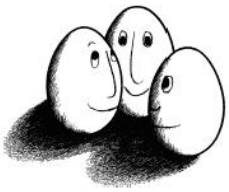
Diplomarbeit

Outlier Detection in USENET Newsgruppen

Stephan Deutsch



Diplomarbeit
am Fachbereich Informatik
der Universität Dortmund

Oktober 2006

Betreuer:

Prof. Dr. Katharina Morik
Dipl.-Inform. Michael Wurst

Inhaltsverzeichnis

Abbildungsverzeichnis

Die im Folgenden aufgelisteten Abbildungen wurden aus den referenzierten Literaturquellen übernommen bzw. anhand deren Quellenangaben nachgebildet. Alle in dieser Tabelle nicht aufgelisteten Abbildungen stammen vom Autor dieser Arbeit (Irrtum/Fehler nicht ausgeschlossen).

Abbildung	**Literaturquelle**
Abbildung 3	[37]
Abbildung 4	[6]
Abbildung 5	[61]
Abbildung 6 – 13	[4]
Abbildung 14 – 17	[5]
Abbildung 18	[68]
Abbildung 19 – 24	[28]
Abbildung 25 – 26	[46]
Abbildung 27	[10]
Abbildung 28	unbekannt (aus altem Vortragsdokument)
Abbildung 29 – 32	Pathlink Technologie Corporation (2005)

Tabellenverzeichnis

Danksagung

Zuallererst möchte ich den Betreuern am Lehrstuhl 8 für „Künstliche Intelligenz“ des Fachbereichs Informatik an der Universität Dortmund, Frau Prof. Dr. Katharina Morik und Dipl.-Inform. Michael Wurst für die Unterstützung und Navigation im noch sehr jungen und durch eine Vielzahl von Meinungen, neuen Erkenntnissen und vor allem ungelösten Problemen gekennzeichneten Feld der Entdeckung von Outliern im Rahmen der Knowledge Discovery in Datenbanken (KDD), herzlich danken.

Gleichsam danke ich für die Unterstützung internationaler Autoren, hier vor allem Zhengyou He, der mir nicht nur neueste Literatur aus der eigenen Forschung, sondern auch Implementierungen experimenteller Algorithmen zur Verfügung stellte.

Mein Dank bei der Erstellung dieser Arbeit gilt jedoch vor allem meiner Ehefrau für ihre besondere Unterstützung in dieser Zeit und auch meinem Sohn Simon, der mir zwar keine fachliche Hilfe war, jedoch eine große Inspiration. Meiner Tochter danke ich dafür, dass sie nach dem dritten Lebensmonat aufhörte, jeden Abend drei Stunden zu schreien, da sich dies sehr positiv auf meine Konzentration ausgewirkt hat.

1. Einleitung

1.1. Knowledge Discovery und Outlier Detection

Der Bereich der Wissensentdeckung (Knowledge Discovery) nimmt im Rahmen der KI-Forschung einen wichtigen Platz ein. Die Aufgaben der Knowledge Discovery sind in vier große Felder geteilt:

- Das Entdecken von Abhängigkeiten
- Die Klassifizierung bzw. das Entdecken von Klassen
- Die Beschreibung von Klassen
- Die Entdeckung von Ausnahmen (sog. Outliern)

Dabei ist das Finden von Strukturen, Mustern und gleichen Eigenschaften, z.B. durch Clustering oder konzeptionelle Generalisierung, meist das Ziel der Forschung. Ausnahmen bilden hingegen nur einen sehr kleinen Prozentsatz der Datenmenge und werden oft entweder ignoriert oder als Rauschen bezeichnet. Daher haben viele existierende Algorithmen und Verfahren des maschinellen Lernens Outlier nur insoweit betrachtet, als dass sie gegenüber diesen Erscheinungen tolerant sind.

Für eine ganze Reihe von Anwendungen sind außergewöhnliche Ereignisse jedoch für die Wissensentdeckung von zentraler Bedeutung. Im Weiteren wird gezeigt, dass Outlier Detection dem Nutzer strategische Vorteile bei der Beurteilung von Situationen geben kann. Dies rechtfertigt eine intensive Auseinandersetzung mit diesem Thema in der vorliegenden Arbeit.

1.2. Sinn und Nutzen von Outlier Detection

„Das Rauschen für den einen ist für den anderen ein Signal.“

Im Rahmen der vielfältigen Betrachtung von Wissensentdeckung in Datenbanken, allgemein auch als KDD – Knowledge Discovery in Databases – bezeichnet, wurden Outlier eine lange Zeit im Bereich des maschinellen Lernens und des Data Mining von existierenden Anwendungen und ihren Algorithmen nur insoweit betrachtet, als dass sie gegenüber diesen Erscheinungen tolerant, bzw. robust waren [3]. Es gibt jedoch eine breite Palette von Anwendungen, für die gerade das Wissen um außergewöhnliche Ereignisse und deren systematische Entdeckung von immenser Bedeutung ist.

Ein wichtiger Aspekt von Outlier Detection ist die Anwendung zur Entdeckung von Anomalien und in ihrer Interpretation die Entscheidung, ob es sich um positive oder negative Abweichungen von dem handelt, was intuitiv als Norm betrachtet oder beschrieben wird. Das wohl namhafteste Beispiel ist die Untersuchung der Transaktionen beim Einsatz von Kreditkarten oder ähnlichen Zahlungsmitteln (z.B. SmartCards) mit dem Ziel, Missbrauch zu identifizieren und erfolgreich zu unterbinden. Die Unterscheidung zwischen normalen und außergewöhnlichen Transaktionsmustern gibt den Kreditkartenfirmen die Möglichkeit, schnell und zielgerichtet einzugreifen und die Kosten von missbräuchlicher Verwendung einzudämmen, Täter ggf. zu identifizieren und trotzdem dem Anwender einen normalen, in Bezug auf diese Aspekte transparenten Zahlungsverkehr zu gewährleisten.

Outlier Detection Anwendungen zur Erkennung von Missbrauch sind jedoch nicht hierauf beschränkt. Die Nutzung von Telefonverbindungen oder Mobilfunkanschlüssen, die Identifizierung der Infiltration von Netzwerken (Intrusion Detection and Prevention), die Analyse von Verkehrsmustern im Internet zur Vermeidung von Denial of Service Attacken (DoS), eCommerce Kriminalität im allgemeinen Sinn, Wahl- und Steuerbetrug (z.B. über die IDEA Software [111] der Prüfer des Finanzamtes), etc. sind alles potentielle und existierende Anwendungen für die Erkennung von Outliern.

Darüber hinaus kann Outlier Detection zu einem strategischen Vorteil durch Wissensgewinn führen. Die Identifizierung von Ausnahmesportlern in diversen Sportarten und ihren Ligen bietet Sportvereinen nicht nur die Möglichkeit, spielerisches Potential zu maximieren. In der Zeit der starken Kommerzialisierung des Sports mit Börsengängen von Fußballvereinen und Sponsoring- und Werbeverträgen von Spitzensportlern, profitiert eine ganze Industrie davon, Wissen um die extraordinären Fähigkeiten von Menschen schnell und effizient zu erwerben.

Die Erkennung abweichender Ereignisse bietet zusätzlich handfeste Vorteile, wenn es um die Vorhersage geht. Extreme Wettersituationen zu erkennen kann genauso überlebenswichtig sein, wie Erkenntnisse über

tektonische Anomalien im geologischen Bereich zu gewinnen, oder außergewöhnliche Zusammenhänge bei der Terrorismusbekämpfung, z.B. durch Rasterfahndung und vergleichbare Methoden richtig zu bewerten.

Auch bei der langfristigen Betrachtung von Systemen, z.B. in der Klima- und Umweltforschung, im Gesundheitswesen oder im Transportwesen spielt Outlier Detection eine immer wichtigere Rolle. Zudem wird sie auch für die neuen Location Based Services eingesetzt.

Dadurch wird deutlich, dass die Outliererkennung einen festen Platz im Rahmen der Wissensentdeckung hat und einen Forschungszweig etabliert, welcher sich fachübergreifend mit den theoretischen Grundlagen und mit praktischen Anwendungen beschäftigt. Er bedient sich dazu den verschiedensten Methoden und Werkzeugen aus der Mathematik (Statistik) und Informatik (theoretische Informatik, Künstliche Intelligenz, etc.) und verbindet diese mit praktischen Anwendungsfeldern. So vielfältig wie die potentiellen Anwendungsmöglichkeiten, so verschieden sind auch die vorgeschlagenen Methoden und Ansätze für die erfolgreiche Identifizierung von Outliern. Insgesamt handelt es sich also um ein junges Wissenschaftsfeld mit täglich neuen Entdeckungen. Dies drückt sich z.B. in einem generellen Fehlen einer formalen, allgemeingültigen und allseits anerkannten Definition dessen aus, was ein Outlier eigentlich ist. Ohne Anspruch auf Vollständigkeit wird im Folgenden versucht, einen Definitionsansatz zu geben, der den Nutzer zumindest befähigt, in der Vielzahl der Forschungsquellen zu navigieren. Denn das Identifizieren von validem, neuem, potentiell sinnvollem und nutzbarem, sowie letztendlich verständlichem Wissen aus Daten ist laut Fayyad und Smyth [52] eine grundlegende Frage der KDD, welche als Problemstellung nicht trivial zu beantworten ist.

1.3. Definition für Outlier

Outlier werden von Barnett und Lewis [33] informell als Beobachtungen definiert, welche zum Rest einer Datenmenge inkonsistent erscheinen. Hawkins [45] definiert Outlier formeller als Beobachtungen, welche so stark von anderen Beobachtungen abweichen, dass dies den Verdacht begründet, ihnen läge ein (gänzlich) andersartiger Mechanismus zugrunde. In den meisten Quellen zu den Themen Outlier und Outlier Detection wird auf diese Definitionen Bezug genommen. Gleichzeitig wird beklagt, dass es keine einheitliche Definition gibt, welche eine genauere Einordnung der vielfältigen Ansätze für Outlier Detection ermöglicht. Auch ist es schwierig, die verschiedenen Ansätze sozusagen rückwärts einzuordnen, indem auf die angewendeten Verfahren Bezug genommen wird. Eine solche Ordnung erlaubt zwar den Vergleich von Ansätzen für Outlier Detection anhand der Art, Kosten und Umsetzung von Algorithmen und in gröberem Rahmen auch eine quasi Ordnung nach den allgemeinen oder statistischen Maßen, welche der Identifizierung zugrunde gelegt werden. Jedoch erscheint die Navigation zwischen den Ansätzen und vor allem die Entscheidung, welches Verfahren für eine Situation konkret am besten geeignet ist, für einen Anwender ohne ausführliches Studium fast aller Ansätze nur sehr schwer möglich zu sein.

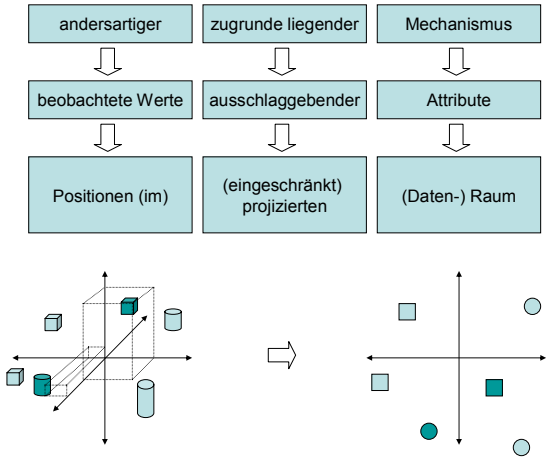


Abbildung 1 - Hawkins Definition von Outliern in Datenmengen

Abbildungsbeschreibung: Die Definition von Outliern nach Hawkins [45] wird grafisch gezeigt und drückt aus, dass sich Mechanismen, welche dem Verhalten von Objekten zugrunde liegen, in der Position dieser Objekte im Datenraum niederschlagen, der durch die Attribute des Objektes aufgespannt wird.

Eine Definition von Outliern sollte demnach vor allem für den Nutzer intuitiv und sinnvoll sein. Sie sollte eine Einordnung und das Finden eines oder mehrerer geeigneter Verfahren möglich machen. Ein geeignetes Verfahren sei hiermit als ein solches bezeichnet, welches Outlier gemäß dieser intuitiven Definition erfolgreich und zu vertretbaren Kosten findet.

Verfeinert man die Definition von Hawkins, indem der Gedanke der Andersartigkeit des zugrunde liegenden Mechanismus konsequent fortgeschrieben wird, so wird deutlich, dass ein solches „Outlier"-Objekt an einem abweichenden ursächlichen Verhalten erkennbar wird. Dieses abweichende Verhalten lässt sich an den beobachteten Werten der ausschlaggebenden Attribute des Objektes ablesen.

Mathematisch bzw. im Rahmen der erfolgreichen Anwendung von statistischen Verfahren äußert sich dies in konkreten Positionen von Objekten in einem eingeschränkt projizierten Raum. Somit lassen sich Objekte also unter Nutzung der globalen Gesamtheit all ihrer möglichen Attribute räumlich ordnen.

Hier wird, noch bevor die verschiedenen Verfahren zur Outliererkennung, welche bereits von der weltweiten Forschungsgemeinschaft im KDD Bereich bereitgestellt werden, zur Anwendung kommen, das größte Dilemma deutlich. Ein Anwender, welcher nach Outliern sucht, kann nach Erkennung von Outliern durch ein Verfahren nur schwer eine Aussage darüber treffen, welche Qualität diese Outlier haben. Noch weniger lässt sich vermuten, warum es sich um Outlier handelt. Allein wenn ein Verfahren die Menge der betrachteten Attribute gezielt oder willkürlich einschränkt (z.B. durch die Anwendung eines Algorithmus, durch fehlende Vollständigkeit von beobachteten Daten, etc.), ergeben sich jeweils verschiedenartige Situationen. Ob in Folge diverse Verfahren gleichartige oder verschiedenartige Outlier identifizieren, lässt sich nur experimentell klären. Inwieweit diese Erkennung dann auf die Systematik der Attributwahl, die Systematik der Verfahrenswahl bzw. überhaupt auf die Konstellation der zu untersuchenden Beobachtungen bezogen werden kann, dafür kann die Forschung derzeit nur Ansätze und erste Überlegungen liefern [55].

Im Rahmen der intensiven Diskussion verschiedener Ansätze bei der Vorstellung einer Auswahl von Verfahren in einem Diplomandenseminar ergab sich die Schlussfolgerung, dass die Antwort auf die Fragestellung nach dem „Warum?" von erkannten Outliern ggf. eine vollständige Lösung des Grundproblems der KDD bedingen würde. Denn maschinelle Lernverfahren zur Erkennung von Outliern müssten die Gründe und Eigenschaften dieser Outlier bei der Identifizierung vollständig beschreiben können, da der Anwender zwar eine Erwartungshaltung hat, jedoch auch ihm unbekanntes Wissen entdecken will, welches sich nicht a priori umfassend beschreiben lässt.

Daher wird im Rahmen dieser Arbeit nicht versucht, Outlier Detection Ansätze zu ordnen bzw. zu klassifizieren. Gleichsam wird im Hinblick auf die Anwendung der USENET Newsgruppen nicht versucht, im Vorfeld eine Einschätzung der Eignung von Verfahren zur Erkennung von Outliern zu treffen. Vielmehr steht in der praktischen Anwendung von unterschiedlichen, gezielt ausgewählten Verfahren – unter der Voraussetzung, diese sind rechentechnisch überhaupt praktikabel – versucht, empirische Ergebnisse vorzuweisen und die erkannten Outlier entsprechend vorzustellen. Dabei steht ein Effizienzvergleich von Verfahren ebenso im Vordergrund, wie die quantitative Menge erkannter Outlier im direkten Verfahrensvergleich. Über die Qualität der erkannten Outlier wird keine Aussage getroffen, da dies mit Sicht auf den Anspruch der Arbeit nicht möglich wäre.

Die hier geforderte ideale Definition existiert somit derzeit noch nicht.

1.4. Ziele und Vorgehensweise dieser Arbeit

1. Die Erstellung eines möglichst umfassenden Überblicks auf bestehende Ansätze zur Erkennung von Outliern.

   Dies wird im ersten Teil der Arbeit durch eine Einführung in die generelle Problematik der Outliererkennung erreicht und für die Verfahren wird eine detaillierte Darstellung der ihnen jeweils zugrunde liegenden Ansätze vorgenommen. Dabei werden zur Begrenzung des Umfangs nicht alle Verfahren vorgestellt. Durch die Aktualität des Forschungsgebietes werden zudem jedes Jahr neue Verfahren eingeführt. Es wird jedoch eine möglichst umfängliche Liste als Abschluss angeboten, welche über entsprechende Literaturverweise eine Vertiefung der Materie erlaubt.

2. Die Einführung in die Anwendungsdomaine der USENET Newsgruppen

   a. Hierbei werden die der Benutzung zugrunde liegenden wesentlichen Mechanismen durch eine Betrachtung des Systems und seiner technischen Funktion, sowie durch die auf den entsprechenden Standards basierende Analyse der Struktur von Newsartikeln und des Anwenderverhaltens vorgestellt.

   b. Im zweiten Schritt wird eine Anforderungsanalyse für die Erkennung von Outliern durch die Beschreibung von Objektkategorien, welche als Outlier erwartet werden könnten, vorgenommen.

   c. Auf Basis der Anforderungen und Kategorien wird ein umfänglicher Benchmark-Datensatz erstellt.

3. Die Implementierung von Outlier Detection Verfahren zum Zweck ihrer Evaluation

   a. Dafür werden ausgewählte Standardverfahren zur Erkennung von Outliern implementiert.

   b. Es werden speziell auf das Problem hin angepasste Vorverarbeitungsverfahren zur Vektorisierung der Testdatenmengen umgesetzt.

   c. Durch die Entwicklung von Zusatzverfahren im Vorverarbeitungsschritt soll untersucht werden, ob diese die Ergebnisse der Standardverfahren entscheidend verbessern können.

4. Die Durchführung und Auswertung von Experimenten

   a. Mittels des Benchmark-Datensatzes und zusätzlicher Datensätze werden die implementierten Verfahren in einer Testumgebung evaluiert.

   b. Im Abschluss der Arbeit wird eine Interpretation der Ergebnisse der Experimente durchgeführt.

Abbildung 2 - Ziele der Diplomarbeit

Diese Arbeit konzentriert sich im theoretischen Teil auf die Betrachtung von Outlier Detection Ansätzen. Dabei wird eine Auswahl von Verfahren eingeführt, wobei durch die rasant fortschreitende Entwicklung kein Anspruch auf Vollständigkeit bestehen kann, da pro Jahr mit Sicherheit mindestens fünf bis zehn echte neue Ansätze mit entsprechenden Verfahren und neuen Algorithmen publiziert werden. Auch würde die ausführliche Listung und Beschreibung aller bekannten Verfahren den Umfang dieser Arbeit sprengen.

Die Abgrenzung der Verfahren wird basierend auf den Aussagen der jeweiligen Autoren der Ansätze vorgenommen. Es ist nicht das Ziel dieser Arbeit, Ansätze systematisch zu ordnen oder bzgl. der generellen Qualität oder Aussagekraft der Ergebnisse der Ansätze verbindliche Aussagen zu machen (z.B. in Form eines Schemas oder Systems). Trotzdem gibt dieser Teil der Arbeit einen guten Überblick über mögliche Verfahren und ihre KDD Grundlagen. Zusätzlich werden Algorithmen und deren Anwendbarkeit auf verschiedene Sachgebiete bzw. Situationen vorgestellt, wobei die Art der Erkennung und die Komplexität der Verfahren in Bezug auf die Rechenzeit eine wichtige Rolle spielt. Durch ausführliche Referenzen der Literaturquellen wird eine weitergehende Betrachtung des Themas durch den interessierten Leser ermöglicht.

Im praktischen Teil der vorliegenden Diplomarbeit wird die Anwendung der Outliererkennung auf ein konkretes Sachgebiet vorgestellt. Dazu werden die USENET Newsgruppen herangezogen. Neben einer Einführung in das USENET Thema wird vor allem auf die zu erwartenden Ergebnisse von Outliererkennungen aus Sicht des Anwenders abgestellt, um eine Einschätzung der Ergebnisse der Anwendung ausgewählter Outlier-Detection Ansätze zu erlauben. Dabei wird auf die speziellen Anwendungsumstände für Newsgruppen hingewiesen, um sowohl die Wahl von geeigneten Verfahren (jedoch nicht in Bezug auf eine systematische Ordnung) als auch deren algorithmische Implementierung zu begründen. Diese Verfahrensauswahl wird in einem experimentellen Set-Up implementiert. Darauf basierend wird eine Reihe von Experimenten umgesetzt, damit empirische Erkenntnisse gewonnen werden können.

Um eine möglichst breite Weiterverwendung der Verfahren zu ermöglichen, setzt die Implementierung auf die Plattform des Systems YALE des Lehrstuhls für Künstliche Intelligenz des Fachbereichs Informatik an der Universität Dortmund auf. Gleichsam werden auch Schnittstellen für die sog. ESOM Tools des Lehrstuhls für Datenbionik am Fachbereich Mathematik der Universität Marburg bereitgestellt, um die Experimente zwischen beiden Systemplattformen zu verbinden. Aufgrund der internationalen Verbreitung beider Systeme ist damit ein Zusatznutzen der praktischen Ergebnisse dieser Arbeit gegeben. Interessierte Leser sind eingeladen, die unter der GNU Public License (GPL) erstellten Implementationen des „Outlier Plugin“ für YALE zu nutzen und ggf. selbst weiterzuentwickeln.

2. Existierende Outlier Detection Ansätze

„Quot capitum vivunt, totidem studiorum milia“ (HORAZ)

In diesem Kapitel werden verschiedene Outliererkennungsverfahren in unterschiedlicher Detailtiefe vorgestellt. Diese unterschiedliche Tiefe ergibt sich aus der Unterstützung der gesamtheitlichen Betrachtung des Themas und aus der notwendigen Beschreibung von Details der Verfahren, welche im praktischen Teil eine konkrete Anwendung finden.

Die verschiedenen Quellen, welche im Rahmen der Beschreibung des von den Autoren jeweils vertretenen Ansatzes für Outlier Detection natürlich auf vorangegangene und vor allem vom eigenen Ansatz abweichende Methoden eingehen, stellen Vergleiche der Verfahren an. Die Nennung der Verfahren und deren Beschreibung stellt hier jedoch keine Einordnung, Kategorisierung oder Priorisierung durch den Autor dieser vorliegenden Arbeit dar. Im Allgemeinen ergibt sich im Forschungsfeld der Outlier-Detection Ansätze der Trend, Verfahren in einer jeweiligen Familie gleicher oder verwandter Ansätze zu verallgemeinern um ein besseres Gefühl für die Anwendbarkeit und eine Basis für allgemein effizientere Umsetzungen in Algorithmen zu bekommen.

Dementsprechend reflektiert die im Kapitel vorgenommene Unterteilung zum einen die Herkunft des Ansatzes, als auch die „Entwicklung“ hin zur Bearbeitung spezieller Probleme bzw. in anderen Fällen die Unifizierung von Verfahren. Stück für Stück werden neue Ideen hinzugenommen, sodass in Teilen ein historisierter Abriß entsteht. Es wäre aber auch eine andere Gliederung unter abweichenden Gesichtspunkten genauso gut möglich.

Die Beschreibungen und Beweise wurden möglichst unverändert aus den Veröffentlichungen der Autoren übernommen und um Anmerkungen anderer Autoren angereichert. Hierbei sei ausdrücklich auf die originalen Quellen verwiesen.

2.1. Generelle Definitionen und Begriffe

Die Autoren der verschiedenen Verfahren setzen eine Reihe unterschiedlicher Begriffe und vor allem Variablendefinitionen ein, mit denen sie ihre Ansätze formal beschreiben. Um eine Vergleichbarkeit der Ansätze zu erleichtern und auch das Verständnis zu fördern, wurden die formalen Definitionen weitgehend vereinheitlicht und weichen daher von den Literaturquellen entsprechend ab.

Definition der Begriffe Datenraum, Datenpunkt, Objekt, Attribut und Distanz sowie Distanzfunktion:

Sei eine Menge X von Datenpunkten (oder im Folgenden auch Punkten bzw. Objekten oder Elementen) gegeben mit $X = \{x_i \mid i = 1,...n_X\}$ und $X \subseteq R^m$ eine echte Teilmenge des Datenraumes R^m mit $n_X = |X|$. Sei n eine Anzahl von Datenpunkten bzw. Objekten und bezeichne ggf. sowohl die Kardinalität der Menge X mit $n = n_X$ oder auch die Kardinalität einer Untermenge von X mit $n \le n_X$.

Sei ferner $\vec{x}$ der *m*-dimensionale Vektor (im Folgenden auch *m*-dimensionales Tupel von Koordinaten), welcher die Position des Datenpunktes x im Datenraum R^m beschreibt, so sei die Attributmenge $A = \{a_j \mid j = 1,...m\}$ des Datenpunktes durch die m Achsen des Datenraumes R^m beschrieben und der Wert des Attributes a_j gleich dem Wert der *j*-ten Koordinate von $\vec{x}$. Die Distanzfunktion zwischen zwei Datenpunkten sei durch $d : X \times X \to R_o^+$ gegeben und die Distanz zwischen zwei verschiedenen Objekten $x, x' \in X$ bezeichnet mit $D = d(x, x')$.

Definition der Begriffe Nachbarschaft und Cluster:

Der Nachbarschaftsbegriff wird von verschiedenen Ansätzen unterschiedlich definiert. Es handelt sich jedoch fast durchgängig um eine Teilmenge $N \subseteq X$, wobei die Besetzung dieser Menge durch eine Abhängigkeit von einem oder mehren Objekten bzw. Datenpunkten bestimmt ist, z.B. $N_k(x) = \{x' \in X \wedge d(x, x') < D_k\}$. Ein Cluster sei durchgängig bezeichnet mit $C \subseteq X$, wobei sich die Zugehörigkeit von Objekten zu einem Cluster durch die Definition der Eigenschaften eines Clusters a.a.O. ergibt.

Definition zusätzlicher Begriffe:

Eine Reihe von Ansätzen führt zusätzliche Begriffsdefinitionen ein und soweit diese spezifisch für den Ansatz sind, werden sie in den folgenden Abschnitten entsprechend definiert.

2.2. Verteilungsbasierte Ansätze

Im Bereich der frühen Auseinandersetzung mit dem Thema Outlier Detection werden eine Vielzahl an Testverfahren vorgestellt, welche sich auf verschiedene statistische Standardverteilungen bzw. Normalverteilungen stützen.

> *Anmerkung:* He, Deng und Xu gruppieren in Ihren Veröffentlichungen [10] mit Bezug auf erste Studien zur Identifizierung von Outliern seitens Barnett und Lewis [33] verteilungsbasierte Tests als eine von zwei Kategorien von statistischen Tests für Outlier Detection. Die zweite Kategorie umfasst in dieser Veröffentlichung die tiefenbasierten Tests. Motivation für diese Art von Kategorisierung ist der Fokus auf starke statistische Maße, welche einen direkten mathematischen Bezug in den Vordergrund stellen. Andere Tests werden eher anhand der intuitiven Idee und der daraus folgenden Definition eines Outliers sowie der algorithmischen Umsetzung oder unter Bezug auf das Verfahren geordnet. Interessant ist in diesem Zusammenhang die Beobachtung, dass trotzdem all diesen Tests in der Regel, d.h. bis auf wenige Ausnahmen, ein oder mehrere spezielle, manchmal auch frei wählbare, statistische Maße zugrunde liegen.

Outlier werden demgemäß in diesen verteilungsbasierten statistischen Ansätzen auch verteilungsabhängig definiert. Die Verteilung wird zur Darstellung normalen Verhaltens der zu beobachtenden Objekte herangezogen. Da für jede Normalverteilung eine Reihe von statistischen Maßen existiert, um Objekte im Rahmen dieser Verteilung zu beschreiben (Mittelwerte bzw. Erwartungswerte und deren Wahrscheinlichkeiten, Varianzen und Standardabweichungen), können Outlier anhand dieser Maße beschrieben und erkannt werden.

Barnett und Lewis stellen mehr als 100 Tests für diverse Verteilungen vor, darunter für $N(\mu;\sigma^2)$-Normalverteilungen, exponentielle Verteilungen, Gamma Verteilungen, Poisson Verteilungen und binomiale Verteilungen. Die Wahl des Tests basiert nach Knorr und Ng [6] unter anderem auf der Verteilung selbst, da es Sinnvollerweise verschiedene optimierte Tests für verschiedene Verteilungen gibt. Wichtig für die Auswahl des Verfahrens ist zudem, ob verteilungsspezifische Parameter, wie z.B. der Erwartungswert oder die Varianz oder beide Größen zusammen, bekannt sind. Auch die Anzahl der erwarteten Outlier und die Typen der erwarteten Outlier, also ob einzelne, Paare, oder eine Anzahl *n* von Outliern erwartet werden, ist entscheidend. Ebenso fließt ein, wo diese Outlier erwartet werden, z.B. im oberen, unteren oder im oberen und unteren Bereich der Verteilung. Allerdings gibt es keine Garantie dafür, dass Outlier auch tatsächlich gefunden werden. Dafür gibt es verschiedene Gründe. Möglicherweise wurde gerade für die vorliegende Verteilung kein Test entwickelt. Oder es gibt keine Standardverteilung, welche die tatsächliche Verteilung der vorliegenden Testmenge an beobachteten Objekten adäquat abbildet.

Die große Zahl der statistischen verteilungsbasierten Testverfahren ist univariat und untersucht nur ein einzelnes Attribut. Dies stellt insbesondere bei der Betrachtung von multivariaten Datenmengen ein Problem dar. Zwar kann unter der naiven Annahme grundsätzlich unabhängiger Attribute in einer mehrdimensionalen Datenmenge eine Reihe von solchen univariaten Verfahren pro Attribut angewendet werden. Im Gesamtergebnis würde dann eine geeignete Zusammenführung der unterschiedlichen Einzelergebnisse angestrebt. Wie diese sinnvoll stattzufinden hat und ob eine Abhängigkeit der Attribute überhaupt ausgeschlossen werden kann, ist jedoch fraglich. Daher ist ein solcher Ansatz in der Praxis sicher nicht effizient durchsetzbar.

Über reine statistische Tests hinaus sind im Rahmen von KI Betrachtungen von Yamanishi, Takeuschi und Williams ([34], [35]) weitere Verfahren vorgeschlagen worden, welche statistische Modelle und deren Untersuchung mit überwachten Lernverfahren kombinieren, um generelle Muster für Outlier zu finden.

Allgemein gehen alle Verfahren davon aus, dass zumindest die Verteilung der Objekte beim Ansatz des für eben diese Verteilung geeigneten Verfahrens im Voraus bekannt ist. Dies ist für eine große Menge an Situationen nur schwer intuitiv anzunehmen und stellt die Praktikabilität der verteilungsbasierten statistischen Tests und darauf basierender weitergehender Outlier Detection Methoden in Frage. Trotzdem sollten statistische verteilungsbasierte Methoden nicht grundsätzlich negativ bewertet werden. Denn zum einen spielen statistische Maße auch bei der überwiegenden Zahl anderer Verfahren eine zentrale Rolle. Zum anderen ist eine Vielzahl von Tests für die unterschiedlichsten Verteilungen bekannt. Daher kann in dem Fall, dass eine vorhandene Testmenge nicht a priori in ihrer Verteilung bekannt ist, bzw. einer Standardverteilung nicht entspricht, diese Testmenge durch Berechnungen in eine Menge mit entsprechender Verteilung umgewandelt werden. Leider sind die Kosten einer solchen Übertragung bzw. Anpassung signifikant und können nicht vernachlässigt werden. Auch ist die Frage zu beantworteten, inwieweit eine Anpassung die Charakteristika der gesuchten Outlier so verändert, dass diese nur noch schwer oder gar nicht mehr identifiziert werden können, und ob sogar Objekte, welche vor einer Umwandlung nicht als Outlier in Frage kamen, nun als solche leicht zu erkennen sind. Da die Definition, was ein Outlier in einem konkreten Fall sein soll, nicht einheitlich ist, sei dies eine Anregung für

weitergehende Überlegungen. Diese sind mit dem Gedanken verbunden, dass sich die statistischen Verfahren ggf. deswegen nicht vorrangig als geeignet erweisen, in praktischen Situationen Outlier erfolgreich bzw. kostengünstig zu identifizieren, weil sie von einer zu starren Definition eines Outliers ausgehen, welche sich immer auf die zugrunde liegende Verteilung bezieht, die das Normalverhalten darstellt.

Verfahren, welche verteilungsbasierte Outlier Detection Ansätze generalisieren, könnten genau deshalb erfolgreicher sein, weil sie die starren Grenzen der statistischen Tests aufweichen und flexibilisieren. Derartige Verfahren werden in diesem Kapitel vorgestellt. Dem stünde allerdings argumentativ entgegen, dass gerade auch die statistisch basierten, verteilungsorientierten Testverfahren eine gewisse Flexibilität erlauben, wenn nicht sogar erfordern, da sich ein spezifisches Verfahren einsetzen lässt, welches gewissen Parametern der Outlier, z.B. im Hinblick auf deren Anzahl, Verteilungsparameter, Typen etc., entspricht. Hier muss jedoch zwischen der Erwartungshaltung an Outlier und der Auswahl eines konkreten Verfahrens unterschieden werden. Letztere erfordert bei den verteilungsbasierten Verfahren detaillierte Kenntnisse über den tatsächlichen statistischen Charakter dessen, was als Outlier gesucht wird. Dieses Wissen ist in der Regel jedoch nicht gegeben. Vielmehr ist davon auszugehen, dass der Anwender gar nicht weiß, was er als Outlier sucht. Daraus ergibt sich ein Folgeproblem, weil der Anwender demgemäß nicht selbst entscheiden kann, welche Qualität Outlier haben, die von einem jeweils vorgeschlagenen Verfahren entdeckt werden. Auf diese Fragestellung bietet die KDD Forschung derzeit noch keine vollständige Antwort und daher wird sich die vorliegende Arbeit auch nicht mit der Lösung dieses Problems befassen können.

2.3. Tiefenbasierte Ansätze

Tiefenbasierte Testverfahren zur Identifizierung von Outliern, z.B. vorgestellt von Ruts und Rousseeuw [36], organisieren die zu prüfenden Objekte im Datenraum anhand einer Tiefendefinition. Basierend auf dieser Definition einer Tiefe werden die Objekte in konvexen Hüllen-Ebenen oder anhand ihrer Schältiefe geordnet. Outlier werden unter den Objekten mit geringem Tiefenwert bzw. in Ebenen mit geringen Tiefenwerten erwartet. Diese Tests wurden entwickelt, um der bei den verteilungsbasierten Tests erforderlichen Bestimmung der Verteilung – welche i.d.R. unbekannt ist – zu entgehen.

Peeling bzw. die Schältiefe ist ein Tiefenbegriff, der ausführlicher von Preparata [38] vorgestellt wird. Dieser Ansatz leidet jedoch darunter, dass er sich zu schnell in Regionen mit einer hohen Punktdichte bewegt und daher nicht so robust wie der Ansatz der Tiefenkonturen mit Halbraumtiefen von Ruts und Rousseeuw ist.

Der Begriff der Halbraumtiefe eines Punktes relativ zu einer multivariaten Datenmenge wurde 1975 von Tukey [39] eingeführt. Im univariaten Fall wird die Tiefe eines Punktes x' relativ zu einer eindimensionalen Menge $X = \{x_1,\ldots,x_n\}$ als das Minimum der Anzahl der Punkte links und rechts von x' definiert:

$$depth_1(x') = \min(|\{i; x_i \le x'\}|, |\{i; x_i \ge x'\}|)$$

Die Halbraumtiefe eines Punktes $x' \in R^m$ relativ zu einer *m*-dimensionalen Datenmenge $X = \{x_1,\ldots,x_n\}$ wird als die geringste Tiefe von x' in jeder eindimensionalen Projektion der Datenmenge definiert und kann auch als die minimale Zahl an Datenpunkten in einem geschlossenen Halbraum gesehen werden, dessen Randebene x' passiert. Es gibt noch weitere Tiefendefinitionen, welche von Small, Niinimaa und Tukey eingeführt werden und für eine weitergehende Betrachtung der Unterschiede zur hier verwendeten Definition sei auf die entsprechenden Quellen ([40], [41] und [42]) verwiesen.

Die Halbraumtiefe ist affin invariant, d.h. wenn x' und X linear transformiert werden, ändert sie sich nicht. Dies impliziert, dass das Konzept der Halbraumtiefe unabhängig vom gewählten Koordinatensystem ist und sich daher in vielfältiger Weise einsetzen lässt. Diese Eigenschaft wird von Donoho und Gasko [43] in zwei Papieren ausführlicher diskutiert.

Die Tiefe steht in enger Beziehung zum Rang. Dies ist besonders deutlich im univariaten Fall zu sehen. Wenn die Datenpunkte einer Dimension mit Rang versehen werden, so erhalten die extremen Punkte mit dem niedrigsten und dem höchsten Rang die Tiefe 1. Datenwerte mit dem nächstniedrigsten und nächsthöchsten Rang erhalten die Tiefe 2, usw. Der Median ist folglich der Punkt mit der maximalen Tiefe.

Der Median ist ein empirisches Lagemaß in der Statistik und wird auch Zentralwert genannt. 50% der Werte einer nach Größe geordneten Menge $x_{(1)},\ldots,x_{(n)}$ sind größer oder gleich und 50% der Werte sind kleiner oder gleich dem Wert des Median. Somit errechnet sich der Median $\tilde{x}_{0,5} = x_{((n+1)/2)}$ falls n ungerade; und $\tilde{x}_{0,5} = \frac{1}{2}(x_{(n/2)} + x_{((n+2)/2)})$ falls n gerade ist.

In höherdimensionalen Fällen gibt die Tiefe eines Punktes einen Eindruck davon, wie „tief“ sich der Punkt in der Datenwolke befindet. Ein Punkt mit maximaler Tiefe kann als multidimensionaler Median interpretiert werden.

Wichtig zur Abgrenzung des Ansatzes ist, dass das Maß der Tiefe nicht äquivalent zum Maß der Dichte ist. Während die Tiefe eines Punktes x' ein globaler Begriff ist, da sie von der Gesamtheit der Datenmenge X abhängt, ist die Dichte von x' lokal in ihrer Natur, da sie nur von den Punkten von X abhängt, welche sich in einer Nachbarschaft von x' befinden. Dichtebasierte Ansätze zur Identifikation von Outliern werden separat in diesem Kapitel eingeführt.

Von Ruts und Rousseeuw werden sogenannte Tiefenkonturen zur Berechnung eingeführt. Sei $X \subset R^m$ eine *m*-dimensionale Datenmenge. Sei die Menge $X_k = \{x \in R^m \mid depth(x; X) \geq k\}$. Die inneren Punkte von X_k haben mindestens die Tiefe k und die Randpunkte von X_k haben eine Tiefe gleich k. Damit ist X_k die Kontur der Tiefe k, wenn auch eine strengere Auslegung dieser Begrifflichkeit auf den Rand von X_k beschränkt ist. Da X_k der Schnitt aller der Halbräume ist, welche mindestens $n+1-k$ Punkte der Wolke enthalten, ist X_k konvex. Die verschiedenen Tiefenkonturen formen eine verschachtelte Reihe, weil X_{k+1} in X_k enthalten ist. Die äußerste Kontur X_1 ist die normale konvexe Hülle von X. Punkte außerhalb dieser konvexen Hülle der Datenmenge haben die Tiefe Null. Die Anzahl der Tiefenkonturen einer gegebenen Menge X und damit deren maximale Tiefe hängen von der Form von X ab. Ist sie nahezu symmetrisch, kann es bis zu $\lceil n/2 \rceil$ Tiefenkonturen geben. Ist sie jedoch sehr asymmetrisch, werden es aller Wahrscheinlichkeit nach sehr viel weniger Tiefenkonturen sein. Abbildung 3 zeigt die ersten 10 Tiefenkonturen einer Datenwolke mit 5000 Punkten als Beispiel.

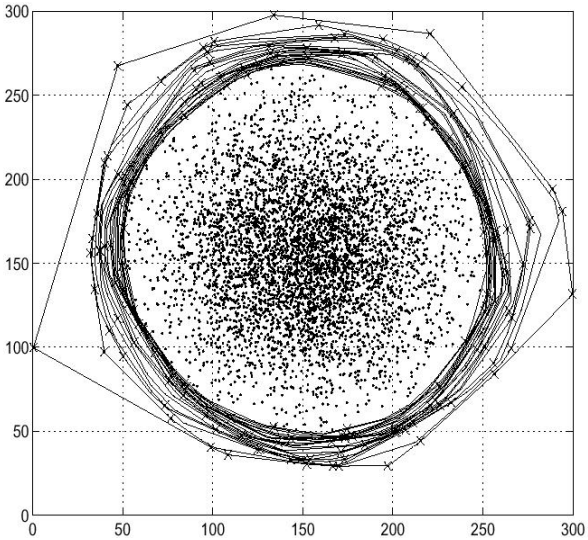


Abbildung 3 - Tiefenkonturen einer zweidimensionalen Datenwolke mit 5000 Punkten

Abbildungsbeschreibung: Eine Datenwolke mit 5000 Punkten in einem zweidimensionalen Datenraum mit einer Gauß-Verteilung wird dargestellt, wobei die unterschiedlichen Tiefenkonturen [36] durch verbundene Linien dargestellt sind.

Zur Berechnung der Konturen werden mehrere Algorithmen vorgeschlagen, wobei sich Ruts und Rousseeuw auf bivariate Datenmengen konzentrieren. Sie bieten einen „naiven“ Algorithmus mit $O(n^5 \log n)$ und einen ISODEPTH Algorithmus mit ca. $O(n^2 \log n)$ an. Johnson, Kwok und Ng [37] entwickelten basierend auf der Idee von ISODEPTH einen schnelleren und robusteren Algorithmus zur Berechnung von zweidimensionalen Tiefenkonturen. Dieser FDC Algorithmus konzentriert sich zur Berechnung der ersten k Tiefenkonturen auf eine kleine, ausgewählte Teilmenge an Datenpunkten, anstatt die gesamte Datenmenge zu evaluieren. Diese Teilmenge wird durch die Berechnung der entsprechenden konvexen Hüllen identifiziert. Da für die Identifizierung von Outliern i.d.R. nur die ersten ≤ 100 Tiefenkonturen interessant sind, ist die erwartete Performance von FDC sehr viel besser, als die von ISODEPTH. Generell ist der Aufwand mit $O(n \log n + h \log^2 n + k'h^3)$ angegeben, wobei *n* die Anzahl der Datenpunkte, *k'* die Zahl der Tiefenkonturen und *h* die maximale Kardinalität der ersten *k'* Elemente in der Serie von Tiefenkonturen ist. Eine nähere Ausführung

und experimentelle Ergebnisse der Performance von FDC geben die Autoren des Algorithmus im Rahmen ihrer Veröffentlichung an.

Theoretisch sind alle diese Ansätze lt. He, Deng und Xu (et al) auch für hochdimensionale Datenräume geeignet. Da sich die geschätzten Rechenkosten jedoch in der Praxis durch den Einsatz von konvexen Hüllen ergeben, eine Technik die einen unteren Grenzaufwand von $O(n^{\frac{m}{2}})$ hat, wobei *n* die Anzahl der Objekte und *m* die Anzahl der Dimensionen darstellt, ist dieser Ansatz unter praktischen Gesichtspunkten auf Datenmengen mit vielen Dimensionen nicht uneingeschränkt anwendbar. Untersuchungen wie die von Rousseeuw haben gezeigt, dass in der Praxis die Performance nur für Datenmengen mit einer Zahl von $m \leq 2$ Dimensionen akzeptabel ist. Solche effizienten zweidimensionalen Verfahren werden z.B. von Johnson, Kwok und Ng [37] vorgeschlagen und einige Peeling-Verfahren für Datenmengen mit $m = 3$ werden von Preparata und Shamos [38] vorgestellt.

2.4. Entfernungsbasierte Outlier

2.4.1. Unifizierende entfernungsbasierte Ansätze

Knorr und Ng [3] stellen in verschiedenen Papieren ([6], [7] und [8]), unter anderem gemeinsam mit Tucakov, ausführlich einen Ansatz für Outlier Detection vor, der in Bezug auf das angewendete statistische Maß entfernungsbasiert ist. Sie motivieren ihren Ansatz mit den in den vorhergegangenen Abschnitten beschriebenen Unzulänglichkeiten der verteilungs- bzw. auch tiefenbasierten Ansätze. Diese sind zum einen von der Auslegung der Tests meist auf univariate Datenmengen bezogen. Solche Tests sind für mehrdimensionale Anwendungen schlicht ungeeignet. Zum anderen sind bei verteilungsbasierten Verfahren die Verteilungen innerhalb der Datenmenge nicht a priori bekannt und es sind intensive Tests nötig, um diese Verteilungen zu identifizieren. Sofern eine beobachtete Verteilung gar keiner Verteilung entspricht, für die ein Test existiert, ist es sehr aufwändig, diese entsprechend umzurechnen. Tiefenbasierte Tests umgehen die Notwendigkeit, die Verteilung zu kennen oder eine bekannte Verteilung aus der beobachteten Datenmenge zu erzeugen. Auch sind sie vom Ansatz her prinzipiell für multivariate Anwendungen einsetzbar.

Die Idee, welche den Überlegungen von Knorr und Ng zugrunde liegt, ist die Einführung eines Outlier Begriffs, der die verteilungsbasierten Begriffe unifiziert und gleichzeitig Algorithmen liefert, welche für mehrdimensionale Fälle einfach und kosteneffizient einsetzbar sind. Dabei sind der statistische Ansatz und das eingesetzte Maß mit den verteilungs- und tiefenbasierten Verfahren vergleichbar. Auch dort werden statistische Entfernungsmaße verwendet, um die Objekte anhand einer Verteilung oder anhand der Einordnung in eine gewisse Tiefe, welche eben anhand eines statistischen Entfernungsmaßes errechnet wird, miteinander zu vergleichen und Outlier entsprechend zu identifizieren.

Definition von entfernungsbasierten (distance based) Outliern:

Ein Objekt *x* in einer Datenmenge *X* ist ein *DB(p,D)*-Outlier, wenn zumindest ein Anteil *p* von Objekten in *X* weiter als die Entfernung *D* von *x* liegt. Ein *DB(p,D)*-Outlier wird also anhand der Parameter *p* und *D* erkannt. Dieser intuitive Outlier Begriff steht mit der Definition nach Hawkins [45] im Einklang und eignet sich auch, aber nicht ausschließlich, für solche Fälle, in denen die beobachtete Verteilung nicht mit einer Standardverteilung übereinstimmt. Die Definition eignet sich für multivariate Anwendungen mit einer unbegrenzten Anzahl *m* an Dimensionen. Knorr und Ng führen aus, dass die Berechnung der Entfernung bei *DB(p,D)*-Outliern auf Basis einer metrischen Distanzfunktion vorgenommen wird, wobei die von den Autoren vorgestellten Algorithmen davon ausgehen, dass diese Funktion euklidisch ist. Obwohl kein Anspruch geltend gemacht wird, dass die entfernungsbasierten Outlier alle anderen Outlier Begriffe ersetzen würden oder universellen Charakter hätten, bezeichnen Knorr und Ng sie trotzdem auch als „unifizierte Outlier“ oder sog. *UO(p,D)*-Outlier. Demgemäß sind *DB(p,D)*-Outlier und *UO(p,D)*-Outlier synonym zu verstehen.

> *Anmerkung:* Da Knorr und Ng den Begriff erst als unifizierte Outlier (*UO(p,D)*) einführten und in der detaillierteren Ausführung der Algorithmen den Begriff *DB(p,D)*-Outlier prägten, sei hier der spätere Begriff, also *DB(p,D)*-Outlier angenommen. Für den interessierten Leser sei noch angefügt, dass das Papier [6], *A Unified Approach for Mining Outliers*, eine erweiterte Version des Papiers [7], *A unified notion of outliers: Properties and computation*, ist (siehe Literaturverzeichnis).

Die Analyse zeigt, dass der *DB(p,D)*-Outlier Begriff die Outlier Begriffe der verteilungsbasierten Tests insofern generalisiert, als dass es für ein Objekt *x*, welches nach einem verteilungsbasierten Test ein Outlier ist, auch eine passende Kombination der Parameter *p* und *D* gibt, sodass *x* auch ein *DB(p,D)*-Outlier ist. Somit sind diverse verteilungsbasierte Outlier dann Instanzen von *DB(p,D)*-Outliern.

Definition der Unifizierung von Outliern durch *DB(p,D)*-Outlier: Ein *DB(p,D)*-Outlier (bzw. ein *UO(p,D)*-Outlier) unifiziert eine andere Outlier Begriffsdefinition „Def“ genau dann, wenn es eine spezifische Wertekombination p_0, D_0 gibt, mit der gilt, dass wenn x ein Outlier nach „Def“ ist, x auch ein $DB(p_0, D_0)$ – Outlier ist und dies für alle $x \in X$ gilt.

Definition der Parameter *p* und *D*: Sei n eine Anzahl von Objekten in einer Test-Datenmenge X. Jedes Objekt x wird mit denselben m Attributen identifiziert, m ist also die Dimensionalität der Menge X. Angenommen, es existiert eine zugrundeliegende metrische Funktion d, welche die Distanz zwischen jedem möglichen Paar von Objekten in X liefert, dann gilt:

1. Für ein Objekt x enthält die D-Nachbarschaft N_D von x die Menge an Objekten $x' \in X$, welche sich maximal in der Entfernung D von x befinden, also $N_D(x) = \{x' \in X \mid d(x, x') \leq D\}$.
2. Der Anteil p ist der minimale Anteil von Objekten in X, welche sich außerhalb der D-Nachbarschaft eines Outliers befinden müssen.

Anhand von zwei Beispielen, dem statistischen verteilungsbasierten Test zu einer Normalverteilung und der Erkennung von Outliern in Regressionsmodellen, soll deutlich werden, wie diese Unifizierung belegt ist, bevor der eigentliche Ansatz zur Berechnung ausgeführt wird [8].

Outlier in einer Normalverteilung sind solche Punkte, welche mehr als die dreifache Standardabweichung $(\geq 3\sigma)$ vom Erwartungswert μ entfernt liegen, vgl. u.a. Freedman, Pisani und Purves [47]. X sei also eine Datenmenge, deren Objekte wirklich normalverteilt sind mit $N(\mu; \sigma^2)$. Damit sei Def_{Normal} wie folgt definiert: $x \in X$ ist ein Outlier, wenn

$$\frac{x - \mu}{\sigma} \geq 3 \text{ oder } \frac{x - \mu}{\sigma} \leq -3 \text{ ist.}$$

Lemma der Unifizierung von Outliern in Normalverteilungen: ein *DB(p,D)*-Outlier unifiziert Def_{Normal} mit $p_0 = 0{,}9988$ und $D_0 = 0{,}13\sigma$. Der Beweis dazu wird in [6] geführt.

Als weiters Beispiel sei die Identifizierung von Outliern in Regressionsmodellen beschrieben. Ein einfaches lineares Regressionsmodell ist durch die Gleichung $y = \alpha + \beta x$ gegeben. Die Datenmenge X enthält Beobachtungen der Form (x_i, y_i) für $i = 1,...n$, welche in dieses Modell eingepasst werden. Eine Möglichkeit zur Erkennung von Outliern ist die Betrachtung der Residuen, d.h. der Unterschiede zwischen beobachteten und eingepassten Werten. Der residuale Fehler der i-ten Beobachtung wird durch e_i in $y_i = \alpha + \beta x_i + e_i$ ausgedrückt. Outlier werden dann als solche Residuen erkannt, die weitaus höhere Unterschiede aufweisen, als die meisten, z.B. wenn sie mehr als 3 Standardabweichungen vom mittleren Erwartungswert der Residuen abweichen (vgl. auch Draper und Smith [49]). Unter der vereinfachenden Annahme, dass die Residuen voneinander unabhängig und damit normalverteilt sind, lassen sich diese Outlier nach Def_{Normal} identifizieren und folglich unifiziert $DB(p_0, D_0)$ auch hier. Der multivariate Fall liegt wesentlich komplexer. Im Allgemeinen ist es schwer, einen verteilungsbasierten Unterscheidungstest für multivariate Regressionsmodelle zu finden. Daher wird der Ansatz der robusten Regressionstechniken nach Rousseeuw und Leroy [50] verwendet. Dieser Methode nach wird die Regressionsgleichung für den mehrheitlichen Anteil der Daten berechnet und Outlier werden als die Punkte erkannt, welche große Residuen gegenüber der robusten Gleichung haben.

Im generellen Regressionsmodell, in dem k Parameter von n Beobachtungen geschätzt werden, sind die Residuen nicht als voneinander unabhängig zu betrachten. Die n Residuen sind lediglich mit $n - k$ Freiheitsgraden assoziiert. Sei zum Beispiel X eine Menge mit $n = 150$ Beobachtungen für $k = 10$ Parameter, welche in ein Regressionsmodell der Form $y = \alpha + \beta_1 x_1 + \cdots + \beta_{10} x_{10}$ eingepasst sind, und seien die Residuen für die i-te Beobachtung bezeichnet als e_i. Dann ist $Def_{Regression}$ definiert durch: $x_i \in X$ ist ein Outlier, wenn e_i nicht im 99%-Konfidenzintervall von Students t-Kurve mit 140 Freiheitsgraden liegt [51].

Lemma der Unifizierung von Outliern in Regressionsmodellen: $DB(p_0, D_0)$ unifiziert $Def_{Regression}$ mit $p_0 = 0{,}99$ und $D_0 = 0{,}258$. Der Beweis für dieses Lemma wird in einem separaten, nicht veröffentlichten Dokument [51] von Knorr und Ng geführt.

Generell ist die Outlier Erkennung in Regressionsmodellen ein wichtiges Thema der Statistik und wird von den hier vorgestellten entfernungsabhängigen Ansätzen unterstützt. Weitere Beispiele für die Unifizierung von verteilungsbasierten Unterscheidungstests für die Exponentialverteilung und die Poisson-Verteilung können direkt in der angegebenen Literaturquelle [3] nachgelesen werden. Allen diesen Beispielen ist gemein, dass die

Werte p_0, D_0 wohldefiniert waren. Aber was geschieht in den Fällen, welche für vorgegebene Entscheidungstests ungeeignet sind? Ein erster Ansatz geht von der Bereitstellung effektiver Visualisierung und effizienter inkrementeller Werkzeuge zur experimentellen Veränderung des Wertepaars p_0, D_0 aus. Ein effizienter Algorithmus zum Finden von $DB(p_0, D_0)$ - Outliern ist eine Voraussetzung für die notwendige Performance derartiger interaktiver Werkzeuge. Allerdings gehen selbst Knorr und Ng davon aus, dass eine heutige Implementierung von $DB(p_0, D_0)$ - Outliern nicht auf die Erarbeitung optimater Werte für *p* und *D* ausgerichtet ist, da dies im Allgemeinen sehr schwer zu erreichen sei. Auch ist zu bedenken, dass eine Visualisierung bei multidimensionalen Datenmengen mit viel mehr als 3 Dimensionen vom Nutzer ggf. ein sehr hohes räumliches Vorstellungsvermögen oder bei der Arbeit mit Projektionen auf handhabbare Unterräume wiederum eine sehr hohe Abstraktion verlangt. Zudem sind solche Projektionsmechanismen sehr rechenintensiv (vgl. [104] und [87]). Ein alternativer zweiter Ansatz bezieht den Nutzer ein, welcher Werte für *p* und *D* vergibt und verändert. Hier sollen Beispieltechniken dem Anwender geeignete Startwertkombinationen zur Verfügung stellen, da sonst ein hoher Kostenaufwand mit ungeeigneten Entfernungswerten für große Datenmengen ergebnislos bleibt. Durch das Sampling eines Beispiels von $\hat{n}$ von n Tupeln in der Menge wird ein Konfidenzintervall für die initialen Schätzungen von *D* bei gegebenem *p* erarbeitet.

Um die unifizierende Natur des Ansatzes entfernungsbasierter Outlier verständlicher zu machen, wird hier auf deren Beziehung zu Clustering-Methoden eingegangen. Im Prinzip klassifizieren Clustering Verfahren gleichartige Objekte und bieten vergleichsweise wenig Unterstützung für Outlier Detection. Outlier werden meist als Beiprodukt angesehen und als „Rauschen“ entfernt, jedoch nicht identifiziert. Das konkrete Verfahren DBSCAN [100] bietet dabei einen direkten Bezug zum Ansatz der $DB(p_0, D_0)$ - Outlier. Es klassifiziert Objekte in Kern-, Rand- und Außen-Bereiche abhängig von der Anzahl der Objekte in einer ε -Nachbarschaft, sowie abhängig von der Erreichbarkeit und der Verbundenheit eines Objekts. Diese ε -Nachbarschaft ist direkt vergleichbar mit der *D*-Nachbarschaft, sie verwendet ein vergleichbares statistisches Maß, arbeitet aber mit kleinen Werten für die Bildung starker Cluster, während *D* entsprechend groß ist, um starke Outlier zu finden. Den Erreichbarkeits- und Verbindungsbegriffen von DBSCAN wird beim $DB(p_0, D_0)$ Ansatz nicht widersprochen. Zusammenfassend kann festgestellt werden, dass DBSCAN auf maximal große Cluster abzielt und sehr zurückhaltend beim Kennzeichnen von Outliern ist. $DB(p_0, D_0)$ hingegen ist so aufgebaut, dass Outlier nach vom Anwender gewählten bzw. vorgegebenen Parametern *p* und *D* erkannt und gekennzeichnet werden. Im Fall $\varepsilon = D$ sind die identifizierten Outlier also gleich, in der Praxis sollte *D* jedoch größer als ε sein. Darüberhinaus sind Clustering-Algorithmen, also die konkreten Umsetzungen der Ansätze in praktikable und kosteneffiziente Verfahren, nicht für die Unifizierung von Unterscheidungstests entworfen. Dies muss berücksichtigt werden, wenn Clustering der Outlier Detection gegenübergestellt wird.

Ein großer Vorteil des einheitlichen entfernungsbasierten Ansatzes ist der Ersatz für viele Unterscheidungstests. Abbildung 4 zeigt einen möglichen Entscheidungsbaum für die Anwendung verteilungsbasierter Unterscheidungstests und unterlegt damit deutlich die Reduzierung des Aufwandes durch die entfernungsbasierte Unifizierung, sofern eine Wahl geeigneter Parameter für *p* und *D* gegeben ist.

Der Anwender muss sonst die Wahl des Tests von vielen Aspekten abhängig machen, wie z.B. Verteilungen, Verteilungswerten und Outlier Charakteristika. Eine Reihe dieser Informationen sind nur schwer festzustellen oder gar nicht vorhanden, wenn die Verteilung unbekannt ist. Mit dem einheitlichen Ansatz können die Fragestellungen und damit verbundenen Entscheidungen vermieden werden. Knorr und Ng beanspruchen keine Universalität des Verfahrens, verweisen aber darauf, dass es besser ist, als die meisten Standard-Statistiktests [3].

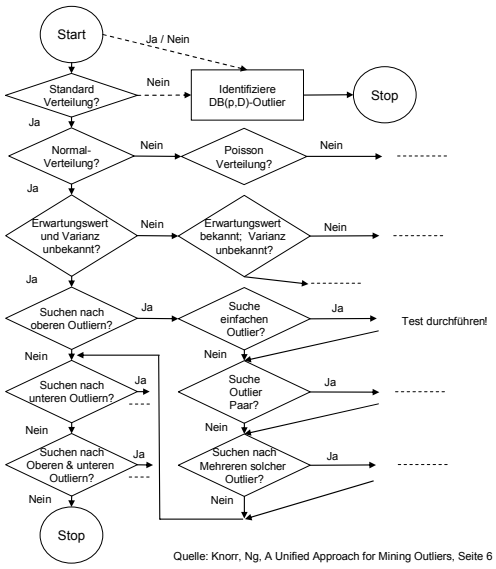


Abbildung 4 - Auswahlbaum für statistische Outliererkennung

Abbildungsbeschreibung: Statistische Erkennungsverfahren [33] für Outlier setzen eine Reihe von Annahmen über die Natur der zu erkennenden Outlier voraus. Der hohe Aufwand der Auswahl eines Verfahrens wird durch den dargestellten Entscheidungsbaum deutlich und motiviert Verfahrensalternativen [3], für die derartige Entscheidungen nicht a priori als Annahmen getroffen werden müssen.

Entfernungsabhängige Tests erlauben es dem Anwender, durch die geeignete Wahl der Parameter *p* und *D* selbst Einfluss auf das Verfahren zu nehmen. Ein weiteres Charakteristikum der entfernungsbasierten Tests ist die homogene Betrachtung aller Dimensionen bezogen auf die Attribute, welche zum Vergleich von Verhalten und zur Identifizierung von Outliern verwendet werden. Zudem ist das statistische Maß, die Entfernungsfunktion *d* kombiniert mit der Entfernung *D*, ein globales Maß, die dadurch entdeckten Outlier haben also einen globalen Charakter.

Für entfernungsbasierte Tests existieren mehrere Algorithmen mit verschiedenen Kostenabschätzungen. Es werden von Knorr und Ng zwei Algorithmen für mehrdimensionale Datenmengen angeboten, welche eine Komplexität von $O(mn^2)$ haben, wobei *n* die Anzahl der Objekte und *m* die Anzahl der Dimensionen mit $m \geq 2$ ist. Zusätzlich existiert ein partitionsbasierter Algorithmus, der eine Komplexität von $O(n)$ bei gegebenem *m* hat, allerdings exponentiell gegenüber wachsendem *m* ist. Eine Abwandlung dieses Algorithmus wird zudem für große, festplattenbasierte Datenmengen angeboten und garantiert eine maximale Zahl von drei Läufen über die Datenmenge auf dem Speichermedium. Dies stellt einen großen Vorteil dar, da bei dieser Art von Datenspeicherung i.d.R. die Zugriffszeiten die größten Kosten verursachen. Auch dieser Algorithmus ist für $m \leq 4$ nach experimentellen Untersuchungen die beste Wahl für diesen generellen Outlier Detection Ansatz. Fraglich bleibt allerdings, wie sich die Performance für Datenmengen mit einer sehr hohen Zahl an Dimensionen entwickelt. Hier legt die Tatsache, dass die Kosten mit *m* exponentiell wachsen, den Einsatz anderer Algorithmen oder generell anderer Outlier Detection Verfahren nahe. Im Folgenden sind die entsprechenden Algorithmen nach [8] kurz beschrieben.

Ein naives Verfahren zum Finden aller $DB(p_0, D_0)$-Outlier in univariaten Testmengen ist das Feststellen der Anzahl von Objekten, welche sich in der *D*-Nachbarschaft von *x* befinden, was eine Komplexität von $O(n^2)$ hat, wenn es für alle *n* Objekte *x* festgestellt wird. Ein optimiertes Verfahren baut eine indizierte räumliche Suchstruktur auf, welche dann mit einer Intervallabfrage dazu benutzt wird, die Anzahl der Objekte in der *D*-Nachbarschaft zu *x* zu finden. Der Aufwand dafür liegt zwischen $O(n \log n)$ und $O(n^2)$, abhängig von der tatsächlichen Verteilung der Objekte in der Datenmenge. Für mehrdimensionale Mengen ist die Berechnung der Entfernung von *m* abhängig, wobei die Komplexität mit $O(mn^2)$ angenommen werden kann, sofern der Aufwand der Bestimmung des Entfernungsmaßes linear von *m* abhängt.

Für eine kleine Zahl an Dimensionen *m* ist ein spezieller, zellenbasierter Algorithmus besonders effizient, weil er mit linearem Aufwand zu *n* bei festem *m* betrieben werden kann. Der Effizienzgewinn resultiert aus der

Reduzierung der Objekt-zu-Objekt Berechnungen durch Rückführung auf eine Zelle-zu-Zelle Berechnung. Ein Schritt für Schritt Algorithmus *FindAllOutsM* in Pseudo-Code wird in der entsprechenden Literatur [3] ausgeführt. Knorr und Ng [8] verweisen darauf, in weiterer Forschung Optimierungen für Datenmengen mit mehr als 10 Dimensionen zu suchen. Außerdem wird von den Autoren bereits ein Algorithmus *FindAllOutsD* angeboten, welcher sich für festplattenbasierte Datenmengen eignet und I/O Zugriffe durch Pufferungsmechanismen minimiert.

2.4.2. Entfernungsbasierte Outlier zum *k*-ten nächsten Nachbarn

Ramaswamy, Rastogi und Shim [13] stellen eine alternative Formulierung entfernungsbasierter Outlier vor, welche sich auf die Distanz eines Punktes zu seinem *k*-ten nächsten Nachbarn bezieht. Jeder Punkt wird hier anhand der Entfernung zum *k*-ten nächsten Nachbarn eingeordnet, und die top-*n* Punkte dieser Anordnung werden als Outlier deklariert.

Die Autoren beschreiben Outlier als solche Objekte, deren Abweichung bzw. Inkonsistenz mit den verbleibenden Daten groß genug ist, um sie als Outlier in Betracht zu ziehen. Sie verweisen auf die eingeführten Verfahren der statistisch basierten Abweichungstests nach Barnett und Lewis [33] und auf die entfernungsbasierten Outlier nach Knorr und Ng [6]. Für die weitere Betrachtung ist es wichtig, nochmals darauf hinzuweisen, dass die Entfernungsfunktion als eine beliebige metrische Funktion zur Beschreibung der Distanz zwischen zwei Punkten angenommen wird, wobei sich die vorgestellten Algorithmen auf eine euklidische Entfernungsfunktion beziehen.

Gegenüber den statistischen Tests, welche eine Kenntnis über die zugrunde liegende Verteilung und entsprechende statistische Maße, sowie auch über den Charakter gesuchter Outlier voraussetzen, hat das entfernungsbasierte Verfahren von Knorr und Ng den Vorteil, dass a priori kein Wissen über die Verteilung erforderlich ist. Das Verfahren ist nicht nur intuitiv und simpel, es ist auch generell genug, um statistische Verteilungstests zu modellieren. Die Algorithmen, welche vorgestellt werden, sind jedoch lediglich in Bezug auf die Anzahl der Objekte von linearer Komplexität und der Aufwand steigt im günstigsten Fall, d.h. unter Anwendung des effizienten zellenbasierten Algorithmus, exponentiell mit der wachsenden Zahl an Dimensionen. Zudem muss der Anwender den Entfernungsparameter *D* festlegen, welcher in Kombination mit dem Parameter *p* das Verfahren auf eine optimale Erkennung von Outliern kalibriert. *p* ist dabei der Anteil der Objekte, welche sich weiter als die Entfernung *D* von einem Objekt *x* entfernt befinden müssen, damit *x* als Outlier klassifiziert wird. Das Finden eines optimalen Wertes für *D* ist möglicherweise sehr schwierig. Knorr und Ng [3] schlagen eine Trial & Error Methode mit Iterationen vor, um von einem Startwert für *D* ausgehend eine passende Wertekombination zu finden. Sie verweisen jedoch auch auf die Schwierigkeiten, welche mit der Parameterwahl verbunden sein können. Zudem liefert das Verfahren keine Ordnung der Outlier, also keinen Hinweis auf die Stärke und den Grad des Outlier-Charakters eines Objekts. Dies wird von Ramaswamy, Rastogi und Shim als Nachteil der Methode aufgeführt, wobei in ihrer Veröffentlichung zwar Bezug auf die Bemühungen von Knorr und Ng [55] genommen wird, Gründe für die Charakterisierung eines Objektes als Outlier durch das Finden von „Intensional Knowledge“ anzuführen, diese Gründe jedoch nicht mit den durch das vorliegende Konzept bereitgestellten Outlier-Charakteristika vergleichbar sind.

Um die genannten Nachteile und Probleme zu adressieren, führen die Autoren die o.g. Definition eines Outliers ein. Diese hat zudem den Vorteil, dass ein Parameter *D* nicht angegeben werden muss. Anstelle dessen bezieht sich die Definition auf die Entfernung zum *k*-ten nächsten Nachbarn eines Punktes.

Definition der Entfernung zum *k*-ten nächsten Nachbarn: Für ein gegebenes *k* und einen Punkt *x* sei $D^k(x)$ die Entfernung zum *k*-ten nächsten Nachbarn [61] von *x*.

Intuitiv ist $D^k(x)$ ein Maß dafür, zu welchem Grad der Punkt *x* ein Outlier ist. Punkte mit einem größeren Wert für $D^k(x)$ haben zum Beispiel spärlicher besetzte Nachbarschaften und sind damit typischere Kandidaten für Outlier, als solche Punkte, welche zu dichten Clustern gehören und damit geringere Werte für $D^k(x)$ haben. Da der Anwender im Allgemeinen an den top-*n* Outliern interessiert ist, seien Outlier demgemäß wie folgt definiert:

Definition von *D(k,n)*-Outliern: Bei gegebenem *k* und *n* ist ein Punkt *x* ein Outlier, wenn nicht mehr als $n-1$ andere Punkte x' in der Datenmenge *X* einen höheren Wert für D^k haben als *x*, also:

$$X' \subset X, \qquad X' = \{x' \mid D^k(x') > D^k(x)\}, \qquad |X'| \leq n-1$$

In anderen Worten werden also die top-*n* Punkte mit maximalen Werten für D^k als Outlier in Betracht gezogen. Diese Outlier werden folgend als D_n^k-Outlier bezeichnet. Da mehr als *n* Outlier diese Definition erfüllen können, seien alle Outlier aus dieser erfüllenden Menge D_n^k-Outlier.

Mit dieser neuen Definition entfällt die Notwendigkeit für den Anwender, einen Wert für *D* vorzugeben. Anstelle dessen soll der Nutzer vorgeben, wieviele Outlier von Interesse sind. Die Definition nutzt generell die Entfernung zwischen dem *k*-ten Nachbarn des *n*-ten Outliers um die Nachbarschaftsdistanz *D* zu definieren. Da der Wert für *n* in der Regel als klein und vor allem als relativ unabhängig von der zu untersuchenden Datenmenge vorausgesetzt werden kann, ist dieser durch den Anwender leichter festzulegen, als *D*. Wichtig ist in diesem Zusammenhang der Unterschied der Nachbarschaftsbegriffe in Abgrenzung zu dichtebasierten lokalen Outliern, wo Verfahren zu deren Erkennung ein Dichtemaß ([13] mit Verweis auf [4]) verwenden. Im Gegensatz hierzu nutzt das hier vorgestellte Verfahren ein Entfernungsmaß.

Für die Entfernung kann jede der L_p Metriken (z.B. die euklidische) eingesetzt werden, aber auch nicht-metrische Entfernungsfunktionen, z.B. für Anwendungsgebiete mit Textdokumenten. Dies macht die Outlier Definition sehr generell. Zusätzlich gilt, dass alle n' *DB(p,D)*-Outlier, welche durch das von Knorr und Ng vorgestellte Verfahren gefunden werden, gleichzeitig auch $D_{n'}^k$-Outlier sind.

Eines der eingesetzten Schlüsselverfahren im Rahmen des vorgestellten Ansatzes ist die Approximation von einer Punktmenge unter Nutzung ihres minimalen Grenzrechtecks („minimum bounding rectangle“) MBR. Durch die Berechnung unterer und oberer Grenzen für $D^k(x)$ für die Punkte in einem MBR können solche MBRs, welche keine D_n^k-Outlier enthalten können, identifiziert und abgeschnitten werden. Die Berechnung derartiger Grenzen für MBRs erfordert die Ermittlung der minimalen und der maximalen Distanz zwischen zwei MBRs. Die Outlier-Erkennung wird zusätzlich durch die Berechnung der minimalen und maximalen Distanz zwischen einem Punkt und einem MBR unterstützt.

Im vorgestellten Ansatz wird das Quadrat der euklidischen Entfernung anstatt der euklidischen Entfernung selbst als metrisches Entfernungsmaß verwendet, weil es weniger und unaufwändigere Berechnungen erfordert. Die Distanz zwischen zwei Punkten *x* und *x'* sei als *d(x, x')* bezeichnet. Weiterhin sei ein Punkt *x* im *m*-dimensionalen Raum mit $[x_1,...x_m]$ bezeichnet und ein *m*-dimensionales Rechteck mit R_q durch seine Eckpunkte der Diagonale $r=[r_1,...r_m]$ und $[r_1',...r_m']$ mit $r_i \le r_i'$ für $1 \le i \le m$ gegeben.

Die minimale Entfernung zwischen einem Punkt *x* und einem Rechteck R_q sei mit $d_{\min}(x,R_q)$ bezeichnet und jeder Punkt innerhalb R_q ist mindestens $d_{\min}(x,R_q)$ von *x* entfernt. Diese minimale Entfernung sei im Folgenden definiert (vgl. auch [61]).

Definition der minimalen Entfernung zwischen einem Punkt und einem Rechteck: $d_{\min}(x,R_q)=\sum_{i=1}^{m} p_i^2$

mit (1) $p_i = r_i - x_i$ für $x_i < r_i$ und (2) $p_i = x_i - r_i'$ für $r_i' < x_i$ und (3) $p_i = 0$ für alle anderen Fälle.

Die maximale Entfernung zwischen einem Punkt *x* und einem Rechteck R_q, wobei kein Punkt in R_q weiter als diese maximale Entfernung von *x* entfernt liegt, sei definiert wie folgt.

Definition der maximalen Entfernung zwischen einem Punkt und einem Rechteck: $d_{\max}(x,R_q)=\sum_{i=1}^{m} p_i^2$

mit (1) $p_i = r_i' - x_i$ für $x_i < \frac{r_i + r_i'}{2}$ und (2) $p_i = x_i - r_i$ für alle anderen Fälle.

Weiterhin wird die minimale und maximale Entfernung zwischen zwei MBRs R_q und S_q, welche durch ihre Diagonalen (r,r') bzw. (s,s') gegeben sind, so definiert, dass im Falle der minimalen Entfernung jeder Punkt in R_q mindestens diese Entfernung von jedem Punkt in S_q (und umgekehrt) liegt und im Falle der maximalen Entfernung jeder Punkt in R_q nicht weiter als diese Entfernung von jedem Punkt in S_q (und umgekehrt) entfernt liegt. Sie können nach den folgenden Formeln ermittelt werden:

Definition der minimalen Entfernung zwischen zwei Rechtecken: $d_{\min}(R_q,S_q)=\sum_{i=1}^{m} p_i^2$ mit (1) $p_i = r_i - s_i'$

für $s_i' < r_i$ und (2) $p_i = s_i - r_i'$ für $r_i' < s_i$ und (3) $p_i = 0$ für alle anderen Fälle.

Definition der maximalen Entfernung zwischen zwei Rechtecken:

$$d_{\max}(R_q, S_q) = \sum_{i=1}^{m} p_i^2 \quad \text{mit} \quad p_i = \max\left\{ |s_i' - r_i|, |r_i' - s_i| \right\}.$$

In der Literaturquelle stellen die Autoren zwei relativ geradlinige Algorithmen zur Lösung der Aufgabe des Findens von D_n^k -Outliern vor. Ein erster verschachtelter Schleifenalgorithmus ermittelt für jeden Eingabepunkt *x* die Distanz zum *k*-ten nächsten Nachbarn, also $D^k(x)$. Danach wählt er die top-*n* Punkte mit den maximalen Werten für D^k aus. Um die D^k -Werte für die Punkte der Datenmenge zu berechnen, scannt der Algorithmus die Datenbank für jeden Punkt *x*. Für jeden Punkt *x* wird eine Liste der *k*-ten nächsten Nachbarn von *x* verwaltet und für jeden in Frage kommenden Punkt x' der Datenmenge wird geprüft, ob *d(x, x')* kleiner ist, als die Entfernung zum bisher gefundenen *k*-ten nächsten Nachbarn. Wenn dieser Test positiv verläuft, wird x' in die Liste der *k*-ten nächsten Nachbarn von *x* aufgenommen und sollte diese Liste bereits *k* Elemente enthalten, wird der am weitesten von *x* entfernte Punkt aus dieser Liste im Gegenzug gelöscht.

Eine I/O Optimierung ist möglich, wenn D^k für ganze Blöcke von Punkten berechnet wird, jedoch auch in diesem Fall beträgt die Komplexität $O(n_X^{\,2})$, wobei n_X die Anzahl der Elemente in der Datenmenge bezeichnet. Allerdings ist dies rechentechnisch sehr teuer, vor allem unter Berücksichtigung einer hohen Zahl an Dimensionen pro Punkt. Durch den Einsatz eines räumlichen Index-Mechanismus, z.B. eines R*-Tree [62], kann die Anzahl der Entfernungsberechnungen signifikant reduziert werden. Sind alle Punkte in einem R*-Tree erfasst, kann folgende Optimierung durch Beschneidung angewandt werden, um die Anzahl der Entfernungsberechnungen herabzusetzen:

Unter der Voraussetzung, dass $D^k(x)$ für *x* durch Betrachtung einer Untermenge der Eingabepunkte berechnet wurde, ist dieser errechnete derzeitige Wert eine obere Grenze für den wirklichen Wert $D^k(x)$. Wenn nun die minimale Entfernung zwischen *x* und dem MBR eines Knotens im R*-Tree diesen derzeitigen Wert für $D^k(x)$ übersteigt, wird keiner der Punkte im Unterbaum dieses Knotens unter den *k*-ten nächsten Nachbarn von *x* sein. Dieser Unterbaum kann somit abgeschnitten werden.[1]

Als weitere Verbesserung kann die folgende Beschneidung der Berechnung von $D^k(x)$ in Bezug auf den Gedanken, die top-*n* Outlier zu finden, eingesetzt werden: Die bereits errechneten top-*n* Outlier werden bei jedem Schritt des Index-basierten Algorithmus vorgehalten. Sobald der für einen Punkt *x* errechnete Wert $D^k(x)$ unter den kleinsten D^k -Wert der bereits errechneten top-*n* Outlier fällt, kann *x* kein top-*n* Outlier mehr sein, weil $D^k(x)$ monoton fällt, je mehr Punkte in die Berechnung einbezogen werden. Diese Beschneidung lässt sich auch auf den verschachtelten Schleifenalgorithmus anwenden.

Die zwei Algorithmen selbst sind ausführlich in der Literatur [13] beschrieben. Ihre Nachteile, vor allem der entsprechend hohe Aufwand, werden durch die Einführung eines partitionsbasierten Algorithmus aufgegriffen. Dieser entsteht durch die Berechnung der Werte für $D^k(x)$, also der Entfernung zwischen *x* und seinem *k*-ten nächsten Nachbarn. Da jedoch nur die top-*n* Outlier von Interesse sind und *n* typischerweise ein kleiner Wert ist, sind die Entfernungen für den Großteil der verbleibenden Punkte nicht von Bedeutung und sollten daher auch nicht berechnet werden müssen. Der vorgeschlagene partitionsbasierte Algorithmus beschneidet die Entfernungsberechnungen für solche Punkte, deren Distanz zu ihrem *k*-ten nächsten Nachbarn so gering ist, dass sie unmöglich in die Gruppe der top-*n* Outlier aufrücken können. Durch die Partitionierung der Datenmenge kann diese Entscheidung getroffen werden, ohne dass der präzise Wert für $D^k(x)$ berechnet werden muss. Die experimentellen Untersuchungen (hier am Beispiel von Daten aus der National Basketball Association – NBA) belegen substantielle Einsparungen an Rechenzeit und I/O Kosten durch diese Beschneidung.

In Abbildung 5 sind die experimentellen Ergebnisse grafisch aufgezeichnet. Da der Partitionsalgorithmus das Vor-Clustering der Daten zum Finden geeigneter Partitionen voraussetzt, wurde dieser Aufwand eingerechnet. Der direkte Aufwand des Outlier-Schritts im Gegensatz zum Clustering ist daher nochmals separat aufgeführt. Das Clustering erfolgte in diesen Experimenten mittels BIRCH [60].

[1] In der Literaturquelle [61] wird ein abweichender Paramerter für die Grenzbestimmung eingesetzt, welcher sich an dem dort verfolgten Ziel orientiert. Hier liefert $d_{\max}(x, MBR)$ lt. [13] eine engere Grenze.

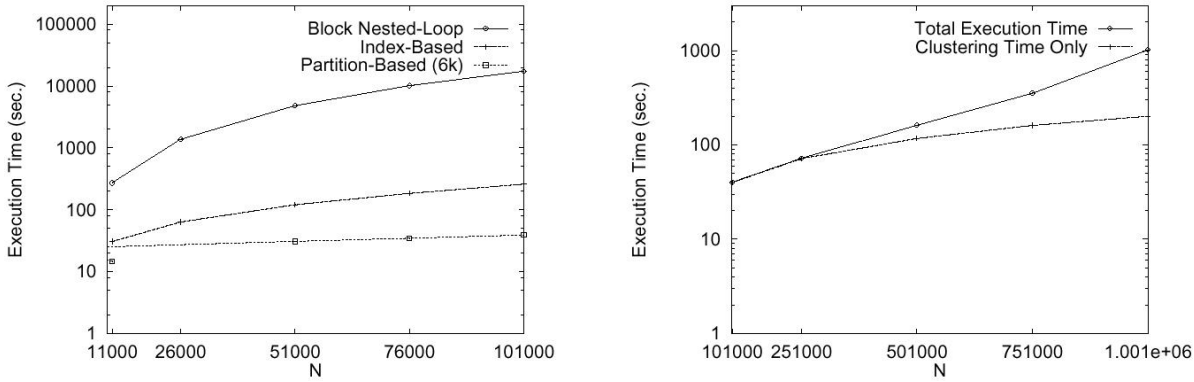


Abbildung 5 - Experimentelle Ergebnisse des Partitionsalgorithmus

Abbildungsbeschreibung: Die Ergebnisse der Experimente des Partitionsalgorithmus nach [61] legen nahe, dass dieser bei einer steigenden Zahl an Dimensionen besser skaliert.

Der partitionsbasierte Algorithmus skaliert besser bei steigender Zahl an Dimensionen, erfordert jedoch die Anpassung des Radius der Cluster beim Vor-Clustering, da bei steigender Dimensionszahl die Punkte relativ spärlich im Raum angeordnet sind und für den Outlier-Erkennungsschritt sichergestellt werden muss, dass Punkte in einem Cluster relativ nah an anderen Punkten in demselben Cluster liegen, wenn sie mit den Punkten in anderen Clustern verglichen werden.

Die Experimente der Autoren haben gezeigt, dass die Anwendung des Verfahrens zur Entdeckung erwarteter aber auch unerwarteter Aspekte der gestesteten Datenmenge aus der NBA führte. Die Experimente mit synthetischen Daten legen eine gute Skalierbarkeit bei wachsender Zahl an Dimensionen nahe. Ein Nachteil des Partitionsverfahrens ist aus Sicht dieser Arbeit jedoch in der notwendigen Clustererkennung zu sehen, insbesondere wenn aufgrund einer spärlichen Besetzung des Suchraumes eine erfolgreiche Clusterzuordnung nahezu ausgeschlossen ist bzw. nicht die Qualität zur erfolgreichen Abgrenzung von Outliern hat.

2.5. Dichtebasierte Outlier Detection Ansätze

2.5.1. Local Outlier Factor (*LOF*)

Breuning, Kriegel, Ng und Sandner [4] führen einen weiteren Begriff im Rahmen der Outlier Detection Ansätze ein. Dieser ist vor allem durch den globalen und gleichzeitig binären Charakter der Outlier in den vorher besprochenen Ansätzen motiviert. Diese Verfahren gehen davon aus, dass ein Objekt durch den Algorithmus als Outlier fest identifiziert werden muss. Dies setzt nicht nur eine sehr stringent angewandte Unterscheidungstechnik voraus, sondern ist zudem auch äußerst unflexibel gegenüber der intuitiven Definition eines Outliers.

Breuning et al führt hierfür einen sogenannten Outlier Faktor ein, also einen Grad, zudem ein Objekt ein Outlier sein kann. Dies erweitert den Interpretationsspielraum erheblich und ist ein sehr viel flexiblerer Ansatz. Zudem wird bei dem vorgestellten Verfahren der lokale Charakter eines Outliers betont. Im Rahmen von Verfahren, welche Outlier in Bezug zur Gesamtheit der Objekte in einer Datenmenge global definieren und demgemäß identifizieren, gibt es keine Möglichkeit, Outlier zu erkennen, die ggf. gegenüber einer Teilmenge von Objekten einen starken Outlier Charakter haben, dieser aber in der globalen Betrachtung durch die Konstellation der restlichen Objektmehrheit so überdeckt wird, dass er sich mit einem globalen Verfahren unmöglich erkennen ließe. Breuning, Kriegel, Ng und Sandner lösen dies indirekt, indem sie einen sogenannten *LOF* (Local Outlier Factor) einführen, welcher für ein Objekt einen lokalen Grad angibt, gegenüber anderen Objekten ein Outlier zu sein. Andere Ansätze für eine solche graduelle Anordnung werden u.a. im Kapitel zu clusterbasierten Outliern (*CBLOF*) vorgestellt.

Nach der dichtebasierten Methode der Erkennung von Outliern wird jedem Objekt ein lokaler Faktor (*LOF*) zugewiesen, der bestimmt, zu welchem Grad ihm ein Outlier Status zugeordnet werden kann. Dies ist ein Konzept, welches die Outlier-Eigenschaft nicht nur qualifiziert, sondern auch quantifiziert. Der angegebene Faktor ist im Sinne einer begrenzten Nachbarschaft lokal. Eben diese Nachbarschaft und vor allem die Dichte der Objekte in derselben werden bei der Bestimmung des Faktors eines Objekts herangezogen. Der Ansatz selbst ist lose verwandt mit dem dichtebasierenden Clustering, jedoch ohne dass eine explizite oder implizite Begrifflichkeit von Clustern für diese Methode notwendig wäre. Im Folgenden wird eine kurze Einführung in

das *LOF*-Konzept, die Begriffe und ihre Eigenschaften, die praktische Anwendung des Verfahrens und die experimentell nachgewiesene Performance der dichtebasierten Outlier Detection Verfahren gegeben.

In Abbildung 6 wird eine Beispielverteilung gezeigt, welche eine Situation in einer Datenmenge illustriert. Zwei Cluster unterschiedlicher Größe und Dichte sind gegeben. Der Cluster C_2 ist dichter als der Cluster C_1.

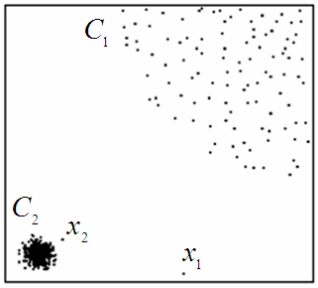


Abbildung 6 - Beispielverteilung für die Erkennung dichtebasierter Outlier (*LOF*)

Abbildungsbeschreibung: Die Objekte x_2 als auch x_1 sind intuitive Outlier, auch wenn x_2 deutlich näher am Cluster C_2 liegt, als die Objekte des Clusters C_1 voneinander entfernt liegen. Dadurch wird eine lokale Betrachtung motiviert, weil rein entfernungsbasierte Verfahren mit hoher Wahrscheinlichkeit x_2 nicht als Outlier erkennen würden.

Nach Hawkins Definition von Outliern, sind sowohl x_2 als auch x_1 mit hoher Tendenz intuitiv mögliche Outlier. Zurückkommend auf die Frage, welche Verfahren diese Outlier erkennen können, merken Breuning und Ng an, dass der von Knorr und Ng vorgestellte entfernungsbasierte Ansatz lediglich x_1 als *DB(p,D)*-Outlier identifiziert, denn wenn für jedes Objekt $x \in C_1$ die Distanz zwischen *x* und dem nächsten Nachbarn größer ist, als die zwischen x_2 und C_2, so gibt es kein geeignetes Wertepaar *p, D* zur Erkennung von x_2 als Outlier, ohne dass nicht auch alle Objekte von C_1 als *DB(p,D)*-Outlier klassifiziert würden [4]. Im Ergebnis sind entfernungsbasierte Verfahren also unter gewissen Bedingungen mit ihrer globalen Outlier Sicht adäquat und sinnvoll, liefern aber keine zufriedenstellenden Ergebnisse bei der Existenz von Clustern verschiedener Dichte.

Der *LOF*-Ansatz wird nun wie folgt formal definiert:

Definition der *k*-Distanz: Bei gegebener Datenmenge *X* für jede positive ganze Zahl *k* sei die *k-distance* (*k*-Distanz) $k_d(x)$ eines Objektes *x* definiert als die Entfernung *d(x, x')* zwischen *x* und *x'* aus *X* mit: (i) für mindestens *k* Objekte $x'' \in X \setminus \{x\}$ gilt $d(x,x'') \le d(x,x')$ und; (ii) für maximal *k*-1 Objekte $x'' \in X \setminus \{x\}$ gilt $d(x,x'') < d(x,x')$.

Definition der *k*-Distanz Nachbarschaft: Bei gegebener *k-distance* von *x* ist die *k-distance-neighbourhood* (Nachbarschaft) von *x* die Menge an Objekten *x'*, welche jedes Objekt enthält, dessen Distanz von *x* nicht größer als die *k-distance* ist: $N_k(x) = \{x' \in X \setminus \{x\} \mid d(x,x') \le k_d(x)\}$. Diese Objekte werden auch als *k*-nächste Nachbarn von *x* bezeichnet. $k_d(x)$ ist wohldefiniert für positive ganzzahlige *k*, da aber das Objekt *x'* nicht einzigartig sein muss, kann die Kardinalität von $N_k(x)$ durchaus größer sein als *k*, also ist $|N_k(x)| \ge k$.

Definition der Erreichbarkeitsdistanz: Die Erreichbarkeitsdistanz eines Objektes *x* bezogen auf ein Objekt *x'*, *reach-dist* mit *k* als Element der natürlichen Zahlen $k \in \mathrm{N}$, sei definiert als $rd_k(x,x') = \max\{k_d(x'), d(x,x')\}$ nach folgendem Konzept: (i) wenn *x* weit von *x'* weg liegt, also weiter als die *k*-distance von *x'*, dann wird die tatsächliche Entfernung als Erreichbarkeits-Distanz angenommen; (ii) ist die tatsächliche Distanz zwischen *x* und *x'* allerdings genügend klein, wird sie durch die *k*-distance von *x'* ersetzt. Abbildung 7 zeigt dieses Vorgehen.

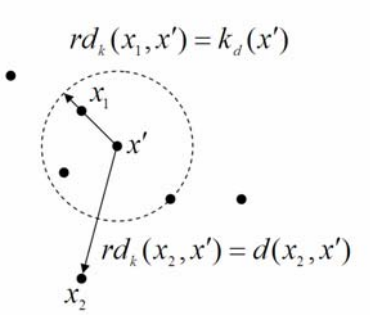


Abbildung 7 - Erreichbarkeitsdistanz von Objekten

Abbildungsbeschreibung: Die Erreichbarkeitsdistanz eines Objektes in Bezug zu einem anderen Objekt ist als das Maximum der *k*-distance des Objektes im Vergleich zur Distanz zwischen diesen Objekten definiert, um statistische Fluktuationen zu vermeiden.

Durch dieses Vorgehen werden statistischen Fluktuationen für die Objekte *x*, welche sich sehr nahe an x' befinden, deutlich reduziert. Die Stärke dieses Glättungseffektes kann durch die Wahl des Parameters *k* gesteuert werden. Je höher *k* gewählt wird, desto ähnlicher sind sich die Erreichbarkeits-Distanzen für Objekte in derselben Nachbarschaft.

Der Parameter *k* stellt die direkte Verbindung zum so genannten dichtebasierten Clusteringverfahren DBSCAN [100] dar. Dieses definiert den Dichtebegriff mit 2 Parametern: (i) mit *MinPts* (spezifiziert dabei eine minimale Anzahl an Objekten) und (ii) mit einem Volumenparameter. Beide bestimmen eine gewisse Dichtebarriere. Objekte oder Regionen gelten als verbunden, wenn ihre Nachbarschaftsdichten diese gegebene Barriere überschreiten. Zur Erkennung von dichtebasierten Outliern muss allerdings die Dichte unterschiedlicher Objektmengen verglichen werden. Dies erfordert eine dynamische Bestimmung der Dichte von Objektmengen. Aus diesem Grund wird für den *LOF*-Ansatz nur der Parameter *MinPts* herangezogen und es wird der Wert der (*MinPts*)-Erreichbarkeitsdistanz $rd_{MinPts}(x,x')$ für $x' \in N_{MinPts}(x)$ als Volumenmessung herangezogen, um die Dichte in der *MinPts*-Nachbarschaft eines Objektes *x* zu bestimmen. Es wird hier also eine spezielle Instanz des Parameters *k* eingesetzt, eben $k = MinPts$.

Definition der lokalen Erreichbarkeitsdichte: Die lokale Erreichbarkeitsdichte eines Objektes *x* (*local-reachibility-distance*) sei definiert als:

$$lrd_{MinPts}(x) = 1/\left(\frac{\sum_{x' \in N_{MinPts}(x)} rd_{MinPts}(x,x')}{|N_{MinPts}(x)|} \right).$$

Intuitiv ist die lokale Erreichbarkeitsdichte eines Objektes *x* der Kehrwert der durchschnittlichen Erreichbarkeitsdistanz basierend auf den *MinPts*-nächsten Nachbarn von *x*. Die lokale Dichte kann durchaus ∞ sein, wenn alle Erreichbarkeitsdistanzen in Summe Null sind. Dies kann z.B. für ein Objekt *x* eintreten, wenn es mindestens *MinPts* von *x* verschiedene Objekte gibt, welche die exakt gleichen räumlichen Koordinaten wie *x* haben, also in der Datenmenge mindestens *MinPts* Duplikate von *x* existieren. Der Vereinfachung halber sei angenommen, es gibt keine Duplikate[2]. Duplikate können dementsprechend so behandelt werden, dass der Nachbarschaftsbegriff so definiert sei, dass es in einer *k*-Distanz mindestens *k* Objekte geben muss, deren räumliche Koordinaten von *x* verschieden sind.

Definition des lokalen Outlier Faktors (*LOF*): Der lokale Outlier-Faktor eines Objektes *x* (Local Outlier Factor – *LOF*) sei definiert als:

$$LOF_{MinPts}(x) = \frac{\sum_{x' \in N_{MinPts}(x)} \frac{lrd_{MinPts}(x')}{lrd_{MinPts}(x)}}{|N_{MinPts}(x)|}.$$

Dieser Wert erfasst den Grad, zu dem wir das Objekt *x* als Outlier bezeichnen. Es ist der Durchschnitt der Rate der lokalen Erreichbarkeits-Dichte von *x* und seiner *MinPts*-nächsten Nachbarn. Dabei ist einfach

[2] Die Schwierigkeit der Implementierung liegt ggf. darin, dass Duplikate sehr wohl vorkommen können.

abzulesen, dass (i) je kleiner die lokale Erreichbarkeitsdichte von *x* ist, und (ii) je höher die lokale Erreichbarkeitsdichte der *MinPts*-nächsten Nachbarn von *x* ist, desto höher ist der *LOF*-Wert von *x* und demnach der Grad, zu dem *x* als Outlier bezeichnet werden kann.

Die formalen Eigenschaften des *LOF* zeigen, dass die Definition des lokalen Outlier Faktors die Idee „lokaler" Outlier erfasst und eine Reihe sinnvoller Eigenschaften ableitbar sind. So lässt sich beweisen, dass der *LOF* Wert für die meisten Objekte in einem Cluster, insbesondere aber für solche Objekte, welche sich tief in einem Cluster befinden, nahe 1 liegt. Dieser Beweis ist ausführlich in der Literatur [4] zu diesem Thema ausgeführt.

Für die anderen Objekte, einschließlich der Objekte außerhalb eines Clusters, wird ein Theorem zur Beschreibung einer generellen oberen und unteren Grenze des *LOF* bereitgestellt. Dieses Theorem stellt geeignete obere und untere Grenzen für solche Objekte bereit, deren *MinPts*-nächste Nachbarn alle demselben Cluster angehören, auch wenn sich das Objekt selbst nicht tief in einem Cluster befindet. Für diese Fälle sind die angegebenen Grenzen eng genug gesetzt. Für alle anderen Fälle wird ein zweites Theorem bereitgestellt, um in spezifischen Situationen, in denen sich das erste Theorem nicht eignet, sinnvolle Grenzen bereitzustellen, welche eng genug sind. Auf diese Theoreme und die Betrachtung von Objekten tief in einem Cluster wird im Folgenden kurz eingegangen, weil verschiedene Aspekte wichtig zum späteren Verständnis der Wahl geeigneter *MinPts*-Werte für das Verfahren in der Praxis sind.

Lemma zu den Grenzen für den *LOF*: Sei C eine Menge von Objekten mit $C \subseteq X$ und (unter Vernachlässigung von vorher benutzen Indizes, sofern dadurch keine Verwirrung entsteht) $rd_{\min}$ die minimale Erreichbarkeitsdistanz von Objekten in *C*, also: $rd_{\min} = \min\{rd(x,x') \mid x,x' \in C\}$. Die maximale Erreichbarkeitsdistanz $rd_{\max} = \max\{rd(x,x') \mid x,x' \in C\}$ sei gleichermaßen definiert und sei

$$\varepsilon = \frac{rd_{\max}}{rd_{\min}} - 1,$$

und wenn für alle Objekte *x* aus *C* gilt: (i) alle *MinPts*-nächsten Nachbarn x' von *x* seien in *C*, und (ii) alle *MinPts*-nächsten Nachbarn x'' von x' seien auch in *C*, so gilt insgesamt, dass

$$\frac{1}{1+\varepsilon} \leq LOF(x) \leq 1+\varepsilon .$$

Der Beweis dafür ist in der Literatur entsprechend angegeben [4]. Das Lemma kann nun wie folgt interpretiert werden: Intuitiv ist *C* ein Cluster und sind Objekte *x* tief in diesem Cluster, so gilt für diese, dass die *MinPts*-nächsten Nachbarn x' von *x* in diesem Cluster liegen und gleichermaßen die *MinPts*-nächsten Nachbarn x'' der *MinPts*-nächsten Nachbarn x' von *x* in diesem Cluster liegen. Für solche tief im Cluster liegenden Objekte *x* ist der *LOF(x)* begrenzt. Sofern der Cluster sehr dicht ist, wird der ε Wert im Lemma sehr klein und damit liegt der *LOF* sehr nahe um 1. Aus diesem Grund sollten Objekte tief in einem Cluster nicht als lokale Outlier markiert werden. Was passiert jedoch mit Objekten am Rand oder außerhalb eines Clusters?

Im folgenden Theorem (1) wird das Lemma so generalisiert, dass es auf alle Objekte in der Datenmenge anwendbar ist und entsprechende Grenzen für den *LOF* eines Objektes *x* liefert, u.a. auch für Objekte tief innerhalb eines Clusters. Für dieses Theorem seien die folgenden Begriffe neu eingeführt:

Definition der direkten und indirekten Erreichbarkeitsdistanzen: Für jedes Objekt *x* sei

$$direct_{\min}(x) = \min\{rd(x,x') \mid x' \in N_{MinPts}(x)\}$$

und

$$direct_{\max}(x) = \max\{rd(x,x') \mid x' \in N_{MinPts}(x)\}$$

die minimale bzw. maximale Erreichbarkeitsdistanz zwischen *x* und einem der *MinPts*-Nachbarn von *x*.

Ebenso sei

$$indirect_{\min}(x) = \min\{rd(x',x'') \mid x' \in N_{MinPts}(x) \vee x'' \in N_{MinPts}(x')\}$$

und

$$indirect_{\max}(x) = \max\{rd(x',x'') \mid x' \in N_{MinPts}(x) \vee x'' \in N_{MinPts}(x')\}$$

die minimale bzw. maximale Erreichbarkeitsdistanz zwischen x' und dem *MinPts*-nächsten Nachbarn von x'. Damit ist die Erreichbarkeit der direkten *MinPts*-Nachbarschaft von *x* durch x' beschrieben und darauf

aufbauend die Erreichbarkeit der indirekten Nachbarschaft zwischen x'' und x', sofern x' ein *MinPts*-nächster Nachbar von *x* ist. Abbildung 8 zeigt die Definitionen am Beispiel, wobei *MinPts* mit dem Wert 3 angenommen ist und *direct* / *indirect* mit *d* / *i* abgekürzt sind.

Theorem (1) zu den Grenzen des LOF: Sei *x* ein Objekt der Datenmenge *X* mit $1 \le MinPts \le |X|$, dann ist

$$\frac{direct_{\min}(x)}{indirect_{\max}(x)} \le LOF(x) \le \frac{direct_{\max}(x)}{indirect_{\min}(x)}.$$

Dabei ist am Beispiel in Abbildung 8 deutlich zu sehen, dass *LOF(x)* einfacherweise eine Funktion der direkten Erreichbarkeitsdistanzen von *x* im Verhältnis zu den Erreichbarkeitsdistanzen in indirekter Nachbarschaft von *x* ist. Der Beweis von Theorem (1) ist in der Quellenliteratur zu dichtebasierten Outliern ausgeführt [4].

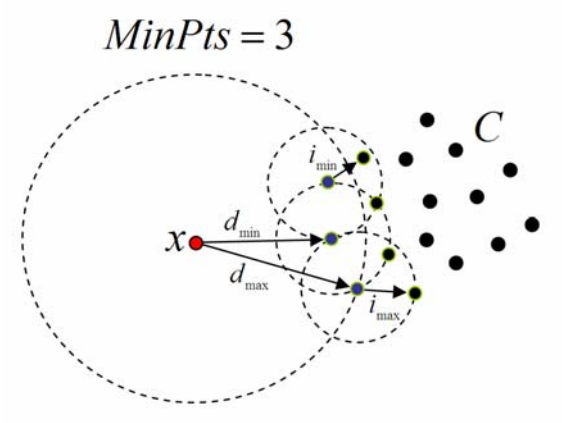


Abbildung 8 - Direkte und indirekte Erreichbarkeit von Objekten

Abbildungsbeschreibung: Um den Punkt *x* sind seine 3-nächsten Nachbarn gekennzeichnet und die minimale und maximale direkte Erreichbarkeitsdistanz (*d*) von *x* ist angedeutet, ebenso die indirekte minimale und maximale Erreichbarkeitsdistanz (*i*) der 3-nächsten Nachbarn der 3-nächsten Nachbarn von *x*. Aus diesen Distanzen lässt sich eine obere und untere Grenze für den *LOF(x)* nach Theorem (1) bestimmen.

Die aufgeworfene Frage nach der Qualität dieser Grenzen nach Theorem (1) für die unteren und oberen *LOF* Werte zu einem Objekt wird in der Literatur entsprechend untersucht und unter der Annahme, dass die statistische Fluktuation der Erreichbarkeitsdistanzen in den direkten und indirekten Nachbarschaften von *x* in derselben Größenordnung (d.h. unter vereinfachter Annahme mit demselben Wert) vorkommt, und diese Fluktuation durch den Parameter *pct* simuliert sei, kann gezeigt werden, dass die Spanne zwischen den oberen und unteren Grenzen des *LOF* nur von der Rate der indirekten zu den direkten Erreichbarkeitsdistanzen abhängt, nicht jedoch von deren absoluten Werten. Abbildung 9 zeigt, dass für eine geringe Fluktuation *pct* die *LOF*-Grenzen sehr geeignete Werte repräsentieren.

Die Fluktuation ist in zwei Fällen als gering einzustufen. Erstens ist die Fluktuation *pct* für ein Objekt *x* dann klein, wenn die Fluktuation der Erreichbarkeitsdistanz relativ homogen ist, z.B. wenn die *MinPts*-nächsten Nachbarn von *x* zum gleichen Cluster gehören. In diesem Fall sind die Werte für $direct_{\min}, direct_{\max}, indirect_{\min}, indirect_{\max}$ alle fast identisch und damit hat der *LOF*, in Konsitenz mit Lemma (1), einen Wert um 1. Aber auch in dem Fall, dass ein Objekt *x* nicht zu einem Cluster gehört, jedoch seine *MinPts*-nächsten Nachbarn zu ein und demselben Cluster gehören, sind die von Theorem (1) vorhergesagten Werte für die *LOF*-Grenzen geeignet, auch wenn der *LOF* nicht um den Wert 1 liegt.

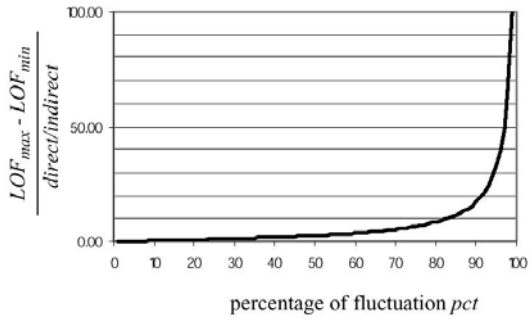


Abbildung 9 - Qualität der Grenzen für *LOF* in Bezug auf statistische Fluktuation

Abbildungsbeschreibung: Für geringe Fluktuationswerte *pct*, welche nur von den indirekten Erreichbarkeitsdistanzen abhängen, bietet Theorem(1) geeignete *LOF*-Grenzen.

Allerdings stellt sich sofort die Frage, ob es Situationen gibt, in denen Theorem (1) keine geeigneten Grenzen für den *LOF* eines Objektes aufzeigt. Ein solcher Fall tritt ein, wenn sich in der *MinPts*-nächsten Nachbarschaft eines Objektes *x* mehrere verschiedene Cluster überlappen, d.h. die *MinPts*-nächsten Nachbarn nicht alle zu demselben Cluster gehören, sondern zu verschiedenen. Abbildung 10 zeigt ein Beispiel mit *MinPts*=6. Hier gehören 6-nächste Nachbarn von *x* zu Cluster 1 und andere zu Cluster 2.

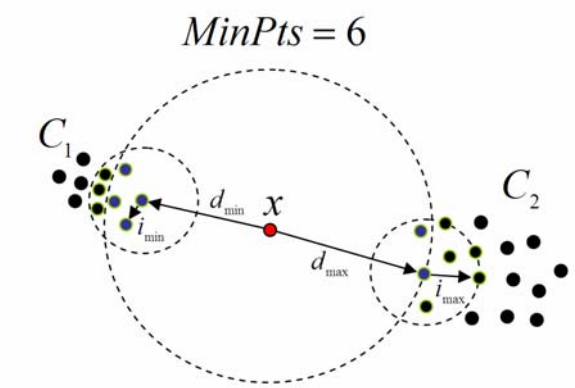


Abbildung 10 - Verschiedene Cluster überlappende *MinPts* Nachbarschaften

Abbildungsbeschreibung: Die direkte 6-Nachbarschaft von *x* überlappt zwei verschiedene Cluster. In dieser Situation liefert Theorem (1) keine geeigneten Grenzen für den *LOF(x)*.

Ein weiteres Theorem (2) versucht durch Generalisierung von Theorem (1) geeignete Grenzen für diesen Fall bereitzustellen. Die Idee dahinter ist, dass die *MinPts*-nächsten Nachbarn in mehrere Gruppen je ihrer Clusterzugehörigkeit partitioniert werden und anteilsmäßig zum *LOF* Wert von *x* beitragen. In Abbildung 10 bilden vier 6-nächste Nachbarn aus Cluster 1 eine Gruppe und zwei 6-nächste Nachbarn aus Cluster 2 eine Gruppe, welche nach der Anzahl der in den Gruppen jeweils enthaltenen Objekte 4:2 zum *LOF(x)* beitragen.

Theorem (2) zu den Grenzen des LOF: Sei *x* ein Objekt aus der Datenmenge *X* mit $1 \le MinPts \le |X|$, und seien $C_1,...,C_n$ die *n* Partitionen der *MinPts*-nächsten Nachbarschaft $N_{MinPts}(x)$ mit $N_{MinPts}(x) = C_1 \cup C_2 \cup ... \cup C_n$ und $C_i \cap C_j = \emptyset$, $C_i \neq \emptyset$ für $1 \le i, j \le n, i \neq j$. Sei weiterhin

$$\xi_i = \frac{|C_i|}{|N_{MinPts}(x)|}$$

der prozentuale Anteil von Objekten in der Nachbarschaft von *x*, welche auch zu C_i gehören. $direct^i_{\min}, direct^i_{\max}, indirect^i_{\min}, indirect^i_{\max}$ seien analog zu Theorem (1) definiert, allerdings auf C_i beschränkt.

Dann gilt:

$$\left(\sum_{i=1}^{n} \xi_i \cdot direct^i_{\min}(x)\right) \cdot \left(\sum_{i=1}^{n} \frac{\xi_i}{indirect^i_{\max}(x)}\right) \leq LOF(x) \leq \left(\sum_{i=1}^{n} \xi_i \cdot direct^i_{\max}(x)\right) \cdot \left(\sum_{i=1}^{n} \frac{\xi_i}{indirect^i_{\min}(x)}\right)$$

Der Beweis zu diesem Theorem kann der Literatur [4] entnommen werden.

Da alle formalen Eigenschaften des *LOF* an gegebenen *MinPts*-Werten aufgezeigt wurden, ist von Interesse, welcher Wert von *MinPts* für die Erzielung bester Ergebnisse mit dem dichtebasierten Outlier Detection Verfahren zu wählen ist. Dazu ist es notwendig, zwei Fragestellungen zu untersuchen. Zum einen: Wie verändert sich der *LOF*-Wert unter der Anpassung von *MinPts*? Und zum anderen: Verändert sich der *LOF*-Wert stetig monoton bei monotoner stetiger Veränderung von *MinPts*? In der Realität ist dies nicht der Fall. Zur Verdeutlichung dieses Umstandes sei auf Abbildung 11 verwiesen, welche die Entwicklung des *LOF* bei steigendem *MinPts* für eine Gauß-Verteilung zeigt.

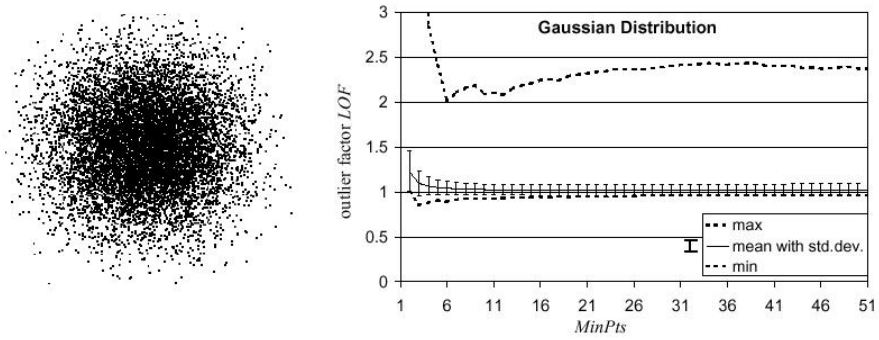


Abbildung 11 - Qualität von *LOF*-Werten bei verändertem *MinPts*

Abbildungsbeschreibung: Schon bei einer so reinen Verteilung, wie der hier gezeigten Gauß-Verteilung, sind starke Schwankungen der *LOF*-Werte bei veränderten *MinPts*-Werten sichtbar. Daher ist *MinPts*=10 als unterste Grenze von den Autoren empfohlen [4].

Bei steigendem *MinPts* Wert werden die Fluktuationen in den Erreichbarkeitswerten geringer, als Folge davon sinkt initial die obere Grenze des *LOF*, um danach wieder anzusteigen und sich dann zu stabilisieren.

Eine vergleichbare, nicht ganz so starke Entwicklung ist auch bei der unteren Grenze des *LOF* zu beobachten. Wenn bereits bei so „reinen" Verteilungen wie der Gauss-Verteilung derartige nicht-monotone Schwankungen des *LOF* in Bezug zu *MinPts* beobachtet werden können, so sind in direkter Schlussfolgerung noch stärkere Schwankungen bei komplexeren Verteilungen zu erwarten. Das Ziel ist also besser die Bestimmung eines geeigneten Intervalls von *MinPts*-Werten als heuristischer Ansatz. Dieses Intervall wird durch die untere und obere *MinPts*-Grenze *MinPtsLB* (lower-bound) und *MinPtsUB* (upper-bound) beschrieben.

Da es von Vorteil ist, ungewollte statistische Schwankungen kleiner *MinPts*-Werte zu vermeiden (siehe auch Abbildung 11), empfehlen Breunig und Kriegel mit einem Wert $MinPts > 10$ als untere Grenze zu starten. Experimente haben gezeigt, dass kleinere *MinPtsLB* Werte zu nicht-intuitiven Beobachtungen bei der Identifizierung von Outliern führen, die sich erst ab einem Wert von 10 stabilisieren. Gleichzeitig ergibt sich für die untere Grenze die Anforderung, dass Outlier auch in dem Fall erkannt werden, dass die dem Outlier naheliegenden Cluster nur wenige Objekte enthalten. Enthalten diese weniger als *MinPts*-Objekte, so wird die *MinPts*-Nachbarschaft der Objekte in einem solchen Cluster auch den Outlier enthalten, der somit nicht unterscheidbar ist. *MinPtsLB* stellt damit auch die minimale Anzahl an Objekten dar, die ein Cluster enthalten muss, damit Outlier ihm gegenüber identifiziert werden können. Generell funktioniert ein *MinPtsLB* Wert zwischen 10 und 20 zufriedenstellend.

Die obere *MinPts*-Grenze stellt eine vergleichbare Assoziation dar, nämlich die maximale Anzahl der Objekte in einer Gruppe (z.B. einem Cluster), welche (noch) als lokale, nahe Outlier-Gruppe von nahe beieinander liegenden Outliern angesehen werden können. Abbildung 12 zeigt hierfür ein Beispiel. S_1 enthält 10 Objekte, S_2 enthält 35 Objekte und S_3 enthält 500 Objekte.

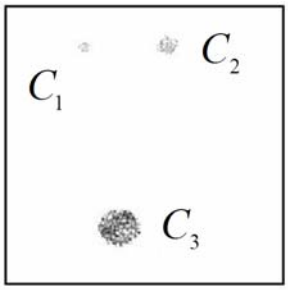


Abbildung 12 - Beispiel für die Bestimmung geeigneter *MinPts*-Werte

Aus der Abbildung wird klar, dass die Objekte in C_3 alle *LOF* Werte um 1 haben und damit keine Outlier sind. Die Objekte in C_1 jedoch sind starke Outlier für *MinPts*-Werte zwischen 10 und 35 und bilden damit eine Outlier-Gruppe zu C_2. Für *MinPts*-Werte zwischen 36 und 45 zeigen C_1 und C_2 ein sehr ähnliches Verhalten. Erst ab einem *MinPts*-Wert von 45 enthalten die Nachbarschaften der Objekte von C_1 und C_2 auch Objekte aus C_3 und beginnen somit einen Outlier Charakter gegenüber C_3 zu entwickeln, sodass sie zusammen als eine Outlier-Gruppe interpretiert werden können. Abhängig von der Anwendung kann nun durch die Festsetzung von *MinPts* entschieden werden, ob eine Gruppe von 35 Objekten, wie z.B. C_2, noch als Gruppe nah beieinander liegender Outlier, oder bereits als Cluster angesehen werden soll. Im ersten Fall dient ein *MinPtsUB*-Wert von größer 35 dazu, C_2 wie eine Gruppe nahe beieinander liegender Outlier zu behandeln, ein *MinPtsUB*-Wert von kleiner 35 behandelt C_2 dagegen bereits als einen Cluster.

Mit der erfolgten Wahl der oberen und unteren Grenze von *MinPts* können nun die *LOF*-Werte innerhalb dieses Intervalls berechnet werden. Dabei empfehlen die Autoren des dichtebasierten Verfahrens, die identifizierten Outlier anhand ihrer maximalen *LOF*-Werte innerhalb des *MinPts*-Intervalls anzuordnen, also basierend auf

$$\max\{LOF(x) \mid MinPtsLB \le MinPts \le MinPtsUB\}.$$

Diese Anordnung anhand der Maxima wird empfohlen, weil die Ordnung nach anderen Maßen (z.B. Minimum oder Durchschnitt) den Outlier Charakter überdecken, im Zweifelsfall jedoch zumindest abschwächen könnte [4].

Abbildung 13 zeigt ein Beispiel der Anwendung des Verfahrens anhand einer Beispieldatenmenge und einem *MinPts*-Wert von 40. Auf der *z*-Achse sind die *LOF*-Werte der Objekte entsprechend abgetragen. Für die Berechnung wird ein zweistufiger Algorithmus vorgeschlagen. Im ersten Schritt werden die *MinPtsUB*-nächsten Nachbarschaften identifiziert und im zweiten Schritt werden die entsprechenden *LOF* Werte berechnet.

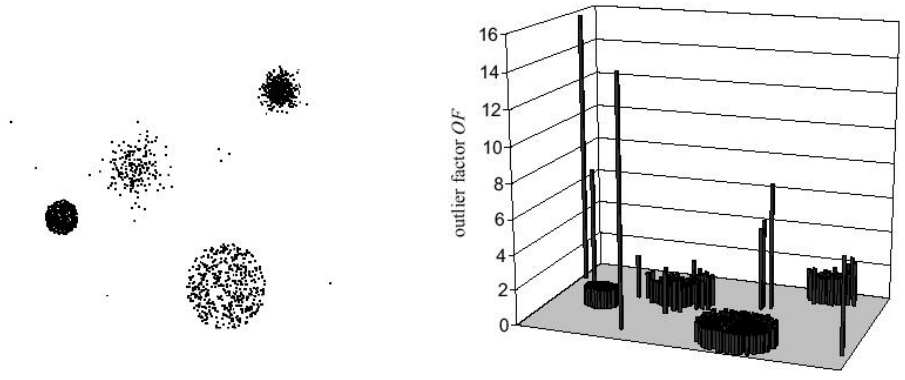


Abbildung 13 - Anwendungsergebnisse des *LOF* Verfahrens

Abbildungsbeschreibung: Für die unterschiedlichen Cluster und intuitive Outlier ist hier der Outlier-Faktor (*LOF*) im Diagramm gezeigt und eine Erkennung der Outlier anhand ihres Faktors wird deutlich nachvollziehbar.

In Schritt 1 des Algorithmus werden also zunächst die *MinPtsUB*-nächsten Nachbarn $N_{MinPtsUB}(x)$ für jedes Objekt x der Datenmenge $X \mid n = |X|$ zusammen mit ihrer Entfernung zu x in einer Datenbank M zusammengefasst, welche einen Umfang von *n*MinPtsUB* Instanzen hat. Die Größe dieser Datenbank ist unabhängig von der Dimensionalität der ursprünglichen Datenmenge. Die Komplexität dieses Schrittes ist mit $O(n \cdot t_{knn})$ angegeben, wobei t_{knn} die benötigte Zeit für eine *k-nn* Abfrage ist. *k-nn* steht hier für *k*-nearest-neighbors (*k*-nächste Nachbarn). Für Datenmengen von wenigen Dimensionen kann hier eine Grid-basierte Abfrage mit konstantem Aufwand eingesetzt werden, sodass die Gesamtkomplexität von Schritt 1 bei $O(n)$ liegt. Für eine Datenmenge mit einer mittleren bis hohen Anzahl an Dimensionen kann ein Index (z.B. ein X-Tree [53]) mit einer Abfragezeit für *k-nn* Anfragen von $O(\log n)$ verwendet werden, sodass sich ein Gesamtaufwand von $O(n \log n)$ für Schritt 1 ergibt. Für eine extrem hohe Zahl m an Dimensionen in der Datenmenge ist ein sequentieller Scan, z.B. ein VA-File [54], notwendig, das eine Komplexität von $O(n)$ bietet, so dass der Gesamtaufwand für den ersten Schritt auf $O(n^2)$ steigt. In den Experimenten zu dichtebasierenden Outlier-Detection Methoden weist Breuning mit den Co-Autoren nach, dass X-Tree für bis zu 5-dimensionale Datenmengen gute Ergebnisse liefert, die Performance jedoch ab 10 bzw. 20 Dimensionen erheblich nachlässt, was auf grundlegende Eigenschaften von Index-Strukturen zurückzuführen ist [4].

In Schritt 2 des Algorithmus werden die *LOF*-Werte anhand der Datenbank *M* errechnet, ohne dass die originale Datenmenge dazu notwendig wäre, da ausreichende Informationen in *M* enthalten sind. *M* wird für jeden *MinPts* Wert im Intervall zwischen *MinPtsLB* und *MinPtsUB* zweimal gescannt. Zuerst werden für alle Objekte die lokalen Erreichbarkeits-Dichten *lrd* ermittelt und dann daraus die finalen *LOF*-Werte berechnet. Anhand dieser Werte können die Objekte nach ihrem maximalen Outlier-Faktor geordnet werden. Der Aufwand für Schritt 2 beträgt $O(n)$, was auch experimentell unterlegt wurde.

In ihren Ausführungen zu weitergehenden Studien geben Breuning, Kriegel, Ng und Sandner vor allem an, Gründe für den Outlier-Charakter der identifizierten Objekte zu suchen, vor allem in Bezug auf vergleichbare Bestrebungen unter homogener Nutzung aller Dimensionen zur Erkennung von Outliern [55].

Außerdem wird von Breuning et al die gedankliche Verzahnung der Outlier-Erkennung mit der Identifizierung von Clustern angeführt. Dabei besteht die Möglichkeit, Clustering als einen Schritt zur Informationsgewinnung zu nutzen, um anhand von Informationen über die Cluster (Dichte, Position, Kardinalität, etc.) geeignete *MinPts* Grenzen für den Outlier-Detection Schritt zu wählen, aber auch um Datenstrukturen gemeinsam zu nutzen und damit Rechenzeit einzusparen und die Algorithmen effizienter zu gestalten.

2.5.2. Top-*n* Local Outlier

Von den Autoren Jin, Tung und Han [5] wird ein erweiterter Ansatz zu lokalen, dichtebasierten Outliern vorgestellt, der sich vor allem auf eine Optimierung der Verfahren zur Identifizierung solcher Outlier bezieht. Sie schätzen das von Breuning, Kriegel, Sandner und Ng vorgestellte Konzept der lokalen Grade für Objekte, welche deren Outlier-Potential anhand einer dichtebasierten Analyse beschreiben, als sehr nützlich ein. Sie verweisen jedoch darauf, dass der Aufwand der Berechnung der Faktoren (*LOF*) für alle Objekte einer Datenmenge sehr hoch ist. Demgemäß schlagen sie eine Optimierung vor, welche die Suche auf eine gewisse Zahl an top-*n* Outliern beschränkt. Dadurch werden nur für Objekte mit einem angenommenen starken Outlier-Charakter diese Faktoren berechnet. Dem liegt die Überlegung zugrunde, dass in einer Datenmenge die Mehrzahl der Objekte intuitiv eben keine Outlier sind und daher die Berechnung der lokalen Outlier-Faktoren dies lediglich bestätigen, aber ansonsten keinen weiteren Wissenszuwachs generieren würde.

Somit ergibt sich die folgende Problemstellung: Alle *LOF* Werte für alle Objekte zu berechnen impliziert die Berechnung für eine Mehrheit an Objekten, welche keine Outlier sind. Das entspricht direkt einer Verschwendung wertvoller Resourcen, z.B. in Bezug auf die aufgewandte Rechenzeit. Allerdings kann bei lokalen Outliern der Outlier-Charakter nicht einfach „global“ ausgeschlossen werden. Ein einfaches Beschneiden der zu untersuchenden Datenmenge ist also nicht möglich. Die Löschung eines Datenpunktes aus der betrachteten Menge durch eine solche Beschneidung hat direkten Einfluss auf die Dichtewerte der Nachbarschaft dieses Punktes bzw. Objektes. Daher kann ohne Kenntnis über die Lage möglicher Outlier die Berechnung der Dichteinformation nicht effektiv beschnitten werden. Ohne die Berechnung der Dichteinformation wiederum ist nicht zu ermitteln, wo sich die möglichen top-*n* Outlier befinden.

Dieses „Henne-Ei-Problem“ lösen Jin, Tung und Han durch die Vorstellung eines Verfahrens, welches die Datenmenge in so genannten Microclustern [60] komprimiert und eine effiziente Schätzung der oberen und unteren Grenzen für die *LOF* Werte für jedes Objekt in der Datenmenge erlaubt. Durch den Vergleich der *LOF*-

Grenzen der Objekt-Microcluster kann nun erkannt werden, welche Objekte nicht für die Reihe der top-*n* Outlier in Frage kommen. Daher wird im Ergebnis die Berechnung der *LOF*-Werte auf die top-*n* Outlier Kandidaten beschränkt.

Die zentrale Idee hinter dem vorgeschlagenen Algorithmus ist das Schätzen der Entfernung zwischen einem Objekt *x* und einer Gruppe von Objekten in einem Microcluster. Der Microcluster wird durch einen Kreis (*Anm. des Autors:* pro Attribut) mit dem Mittelpunkt im Mittelwert der Punkte der Objekte der Microclustergruppe beschrieben. Wenn allerdings das Objekt *x* im Kreis des Microclusters liegt, wird die Entfernungsschätzung sehr ungenau. Die Lösung hiefür bietet der Einsatz einer speziellen Schnittebenen-Methode zur Bestimmung der Grenze zwischen dem Datenobjekt und dem Microcluster.

Definition lokaler dichtebasierter Outlier analog zu [4]: Lokale dichtebasierte Outlier werden für ein Objekt *x* in einer Datenmenge *X* mit den Begriffen *k*-distance $k_d(x)$, *k*-distance Nachbarschaft $N_k(x)$, Erreichbarkeitsdistanz $rd_k(x,x')$, lokale Erreichbarkeitsdichte $lrd_k(x)$ und lokaler Outlier Faktor $LOF_k(x)$ analog zu Kapitel 2.5.1 formal definiert, sodass diese Definitionen hier nicht nochmals gesondert aufgeführt werden.

Dabei ist der lokale Outlierfaktor *LOF* der Durchschnitt der Rate der Erreichbarkeitsdichte von den *k*-nächsten Nachbarn von *x* und der eigenen Erreichbarkeitsdichte von *x*. Intuitiv wird der *LOF(x)* groß, wenn die lokale Erreichbarkeitsdichte von *x* kleiner ist, als die der besagten Nachbarn.

Da die von den Autoren getroffene Beschreibung des Algorithmus zur Erkennung der top-*n* dichtebasierten lokalen Outlier einige Grunderkenntnisse über die Natur dieser Outlier formal sehr gut vertieft und damit einen Einblick in diese Zusammenhänge liefert, wird dies hier ausführlicher dargestellt. Die Größen *n* und *k* werden vom Anwender bestimmt und für das Verfahren vorgegeben. Nach einer detaillierten Analyse des Problems des Findens sinnvoller Grenzen für die *LOF*-Werte können diese bestimmt werden, wenn obere und untere Grenzen für die lokalen Erreichbarkeitsdichten vorliegen.

Theorem (1) zu oberen und unteren Grenzen der lokalen Erreichbarkeitsdichte: $lrd_k(x').lower$ und $lrd_k(x').upper$ seien die untere und obere Grenze der lokalen Erreichbarkeitsdichte eines Objektes $x' \in N_k(x)$ genau dann, wenn

$$\frac{\min(lrd_k(x').lower)}{lrd_k(x).upper} < LOF_k(x) < \frac{\max(lrd_k(x').upper)}{lrd_k(x).lower}.$$

Beweis: Der *LOF* ist der Durchschnitt der Raten von $lrd_k(x')_{o \in N_k(x)}$ und $lrd_k(x)$ [4]. Der durchschnittliche Quotient muss also größer sein als der Quotient mit der minimalen Erreichbarkeitsdichte von den *k*-nächsten Nachbarn von *x* im Zähler und der maximalen Erreichbarkeitsdichte von *x* im Nenner. Gleichsam muss er kleiner sein als der Quotient mit der maximalen Erreichbarkeitsdichte von den *k*-nächsten Nachbarn von *x* im Zähler und der minimalen Erreichbarkeitsdichte von *x* im Nenner.

Schlussfolgerung: Für ein Objekt *x* und seine *k*-nächsten Nachbarn $x' \in N_k(x)$ ist die Erreichbarkeitsdichte folgendermaßen begrenzt:

$$\frac{1}{\max\{rd_k(x,x')\}} \le lrd_k(x) \le \frac{1}{\min\{rd_k(x,x')\}}.$$

Weil die lokale Erreichbarkeitsdichte lt. Definition des Begriffs der Kehrwert des Durchschnitts der Erreichbarkeitsdistanz ist, ist er folglich kleiner als der Kehrwert des Maximums und größer als der Kehrwert des Minimums der Erreichbarkeitsdistanz zwischen *x* und seinen *k*-nächsten Nachbarn.

Theorem (2) zu oberen und unteren Grenzen der Erreichbarkeitsdistanz: Sei $k_d(x').lower$ und $k_d(x').upper$ die untere und obere Grenze der *k*-distance $k_d(x')$ eines Objektes x'. Und sei $d(x',x).lower$ und $d(x',x).upper$ die untere und obere Grenze der Entfernung zwischen zwei Objekten *x* und x', so gilt:

$$\max(d(x',x).lower, k_d(x').lower) \le rd_k(x,x') \le \max(d(x',x).upper, k_d(x').upper).$$

Dies ergibt sich aus der Definition der Erreichbarkeitsdistanz [4]. Es zeigt sich im Ergebnis der Analyse, dass die Berechnung einer oberen und unteren Grenze des *LOF* von allen Punkten einer Datenmenge von einem effizienten Weg abhängt, obere und untere Grenzen der Distanzen zwischen Punkten und obere und untere Grenzen der *k*-distance Werte dieser Punkte zu bestimmen.

Weiterhin wird hier kurz in das Konzept der Microcluster eingeführt, da diese in der folgenden Betrachtung des Verfahrens und der Algorithmen eine zentrale Rolle spielen.

Definition von Microclustern: *MC(n,c,r)* ist als „Microcluster“ eine summarische Repräsentation einer Gruppe von Objekten $x_1,...x_n$, welche sich so eng aneinander befinden, dass sie wahrscheinlich zum selben Cluster gehören.

$$c = \frac{\sum_{i=1}^{n} x_i}{n}$$

sei (vereinfacht) das Mittelwertzentrum und $r = \max\{d(x_i, c)\}_{i=1,...n}$ der Radius des Microclusters, wobei dieser *n* Objekte enthalten soll. Zur Vermeidung zu geringer Genauigkeit durch den Einsatz von Microclustern wird dieser durch einen vom Nutzer vorgegebenen Wert max*radius* begrenzt. Wird ein *MC* mit $r > \max radius$ entdeckt, so wird dieser geteilt. Dafür werden die zwei voneinander am weitesten befindlichen Objekte als neue Samen der durch Teilung entstehenden Microcluster definiert und die Objekte des ursprünglichen Microclusters werden dem jeweils naheliegensten Samen zugeordnet.

Zuletzt ist es notwendig, die Entfernungsmessung zu einem Microcluster zu definieren. Sei *X* eine Datenmenge und *M* eine Menge von Microclustern. Ein Punkt in der *m*-dimensionalen Datenmenge ist bezeichnet als $x(x_1,...x_m)$ und ein Microcluster ist bezeichnet mit *MC(n,c,r)*.

Theorem zur minimalen und maximalen Entfernung eines Objektes zu einem Microcluster: Sei *x* ein Objekt und *MC(n,c,r)* ein Microcluster, so sei in dem Falle, dass *x* weiter als die Distanz *r* von *c* entfernt liegt, die minimale Entfernung zwischen *x* und *MC* definiert als $d_{\min}(x, MC) = d(x, c) - r$ und die maximale Entfernung zwischen *x* und *MC* definiert als $d_{\max}(x, MC) = d(x, c) + r$. Damit liegt jeder Punkt in *MC* mindestens $d_{\min}$ und höchstens $d_{\max}$ von *x* entfernt. Abbildung 14 zeigt diesen Fall, in dem sich *x* und *MC* nicht überlappen.

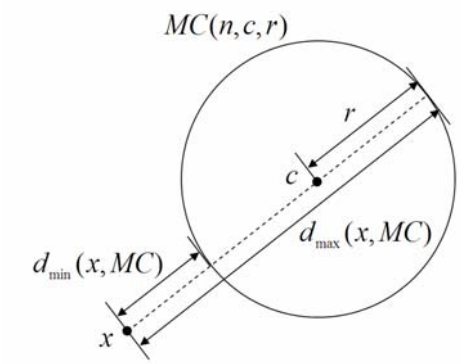


Abbildung 14 - Microcluster ohne Überlappung mit einem Objekt *x*

Abbildungsbeschreibung: Ohne Überlappung eines Microclusters (*MC*) mit einem Objekt *x* liegt jeder Punkt in *MC* nach Theorem (3) *mindestens* $d_{\min}$ und höchstens $d_{\max}$ von *x* entfernt.

Im Fall, dass sich *x* und *MC* überlappen, also wenn *x* innerhalb der Entfernung *r* von *c* liegt, bleibt zwar die maximale Distanz zwischen *MC* und *x* unverändert, die minimale Distanz muss jedoch auf Null gesetzt werden. Dies führt aber zur Messung von extrem hohen Dichten, was für die Outlier Erkennung nicht wünschenswert ist. Dieses Problem kann gelöst werden, indem sichergestellt wird, dass jedes Objekt in einem Microcluster *MC* näher an *c* als an den Zentren aller anderen Microcluster liegt. Das wird durch folgendes Vorgehen erreicht:

1. Identifizierung aller Microcluster mit einem geeigneten Algorithmus, z.B. BIRCH [60]
2. Fixieren der Zentren aller Microcluster
3. Neuzuordnung aller Objekte zu den nächsten neuen Zentren

Unter dieser Annahme kann nun ein Schnittebenen-Konzept eingeführt werden.

Definition der Schnittebene zwischen zwei Microclustern: Seien MC_i und MC_j zwei Microcluster, so ist die Schnittebene (cut plane) $cp(MC_i, MC_j)$ eine Hyperebene, welche senkrecht auf der Linie zwischen den Zentren der Microcluster c_i und c_j steht und diese exakt in zwei Hälften teilt. Dies wird in Abbildung 15 illustriert.

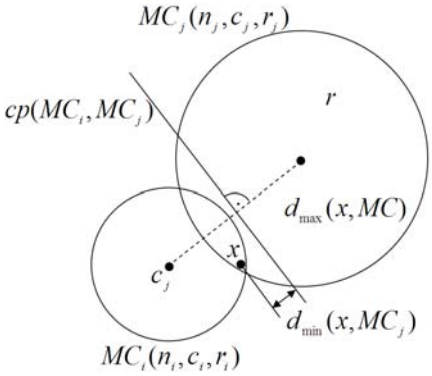


Abbildung 15 - Schnittebenenkonzept für Microcluster

Abbildungsbeschreibung: Das Schnittebenenkonzept für Microcluster wird gezeigt, wobei die minimale Entfernung zwischen einem Objekt und dem Microcluster, zu dessen Zentrum das Objekt weiter entfernt ist, als zu dem anderen überlappenden Microcluster, eingeführt wird.

Definition der minimalen Entfernung zwischen einem Objekt und einem Microcluster: Sei *x* ein Objekt in einem Microcluster $MC_i(n_i, c_i, r_i)$. Wenn *x* sich innerhalb des Radius r_j zu einem Microcluster $MC_j(n_j, c_j, r_j)$ befindet, so sei $d(x, cp(MC_i, MC_j))$ die senkrecht auf $cp(MC_i, MC_j)$ stehende Entfernung zwischen *x* und $cp(MC_i, MC_j)$. Dies sei die minimale Entfernung $d_{\min}(x, MC_j)$ zwischen *x* und MC_j. Diese Definition basiert auf der Annahme, dass *x* näher an c_i als an c_j liegt. Dies wird aus Abbildung 15 ersichtlich.

Nun muss noch die minimale und maximale Entfernung zwischen zwei Microclustern MC_i und MC_j definiert werden.

Definition der minimalen und maximalen Entfernung zwischen zwei Microclustern: Bei gegebenen Microclustern $MC_i(n_i, c_i, r_i)$ und $MC_j(n_j, c_j, r_j)$ sei die minimale und maximale Entfernung zwischen diesen definiert als: $d_{\min}(MC_i, MC_j) = d(c_i, c_j) - (r_i + r_j)$ und $d_{\max}(MC_i, MC_j) = d(c_i, c_j) + (r_i + r_j)$. Diese Definition gilt unter der Annahme, dass es keine Überlappung gibt. Bei Überlappung gilt, dass $d_{\min}(MC_i, MC_j) = 0!$ ist. Dies ist in Abbildung 16 aufgezeigt (wobei in dieser *d* mit „Dist“ bezeichnet ist).

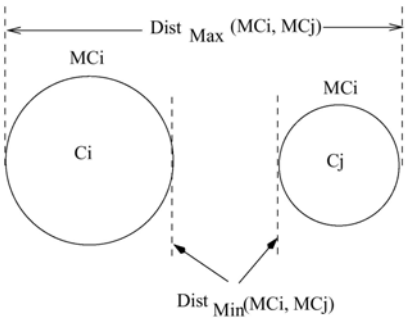


Abbildung 16 - Minimale und maximale Entfernungen zwischen Microclustern

Abbildungsbeschreibung: Die maximale Entfernung zwischen zwei nicht überlappenden Microclustern ist definiert als die Summe der Entfernung zwischen den Zentren und den jeweiligen Radien der Microcluster. Die minimale Entfernung ist definiert als die Entfernung zwischen den Zentren abzüglich dieser jeweiligen Radien.

Hieraus kann die folgende **Schlussfolgerung** gezogen werden: Sei *x* ein Objekt und *MC(n,c,r)* ein Mircocluster mit $x \in MC(c, n, r)$ und seien $MC_1(n_1, c_1, r_1), ...MC_l(n_l, c_l, r_l)$ eine Menge von Microclustern, welche potentiell die *k*-nächsten Nachbarn von *x* enthalten könnten. Zur Vereinfachung seien alle anderen *n-1* Objekte

x'_i selbst Microcluster mit $MC_i(1, x'_i, 0)$, also jeweils einem Objekt, welches gleichzeitig das Zentrum des Microclusters ist und mit einem Radius Null. Damit gibt es $l+n-1$ Microcluster.

1. Sei $\{d_{\min}(x, MC_1), ...d_{\min}(x, MC_{l+n-1})\}$ in wachsender Reihenfolge sortiert, dann ist die untere Grenze der *k*-Distanz $k_d(x)$ gegeben mit $k_{\min}(x) = d_{\min}(x, MC_i)$ mit $n_1 + \cdots + n_{i-1} < k \leq n_1 + \cdots + n_i$.

2. Sei $\{d_{\max}(x, MC_1), ...d_{\max}(x, MC_{l+n-1})\}$ gleichsam in wachsender Reihenfolge sortiert, dann ist die obere Grenze der *k*-Distanz $k_d(x)$ gegeben mit $k_{\max}(x) = d_{\max}(x, MC_i)$ mit $n_1 + \cdots + n_{i-1} < k \leq n_1 + \cdots + n_i$.

Definition der internen Erreichbarkeitsgrenzen von Microclustern: Die interne obere und untere Erreichbarkeitsgrenze eines Microclusters *MC(n,c,r)* sei unter der Annahme, dass *MC* die Objekte $x_1, ...x_n$ enthalte, wie folgt definiert:

1. $rd_{\max}(MC) = \max(2r, \max(k_{\max}(MC)))$ mit $k_{\max}(MC) = \max(k_{\max}(x_1), ...k_{\max}(x_n))$
2. $rd_{\min}(MC) = k_{\min}(MC)$ mit $k_{\min}(MC) = \min(k_{\min}(x_1), ...k_{\min}(x_n))$

Intuitiv kann dies so interpretiert werden, dass, bei zwei gegebenen Objekten *x* und x' in einem Microcluster *MC*, $rd_{\min}(MC)$ und $rd_{\max}(MC)$ die jeweils untere bzw. obere Grenze für $rd(x, x')$ repräsentieren. Dies wird benutzt, um die Grenzen der Erreichbarkeitsdistanz eines Paares von Objekten innerhalb eines Microclusters abzuschätzen.

Definition der externen Erreichbarkeitsgrenzen von Microclustern: Die externe obere und untere Erreichbarkeitsgrenze eines Microclusters MC_i in Bezug auf einen anderen Microcluster MC_j unter Annahme gleicher Definitionen für die Bezugsgrößen wie vorher sei wie folgt definiert:

1. $rd_{\max}(MC_i, MC_j) = \max(d_{\max}(MC_i, MC_j), k_{\max}(MC_j))$
2. $rd_{\min}(MC_i, MC_j) = \max(d_{\min}(MC_i, MC_j), k_{\min}(MC_j))$

Intuitiv repräsentieren für zwei Objekte $x \in MC_i$ und $x' \in MC_j$ die Werte $rd_{\max}(MC_i, MC_j)$ und $rd_{\min}(MC_i, MC_j)$ jeweils die obere bzw. untere Grenze für $rd(x, x')$. Dies wird benutzt, um die Grenzen der Erreichbarkeitsdistanz eines Paares von Objekten in verschiedenen Microclustern abzuschätzen. Diese externen Grenzen werden von einer eventuellen Überlappung nicht beeinflusst, sofern es eine gute Schätzung der *k*-Distanz-Grenzen innerhalb eines Microclusters gibt.

Im Folgenden wird der Algorithmus zur Erkennung der top-*n* Outlier, welcher auf dem Konzept der Microcluster basiert, beschrieben. Er besteht generell aus drei Schritten, dem Preprocessing, der Berechnung der *LOF*-Grenzen für die Microcluster und dem Ordnen der top-*n* lokalen Outlier.

Im Preprocessing Schritt werden zur effizienten Bestimmung der k-Distanz-Grenzen $k_{\max}$ und $k_{\min}$ die Daten durch einen sequentiellen Scan in einen CF-Tree [60] geladen. Am Ende des Prozesses werden die Zentren aller CFs in einen speicherbasierten X-Tree [53] geladen. In einem zweiten Unterschritt werden nun durch einen zweiten sequentiellen Scan der Daten die Objekte zu den nächstgelegenen Zentren neu zugeordnet, welche vorher berechnet wurden. Gleichzeitig werden die Anzahl der Objekte *n* und der Radius *r* eines jeden Microclusters MC aufgezeichnet. Im dritten Unterschritt werden die Microcluster in einen eigenen X-Tree eingeordnet.

Der zweite Schritt zur Berechnung der *LOF*-Grenzen für die Microcluster benötigt einen Scan durch die Datenbank und einen durch die Microcluster. Mit dem Datenbank Scan werden die Grenzen $k_d(MC) \rightarrow k_{\max}(MC)$ und $k_d(MC) \rightarrow k_{\min}(MC)$ geschätzt. Der zweite Scan durch die Microcluster ermittelt die geschätzten Grenzen für die Werte von $rd(MC)$, also für die Erreichbarkeitsdistanzen der Microcluster (sowie für die von deren benachbarten Microclustern). Hier kommt zum effektiven Beschneiden die Cut-Plane Lösung innerhalb der Algorithmen zum Einsatz.

Im dritten Schritt werden die top-*n* Outlier unter Ausnutzung der Kenntnisse über die oberen und unteren *LOF*-Grenzen aller Microcluster geordnet.

Alle Algorithmen sind ausführlicher in der Literatur [5] beschrieben. Die Autoren führen keinen formellen Beweis über die Komplexität ihrer vorgeschlagenen Algorithmen, bieten aber einen empirischen, auf

Experimenten mit synthetischen Datenmengen basierenden Überblick über die Performance im Vergleich zur Berechnung aller *LOF*-Werte einer Datenmenge nach der klassischen Methode [4] unter Nutzung eines X-Tree. Es wird auch auf Experimente mit realen Daten verwiesen, diese werden jedoch nicht vorgestellt. Generell arbeitet die Methode deutlich effizienter, vor allem bei einer wachsenden Zahl von Dimensionen, und bei einer steigenden Zahl an Testdaten legen die Ergebnisse einen linear steigenden Aufwand nahe. Die Belastbarkeit dieses Eindrucks kann jedoch in dieser Arbeit nicht näher beurteilt werden.

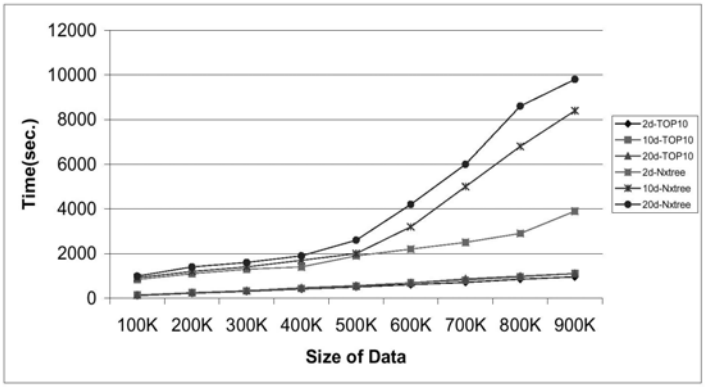


Abbildung 17 - Experimentelle Ergebnisse des top-*n* *LOF* Verfahrens

Abbildungsbeschreibung: Die Ergebnisse des Experiments zeigen robuste Performance insbesondere gegenüber einer steigenden Zahl an Dimensionen.

Abbildung 17 zeigt grafisch die Ergebnisse der Experimente. Es werden die klassischen *LOF*-Berechnungen mit X-Tree und jeweils 2, 10 und 20 Dimensionen für Datenmengen mit 100.000 bis zu nahe 1 Million Elementen bzw. Objekten dargestellt. Hier ist ein exponentieller Verlauf abzulesen. Demgegenüber erweisen sich die top-*n* Experimente zur Erkennung der Top-10 lokalen Outlier als sehr robust, insbesondere bzgl. der steigenden Anzahl von Dimensionen. Nicht ersichtlich aus den Quellen ist jedoch, ob die Verfahren jeweils dieselben Outlier gefunden haben. Auch wird keine Aussage über die Wahl von *MinPts*-Werten gemacht.

Als zukünftiger Fokus der Forschung im Bereich der top-*n* lokalen Outlier Ansätze werden von Jin, Han und Tung das Finden von starken Outlier Gruppen sowie das rekursive Finden von lokalen Outliern bei unterschiedlicher Granularität der betrachteten Datenräume vorgeschlagen.

2.6. Outliererkennung durch Dichtemessung in Projektionen

Aggarwal und Yu [68] stellen in ihrem Ansatz für die Erkennung von Outliern in hochdimensionalen Räumen viele der bisher vorgestellten Ansätze unter spezifischen, auf die Anzahl der Dimensionen bezogenen Anforderungen, auf den Prüfstand. Dabei kommen Sie unter der Voraussetzung, dass die meisten Anwendungen für Outlier Detection mit vielen, z.B. einigen hundert Dimensionen, arbeiten, zu einem sehr interessanten Ergebnis. Die meisten Algorithmen nutzen das Konzept der absoluten oder relativen Nähe von Objekten zueinander, um Outlier durch die Betrachtung ihrer Beziehung zum Rest der Datenmenge zu erkennen. Bei einer wachsenden Zahl an Dimensionen wird die Anordnung von Daten im Raum natürgemäß jedoch schnell spärlicher und es ist fraglich, ob die verwendeten Nähebegriffe ihre Aussagekraft behalten. Die Spärlichkeit von Daten in hochdimensionalen Räumen legt nahe, dass fast alle Objekte gleichsam als Outlier in Frage kommen, wenn diese Charaktereigenschaft aufgrund von räumlicher Nähe zugeordnet wird. Durch diese Entwicklung wird das Finden von sinnvollen und aussagekräftigen Outliern substantiell komplexer und ist nicht mehr so offensichtlich, wie dies in Räumen mit einer niedrigen Zahl an Dimensionen der Fall ist. Die Autoren schlagen daher die Betrachtung von Datenprojektionen auf Subräume vor, in der das Verhalten von Objekten studiert werden kann.

Ein Outlier ist gemäß Aggarwal et al definiert als ein Datenpunkt, welcher sich vom Rest der Datenmenge aufgrund eines gewissen Maßes unterscheidet. Dieser Datenpunkt enthält in vielen Fällen nützliche Informationen über anormales Verhalten in einem System, welches durch die Datenmenge beschrieben ist. Viele der vorgestellten Outlier Detection Methoden (vgl. Kapitel 2.10) machen eine implizite Annahme dahingehend, dass die Dimensionalität der Datenmengen relativ gering ausfällt. Folgerichtig ergeben sich daraus Probleme, sobald diese Methoden auf hochdimensionale Räume angewandt werden.

> *Anmerkung:* Diese Probleme werden i.d.R. auch von den Autoren der entsprechenden Ansätze eingeräumt. In Bezug auf andere Ansätze werden entweder Optimierungen in den Algorithmen vorgeschlagen, um hochdimensionale Räume und in diesen die Outliererkennung zu bewältigen, oder es werden Kompromisse gesucht, welche eine effektive Beschneidung der Datenmenge oder der Anzahl der zu findenden Outlier erlauben, ohne dass die Autoren davon ausgehen, dass die Ergebnisse der Outlier-Suche zu stark kompromittiert werden. Meist jedoch werden am eigentlichen Ansatz, d.h. am (statistischen) Maß selbst, keine tiefgreifenden Änderungen vorgenommen. Das hier vorgestellte Verfahren wird deshalb so detailliert beschrieben, weil die theoretischen Beschreibungen zum Verständnis verschiedener Anwendungsdomainen für die Outliererkennung – unabhängig von der Einschätzung der Praktikabilität dieses Ansatzes – sehr hilfreich sind.

Viele Data Mining Algorithmen definieren Outlier als Nebenprodukt der Clustering Verfahren (vgl. auch Kapitel 2.8). Somit werden Outlier als solche Datenpunkte definiert, die nicht in Clustern liegen. Implizit definieren diese Techniken Outlier als Hintergrundrauschen, aus welchem die Cluster hervorstehen.

Eine andere Klasse von Techniken definiert Outlier als Punkte, welche weder Teil eines Clusters sind, noch Teil des Hintergrundrauschens. Sie werden als spezifische Punkte definiert, welche sehr von der Norm des Verhaltens von Punkten in einem System abweichen. Diese Form der Definition wird von Aggarwal und Yu als sehr viel nützlicher beschrieben, weil die charakteristischen Abweichungen vom Normalverhalten und deren Diagnose es erlauben, Anhaltspunkte für Zusammenhänge und Abhängigkeiten anormalen Verhaltens in der zugrundeliegenden Anwendung zu finden und entsprechende Schlussfolgerungen zu ziehen. Aufgrund dessen bezieht sich der nun vorgestellte Ansatz auf die Outliererkennung anhand von Abweichungswerten.

Die Autoren beschreiben zusätzlich zum allgemeinen Ansatz noch einmal die Abgrenzung zu anderen Verfahren, um ihr Vorgehen zu motivieren. Bezugnehmend auf die statistischen Verfahren zur Outliererkennung (vgl. auch Kapitel 2.1), haben diese den Nachteil, dass es mit einer steigenden Zahl an Dimensionen sehr schwierig wird, die zugrundeliegende Verteilung der Daten zu bestimmen, um ein passendes statistisches Verfahren anzuwenden.

> *Anmerkung:* Bei solchen statistischen Verfahren, welche das Wissen um eine Verteilung nicht erfordern, ist in Kapitel 2.3 bereits ausgeführt, dass allein der Aufwand der Berechnung konvexer Hüllenebenen diese Verfahren für eine hohe Zahl an Dimensionen nicht geeignet macht.

Die entfernungsbasierten Ansätze von Knorr und Ng [3] (*DB(p,D)*-Outlier vgl. Kapitel 2.4.1) und Ramaswamy, Rastogy und Shim [13] (Outlier auf Basis der Entfernung zu den *k*-ten nächsten Nachbarn, vgl. Kapitel 2.4.2), welche die auf alle Dimensionen bezogene Entfernung zwischen zwei Punkten nutzen, um Outlier zu erkennen, leiden direkt unter den Nachteilen einer hohen Anzahl an Dimensionen.

Bei den *DB(p,D)*-Outliern befinden sich gemäß Definition nicht mehr als die Anzahl von *p* Datenpunkten in einer Entfernung kleiner oder gleich *D* von einem Outlier. Dieses Verfahren ist naturgemäß sehr sensibel gegenüber dem Parameter *D*, welcher vom Nutzer bestimmt werden soll. Dies ist apriori sehr schwierig, weshalb auch Knorr und Ng ein interatives Verfahren zur Annäherung an passende Wertekombinationen von *p* und *D* vorschlagen, sowie auf die Notwendigkeit für gute Startwerte für diese Iteration verweisen. Wenn sich die Zahl der Dimensionen erhöht, wird es mit steigender Tendenz schwieriger, *D* geeignet zu wählen, weil die meisten Punkte mit einer hohen Wahrscheinlichkeit in einer dünnen Hülle um alle anderen Punkte liegen. Wird nun *D* geringfügig kleiner als der durchschnittliche Hüllenradius gewählt, sind fast alle Punkt Outlier. Wird *D* geringfügig höher als der durchschnittliche Hüllenradius gewählt, sind gegebenenfalls gar keine Punkte Outlier. Daher müsste *D* mit hoher Genauigkeit gewählt werden, um eine moderate Anzahl an Punkten als sinnvolle Outlier zu identifizieren. Abgesehen davon tendieren Daten aus realen Anwendungen zu einem starken Rauschen und anormale Abweichungen wären nur in Subräumen mit einer niedrigeren Zahl an Dimensionen offensichtlich erkennbar. Diese eingebetteten Abweichungen könnten mit einem Verfahren, welches auf dem gesamten Dimensionsraum arbeitet, nur ungleich schwerer bzw. nicht erkannt werden.

Die Arbeiten zu Outliern in Bezug auf die *k*-ten nächsten Nachbarn haben zwar gegenüber den *DB(p,D)*-Outliern eine Reihe von Vorteilen, aber auch dieses Verfahren ist nicht für eine hohe Zahl an Dimensionen geschaffen. Allein durch die hohe Zahl an Dimensionen steigt die Wahrscheinlichkeit, dass sich Punkte in sehr ähnlichen Abständen zueinander befinden.

Das von Breuning, Kriegel, Ng und Sandner [4] vorgestellte *LOF*-Verfahren identifiziert Outlier anhand eines lokalen Dichtemaßes (vgl. auch Kapitel 2.5.1). Hier wird die lokale Erreichbarkeitsdichte unter Verwendung einer geglätteten Durchschnittsdistanz zu Punkten in der lokalen Umgebung eines betrachteten Objektes ermittelt. Dieses Verfahren ist sehr aufwändig, da bei steigender Zahl an Dimensionen die Messung der lokalen Dichte aufgrund der Spärlichkeit der Daten schwierig durchzuführen ist. Wenn also kein sinnvolles

Konzept lokaler Dichte in spärlich besetzten Räumen (hoher Dimension) existiert, sind die resultierenden Outlier wahrscheinlich von geringerer Aussagekraft.

Diese Techniken nutzen also in der einen oder anderen Weise (homogen) die gesamte Zahl an Dimensionen zur Definition der Verfahren zur Erkennung von Outliern (vgl. auch Kapitel 2.7.1). Beyer, Goldstein, Ramakrishnan und Shaft [69] legen in ihrer Arbeit Grundlagen für die Interpretationsfähigkeit von spärlichen Daten in hochdimensionalen Räumen. Aggarwal und Yu beziehen sich darauf bei der Schlussfolgerung, dass diese Spärlichkeit so differenziert ausgelegt werden kann, um damit zu begründen, dass in hochdimensionalen Räumen jeder Punkt mit gleicher Güte als Outlier klassifiziert werden kann. Dies ist der Fall, weil unter der Voraussetzung, dass alle Paare von Punkten sich in etwa gleicher Distanz voneinander befinden, Cluster (vgl. [70] und [71]) nicht erfolgreich identifiziert werden können. Somit ist es folgend genauso schwer, anormale Abweichungen zu erkennen. Durch die Betrachtung des Verhaltens der Daten in Subräumen [72] sei es jedoch möglich, effektivere Algorithmen zu entwickeln, da verschiedene Lokalitäten eine Dichte in Bezug auf unterschiedliche Untermengen von Attributen aufweisen. Die gleiche Einsicht gilt für Outlier, weil für typische Anwendungen lediglich eine Untermenge der Attribute tatsächlich von abweichendem Verhalten beeinflusst wird und damit für dessen Erkennung nützlich ist.

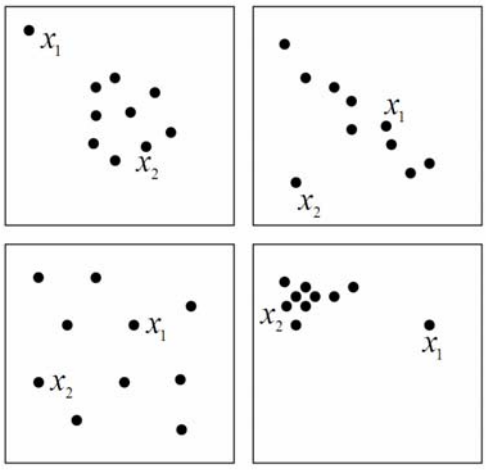


Abbildung 18 - Projektionen hochdimensionaler Datenräume im Beispiel

Abbildungsbeschreibung: Die Outliereigenschaften von Objekten in einer hochdimensionalen Menge werden ggf. nur in bestimmten Projektionen auf niedrigdimensionale Subräume sichtbar. In anderen Projektionen sticht die Outliereigenschaft dieser Objekte nicht hervor.

Abbildung 18 verdeutlicht dies an einem Beispiel. Es zeigt Ansichten von verschiedenen Projektionen eines hochdimensionalen Datenraums auf 2-dimensionale Ebenen. Für eine gewisse Zahl an solchen Ansichten aus einem Raum mit vielen Dimensionen ist es wahrscheinlich, dass diese strukturiert sind, während andere starkes Rauschen aufweisen. Im Beispiel zeigen die Punkte x_1 und x_2 anormales Verhalten in jeweils einer bzw. in zwei Ansichten der Datenmenge. In den anderen Ansichten zeigen die Punkte jedoch durchschnittliches Verhalten. Im Kontext einer Anwendung zur Erkennung von Missbrauch könnten sie mit jeweils unterschiedlichen Arten von Missbrauch korrespondieren. Über alle Dimensionen hinweg homogen betrachtet würde ihr Verhalten jedoch durchschnittlich sein. Daraus ergibt sich ein höherer Schwierigkeitsgrad für die Erkennung von Outliern beim Einbeziehen aller Dimensionen in die Entfernungsmessung zwischen Objekten, weil diese Messung durch das vermehrte Rauschen und irrelevante Dimensionen erschwert wird. Des Weiteren kann a priori die Menge der Ansichten nicht effektiv beschnitten werden, weil verschiedene Punkte in verschiedenen Ansichten verschiedene Muster anormalen Verhaltens zeigen können.

Somit unterscheidet sich das Problem der Outliererkennung nicht wesentlich von vielen anderen in der Data Mining Literatur im Hinblick auf die Tatsache, dass die Algorithmen ihre Effektivität mit steigender Zahl von Dimensionen verlieren. Bisherige Ansätze tendierten dazu, dieses spezifische Problem zu ignorieren und sich auf eine niedrige Anzahl von Dimensionen zu konzentrieren. Auch das viel versprechende Konzept von Knorr und Ng [55] zur Beschreibung der Gründe dafür, ein Objekt als entfernungsbasierten Outlier zu betrachten, nutzt eine Methode, die für eine hohe Zahl an Dimensionen ungleich komplexere Kosten verursacht.

Effektive Algorithmen für die Erkennung von Outliern in Räumen mit vielen Dimensionen sollten laut Aggarwal und Yu folgenden Anforderungen erfüllen: (1) Sie müssen in der Lage sein, dass Problem der

spärlichen Verteilung von Datenpunkten bei einer hohen Zahl an Dimensionen erfolgreich zu behandeln. (2) Sie sollten eine Interpretationsfähigkeit für den Grund des anormalen Verhaltens liefern. Im Idealfall sollte dies durch eine Wahrscheinlichkeit beschrieben werden, die Aussagen über die Signifikanz des Grundes der Verhaltensabweichung macht. (3) Vernünftige Messkriterien für die physikalische Signifikanz der Definition eines Outliers in einem *m*-dimensionalen Unterraum müssen identifiziert werden, weil z.B. ein Schwellwert für die Entfernung zwischen zwei Objekten in einem *m*-dimensionalen Raum nicht direkt vergleichbar ist mit einem ebensolchen Schwellwert in einem (*m*+1)-dimensionalen Raum. (4) Der Algorithmus soll weiterhin effizient in Bezug auf den Rechenaufwand für hochdimensionale Fälle sein und eine kombinatorische Untersuchung von Suchräumen im besten Fall ganz vermeiden; und letztendlich sollte (5) der Algorithmus die Wichtigkeit lokalen Verhaltens der Daten in Betracht ziehen, wenn Outlier identifiziert werden.

Einige dieser Anforderungen werden von den referenzierten Verfahren zur Erkennung von entfernungsbasierten und dichtebasierten Outliern sowie durch die Untersuchung von Wissen um den Charakter von Outliern (Intensional Knowledge) bereits erfolgreich erfüllt. Auch der Ansatz der Outliererkennung durch die Erkennung linearer Abweichungen von Arning, Agrawal und Raghavan [73] orientiert sich an einem Teil dieser Anforderungen. Unglücklicherweise ist keines dieser Verfahren im Hinblick auf eine hohe Zahl an Dimensionen effizient. Daher stellen Aggarwal[3] und Yu eine neue Technik vor, die Outlier anhand der Beobachtung von Dichteverteilungen in Projektionen des Datenraumes findet. Intuitiv sind somit solche Punkte Outlier, wenn sie sich in einigen Projektionen geringerer Dimensionalität in Regionen anormal geringer Dichte befinden.

Essentiell ist dabei die Idee, solche Projektionen mit anormal geringer Dichte zu untersuchen. In einem ersten Schritt werden somit die Muster identifiziert, für welche die reine Zufälligkeit die anormale geringe Dichte nicht mehr ausreichend begründen kann. Wichtig ist dies für die Betrachtung von Outliern unter dem Gesichtspunkt ihrer Abweichungswerte und nicht in Hinblick auf generelles Rauschen in der Datenmenge. Sind diese Muster identifiziert, werden Outlier als solche Datentupel definiert, die derartige Muster in sich repräsentieren. Solche niedrigdimensionalen Projektionen können im Übrigen selbst für Datenmengen untersucht werden, bei denen Attributwerte in bestimmten Bereichen fehlen oder nicht verfügbar sind. Dies ist für reale Anwendungen von Bedeutung, für die die Extraktion von Attributwerten in manchen Bereichen schwierig ist oder für die vollständige Attributbeschreibungen und Auswertungen nicht existieren.

Eine anormale niedrigdimensionale Projektion ist eine Projektion, in der die Dichte außerordentlich geringer ist, als im Durchschnitt aller Projektionen. Um solche Projektionen zu definieren, werden die Daten zuerst in diskrete Zellen eingeordnet. Jedes Attribut der Daten wird in ϕ Intervalle gleicher Tiefe unterteilt. Jedes dieser Intervalle enthält entsprechend einen Anteil $f = 1/\phi$ Tupel. Der Grund für den Einsatz von Intervallen gleicher Tiefe anstatt von Intervallen gleicher Breite resultiert in der Tatsache, dass verschiedene lokale Umgebungen der Daten auch verschiedene Dichten haben (vgl. auch Kapitel 2.5.1) und dies bei der Erkennung von Outliern eine Rolle spielen soll. Insgesamt formen diese Intervalle die lokalen Einheiten, mithilfe derer die Regionen geringerer Dimension definiert werden, welche unbegründet spärlich besetzt sind.

Sei ein *m*-dimensionaler Würfel bestimmt durch Zellenintervalle aus *m* verschiedenen Dimensionen. Wären die Attribute statistisch voneinander unabhängig, so wäre der erwartete Anteil an Tupeln in diesem Würfel gleich f^m. Selbstverständlich sind die Daten nicht unabhängig und daher wird die Verteilung der Daten im Würfel signifikant vom Durchschnittsverhalten abweichen. Im Wesentlichen sind solche Abweichungen unterhalb des Durchschnitts für die Outlier Erkennung von Nutzen.

Sei *n* die Zahl an Datenobjekten in der betrachteten Datenmenge. Sind die Daten gleichmäßig im Raum verteilt, ist die Anwesenheit oder Abwesenheit jedes Punktes in einem *m*-dimensionalen Würfel eine Bernouillische Zufallsvariable mit einer Wahrscheinlichkeit f^m. Unter dieser Annahme kann die Anzahl von Punkten in einem Würfel durch eine Normalverteilung approximiert werden, da der Zentrale Grenzwertsatz[4] gilt. Damit ist der erwartete Anteil der Punkte in einem *m*-dimensionalen Hyperwürfel gegeben durch $n \cdot f^n$ und die Standardabweichung dieses Anteils durch $\sqrt{n \cdot f^m \cdot (1 - f^n)}$.

Sei weiterhin $n(W)$ die Anzahl der Punkte in einem *m*-dimensionalen (Hyper)Würfel *W*. Der Spärlichkeitskoeffizient $S(W)$ des Würfels W ist somit definiert als:

$$S(W) = \frac{n(W) - n \cdot f^m}{\sqrt{n \cdot f^m \cdot (1 - f^m)}}.$$

[3] Anm. R. Agrawal und C. C. Aggarwal sind zwei verschiedene Autoren, vgl. Literaturverzeichnis
[4] vgl. auch http://de.wikipedia.org/wiki/Zentraler_Grenzwertsatz

Lediglich negative Spärlichkeitskoeffizienten deuten auf Würfel hin, in denen die Präsenz von Punkten signifikant unterhalb des erwarteten Durchschnitts liegt.

Unter der Voraussetzung, dass $n(W)$ normalverteilt sei, können die Tabellen [58] der Normalverteilung Aufschluss zur Quantifizierung der wahrscheinlichen Signifikanz für die Abweichung eines Punktes vom durchschnittlichen Normalverhalten geben, wenn weiterhin angenommen wird, die Daten seien gleichmäßig verteilt. Da dies jedoch in der Regel nicht der Fall ist, gibt der Spärlichkeitskoeffizient zumindest eine inutitive Idee des Signifikanzniveaus für eine gegebene Projektion.

Die Autoren dieses Ansatzes weisen auf ein grundlegendes Problem bei der Erkennung von Outliern in Projektionen hin. Dieses bezieht sich auf das Finden der am spärlichsten besetzten Würfel in einem m-dimensionalen Raum. Es existieren keine nach oben oder unten abgeschlossenen Bestandteile in der Menge der Dimensionen und der mit diesen assoziierten Intervalle, welche ungewöhnlich spärlich besetzt sind. Diese Tatsache ist nicht unerwartet, da das Suchen nach Untermengen an Dimensionen, welche spärlich besetzt sind, der sprichwörtlichen Suche nach der Nadel im Heuhaufen gleicht. Zusätzlich kann der Fall auftreten, dass gewisse Regionen in einer gewissen Menge an Dimensionen durchaus gut bevölkert sind, jedoch in der Kombination dieser Dimensionen eher spärlich besetzt sind. Dies wird am Beispiel von Menschen mit einem gewissen Alter und gewissen Krankheiten deutlich. So gibt es durchaus viele Menschen im Alter um 25 Jahre und auch viele Menschen mit Diabetis. Jedoch gibt es sehr wenig Menschen mit Diabetis im Alter um 25 Jahre. Aus der Sicht der Outlier Erkennung sind gerade solche Kombinationen sehr interessant. Allerdings stellt es sich als schwierig heraus, derartige Kombinationen mit struktierten Suchmethoden zu finden. Genau aus diesem Grund werden die besten Projektionen durch a priori unbekannte Kombinationen von Dimensionen erstellt, wobei dies nicht ausgehend von irgendeiner Unter- oder Obermengenprojektion zielgerichtet festgestellt werden kann. Eine Möglichkeit ist die Veränderung des eingesetzten Maßes um bessere Konvergenz- oder Beschneidungseigenschaften im Suchraum zu erlangen. Jedoch kann die Tatsache, dass nun das Maß aus algorithmischen Erwägungen angepasst wird, zu einer qualitativen Verschlechterung der Gesamtlösung führen. Grundsätzlich ist es nämlich nicht möglich, das Verhalten der Daten bei der Kombination von zwei Dimensionen vorauszusagen. Die qualitativ beste Option ist daher die Entwicklung von Suchmethoden, welche speziell solche verborgenen Kombinationen von Dimensionen entdecken können.

Um diese Suche im exponentiell wachsenden Raum der möglichen Dimensionskombinationen durchzuführen, haben sich die Autoren an Erfahrungen aus dem Bereich der evolutionären Suchmethoden angelehnt, um effiziente Algorithmen für die Identifizierung von Outliern zu erstellen. Vergleichbare Methoden werden erfolgreich für die Suche nach nächsten Nachbarn [69] in Räumen mit vielen Dimensionen angewandt. Zwei Algorithmen werden von den Autoren vorgestellt. Beim ersten Algorithmus handelt es sich um einen Brute-Force-Algorithmus, der alle Untermengen an Dimensionen durchsucht, um spärlich besetzte Projektionen zu finden. Die Muster dieser Projektionen werden zur Bestimmung von Outliern genutzt. Durch die aufwändige Suche im gesamten Raum an Kombinationen ist der Brute-Force-Algorithmus sehr langsam. Als zweiter Algorithmus wird ein Evolutionsverfahren vorgestellt, welches in der Lage ist, verborgene Kombinationen von Dimensionen in der Datenmenge, die spärlich besetzt ist, schneller zu finden.

Evolutionäre Algorithmen [74] sind Methoden, welche die organische Evolution [75] imitieren, um das Problem der Parameteroptimierung zu lösen. Die fundamentale Idee ist die begrenzte Verfügbarkeit von Ressourcen in der Natur, welche zu einem Wettbewerb zwischen den Spezies führt und bei diesen selbst einen Selektionsmechanismus auslöst, unter dem sich die fitteren Organismen miteinander paaren, um noch fittere Nachkommen miteinander zu zeugen. Zugleich führt ein zufälliger Mutationsmechanismus zu größerer Diversifizierung und damit zu einer Erweiterung des Verbesserungsraumes. Daran lehnen sich evolutionäre Suchtechniken eng an, denn jedes Optimierungsproblem kann als Individuum in einem evolutionären System angesehen werden. Der Grad an Fitness des „Individuums“ ist gleich dem objektiven Funktionswert der korrespondierenden Lösung und die anderen Spezies mit denen das Individuum im Wettbewerb steht, sind eine Gruppe anderer Problemlösungen. Ungleich anderen Optimierungsansätzen, wie z.B. dem Bergsteigen oder simuliertem Härten [76], betrachtet der evolutionäre Ansatz die Gesamtheit derzeit verfügbarer Lösungen anstatt sich auf eine Lösung zu konzentrieren.

Somit ergibt sich der Vorteil der evolutionären Methode im Vergleich zu anderen Methoden durch die ganzheitliche Betrachtung aller Techniken (in deren Essenz sozusagen) und durch deren kombinatorische Anwendung, wobei entsprechende Operationen die Vorgänge der Rekombination, Selektion und Mutation simulieren. Jede Problemlösung wird als ein Individuum definiert und durch eine genetische Repräsentation in Form eines Strings dargestellt. Diese Repräsentation wird durch sogenanntes Coding gewonnen und problembeschreibende Bereiche innerhalb des Strings werden als Gene und deren potentielle Werte als Allele bezeichnet. Der Grad der Fitness eines Individuums wird durch die Fitnessfunktion ermittelt. Sie ist analog zur objektiven Wertfunktion der Lösung zu sehen, je besser deren Wert, desto besser der Fitnessgrad. Mit fortschreitender Evolution werden sich die Individuen im genetischen Aufbau ähnlicher, konvergieren also zu gemeinsamen Genen (bei z.B. 95% Anteil der Population mit einem gleichen Gen bedeutet Konvergenz der

Population in Bezug auf dieses spezielle Gen [77]) und die Population insgesamt wird konvergent, wenn alle Gene Konvergenz erreicht haben. Die Anwendung evolutionärer Suchprozesse für ein vorliegendes Problem erfordert jedoch tiefes Wissen über dieses Problem, da es in der Regel schwierig ist, die Rekombinationstechniken, Selektionen und Mutationen geeignet zu kalibrieren. Die Algorithmen sind in der Literaturquelle ausführlich beschrieben und für die weitergehende Betrachtung der Materie über diese Arbeit hinaus ist es sicher interessant, die praktische Umsetzung solcher Verfahren zu untersuchen.

Ein grundsätzliches Problem bei diesem hier vorgeschlagenen Verfahren ist die Annahme der Autoren, dass die Objekte innerhalb einer Projektion als normalverteilt angenommen werden. Dies ist gerade in hochdimensionalen Räumen nicht immer der Fall und die Erfahrungen der Statistik und Wahrscheinlichkeitsrechnung legen nahe, dass gewisse Anwendungsdomainen mit einer Verteilung der Objekte in Teilprojektion entlang von durch die Anwendung vorgegebenen Hyperebenen aufwarten. Derartige Verteilungen würden bei einem so wie hier vorgeschlagenen Verfahren mglw. nicht hinreichend genug erkannt.

2.7. Räumliche Outlier Erkennung

2.7.1. Spatial Outlier

Shekhar, Lu und Zhang [28] geben bei der Betrachtung von Outliern eine Unterscheidung bzw. Klassifizierung an, welche eine Reihe von existierenden Ansätzen ordnet. Diese Ordnung ist in Abbildung 19 gezeigt. Gleichzeitig ordnen die Autoren die Definition von Nähe (Nachbarschaft) und den Vergleichsprozess zwischen den Attributen, d.h. welche Attribute zum Vergleich von normalem und anormalem Verhalten herangezogen werden. Dabei unterscheiden sie zwischen eindimensionalen und multidimensionalen Outlier Detection Ansätzen und im Bereich der multidimensionalen Ansätze zwischen der homogenen und der paritätischen Betrachtung von Attributen. Während homogene Verfahren alle Dimensionen zur Definition von Nähe von Objekten zueinander in Betracht ziehen und auch alle diese Attribute in den Verhaltensvergleich einbeziehen, ordnen paritätische Verfahren die Objekte in einen festen Raum ein, der von echten räumlichen Attributen im Sinne eines realen Raumes gekennzeichnet ist. Der Verhaltensvergleich findet sodann mit Hilfe genau der Attribute statt, welche nicht-räumlich sind, d.h. welche die Position eines Objektes in einem „realen“ Raum nicht bestimmen.

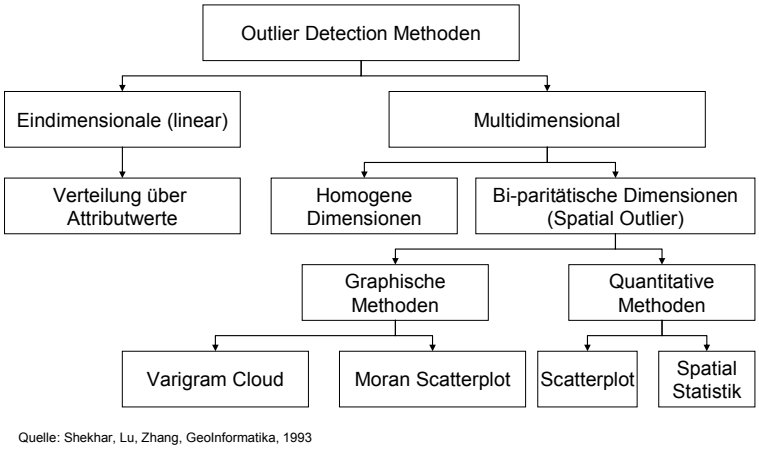


Abbildung 19 - Klassifizierung von Outlier Ansätzen nach Shekhar

Abbildungsbeschreibung: Im Rahmen der Ansätze für räumliche Outlier werden die unterschiedlichen Verfahren von den Autoren [28] klassifiziert. Dabei wird unterschieden, ob der Attributraum homogen über alle Attribute hinweg betrachtet wird, oder ob der Attributraum geteilt wird.

Daher prägt Shekhar et al den Begriff der räumlichen Outlier (Spatial Outlier – *SPO*). Anwendungsbereiche für diese Spezialform von Outliern finden sich z.B. im Transportwesen, in der Klimaforschung, oder bei Location Based Services. So werden Verkehrsknotenpunkte als Objekte im 2- oder 3-dimensionalen Raum angeordnet und Nachbarschaft von Objekten anhand dieser Ordnung bestimmt. Der Vergleich von normalem und nicht-normalem Verhalten findet danach jedoch anhand von nicht-räumlichen Attributen statt, wie z.B. unter Bezug auf die Werte von Verkehrsdichtesensoren bzgl. der Verkehrsknoten. Es wird eine Reihe von spezifischen Verfahren angegeben, welche sich gerade für diese Situationen eignen. Abbildung 20 zeigt ein vereinfachtes

grafisches Beispiel, welches Shekhar, Lu und Zhang in ihrer Arbeit angeben. Outlier, welche mit *x* gekennzeichnet sind, sind solche Objekte, welche in einer generellen Nachbarschaft bzgl. ihrer eindimensionalen räumlichen Position (*Location*-Achse) einen stark abweichenden nichträumlichen Attributwert (*Value*-Achse) aufweisen. Dabei sind die Nachbarschaften, gegenüber denen diese Abweichung im Attributwert stattfindet, als umrandete Boxen gekennzeichnet[5]. Bei einer homogenen Betrachtung, also unter Einbeziehung der *Location*-Achse und *Value*-Achse würden andere in dieser Arbeit beschriebene Verfahren mglw. auch andere Outlier identifizieren, hier z.B. nur x_1 und x_6, oder beispielsweise Objekte, welche an den Rändern von (*Location*, *Value*)-Nachbarschaften liegen.

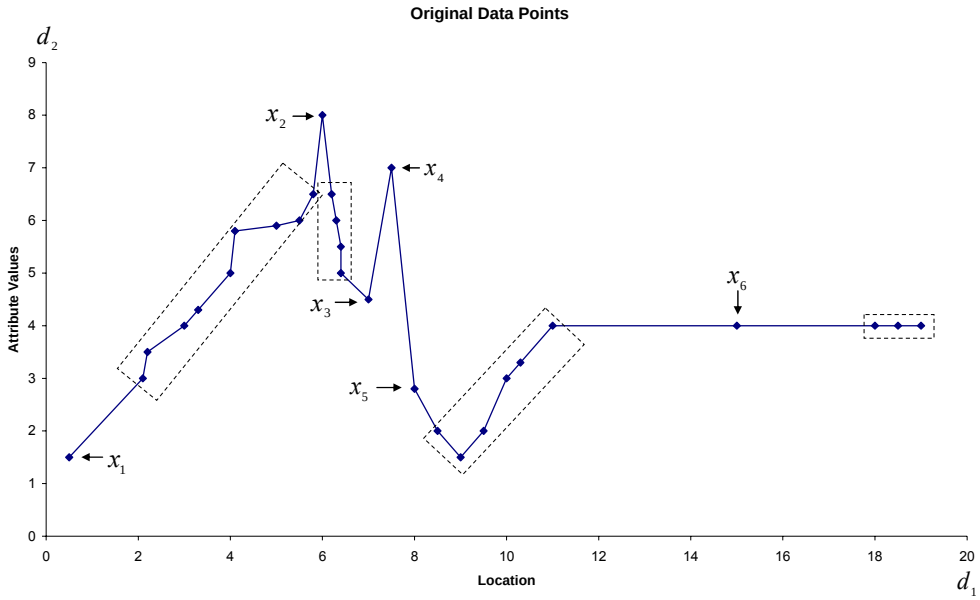


Abbildung 20 - Beispiel für Spatial Outlier

Abbildungsbeschreibung: Als Spatial Outlier werden solche Objekte *x* gekennzeichnet, welche gegenüber ihrer direkten räumlichen Nachbarschaft (d_1-Achse) starke Abweichungen in den nichträumlichen Attributwerten (d_2-Achse) aufweisen.

Räumliche Attribute werden also zur Charakterisierung von Position, Nachbarschaft und Entfernung benutzt. Nicht-räumliche Attribute werden verwendet, um die räumlich referenzierten Objekte mit ihren Nachbarn zu vergleichen. In ihren Arbeiten verweisen die Autoren Shekhar, Zhang, Huang und Vatsavai [27] auf zwei Arten von bi-paritätischen Verfahren für multidimensionale *SPO*-Tests. Zum einen sind dies grafische und zum anderen quantitative Tests. Grafische Tests, welche auf Visualisierung basieren, stellen Outlier optisch heraus. Beispielmethoden umfassen unter anderem Variogram Wolken und sogenannte Scatterplots und Moran Scatterplots. Quantitative Methoden stellen einen präzisen Test zur Verfügung, um Spatial Outlier vom Rest der Datenmenge abzusondern. Dabei sind Scatterplots selbst eine Repräsentationstechnik aus der Familie der quantitativen Methoden.

Eine Variogram Wolke (Cloud), welche ausführlich von Cressie [64] vorgestellt wird, stellt Datenpunkte in Bezug auf deren Nachbarschaftsbeziehungen dar. Für jedes Paar an Positionen wird die Quadratwurzel des absoluten Unterschieds der Attributwerte an diesen Positionen gegenüber der euklidischen Distanz zwischen den Positionen dargestellt. In Datenmengen mit starken räumlichen Abhängigkeiten wird die Abweichung der Attribute gleichsam mit steigender Distanz zwischen Positionen wachsen. Positionen, welche sich nah aneinander befinden, aber in ihren Attributwerten stark abweichen, könnten den Verdacht auf räumliche Outlier nahe legen, auch wenn die Werte an beiden Orten im normalen Rahmen liegen, würde eine nicht-räumliche Auswertung vorgenommen (*Anmerkung:* d.h. unter Einbeziehung aller Dimensionen eine Auswertung, welche Attribute nicht paritätisch trennt).

Abbildung 21 zeigt eine Variogram Wolke für die Datenmenge, welche in Abbildung 20 vorgestellt wurde. Diese Zeichnung zeigt zwei Paare ((x_5, x_4) und (x_3, x_4)), welche über der allgemeinen Gruppe von Paaren liegen und damit möglicherweise in Beziehung zu Spatial Outliern stehen. Der Punkt x_4 könnte als Spatial

[5] Diese Boxen wurden der originalen Quellengrafik zum besseren Verständnis hier hinzugefügt.

Outlier identifiziert werden, weil er in beiden Paaren vorkommt. Jedoch sind grafische Verfahren in ihrer Effektivität meistens durch das Fehlen präziser Kriterien zur Kennzeichnung von räumlichen Outliern begrenzt. Zusätzlich verlangt das Variogram Cloud Verfahren ein aufwändiges Post-Processing um Spatial Outlier von ihren Nachbarn zu separieren, vor allem wenn mehrere Outlier vorkommen oder wenn die Dichten stark variieren.

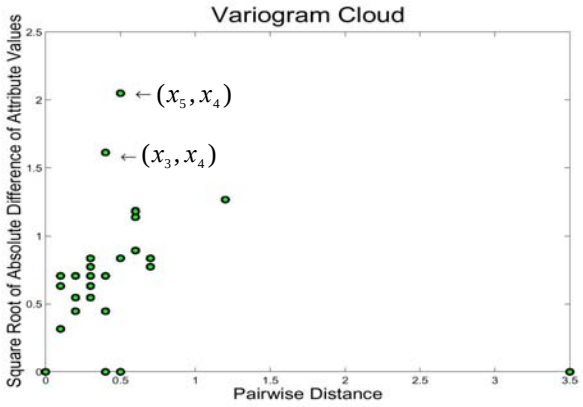


Abbildung 21 - Variogram Cloud für Spatial Outlier

Abbildungsbeschreibung: Der Punkt x_4 könnte als Spatial Outluer identifiziert werden, weil er in beiden gekennzeichneten Paaren vorkommt und hohe nichträumliche Attributabweichungen im Vergleich zu den anderen Wertepaaren zeigt.

Ein Moran Scatterplott, wie von Anselin [66] beschrieben, ist eine Aufzeichnung der normierten Attributwerte gegenüber dem Durchschnitt der normierten Attributwerte der Nachbarschaft. Ausführlich werden diese Formeln in der Literatur [27] dargestellt. Abbildung 22 zeigt beim Moran Scatterplot für die Datenmenge aus Abbildung 20 den oberen linken und unteren rechten Quadranten als Indikation einer räumlichen Beziehung abweichender Werte, z.B. niedrigwertige Punkte, welche von hochwertigen Nachbarn umgeben sind (z.B. die Punkte x_5 und x_3), sowie hochwertige Punkte mit niedrigwertigen Nachbarn (z.B. x_4). Somit können Punkte identifiziert werden, die von einer unüblich hohen Zahl hochwertiger bzw. niedrigwertiger Nachbarn umgeben sind. Diese Punkte können dann wie Outlier behandelt werden.

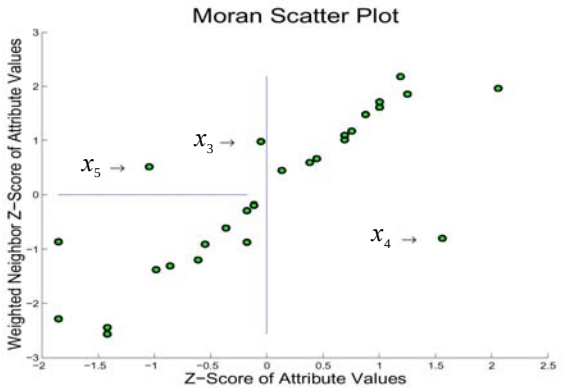


Abbildung 22 - Moran Scatter Plot für Spatial Outlier

Abbildungsbeschreibung: Objekte, welche sich im oberen linken bzw. im unteren rechten Quadranten dieses Plots befinden, stechen als Outlierkandidaten heraus, weil sie im Vergleich zu den Nachbarn im Attributwert stark nach unten bzw. nach oben abweichen.

Ein (normaler) Scatterplot nach Anselin [65] zeichnet die Attributwerte auf der d_1-Achse auf und die durchschnittlichen Attributwerte der Nachbarschaft auf der d_2-Achse. Eine Linie der minimalen quadratischen Regression wird verwendet, um die Outlier zu identifizieren. Eine aufgeworfene Gruppierung (Scatter), welche

sich nach oben und rechts zieht (Slope), weist auf eine positive räumliche Autokorrelation hin, d.h. aneinander angrenzende Werte sind im Trend gleichartig. Ein Scatter mit einem Slope nach oben und links weist auf eine negative räumliche Autokorrelation hin. Das Residuum ist als die vertikale Distanz zwischen einem Punkt und der Regressionslinie $d_2 = md_1 + b$, also als $\varepsilon = d_{2p} - (md_{1p} + b)$ definiert. Fälle mit standardisierten Residuen

$$-3{,}0 < \varepsilon_s = \frac{\varepsilon - \mu_\varepsilon}{\sigma_\varepsilon} < 3{,}0$$

werden als mögliche Spatial Outlier vermerkt. Dabei stellen μ_ε den mittleren Erwartungswert und σ_ε die Standardabweichung der Verteilung des Fehlerterms ε dar.

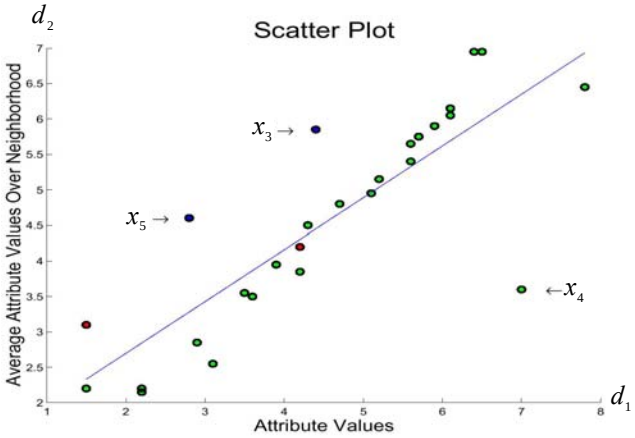


Abbildung 23 - Scatter Plot für Spatial Outlier

Abbildungsbeschreibung: Objekte, welche von der durch Autokorrelation gekennzeichnete Slope abweichen, werden als Outlierkandidaten markiert.

Abbildung 23 stellt wiederum für die schon mehrfach referenzierte Datenmenge einen Scatterplot vor. Der Punkt x_4 erscheint hier am weitesten von der Regressionslinie entfernt zu liegen und kann als potentieller Spatial Outlier identifiziert werden.

Im Rahmen von quantitativen Tests wird nun ein weiteres Beispiel eingeführt, welches ausführlich in der Literatur [28] dargestellt ist. Hierbei wird eine Position (als Repräsentation eines lokalen Sensors) mit der Nachbarschaft unter Verwendung einer Sensorfunktion verglichen. Die Statistikfunktion *S*(*x*) bezeichnet den Unterschied des Attributwertes eines Sensors für einen Datenpunkt *x* mit dem durchschnittlichen Attributwert von den Nachbarn von *x*. Die räumliche Statistik *S*(*x*) ist normalverteilt, wenn der Attributwert *f*(*x*) normalverteilt ist.

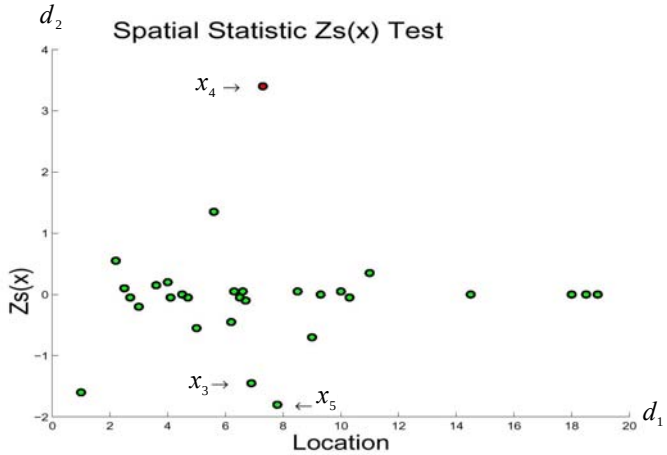


Abbildung 24 - Statistischer *Zs(x)* Test für Spatial Outlier

Abbildungsbeschreibung: Die räumliche Statistikfunktion weist für x_4 mit hoher Sicherheit eine Outliereigenschaft nach.

Ein populärer Test für die Erkennung von Spatial Outliern für normalverteilte *f*(*x*) kann wie folgt beschrieben werden:

$$Z_{s(x)} = \left| \frac{S(x) - \mu_s}{\sigma_s} \right| > \theta .$$

Für jeden Datenpunkt *x* mit einem Attributwert $f(x)$ ist für $S(x)$ jeweils μ_s der mittlere Erwartungswert von $S(x)$ und σ_s die Standardabweichung von $S(x)$ über alle Datenpunkte. Die Wahl für θ hängt von einem zu spezifizierenden Konfidenzniveau ab, wobei z.B. ein Konfidenzniveau von 95% zu einem Wert $\theta \approx 2$ führt.

Abbildung 24 zeigt die beschriebene statistische Methode in einer Visualisierung. Die d_1-Achse zeigt die Position eines Punktes im eindimensionalen Raum und die d_2-Achse den Wert der räumlichen Statistikfunktion $Z_{s(x)}$ für jeden dieser Punkte. x_4 hat z.B. einen Wert $Z_{s(x)} > 3{,}0$ und wird damit leicht als räumlicher Outlier identifiziert. Bemerkenswert ist, dass gerade die anderen Outlierkandidaten der vorher beschriebenen Verfahren x_3 und x_5 Werte von $Z_{s(x)}$ um -2 haben, weil in ihrer Nachbarschaft Spatial Outlier vorkommen (z.B. x_4).

Von den Autoren wird für diesen Ansatz Forschungsbedarf in verschiedenen Feldern angeführt. So müssen zum Beispiel klassische Data Mining Techniken mit den Techniken des Spatial Data Mining intensiv verglichen werden. Da die im Spatial Data Mining vorkommenden impliziten Abhängigkeiten zwar durch diverse Techniken in traditionelle Eingabedaten(-spalten) für klassische Data Mining Verfahren eingebracht werden können und somit im Prinzip auch mit allen klassisch vorgestellten Outlier Detection Verfahren die Spatial Outlier erkennbar würden, wurden eben hier spezielle Techniken eingeführt, um diese impliziten Abhängigkeiten alternativ zu handhaben. Es gibt in der Literatur jedoch keine Anleitung zur Wahl zwischen diesen Alternativen. Dafür müssten die zwei Ansatzwege in Bezug auf Effektivität und Effizienz miteinander innerhalb der Forschung verglichen werden.

Auch wird darauf hingewiesen, dass die räumlichen Beziehungen zwischen Lokationen oft viel komplexer sind, als dies in den derzeit verwendeten Modellen berücksichtigt werden kann. So ist es zum Beispiel notwendig, für die Modellierung von Topologie die Datenmodelle stark anzureichern.[6]

Räumliche Muster (Pattern), wie z.B. Spatial Outlier oder Spatial Co-location Rules, werden im Data Mining Prozess mit unüberwachtem Lernen identifiziert. Es ergibt sich dadurch die Anforderung, die statistische Signifikanz solchermaßen erkannter räumlicher Muster zu beurteilen.

Neben einigen weiteren angeführten und sehr interessanten Forschungsrichtungen, für die auf die Literaturquellen [28] verwiesen sei, spielt die Effektivität der Visualisierung eine entscheidende Rolle. Es gibt in den vorgestellten Visualisierungen bisher keine Möglichkeit, Spatial Outlier optisch herauszustellen. Meist ist diese räumliche Beziehung nicht offensichtlich und es ist notwendig, die Informationen zurück in den ursprünglichen Raum zu transferieren, um die Nachbarschaftsbeziehungen zu testen. Da ein einzelner Spatial Outlier nicht nur eine lokale Instabilität erkennen lässt, sondern meist auch Instabilität in seiner direkten Nachbarschaft, ist es wichtig, erkannte Positionen zu gruppieren und echte Spatial Outlier aus diesen Gruppen im Post-Processing zu separieren.

Gleichzeitig wird auf den oft hohen Aufwand zur Erkennung räumlicher Outlier verwiesen. Hier wird vor allem die Nutzung klassischer Outlier Detection Methoden (als potentielle Filter oder zusätzliche Komponenten) vorgeschlagen.

[6] *Anmerkung:* Bei Verkehrsflussmodellen spielt z.B. auch die Steigung auf Distanzen, der Straßenbelag, die Umweltbeschaffenheit u.v.m. potentiell eine große Rolle.

2.7.2. Spatial Temporal Outlier

Von Cheng und Li [46] werden so genannte Spatial Temporal Outlier eingeführt, welche die Idee der Spatial Outlier weiterentwickeln und sich vor allem auf die Notwendigkeit beziehen, im geologischen Bereich solche Orte als lokale temporale Anomalien zu identifizieren, welche sehr verschieden von ihrer Nachbarschaft sind, auch wenn sie nicht sehr verschieden von der betrachteten Gesamtmenge bzw. Datenpopulation zu sein scheinen.

Der Definition nach ist ein temporärer räumlicher Outlier (*STO*) ein räumlich und zeitlich referenziertes Objekt, dessen thematische Attributwerte von den räumlich und zeitlich referenzierten Objekte in seiner jeweils entweder räumlichen und/oder zeitlichen Nachbarschaft signifikant abweichen. Zweck ist die Entdeckung von implizitem Wissen, insbesondere hinsichtlich lokaler Instabilitäten.

Spatial Temporal Outlier werden durch einen multidimensionalen Ansatz gefunden, bei dem räumliche und zeitliche Achsen hinsichtlich der aufeinanderfolgenden Änderungen von Objektattributen geprüft werden. Die Autoren weisen darauf hin, dass es räumliche und zeitliche Abhängigkeiten zwischen räumlichen Objekten auf sehr unterschiedlichen Ebenen gibt (vgl. Yao [67]). Solche Beziehungen sollten bei der Erkennung von *STO*-Outliern einbezogen werden.

Um räumliche temporäre Outlier zu entdecken, können existierende Methoden zur Erkennung von Spatial Outliern so modifiziert werden, dass die semantischen und dynamischen Aspekte entsprechend in einem multidimensionalen Raum berücksichtigt werden können. Dabei wird mit einem Mehrschritt-Verfahren, welches in Abbildung 25 gezeigt wird, operiert. Dieses Verfahren basiert auf 4 Schritten und ist ein Multi-Achsen Ansatz, da die Aggregation und die Verifikation als Schritte jeweils die Änderungen zwischen zwei aufeinander folgenden Achsen in Raum und Zeit vergleichen.

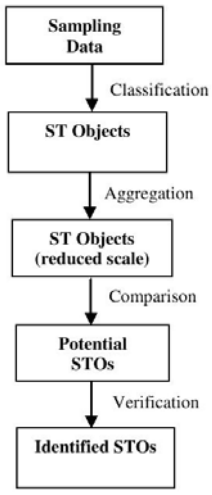


Abbildung 25 - Mehrschrittverfahren für Spatial Temporal Outlier

Schritt 1 (**Klassifikation**) umfasst die Klassifikation bzw. das Clustering der Eingangsdaten basierend auf dem Hintergrundwissen zu diesen Daten. Die Clustering Methode wird auf Grundlage des Ausgangswissens und der Charakteristika der Daten gewählt. Wenn die Daten rasterbasierte Bilder sind, ist z.B. überwachte Klassifikation anzuwenden. Wenn kein Wissen über die Daten a priori vorhanden ist, kann zum Beispiel mit Hilfe von neuronalen Netzen geclustert werden. Der Zweck dieses Schrittes ist die Formung von Regionen, welche eine signifikante semantische Bedeutung haben.

Schritt 2 (**Aggregation**) aggregiert die geclusterten Ergebnisse des ersten Schrittes um die Stabilität der Daten zu prüfen. In diesem Schritt werden Outlier ggf. ausgeschnitten, daher ist dieser Schritt auch ein Rauschfilter.

Schritt 3 (**Vergleich**) vergleicht nun die Ergebnisse von Schritt 1 mit denen von Schritt 2 und identifiziert die Objekte, welche in Schritt 2 ausgefiltert wurden. Diese sind potentielle *STO*-Outlier. Der Vergleich erfolgt auf jeweils zwei zeitlich aufeinander folgenden räumlichen Achsen.

Schritt 4 (**Verifizierung**) überprüft die temporären Nachbarn der potentiellen in Schritt 3 identifizierten *STOs*. Wenn der semantische Wert eines solchen *STO* keine signifikanten Unterschiede zu den Werten seiner temporären Nachbarn aufweist, ist er kein Spatial Temporal Outlier. Im anderen Fall wird er als *STO* bestätigt.

Das vorgestellte Verfahren wurde von den Autoren an experimentellen Daten am Beispiel der Küstenentwicklung von Ameland in den Niederlanden untersucht. Dabei wurde die Entwicklung über einen Zeitraum von 6 Jahren hinweg (1989 – 1995) überprüft. Im Ergebnis interessant ist die Entdeckung von Entwicklungen in den Küstenlinien (in verschiedenen Wasserhöhenlinien), welche temporär, also nicht anhaltend sind. Eben diese waren von anhaltenden Entwicklungen zu unterscheiden.

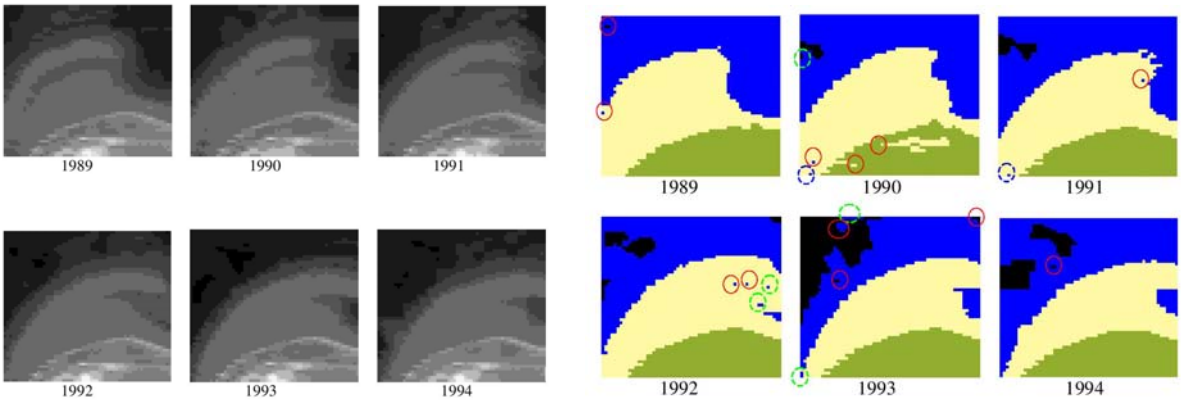


Abbildung 26 - Spatial Temporal Outlier am Beispiel von Wasserstandsdaten

Abbildungsbeschreibung: Hier sind die Eingangsdaten (Wasserhöhen) und dem gegenübergestellt das Ergebnis der Outlier Detection *für* Spatial Temporal Outlier visualisiert, wobei alle eingekreisten Objekte entsprechende *STO* Kandidaten sind, jedoch nur die komplett markierten Objekte verifizierte *STO* sind, wohingegen die gestrichelt eingekreisten Objekte keine räumlich-temporären Outlier sind.

2.8. Clustering und Outlier Detection

2.8.1. Clustering Verfahren im Einsatz zur Outliererkennung

Beim Einsatz von Clustering-Verfahren ist bereits bei der Beschreibung verschiedener anderer Outlier Ansätze erläutert worden, dass Clustering als Ansatz prinzipiell in der Lage ist, Outlier zu behandeln. Outlier stellen in diesem Sinne genau solche Objekte dar, welche nach dem Clustering keinem Cluster selbst geeignet zugeordnet werden konnten. Dabei ist der Einsatz auf solche Verfahren beschränkt, die nicht zwingend alle Objekte einer definierten Zahl von Clustern zuweisen. Verschiedene Verfahren weisen solche Objekte als Rauschen (Noise) oder direkt als Kandidaten für Outlier aus. Die indirekte Erkennung belegt für die entsprechenden Outlier dann eine Grundlage (z.B. in Form eines statistischen Maßes), wie sie auch dem Clustering-Verfahren zugrunde liegt.

Für den Einsatz von Clustering zur Outliererkennung spricht die Tatsache, dass Clustering und Outlier-Erkennung prinzipiell zwei Sichtweisen auf einen vergleichbaren Prozess sind. Während beim Clustering ähnliche Objekte von verschiedenartigen aus der Gesamtmasse getrennt werden und so lange zu Gruppen zusammengefasst werden, bis lediglich ein Rest von Objekten verbleibt, bei dem dies nicht mehr möglich oder sinnvoll erscheint, wird bei der Outliererkennung sofort nach den Objekten gesucht, welche sich durch eine Differenzierung von der Gesamtmenge unterscheiden. Oft ist ein Clustering auch eine fördernde oder sogar notwendige Voraussetzung für die erfolgreiche Anwendung eines Outlier-Identifizierungsansatzes.

Gegen den Einsatz von Clustering zur Outlier-Detection selbst spricht vor allem die fehlende Spezialisierung auf die Erkennung von Outliern, welche oft nur ein (lästiges) Beiprodukt sind. Somit fehlt den meisten Clustering-Verfahren auch die Möglichkeit, die Outlier-Eigenschaft nachvollziehbar zu qualifizieren[7] (Maß, welches den Status als Outlier begründet) oder zu quantifizieren (Grad einer Outlier-Eigenschaft). Zudem ist die Outliererkennung auch in solchen Datenmengen sinnvoll, in denen effektiv keine Cluster erkannt werden können, wobei es Anwendungsdomainen gibt, die hohe Hürden für ein erfolgreiches Clustering setzen.

Eines der einfachsten Verfahren zur Erkennung von Outliern ist die Nutzung von Clustereliminierung. Hierfür kann im Prinzip jedes Clusteringverfahren eingesetzt werden, dass Elemente, welche sich nicht für die Zuordnung zu einem Cluster eignen, separat ausweist, indem es diese z.B. als Rauschelemente oder direkt als Outlierkandidaten kennzeichnet. Dabei ist zu berücksichtigen, dass jedoch lediglich die fehlende Zuordnungsmöglichkeit ausschlaggebend für die Einordnung als Outlier ist, aber keine gezielte Suche nach

[7] *Anmerkung:* Hiermit ist nicht die eigentliche Qualität des Outliers gemeint, wie z.B. nach [55].

Outliern selbst (unter Anwendung eines gerichteten Ansatzes). Hier ist der Outlier also ein echtes Abfallprodukt eines indirekten Ansatzes, welcher direkt nicht nach Outliern sucht. Dies muss der Tatsache, dass Outlier gut gefunden werden, zwar nicht entgegenstehen, aber Knorr und Ng ([3], [6] und [8]) weisen darauf hin, dass z.B. das Clusteringverfahren DBSCAN [100] bzgl. der Markierung von Outliern sehr zurückhaltend ist, da das Verfahren auf die Erkennung möglichst großer und umfassender Cluster ausgerichtet ist.

Bzgl. der Komplexität der Verfahren ist anzumerken, dass der Rechenaufwand für das Clustering dem Aufwand für die Erkennung des „Beiproduktes“ der Outlier entspricht. Abhängig vom eingesetzten Clustering kann dieses Verfahren sehr effizient sein, oder auch nicht. Eine Reihe von Clusteringverfahren erfordern Angaben, welche mit der Outlierfindung nur wenig zu tun haben (z.B. die Eingabe, welche Zahl an Clustern gesucht bzw. erwartet wird). Oft kann eine den anderen Outlierverfahren typische Angabe zur Optimierung der Erkennung von Outliern nicht vorgenommen werden. Es muss also bei der Verfahrensoptimierung auf Eigenschaften abgestellt werden, die nicht direkt mit der Outliersuche verbunden sind.

In der Literatur [19] werden ausführlichere Angaben zum Vergleich von Clustering und Outlier Detection gemacht, sodass für weitergehende Betrachtungen auf diese Quellen verwiesen sei.

2.8.2. Cluster Based Local Outlier – *CBLOF*

Die Autoren He, Deng und Xu [10] stellen mit Cluster-basierten lokalen Outliern einen weiteren Ansatz vor. Dieser orientiert sich vor allem daran, dass die große Zahl der existierenden Algorithmen hohe Rechenkosten haben. Dies ist im Zusammenhang mit großen, oftmals in verteilten Festplattensystemen oder verteilten Speichernetzen (Anm.: NAP – Network applied Storage oder SAN – Storage Area Network) abgelegten Datenbanken, in deren Datenmengen gesucht wird, nicht sehr sinnvoll. Weiterhin nutzen eine Menge der existierenden Algorithmen den vollen, aus der Betrachtung aller Dimensionen resultierenden Abstand zwischen zwei Punkten mit dem Ergebnis von unerwarteten Performance- und Qualitätseinbußen. Dies bezieht sich sowohl auf ein den Abstand definierendes statistisches Maß, als auch auf die homogene Betrachtung der Dimensionen in verschiedenen vorgestellten Ansätzen. Da sich der vorgestellte Ansatz stark an Clustering als eine der möglichen Methoden anlehnt, Outlier zu identifizieren, weisen die Autoren auf die Einschränkungen der Outlier Detection durch Clustering-Verfahren hin. Zwar sind diese im Prinzip in der Lage, Outlier auch zu behandeln, allerdings meist in Form von Rauschen. Die initialen Arbeiten der Cluster based Outlier Detection, z.B. durch Su, Jiang und Tseng [56], weisen einige Probleme auf. Su et al betrachtet nur kleine Cluster als mögliche Outlier, liefert aber kein Maß zur Identifizierung des Outlier-Grades. Außerdem ist es auch bei clusterbasierten Outliern sinnvoll, nur eine gewisse Menge an Outlier-Graden für die potentiellen top-*n* Outlier zu berechnen, da in der Regel davon ausgegangen werden kann, dass die meisten Objekte keine Outlier sind. Die von Su vorgestellte Methode konnte dies ebenfalls nicht realisieren.

> *Anmerkung:* Im Prinzip wird auch hier die Nutzung derselben Methode und Funktionalität zum Lösen sowohl des Clustering-, als auch des Outlier-Problems als wünschenswert angesehen. Ein Vorteil ist die Tatsache, dass sich der Anwender durch die Integration nicht um die Auswahl separater, geeigneter Algorithmen kümmern müsste.

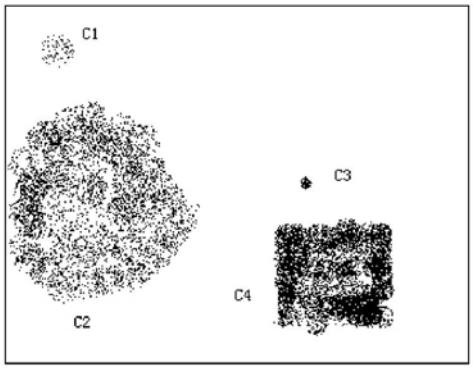


Abbildung 27 - Cluster als Outlier am Beispiel einer Datenmenge

> *Abbildungsbeschreibung:* Kleine Cluster und alle in diesen vorhandene Objekte können gegenüber großen Clustern durchaus eine Outliereigenschaft aufweisen, wie an diesem Beispiel intuitiv deutlich wird, sofern der Unterschied zwischen den Clustern signifikant genug ist.

Die Idee von clusterbasierten lokalen Outliern fußt auf der Überlegung, dass alle Datenpunkte, welche nicht in einem großen Cluster liegen, als Outlier betrachtet werden können. Die grundlegende Frage ist also die der Dominanz von Clustern in einer zu untersuchenden Datenmenge. Um den Gedanken der Lokalität, wie er z.B. von Breuning et al eingeführt wird, aufrecht zu erhalten, sollten clusterbasierte lokale Outlier (*CBLO*) nachweisen, dass sie gegenüber spezifizierten Clustern lokal sind. Dazu wird jedem Objekt ein verfahrensspezifischer Outlier Faktor (*CBLOF*) zugewiesen. Dieser wird im Prinzip zum einen an der Größe des Clusters gemessen, zudem das Objekt zugehörig ist. Zum anderen fließt die Entfernung zwischen dem Objekt und seinem nächsten Cluster ein, sofern das Objekt selbst in einem kleinen Cluster liegt, wobei Einzelobjekte ggf. sehr kleinen Clustern gleichzusetzen sind. Abbildung 27 zeigt dies am Beispiel. *C1* und *C4* bezeichnen große Cluster, *C1* und *C3* sind Outlier-Kandidaten und die Punkte in *C1* sind z.B. lokal zu *C2*.

Formal sind clusterbasiere lokale Outlier wie folgt definiert:

Definition der Partitionierung einer Datenmenge in Cluster: Sei $a_1,...a_m$ eine Menge von Attributen für ein Objekt *x* aus einer *m*-dimensionalen Datenmenge *X*, also $x \in X$, so wird das Ergebnis eines Clustering Verfahrens auf *X* mit *k* Clustern eingeführt als:

$$C = \{C_1,...C_k \mid C_i \cap C_j = \emptyset_{i,j=1,...k}; C_1 \cup ... \cup C_k = X\}.$$

Der Clustering Algorithmus, welcher die Menge X in disjunkte Teilmengen von Objekten partitioniert, ist frei wählbar, sollte jedoch nachvollziehbarer Weise gute Clusteringergebnisse liefern. Ein zu lösendes Problem bei der Bewertung des Outlier Status ist die Beantwortung der Frage, ob ein Cluster groß oder klein ist.

Definition der quantitativen Größe von Clustern: Angenommen, $C = \{C_1,...C_k\}$ sei eine Menge Cluster in der Reihenfolge $|C_1| \geq \cdots \geq |C_k|$. Gegeben sind zwei numerische Parameter α und β. Der Parameter *b* sei als Grenze zwischen großen und kleinen Clustern definiert, wenn eine der zwei folgenden Formeln gilt: (i) $(|C_1| + \cdots + |C_b|) \geq |X| \cdot \alpha$ oder (ii) $|C_b|/|C_{b+1}| \geq \beta$. Dann sei die Teilmenge großer Cluster definiert als $LC = \{C_i \mid i \leq b\}$ und die Menge kleiner Cluster definiert als $SC = \{C_j \mid j > b\}$. Diese Definition stellt eine quantitative Größe zur Unterscheidung von großen und kleinen Clustern bereit. Sie stellt zum einen fest, dass Cluster mit einer großen Zahl an Datenpunkten als große Cluster gelten sollen, weil die meisten Datenpunkte in einer Menge i.d.R. keine Outlier sind. Dies wird durch den Parameter α beschrieben. Ist er z.B. $\alpha = 0{,}9$, so wird beschrieben, dass die Cluster, welche in Summe 90% der Datenpunkte enthalten, große Cluster sind. Zum anderen betrachtet die Definition die Tatsache, dass große und kleine Cluster sich in ihrer Größe signifikant unterscheiden sollten. So ist z.B. für $\beta = 5$ ein großer Cluster mindestens fünfmal so groß, wie ein kleiner Cluster.

Definition des clusterbasierten lokalen Outlierfaktors *CBLOF*: Angenommen, $C = \{C_1,...C_k\}$ sei eine Menge Cluster in der Reihenfolge $|C_1| \geq \cdots \geq |C_k|$ und α, β, b, LC, SC seien wie vorher definiert. Für jedes Objekt *x*, sei der clusterbasierte lokale Outlierfaktor *CBLOF* definiert als:

(i) $CBLOF(x) = |C_i| * \min(d(x, C_j)), x \in C_i, C_i \in SC, C_j \in LC, 1 \leq j \leq b$ bzw.

(ii) $CBLOF(x) = |C_i| * d(x, C_i), x \in C_i, C_i \in LC$

Der Faktor wird aus der Größe eines Clusters und der Entfernung zwischen dem Objekt und seinem nächsten Cluster errechnet, wenn *x* in einem kleinen Cluster liegt. Ansonsten wird er aus der Entfernung zwischen dem Objekt und dem Cluster, zu dem es gehört, errechnet. Dies ist der Fall wenn das Objekt zu einem großen Cluster gehört und unterstreicht das lokale Verhalten der Daten. Für die Berechnung der Entfernung ($d(x, C)$) zwischen dem Objekt und dem Cluster, zu dem es gehört, kann z.B. die Ähnlichkeitsfunktion des Clusteringverfahrens verwendet werden, aber auch eine anderes, frei wählbares und adäquates Entfernungsmaß.

Mit dem beschriebenen Faktor *CBLOF* kann der Grad der Abweichung eines Datenelements festgestellt werden. Die Berechnung des Grades erfordert die vorherige Anwendung eines Clustering Algorithmus. Im in der Literaturquelle gegebenen Beispiel setzen die Autoren den selbst entwickelten Squeezer Algorithmus [58] ein. Im Prinzip kann aber auch jeder andere Clustering-Algorithmus, wie z.B. BIRCH [60] oder DBSCAN [100], eingesetzt werden.

Der Algorithmus [10] zur Berechnung der Outlier, *FindCBLOF*, setzt auf den Ergebnissen des Clusterings auf und berechnet für die Menge der identifizierten Cluster $C = \{C_1,...C_k\}$ mit $|C_1| \geq \cdots \geq |C_k|$ zuerst anhand der Parameter α und β die Mengen *LC* und *SC*. Im nächsten Schritt werden sodann die *CBLOF* Werte für jedes Objekt der Datenmenge berechnet, wobei in Abhängigkeit von der Zugehörigkeit der Objekte zu einem Cluster

in *LC* oder *SC* die entsprechenden Formeln angewendet werden. Diese Outlier Werte werden im Ergebnis ausgegeben. Der Aufwand des Algorithmus ist mit $O(n)$ für eine Datenmenge von *n* Elementen angegebenen. Rechnet man den Aufwand des Clusterings mit $O(H)$ ein, so ist der Gesamtaufwand $O(H+n)$. Da Squeezer [58] auf kategorisierten Daten arbeitet, muss eine Datenmenge gegebenenfalls in eine Menge an diskreten Daten umgewandelt werden, wozu es eine Reihe von Verfahren gibt, welche in der Literatur beschrieben sind, vgl. auch Liu, Hsu und Ma [59].

Als Felder für zukünftige Forschungen haben die Autoren des Verfahrens, He, Deng und Xu, die Suche nach top-*k CBLOF*-Outliern angegeben [10].

2.9. Outlier unter Einbeziehung semantischen Wissens

Von He, Deng und Xu [11] werden Outlier in einem weiteren Ansatz betrachtet, welcher sich nicht nur auf die Daten der gegebenen Menge selbst bezieht, sondern Attribute, welche Klasseninformationen zu den in der Datenmenge enthaltenen Objekten enthalten, für die Identifizierung nutzt. Die Idee dahinter ist in der Annahme begründet, dass Objekte gleicher Klasse ein in einem gewissen Rahmen gleichartiges Verhalten zeigen und dass solche Objekte innerhalb derselben Klasse, welche von diesem Standardverhalten abweichen, als Outlier stark in Frage kommen.

Informell werden semantische Outlier daher als solche Objekte definiert, welche ein anderes Verhalten zeigen, als dies in der Klasse dieses Objektes üblich ist. Formal werden semantische Outlier wie folgt definiert:

Definition der Partitionierung einer Datenmenge in Cluster: Sei $a_1,...a_m$ eine Menge von Attributen für ein Objekt *x* aus einer *m*-dimensionalen Datenmenge *X*, also $x \in X$, so wird das Ergebnis eines Clustering Verfahrens auf *X* mit *k* Clustern eingeführt als:

$$C = \{C_1,...C_k \mid C_i \cap C_j = \emptyset_{i,j=1,...k}; C_1 \cup ... \cup C_k = X\}.$$

Definition von Frequenzen zusätzlicher Attribute: Sei *CL* ein zusätzliches Attribut für *X*, welches eine Objektklasse bezeichnet und die verschiedenen Werte $CL = \{cl_1,...cl_p\}$ annehmen kann, so wird unter Annahme der Ergebnisse eines Clusterings auf *X* nach vorheriger Definition die Frequenz von cl_i in *X* und die Frequenz von cl_i in einem Cluster C_j entsprechend definiert als:

$$\Pr(cl_i \mid X) = \frac{|\{x \mid x.CL = cl_i, x \in X\}|}{|X|} \text{ und } \Pr(cl_i \mid C_j) = \frac{|\{x \mid x.CL = cl_i, x \in C_j\}|}{|C_j|}.$$

Das Argument hinter dieser Definition ist die Annahme, dass ein Clusteringverfahren über *X* dazu führt, dass Cluster gebildet werden, deren Objekte das gleiche Klassenlabel erhalten. Dies ist in der Praxis oft nicht der Fall und es ist durchaus intuitiv, zu vertreten, dass diese Objekte als Outlier-Kandidaten anzunehmen seien. Da $\Pr(cl_i \mid C_j)$ die Rate der Objekte der Klasse cl_i innerhalb des Clusters C_j repräsentiert, ist es wahrscheinlich, dass bei einem kleinen $\Pr(cl_i \mid C_j)$-Wert eben diese Objekte Outlier sind, weil die Majorität der Objekte im Cluster naturgemäß einer anderen Klasse (ähnlichen Verhaltens) angehört.

Definition der Ähnlichkeit zwischen einem Objekt und einer Datenmenge: Bei einer gegebenen Menge X_R und einem Objekt *x* wird die (durchschnittliche) Ähnlichkeit zwischen X_R und *x* definiert als:

$$sim(x, X_R) = \frac{\sum_{i=1}^{|X_R|} similarity(x, x'_i)}{|X_R|}; \forall x'_i \in X_R.$$

Für die Ähnlichkeit zwischen zwei Objekten *x* und $x'_i \in X_R$ wird die Ähnlichkeitsfunktion des Clusteringverfahrens verwendet, auf dessen individuelle vom gewählten Verfahren abhängige Definition hier der Einfachheit halber verwiesen sei.

Definition des semantischen Outlierfaktors *SOF*: Der semantische Outlier-Grad eines Objektes *x* ist unter der Annahme, dass ein Clusteringverfahren *x* dem Cluster C_k zuordnet und dass der Klassenwert von *x* gleich cl_i sei, mit der zusätzlichen Annahme, dass X_R eine Teilmenge von *X* mit dem Klassenwert cl_i sei, definiert:

$$SOF(x) = \frac{\Pr(cl_i \mid C_k) \cdot sim(x, X_R)}{\Pr(cl_i \mid X)}.$$

$\Pr(cl_i \mid C_k)$ repräsentiert die Rate der Objekte mit dem Klassenlabel cl_i in C_k. Wenn diese Rate klein ist, weist dies darauf hin, dass solche Objekte mit Klasse cl_i Outlier sein könnten. Um eine statistische Diskriminierung von Klassen geringer Größe und damit eine mögliche Verfälschung der Ergebnisse zu vermeiden, wird die Gesamtrate der Objekte mit dem Klassenlabel cl_i in X benutzt, um diesen Effekt auszubalancieren. Das Maß $sim(x, X_R)$ beschreibt, wie sehr das Objekt *x* den Objekten derselben Klasse im Durchschnitt ähnelt. Ist dieser Wert sehr klein, impliziert dies, dass das Objekt ein Outlier ist. Ohne nun das Clusteringverfahren, welches die Funktion für die Ähnlichkeit liefert, in Betracht zu ziehen, wird der Wert von $\Pr(cl_i \mid X)$ von der Charakteristik der Datenmenge bestimmt. Geht man davon aus, dass Clustering Algorithmen Objekte aufgrund von Ähnlichkeit jeweils demselben Cluster zuordnen, und wenn grundsätzlich vorausgesetzt wird, dass Objekte gleicher Klassen ähnliches Verhalten aufweisen, so wird der Zusammenhang zwischen $\Pr(cl_i \mid C_k)$ und $sim(x, X_R)$ klar: ein kleinerer Wert für $\Pr(cl_i \mid C_k)$ führt zu kleinerer Ähnlichkeit $sim(x, X_R)$ und umgekehrt. Damit rechtfertigt dies eine Verstärkung durch Multiplikation in der Formel für den *SOF*.

> *Anmerkung:* Nach der Definition beschreibt der $SOF(x)$ den Grad der semantischen Ähnlichkeit des Objektes anhand der global nivellierten Rate, mit welcher die Objekte derselben Klasse wie *x* in dem Cluster, dem auch *x* zugeordnet ist, vorkommen; kombiniert mit der durchschnittlichen Ähnlichkeit von *x* zu all den Objekten derselben Klasse, in der sich *x* befindet. Demnach wird ein Outlier durch die Abwesenheit dieser Ähnlichkeit, also im Ergebnis durch einen sehr kleinen $SOF(x)$ beschrieben. Da die meisten anderen Outlier Detection Methoden den Grad des Outlier Status eher wachsend interpretieren, kann hier der Kehrwert zum Vergleich eingesetzt werden, z.B. $SOF^R(x) = 1/SOF(x)$, sozusagen als reziproker Semantischer Outlierfaktor.

In der Literatur wird von He, Deng und Xu [11] ein Algorithmus *FindSOF* vorgestellt, welcher im Prinzip nach Anwendung eines beliebigen Clusteringverfahrens mit einem verfahrensabhängigen Aufwand $O(H)$ die Datenmenge zweifach scannt und im ersten Scan entsprechende Zähler für die Klassenzugehörigkeiten bzgl. der Cluster und bzgl. der Gesamtmenge inkrementiert, sowie die Ähnlichkeitswerte für die Objekte gegenüber den Objekten derselben Klasse berechnet. Im zweiten Durchlauf werden dann die *SOF*-Werte für alle Objekte berechnet. Da die Autoren den Aufwand für diese Schritte mit $O(n)$ mit *n* Objekten in der Datenmenge angeben, ergibt sich ein Gesamtaufwand von $O(H + n)$ für dieses Verfahren.

Der vergleichsweise niedrige Aufwand resultiert aus der linearen Betrachtung eines Attributs, nämlich dem der Klassenzugehörigkeit und der engen Verzahnung der Ähnlichkeitsberechnungen mit dem Clustering-Algorithmus. Da hier auf Informationen zurückgegriffen wird, welche sich ggf. im Rahmen des Clusteringschritts bereits gespeichert bereitstellen lassen, reduziert dies den Aufwand des eigentlichen Outlier-Detection Schrittes.

Semantische Outlier wählen im Fazit einen Ansatz, die homogene Betrachtung des gesamten Attributraumes aufzubrechen und verschiedene Attribute auch verschieden auf Verhaltensabweichungen hin zu untersuchen. Er ist vergleichbar mit dem Ansatz der Spatial Outlier (*SPO*), der die signifikante Attributmenge für den Verhaltensvergleich in räumliche und nicht-räumliche Attribute aufteilt und die grundsätzliche Frage aufwirft, ob sich Objekte, welche sich räumlich nah beieinander befinden bzgl. ihrer nicht-räumlichen Eigenschaften im Verhalten signifikant unterscheiden. Dies stünde im Widerspruch zu ihrer räumlichen Nähe und macht sie somit als Outlier verdächtig. Gleichsam wird im Bereich semantischer Outlier die Frage gestellt, ob sich Objekte, welche sich zwar ähnlich verhalten, jedoch unterschiedlichen Klassen angehören, nicht im Widerspruch zu dem a priori anzunehmenden Verhalten bzgl. einer globalen Klassendisziplin befinden und somit Kandidaten für Outlier sind. Diese indirekte Fragestellung deckt solche Objekte auf, welche sich mit ihrem Verhalten nicht in der für dieses Verhalten typischen Region, welche durch ihre Klasse repräsentiert wird, befinden.

Beide Verfahren stellen intelligente und vor allem intuitive Fragen. In beiden Ansätzen wird der Attributraum willkürlich unterteilt und die Fragestellung auf die Objekte angewandt. Ein gutes Beispiel liefert das von He, Deng und Zu beschriebene Experiment der Wahlentscheidungsdaten im amerikanischen Zweiparteiensystem. Hier wurden die Entscheidungen von Kongressabgeordneten untersucht, welche prinzipiell zwei Klassen entsprechend ihrer Parteizugehörigkeit (Demokraten oder Republikaner) zuzuordnen sind. Outlier wurden hier als solche Abgeordnete erkannt, welche in ihrem Stimmverhalten nicht der Mehrheit der parteiüblichen Abstimmung entsprachen. Die Autoren weisen darauf hin, dass derartige Outlier von anderen Tests nicht erkannt werden. Dies ist nachvollziehbar, weil derartige Tests eher global nach solchen Abgeordneten suchen, deren Abstimmungsverhalten sich entweder generell von dem aller anderen Abgeordneten in Gänze (z.B. entfernungsbasierte Outlier) oder zumindest von dem großer Meinungsblöcke (z.B. lokale dichtebasierte Outlier) unabhängig von einer Klassenzugehörigkeit unterscheidet. Demgegenüber suchen die

benannten semantischen Outlier z.B. nach Demokraten, welche nicht wie ein typischer Demokrat abstimmen oder nach Republikanern, welche nicht wie typische Republikaner abstimmen. Eine andere mögliche Situation wäre jedoch die gleiche Fragestellung bei Einbeziehung von Wissen über solche Abstimmungen, in denen die Fraktionsdisziplin aufgehoben wurde und im Rahmen einer Abstimmung jeder Abgeordnete unabhängig von seiner Partei nur seinem Gewissen verpflichtet abstimmt. Hier sind ggf. andere Outlier zu erwarten.

2.10. Übersicht über Outlier Detection Ansätze

In der folgenden Übersicht ist noch einmal eine Anzahl von Outlier Detection Ansätzen aufgeführt. Die Liste erhebt wegen der starken Dynamik der Outlier Forschung und daraus resultierender neuer Ansätze, welche ggf. nicht betrachtet werden konnten, keinen Anspruch auf Vollständigkeit. Die Verfahren sind sporadisch nach der Zugehörigkeit zu generellen Verfahrensweisen im Ansatz geordnet, jedoch liegt dieser Ordnung keine Systematik zugrunde, aus der sich Grundsätze für die Wahl von Anwendung und Verfahren ergeben würden. Zudem wird keine Aussage über Effizienz der Algorithmen oder Quantität oder Qualität der erkannten Outlier gemacht. Dies ergibt sich insbesondere durch die noch zu lösenden Probleme im Bereich der Outlier Erkennungsverfahren. Die Wahl von Verfahren und der Grund für diese Auswahl zur Anwendung der Outlier Verfahren auf das untersuchte Thema der USENET Newsgruppen wird in dieser Arbeit im späteren näher ausgeführt.

- Verteilungsbasiert / statistisch
    - Outlier in Normalverteilungen (Freedman, Pisani und Purves) [47]
    - Outlier in univariaten Regressionsmodellen (Draper und Smith) [49]
    - Outlier in multivariaten Regressionsmodellen (Rousseeuw und Leroy) [50]
    - Outlier in Exponentialverteilungen (Pawlitschko) [17]
    - statistische Unterscheidbarkeitstests (Barnett & Lewis) [33]
    - überwachte Lernverfahren (Yamanishi, Takeuschi und Williams) [34], [35]
- Tiefenbasiert
    - tiefenbasiere konvexe Hüllen / Tiefenkonturen / *ISODEPTH* (Ruts & Rousseeuw) [36]
    - verbesserte Tiefenkonturen, *FDC* (Johnson, Kwok, und Ng) [37]
    - Schältiefen (Preparata und Shamos) [38]
- Entfernungsbasiert
    - unifizierte entfernungsbasierte *UO(p,D)* bzw. *DB(p,D)*-Outlier (Knorr, Ng, Tukakov) [3], [6], [7], [8], [51]
    - distance based / Intensional Knowledge (Knorr, Ng) [55]
    - entfernungsbasiert zu *k*-nächsten Nachbarn (Ramaswamy, Rastogy, Shim) [13], [63]
    - entfernungsbasiert mit zufälliger einfacher Beschneidung (Bay, Schwabacher) [9]
    - Extended Distance based Outlier (*EDB*) (Cheng, Fu, Tang) [19]
- Dichtebasiert
    - lokale dichtebasierte Outlier - *LOF* (Breuning, Kriegel, Ng, Sandner) [4]
    - top-*n LOF* Outlier (Jin, Tung und Han) [5]
    - dichtebasierte Outlier in Projektionen (Aggarwal und Yu) [68]
- Sequentielle Ausnahmen
    - Simple Deviation with Smoothing Factor (Arning, Agrawal, Raghavan) [73]
    - *K-d* Baum Outlier (Chaudhary, Szalay, Moore) [16]
- Clusteringverfahren und Outlier Detection
    - Clustereleminierung
    - Clustering mit Outlierbehandlung (Su, Jiang, Tseng) [56]
    - clusterbasierte lokale Outlier *CBLOF* (He, Deng, Xu) [10]
- Spatial Outlier
    - Unified Spatial Outlier (Shekhar, Zhang, Lu, (Huang)) [27], [28]
    - Graph-based Spatial Outlier (Shekhar, Zhang, Lu) [29]
    - Spatial Temporal Outlier (Cheng, Li) [46]

- Semantische und klassenbasierte Outlier
    - semantische Outlier – *SOF* (He, Deng, Xu) [11]
    - klassenbasierte Outlier (He, Deng, Xu) [108]
- Weitere Ansätze
    - Biased Sampling für Clustering und Outlier Detection (Kollios, Gunopulos, Koudas, Berchthold) [12]
    - Complementarity of Clustering and Outlier Detection (Chen, Fu, Tang) [19]
    - Outlier Detection mit Replicator Neural Networks (Hawkins, He, Williams, Baxter) [14]
    - Hypergraph basierte Outlier – *HOT* (Wei, Quian, Zhou, Jin, Yu) [15]
    - Connectivity Based Outlier – *COF* (Tang, Chen, Fu, Cheung) [18]
    - Local heuristic based Outlier search (He, Deng, Xu)
    - Frequent Pattern based Outlier (He, Deng, Xu) [110]
    - Unified Subspace Outlier Ensemble Framework (He, Deng, Xu) [109]

Eine Anzahl der angeführten Verfahren wurden in den vorhergehenden Kapiteln in bereits im Überblick bzw. auch im Detail vorgestellt. Diese Vorstellung diente dazu, einen Einblick in die Thematik zu geben, damit der Leser sich ein generelles und recht umfassendes Bild der Verfahren und der unterschiedlichen Ansätze machen kann. Eine Reihe von Verfahren wurde nicht in die Detailvorstellung übernommen, weil diese entweder keine neuen Erkenntnisse für den generellen Einblick vermitteln würden, oder weil eine weitergehende Betrachtung den Fokus der Arbeit unangemessen auf die theoretische Analyse von Verfahren verengt hätte, da im Folgenden die empirische Anwendung auf USENET News als zweiter Aspekt eingebracht werden soll.

Allerdings erhebt der Autor keinen Anspruch auf eine Wertung eines der Verfahren. Die in der Detailvorstellung vorgebrachten Wertungen entsprechen denen der Autoren der jeweiligen Verfahren, da jeder Autor sein eigenes Verfahren gegen andere Ansätze abgrenzt und dies meist mit einer Effizienzbetrachtung oder einem Vergleich der Outliererkennung quantitativ oder in engen Grenzen qualitativ verbindet. Derartige Vergleiche wurden vom Autor dieser Arbeit sodann bei Querreferenzen wieder verwendet.

Im Ergebnis kann also eine weitere Untersuchung von in Kapitel 2 nicht vorgestellten Ansätzen durchaus zu einer neuen Sichtweise auf die Verfahren in Gänze und auf die folgende Anwendung auf die USENET Newsgruppen erlauben. Dies wird hiermit nicht ausgeschlossen.

3. USENET Newsgruppen als Anwendungsdomaine

3.1. Einführung in USENET News

USENET News ist ein bekannter und weltweit stark genutzter Dienst im Internet. Das USENET System hat den Charakter eines weltweit verteilten Bulletin Board Systems, in dem Bekanntmachungen, Nachrichten und andere beliebige Inhalte in Form von Anhängen (Artikeln) an schwarzen Brettern (Newsgroups) in thematisch geordneter Form (Gruppenhierarchie) zwischen Nutzern ausgetauscht werden können. Dabei hat die Intensität der Nutzung über die Jahre hinweg stark zugenommen. Obwohl die USENET News nicht denselben medialen Schub erfahren haben wie das World Wide Web (WWW) und heute einer Vielzahl der Internetanwender offenbar unbekannt sind, haben sie sich in akademischen, öffentlichen und privaten Interessensgruppen mit ungebremster Popularität etabliert. Ursprünglich als ein Medium zum Austausch reiner Textnachrichten konzipiert, werden heute über News auch viele multimediale Inhalte in Form von Filmen, Musik, Software, etc. ausgetauscht.

Die Architektur des Newssystems ist denkbar einfach. Es kombiniert die zentrale Verfügbarkeit einer global einheitlichen und standardisierten Themenhierarchie mit der Möglichkeit, Nachrichten innerhalb dieser Hierarchie dezentral einzuspeisen, zu entnehmen, und sie mithilfe eines robusten Systems weltweit zu verteilen. Dabei ist eine globale und lokale Erweiterung der Hierarchie möglich. Erstere wird über ein quasi basisdemokratisches Verfahren möglich. Sofern sich ausreichend viele Nutzer für ein Thema interessieren, welches in der bereits vorhandenen Hierarchie unzureichend vertreten ist, bzw. welches bereits einen hohen Stellenwert ohne eigenständige Repräsentation in einer Newsgruppe hat, so kann für dieses Thema eine neue, eigenständige Newsgruppe ins Leben gerufen werden, deren Verteilung dann auch global gegeben ist. Zusätzlich können Anwender unreguliert die Hierarchie im Bereich der lokalen Verteilung erweitern, sodass dies unbemerkt von der globalen Verteilung durchgeführt wird. Hier stehen z.B. organisationsbezogene Newsgruppenhierarchien im Vordergrund, welche in Intranets von Unternehmen, Behörden oder Bildungseinrichtungen häufig anzutreffen sind. Lokale Verteiler von News, d.h. Organisationen, welche eigene Newsserver innerhalb des globalen Newssystems betreiben, entscheiden dann darüber, welche Teile der globalen Newsgruppenhierarchie sie von jeweils anderen Verteilern beziehen und welche sie an angeschlossene Dritte weiterleiten (Newsfeed). Durch die dezentralisierte Struktur und durch die Mehrfachvernetzung zentraler Player im USENET Verbund ergibt sich somit ein Verteilungssystem, welches den lokalen Bezug eines Großteils der Hierarchie für den Endanwender sicherstellt. Meist wird zudem der Verteilerbezug auf Anforderung von Endnutzern erweitert, sodass die Feeds sich dynamisch den Interessen der Nutzer anpassen.

Die Verteilung der News zwischen den so genannten Newsservern wird standardmäßig über das Network News Transfer Protocol – NNTP [81] realisiert, welches in einem Request for Comment (RFC) der Internet Engineering Task Force (IETF) spezifiziert ist und die Grundlage für die Implementierung der Newssysteme bildet. Abbildung 28 zeigt eine grafische Umsetzung eines Ausschnitts der USENET Topologie. Da innerhalb der Newsgruppen so genannte Newsartikel durch den Vorgang des „Postings“ versandt werden, entspricht dies der Eingabe von Informationen in einer Nachricht durch das Anheften an ein global verteiltes schwarzes Brett. Der Artikel ist dann für die gesamte Leserschaft dieser Gruppe zu sehen. In diese Leserschaft kann sich ein Anwender durch das Abonnieren einer Newsgruppe integrieren und hat sodann zum einen die Möglichkeit, mit einem Newsreader Programm (entspricht heute in der Ausgestaltung einem modernen Mail- oder Browserprogramm (z.B. Outlook Express) bzw. ist in solche bereits eingebunden (z.B. Mozilla)) diese Artikel zu lesen und kann zum anderen auf solche Artikel antworten bzw. eigene Artikel in das System einstellen.

Aus dieser Methodik ergibt sich die Möglichkeit, innerhalb einer Newsgruppe Themen dieser Newsgruppe selbst zu diskutieren. Dies geschieht durch wiederholtes anheften von Artikeln an das entsprechende schwarze Brett, d.h. die Newsgruppe, wobei sich der Bezug sowohl auf den jeweils letzten geposteten Artikel ergeben kann, oder aber auf mehrere bzw. alle vorangegangenen. Dadurch entsteht ein Diskussions-Thread, in welchem der generelle Aspekt oder auch Teilaspekte im Zusammenhang, getrennt voneinander oder durch Zusammenführen wiederverbindend diskutiert werden können. An dieser Diskussion können beliebig viele Nutzer teilnehmen.

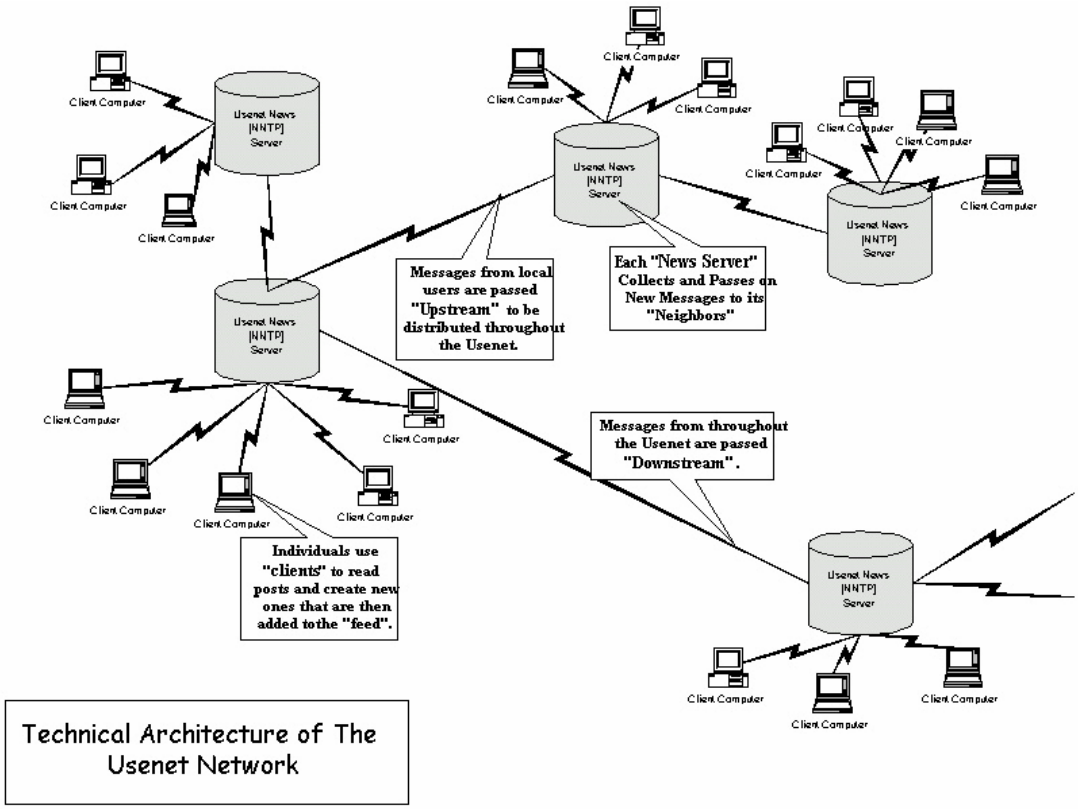


Abbildung 28 - USENET Topologie (Ausschnitt)

Abbildungsbeschreibung: Die Topologie weist das USENET News System als ein global verteiltes dezentrales System aus, dessen lokale Verästelungen sowohl als Quelle und auch als Senke der vom System übertragenen Nachrichten fungieren. Die Verwaltung der Datenströme erfolgt ebenfalls dezentral nach global gültigen Grundsätzen der Kaskadierung der Nachrichtenübermittlung.

Die Themen der Newsgruppen sind hierarchisch in einem Gruppenbaum geordnet. Hierbei wird unter der Baumwurzel global in 3 unterschiedliche Bereiche verzweigt. Zum einen gibt es den Bereich der großen acht Hierarchien comp, humanities, misc, news, rec, soc, sci und talk, die so genannten „Big-8“. Innerhalb dieser Hierarchienen sind Gruppen thematisch in Teilbäumen angeordnet. Die Regeln und Anforderungen für die Einrichtung neuer Gruppen sind in diesen Bereichen relativ strikt. In einem zweiten großen Bereich, den alternativen Gruppen unter alt.*, ist die Gestaltung der Gruppenhierarchie sehr frei möglich und daher die Struktur auch nicht so sehr geordnet bzw. konsistent. Oft werden Themenbereiche aus den Big-8 hier noch einmal im Rahmen einer freieren Diskussion repliziert. Der dritte Bereich beschreibt die landesspezifischen Gruppenteilbäume. Während in den Big-8 und im alt.* Bereich Englisch als Sprache dominiert, ist die Assoziation landesspezifischer Inhalte mit der entsprechenden Sprache des Landes einer Gruppenteilhierarchie sehr viel stärker. Eine sprachliche Einschränkung wird jedoch nicht verordnet, sondern es ist den Nutzern überlassen, dies im Rahmen von „Angebot und Nachfrage“ zu regeln. Im Folgenden ist eine Auflistung der Hierarchien gezeigt:

- Big-8 Hierarchien
  - comp.* Computer, Software, Hardware, Betriebssysteme
  - humanities.* Geisteswissenschaften
  - misc.* Vermischtes, Sonstiges
  - news.* Verwaltung der Newsgruppen
  - rec.* Freizeitaktivitäten, Hobbies etc.
  - soc.* Soziale Themen
  - sci.* Natur-, technische und mathematische Wissenschaften
  - talk.* Smalltalk, Unterhaltungen
- .alt Hierarchie
- Landesspezifische Hierarchien (de., fr., usw.)
- Lokale Hierarchien t-online., uni-dortmund., thd., usw

In der letzten Zeile der Aufführung der Gruppenteilbäume wird noch einmal auf die lokalen Distributionen verwiesen. Da diese i.d.R. nicht global transportiert werden, ist die Organisation, welche diese Teilbäume verwaltet, in der Gestaltung sehr frei. Oft orientiert sich die Unterhierarchie lokaler Distributionen an den Anforderungen der Organisation, d.h. ihrer Gestaltung bzw. auch bzgl. ihrer maßgebenden Themen.

Das Format für den Austausch von Newsartikeln ist auch in einem RFC zu USENET News [80] beschrieben. In diesem Request for Comment, einer de facto Standardisierungsform, welche wie viele andere im Verbund die Grundpfeiler der Funktion des Internets beschreibt, sind sowohl die technischen Aspekte der Verteilungsmethodik berücksichtig, als auch Teile der Verbindung von Artikeln mit einer Newsgruppe und einer themenbezogenen Diskussion innerhalb eines Threads. Allerdings ist der letztere Aspekt nur sehr rudimentär realisiert, was in der fortlaufenden Betrachtung von News innerhalb dieser Arbeit noch deutlich wird. Da es sich bei den Newsartikeln generell um sog. Klartextnachrichten handelt, erfolgt die Beschreibung dieser Aspekte durch standardisierte Feldbezeichner in einem Header für den Artikel. An diesen Header schließt sich ein Textkörper (Body) an, welcher innerhalb des Headers auf eine dort bestimmte Zeilenzahl spezifiziert ist, jedoch beliebigen Inhalt haben kann. Ein Tail bzw. ein Footer für einen Artikel ist nicht spezifiziert. Insofern ähneln Newsartikel in ihrer Ausprägung sehr deutlich der Ausprägung einer normalen eMail.

Im Folgenden ist ein Beispiel eines Newsartikel-Headers (in Auszügen) dargestellt.

```
Path: gate.krell.zikzak.de!not-for-mail
From: cg@garbers.org (Christoph Garbers)
Reply-To: cgarbers@gmx.de (Christoph Garbers)
Sender: de-newusers-infos@spamfence.net
Subject: <2003-02-20> Headerzeilen
Newsgroups: de.newusers.infos,de.newusers.questions
Followup-To: de.newusers.questions
Message-ID: <de-newusers-infos/headerzeilen/20050708-1@krell.zikzak.de>
Supersedes: <de-newusers-infos/headerzeilen/20050701-1@krell.zikzak.de>
Expires: Sat, 08 Oct 2005 22:00:21 +0000
Approved: de-newusers-infos@spamfence.net
MIME-Version: 1.0
Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: 8bit
Organization: Moderation von de.newusers.infos
Archive-name: de-newusers/headerzeilen
Posting-frequency: weekly
Last-modified: 2003-02-20
URL: http://www.cgarbers.de/usenet/headerzeilen.txt
URL: http://www.kirchwitz.de/~amk/dni/headerzeilen
```

Ein wichtiger Aspekt der Newsgruppen ist die Tatsache, dass das System selbst auf einer zeitlich begrenzten Vorhaltung der Artikel basiert. Somit durchfließen Newsartikel das System und werden automatisch nach einer gewissen Zeit vom Brett abgehängt. Dies begünstigt zum einen den bewusst als Ziel festgelegten Charakter der Aktualität – daher auch der Name des Systems: „News“. Zum anderen sorgt es für eine Minimierung der Resourcen, welche erforderlich wären, die sonst stetig anwachsende Menge an Artikeln in der auch wachsenden Hierarchie überall dezentral vorzuhalten bzw. kontinuierlich zu verteilen. Seit geraumer Zeit gibt es im Internet Dienste und Systeme, welche es sich zur Aufgabe gemacht haben, die Inhalte der USENET News genau wie die Inhalte des World Wide Web zu archivieren. Damit ist zwar ein prinzipieller Zugriff auf einen großen Teil der Inhalte der News auch langfristig möglich, allerdings außerhalb des Systems und ohne den angeschlossenen Diskussionsaspekt innerhalb des USENET Systems. Dieses selbst lebt in seiner Dynamik von der großen Zahl der teilnehmenden Anwender und respektive der Vielzahl an diskutierten Themen mit einem hohen Aktualitätsbezug.

Ein weiterer wichtiger Aspekt ist die generelle Unterteilung von Newsgruppen in moderierte und unmoderierte Gruppen. Während in einer unmoderierten Gruppe jeder Nutzer frei seine Inhalte an das schwarze Brett heften kann, also einen Artikel in die Gruppe postet, werden Postings in moderierten Gruppen zuerst vom System automatisch an einen Moderator für diese Newsgruppe geschickt. Dieser entscheidet dann, ob das Posting die Anforderungen, welche für die Newsgruppe von den Moderatoren oder von den Nutzern bei der Einrichtung eben dieser Gruppe festgeschrieben wurden, erfüllt, und es somit vom Moderator selbst in die Gruppe gepostet werden kann. Moderierte Gruppen sind am Namen, der auf .moderated endet, zu erkennen. Der größte Teil der USENET News ist allerdings unmoderiert.

Für weitergehende Ausführungen zu News jenseits dieser kurzen Einführung sei auf die einschlägige Literatur ([82] und [83]) zum Thema verwiesen. Zusätzlich bietet das Internet eine Vielzahl von Quellen[8], welche sich mit der Anwendung von News im Allgemeinen und den Inhalten spezieller Gruppen im Besonderen beschäftigen. Hierzu gibt es z.B. in jeder Newsgruppe einen „lebenden", d.h. von Initiatoren der Gruppe gepflegten FAQ (Frequently Asked Questions). Der für die Erläuterung der Headerzeilen verwendete Artikel verweist z.B. auf eine spezielle Newsgruppe, welche Informationen über USENET News zum Inhalt hat und sich an neue Nutzer richtet.

3.2. Motivation für Outlier Detection in USENET Newsgruppen

Für die Themengestaltung dieser vorliegenden Arbeit ist es natürlich von zentraler Bedeutung, die wissenschaftliche Motivation der Untersuchung von Outliern in Newsgruppen zu erklären. Diese leitet sich aus zwei grundlegenden Aspekten ab. Zum einen ergibt sich die Notwendigkeit wissenschaftlicher Bertrachtung bei USENET News durch den praktischen Charakter der Aufgabenstellung. Ähnlich wie bei eMail ist eine Erkennung von Outliern in den Newsgruppen und die Identifizierung von Nachrichten mit Outlier-Charakter prinzipiell nicht nur sinnvoll, sondern auch sehr nützlich. Dies wird im Folgenden durch Statistiken begründet.

Zum anderen erscheint die Betrachtung von USENET News aufgrund der Eigenschaften der Nachrichten als Objekte in einem aufgespannten Nachrichtenraum durch die Spezifika der News, auf welche im Kapitel 4 detailliert verwiesen wird, viele der grundlegenden Fragen zu Outlier Detection Ansätzen und Verfahren aufzuwerfen. Da News Nachrichten (bzw. Artikel) eine hohe Anzahl an Attribut-Dimensionen haben, sofern der Gesamtraum aus festen Attributen (Nachrichtenfeldern) oder z.B. der Vektorisierung der Freitextteile (Body eines Artikels) betrachtet wird, stellt Outlier Detection in USENET News hohe Anforderungen an die Verfahren und wirft eine Reihe von Problemen auf, welche die Verfahrenseignung als auch die Erwartung an identifizierbare Outlier betrifft.

Beide Aspekte lassen eine wissenschaftliche Untersuchung nützlich erscheinen, zumal im Rahmen von Outlier Verfahren (vgl. Kapitel 2) oft wenige gleichartige Beispiele (Spitzensport, synthetische Datenmengen) oder sehr spezialisierte Anwendungen (Geologie, Verkehrsanalyse) betrachtet werden. Auf der anderen Seite kann diese Arbeit aufgrund des Anspruchs einer Diplomarbeit nur eine anfängliche Betrachtung der aufgeworfenen Probleme liefern, jedoch keine vollumfänglichen Lösungen anbieten. Wie bereits in Kapitel 1.2 angeführt, ist eine der am tiefsten greifenden Fragestellungen der Outliererkennung die, warum ein Objekt als Outlier erkannt worden ist (abgesehen von der Erfüllung der Kriterien, welche das Verfahren an die Identifizierung stellt) und welche weiteren Aussagen insbesondere auf die Entdeckung bisher unbekannten Wissens nicht nur intuitiv sondern vor allem formal vollständig abgeleitet werden können. Da a priori vom Autor dieser Arbeit keine Aussage gemacht werden kann, warum Outlier erkannt werden – da ein umfängliches Hintergrundwissen zu den Inhalten, d.h. dem Wissen, um die als Outlier entdeckten Newsartikel nicht vorausgesetzt werden kann – kann die Eignung bzw. die Ergebnisse der Verfahren auch nur empirisch betrachtet werden. Im Fazit kann dann eine Aussage darüber getroffen werden, welche Objekte jeweils von einem Verfahren in einer gegebenen Testmenge als Outlier erkannt wurden. Und es können Vergleiche angestellt werden, welche Outlier von mehreren Verfahren erkannt wurden bzw. welches Verfahren quantitativ auf derselben Testmenge wie viele Outlier erkannt hat. Eine qualitative Analyse der Outlier und damit ein qualitativer oder streng formaler Rückschluss auf die Verfahren ist nach dem derzeitigen Stand der Outlier-Forschung in dieser Arbeit nicht möglich und daher auch nicht Teil der Betrachtung. Vielmehr stellt die Arbeit auf den experimentellen Vergleich mit nach gewissen Kriterien vorkategorisierten Outlierkandidaten ab, ohne daraus eine formale Verfahrensbewertung herzuleiten (vgl. Kapitel 4.4).

Zurückkommend auf den ersten Aspekt zeigen USENET News pro Monat eine Anzahl von global ca. 100 Millionen Postings, d.h. jeden Monat kommt somit ein beträchtliches Volumen, welches ca. 20 bis 30 Gigabyte an Rohdaten entspricht, an neuen Artikeln in die Landschaft der thematisch geordneten schwarzen Bretter. Abbildung 29 zeigt diese Situation[9].

Es sind momentan keine detaillierten statistischen Untersuchungen zur Verteilung dieses Artikelvolumens auf die einzelnen Newsgruppen, von denen es mehrere zehntausend mit wachsender Tendenz gibt, bekannt. Jedoch kann angenommen werden, dass das Nachrichtenaufkommen bei stärker frequentierten Newsgruppen in einer so hohen Anzahl an Artikeln resultiert, dass eine automatische Erkennung von zugehörigen und vor allem nicht-zugehörigen Artikeln eine erhebliche Erleichterung für die Anwender bedeuten könnte. Auf der Seite der moderierten Gruppen könnte so ein Moderator unterstützende Informationen bei der Zuordnung neuer Artikel gewinnen, was seinen Entscheidungsprozess gleichsam erleichtern und beschleunigen würde.

[8] Siehe auch: http://www.newsadmin.org/ (u.a. Quelle der in dieser Arbeit abgebildeten Statistiken zu News)
[9] © Pathlink Technology Corporation / Quelle: www.newsadmin.org, Stand Juli 2005

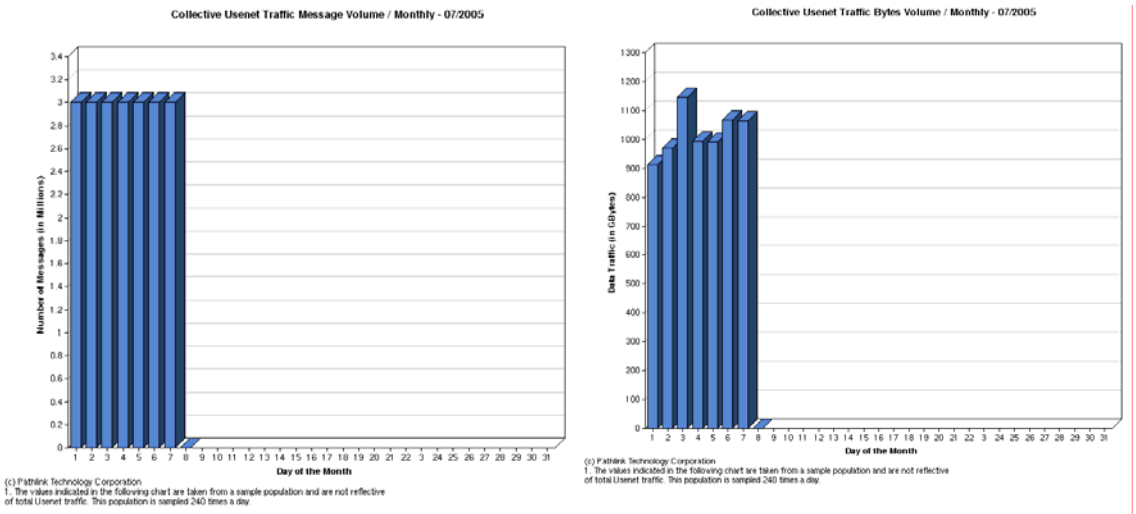


Abbildung 29 - USENET News Statistiken

Abbildungsbeschreibung: Über das USENET News System werden monatlich mehr als 100 Millionen Nachrichten ausgetauscht, die einem Rohdatenvolumen von 20-30 Gigabyte entsprechen.

Vor allem aber in nicht-moderierten Gruppen ist es für einen Anwender ohne das Hintergrundwissen eines Moderators oft von entscheidender Bedeutung, zugehörige und damit relevante Artikel von nicht zugehörigen Artikeln zu trennen. Hierbei geht die Betrachtung weiter, als die bloße Erkennung von sog. SPAM Postings.

Eine detailliertere Analyse der in Abbildung 30 dargestellten Statistik zeigt, dass über USENET News nicht nur reine Textnachrichten ausgetauscht werden. Vielmehr hat sich das USENET zu einem Medium für den Austausch von multimedialen Inhalten entwickelt. Ähnlich den Peer-to-Peer Netzwerken oder sogenannten Multimedia-Mailinglisten werden komplette Audio- und Videodateien oder auch Software-Pakete über Newsartikel übermittelt.

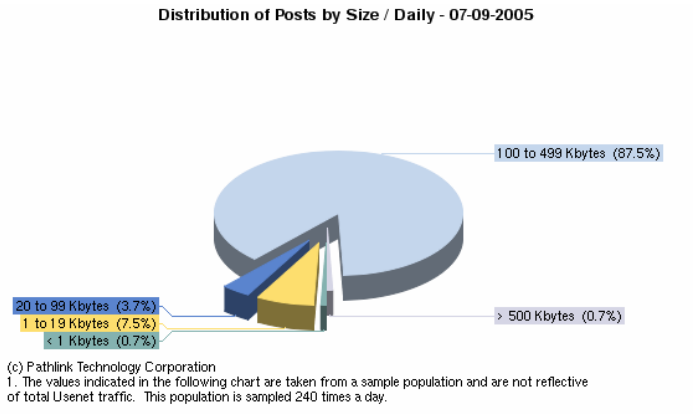


Abbildung 30 - Verteilung der Artikelgrößen von News

Abbildungsbeschreibung: Ein Großteil der Nachrichten in den USENET News enthält multimediale Inhalte, welche über viele Nachrichten verteilt übermittelt werden. Dies ist an der Größe der Nachrichten nachvollziehbar.

Da dies aufgrund der Beschränkungen von Newsartikeln technisch nicht voll transparent zu bewerkstelligen ist, wird hierfür eine Overlay-Technik angewendet. Die Multimediadateien werden zuerst für eine reine Textübermittlung neu kodiert und sodann mittels verschiedener Verfahren in für den Transport geeignete Teilpakete zerschnitten. Diese werden sodann über die Newsgruppen verteilt. Auf Empfängerseite werden diese verschiedenen Artikel dann mit einem spezialisierten Newsreader (z.B. newsbin) gezielt zur Zusammenstellung

eines Gesamtpaketes abgerufen, heruntergeladen, wieder zusammengesetzt und in binäre Form rekodiert. In der Regel wird gleichzeitig eine Reihe von Checksummendateien übertragen, um bei Verlust oder Beschädigung von Artikeln eine entsprechende Reparatur empfängerseitig zu ermöglichen.

Im Ergebnis propagiert das Newssystem also eine Reihe von Artikeln, welche ohne sinnvollen textuellen Inhalt – für den Leser versteckt – große Datenmengen übertragen. Durch die zentrale Verteilung (ein Einspeisepunkt) und das dezentrale Abrufmodell (Leser weltweit beziehen vom nächstgelegenen Newssystem) ist diese Distributionsmethode trotz der technischen Widrigkeiten sehr effizient. Nachteilig wirkt sich lediglich die nur temporäre Sichtbarkeit der für die Zusammenstückelung notwendigen Inhalte aus.

Derartige Übertragungen sind statistisch in Artikeln mit hohem Volumen größer 100 kByte pro Artikel zu vermuten. Da normalerweise ein Artikel mit News-Charakter einen Umfang mehrerer Textseiten nicht überschreitet (10 Volltextseiten a 80 Zeichen pro Zeile und 55 Zeilen pro Seite belegen nicht mehr als 44 kByte) und selbst ein Artikel mit eingebettetem illustrativen Bild (GIF oder JPEG Format mit Bild/Text Kombination) im Regelfall kleiner 90 kByte ist, kann o.B.d.A. geschlossen werden, dass Artikel mit einer Größe von mehr als 100 kByte entweder einen primär bildlichen Inhalt haben (Multimedia-Bilddatei) oder geplittete Multimediadateien transportieren. Dieser Bereich bildet ca. 90 Prozent des transportierten Volumens. Da eine Outlier-Erkennung für diesen Bereich jedoch einen direkten Zugriff auf die rekodierten Inhalte braucht, diese aber im Newssystem transparent nicht zur Verfügung stehen, wird dieses Teilthemenfeld innerhalb der vorliegenden Arbeit ausgeklammert.

Circa 8 bis 10 Prozent der News sind somit statistisch gesehen reine Textnachrichten oder Nachrichten mit einem primär textorientierten Charakter. Abbildung 31 zeigt noch einmal, dass diese Verteilung auch über die Zeit stabil und relativ gleich bleibend ist und daher die getroffenen Schlussfolgerungen stützt.

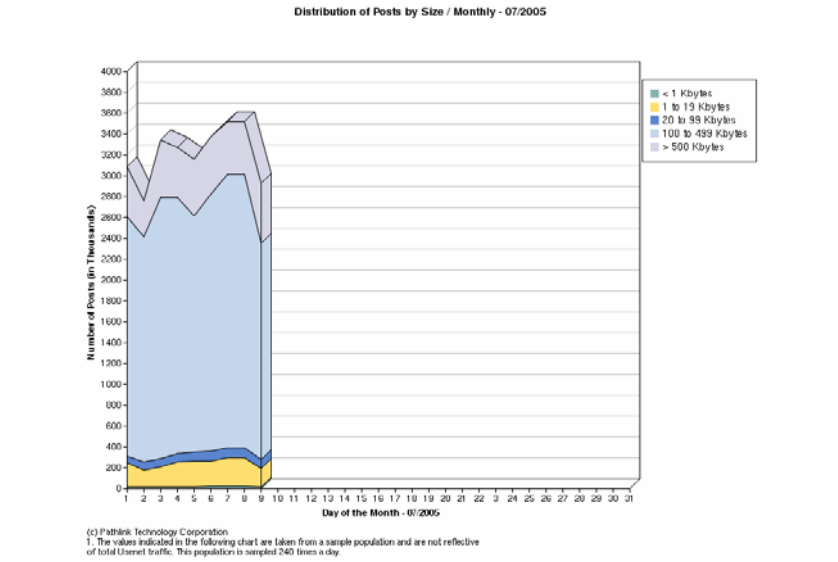


Abbildung 31 - Zeitliche Verteilung der Nachrichtengröße in Newsgruppen

Abbildungsbeschreibung: Zwischen 8 und 10 Prozent der Nachrichten sind reine Textnachrichten, wobei die Statistik zeigt, dass dieser Anteil stabil ist.

Die Erkennung von SPAM ist mit Sicherheit eine primäre Aufgabe bei der Benutzung von USENET News und daher ein ideales Anwendungsfeld für Outlier-Detection Verfahren. Zwar belegt die Statistik nur einen Anteil von rund 3 Prozent SPAM in der Gesamtmenge der neuen Nachrichten in den News (vgl. Abbildung 32). Wird jedoch zugrunde gelegt, dass der reine Textanteil zwischen 8 und 10 Prozent beträgt, wobei gesplittete Nachrichten mit kodierten Multimedia-Inhalten ausgeschlossen, Nachrichten mit Einzelbildern jedoch noch einbezogen werden, so ist der reale Anteil von SPAM in USENET News bei ca. 30 Prozent anzusiedeln. Die Erkennung eines so hohen Anteils an SPAM ist natürlich sinnvoll. Inwieweit die Outlier-Detection Verfahren hier Hilfestellungen bieten können, wird eine empirische Betrachtung mit der experimentellen Anwendung mehrerer Verfahren zeigen können. Durch den hohen Anteil von SPAM in manchen Gruppen könnte jedoch (theoretisch betrachtet) das Problem entstehen, dass SPAM durch die Masse an Objekten nicht notwendigerweise als Outlier betrachtet wird, da gleichartige SPAM-Objekte ggf. einen signifikanten Teil der Gesamtmenge dominieren und daher als Outlier nach Hawkins [45] nicht in Frage kommen, wenn die

Andersartigkeit des zugrunde liegenden Mechanismus nicht in Erscheinung tritt, sofern der Mechanismus „SPAM“ zu den Hauptmechanismen der Datenmenge zugeordnet werden muss.

Die Erkennung und Bekämpfung von SPAM nimmt in anderen Kommunikationsbereichen, wie z.B. bei eMail, bereits einen breiten Raum in der Forschung, aber auch in der praktischen Anwendung ein. Dahinter steht das Ziel, der kontinuierlich wachsenden Rate von SPAM in normalen eMail Nachrichten und den daraus resultierenden enormen Kosten in Organisationen Herr zu werden. Gleichzeitig werden über SPAM in eMails auch viele Viren, Würmer und andere elektronische Schädlinge verbreitet, sodass sich SPAM Bekämpfung gleich zweifach rentiert. Allerdings liegt der SPAM Erkennung [84] bei eMail eine andere Idee, als die der Erkennung von Outliern zugrunde. Es wird versucht, anhand von Attributeigenschaften durch Heuristiken oder auf Basis von historisch vorhandenen Daten die Wahrscheinlichkeit der SPAM-Eigenschaft einer Nachricht zu bestimmen. Darauf aufbauend wird der Nachricht meist ein Grad, SPAM zu sein, zugeordnet. Allerdings wird die Restmenge der Nachrichten lediglich auf die SPAM-Eigenschaft hin überprüft, die Inhalte von Objekten in einer Betrachtungsmenge werden aber nicht immer in Beziehung gesetzt.

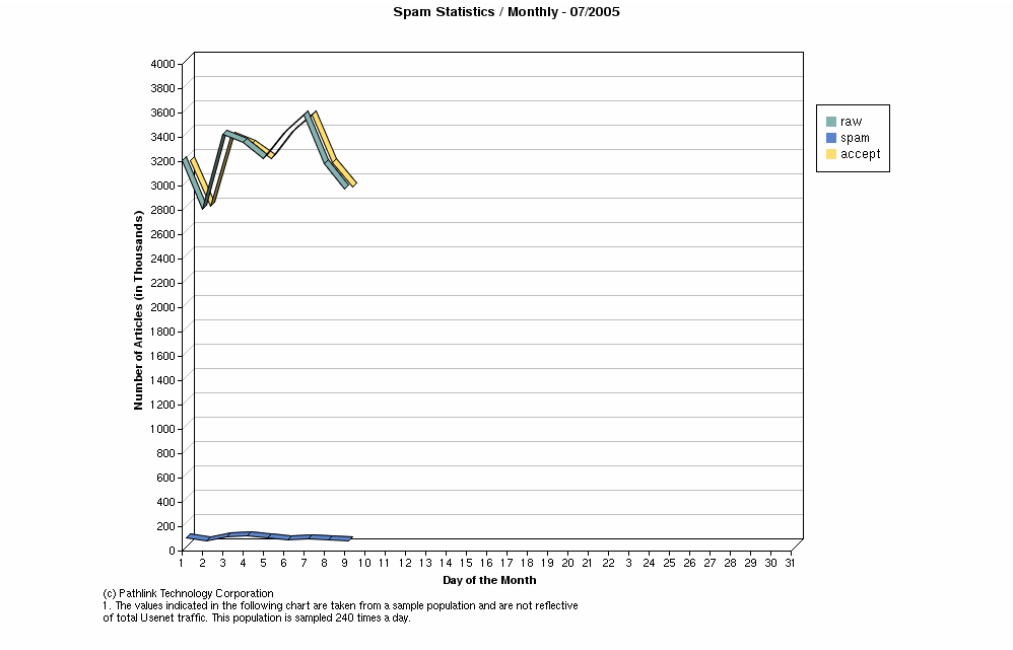


Abbildung 32 - Monatliche Spam Statiskiken für USENET News

Abbildungsbeschreibung: Zwar beträgt der Anteil an SPAM in USENET News Nachrichten nur ca. 3 Prozent, da jedoch lediglich 8-10 Prozent der Nachrichten auch Textnachrichten sind und SPAM genau in diese Gruppe fällt, ist von ungefährt 30% SPAM effektiv auszugehen.

Daher sollte die Outlier-Betrachtung nicht auf die Erkennung von SPAM in News reduziert werden. Zwar kann die Identifikation von SPAM einen Teilaspekt in der Erkennung von Outliern darstellen, vor allem, weil sich die SPAM Eigenschaft einer Nachricht in USENET News ganz wesentlich von der in einer eMail unterscheidet und viel intensiver mit der Thematik der Gruppe und mit den Inhalten der anderen Nicht-SPAM Objekte in dieser Gruppe verbunden ist. Jedoch geht Outliererkennung über dieses Themenfeld hinaus, da vor allem wissenstechnisch relevante Ausnahmen der Regel von Interesse sind. Dies kann z.B. die Erkennung inhaltlich zugehöriger aber trotzdem abweichender Objekte, z.B. in erfahrungsgemäß komplett SPAM-freien Newsgruppen, umfassen.

4. Outlier Detection in USENET News

Für die erfolgreiche Identifizierung von Outliern in Newsgruppen ist zunächst eine Untersuchung des grundlegenden Aufbaus von Newsartikeln im Detail notwendig, um mögliche Attributierungen zu bestimmen oder erfolgversprechende Verfahren für den Aufbau von Annotationen für nicht attributierte Teile zu wählen. Anhand dieser Attribute und Annotationen können dann passende Verfahren aus algorithmischer Sicht gewählt werden. Dabei erstreckt sich diese Wahl jedoch nur auf eine Auswahl anhand von verschiedenen, möglicherweise passenden statistischen Maßen und auf Verfahren, deren algorithmische Eigenschaften geneigt scheinen, sich besonders leicht auf Newsgruppen anwenden zu lassen. Es wird keine Aussage über eine formelle Auswahl anhand eines die Newsgruppen vollständig beschreibenden Systems getroffen. Auch ist a priori nicht bekannt, welche Outlier von diesen gewählten Verfahren identifiziert werden und welche qualitativen Eigenschaften diese Outlier haben. Da es um die Identifizierung von teilweise unbekanntem Wissen geht, wäre selbst ein Moderator, welcher den konkreten Kontext einer Newsgruppe und ihrer Nachrichten über eine lange Zeit nahezu vollständig verfolgt hat – und dem aus diesem Grund ein hohes Maß an Hintergrundwissen zur Newsgruppe, den Diskussionsteilnehmern und den Themen dieser Gruppe theroretisch zugestanden werden kann – somit nur schwer in der Lage, die qualitativen Eigenschaften aller erkannten Outlier umfassend oder formell zu beschreiben.

Die vorliegende Arbeit erhofft sich aber durch die Anwendung von Outlier Verfahren auf Newsgruppen empirische Erkenntnisse, welche einen Vergleich der Verfahren erlauben, der auf die quantitative Erkennung und die algorithmische Effizienz abzielt. Zwar sind im Bereich der unterschiedlichen Ansätze Vergleichstests zwischen einzelnen Verfahren meist an synthetischen Testmengen durchgeführt worden, jedoch erfolgten diese Tests bereits im Hinblick auf Effizienzeigenschaften der Algorithmen. Selten, d.h. nur in einigen Fällen, sind den Vergleichstests rein praktische Testmengen zugrunde gelegt. Und auch dann ist das Ziel der Tests meist vorherbestimmt und soll die besseren Effizienz- und Erkennungseigenschaften eines Verfahres gegenüber anderen, stellenweise ähnlichen Vorgängerverfahren belegen. Es bleibt abzuwarten, ob die Vergleiche von (unterschiedlichen) Verfahren anhand von Newsgruppen neue Erkenntnisse bringen werden und welche Schlussfolgerungen daraus abgeleitet werden können.

4.1. Feldbeschreibungen und Attribute von Newsartikeln

Wie bereits ausgeführt, bestehen Newsartikel als Objekte aus zwei Teilen, einem Header und einem Body. Der Header wiederum besteht aus (Attributbezeichner/Attributinhalt)-Paaren. Er muss eine gewisse Anzahl obligatorischer Paare aufweisen, um erfolgreich im Newssystem transportiert werden zu können. Andere Paare sind optional, d.h. deren Verfügbarkeit kann nicht vorausgesetzt werden. Die Attributbezeichner und die Attributinhalte gehorchen Vorschriften, welche im RFC 1036 [80] festgelegt sind. Der Body einer Newsnachricht (Artikel) folgt dem Header nach einer Leerzeile und enthält Freitext.

Die folgenden Feldbezeichner im Header eines Newsartikels sind obligatorisch

Feldbezeichner	**Beschreibung**
From	Die „From" Zeile enthält die elektronische Mailadresse der Person, welche die Nachricht gesendet hat, in Internet Syntax (entspricht RFC 822 [85]). Diese Person ist innerhalb des Newssystems mit der originären „Quelle" der Newsgruppennachricht gleichbedeutend. Wichtig ist in diesem Zusammenhang, dass unterschiedliche textliche Darstellungen gleichbedeutend sein können, da der Name des Senders und Kommentare an unterschiedlichen Stellen standardkonform stehen können und die transmittierenden Systeme diese Zusatzkommentare nicht weitervermitteln müssen, um die USENET Funktionalität zu erhalten. Bezüglich einer möglichen Attributauswertung müsste also eine Entscheidung getroffen werden, ob eine Auswertung auf der Datenbasis des lokal vorliegenden Newsartikels getroffen werden soll, oder ob eine globale Sicht geltend gemacht wird. Hier würde die Auswertung sich auf das unveränderliche Kernelement beschränken. Dieses umfasst die Adresse in der Syntax <user@domain>, wobei „user" sensitiv gegenüber Groß- und Kleinschreibung ist, d.h. „user" unterschiedlich zu „User" ist, „domain" hingegen nicht, also „Domain" und „domain" gleichwertig ist. Auch hier muss entschieden werden, ob ggf. eine Normierung von „domain" auf Kleinschreibung vorgenommen werden soll. Mit der Einführung sog. Internationalized Domain Names (IDN) lt. RFC 3490 [86] sind zusätzlich nun viele kodierte Zeichen jenseits des ASCII Standards möglich, wobei diese in der Domain nicht im Klartext dargestellt werden. Die „From" Zeile wird nur

ausgewertet, wenn keine „Sender“ Zeile vorhanden ist. Ansonsten wird sie ignoriert.

Date — Das „Date“ Feld spezifiziert den Zeitpunkt, an dem eine Nachricht in das USENET Newssystem hineinpropagiert wurde, entspricht also einem Sendedatum. Dieses Datum wird nicht verändert und kann ggf. innerhalb des Systems aber auch für Auswertungen dazu dienen, die zeitliche Sendefolge von Nachrichten zu sequentialisieren und dies für Ordnungsprinzipien zu verwenden.

Newsgroups — Das „Newsgroups“ Feld enthält eine durch Kommata separierte Liste der Newsgruppen, in welche die Nachricht gepostet wurde. Wenn mehrere Newsgruppen spezifiziert sind, handelt es sich um ein sogenanntes Cross-Posting. Dies ist für eine Auswertung eine mglw. wertvolle Information, sofern aus ihr eine allgemeinere thematische Ausrichtung eines Newsartikels gefolgert werden könnte.
Wichtig: Newsgruppen, welche nicht existieren oder Wildcard Einträge für mehrere Newsgruppen sind nicht erlaubt. Sie werden jedoch nicht verändert bzw. gelöscht und einfach i.d.R. vom System ignoriert. Dies muss bei der Auswertung berücksichtig werden.

Subject — Die „Subject“ Zeile enthält einen Titel für die Newsnachricht. Obwohl verschiedene Netiquetten auf einen sinnvollen Umgang mit Subject Zeilen (vgl. auch eMail) verweisen, ist dieser standardmäßig nicht verordnet. So wird z.B. bei Antwort-Postings auf Newsartikel eines bestimmten Subject der Vorsatz „Re:“ empfohlen, er ist jedoch nicht verpflichtend. Meist gibt der verwendete Newsreader diese Nutzung vor, jedoch ist global nicht sichergestellt, ob Vervielfachungen „Re: Re: Re: Subject“ oder Sprachmischungen („Aw:“) oder das manuelle Löschen des „Re:“ ausgeschlossen werden können.
Sinnvoll ist bei einer Antwort die Vorschrift des zusätzlichen „References“ Headerfelds. Dadurch kann ggf. ein direkter Thread-Bezug zwischen Artikeln hergestellt werden. Bei einem Follow-Up Posting ist die Nutzung des „Summary“-Feldes empfohlen, jedoch nicht vorgeschrieben, was die Nutzbarkeit entsprechend einschränkt.

Message-ID — Die „Message-ID“ ist ein unverwechselbarer Identifier für eine Newsgruppennachricht. Somit dürfen keine Message-IDs verwendet werden, sofern andere Newsartikel diese bereits verwenden. Danach wird empfohlen, eine ID erst nach 2 Jahren wieder zu verwenden, dies ist aber nicht vorgeschrieben. Somit kann davon ausgegangen werden, dass keine „lebende“ Nachricht eine mit einer anderen „lebenden“ Nachricht gleiche ID hat, allerdings kann bei historischer Betrachung nicht ausgeschlossen werden, dass zwei Artikel zufällig dieselbe ID besitzen. Das Format der ID ist <unique@full-domain-name>, wobei „unique“ ein ASCII String beliebigen Inhalts und Länge ist. „full-domain-name“ beschreibt den kompletten Domainnamen inkl. Hostname des Senders, kann sich also von „domain“ in der From-Zeile einer Nachricht durchaus unterscheiden.

Path — Das „Path“ Feld beschreibt den Pfad eines Artikels, d.h. die Stationen, welche eine Nachricht durch das Newssystem genommen hat, um die Station zu erreichen, an der sie z.B. mit einem Newsreader ausgelesen wird.

Die folgenden Feldbezeichner im Header eines Newsartikels sind optional

Feldbezeichner — **Beschreibung**

Reply-To — Das „Reply-To“ Feld hat den Zweck, als Adressangabe für Nachrichtensendungen direkt an den Autor des Newsartikels verwendet zu werden. Es hat dasselbe Format wie die „From“ Zeile. Ist dieses Feld nicht vorhanden, werden Nachrichten ggf. an die Adresse in der From-Zeile gesendet.

Sender — Dieses Feld ist nur präsent, wenn der Versender der Nachricht manuell eine „From“-Zeile eingibt, die ggf. von der automatisch erzeugten Senderadresse des versendenden Systems abweicht. Da es eingesetzt wird, um die Verantwortlichkeit für das Senden der Nachricht zu dokumentieren, sollte die Software des sendenden Hosts diese Adresse verifizieren. Gleichzeitig erlaubt dieser Mechanismus von From/Sender Zeilen einem Anwender, von unterschiedlichen Systemen aus – oder unter Nutzung verschiedener Accounts – Nachrichten unter dem eigenen Namen zu versenden (From). Das System verzeichnet sodann auch die Netzwerkeinheit, welche den Artikel geposted hat (Sender).

Für die Auswertung hat dies folgende mögliche Implikation: Zum einen ist die Konsistenz und Authentizität eines Autors (From) nicht verifiziert. Zum anderen muss der verifizierte Sender nicht mit einem Autor der Nachricht, die der Sender sendet, übereinstimmen. Zwar wird diese Situation statistisch eher selten zu beobachten sein, aber gerade wenn es um außergewöhnliche Beobachtungen geht, kompromittiert dies ggf. die Möglichkeiten einer transparenten Erkennung.

Follow-Up

Diese Zeile hat dasselbe Format wie die obligatorische Newsgroups-Zeile. Ist die „Follow-Up“ Zeile vorhanden, werden Antworten auf die Nachricht direkt in die in dieser Zeile angegebenen Newsgruppen gepostet. Ansonsten werden sie in die Newsgruppen gepostet, welche durch die Zeile „Newsgroups“ spezifiziert sind.
Dies erlaubt einen Mechanismus, initial z.B. Crosspostings vorzunehmen und für Rückmeldungen eine Einschränkung auf einzelne Newsgruppen oder eine einzige Newsgruppe vorzunehmen. Zudem wird in manchen (ggf. moderierten) Newsgruppen mit Multimedia-Inhalten eine Follow-Up Diskussion in einer speziellen Diskussionsgruppe vorgenommen, sodass Rückmeldungen von geposteten Inhalten getrennt sind, was die Handhabung z.B. der Downloads erleichtert.

Expires

Das „Expires“ Feld schlägt ein Verfallsdatum für eine Nachricht vor. Das Feld sollte allerdings nur sehr selten, d.h. mit gutem Grund genutzt werden, da in der Regel lokale Expire-Policies der Systeme im USENET System den Verfall von Artikeln anhand von Resourcennutzung u.ä. vorgeben. Es ist nicht gewährleistet, dass ein solches Expire Datum berücksichtigt wird. Lokale Newssoftware sollte nicht automatisch ein Verfallsdatum vorgeben, wenn eine Nachricht gesendet wird.
Allerdings weist die Verwendung von einem Expire Datum darauf hin, dass eine Nachricht ggf. nur sehr kurz oder auch sehr lang von Wert sein kann. Somit ist es ein (nicht gesichertes) Attribut, welches auf einen außerordentlichen Inhalt verweisen könnte.

References

In diesem Feld werden die Message-IDs für alle Nachrichten gelistet, welche diese Nachricht ausgelöst haben. Das Feld ist für ein Follow-Up gefordert und verboten, wenn ein neues Subject erstellt wird.
Implementationen unterstützen in der Regel ein Follow-Up Kommando, welches eine Nachricht mit der gleichen Subject-Zeile unterstützt, ihr jedoch die Zeichenfolge „Re: “ voranstellt, sofern diese in der Subject-Zeile noch nicht enthalten ist.
Enthält die Originalnachricht keine References Zeile, wird in der References-Zeile der Follow-up Nachricht die Message-ID der Originalnachricht (in „<“, „>“) eingefügt. Ansonsten wird der Inhalt der References-Zeile der Originalnachricht übernommen und die Message-ID des Originals mit Leerzeichen angehängt. Dabei wird die References-Zeile ggf. am Anfang beschnitten, sodass eine vernünftige Zahl von Rückwärts-Referenzen verfügbar ist, ohne das die Zeile zu lang wird.
Der Zweck der References Zeile dient der Gruppierung von Nachrichten in Konversationen durch entsprechende Schnittstellenprogramme (z.B. Newsreader). Zwar ist die Nutzung dieser Zeile nicht vorgeschrieben, jedoch sollten alle Programme für die automatische Erzeugung dieses Feature verwenden.
Für die Erkennung von Outliern ist dieses Merkmal ggf. nutzbar für einen Check des Clusterings von Newsartikeln, sodass mglw. Informationen darüber genutzt werden können, welche Anwender auf welche Nachricht mit hoher Wahrscheinlichkeit geantwortet und daher auf den Inhalt der Originalnachricht (bzw. innerhalb einer Konversation = Gruppe von Nachrichten) Bezug genommen haben.

Control

Nachrichten mit einer Kontrollzeile werden für die Kommunikation zwischen Newssystemen mit Kontrollnachrichten verwendet. Diese Nachrichten sind nicht für den Anwender bestimmt und sollen von ihm nicht gelesen werden. Manchmal wird zusätzlich die Subject-Zeile um den Vorsatz „cmsg“ erweitert, sodass der Rest der Subject-Zeile als Kontrollnachricht interpretiert werden kann.

Distribution

Diese Zeile wird zur Eingrenzung der Distribution von Newsgruppen verwendet. Somit müssen Systeme, welche eine Nachricht mit Inhalten in der Distribution-Zeile empfangen wollen, sowohl die Newsgruppen in der Newsgroups-Zeile empfangen, als auch die vorgeschlagene Distribution. Dieser Mechanismus dient dazu, bei Antworten den Kreis der erreichbaren Abonnenten zu erweitern oder einzugrenzen. Da es sich bei Distributionen um Verteilungsgebiete im Rahmen der Newssystem-Topologie handelt,

	können dadurch lokale und globalen Distributionen (im Sinne der Newsgruppenhierarchie) genauso realisiert werden, wie geografische Eingrenzungen.
Organization	Dieses Feld dient der kurzen, prägnanten Beschreibung der Organisation des Senders bzw. des sendenden Systems. Da die Namen der Systeme selbst oft kryptisch sind, soll so die Identifikation des Senders bzw. der Senderorganisation erleichtert werden.
Keywords	In dieser Zeile sollten einige sorgfältig gewählte Schlüsselwörter die Nachricht identifizieren bzw. beschreiben, um dem Nutzer eine Hilfestellung für seine Entscheidung zu geben, ob die Nachricht für ihn interessant ist. Hier ist zu erheben, ob dieses Feld tatsächlich in hohem Umfang genutzt wird. Ist dies der Fall, lassen sich ggf. Rückschlüsse auf Themengruppierungen ziehen bzw. auf Nachrichten, welche sich keinem Schlüsselwort zuordnen lassen.
Summary	Diese Zeile soll eine kurze Zusammenfassung der Nachricht enthalten. I.d.R. soll dies die Follow-Up Mechanismen unterstützten. Hier ist in der Testmenge zu erheben, ob dieses Feld tatsächlich in hohem Umfang genutzt wird. Ist dies der Fall, lassen sich ggf. Rückschlüsse auf Themengruppierungen ziehen bzw. auf Nachrichten, welche sich keinem Schlüsselwort zuordnen lassen.
Approved	Diese Zeile ist notwendig, sobald eine Nachricht zu einer moderierten Newsgruppe gepostet wird. Sie wird vom Moderator eingesetzt und sollte im Inhalt seiner Mailadresse entsprechen. Dieses Feld ist auch für Kontrollnachrichten obligatorisch.
Lines	Diese Zeile enthält die Anzahl der Zeilen des Body (Textkörper) einer Nachricht.
Xref	Dieses Feld enthält Informationen über Newsgruppennamen und Nachrichtennummern (z.B. die wievielte Nachricht die vorliegende in der Newsgruppe (aus dem Newsgroups-Feld) auf dem lokalen System ist. Diese Zeilen werden im Rahmen des globalen Newsaustausches nicht übertragen und haben i.d.R. nur für lokale Interfaces (z.B. Newsreader) einen Nutzen.

Alle beschriebenen Mechanismen, welche sich durch die Anwendung der verschiedenen Felder etablieren lassen, werden i.d.R. noch durch weitere, gruppenspezifische Mechanismen ergänzt. Diese sind meist in den FAQs (Frequently Asked Questions) oder in Handlungsanweisungen (Gruppen-Netiquette, Do's & Don'ts) für die spezielle Gruppe festgeschrieben. Alternativ haben sie sich durch kontinuierliche Nutzung etabliert. Solche Mechanismen umfassen die Handhabung von Diskussionen, die Art des „Quotings" (siehe auch Kapitel 4.2.3), die Anforderung von speziellen Inhalten, die Frequenz spezieller Postings oder die Kennzeichnung von Themen und Inhalten zur automatischen Erkennung und Filterung.

4.2. Nutzungsmechanismen von Newsgruppen

4.2.1. Mechanismen bezogen auf Newsgruppen

Interessant sind im Zusammenhang mit der späteren Anwendung von Outlier-Erkennungsverfahren auf Newsgruppen vor allem die Mechanismen, welche sich bei der Nutzung von News in den unterschiedlichen Gruppen etabliert haben. Hierbei sind vor allem folgende Bereiche grundlegend zu unterscheiden:

- reiner Austausch von Multimedia-Inhalten
- Austausch von Multimedia-Inhalten inkl. Diskussion dieser Inhalte
- Austausch von Inhalten und Diskussion über diese getrennt in Austausch- und Diskussionsgruppen
- moderierte Newsgruppen
- unmoderierte Newsgruppen und moderierte Newsgruppe zum gleichen Thema
- reine Diskussionsgruppen (unmoderiert)
    - Diskussionsgruppen mit freien Themen (allgemein)
    - Diskussionsgruppen mit spezifischen Themen
    - Diskussionsgruppen mit freien Teilnehmern
    - Diskussionsgruppen mit spezifischen Teilnehmern

In Gruppen, welche dem reinen Austausch von Multimediainhalten vorbehalten sind, werden i.d.R. die Inhalte als Dateien am Stück oder in Teilen mittels spezieller Software codiert (um z.B. eine reibungslose 7-bit Übertragung zu gewährleisten). Sodann werden diese codierten Elemente in vom USENET System handhabbare Teile zerschnitten, welche in separate Nachrichten im Textkörper eingebettet werden. In der Subject-Zeile werden dann die Zieldateinamen sowie die Nummerierung der einzelnen Teilstücke chronologisch referenziert. Meist wird das Inhaltspaket insgesamt geposted, damit Anwender dieses innerhalb der für Multimediagruppen meist sehr kurzen Expiration-Zeit abrufen können. Oft wird zudem entweder ein Checksummen- oder Reparaturpaket als zusätzlicher Inhalt übermittelt. Ansonsten erstreckt sich die Kommunikation in solchen Gruppen auf die Anforderung fehlender Teile, die generelle Suche nach bestimmten Inhalten bzw. Kommentare zur Qualität der übermittelten Daten. Eine ausführliche thematische Diskussion findet meist nicht statt.

Es gibt eine Reihe von Multimediagruppen, in welchen eine thematische Diskussion zu den publizierten multimedialen Inhalten in einer gleichnamigen Diskussionsgruppe besprochen wird. Hier findet ein themenbezogener Austausch statt, allerdings orientiert sich dieser primär an dem in der eigentlichen Multimediagruppe gepostetem Inhalt, d.h. die Querbezüge und Referenzen spielen zum einen eine sehr große Rolle, zum anderen müsste der Multimediainhalt selbst in die Betrachtung einbezogen werden, um das Diskussionsverhalten zu beleuchten.

Da sich gestückelte Multimediainhalte generell nicht direkt im USENET System zusammensetzen und betrachten lassen, ist eine Outliererkennung in diesem Fall nur von geringem Wert, da eine Datenbasis innerhalb des Systems, welche den Inhalt uncodiert widerspiegelt, nicht vorhanden ist. Eventuell wäre eine teilweise (An)-Dekodierung von Inhalten oder Inhaltsteilstücken sinnvoll, um innerhalb der kodierten Inhalte Rückschlüsse auf die thematische Zugehörigkeit zur Gruppe zu gewinnen. Allerdings ergibt sich für die generelle Outliererkennung in Newsgruppen dadurch kein signifikanter Erkenntnisgewinn, der über den der Outlieridentifikation in Textgruppen hinausgeht. Daher steht der ungleich komplexere Aufwand dem Ziel dieser Arbeit nicht gleichberechtigt gegenüber. Andererseits gibt es eine ganze Reihe von wissenschaftlichen Arbeiten zur Inhaltsanalyse und Bewertung von multimedialen Inhalten jedweder Art, sodass auch hier ein relevanter Beitrag nur schwer abschätzbar ist. Aus diesem Grund werden Gruppen mit primär multimedialen Inhalten (Bilder, Audiodateien, Videodateien oder Programme und Programmpakete) im Rahmen dieser Arbeit nicht für die Erkennung von Outliern herangezogen.

Da bei textuellen Newsgruppen generell zwischen moderierten und unmoderierten Newsgruppen unterschieden werden kann[10], erfolgt hiermit eine getrennte Beschreibung der Mechanismen. Der maßgebliche Unterschied besteht in der Art des Postings von neuen Nachrichten in die Gruppe. In unmoderierten Gruppen werden die Artikel bzw. Nachrichten direkt in die Gruppe gepostet und sind für alle Diskussionsteilnehmer unmittelbar sichtbar. Bei moderierten Newsgruppen werden neue Nachrichten zuerst per eMail an den Moderator der Gruppe gesendet. Dieser entscheidet dann, ob die Nachricht zu der Gruppe thematisch gehört und ob sie allen festgelegten zusätzlichen Anforderungen, welche für die Diskussion in dieser Gruppe von den Nutzern festgeschrieben wurden – und über die der Moderator letztendlich zu wachen hat – entspricht, und ob sie damit in die Newsgruppe geposted werden kann. Dieser Vorgang des Postens wird dann vom Moderator übernommen.

Aus diesem Grund ist bei moderierten Newsgruppen nur sehr eingeschränkt mit Outliern zu rechnen. Diese ergeben sich zum einen dann, wenn z.B. einzelne Artikel zwar in den generellen, von der Gruppe vorgegebenen Themenblock gehören, aber innerhalb dieses Blocks stark voneinander abweichen. Aus praktischer Sicht von hervorgehobenem Interesse sind mit Sicherheit die Eingangsdaten für moderierte Newsgruppen, welche unmoderierten Gruppen entsprechen. Auch hier ist mit einer eingeschränkten Menge an Outliern zu rechnen, weil die Sender von Nachrichten in eine moderierte Gruppe im Allgemeinen mehr Sorgfalt walten lassen und SPAM Versender innerhalb der News nur eingeschränkt versuchen, in moderierte Gruppen zu spammen, weil die Erfolgsaussichten nahe Null sind und SPAM sich vor allem an den Leser richtet und nicht Mittel zur Geißelung eines Moderators ist, da SPAM seit geraumer Zeit vor allem kommerziellen Zielen dient und damit auch kommerziellen Gesetzmäßigkeiten unterliegt. Daher werden auch moderierte Gruppen in dieser Arbeit nicht untersucht.

Es gibt stellenweise Newsgruppen, welche zum einen gleichartige Themen behandeln, und von denen eine Gruppe moderiert und eine andere unmoderiert ist. Zum anderen gibt es seltene Fälle, in denen zu einer Newsgruppe jeweils eine moderierte und eine unmoderierte Ausprägung gehört. Da aber nicht davon auszugehen ist, dass die unmoderierte Gruppe den Status der moderierten vor der Moderation zeigt, ist eine Differenzanalyse nicht mit Sicherheit erfolgreich und daher auch im Bezug auf die Erkennung von Outliern nicht unbedingt aussagekräftig.

[10] Diese Unterscheidung ist nicht auf textuelle Gruppen beschränkt, auch Gruppen mit Multimediainhalten werden moderiert und unmoderiert betrieben.

Zur Untersuchung von Outlierverfahren in Newsgruppen sind die unmoderierten Newsgruppen besonders interessant, weil sie zum einen rein pragmatisch für die Erkennung von nicht zur Gesamtheit zugehörigen Objekten (d.h. Nachrichten) den größten Nutzen durch darauf aufbauenden Wissensgewinn für Analyse, Filtering, Sortierung, etc. versprechen, und weil zum anderen durch die Vielschichtigkeit des Anwenderverhaltens in diesen Gruppen die Outlier auch mit einiger Wahrscheinlichkeit einen prägnanten Charakter haben.

Bei den unmoderierten Gruppen ist unter den Unterscheidungskriterien Themen und Teilnehmer das folgende verschiedene Verhalten innerhalb der Gruppe zu beobachten, wobei sich diese Annahmen nicht formell verallgemeinern lassen, sondern lediglich auf direkter empirischer Beobachtung im Rahmen einer Stichprobenauswertung basieren. Gruppen mit einem Themenkomplex, der viele Unter- bzw. Teilthemen zulässt, enthalten mit einiger Wahrscheinlichkeit stark fragmentierte Diskussionen. Diese umfassen z.B. wenige Artikel bzw. Nachrichten innerhalb eines Diskussionsthreads. Oder einzelne Diskussionen liegen thematisch sehr weit auseinander, lassen sich aber dennoch klar dem Thema der Gruppe selbst zuordnen. Andere Gruppen mit einem sehr spezifischen Oberthema enthalten normalerweise auch nur eine sehr eingeschränkte Reihe von Diskussionen rund um dieses Thema mit einem schmalen Band an Unterthemen. Verstärkt wird diese Unterscheidung meist dadurch, dass in sehr spezifischen Gruppen auch nur ein stark eingrenzbarer Teilnehmerkreis mit tiefem Wissen um das spezifische Thema selbst aktiv an der Diskussion teilnimmt. Bei thematisch breit angelegten Gruppen ist in vielen Fällen auch der Kreis der Diskussionsteilnehmer sehr breit angelegt und damit auch das Niveau von Diskussionen sehr vielfältig. Wie aber bereits ausgeführt, ist dies nicht zwingend zu verallgemeinern: In einer spezifischen Gruppe kann durchaus ein dem normalen Diskussionsteilnehmerkreis nicht zugehöriger Teilnehmer eine Frage auf einem Niveau stellen, welches nicht der Menge der normalen Diskussionen entspricht. Sehr wohl kann eine solche Frage aber zum einen dem Thema entsprechen und damit in die Gruppe gehören bzw. auch durch die Antwort eines anderen Teilnehmers eine der Gruppe zuzuordnende Diskussion auslösen.

Im Hinblick auf die ursprünglich aufgeworfene Frage, warum Outlier als solche identifiziert werden (vgl. auch [55]) wird deutlich, dass sich der Outlier Charakter in Bezug auf die jeweils zur Verfügung stehende Gesamtmenge an Objekten ergibt und damit im Fall der USENET News sehr situationsabhängig sein kann.

4.2.2. Mechanismen bezogen auf den Artikelfluss

Einen besonderen Aspekt bieten die USENET News zusätzlich zur Vielfalt der Themen und Nutzerkreise durch den hohen Grad an Dynamik beim Durchfluss der Artikel durch das System. Abhängig von den jeweils für die Gruppen global gesetzten, für einzelne Artikel vom Anwender empfohlenen und letztendlich von den lokalen Systemen durchgesetzten Vorhaltezeiten der Artikel sieht der Anwender für einen gewissen Zeitraum einen temporären Ausschnitt aus der Gesamtbeobachtungsmenge. Dies ist in Abbildung 33 dargestellt.

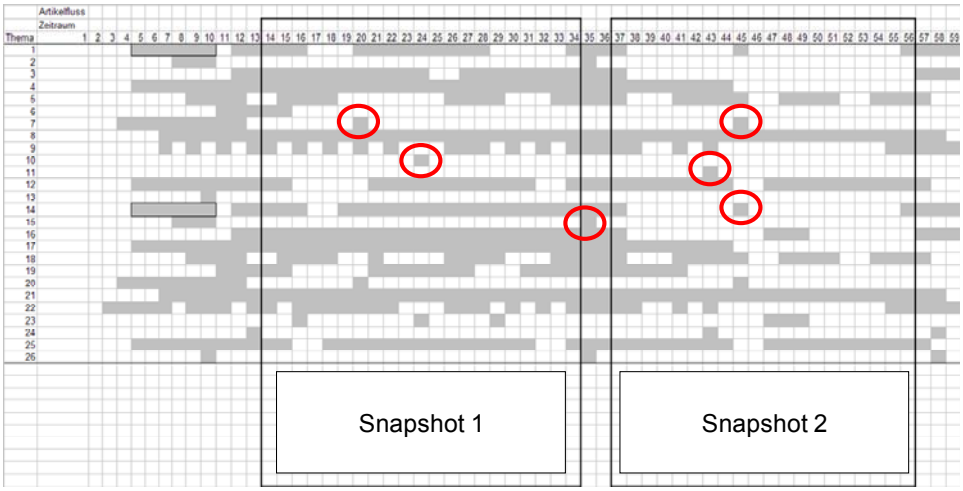


Abbildung 33 - Exemplarischer Artikelfluss als Snapshot einer Newsgruppe

Abbildungsbeschreibung: Durch den Expire-Mechanismus ist zeitlich jeweils nur ein Ausschnitt der durch das System fließenden Artikel sichtbar und somit als Datenmenge effektiv verfügbar. Die Nutzungsmechanismen reichen aber über diesen Ausschnitt hinaus.

Die Abbildung zeigt schematisch grob einen Überblick über Artikel in einer Newsgruppe, welche thematisch zueinandergehören, also z.B. bei entsprechender Clusterbildung durch ein gewähltes Clusteringverfahren mit einiger Wahrscheinlichkeit innerhalb eines Attributraumes ein und demselben Cluster zugeordnet würden. Diese angenommenen Cluster wurden in der Tabelle entsprechend nummeriert und Elemente pro Zeitraum, die diesem Cluster zugeordnet würden, jeweils auf einer Zeitachse gemäß ihrer Präsenz im USENET Newssystem (durch Vorhaltung auf dem für den relevanten Nutzer maßgeblichen lokalen System) angeordnet.

Hierbei zeigen Snapshot 1 und Snapshot 2 jeweils zwei Zeiträume an, welche durch die maximale Vorhaltezeit gekennzeichnet sind. Da Newsartikel gemäß ihres Eintrittszeitraumes in das System entsprechend nach Ablauf der Vorhaltezeit individuell ausgesondert, bzw. im Fachbegriff „expired“ werden, wäre das Snapshot-Fenster natürlich entsprechend pro Themenzeile verschoben. Zur vereinfachten Betrachtung sei angenommen, die Themen seien zeitlich auf den Vorhaltezeitraum der Newsgruppe hin bereits normiert worden. Nach dieser Normierung ist nun deutlich zu sehen, dass in der Historisierung wesentlich mehr Nachrichten als Objekte einem Themencluster zugeordnet werden könnten, würde eine maximale Vorhaltezeit nicht bestehen. Da aber nur ein zeitlicher Ausschnitt des Systems als Testmenge zur Verfügung steht, kann eine Outliererkennung nur Outlier innerhalb der Testmenge, d.h. innerhalb der Snapshot-Zeit identifizieren. Alternativ könnte eine Outlier Identifikation über die dazu vorgehaltenen historisierten Daten der Attribute von Objekten, die im System selbst bereits expired sind, auch auf Zeiträume außerhalb des Snapshot erweitert werden. Hierbei ist davon auszugehen, dass ggf. verschiedene Outlier jeweils bei verschiedener Testmengenbetrachtung erkannt werden. Dies unterstreicht eindrücklich den Bezugscharakter von Outliern zu der Menge der beobachteten Gesamtobjekte.

Abbildung 33 verdeutlicht dies noch einmal durch die Kennzeichnung von potentiellen Outliern durch Einkreisung. Hier ist zu erkennen, dass gewisse Outlierkandidaten innerhalb eines Snapshot theoretisch bei Erweiterung der Datenmenge diesen Charakter nicht mehr in derselben Art und Weise hätten. Signifikant wird dieses Problem durch die teilweise geringe Vorhaltezeit in Newsgruppen verglichen mit der Persistenz gewisser diskutierter Themen. Dies äußert sich dadurch, dass Diskussionsthreads (gebildet durch die References Felder mit den der Diskussion zugeordneten Message-ID) jeweils weit über einen Snapshot hinausgehen können.

> *Anmerkung:* Der Begriff des Snapshot ist hier willkürlich zur besseren Beschreibung der Auswirkungen des Sachverhalts gewählt worden. Snapshots sowie Threads sind keine standardisierten Vorgänge innerhalb der News, sie ergeben sich allein auf Basis der freien Nutzung und „eingefahrener“ Vorgänge und Abläufe der Anwender. Da es sich bei der Outliererkennung in Newsgruppen im Rahmen dieser Arbeit allerdings um die empirische Untersuchung praktischer Anwendungen handelt, welche das Problem nicht formell vollständig beschreiben sollen, geben diese eingeführten Begriffe eine gute Hilfestellung zur Erreichung dieses Ziels.

4.2.3. Mechanismen bezogen auf Newsartikel

Auch der von Nutzungsmechanismen gekennzeichnete Aufbau des Textkörpers von Artikeln einer Newsgruppe muss betrachtet werden, um die Voraussetzungen für die Anwendung von Outlier-Detection auf USENET News zu schaffen. Folgende Bestandteile bzw. Aspekte kommen vor:

- Bilderinhalte in Newsgruppen
- Hyperlinks in Newsgruppen
- Zitate in Newsgruppen
    - Senderverweise
    - einfache Zitierungen (quotes)
    - mehrfach historisierte Zitierungen (multiple quotes)
    - Blockzitate
    - Splitterzitate
- Unterschriften und Signaturen
    - einfache Unterschriften
    - statische Signaturen
    - dynamische Signaturen
    - Werbesignaturen
    - Signature-quotes

Auf Newsgruppen mit multimedialen Inhalten wurde schon hingewiesen. Da es innerhalb der News nur durch die Nutzung (und in einigen Fällen die thematische Eingrenzung über die Gruppenhierarchie) bestimmt wird, ob in einer Gruppe primär textuelle oder auch multimediale Inhalte ausgetauscht und diskutiert werden, besteht keine harte Unterscheidung. Es kann also durchaus vorkommen, dass in einer i.d.R. rein textuellen

Newsgruppe ein Bild in eine Nachricht eingebettet ist. Dies kann dem Ziel der besseren Verständlichkeit genauso dienen, wie einer bloßen Anreicherung der Nachricht durch Standardinhalte. Beispiel für Letzteres sind die in der Internet Kommunikation weit verbreiteten „Smilies“ zur Unterstützung der Stimmungsäußerung in Nachrichten. Während sich in eMails und News meist die textuellen Umsetzungen solcher Smilies durchgesetzt hat (:-), ;-), :-(, 8=), usw.), wird in Chat-Foren, Blogs und beim Austausch von Instant Messages ein Smiley meist grafisch übertragen (im Zeichensatz wie hier: ☺, ☹; oder als direkte Grafik wie in Abbildung 34) und in gewissen Fällen noch animiert. Inwieweit sich diese Entwicklung auch auf die Newsgruppen bereits übertragen hat oder noch überträgt, kann vom Autor nicht beurteilt werden.

Abbildung 34 - Grafische Textelemente in Nachrichten

In den Antworten kann ggf. auf den Bildinhalt oder das Bild selbst Bezug genommen werden, ohne dass dieses in der Antwort erneut eingebettet ist. Bei der Übertragung ist das Bild sodann zusätzlich codiert, sodass erst eine Decodierung und Wiedereinbettung notwendig ist, um eine Beurteilung der Nachricht in der Text-Bild Kombination zu gewährleisten. Natürlich kann auch davon ausgegangen werden, dass sich Nachrichten mit und ohne Bilder dann inhaltlich signifikant unterscheiden, wenn der Großteil der Nachrichten in einer Gruppe keine Bilder enthält.

Ähnlich sind in den Nachrichten enthaltene Hyperlinks zu betrachten. Zwar haben diese auch eine rein textuelle Entsprechung, allerdings ergibt sich für den Leser einer Nachricht durch die in der Newsreader-Anwendung durchgeführte Verkettung eines Hyperlinks mit einem Browser durch einen Klick auf den Link die Möglichkeit, den Inhalt der Nachricht wesentlich zu erweitern. Somit folgt die Fragestellung, ob der Inhalt einer Nachricht mit oder ohne den durch den Hyperlink indirekt hinterlegten Zusatzinhalt mit den anderen beobachteten Objekten der Gesamtmenge verglichen werden soll, oder ob sich durch die Tatsache, dass ein Hyperlink vorhanden ist, eine starke inhaltliche Unterscheidungsqualität ergibt.

Ein ganz wesentlicher Aspekt in den Diskussionsthreads sind vom Sender einer Antwortnachricht in dieser Antwort belassene Teile der Originalnachricht. Diese Zitate bzw. sogenannten „Quotes“ sind im Internet allgemein stark verbreitet (z.B. bei eMail), stellen aber für News Anwender ein Kernelement des Mediums dar. Oft ersetzen diese Quotes die eigentlich durch das Summary-Feld beabsichtigte Zusammenfassung und verketten Inhalte sehr stark durch direkten Detailbezug weit über die normale Verkettung im Rahmen der „References“ und gleicher „Subject“-Zeilen hinaus. Dabei werden Quotes durch sog. Quotierungen gekennzeichnet. Diese bestehen meist aus einer Einleitungszeile mit einer direkten Referenz auf den Ursprungsautor und einem oder mehreren Einrückzeichen, wie im folgenden Beispiel gezeigt:

```
User sysadmin wrote on Mon 12 Oct 2005:

> Wenn dies nicht hilft, am besten das System
> neu starten und dann die Option –s beim Pro-
> grammstart verwenden.

Dies habe ich versucht, allerdings hat es
nicht geholfen, das Problem besteht weiterhin.
```

Allerdings ist weder die Einleitung der Quotierung noch die Nutzung eines bestimmten Quotierungszeichens als Standard vorgeschrieben, sodass es eine ganze Reihe von Quotierungen geben kann. Oft wird z.B. noch ein Zusatz vor das Quote-Zeichen gestellt.

```
User sysadmin wrote on Mon 12 Oct 2005:

sysadmin > Wenn dies nicht hilft, am besten das System
sysadmin > neu starten und dann die Option –s beim Pro-
sysadmin > grammstart verwenden.

Dies habe ich versucht, allerdings hat es
nicht geholfen, das Problem besteht weiterhin.
```

Im Rahmen der Diskussion verästelt sich diese immer weiter, je mehr Nutzer auf eine Nachricht innerhalb des Thread-Baumes Bezug nehmen und dadurch einen neuen Nachrichtenknoten im Thread unter der Ursprungsnachricht als Wurzel schaffen. Innerhalb dieser Antworten auf Antworten werden manchmal verschachtelte Quotierungen vorgenommen.

```
User stephan wrote on Tue 13 Oct 2005:

>> Wenn dies nicht hilft, am besten das System
>> neu starten und dann die Option –s beim Pro-
>> grammstart verwenden.
>
> Dies habe ich versucht, allerdings hat es
> nicht geholfen, das Problem besteht weiterhin.

Du musst ja auch die Option –s nehmen, hast Du das
getan? Sonst funktioniert es natürlich nicht.
Mit freundlichen Grüßen
Sysadmin
```

Dabei kann eine Quotierung sowohl am Block vorgenommen werden (wie es z.B. bei eMails oft der Fall ist) und es ist dem Antwortenden überlassen, oberhalb des Block-Quotes oder unterhalb oder mittendrin Antworten einzufügen. In diesem Zusammenhang werden oft Teile aus den Blöcken gelöscht. Manchmal wird dies mit […] gekennzeichnet. Auch hier gibt es keine Vorschriften. Zudem gibt es keine Pflicht zur Quotierung, ein thematischer Austausch kann also auch so aussehen, wie im Folgenden dargestellt.

Ursprungsnachricht:

```
Subject: Suche Unterlagen zur KI

Hallo KI-Gemeinde,

mich interessieren Verfahren zum maschinellen Lernen,
ich bin mir aber nicht sicher, wo ich suchen muss, um
auch wirklich gute wissenschaftliche Unterlagen zu
finden.

Vielen Dank im Voraus,
Suchender KI’ler
```

Antwort:

```
Subject: Re: Suche Unterlagen zur KI

Hallo Interessent,

solche findest Du z.B. am LS8 der Uni Dortmund,
Fr. Prof. Morik, http://ls8.informatik.uni-dortmund.de

Viel Spass,
WiMi
```

Rein thematisch gehören diese Nachrichten stark zueinander. Wird jedoch die textuelle Gemeinsamkeit betrachtet, bezieht sich diese vor allem auf die Subject-Zeile und beschränkt sich in der Analyse auf die ähnlich verwendeten Grußformeln.

Ein weiterer Level an Komplexität wird durch den möglichen Querbezug auf dritte Nachrichten innerhalb eines Threads eingeführt. Auch dies ist nicht ungewöhnlich, sofern ein Anwender zwei getrennte Diskussionsfäden an einer Stelle wieder zusammenführen will. Es wird in diesem Fall ein Querverweis auf eine Nachricht in einem anderen Teilbaum des Threads vorgenommen. Dieser Querverweis ist in der Regel nicht explizit, sondern wird im Text durch das Übernehmen eines Quote-Splitters aus der anderen Nachricht realisiert.

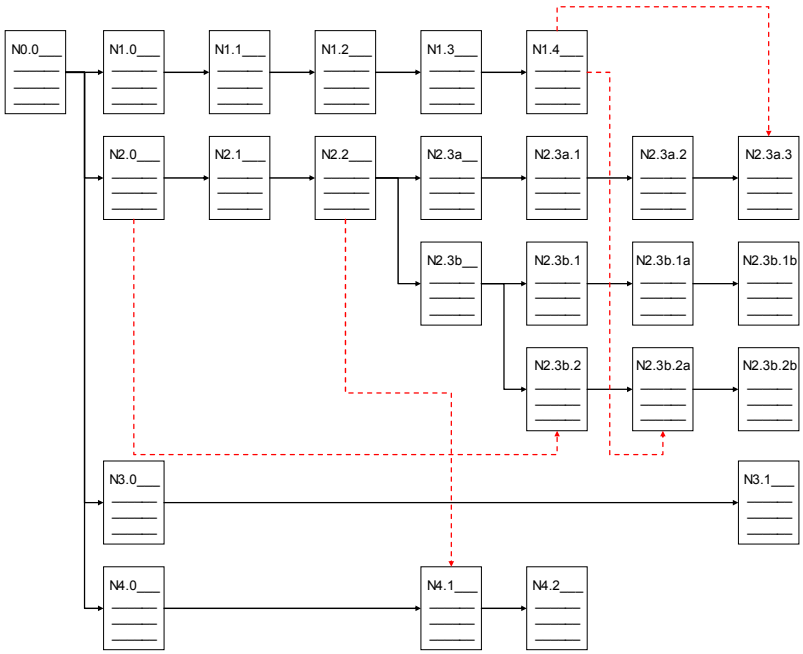


Abbildung 35 - Textuelle Querverweise in Diskussionsthreads

Abbildungsbeschreibung: Querverweise zwischen Artikeln werden von Nutzern bezogen auf deren konkreten Kontext auch zwischen solchen Nachrichten verwendet, die nicht dem gleichen Thread bzw. Subthread zuzuordnen sind.

Abbildung 35 verdeutlicht dies an einem Beispiel. Hier wird ein Diskussionsthread in einem Baum dargestellt, dessen Knoten die Nachrichten zeigen und deren Kanten durch die Beziehung einer direkten Follow-Up Nachricht hergestellt werden. Die Wurzelnachricht N0.0 wird durch vier Follow-Up Nachrichten (N1.0, N2.0, N3.0 und N4.0) beantwortet. Diese Nachrichten müssen im Übrigen nicht von vier verschiedenen Nutzern versendet werden. Da der Follow-Up Bezug sich auf die Verknüpfung von Message-IDs bezieht und diese in den References als kompletter Pfad (oder als beschnittener Teilpfad zum Ende der letzten Antwort hin) gespeichert werden, wird die From- bzw. Sender-Zeile nicht für diesen Aufbau ausgewertet. Denn obwohl die Domain des sendenden Systems Bestandteil der Message-ID ist, kann der Sender und der Autor durchaus verschieden sein. Somit reflektiert die Message-ID keine Beziehung zwischen den Nutzern, sondern nur zwischen den Nachrichten selbst in einem Follow-Up bzw. Antwortgefüge.

In der Abbildung sind weiterhin vier verschiedene indirekte Bezüge ohne ein direktes Follow-Up dargestellt. So nimmt z.B. der Inhalt von Nachricht N4.1 auf den Inhalt der Nachricht N2.2 Bezug, ohne dass es eine Follow-Up Beziehung gäbe. Da die Anwender davon ausgehen, dass ein Interessent und Teilnehmer an der Diskussion den gesamten Thread liest, lassen sich Querbezüge auch außerhalb der Follow-Up Pfade erreichen. Zudem ist nicht gesichert, dass der Newsreader Threads auch unterstützt, also z.B. die References-Zeile entsprechend für eine Thread-Ansicht der Artikel auswertet. Oder der Anwender entscheidet sich, die Sortierung der Nachrichten nach Subject und sekundär nach dem Versendedatum durchzuführen. Dem Newssystem selbst bleiben solche Querbeziehungen jedoch aufgrund der fehlenden Protokollunterstützung verborgen.

Es ist sehr selten, dass ein Follow-Up auf eine Nachricht durchgeführt wird und dann ausschließlich ein Querbezug zu einer ganz anderen Nachricht erfolgt. Durch einen Benutzerfehler ist aber auch diese Situation nicht auszuschließen. Der inhaltliche Bezug selbst kann durch ein Querbezugs-Quoting hergestellt werden.

```
User stephan wrote on Tue 13 Oct 2005:

>> Wenn dies nicht hilft, am besten das System
>> neu starten und dann die Option –s beim Pro-
>> grammstart verwenden.
>
> Dies habe ich versucht, allerdings hat es
> nicht geholfen, das Problem besteht weiterhin.

Du musst ja auch die Option –s nehmen, hast Du das
```

```
getan? Sonst funktioniert es natürlich nicht. Wenn es
allerdings daran nicht liegt, dann versuche doch mal
den Tip von netadmin, der schrieb neulich:


> […] manchmal hilft es auch, den Prozess im Task-Manager
> zu beobachten. Sobald er 100% CPU Last erzeugt, sollten
> alle Plug-Ins des Programms deinstalliert werden.
> Grüße – netadmin

Mit freundlichen Grüßen
Sysadmin
```

Hier wurde mit Absicht ein Beispiel gewählt, in dem der gleiche Quote-Mechanismus verwendet wird, wie für den ursprünglichen Quote (inkl. des verschachtelten Quotes). Somit sind solche Querbezüge nur schwer maschinell zu unterscheiden, obwohl dies dem Anwender offensichtlich nicht schwer fällt. Derartige Querbeziehungen können übrigens auch zwischen Artikeln bzw. Nachrichten aus ganz unterschiedlichen Threads bestehen, wenn z.B. auf einen FAQ der Gruppe oder eine andere Diskussion mit verwandten Themen hingewiesen wird. Sehr selten kann auch eine Querbeziehung zwischen zwei Artikeln ganz unterschiedlicher Gruppen bestehen. Diese Querverweise werden im Übrigen nicht nur indirekt über Quotes hergestellt, sondern können auch durch Cross-Postings der Follow-Up Nachricht in einen erweiterten Kreis von Gruppen oder erweiterte Distributionen gezielt oder unbeabsichtigt erreicht werden.

Letztendlich besteht auch im Bereich der News die Kultur, eigene Nachrichten zu signieren. Diese Form der Signierung erfolgt nicht im Sinne eines Fingerprints der Nachricht, sondern durch eine in den Text eingefügte sog. Signatur („Signature“). Entsprechend der Netiquette im Internet sollten derartige Signaturen eine gewisse Anzahl von Zeilen nicht überschreiten. Sie werden am Ende einer Nachricht unterhalb der abschließenden Grußformel eingefügt und enthalten manchmal die im eMail Verkehr üblichen Kontakt- und Adressdaten (inkl. Titel, Funktion, etc.). Im Newsbereich, in dem viele Anwender auch unter Pseudonym oder ganz anonym (bis auf die obligatorische eMail Adresse, welche z.B. auch mickeymouse@gmx.de sein kann) agieren, enthalten die Signaturen meist Verweise auf grundlegende Meinungen des Autors, Zitate der Weltliteratur, kluge Sprüche oder auch Witze. Dies lockert manchmal die Kommunikation entsprechend auf und ist wahrscheinlich der primäre Einsatzzweck.

Es gibt keine Vorschriften zum Aufbau einer Signatur, daher ist es auch schwierig, diese vom eigentlichen Teil der Nachricht zu separieren. Manchmal werden Teile der Signatur durch Signaturprogramme dynamisch mit alterierenden Inhalten erzeugt (z.B. Sprichwort des Tages, etc.). Es gibt auch eine Reihe von Werbesignaturen, welche sodann wieder multimediale Inhalte enthalten können, also z.B. Werbebanner. Besonders gute Signaturen veranlassen manchmal ein Follow-Up auf die Signatur selbst in einer Antwortnachricht oder zumindest einen indirekten Bezug durch ein Quoting und ein Gegensprichwort oder eine anerkennende Äußerung.

4.3. Vektorisierung von Texten für die Outliererkennung

Um die textuellen Inhalte der Newsartikel mit statistischen Outliererkennungsverfahren untersuchen zu können, ist es notwendig, jeden dieser Texte als Objekte in einem *m*-dimensionalen Suchraum darzustellen. Da die meisten Verfahren sich eine Verteilung der Objekte in diesem Raum bzw. auch die Positionierung der Objekte in Räumen zu nutze machen, um die Beziehungen zwischen diesen Objekten zu untersuchen, spielt die Entfernung zwischen den Objekten sowohl bei entfernungsbasierten, als auch bei dichtebasierten Outlier-Ansätzen eine große Rolle. Um diese Verfahren daher sinnvoll anwenden zu können, wird eine Umwandlung der Texte in vorzugsweise in einem metrischen Raum positionierbare Objekte vorgenommen. Dies geschieht durch Vektorisierung der einzelnen Texte anhand der Häufigkeit von Termen, welche in diesen Objekten vorkommen.

Der Wissenschaftszweig des Information Retreival [1] innerhalb der Informatik beschäftigt sich eingehend mit der Vektorisierung von Texten, sodass in dieser Arbeit lediglich auf die angewandten Mechanismen und eine Beschreibung in der Literatur [2] verwiesen werden soll. Für die praktischen Versuche kommt eine Vektorisierung der Newsartikel mit Hilfe des Word Vector Tools (WVtool) PlugIns [99] für die Lernumgebung YALE – Yet Another Learning Environment ([92] und [94]) des Lehrstuhls für Künstliche Intelligenz am Fachbereich Informatik der Universität Dortmund zum Einsatz.

Mit Hilfe dieses Tools wird jeder Text in einen Vektor umgewandelt. Dies geschieht in mehreren Schritten. Zuerst wird aus den Texten eine Wortliste erstellt. Diese Wortliste enthält alle Terme aller Texte für die Vektorisierung, die im zweiten Schritt erfolgt. Sie enthält daneben auch Statistiken, z.B. in wie vielen Texten ein Term vorkommt. Im Vektorisierungsschritt wird auf diese Statistiken Bezug genommen, um zu entscheiden,

welche Terme als Dimensionen im Vektorraum verwendet und wie sie gewichtet werden. Hierbei kommen die Betrachtung der Term Frequency (TF) und der Inverse Document Frequency (IDF) zum Einsatz [103]. Vorher werden jedoch grammatikalisch ähnliche Terme durch Reduktion und Mapping mittels Stemming (z.B. Porter Stemmer [102] oder Lovins Stemmer [101]) zusammengelegt.

Wie später in der Arbeit noch ausgeführt wird, erzeugt diese Vektorisierung in der Regel bei Texten hochdimensionale Räume. Zwar wird durch das Stemming die Zahl der Terme und damit die Zahl der Dimensionen stark reduziert, jedoch ist trotzdem mit einer Zahl von einigen hundert bis zu mehreren tausend Dimensionen als Ergebnis der Vektorisierung von Texten im Newsbereich zu rechnen. Diese Form der Darstellung von Texten in Vektorräumen als Projektion erlaubt aber trotz der dadurch erwachsenden Nachteile die Möglichkeit, sie wie ein einziges Objekt zu betrachten und damit durch die Anwendung von auf Objekte bezogenen Vergleichsverfahren, wie z.B. Clustering oder Outliererkennung, neue Erkenntnisse zu gewinnen.

Die Nachteile durch die hohe Zahl der Dimensionen und die im Vergleich relativ spärliche Besetzung der Gesamtbetrachtungsmenge an Newsartikeln (es handelt sich normalerweise um einige hundert bzw. einige tausend Texte innerhalb eines verfügbaren Snapshots in einer Newsgruppe, alle Texte sind jedoch relativ kurz und enthalten ggf. neben dem Header als Body nur wenig Informationen – eine Tatsache, welche dem Diskussionscharakter geschuldet ist) sind spärlich besetzte hochdimensionale Räume zu erwarten. Neben dem sofort augenscheinlichen Fakt, dass Verfahren für eine geringe Zahl an Dimensionen und Verfahren mit hohem Rechenaufwand ungeeignet erscheinen, birgt auch diese Spärlichkeit die in [68] erwähnten Unwägbarkeiten. Zum einen ist Textclustering, d.h. die Identifizierung von zueinander gehörenden Texten als vorgeschalteter Schritt für Outliererkennung möglicherweise nur sehr schwer mit guten Ergebnissen möglich. Zum anderen könnte aus demselben Grund auch die Aussagefähigkeit des Outliercharakters anhand verschiedener statistischer Maße nicht ausreichen, um wirklich intuitive und sinnvolle Outlier zu identifizieren.

Auf der anderen Seite erscheint eine zu starke Veränderung der Ursprungsmenge nicht sinnvoll, weil sich dadurch die Repräsentation der Anwendung selbst stark verfälschen könnte. Unter realen Bedingungen, d.h. auch auf komplexen Datenmengen, die für optimierte und vereinfachte Verfahren vielleicht nicht trivial zu erschließen sind, und insofern einen Spiegel der Umwelt darstellen, muss sich die Outliererkennung bewähren. Die Tatsache, dass die meisten Verfahren für eine hohe Zahl an Dimensionen nicht optimiert sind bzw. auch nicht optimierbar sein könnten (siehe unter anderem Verweise in [4] auf die Limitationen von Indizierungsverfahren zur Senkung des Rechenaufwandes bei einer hohen Zahl an Dimensionen), sollte nicht von einem Test auf den entsprechenden Rohdaten abhalten, um empirische Erfahrungen zu gewinnen.

Daher wird im Folgenden vor allem die Rohdatenmenge der Newsgruppen nach Vektorisierung betrachtet. Zum Vergleich wird eine Reihe von Ansätzen verfolgt, um die Zahl der Dimensionen zu reduzieren und damit die Anwendbarkeit niedrigdimensionaler Verfahren im Vergleich zu erleichtern.

- Die Ergebnisse der Vektorisierung werden optimiert, indem sehr häufig oder sehr selten vorkommende Begriffe beschnitten werden (Pruning)
- In den Newsartikeln selbst wird versuchsweise nur der Body-Teil erfasst und der Header nicht berücksichtigt (Splitting), wodurch sich die Vektorisierung auf den reinen Diskussionstext bezieht und beeinflussendes Zusatzwissen, welche die Positionierung der Objekte im Raum durch zusätzliche Dimensionen verändert, ausgeblendet wird.
- Als weitere Möglichkeit kommt die Ausblendung von Dimensionen nach der durchschnittlichen Gewichtung der Terme nach erfolgter Vektorisierung in Betracht.

Neben diesen Ansätzen kann es für weitergehende Betrachtungen des interessierten Lesers sinnvoll sein, verschiedene Newsartikel zusammenzufassen und nach Autoren oder Thread-Zugehörigkeit auszuwerten. Ein zusätzlicher Weg der Verbesserung der Vektorisierung ist der Einsatz von Hintergrundwissen mittels eines Thesaurus häufig vorkommender Terme (Wörter und Wortteile). Beide Wege werden jedoch im Rahmen dieser Arbeit nicht gesondert verfolgt, wobei neuere Implementierungen des WVTools diese unterstützen.

Um das Problem der spärlich besetzten Räume zu adressieren, wird auf folgende Methoden gesetzt:

- Reduktion der durch Vektorisierung entstandenen hochdimensionalen Räume durch Singular Value Decomposition [104] auf eine Zahl niedriger Dimensionen (ca. 2 – 5 Dimensionen).
- Reduktion der hohen Zahl an Dimensionen durch den Einsatz emergenter selbstorganisierender Netze (ESOM) mit abgeleiteter grafischer Darstellung dieser neuronalen Netze durch sog. U*Maps ([88] und [89]).

Gleichzeitig wird auch versucht, in hochdimensionalen Räumen mit klassischen Verfahren Ergebnisse zu erzielen, um eine Vergleichsbasis für die Ergebnisse der Ansätze zu erhalten und zu untersuchen, ob mittels solcher Reduktion veränderte Testmengen eine sinnvolle Alternative darstellen.

4.4. Mögliche Outlier-Kategorien für USENET News

Um in der praktischen Anwendung die erzielten Ergebnisse sinnvoll in Beziehung zu einer Erwartungshaltung zu setzen, welche eine empirische Betrachtung von Outliererkennung für News nutzbringend macht, bedarf es einer Vergleichsbasis von intuitiv zu erwartenden Outlierkategorien. Hierbei ergibt sich wiederum das schon beschriebene grundlegende Problem, dass Outlier nur sehr vage definiert sind [45]. Wenn davon auszugehen ist, dass Outlier solche Objekte sind, welche durch andersartiges Verhalten einen Verdacht dahingehend rechtfertigen, dass ihnen ein grundlegend anderer Mechanismus zugrunde liegt, so ist a priori nicht gleichzeitig definiert, wie sich diese Andersartigkeit äußert. Auch wird der Mechanismus, der solche Outlier-Kandidaten hervorbringt und welcher sich von den Mechanismen für die Objekte, welche in ihrer Gesamtheit als „normales Verhalten zeigend“ angesehen werden, grundlegend unterscheidet, durch die bloße Erkennung der Outlier selbst nicht erkannt. Somit ist auch nicht sichergestellt, welche Mechanismen dafür sorgen, dass es in einer Datenmenge Outlier gibt. Genausowenig ist bekannt, ob es sich dabei um einen oder mehrere Mechanismen handelt, d.h. ob Outlier dieselbe oder verschiedene Ursachen haben. Zusätzlich ist meist nicht bekannt, wie diese Mechanismen wirken, geschweige denn warum.

Insofern kann es ohne Betrachtung dieser zusätzlichen Aspekte keine umfassende und vor allem keine systematische Ordnung von Outliern oder Outlier-Erkennungsverfahren geben. Auch sind Aussagen zur Qualität von erkannten Outliern nur sehr schwer zu treffen. Zwar machen die meisten Autoren der in Kapitel 2 vorgestellten Verfahren vage Angaben darüber, inwieweit ihre Verfahren Outlier intuitiv erkennen. Auch werden auf gleichen Datenmengen Vergleiche zwischen den durch die Anwendung der jeweilig „im Wettbewerb“ befindlichen Ansätze erkannten Outliermengen gezogen. Im Ergebnis steht jedoch nur die Antwort auf die Frage, ob diese Verfahren gleiche, unterschiedliche, gleich viele, weniger oder mehr Outlier erkannt haben. Meist wird für den Vergleich der erkannten Outlier in realen Datenmengen Expertenwissen genutzt, um einen Bezug zu durch den Anwender erwarteter Outliermengen herzustellen, konkret z.B. bei der Krebserkennung, beim Kreditkartenbetrug, etc.

Eine wohldefinierte und systematisch umfassende qualitative Betrachtung von Outliern, welche durch diverse Verfahren erkannt werden, ist nicht gegeben. Hierbei handelt es sich um ein Grundproblem der KDD, aber auch der Outliererkennung im speziellen. Es ist dadurch begründet, dass bisher unbekanntes Wissen entdeckt wird, welches jedoch lediglich Rückschlüsse auf weitere Erkenntnisse ermöglicht, um die Ursachen, die den Wirkungen vorausgehen, zu erforschen. Hawkins Definition von Outliern gesteht diese Einschränkung bereits durch die vorsichtige Wortwahl ein, denn sie spricht vom Verdacht, nicht vom Fakt. Und sie versucht nicht, die Mechanismen, auf welche Bezug genommen wird, formal zu beschreiben.

Wie ist nun eine nutzbare Vergleichsbasis trotzdem herzustellen? Dies kann dadurch erreicht werden, dass Expertenwissen und Hintergrundwissen eingesetzt wird, um Kategorien der durch den Anwender zu erwartenden Outlier zu beschreiben. Aber es muss deutlich gemacht werden, dass ein Vergleich zwischen dieser Basis und den Ergebnissen eben nur sichtbar macht, wo sich diese anwendungsbezogenen, intuitiven, aber stark eingegrenzten Erwartungskategorien von den durch Verfahren erkannten Outliern unterscheiden. Eine ableitbare Aussage zur Qualität der erkannten Outlier ist nicht zu treffen, da sie keine systematische Basis hat. Die Forschung im Bereich der Qualität der erkannten Outlier steht noch am Anfang, wird aber von einigen Wissenschaftlern vorangetrieben [55]. Im Fokus dieser Arbeit liegt jedoch die empirische Betrachtung einer konkreten Anwendung von Outlierverfahren. Ein Anspruch auf die systematische Ordnung besteht nicht.

Ein Weg zur Gewinnung von Expertenwissen über die Inhalte von Newsgruppen besteht darin, einen Fachgebietsexperten heranzuziehen, um Newsartikel inhaltlich in von ihm kategorisierte Outlier und Nicht-Outlier mengenmäßig zu trennen. Dies ist z.B. bei Gruppen mit sehr engen, fachlich äußerst spezifischen Themen oder bei hauptsächlich für vergleichbare Experten zugänglichen Newsgruppen ein sinnvoller Weg. Sobald jedoch die Zahl und Art der Nutzer breiter angelegt ist, vor allem aber auch, weil USENET News als Medium eine freie Diskussion und ein umfängliches Frage-Antwort-System zwischen Experten und Nicht-Experten darstellen, ist dieser Ansatz nur schwer umzusetzen. Zudem ist nicht sichergestellt, dass der Experte wirklich ein plausibler und repräsentativer Experte ist. Da derartige Experten für diese Arbeit auch nicht zur Verfügung standen, nimmt der Autor auf einen derartigen Ansatz keinen Bezug.

Eine zweite Möglichkeit ist die Nutzung von Hintergrundwissen um das USENET Newssystem und seine Mechanismen und das Verhalten der Anwender, welches innerhalb des Systems technisch dokumentiert ist. Auf Basis dieses Wissens werden die folgenden Kategorien zu erwartender Outlier vorgeschlagen und definiert. Mit Hilfe dieser Definition wiederum werden die Testmengen in Outlier und Nicht-Outlier („Average“) anhand genereller Nutzungsmechanismen unterteilt.

4.4.1. Übersicht über Kategorien von zu erwartenden Outliern

Für die experimentelle Evaluation von Verfahren, sowie geeigneter Vor- und Nachbearbeitungsschritte ist es sinnvoll, auf einer Testmenge mögliche Outlierkandidaten bei USENET Newsgruppen unter den Objekten der Gesamtmenge vorzuklassifizieren. Dies erlaubt den späteren Vergleich zwischen erkannten und erwarteten Outliern mit der Möglichkeit, Überlappungen und Differenzen entsprechend zu interpretieren. Abbildung 36 zeigt die für diese Arbeit gewählte Kategorisierung von erwarteten Outliern in einer Übersicht.

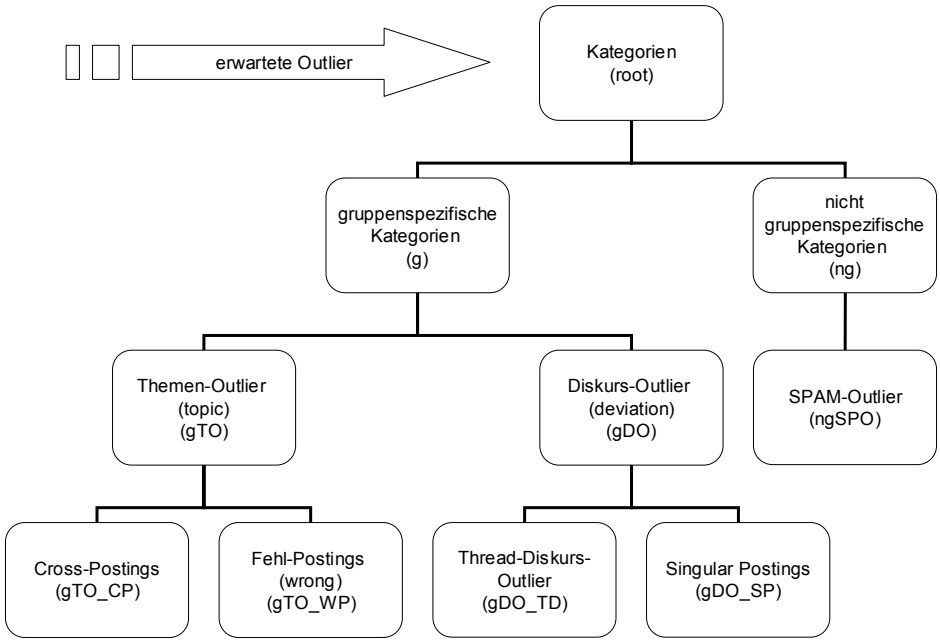


Abbildung 36 - Übersicht erwarteter Outlier-Kategorien in USENET News

Abbildungsbeschreibung: Eine Reihe von Kategorien für zu erwartende Outlier werden als Vergleichsbasis eingeführt, wobei die Einordnung der Nachrichten möglichst disjunkt ist.

Die Kategorien wurden möglichst disjunkt definiert. Zuerst wird zwischen jeweils für die Gruppe spezifischen Outliern und nichtspezifischen Outliern unterschieden. Gruppenspezifische Outlier beziehen sich konkret auf die Gruppe, d.h. ihren thematischen Inhalt bzw. ihre Diskussionsteilnehmer. Hier wird zwischen auf Themen und auf Diskussionsteilnehmer bezogenen Outliern unterschieden. Unter die erste Kategorie fallen Cross-Postings, also Postings, welche sich zwar auf die Gruppe beziehen und somit für sie spezifisch sind, jedoch auch in andere Gruppen versendet wurden und daher entweder für diese Gruppe mit einem anderen Thema genauso spezifisch sind, was die Spezifiziertheit für die konkrete untersuchte Gruppe beeinflusst. Aber auch Fehl-Postings, die ein der Gruppe nicht zugehörendes Thema betreffen, gehören dieser Kategorie an. Sie werden von SPAM-Outliern dadurch abgegrenzt, dass es sich um ein unbewusstes Posting handelt, während SPAM-Outlier bewusst die fehlende Themennähe in Kauf nehmen. Unter die zweite Kategorie fallen Einzelpostings, also solche, denen keine Diskussion folgt bzw. zugeordnet werden kann, die aber themenzugehörig sind. Und weiterhin fallen darunter Thread-Diskurs-Outlier, also Artikel innerhalb eines Threads, welche aber dazu geeignet sind, durch einen Diskurs im Thread den Diskussionsfluss zu unterbrechen oder zu sabotieren. Alle diese Kategorien werden im Folgenden detailliert beschrieben.

4.4.2. Nicht-gruppenspezifische Kategorien (ng)

SPAM-Outlier / ngSPO

Als SPAM-Content-Outlier seien solche Newsartikel in einer Gruppe definiert, welche alle Voraussetzungen erfüllen, die für SPAM [84] im Generellen gelten. Unter diese Kategorie fallen Newsartikel mit eindeutiger Einordnung in UBM – Unsolicited Bulk Messages (ähnlich Unsolicited Bulk Email – UBE), d.h. Nachrichten mit einem unangebrachten Werbeinhalt oder Schwemmnachrichten zur Propagierung von Produkten, Themen und Ansichten, die mit dem eigentlichen Thema der Newsgruppe nicht im Zusammenhang stehen müssen, aber können. Solche Artikel enthalten oft Anpreisungen für sexuelle Inhalte, Webseiten oder Stimulanzien,

propagieren Schönheitsoperationen und deren Vorteile in Partnerschaften bzw. werben für Arzneien, Partnerschaftsvermittlungen etc. In vielen Fällen enthalten derartige SPAM Nachrichten auch sog. Scams, d.h. bahnen Kontakte mit einer betrügerischen Absicht an. Beispiele hierfür sind Kettenbriefe, Nachrichten im Sinne der unter diesem Typnamen bekannt gewordenen Nigeria-Connection oder Phishing-Nachrichten, welche auf Webseiten weiterleiten mit dem Ziel, durch Aufruf von kompromittierten Programmbestandteilen Kontoinformationen (Persönliche Identifikationsnummern (PIN) oder Transaktionsnummern (TAN) oder Passwörter im allgemeinen) abzufischen, Zugangsdaten zu erhalten oder Viren, Trojaner und Würmer, z.B. zum Aufbau von BotNets auf dem dann infizierten System, vom Anwender unbemerkt zu installieren. Letztlich umfasst diese Gruppe auch Meinungsäußerungen unerwünschter Art, inkl. rasisstischer, entwürdigender oder fehlleitender Inhalte.

Gemein ist allen diesen Nachrichten, dass trotz einer fehlenden inhaltlichen Eingrenzung (Wohldefiniertheit) der Anwender intiutiv sofort in der Lage ist, diese zu identifizieren. Daher ist zwar die Unterteilung der Datenmenge in diese Kategorie zur Schaffung einer Vergleichsbasis nicht wohldefiniert. Dies schadet aber dem Erkenntnisgewinn nicht, da mit Bezug auf die Ausführungen der vorangegangen Kapitel gar nicht von einer abschließenden qualitativen Beurteilung im Sinne einer Verfahrensbewertung ausgegangen werden kann.

4.4.3. Gruppenspezifische Kategorien (g)

1. Themen-Outlier (gTO)

1.1. Cross-Posting-Outlier / gTO_CP

Als Cross-Posting-Outlier werden solche Nachrichten in einer Newsgruppe definiert, welche in ihrer Header-Zeile zur Spezifizierung der Newsgruppe mehr als eine Newsgruppe enthalten. Dieser Artikel ist vom Sender willentlich, oder in Erwiderung eines bereits als Cross-Posting versandten Artikels mglw. auch unwissentlich, an mehr als eine Newsgruppe versandt worden. Dies deutet darauf hin, dass es sich um einen newsgruppenübergreifenden Inhalt handelt. Mindestens lässt sich aber der Vorgang eines Cross-Postings im Vergleich zum Normvorgang eines Postings (Versand nur an diese eine Newsgruppe) als grundlegend verschiedenartiger Mechanismus definieren. Cross-Posting-Outlier ergeben somit eine Kategorisierung, welche einen sinnvollen direkten Vergleich zugrunde liegender verschiedenartiger Mechanismen erlauben würde. Da die Outlier Definition nach Hawkins genau den Verdacht auf solch verschiedenartige Mechanismen erhärten soll, ist eine Erkennung dieser erwarteten Outlier Form durch die angewandten Verfahren von besonderem Interesse.

Insgesamt kommen Cross-Postings selten vor, häufig dann jedoch in der Ausprägung eines Cross-Postings in inhaltsverwandte Newsgruppen (also solche, welche entweder in der Hierarchie oder themenbezogen eng beieinander liegen).

1.2. Fehlpostings (Wrong-Postings) / gTO_WP

Als Fehlpostings seien solche Artikel definiert, welche mit dem Thema der Newsgruppe so offensichtlich keinen Zusammenhang darstellen, dass sie sofort intuitiv von einem Leser, selbst wenn er keinerlei Experten- oder Hintergrundwissen dieser Gruppe hat, als solche identifiziert werden können. Es seien also Artikel, die aufgrund eines Anwendungsfehlers in eine falsche Newsgruppe gepostet wurden. Eine klarere Definition dieser Kategorie ist jedoch leider nicht möglich, vor allem weil das Newssystem dynamische Interaktionen der Nutzer auslöst und fördert. Es ist also zu erwarten, dass hilfreiche Nutzer (auch mehr als einer) in manchen Gruppen solche Fehlpostings mit einem Hinweis auf den Fehler oder die richtige Gruppe beantworten. Zusätzlich liegt es in der menschlichen Natur – und ist insofern in den News oft zu beobachten – das solchen Fehlern auch mit harscher Kritik begegnet wird. Daher besteht die Möglichkeit, dass andere Nutzer bezugnehmend auf die Fehlernachricht mit Meinungsäußerungen zum ursprünglichen Sender der themenfremdem Nachricht Stellung nehmen. Ein solcher, wenn auch kleiner Diskussionsthread führt zu einer Veränderung der Datenmenge. In Newsgruppen mit einer durchschnittlich eher stark begrenzten Anzahl an Artikeln pro Thread (ein Thread entspricht in etwa einem sinnvollen Objektcluster) würde also ein Cluster von Nachrichten rund um ein Fehlposting eine ähnliche Größe aufweisen, wie ein regulärer „Themencluster“. Die Unterscheidbarkeit anhand der Clustergröße wäre erwartungsgemäß sehr gering.

Trotz all dieser Probleme erscheint die Einbeziehung von generellen Fehlpostings als sinnvoll, da diese Outlier-Form in einem System immanent ist, welches von Menschen benutzt wird, die Fehler in der Anwendung als natürlichem Bestandteil dieser Nutzung machen.

2. Diskurs Outlier (gDO)

2.1. Thread-Diskurs-Outlier / gDO_TD

Als Thread-Diskurs Outlier seien solche Diskussionsbeiträge innerhalb eines Diskussionsthreads definiert, welche innerhalb des Threads einen Diskurs auslösen, also z.B. einen neuen, eigentlich themenfremden Unterbaum im Thread erzeugen, der nachweislich nichts mit dem ursprünglichen Thema des Threads zu tun hat.

Beispiele für solche Outlier sind z.B. Diskussionsbeiträge, welche geeignet sind, die ursprüngliche Diskussion zu sabotieren. Ausprägungen sind z.B. nicht themenrelevante Einwände, welche auch keinen Nutzen zur Diskussion besteuern. Genannt sei hier die alte Diskussion um die Vermeidung aller augescheinlichen Probleme, sofern die Eigentümer dieses Problems nur die „richtige“ Software oder das „richtige“ Betriebssystem benutzen würden. Aber auch religiös fanatische Äußerungen, Beschimpfungen oder Ablenkungsmanöver auf andere Themen sind Beispiele dafür, wie sich Thread-Diskurs-Outlier äußern können.

2.2. Singular Postings /gDO_SP

Als Singular Postings sollen solche Newsartikel definiert sein, die nicht Bestandteil eines Diskussionsthreads sind. Threads werden in den USENET News, wie bereits beschrieben, durch die Rückwärtsreferenzierung im „References“-Feld im Header eines Artikels mit einem technischen Mechanismus hinterlegt, der es dem System erlaubt, Artikel im Newsreader als Thread zusammengefasst anzubieten. Meist sind zudem die Subject-Felder im Header eines Newsartikels gleichlautend, wobei ein Zusatzkürzel „Re:“ oder „AW:“ den Diskussionsprozess verdeutlicht. Der Newsreader ist zudem in der Lage, Newsartikel mit überlappenden References-Feldinhalten zu einem Thread zu vereinen (da lt. Standard immer nur eine begrenzte Zahl von Rückwärtsreferenzierungen vorgehalten wird).

Zur Gruppe sollen gemäß dieser Kategorisierung ausdrücklich keine virtuellen Threads durch Querreferenzierungen (z.B. durch Zitate mit Artikelquerverweis oder Quotings aus anderen Artikeln eines fremden Threads) gehören und auch keine abgeleiteten Threads (meist dargestellt durch „<Subject> (was: <old-Subject>)“. Da diesen virtuellen Threads kein im Newssystem selbst verankerter technischer Mechanismus zugrunde liegt, wären sie im Gegensatz zu den gDO_SP nicht stichhaltig definierbar, erlaubten auf der anderen Seite aber auch keine einfache intuitive Erkennung durch den Anwender wie im Fall der offensichtlichen SPAM-Artikel (ngSPO). Daher erscheint eine Kategorisierung inkl. virtueller Threads als zu wenig abgrenzend, um dem Erkenntnisprozess zu nutzen.

4.5. Auswahl von Standardverfahren

Bei der Auswahl der für die praktische Implementierung umzusetzenden Outlier-Erkennungsverfahren standen zwei Überlegungen im Vordergrund. Erstens sollten es Verfahren sein, welche sich gut auf die Applikation USENET News anwenden lassen, bei denen also ein echter Erkenntnisgewinn zu erwarten ist. Zum anderen erschien ein Schnitt durch die Reihe der maßgeblichen statistischen Verfahrensansätze sinnvoll, um möglichst gute Sichten auf das Outlierthema in Bezug auf Newsgruppen von verschiedenen Seiten her zu erhalten. Daher wurden die folgenden Verfahren für eine Umsetzung betrachtet:

Entfernungsbasierte *DB(p,D)*-Outlier

Dieses von Knorr und Ng [3] vorgestellte Outliererkennungsverfahren ist insbesondere dadurch interessant, dass es alle bekannten statistischen Outliererkennungsverfahren unifiziert. Diese lassen sich über eine geeignete Wahl der Parameter *p* und *D* nachbilden. Zudem erfolgt über dieses Verfahren eine homogene und globale Betrachtung der Datenmenge über ein statistisches Entfernungsmaß, welches dazu dient, die Objektmenge anhand von prozentualen Vorgaben bzgl. dieses Entfernungsmaßes in Outlier und Nicht-Outlier zu teilen.

Hier ist vor der Anwendung des Verfahrens nicht bekannt, wie viele Outlier identifiziert werden.

Entfernungsbasierte *D(k,n)*-Outlier

Dieses Outlier-Verfahren wurde von Ramaswamy, Rastogy und Shim [13] vorgestellt und bezieht sich auf das entfernungsbasierte Outlierverfahren. Es bietet jedoch gegenüber dem unifizierenden Verfahren von Knorr und Ng den Vorteil, dass der Anwender die Parameter *p* und *D* nicht vorher festlegen muss. Vielmehr legt er einfach eine Anzahl der top-*n* gesuchten Outlier fest, welche anhand eines statistischen Entfernungsmaßes identifiziert werden. Dieses muss der Anwender aber nicht ex ante festlegen, sondern er bezieht sich einfach auf die Entfernung eines Objektes zu seinem *k*-ten nächsten Nachbarn. Somit bildet das Verfahren auch eine gewisse Lokalität ab, da es sich um ein Nachbarschaftsmaß handelt.

Dichtebasierte *LOF(MinPts)*-Outlier

Von Kriegel, Sandner, Breuning und Ng [4] definierte Outlierverfahren nutzen ein statistisches Dichtemaß zur Indentifizierung von Objekten, welche Outlier sein könnten. Dabei wird der Begriff der Lokalität durch den Bezug auf lokale Nachbarschaftsdichten eingeführt und die Outliererkennung resultiert nicht in einem Status, sondern in einem entsprechenden Faktor (Lokaler Outlier Faktor – *LOF*). Da dieses Verfahren einen grundsätzlich verschiedenartigen Ansatz gegenüber dem Entfernungsmaß wählt und ein sehr grundlegendes Verfahren ist, wurde es für die Implementierung ausgewählt.

Aufgrund der zu erwartenden hohen Zahl an Dimensionen und der Spärlichkeit der Besetzung dieses Raumes wurde auf die Umsetzung eines optimierten Verfahrens zu dichtebasierten lokalen Outliern (top-*n LOF*) nach Jin, Tung und Han [5] verzichtet. Dieses setzt auf Microclustering mittels BIRCH [60] auf, einem notwendigen Vorausschritt, welcher in hochdimensionalen Räumen den Performancegewinn durch die Optimierung der Outliersuche gegenüber dem klassischen *LOF*-Verfahren erreicht. Aber es besteht bei USENET News die Gefahr, dass beim Vorclustering keine ausreichend abgrenzbaren Cluster erkannt werden, um das top-*n* Verfahren für den *LOF* hier überhaupt einzusetzen.

Erkennung von Outliern durch Clustereleminierung

Bei diesem Ansatz wird ein geeignetes Clusteringverfahren eingesetzt, um alle möglichen Cluster zu identifizieren. Im Ergebnis sollte ein solcher Verfahrenseinsatz eine Menge von Clustern identifiziert haben, deren zusammengefassten Elemente eine Untermenge der Gesamtmenge darstellen, welche mit hoher Wahrscheinlichkeit keine Outlier sind. Im Gegensatz dazu sind folgerichtig die Elemente der disjunkten Untermenge aller Objekte, welche keinem Cluster zugeordnet werden konnten, mit hoher Wahrscheinlichkeit Outlier. Ggf. müssen solche Clusteringverfahren, welche Outlierkandidaten nicht bereits selbst als Restmenge kennzeichnen können, um eine Nachbereitung zur Kennzeichnung all der Elemente, welche nicht Bestandteil eines erkannten Clusters sind, erweitert werden. Zudem ist die Wahl der Clusteringverfahren auf solche beschränkt, welche nicht zwangsweise alle Elemente oder einen Großteil der Elemente zu Clustern als Verfahrenspräferenz zuordnen, weil eine Restmenge als solche verfahrenstechnisch nicht vorgesehen ist.

Da für alle Clustereleminierungsansätze die gleichen Nachteile zu erwarten sind, wie für das top-*n LOF*-Verfahren, wurde auf detaillierte Experimente nach anfänglichen Tests, welche diese Nachteile bestätigten, verzichtet.

Dichtemessung in Projektionen

Im Prinzip legt die Betrachtung der Eigenschaften von USENET Newsartikeln in Bezug auf eine Suche nach Outliern nahe, Dichtemessungen in Projektionen (wie sie von Aggarwal und Yu vorgeschlagen werden [68]) einzusetzen, da durch die Vektorisierung ein hochdimensionaler Raum entsteht und bei dem Verhältnis von Anzahl der Dimensionen zu der Kardinalität der zu untersuchenden Gesamtmenge auch schnell eine spärliche Besetzung vermutet werden kann. Allerdings bezieht sich die Methode der Dichtemessung in Projektionen auf euklidische Räume, wobei von einer metrischen Entfernungsverteilung der Objekte ausgegangen wird. Im Bereich der Textvektorisierung wird jedoch oft das Kosinusmaß als Entfernungsmaß vorgeschlagen, welches sich auf den Winkel zwischen den Vektoren bezieht. Im späteren wird noch untersucht, ob eine Umstellung vom euklidischen Maß auf dieses Entfernungsmaß im Rahmen anderer Verfahren Vorteile bringt. Auf eine Umsetzung wurde im Rahmen der Experimente dieser Arbeit verzichtet, weil das Verfahren in den Projektionen von Normalverteilungen ausgeht, die nicht zu erwarten sind, und weil die Performance des Algorithmus nicht praktikabel erscheint.

Emergente selbstorganisierende Karten (ESOM)

Ein weit verbreiteter Ansatz zur Visualisierung und zum Erkenntnisgewinn in hochdimensionalen Räumen ist der Einsatz von aus der biologischen Selbstorganisation abgeleiteten Strukturierungsmechanismen. Kohonen hat dazu die sog. Self Organising Maps – SOM [88] eingeführt. Diese bedienen sich eines lernenden neuronalen Netzes, welches hochdimensionale Lernbeispiele durch die Veränderung von Gewichtsvektoren auf meist zweidimensionalen Karten mit Neuronen abbildet. Zusätzlich wird ein aus der Biologie adaptiertes netzinternes Anpassungsverfahren genutzt, welches mit einer abnehmenden Intensitätsfunktion die gelernten Gewichtsveränderungen in der Karte selbst propagiert. Dadurch ergibt sich in der SOM eine Repräsentation des Urprungsraumes durch die entsprechenden Neuronen (Netzknoten) und ihre Ausprägung.

Um diese Mechanismen für die Erkennung von Strukturen in hochdimensionalen Räumen mit vielen Objekten besser nutzbar zu machen, wurden von Ultsch [89] im Bereich Datenbionik gewisse Erweiterungen der SOM vorgenommen. Zum einen ergibt sich durch den Einsatz von Karten mit sehr vielen Neuronen ein emergentes Verhalten, welches die Abbildung von hochdimensionalen Räumen innerhalb des Netzes besser erlaubt. Zum anderen wurden diverse grafische Kartenadaptionen eingeführt [87], welche eine grafische Visualisierung unter Einsatz von Informationen über statistische Entfernungs- bzw. Dichtemaße (oder eine

Kombination derselben) im Urprungsraum erlauben. Darauf aufbauend wird eine grafische Erkennung von Clustern (U*F Clustering [90]) genauso möglich, wie eine Identifikation von Outliern durch den Nutzer.

4.6. Anpassung von Standardverfahren

Zwei der o.g. Verfahren zur Erkennung von Outliern (*DB(p,D)* und *D(k,n)*) definieren als grundlegendes statistisches Maß eine euklidische Entfernung, welche zwischen den Objektvektoren gemessen wird. Bei der Beurteilung von Texten ergibt sich damit ein durch die Länge der Vektoren beeinflusstes Maß, was im Rahmen der Anwendungsdomaine des Information Retreival [1] in Texten, speziell in Newsartikeln, nicht als günstig erscheint. Erfahrungsgemäß eignet sich die Kosinusdistanz wesentlich besser, da hierbei der Winkel zwischen den Vektoren als Ähnlichkeitsmaß entscheidend ist. Damit ergibt sich auch eine mögliche Anpassung von Standardoutlierverfahren für USENET Newsgruppen, indem als Entfernungsmaß statt der euklidischen Distanz die Kosinusdistanz verwendet wird.

euklidische Distanz:

$$d_e(\vec{x},\vec{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

Kosinusdistanz:

$$d_c(\vec{x},\vec{y}) = 1 - \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2} + \sqrt{\sum_i y_i^2}}$$

Die Kosinusdistanz als ein Abstandsmaß betrachtet Dokumente als gleich, die aus der gleichen Zusammensetzung von Termen bestehen, auch wenn diese einzelnen Terme mit unterschiedlicher Häufigkeit vorkommen. Die euklidische Distanz sieht in diesem Zusammenhang einen größeren Unterschied durch die Häufigkeit, als z.B. bei kleinen Dokumenten mit wenigen unterschiedlichen Termen. Da der Kosinus des Winkels der Vektoren zueinander als Ähnlichkeitsmaß fungiert, wird für die Kosinusdistanz als Abstandsmaß die o.g. Invervierung durchgeführt. Da Textvektoren in Bezug auf deren Termhäufigkeiten als Werte auf einem Koordinatensystem aller Terme nur im ersten Quadranten vorkommen, ergibt sich das in Abbildung 37 gezeigte Bild.

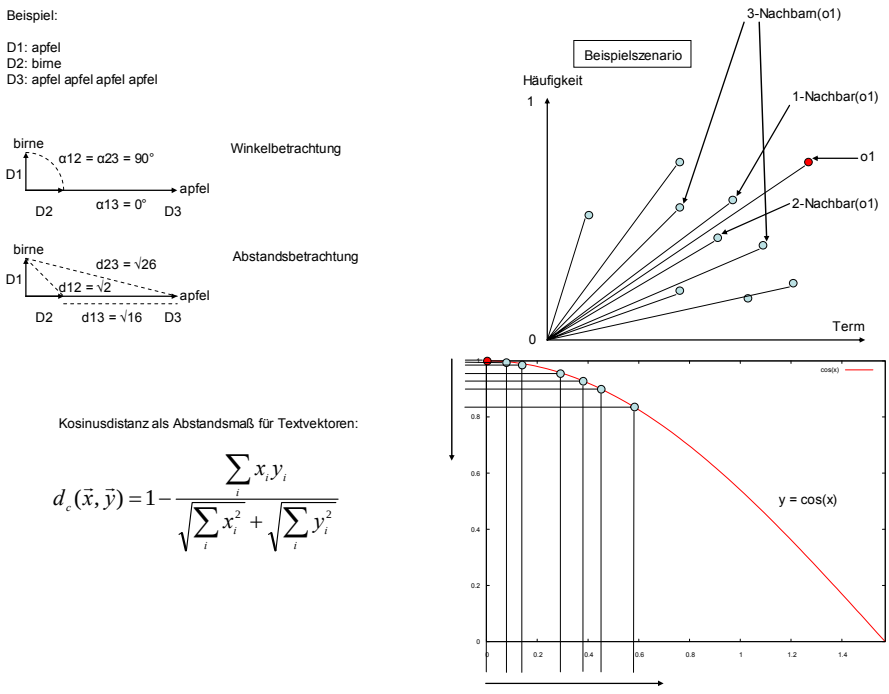


Abbildung 37 - Anwendung der Kosinusdistanz für Textvektoren

Abbildungsbeschreibung: Als alternatives Distanzmaß findet im Bereich des Information Retreival z.B. die Kosinusdistanz Anwendung. Im Unterschied zur euklidischen Distanz spielt die Häufigkeit der Terme eine geringere Rolle, um die Unterschiede oder Ähnlichkeiten zwischen Texten zu beurteilen.

Die Kosinusdistanz bestimmt also mit steigendem Winkel zwischen zwei Vektoren eine sinkende Ähnlichkeit zwischen diesen und dementsprechend einen höheren Abstand. Das Beispiel verdeutlicht noch einmal den Unterschied zwischen der Auswirkung der Anwendung der euklidischen und der Kosinusdistanz. Alternativ wäre eine bloße Anwendung des Winkels (in Radiant) denkbar, um einen linearen Zusammenhang zwischen Winkel und Abstand herzustellen, während die Kosinusdistanz die Ähnlichkeit kleiner Winkelabstände stärker betont.

Abbildung 38 zeigt an einer zweidimensionalen Beispielmenge mit 50 Objekten die Auswirkungen der Anwendung verschiedener Distanzmaße für das *D(k,n)*-Verfahren mit dem Wert *k*=5 und *n*=10. Dabei ergeben sich für die euklidische Distanz (hier mit denselben Ergebnissen für das Quadrat der euklidischen Distanz) andere *D(k,n)*-Outlier, als für die Anwendung der Kosinusdistanz als Abstandsmaß (hier mit denselben Ergebnissen wie für das Winkelmaß in Radiant) bzw. für die Anwendung des Kosinus als Ähnlichkeitsmaß.

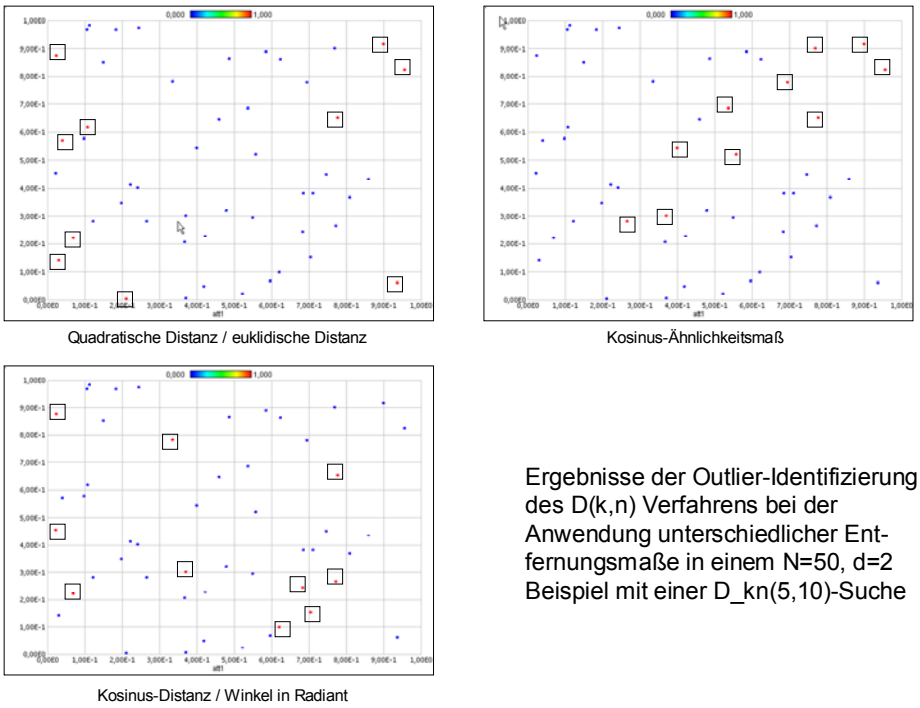


Abbildung 38 - Anwendung von verschiedenen Distanzen im *D(k,n)*-Verfahren

Da auch das Verfahren der dichtebasierenden Outlier letztendlich auf ein Entfernungsmaß zurückgreift, um aus den lokalen Erreichbarkeitsdistanzen die lokalen Erreichbarkeitsdichten und letztendlich aus deren Verhältnis die lokalen Outliergrade zu berechnen, kann auch auf das *LOF(MinPts)*-Verfahren versuchsweise als Anpassung die Kosinusdistanz angewendet werden. Allerdings sei hier darauf verwiesen, dass zumindest die in [4] angeführten Theoreme sich ausdrücklich auf die verwendete euklidische Distanz stützen, um Aussagen über statistische Schwankungen und geeignete Parametergrenzen zu machen. Auch auf das ESOM Verfahren lassen sich sowohl euklidische, als auch Kosinusdistanz anwenden.

Durch die experimentelle Evaluation kann sodann gezeigt werden, ob sich eventuell Verbesserungen bei der Outliererkennung durch Verwendung der Kosinusdistanz erreichen lassen, oder ob die Anwendung zur Identifizierung unterschiedlicher Gruppen von Outliern führt.

4.7. Angepasste Vorverarbeitungsverfahren

Wie bereits ausgeführt, sind die meisten Verfahren für die Erkennung von Outliern lediglich auf einer stark begrenzten Zahl an Dimensionen der betrachteten Datenmenge effizient durchführbar. Zudem ergibt sich durch eine ggf. spärliche Besetzung des Datenraumes zusätzlich die Herausforderung, Outlier mit einer signifikanten Abweichung zu den anderen Objekten der Gesamtmenge zu erkennen, obwohl diese Abweichung durch die Spärlichkeit im hochdimensionalen Raum mit großer Wahrscheinlichkeit nicht besonders ausgeprägt ist. Diesem Problem kann durch den Einsatz von Vorverarbeitungsverfahren begegnet werden. Diese sind dazu angelegt, speziell für die Domaine der USENET News und ihrer Newsartikel Ergebnisse von Verfahren durch Reduktion der Dimensionen bzw. Herabsetzung der Spärlichkeit zu verbessern. Hierbei lässt sich vor und nach der Vektorisierung der Texte ansetzen.

Pruning von Termen minimaler und maximaler Häufigkeit

Die Implementierung des WordVektor Tools in YALE [99] erlaubt bereits standardmäßig die Begrenzung der Vektorisierung von Termen anhand einer oberen und unteren Termanzahl. Dadurch lassen sich sehr seltene oder sehr häufig vorkommende Terme durch das Setzen dieser Grenzen aus der Menge der Dimensionen entfernen.

Textsplitting bzw. Teiltextausblendungen

Als ein Ansatz, der direkt nicht zur Identifizierung von Outliern dient, jedoch neue Erkenntnisse bezüglich einer Erleichterung der Outliererkennung mittels Beschneidung des Textumfangs vor der Vektorisierung liefern kann, wird ein Splitting von Texten unterstützt. Durch dieses Splitting kann der Header-Text eines Newsartikels für die Vektorisierung ausgeblendet werden, sodass nur der Body erfasst wird. Sinn dieser Bearbeitung ist es, wiederkehrende und bei der Vektorisierung die eigentliche Objektpositionierung möglicherweise verfälschende Artikelbestandteile zu beschneiden und so eine transparentere Vektorisierung und als Zusatznutzen eine niedrigere Dimensionszahl zu erreichen.

Dimensionsreduktion mit Singular Value Decomposition (Outlier Dimensionsreduktion)

Diese spezielle Implementierung setzt kein eigenes Verfahren um, sondern erlaubt die Anwendung der Singular Value Decomposition Methode [104] zur Reduktion der Anzahl an Dimensionen des Suchraumes auf eine sehr niedrige Zahl (i.d.R. zwischen zwei und fünf Dimensionen). Dabei wird die Verteilung der Objekte zueinander weitestgehend erhalten und es kann die Fragestellung beantwortet werden, ob eine Anwendung klassischer Verfahren auf der reduzierten Dimensionsmenge ähnliche verwertbare Ergebnisse erbringt, wie die Anwendung dieser Verfahren auf den Raum mit der ursprünglichen Zahl an Dimensionen.

Zusätzlich wird diese Implementierung dazu verwendet, die Ergebnisse von den anderen Outlierverfahren im zwei- bzw. dreidimensionalen Raum grafisch darstellbar zu machen. In diesem zweiten Anwendungsfall wird jedoch auf den zu untersuchenden Datenraum kein Einfluss genommen. Lediglich eine Projektion des Ursprungsraumes dient zur Visualisierung der Ergebnisse.

Da für eine zweite Variante der Dimensionsreduktion durch die Nutzung von selbstorganisierenden Karten (SOM) bereits ein Standardverfahren durch die ESOM Tools zur Verfügung steht, ist dafür keine spezielle Vorverarbeitung notwendig.

4.8. Ergänzung mit Hintergrundwissen über Autoren

Die im vorhergehenden Kapitel beschriebenen Vorverarbeitungsverfahren sind zwar bereits speziell nach deren Eignung für Texte und damit für die USENET News Domaine gewählt worden, sie sind aber noch nicht auf spezielle Möglichkeiten hin optimiert, die Ergebnisse zur Erkennung von Outliern durch Standardverfahren in den News zu verbessern. Sie verbessern somit also die generelle Wirkungsweise von Standardverfahren, ergänzen diese jedoch nicht.

Eine sofort ins Auge fallende Möglichkeit der potentiellen Verbesserung der Ergebnisse von Outlierverfahren ergibt sich durch die Anwendung von Wissen um die Nutzung von Newsgruppen. Solches Hintergrundwissen kann für die Ergänzung der Gesamtdatenmenge verwendet werden, um z.B. den Standardverfahren zusätzliche Informationen durch ergänzte oder reduzierte Dimensionen im Suchraum zur Verfügung zu stellen. Vor allem das Verhalten der Autoren, welches sich durch die Anzahl und Art der verfassten Newsartikel teilweise beschreiben lässt, kann als Wissen herangezogen werden.

Erhebung von Autorenwissen aus kategorisierten Beispielmengen / maschinelles Lernen

Im Rahmen der Vorverarbeitung bietet sich an, vorhandenes Wissen um die Autoren von Newsartikeln in einer Newsgruppe bereitzustellen und ggf. für die Identifizierung von Outliern direkt zu nutzen. Durch geführtes Lernen anhand einer kategorisierten Beispielmenge kann z.B. die Wahrscheinlichkeit, dass ein von einem Autor verfasster Newsartikel ein Outlier ist, anhand des Verhältnisses der vom Autor verfassten und als Outlier kategorisierten Artikel in der Beispielmenge zur Gesamtzahl der vom Autor verfassten Artikel in der Beispielmenge bestimmt werden.

$$E(Autor\_verfasst\_Outlier) = \frac{|\{vom\_Autor\_verfasste\_Outlier\}|}{|\{vom\_Autor\_verfasste\_Artikel\}|}$$ bezogen auf die Beispielmenge.

Diese Wahrscheinlichkeit wird z.B. in einem Autorenprofil gespeichert und für die Erkennung von Outliern auf eine Testmenge angewandt. Die Anwendung kann direkt geschehen, indem alle Artikel in der Testmenge, die vom Autor verfasst wurden, mit der Outlierwahrscheinlichkeit dieses Autors versehen werden und der Anwender dadurch eine Outliergewichtung pro Artikel erhält. Diese Gewichtung kann nun durch den Anwender

ausgewertet werden, indem er z.B. die Testmenge nach der Gewichtung in auf- oder absteigender Reihenfolge sortiert bzw. die Objekte der Testmenge anhand geeigneter Schwellwerte kategorisiert und so ein jeweiliges Gewicht auf einen Status abbildet.

$Outlierfaktor(Artikel\_von\_Autor) = E(Autor\_verfasst\_Outlier)$ bezogen auf die Testmenge;

$Outlierfaktor(Artikel\_von\_Autor) > Schwellwert \rightarrow Outlierstatus(Artikel\_von\_Autor) = WAHR!$

Bei diesem Verfahren gibt das Hintergrundwissen, welches aus einer kategorisierten Beispielmenge gewonnen wurde, den alleinigen Ausschlag. Dabei ist a posteriori nicht notwendigerweise bekannt, wie die ursprüngliche Kategorisierung vorgenommen wurde und welche Analyse der Beispielmenge ihr zugrunde liegt. Auch die Frage, wie repräsentativ bzw. im Allgemeinen, wie vollständig die Beispielmenge bezogen auf die Testmenge ist, kann vom Anwender nur dann beurteilt werden, wenn er entweder selbst die Kategorisierung vorgenommen hat oder wenn detaillierte Informationen über die Kategorisierung vorliegen.

Eine mögliche Ausprägung des Hintergrundwissens kann zusätzlich sein, dass jedem Autor ein harter Status (WAHR bzw. FALSCH) bzgl. seiner Outliereigenschaft zugeordnet wurde und dies als Hintergrundwissen verfügbar ist. Diese Ausprägung ist unter anderem aus dem Bereich der Black- bzw- White-List Kategorisierungen von Nachrichtenabsendern im Bereich des SPAM bekannt [84]. Ein Nachteil der direkten Anwendung solchen Wissens liegt darin, dass alle Objekte der Testmenge nur bzgl. ihrer Eigenschaft, von welchem Autor sie verfasst wurden, untersucht werden. Folgerichtig bietet es sich an, auch die anderen Eigenschaften mit Verfahren zu untersuchen, die eine Identifizierung von Outliern ermöglichen und die Ergebnisse dieser Verfahren mit dem vorhandenen Wissen geeignet zu verbinden. Hieraus lässt sich somit sowohl eine Beurteilung der Verfahrensergebnisse auf der einen Seite, als auch die Verfeinerung des vorhandenen Hintergrundwissens auf der anderen Seite erreichen.

Kombination von Autorenwissen mit Outlier-Verfahren / neue paritätische Verfahrensansätze

Alternativ kann also das vorhandene Hintergrundwissen über die Autoren mit den Erkenntnissen eines anderen Erkennungsverfahrens für Outlier zusammengeführt werden, indem die Ergebnisse der Erkennung mit dem Wissen um die Autoren sinnvoll kombiniert werden. Dabei bieten sich mehrere Wege an, diese Kombination vorzunehmen.

(a) Kombination von Kategorisierungen bzw. Wahrheitswerten

(b) Kombination von Outlierfaktoren, die durch Gewichte repräsentiert werden

Die Kombination von Wahrheitswerten bietet sich für solche Verfahren an, die feste Kategorisierungen von Objekten im Ergebnis der Outlier-Identifikation liefern und wo das Autorenwissen entweder auch durch Wahrheitswerte der Outliereigenschaft eines Autors vorliegt oder es sich zumindest mit einem geeigneten Schwellwert in Wahrheitswerte umwandeln lässt. Die abschließende Kombination ist durch eine logische UND bzw. ODER Verknüpfung erreichbar.

$Outlierstatus(Artikel\_von\_Autor) \wedge Outlierstatus(Autor) \rightarrow Outlierbewertung$ bzw.

$Outlierstatus(Artikel\_von\_Autor) \vee Outlierstatus(Autor) \rightarrow Outlierbewertung$.

Dabei kann die Bewertung im Ergebnis sowohl auf den Status der Artikel als auch auf den Status des Autors angewendet werden. Bei einer UND Verknüpfung ergibt sich in Richtung der Artikelbewertung eine einschränkende Prüfung derart, dass nur als Outlierautoren bekannte Verfasser auch Outlier produzieren. In Richtung der Autorenbewertung kann gefolgert werden, dass ein Autor nur dann Outlier verfasst, wenn (alle) von ihm in der Testmenge verfassten Artikel als Outlier von einem Verfahren erkannt wurden. Bei einer ODER Verknüpfung erfolgt die Prüfung in beide Richtungen erweiternd, d.h. noch nicht als Outlier vom Verfahren erkannte Artikel, welche von einem Outlier-Autor verfasst wurden, werden zusätzlich als Outlier erkannt. Und solche Autoren, die als Outlier kategorisierte Artikel der Testmenge verfasst haben, aber noch keinen eigenen Outlier-Status haben, werden nun auch als Outlier verfassende Autoren geführt. Selbstverständlich ist auch eine Kombination aus UND und ODER Verknüpfung je Richtung möglich, um das Ergebnis je Richtung entweder zu erweitern oder zu beschränken.

Eine Kombination von Gewichtungen, welche sich aus Outlierfaktoren sowohl für die Artikel als auch für die Autoren ergeben, ist für eine sehr viel differenziertere Betrachtung sinnvoll. Sie erfordert auf der anderen Seite jedoch auch eine intensivere Interpretation der Ergebnisse. Zuallererst müssen die Gewichtungen aufeinander angepasst werden, damit eine Verrechnung der Gewichte überhaupt zu sinnvollen Ergebnissen führt. Dafür sind als Wahrheitswerte vorliegende Informationen über Autoren in Gewichte umzuwandeln, die z.B. mit Maximalwerten oder Schwellwerten der zu erwartenden Gewichtsintervalle, welche von den Outliererkennungsverfahren ermittelt werden, korrespondieren. Ermittelt das Outlierverfahren selbst Wahrheitswerte und die Autoreninformation soll zur Beurteilung dieser Werte herangezogen werden, so sind die

Wahrheitswerte umgekehrt in Gewichte umzuwandeln, die eine Bewertung durch Verrechnung mit dem Autorenwissen erlauben. Liegen beide Mengen an Eingangsgrößen bereits als Gewichte vor, so ist eine geeignete Normierung erforderlich, um beide Seiten zweckgebunden zu verrechnen. Nach dieser Zweckbindung richtet sich die Verrechnungsmethode, also z.B. die Addition bzw. Multiplikation von Gewichten oder auch die Durchschnittsbildung aus Gewichten. In Kapitel 5.2.9 wird eine Reihe von möglichen Verrechnungsmethoden vorgestellt. Folgende Zweckbindungen der Verrechnung sind u.a. denkbar:

1. Verstärkung bzw. Dämpfung der ermittelten Outliereigenschaft durch Hinzuziehen von Autorenwissen. In diesem Fall wird durch Addition (von Gewichtswerten in geeigneten Intervallen) ein vom Verfahren ermitteltes Gewicht gegenüber einem Vergleichswert verstärkt oder verringert. Beidseitig stark als Outlier gewichtete Objekte werden somit stärker herausgehoben als nur einseitig stark gewichtete oder beidseitig schwach gewichtete Objekte.
2. Gewichtete Zusammenführung der Bewertung der Outliereigenschaft, welche durch Gewichtsfaktoren vorliegt. Hierbei kann durch Multiplikation der Gewichte eine weiche Schablonierung durchgeführt werden, welche sich in der Natur an der harten logischen Verknüpfung orientiert, jedoch Abstufungen und Schwellwertbetrachtungen erlaubt.
3. Revidierung vorhandenen Autorenwissens durch Anwendung einer durch ein Verfahren ermittelten Outliergewichtung für Artikel auf das herangezogene Autorenwissen. Dies kann z.B. durch Addition, Multiplikation oder Durchschnittsbildung auf einer oder beiden Seiten der Gewichtungen erreicht werden. Hierbei sind zum einen iterative Verbesserungen von Autorenwissen denkbar, aber auch die Zusammenführung von Ergebnissen verschiedener Outlierverfahren auf ein und derselben Testmenge bezogen auf die Autorenmenge.

Im Fazit bietet die Anwendung von Autorenwissen vielfältige Möglichkeiten, die Ergebnisse von Outliererkennungsverfahren zu verbessern oder mit vorhandenem Wissen abzugleichen. Allerdings ist hiermit kein abschließendes System beschrieben oder über dessen Wirkung ein formeller Nachweis geführt. Es ist jedoch absehbar, dass allein die Anwendung von Autorenwissen (unabhängig von dessen Gewinnung) geeignete statistische Outlierverfahren nicht vollständig substituieren, sondern vor allem sinnvoll ergänzen kann. Zudem werden Gemeinsamkeiten mit den vorgestellten paritätischen Verfahren (Spatial Outlier, Spatial Temporal Outlier, semantische Outlier) deutlich, da auch hier eine Unterteilung der Attributmenge und sodann eine Zusammenführung von Wissen, welches in getrennten Schritten aus disjunkten Attributmengen gewonnen wurde, stattfindet. Hierbei scheint sich eine Tendenz bei der Optimierung von Ansätzen der Outliererkennung für spezifische Anwendungsdomainen abzuzeichnen.

In dieser Arbeit wird aufgrund der breiten Möglichkeiten, Hintergrundwissen von Autoren mit anderen Verfahrensergebnissen zu kombinieren, die Awendung von Autorenwissen lediglich grundsätzlich untersucht. Dies geschieht zum einen in der Bildung von Autorenwissen aus der Vorkategorisierung durch Ermittlung von Erwartungswerten für Outliereigenschaften von Autoren aufgrund der Anzahl der von einzelnen Autoren verfassten vorkategorisierten Outliern im Verhältnis zu allen von diesen einzelnen Autoren verfassten Artikeln. Zum anderen wird eine Cross-Validierung durchgeführt, die in der Experimentbeschreibung in Kapitel 6.3.5 detailliert ausgeführt ist.

4.9. Erstellung einer Testdatenmenge

Um eine Evaluation der verschiedenen Verfahren zu ermöglichen, ist die Erstellung einer Testmenge notwendig. In dieser können Objekte (d.h. konkret Newsartikel) in Bezug auf deren potentielle Outliereigenschaft aus Nutzersicht nach dem vorgestellten Schema in Kapitel 4.4 vorkategorisiert werden. Bei Anwendung eines Verfahrens ist ein Vergleich erkannter Outlier und der entsprechenden Kategorisierung der Objekte möglich, um Rückschlüsse auf die Verfahren zu ziehen. Dabei ist besondere Vorsicht geboten, da eine theoretisch fundierte Aussage über die Ursachen der Einstufung eines Objektes aus Verfahrenssicht und der „Qualität“ dieser Einstufung von den meisten Verfahren nicht unterstützt wird und Thema weitergehender Forschung ist.

Die konkret für die Anwendung der Verfahren eingesetzten Testmengen werden in Kapitel 6.2 beschrieben und gleichzeitig zur Nachvollziehbarkeit der Experimente im Rahmen dieser Arbeit über das Internet verfügbar gemacht.

5. Praktische Umsetzung und Implementierung

In diesem Kapitel wird die praktische Umsetzung der ausgewählten Verfahren beschrieben. Diese Beschreibung dient zum Verständnis der Ergebnisse von Experimenten, welche im Anschluss daran dargelegt werden. Auch erlaubt sie dem Leser eine eigene Anwendung oder eine weitere Entwicklung der vorhandenen Implementierungen. Alle Umsetzungen sind unter der Gnu Public License (GPL) verfügbar. Alle weiteren Werkzeuge, Systeme und Programme, welche Einsatz im Rahmen dieser Arbeit finden, sind zum Zeitpunkt der Veröffentlichung der Arbeit zur freien Nutzung unter ähnlichen Lizensierungsformen verfügbar.

5.1. Anwendung der YALE Umgebung und des Outlier-PlugIn

Die eigenen Implementierungen und Verfahrensumsetzungen in dieser Diplomarbeit bedienen sich der allgemeinen Lernumgebung YALE – Yet Another Learning Environment ([92], [93] und [94]), welche am Lehrstuhl Künstliche Intelligenz des Fachbereiches Informatik der Universität Dortmund entwickelt wurde und angeboten wird. YALE, eingesetzt in der zur Zeit aktuellen Version 3.2 und 3.3, wird als Lernumgebung für KI und KDD Anwendungen ihrerseits in der Programmiersprache Java implementiert und gepflegt. Für die Ausführung ist mindestens ein Java Runtime Environment (JRE) erforderlich. Für die Weiterentwicklung wird ein Java Development Kit (JDK) im Rahmen der Java Plattform 2 Standard Edition (J2SE) benötigt.

Für die Experimente wurde zusätzlich auf zwei für YALE 3.2 verfügbare PlugIns zurückgegriffen, namentlich das Clustering-PlugIn [97] und das Word-Vektor-PlugIn [99]. Die Einbindung beider PlugIns ist in [96] beschrieben und die Nutzung in Verbindung mit der grafischen Benutzeroberfläche von YALE in [95]. Auch diese PlugIns sind im Source und mit Dokumentationen und Beispielen unter der GPL verfügbar.

Die implementierten Verfahren dieser Arbeit wurden in einem eigenen PlugIn für YALE 3.3, dem sogenannten Outlier-PlugIn, umgesetzt und sind ebenfalls in Java entwickelt und unter der GPL verfügbar. Dieses PlugIn realisiert eine Reihe von Operatoren, welche sich in YALE Experimente einbinden lassen. Das PlugIn selbst ist ausführlich in [105] beschrieben. Zusammen mit dem PlugIn (in der Java .jar Archivdatei enthalten) wird ein kommentiertes Javadoc als Sourcedokumentation geliefert und die Sourcen selbst sind ebenfalls im Auslieferungsumfang enthalten. Somit ergibt sich eine ausführliche Dokumentation aller Klassen, Schnittstellen und Funktionen, welche dem interessierten Leser zur Weiterentwicklung bzw. auch zur informierten Anwendung in eigenen Experimenten dienen können.

Für die Umsetzung der referenzierten ESOM Verfahren ([89], [90] und [91]) wurden direkt die unter der GPL verfügbaren ESOM Tools [87] des Lehrstuhls für Datenbionik am Fachbereich Mathematik der Universität Marburg eingesetzt. Das Outlier-PlugIn unterstützt einen entsprechenden Datenexport zu diesen Tools. Die Anwendung der ESOM Tools selbst ist in der Literatur so ausführlich dokumentiert, dass hier darauf verzichtet wird, diese weiter auszuführen.

5.2. Implementierung der Verfahren

5.2.1. Operatoren-Testmenge

Für die Veranschaulichung der Arbeitsweise der implementierten Operatoren wird im Folgenden eine Testmenge mit grafischer Repräsentation vorgestellt. Diese ist in Abbildung 39 gezeigt. Deutlich ist zu sehen, dass diese aus drei Clustern besteht, zu denen jeweils mehrere Outlier positioniert sind, wobei manche Outlier sich dicht an Clustern befinden (die wiederum eine hohe Dichte aufweisen), aber auch solche, welche näher oder weiter entfernt zu Clustern mit geringerer Dichte liegen. Die Ergebnisse der Anwendung der beschriebenen Outlier Verfahren soll anhand dieser Testmenge kurz deutlich gemacht werden, da dies besonders anschaulich möglich ist. Die durchschnittliche Entfernung zwischen den Objekten der Testmenge beträgt ca. 6,29. Die Varianz beträgt 30,12 und die Standardabweichung beträgt 5,49 (alles gerundete Werte). In der Testmenge befinden sich 8 intuitive Outlier.

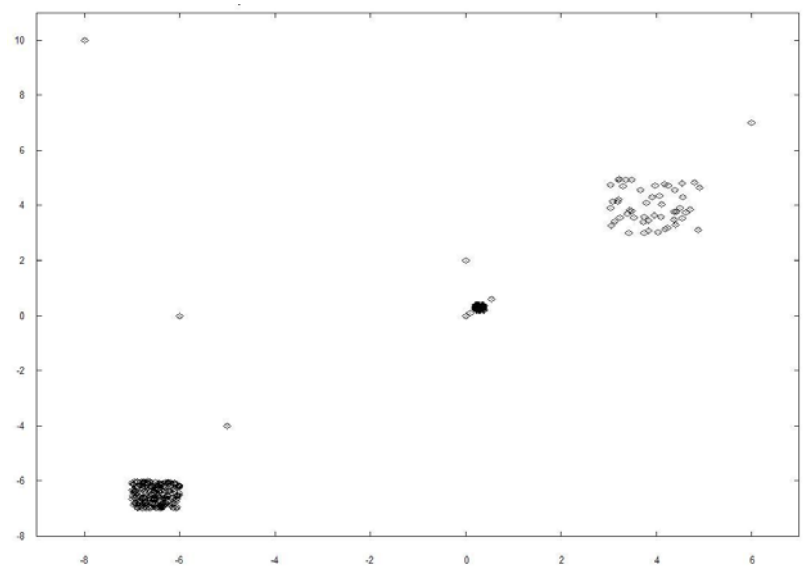


Abbildung 39 - Testmenge für Outlier Operatoren

Abbildungsbeschreibung: Für das Testen der Outlier Operatoren wurde eine intuitive Testmenge erstellt, welche Cluster verschiedener Dichten und Größen sowie 8 Outlierkandidaten enthält. Durch die spätere Anwendung der Verfahren auf diese Testmenge werden auch Unterschiede in den Ansätzen deutlich.

Diese Outlier gruppieren sich jeweils um drei die Datenmenge maßgeblich bestimmende Cluster. Bei der Ausgestaltung dieser Cluster wurde darauf geachtet, unterschiedliche Größen und Dichten zu realisieren. Der mittlere Cluster sehr kleiner Größe hat eine sehr hohe Dichte, die sich durch die Anzahl von 100 Elementen ergibt. Zu diesem Cluster sind in direkter Nähe drei Outlierkandidaten positioniert, wobei zwei davon sehr nahe aber doch deutlich abgegrenzt liegen und der dritte vergleichsweise weit entfernt. Der Cluster links unten ist groß und hat eine hohe bis mittlere Dichte mit 200 Elementen. Zu ihm sind zwei Outlier direkt positioniert. Der Cluster rechts oben ist sehr groß und hat eine geringe Dichte mit 50 Elementen. Ihm ist ein Outlier zugeordnet, wobei die Ränder des Clusters stark ausgefranst sind und damit auch Randobjekte als Outlier in Frage kommen. Zwei Outlier im linken Bereich (Mitte und oben) sind als Outlierkandidaten zu allen Clustern positioniert.

5.2.2. *DB(p,D)*-Outlier Operator

Der *DB(p,D)*-Outlier Operator setzt den entfernungsbasierten Outlier-Erkennungsansatz nach Knorr und Ng [3] um. Im Rahmen des Outlier-PlugIns [105] ist er wie die anderen Operatoren des PlugIns über die Outlier-Subgruppe im Operatoren-Auswahlzweig verfügbar. Der Anwender muss lediglich die Parameter *p* und *D* entsprechend dem Ansatz angeben. Gleichzeitig besteht die Möglichkeit, die Gesamtdatenmenge untersuchen zu lassen, um die statistischen Verteilungsdaten (Mittelwert, Standardabweichung, Varianz) errechnen zu lassen. Bei einer erneuten Anwendung des Operators können diese Werte benutzt werden, um die Wahl für *p* und *D* zu optimieren bzw. nach anderen Gesichtspunkten anzupassen.

Wie alle anderen Operatoren auch, empfängt der *DB(p,D)*-Operator in der Operatorenkette eines YALE Experiments eine Beispielmenge (ExampleSet). Er überträgt diese in einen eigenen Suchraum (SearchRoom) und ordnet die Beispiele (Examples) als Suchobjekte (SearchObject) in den Suchraum ein. Die Suchobjekte selbst verfügen über Felder und Methoden zur Bereitstellung relevanter Informationen (Position im Raum, Statusinformationen, usw.), sind aber auch in der Lage, z.B. ihre Entfernung zu anderen Objekten zu errechnen. Der Suchraum wiederum ist in der Lage, mit Hilfe der Objekte und eigenen Methoden die relevanten Erkennungsverfahren – hier die distanzbasierte Outlieridentifikation – durchzuführen und die Ergebnisse in den Suchobjekten zu speichern. Der Operator wiederum überträgt die Ergebnisse der Outlier-Erkennung in das ursprüngliche ExampleSet und ergänzt dieses durch ein Prediction-Label, welches den Outlier-Status enthält. Dabei steht „1.0“ für Outlier und „0.0“ für Average, also Nicht-Outlier.

Die Implementierung setzt einen BruteForce Radius-Search für die Identifizierung der Outlier ein. Zwar stellen die Autoren des Ansatzes in [8] mit einem Zellenalgorithmus Optimierungsalternativen zur schlichten Radiensuche vor, welche auch in der Beschreibung des Ansatzes in dieser Arbeit referenziert werden. Dieser Algorithmus eignet sich jedoch nicht für hochdimensionale Räume, da die Anzahl der Zellen mit der Anzahl der Dimensionen exponentiell steigt und damit der Verwaltungsaufwand für die Zellen die Optimierung mit hoher

Wahrscheinlichkeit spätestens dann aufwiegt, sobald die Zellenzahl die Zahl der Objekte in der Datenmenge übersteigt. Daher wurde auf eine Umsetzung niedrigdimensionaler Optimierungen aus Praktikabilitätsgründen verzichtet. Der umgesetzte Algorithmus hat eine Komplexität von $O(m \cdot n^2)$, wobei m die Anzahl der Dimensionen ist (welche Einfluss auf die Berechnung des Abstands als Betrag der Differenz der Vektoren hat) und n die Anzahl an Objekten in der Datenmenge.

Auf die Testdatenmenge angewendet (siehe Abbildung 40), ergibt sich für diesen Operator das folgende Ergebnis, wenn die Standardabweichung der Datenmenge als Parameter D = 5,49 festgelegt wird. Versuche haben gezeigt, dass die beste Qualität in Bezug auf die zu erwartenden Outlier mit einem Wert p = 85,87% erreicht wird. Mit höherem Anteil p werden nur noch zwei von acht Outliern erkannt, mit niedrigerem Parameter p werden zu viele Objekte des Clusters mit geringer Dichte als Outlier falsch erkannt. Dies zeigt, das stark unterschiedliche Dichten von Clustern bei einer globalen Betrachtung von Datenmengen in Bezug auf die Entfernung der Objekte voneinander schnell zu verfälschten Ergebnissen führen könnten. Allerdings ist auch festzustellen, dass drei besonders starke Outlier sofort und mit einer Vielzahl an Wertekombinationen von p und D sicher erkannt werden.

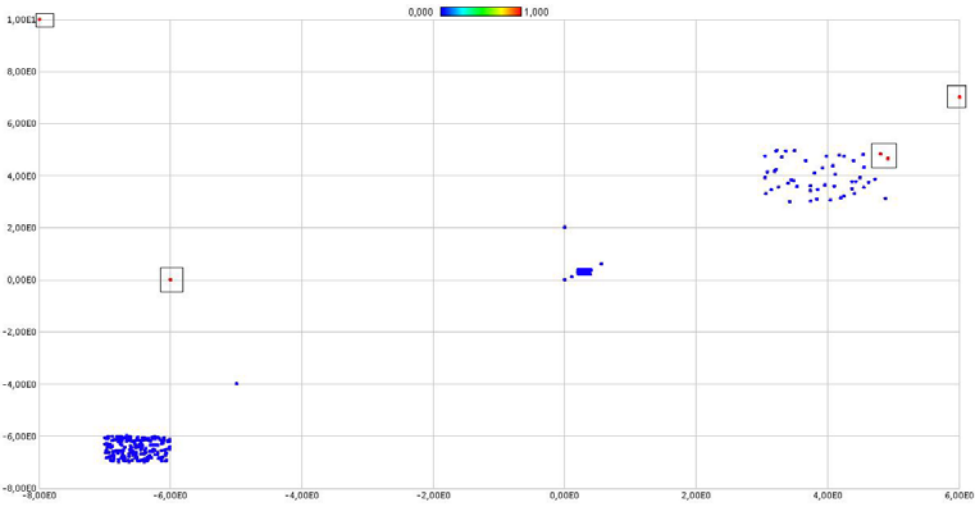


Abbildung 40 - Anwendung von *DB(p,D)*-Outlierverfahren auf die Testmenge

Abbildungsbeschreibung: Das *DB(p,D)*-Verfahren erkennt Outlier mit hohen Entfernungen zu den Clustern. Solche Outlierkandidaten, welche sich zu nah an Clustern befinden, werden jedoch nicht erkannt und manche Clusterobjekte irrtümlich als Outlier klassifiziert.

In der Abbildung sind die erkannten Outlier durch Eingrenzung mit Rechtecken noch einmal gesondert gekennzeichnet worden. Zum Vergleich zur Ursprungsmenge siehe Abbildung 39. Die grafische Darstellung wurde direkt aus YALE heraus erstellt, wobei hier die Outlier durch einen binären Farbwert dargestellt werden. Für Datenmengen mit mehr als zwei Dimensionen kann eine grafische Reduktion der Ursprungsmenge auf z.B. zwei Dimensionen erfolgen. Das Outlier-PlugIn bietet hier ebenso wie das YALE Clustering PlugIn [97] eine Reduktution über einen Operator zur Singular Value Decomposition an.

5.2.3. *D(k,n)*-Outlier Operator

Der *D(k,n)*-Outlier Operator implementiert die Identifizierung von Outliern durch die Betrachtung der Entfernung zu den *k*-ten Nachbarn von Objekten nach der von Ramaswamy, Rastogi und Shim [13] vorgeschlagenen Methode. Hierbei werden die top-*n* Outlier dadurch gefunden, dass von allen Objekten die Entfernung derselben zu ihren *k*-ten Nachbarn verglichen wird. Unter der Annahme, dass Objekte mit einer vergleichsweise großen Entfernung zu ihren *k*-ten Nachbarn auch eine zu erwartende spärlich besetzte Nachbarschaft haben, kommen diese unter Bezug auf ein statistisches Entfernungsmaß als Outlierkandidaten in Betracht. Als top-*n* Outlier werden sodann die *n* Objekte aus der Gesamtmenge mit den größten Entfernungen zu ihren *k*-ten Nachbarn erkannt. Die Gesamtmenge X soll zur Unterscheidbarkeit n_X Elemente enthalten

Der *D(k,n)*-Operator nutzt die bereits für den *DB(p,D)*-Operator beschriebenen PlugIn-Funktionen. Die Implementierung des Operators ist durch einen Mehrschritt-Algorithmus gekennzeichnet. Im ersten Schritt werden die Nachbarn aller Objekte identifiziert und entsprechend ihrer Entfernungen in Objektcontainer

eingeordnet. Objekte mit gleicher Distanz zum untersuchten Objekt werden in denselben Container eingeordnet. Dieser Schritt des Algorithmus hat eine Komplexität von $O(m \cdot n_X^{\,2})$, da für jedes Objekt alle anderen Objekte als entfernungsabhängige Nachbarschaften geordnet werden müssen, somit pro Objekt eine sequentielle Untersuchung aller Objekte notwendig wird. Der Parameter *m* geht als die Anzahl der Dimensionen in den Rechenaufwand ein, da der Aufwand der Entfernungsberechnung bei Verwendung der euklidischen Distanz als Betrag des Abstandes zweier Vektoren mit Faktor *m* steigt. Ein einfacher möglicher Optimierungsansatz wäre die Berechnung aller Objektdistanzen inkl. der Speicherung dieser und Zugriff auf eine Distanztabelle. Dies ist durch einen Ansatz mit

$$O(m \cdot (n_X^{\,2} / 2 - n_X))$$

möglich, da die Matrix aller Distanzen an der Diagonalen gespiegelt ist und die Distanz eines Objektes zu sich selbst Null ist. Allerdings verändert diese Optimierung nicht den exponentiellen Charakter der Komplexität. Die Möglichkeit der Identifikation von Nachbarschaften durch die Anwendung spezieller Indizierungsformen (R*-Tree, X-Tree) ist eine Option für die Optimierung von Anwendungen mit niedriger Dimensionszahl, allerdings verlieren die Indizierungsverfahren schnell an Effizienz, wenn die Zahl der Dimensionen deutlich über 5 bis 10 Dimensionen ansteigt (vgl. auch [4]). Da dies für den Fall der praktischen Anwendung von USENET News gegeben ist, wurde auf eine indexbasierte Optimierung verzichtet.

Im zweiten Schritt werden für jedes Objekt die *k*-Distanzwerte durch Abzählung der in den Distanzcontainern enthaltenen Objekte entsprechend der in [4] von Breuning & Co. beschriebenen Bedingungen zur Bestimmung von *k*-distance Werten ermittelt. Dadurch sind in den Containern bereits kaskadierend auch die Objekte der *k*-Nachbarschaften für steigende k-Werte enthalten. Im Fall des *D(k,n)*-Outlier Operators ist jedoch nur die Distanz zum *k*-ten Nachbarn selbst von Interesse. Diese wird im Objekt selbst gespeichert. Der Anwendungsschritt hat eine Komplexität von $O(k \cdot n_X)$, da nur die Distanzen bis *k* von Interesse sind.

Im letzten Schritt werden nun die Distanzen aller Objekte zu ihren *k*-ten Nachbarn miteinander verglichen und die *n* Objekte mit den größten Distanzen werden als Outlier erkannt. Dieser Schritt, welcher eine Komplexität von $O(n_X)$ hat, ließe sich auch noch zur Steigerung der Effizienz mit dem vorherigen Schritt verbinden, allerdings ist der Performance-Gewinn von der Größenordnung her nicht ausschlaggebend. Die Gesamtkomplexität liegt also bei $O(mn_X^{\,2} + (k+1)n_X)$.

Die Autoren des Outlier Ansatzes [13] schlagen zusätzlich noch einen anderen Algorithmus vor. Dieser basiert auf einer Partitionierung der Datenmenge und erfordert als ersten Schritt ein Clustering mittels des von Zhang, Ramakrishnan und Livny vorgeschlagenen Verfahrens BIRCH [60]. Im Weiteren wird dann wieder auf eine Indizierung mittels R*-Tree gesetzt, um die Performance zu steigern. Dieser Algorithmus wurde aber für die praktischen Tests dieser Arbeit nicht umgesetzt, weil es fraglich erscheint, dass ein Clustering auf einem zu erwartenden spärlich besetzten Datenraum zu einer effizienten Partitionierung führt und damit Effizienzgewinne wirklich realisiert.

Angewendet auf die eingeführte Testdatenmenge ergibt sich das in Abbildung 41 gezeigte Ergebnis. Der Wert für *k* wurde auf 5 gesetzt (ein von den Autoren des Ansatzes als guter Ausgangspunkt beschriebener Wert). Da in der Testdatenmenge 8 intuitive Outlierkandidaten enthalten sind, wurde *n* auf 10 gesetzt, um zu sehen, welche top-10 Outlier das Verfahren identifiziert. Hierbei wird deutlich, dass der Ansatz zu erwartende Outlier gut identifiziert, wobei nicht mehrfach mit den Parametern experimentiert werden muss, wie bei dem auf einem vergleichbaren statistischen Maß aufsetzenden *DB(p,D)*-Outlierverfahren. Es ist aber auch ein Schwachpunkt zu erkennen: Weicht die Entfernung von Objekten von Cluster zu Cluster durch die unterschiedliche Dichte der Cluster so weit ab, dass sie die Entfernung von Outlier-Objekten zu dichten Clustern übersteigt, werden automatisch erst Objekte spärlicher Cluster als Outlier-Kandidaten identifiziert. Die drei intuitiven Outlier nahe dem sehr kleinen dichten Cluster in der Mitte der Testmenge werden vom Verfahren auch dann nicht erkannt, wenn *n* auf einen sehr hohen Wert gesetzt wird (z.B. 60). Ursache dafür ist die globale Anwendung des Entfernungsmaßes im Sinne des absoluten Vergleiches der Entfernungswerte. Zwar ergibt sich durch die Betrachtung zum *k*-ten Nachbarn eine gewisse Lokalität, diese fließt aber nur begrenzt in die finale top-*n* Auswahl ein.

Eine Vergrößerung des *k*-Wertes erbrachte keine wesentlichen Änderungen der Erkennung. In der Abbildung sind die erkannten Outlier durch Eingrenzung mit Rechtecken noch einmal gesondert gekennzeichnet worden. Zum Vergleich zur Ursprungsmenge siehe Abbildung 39. Die grafische Darstellung wurde direkt aus YALE heraus erstellt, wobei hier die Outlier durch einen binären Farbwert dargestellt werden.

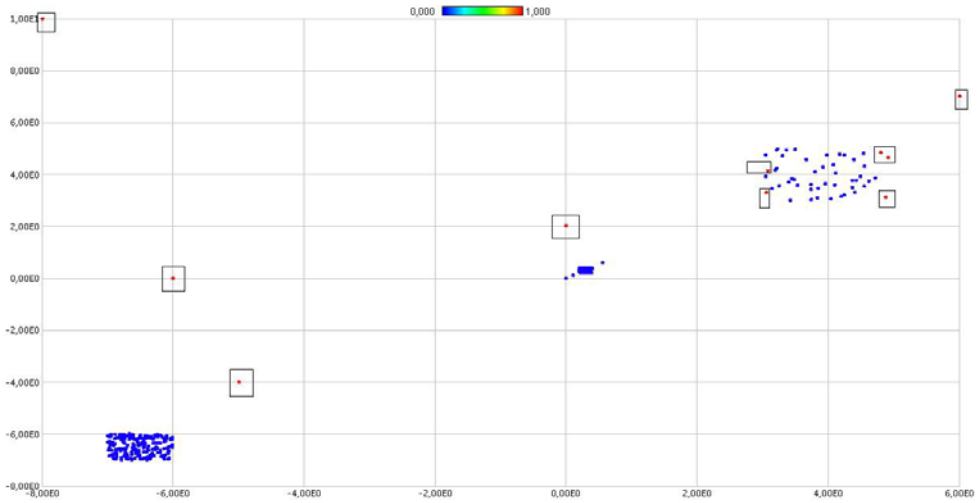


Abbildung 41 - Outlier der Testdatenmenge nach dem *D(k,n)*-Verfahren

Abbildungsbeschreibung: Das *D(k,n)*-Verfahren erkennt zuverlässig Outlier mit großen Entfernungen zu Nachbarschaften, die durch Cluster gekennzeichnet sind. Allerdings zeigt sich, dass bevor Outlier in sehr lokalen Nachbarschaften erkannt werden, erst alle Objekte von Clustern mit geringerer Dichte (und damit höherem Abstand) fälschlich als Outlier markiert werden.

5.2.4. *LOF(MinPts)*-Outlier Operator

Der *LOF(MinPts)*-Outlier Operator setzt das bekannte, von Breuning, Kriegel, Sandner und Ng [4] vorgestellte Verfahren der dichtebasierten Outliererkennung um. Das Verfahren selbst ist im theoretischen Teil dieser Arbeit ausführlich beschrieben, sodass hier nur anzumerken ist, dass es lokale Dichten innerhalb der Datenmenge misst und diese so in Relation zueinander setzt, dass allen Objekten ein lokaler Faktor des Grades, als Outlier verdächtig zu sein, zugeordnet wird. Damit rückt dieses Verfahren von der klassischen Ja/Nein Einordnung in Outlier und Nicht-Outlier ab. Dem Anwender ist im Ergebnis dann überlassen, die *LOF*-Werte der Objekte zu interpretieren. Dabei ist vor allem das Verhältnis dieser Werte zueinander interessant, wobei sich Erkenntnisse über Teilmengenbeziehungen innerhalb der Gesamtdatenmenge ableiten lassen. Dies wird im Testbeispiel später gezeigt.

Das Verfahren stellt nur auf einen einzigen Parameter *MinPts* ab. Dieser hat die Funktion, das Verfahren anzuweisen, ein statistisches Dichtemaß basierend auf der *MinPts*-Nachbarschaft für die Berechnung der Outliergrade zu analysieren. Der Operator selbst kann mit einem *MinPts*-Intervall verwendet werden, wobei die Autoren Verfahren zur Bestimmung sinnvoller *MinPts*-Werte und *MinPts*-Intervallgrenzen vorschlagen. Dabei spielen zwei Überlegungen eine Rolle. Die absolute untere Grenze für *MinPts* sollte bei 10 liegen, da erst ab diesem Wert die statistischen Schwankungen in der Datenmenge soweit nachlassen, dass sinnvolle *LOF*-Werte gefunden werden können. Im Weiteren wird für die Anwendung eines oder mehrerer *MinPts*-Intervalle empfohlen, die Datenmenge bzgl. einer Clusterstruktur vorher zu untersuchen. Nähere Angaben sind in Kapitel 2.5.1 ausgeführt.

Wie alle anderen Operatoren auch, empfängt der *LOF(MinPts)*-Operator in der Operatorenkette eines YALE Experiments eine Beispielmenge (ExampleSet). Er überträgt diese in einen eigenen Suchraum (SearchRoom) und ordnet die Beispiele (Examples) als Suchobjekte (SearchObject) in den Suchraum ein, welcher die Berechnung der *LOF*-Werte pro Suchobjekt durchführt. Der Operator wiederum überträgt die Ergebnisse der Outlier-Erkennung in das ursprüngliche ExampleSet und ergänzt dieses durch ein Prediction-Label, welches den Outlier-Grad, also den *LOF*-Wert enthält. Dabei wird nicht zwischen Outlier und Nicht-Outlier unterschieden. Allerdings weisen die Autoren des Verfahrens in [4] auch nach, dass Objekte innerhalb eines Clusters einen Outlierwert um 1 haben. Die Anwendung der vorliegenden Implementierung hat gezeigt, dass die *LOF*-Werte von Objekten tief innerhalb eines Clusters im Intervall [0;1] liegen, meist jedoch einen sehr kleinen Wert näher an der unteren als an der oberen Intervallgrenze aufweisen. Ob dies am Testbeispiel oder an der Implementierung liegt, ist derzeit nicht bekannt. Diese liefert jedoch für Outlier im Vergleich zu Nicht-Outliern sehr plausible Werte. Starke Outlier haben entsprechend hohe bis sehr hohe *LOF*-Werte.

Die Implementierung bedient sich des schon in Kapitel 5.2.3 vorgestellten ersten Schrittes eines Algorithmus, der mit Komplexität $O(m \cdot n^2)$ die Objekte in Distanzcontainer einordnet. Dies entspricht der Empfehlung in [4], für hochdimensionale Datenmengen nicht auf eine Implementierung mittels X-tree [53] zu setzen, sondern ein VA-File [54] oder einen sequentiellen Scan zu benutzen, wobei die vorliegende Umsetzung letzteren benutzt. Sowohl die Kardinalitäten der Objektmengen in den Containern, als auch die Distanzen werden für die spätere Verwendung ebenfalls in den Containerobjekten gespeichert. Im zweiten Schritt werden für die Objekte entsprechend die *k*-Distanzen für alle $k \leq MinPtsUB$ in dem Objekt selbst gespeichert, was einen Rechenaufwand von $O(MinPtsUB \cdot n)$ hat, wobei *MinPtsUB* die obere Intervallgrenze für *MinPts* für den Operator ist und *MinPtsLB* die untere. Danach werden für alle Objekte die Erreichbarkeitsdistanzen für die Objekte in den *k*-Nachbarschaften errechnet und im gleichen Moment daraus auch die lokalen Erreichbarkeitsdichten ermittelt. Die Komplexität dieses Schritts beträgt ebenfalls $O(MinPtsUB \cdot n)$. Im letzten Schritt werden die *LOF*-Werte für alle Objekte innerhalb des *MinPts*-Intervalls errechnet, jedes Objekt erhält einen *LOF*-Wert für jeden *MinPts*-Wert innerhalb des Intervalls. Letztendlich werden die maximalen *LOF*-Werte jedes Objektes innerhalb dieses Intervalls als der finale *LOF*-Parameter der Objekte vom Operator zurück in das ExampleSet geschrieben. Dieser letzte Schritt braucht zusammengefasst $O((MinPtsUB - MinPtsLB) \cdot n)$. Wenn für den Worst-Case davon ausgegangen wird, dass die untere Grenze von *MinPtsLB* auf 1 gesetzt wird, ergibt sich für den Algorithmus ein Gesamtrechenaufwand von $O(mn^2 + 3 \cdot MinPtsUB \cdot n)$. Es sei allerdings erwähnt, dass bei Räumen mit extrem vielen Dimensionen für Fälle, in denen $m \approx n$ gilt, der Aufwand auf eine Größenordnung von vereinfacht $O(n^3 + cn)$ ansteigen kann.

Angewendet auf die Testdatenmenge ergibt sich das in Abbildung 42 gezeigte Ergebnis. Dabei wird deutlich, dass besonders starke Outlier der Testmenge sehr gut erkannt werden, aber auch solche Outlier nahe an dichten Clustern in Relation zu diesen noch identifiziert werden können. Dies zeigt der vergrößerte Ausschnitt in der Abbildung, wobei die durch ein Rechteck eingegrenzte Objektmenge in der Vergrößerung selbst der sehr kleine sehr dichte Cluster mit 100 Objekten ist. Hier werden die Vorteile der lokalen Betrachtung deutlich, denn nur von diesem Verfahren werden die drei Outlier in der Nähe des kleinen Clusters erkannt, sofern die Clusterumgebung selbst untersucht und die *LOF*-Werte in Relation gesetzt werden. Von Jin, Tung und Han [5] wird noch eine Methode zur schnelleren Erkennung der top-*n* Outlier vorgestellt, die wiederum ein Vorabclusterung mittels BIRCH [60] erfordert. Sie ist mit derselben Begründung wie in Kapitel 5.2.3 hier nicht umgesetzt worden. Auch stände mglw. zu erwarten, dass die o.g. Outlier nahe dem kleinen dichten Cluster nicht unter den top-*n* Outliern sein werden, da die Objekte des Clusters mit geringer Dichte ähnlich hohe *LOF*-Werte haben, wie die Outlierkandidaten. Daher ist ein lokaler Vergleich empfehlenswert, für den sich z.B. paritätische Verfahren für das Postprocessing anbieten würden (vgl. Kapitel 2.7).

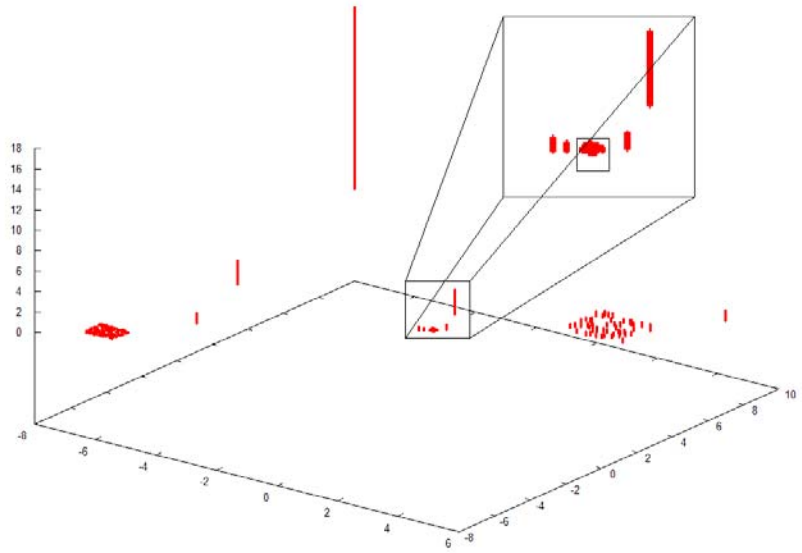


Abbildung 42 - *LOF*-Werte für die Testdatenmenge / dichtebasierte Outlier

Abbildungsbeschreibung: Das *LOF*-Verfahren erkennt auch solche Outlier, die intuitiv nur mit lokalem Bezug zu erkennen wären. Auf der *z*-Achse ist in der Abbildung der *LOF* jedes Objektes durch die Höhe des Balkens abgetragen.

In der Abbildung sind die Outlier-Faktoren durch die Höhe des Balkens gekennzeichnet worden. Zum Vergleich zur Ursprungsmenge siehe Abbildung 39. Die grafische Darstellung wurde aus YALE heraus an das Programm GNUplot exportiert. Unter UNIX Betriebssystemen besteht die Möglichkeit einer nahtlosen Integration von GNUplot in YALE. Es besteht auch die Möglichkeit, die *LOF*-Werte farblich in einer zweidimensionalen Darstellung mit dem in YALE integrierten Plotter anzuzeigen, wobei der Farbverlauf der Punkte den Outlier-Grad visualisiert. Dies ist jedoch nur sinnvoll, wenn der Farbverlauf in etwa gleichmäßig verläuft. Ab Version 3.3 unterstützt YALE auch die dreidimensionale Darstellung selbst.

5.2.5. ESOM-Export Operator

Im Prinzip stellt dieser Operator lediglich eine Anbindung der Lernumgebung YALE an die Databionik ESOM Tools [87] her. Dies geschieht durch den Export eines YALE ExampleSets (welches z.B. durch Vektorisierung von Textdateien gewonnen wird) in das ESOM Dateiformat durch den ESOMFileSetWriter Operator des Outlier PlugIns in YALE. Damit stehen die ESOM Tools für eine Auswertung der Datenmenge durch emergente selbstorganisierende Karten zur Verfügung. Angewendet auf die so exportierte Datenmenge zeigt Abbildung 43 die Testmenge in der ESOM Analyse mittels einer emergenten SOM von 52 x 80 Neuronen (mit 20 Lernzyklen).

Abbildung 43 - Testmengenanalyse durch ESOM Tools

Abbildungsbeschreibung: Die drei Cluster sind durch die Wasserbereiche gut visualisiert, wobei auch die geringe Dichte des dritten Clusters unterscheidbar ist. Die Outlier befinden sich an den Hängen von Bergmassiven bzw. sind gänzlich von diesen eingeschlossen.

Wie der Abbildung, welche eine U*Matrix und das Neuronengitter zeigt, deutlich zu entnehmen ist, sind die drei Cluster sehr schön als Wasserflächen visualisiert. Gemein ist allen Anwendungen einer SOM, dass sie die wesentlichen Informationen der Datenverteilung auf das Gitter an Neuronen durch einen konvergierenden Lernprozess projiziert. Durch die U*Matrix, welche Dichte- und Entfernungswerte der Ursprungsobjekte zueinander dem Neuronengitter grafisch hinterlegt, werden sehr schnell wichtige Informationen deutlich: Gebiete mit geringer Höhe (hier als Wasserflächen intuitiv dargestellt) zeigen Gruppierungen von nahe aneinanderliegenden Objekten. Gut zu sehen ist das Seegebiet links, welches den Cluster der Testmenge in Abbildung 39 links unten mit 200 Objekten zeigt. Dieses ist vom mittleren Seegebiet in der rechten Hälfte durch ein hohes und ausgeprägtes Massiv getrennt, welches Gebiete mit sehr geringer Dichte an Daten darstellt. Das zweite Seegebiet visualisiert den mittleren kleinen Cluster mit hoher Dichte (200 Objekte). Es ist rechts durch ein weiteres Massiv begrenzt, dessen höchste Ausprägung sich unten rechts im Bild zeigt. Im rechten oberen Rand der Karte liegt jenseits des zweiten Massivs der dritte Cluster mit geringer Dichte (50 Objekte).

Die acht Outlier sind hier aufgrund der verfügbaren Information der Ursprungsmenge gekennzeichnet worden. Als Outlierkandidaten können solche Objekte erkannt werden, welche sich entweder sehr nahe an Bergmassiven befinden, direkt auf Bergmassiven liegen oder ganz von solchen eingeschlossen sind. Letztere Form zeigt die wohl stärksten Outlier, welche zur Gesamtdatenmenge die größten Abweichungen zeigen. Die vorher genannten Formen zeigen Outlier, welche Bezüge zu Clustern zu haben scheinen.

Es sollte also im Fazit bei der Analyse einer ESOM nach Neuronen gesucht werden, welche sich an Hängen von Massiven befinden bzw. die von solchen eingeschlossen sind. Von Ihrer Ausprägung her sind die ESOM Tools darauf ausgerichtet, z.B. Clustering von hochdimensionalen Datenmengen durch U*C Clustering [90] oder U*F Clustering [91] zu erlauben. Die Outlierbehandlung läuft hier darauf hinaus, Outliereffekte durch Beschneidungen des Effektnivaus zu begrenzen, um Clusteringstrukturen grafisch besser herausarbeiten zu können. Auf der anderen Seite sind die ESOM Tools ein sehr wichtiges Werkzeug zur Analyse hochdimensionaler Datenstrukturen, welche es erlauben, diese grafisch zu interpretieren. Dies ist sonst bei Datenmengen mit mehr als 3 Dimensionen äußerst schwierig.

Ab Version 3.3 unterstützt YALE auch die Darstellung von SOM direkt.

5.2.6. OutlierDimensionReduction Operator

Der Operator zur Reduzierung der Dimensionszahl basiert auf einem Clone des SVDReduction Operators der für YALE 3.0 gültigen Version des Clustering-Plugins [97] für YALE. Dieser setzt eine Singular Value Decomposition [104] um und ermöglicht so die Reduktion der Anzahl der Dimensionen im Suchraum von einigen hundert bis tausend auf normalerweise zwei bis fünf Dimensionen. Dabei bleiben die Verteilungsgrundsätze der Objekte im Raum in etwa für die niedrigdimensionale Projektion erhalten.

Die Implementierung des Clones wurde im Gegensatz zum Original des SVDReduction Operators lediglich um die Funktionalität erweitert, Prediction-Label in YALE an den nächsten Operator durchzureichen. Es besteht für das Clustering PlugIn in der Version für YALE 3.1 ein SVDReduction Operator, der diese Möglichkeit auch bietet. Der OutlierDimReduction Operator wird aber dennoch angeboten, falls das Clustering PlugIn nicht zur Verfügung stehen sollte. Sofern es vorhanden ist, bietet der Original SVDReduction Operator für YALE 3.1 eine noch weiter verbesserte Laufzeit-Implementierung der Singular Value Decomposition als der Operator für Version 3.0 bzw. der Clone des Outlier-PlugIns an.

In den praktischen Experimenten wird dieser Operator zur Reduktion der Dimensionen vor der Anwendung klassischer niedrigdimensionaler Verfahren zur Outlier-Identifizierung genauso angewendet, wie zur ggf. notwendigen Reduktion der Dimensionen für eine grafische Visualisierung des Endergebnisses, wobei im letzteren Fall die Outlier-Erkennung selbst auf der Ursprungsmenge stattfindet. Für Performance und Komplexität des Operators sei auf die Ausführungen zur Ursprungsimplementierung in [97] verwiesen.

5.2.7. Textsplitting / NewsArticleSplitter Operator

Um eine Optimierung der Identifikationsergebnisse für Outlier durch das Reduzieren der Originaltextmenge zu unterstützen, wurde ein Operator für YALE im Outlier Plugin implementiert, der es dem Anwender erlaubt, den Header eines Newsartikels abzuschneiden und lediglich den reinen Textkörper (Body) zu behalten. Gleichzeitig kann der Anwender eine Reihe von Header-Feldern definieren, für welche die Header-Zeilen übernommen werden. So ist es zum Beispiel möglich, nur die „Subject“ oder „From“ Zeile im Header zu übernehmen und alle anderen Headerinformation auszublenden. Der Operator setzt dies mit einer entsprechenden Abfrage nach einem Verzeichnis mit Newsartikeln als Textfiles um und erstellt reduzierte Kopien jedes Artikels in einem entsprechenden Unterverzeichnis.

5.2.8. Implementierung unterschiedlicher Abstandsmaße

Um die in Kapitel 4.6 vorgeschlagenen unterschiedlichen Abstandsmaße durchgängig für alle Verfahren zu realisieren, welche solche Maße zur Beurteilungen von Abständen zwischen Objekten bzw. zur Einordnung in Nachbarschaften von Objekten oder zur Bestimmung lokaler Dichten verwenden, wurde der Implementierungskern mit einer flexiblen Distanzfunktion ausgestattet, welche den Abstand zwischen zwei Objekten nach Vorgabe eines Distanzmaßes bestimmt. Die restliche Verarbeitung ist dann für das Verfahren transparent. Zur Sicherstellung der korrekten Funktionalität sind derzeit folgende Abstandsmaße implementiert:

- Euklidische Distanz
- Kosinusdistanz (als Kosinus-Ähnlichkeitsmaß und als Kosinus-Abstandsmaß)
- Quadratische Distanz (der euklidischen Distanz)
- Winkelmaß (in Radiant)

Als potentielle Erweiterung empfiehlt sich die Implementation der Minkowski-Metric, welche zusätzliche Distanzmaße realisierbar macht. Auch ist eine Erweiterung auf andere Abstandsmaßgruppen (z.B. Korrelation,

Hamming Distanz, Jaccard Distanz oder Chebychev Distanz) ggf. sinnvoll, um die Anwendbarkeit des Outlier-PlugIns und seiner Operatoren in anderen Sachgebieten zu verbessern. Hierfür ist eine stärkere Kapselung der Distanzberechnung im Kern des PlugIns empfehlenswert [105].

Die Distanz, welche bei einem Verfahren und dem dazugehörigen Operator zum Einsatz kommt, lässt sich im grafischen Nutzerinterface von YALE durch eine Auswahl festlegen, wobei diese Information dann dem Implementierungskern übermittelt wird.

5.2.9. AuthorBackgroundKnowledgeApplier Operator

Eine mögliche Optimierung der Erkennung von Outliern bezogen auf die direkte Anwendungsdomaine der USENET Newsgruppen besteht in der Hinzuziehung von Hintergrundwissen über die Autoren von Newsartikeln. Dies ermöglicht der AuthorBackgroundKnowledgeApplier Operator im Outlier Plugin für YALE. Er ist in der Lage, drei Quellen von Informationen miteinander zu verbinden. Er übernimmt aus der Operatorenkette ein ExampleSet mit bereits vorhandenen Outlierinformationen pro Objekt. Gleichzeitig analysiert er die Newsartikel, aus denen die Vektorisierung für das ExampleSet erstellt worden ist und erstellt eine Liste von Autoren. Für die Autorenerhebung wird von den vom Operator genutzten Kernfunktionen lediglich die „From“-Zeile des Headers des Artikels analysiert. Auf die Konsistenzprobleme des „From/Sender“ Mechanismus und mögliche Unterschiede in der Darstellung der „From“-Zeile (vgl. Kapitel 4.1) sei daher ausdrücklich hingewiesen.

Der Operator liest für diese Autoren zusätzlich eine separate Autorenliste ein, welche Hintergrundwissen über den Autor enthält, das durch ein Gewicht repräsentiert, ob der Autor vornehmlich Outlier im Newssystem versendet. Der Operator ist nun in der Lage, die Gewichte der zu betrachtenden Objekte (Examples, welche die Artikel repräsentieren) und die Gewichte der Autoren auf unterschiedliche Art und Weise zusammenzuführen und in jeweils eine oder beide Richtungen auf die Quellen anzuwenden. Im Folgenden werden Variablen eingeführt, deren Bedeutung von den Definitionen in Kapitel 2.1 abweicht.

Sei $A = \{a_i \mid i = 1,...k\}$ die Menge von *k* Autoren der Menge $N = \{n_j \mid j = 1,...m\}$ von *m* Newsartikeln und sei $N_i = \{n_x \mid x = 1,...m \wedge a_i = Author(n_x)\}$ die Menge von Artikeln, welche vom *i*-ten Autor verfasst wurden. Dann sei w_{ai} das Gewicht des *i*-ten Autors und w_{nj} das Gewicht des *j*-ten Artikels.

Der Operator ist nun in der Lage, pro Autor dessen Gewicht mit den Gewichten der Artikel, die dieser Autor verfasst hat, zu verrechnen. Dazu bietet er die folgenden Verrechnungsverfahren an.

1. Addition der Gewichte: $w_{nj\_neu} = w_{nj} + w_{ai}; \forall n_j \in N_i$ und $w_{ai\_neu} = w_{ai} + \sum_{n_j \in N_i} w_{nj}$
2. Multiplikation der Gewichte: $w_{nj\_neu} = w_{nj} \cdot w_{ai}; \forall n_j \in N_i$ und $w_{ai\_neu} = w_{ai} \cdot \prod_{n_j \in N_i} w_{nj}$
3. Multiplikation der normierten Gewichte (wie Multiplikation, allerdings werden die Autorengewichte und die Artikelgewichte jeweils auf das Intervall [0,1] normiert).
4. Durchschnitt der Gewichte: $w_{nj\_neu} = \frac{w_{nj} + w_{ai}}{2}; \forall n_j \in N_i$ und $w_{ai\_neu} = (w_{ai} + \frac{\sum_{n_j \in N_i} w_{nj}}{|N_i|}) / 2$
5. Durchschnitt der Artikelgewichte: $w_{ai\_neu} = \frac{\sum_{n_j \in N_i} w_{nj}}{|N_i|}$
6. Logische UND bzw. ODER Verknüpfung: $w_{nj\_neu} = w_{nj} \wedge w_{ai}; \forall n_j \in N_i$ und $w_{ai\_neu} = w_{ai} \wedge w_{n1} \wedge ... \wedge w_{nm}; n_j \in N_i \mid j = 1,...m$, wobei die Gewichte hier jeweils Wahrheitswerte sind und die ODER Verknüpfung analog zur UND Verknüpfung realisiert ist.

Dabei ist die Anwendung der Zusammenführung auf die Quellinformationen unidirektional in jeweils beide Richtungen genauso möglich, wie bidirektional (also in beide Richtungen gleichzeitig). Da es eine 1:*n* Beziehung zwischen einem Autor und *n* von ihm verfassten Artikeln gibt, erfolgt die Gewichtsverrechnung in Richtung von Autor zu Artikeln immer 1:*n* und von den Artikeln in Richtung Autor immer *n*:1.

Mit dem Operator kann nun eine ganze Reihe von Verrechnungen an Gewichten je nach Natur der Quellinformationen und der Art der Gewichte durchgeführt werden und er lässt eine flexible Modellierung unterschiedlicher Anwendungen von repräsentiertem Hintergrundwissen zu.

5.2.10. LabelPredictionApplier Operator

Dieser simple Operator erlaubt es dem Anwender, für ein Example aus einem ExampleSet anhand von nutzerdefinierten positiven und negativen Labels einen bestimmten Wert für das PredictionLabel dieses Examples zu setzen. Dieser Operator ist vor allem hilfreich, um aus einer Vorkategorisierung mit bestimmten Labeln eine Besetzung der Outlierfaktoren anhand der Label vorzunehmen, also die Kategorisierung in eine direkte Outlierfaktorenbestimmung der Beispielmenge zu übernehmen. In einem zweiten Schritt kann mit dem AuthorBackgroundKnowledgeApplier Operator eben diese Information mit den Autoren der Newsartikelmenge, welche durch das ExampleSet beschrieben wird, in ein Autorenprofil eingebracht werden, welches später zur Bestimmung von Outliern oder als Hintergrundwissen verwendet werden kann. Die Verkettung der beiden Operatoren kann also zum Modellieren von geführtem Lernen von Autorenwissen anhand von Beispielen dienen.

5.2.11. OutlierPerformanceEvaluator zur Ergebnisauswertung

Für die Auswertung der Ergebnisse eines Outliererkennungsverfahrens bezüglich einer a priori vorgenommenen Kategorisierung wurde ein PerformanceEvaluator als YALE Operator im Outlier PlugIn umgesetzt. Dieser berechnet für eine vorgegebene Kategorisierung relevanter und nicht relevanter Labels im Vergleich mit dem durch das Verfahren ermittelten Outlier Status bzw. Outlier Faktor die Anzahl der korrekt identifizierten (TP – true positive), korrekt nicht identifizierten (TN – true negative), inkorrekt identifizierten (FP – false positive) und inkorrekt nicht identifizierten (FN – false negative) Elemente der Datenmenge. Dabei wird ein Schwellwertvergleich zum ermittelten Status bzw. Faktor vorgenommen.

$$TP = \left|\{label \subseteq relevant\_labels \wedge prediction \geq threshold\}\right|$$
$$TN = \left|\{label \subseteq irrelevant\_labels \wedge prediction < threshold\}\right|$$
$$FP = \left|\{label \subseteq irrelevant\_labels \wedge prediction \geq threshold\}\right|$$
$$FN = \left|\{label \subseteq relevant\_labels \wedge prediction < threshold\}\right|$$

Aufgrund dieser Werte wird nun die Genauigkeit (Precision), Vollständigkeit (Recall) und in negativer Weise die Güte des Verfahrens (Fall Out) berechnet. In der probabilistischen Interpretation (vgl. auch [106] und [107]) ist es auch möglich, die Maße als Wahrscheinlichkeit zu interpretieren:

- Recall ist die Wahrscheinlichkeit mit der ein (zufällig ausgewähltes) relevantes Dokument gefunden wird.
- Precision ist die Wahrscheinlichkeit mit der ein (zufällig ausgewähltes) gefundenes Dokument relevant ist.
- Fallout ist die Wahrscheinlichkeit mit der ein (zufällig ausgewähltes) irrelevantes Dokument gefunden wird.

Zusätzlich berechnet der Operator das gewichtete harmonische Mittel von Precision und Recall, das sog. F-Measure bzw. F_1-Maß, bei dem Precision und Recall gleich gewichtet werden.

Mittels dieser Standard-Performance-Maße lassen sich die Ergebnisse der Experimente der Operatoren entsprechend vergleichen. Gleichzeitig ermöglicht die flexible Gestaltung des Operators, dem über die Benutzerschnittstelle sowohl eine Zuordnung von Labels in die relevante bzw. nicht relevante Kategorienmengen sowie auch ein flexibler Schwellwert mitgeteilt werden kann, eine komfortable Auswertung eines Experiments. Für die Schwellwertanalyse kann der Operator mehrfach hintereinander geschaltet werden.

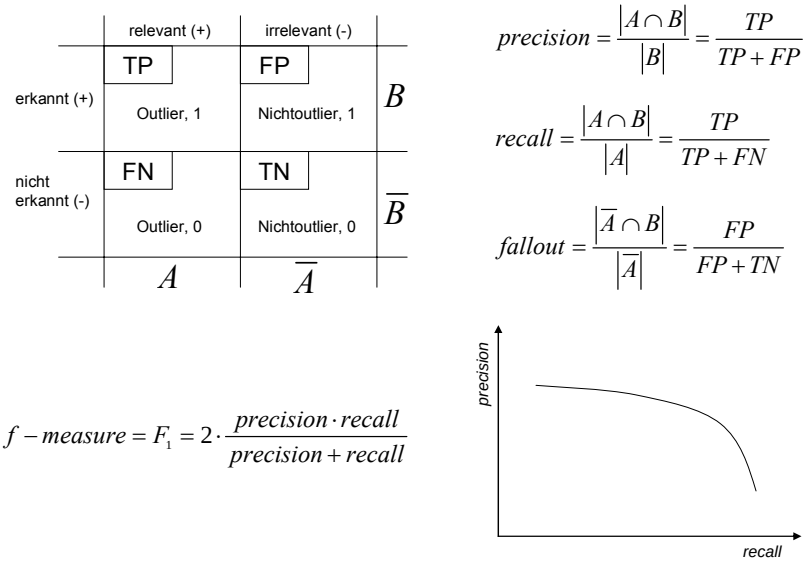


Abbildung 44 - Performance-Maße für Erkennung kategorisierter Objekte

Abbildungsbeschreibung: Für einen Vergleich der Performance eines Verfahrens zur Erkennung von Eigenschaften gegenüber einer Vorkategorisierung werden Standardmaße herangezogen, deren Ermittlung ein entsprechender YALE-Operator des Outlier-PlugIns direkt unterstützt.

5.3. Mögliche Verbesserungen und Entwicklerhinweise

Im Zusammenhang mit möglichen Verbesserungen sei bemerkt, dass die Implementierungen das Ziel hatten, eine experimentelle Umgebung zur Gewinnung von Erkenntnissen bei der Anwendung von Outlier-Verfahren anhand der speziellen Applikation der USENET News bereitzustellen. Diese ist natürlich weit von einem kommerziellen Produkt entfernt und verzichtet absichtlich auf eine nahtlose Integration, z.B. in einen Newsreader. Auf der anderen Seite wurden die Verfahren im Hinblick auf die zu erwartenden Testmengen umgesetzt, was einen Verzicht auf weitergehende Optimierungen der Algorithmen erklärt. Daraus ergeben sich direkte Erkenntnisse und Hinweise für eine potentielle Weiterentwicklung. Die Umsetzung zusätzlicher Outlier-Verfahren in YALE ist mit Sicherheit im Interesse der Nutzergemeinschaft dieser in der KI-Community anerkannten Lernumgebung. Ebenso kann die weitere Optimierung der bereits umgesetzten Verfahren, insbesondere durch die Implementierung von Algorithmen mit verbesserter Performance für niedrigdimensionale Suchräume, als Verbesserung oder Entwicklungsziel verstanden werden.

Umfassendes Potential bietet die Erweiterung der implementierten Verfahren um zusätzliche statistische Maße. Derzeit sind eine Reihe von Abstandsmaßen und ein Ähnlichkeitsmaß implementiert. Um die Verfahren auch für andere Anwendungsgebiete nutzbar zu machen, kann der Kern des PlugIns für YALE um zusätzliche statistische Maße ergänzt werden, welche sodann durch die YALE Operatoren des PlugIns mit miminalem ergänzendem Programmieraufwand bedient werden können.

Der Autorenoperator wertet derzeit nur „From“-Zeilen aus. Eine kombinierte Auswertung von „From“- und „Sender“-Zeilen würde die Standardkonformität des Operators erhöhen. Auch eine semantische Auswertung der Adressdarstellung selbst würde in Kombination mit vorgenannter Verbesserung eine noch tiefergehende Analyse der Autoren erlauben. Für das *DB(p,D)*-Verfahren ist die Unterstützung einer iterativen Parameteroptimierung für das Wertepaar *p* und *D* durch Hinzuschalten des OutlierPerformanceEvaluators in einem Schleifenoperator denkbar, wobei eine auf Precision, Recall oder F_Measure ausgerichtete Optimierung einer durch (zumindest teilweise) Vorkategorisierung gewonnenen Vergleichsbasis bedarf.

Eine tiefere Integration der gewonnenen Ergebnisse in die Anwendungen, welche die Untersuchungsmengen bereitstellen (in diesem Fall z.B. der Newsreader), ist jedoch ein relativ weitgehendes Ziel, welches spezifische Anpassungen notwendig machen würde. Alle genannten Verbesserungen befinden sich aber außerhalb des Fokus dieser Arbeit.

6. Evaluation: Experimente und Ergebnisse

6.1. Experimentelles SetUp

Für die Durchführung der Experimente wurde das in Abbildung 45 gezeigte Setup eingerichtet. Dabei wurde darauf geachtet, dass dieses einfach von einem anderen Anwender nachzuvollziehen ist und lediglich Standardkomponenten oder verfügbare Werkzeuge beinhaltet. Auch sollte es einer praktischen Anwendungsumgebung relativ nahe kommen.

Aus dem USENET News System heraus wird auf herkömmlichem Weg eine Newsgruppe über einen NNTP [81] Newsreader abonniert und ist damit im lesenden Modus des Readers verfügbar. Xnews wurde gewählt, weil dieser Reader (Freeware) eine bequeme Funktion zum Abspeichern aller Artikel einer Gruppe im Klartextformat als *.txt Dateien auf der Festplatte in einer Verzeichnisstruktur erlaubt, welche dann als Experimentierraum dienen kann. Somit lässt sich leicht ein Snapshot einer Gruppe erzeugen. Auch sichert das Programm die Artikel mit der Subject Zeile als Dateiname. Das WordVektor Plugin [99] von YALE [92] mit dem WVtool Operator übernimmt diese Dateinamen als Id's in das ExampleSet von YALE und diese können per MouseOver Funktion in der grafischen Datenmengenanzeige (YALE Plotter) gezeigt werden. Das dazugehörige Textfile kann nun per Doppelklick geöffnet werden. Dies ermöglicht eine gute Analyse der Ergebnisse von Outliertests mit zusätzlichem Zugriff auf die Ursprungsdatenmenge der Textfiles. Zudem ist der AutorBackground-KnowledgeApplier Operator auf diese Dateibenennung zur Verbindung der Quellen angewiesen.

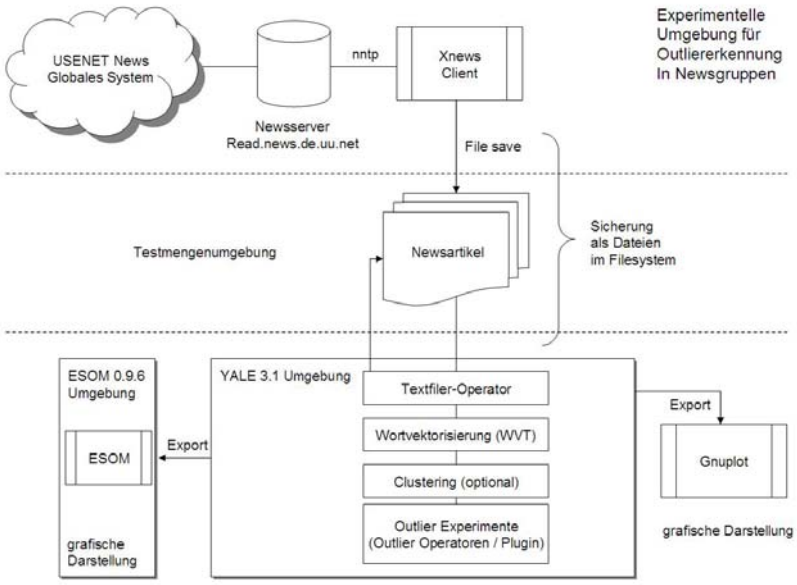


Abbildung 45 - Testumgebung für praktische Experimente

Abbildungsbeschreibung: Für das Test-SetUp wurden nur frei verfügbare Komponenten verwendet, deren Anwendung sich leicht nachvollziehen lässt. Dadurch können die Experimente flexibel realisiert und beliebig wiederholt werden.

Die Newsartikel können nun innerhalb der Filestruktur nach Gruppen und nach Kandidatenkategorien (vergleiche Kapitel 4.4) geordnet werden. Werden die Artikel nach Kategorien in Subverzeichnisse verschoben, so erlaubt das WVtool in YALE die Kennzeichnung jeder Kategorie (inkl. der Nicht-Outlier Kategorie) als Label eines Objektes in YALE. Auch hier kann über den Plotter die Datenmenge dann jeweils nach Label und auch nach den Ergebnissen der Outliererkennung farblich gekennzeichnet werden. Dies erleichtert den Vergleich zwischen Erwartung und Ergebnis erheblich.

Innerhalb der Lernumgebung YALE werden nun die einzelnen Experimente durchgeführt. Diese sind in den folgenden Kapiteln im Detail ausgeführt. Prinzipiell wird nach optionaler Filterung von Textteilen der Artikel eine Vektorisierung der Artikel vorgenommen und der so gewonnene Ursprungsdatenraum mit Outlier-Verfahren untersucht. Die Ergebnisse werden entweder direkt in YALE grafisch ausgewertet, oder die Rohdaten der experimentellen Ergebnisse werden nach GNUplot exportiert, um eine 3-dimensionale Darstellung zu realisieren.

Im Bereich der Anwendung der ESOM Tools [87] erlauben spezielle Operatoren (vgl. auch Kapitel 5.2.5) den Export des ExampleSets aus YALE in das ESOM Learner Format. Die ESOM Tools selbst werden dann zur Visualisierung der Ergebnisse verwendet.

6.2. Testmengenbeschreibung

6.2.1. Generelle Hinweise

In diesem Kapitel wird die für die Experimente herangezogene Testmenge detailliert beschrieben. Für die Auswahl der Testmenge wurden folgende Kriterien angewandt:

- Die Newgruppe muss hinreichend stark frequentiert sein, um einen Snapshot ausreichender Größe über einen kleinen Zeitraum zu erlauben, d.h. die Zeitpunkte der Postings sollten nicht zu lange auseinander liegen um zusammenhängende Diskussionsthreads zu repräsentieren; gleichzeitig darf die Gruppe nicht „schlafend“ sein, denn es gibt gewisse Gruppen, in denen kein sinnvoller Austausch (außer SPAM Postings) stattfindet.
- Die Inhalte müssen (bereits vom Thema her) rein textueller Natur sein, d.h. ohne multimediale Inhalte, welche in den Nachrichtenverkehr eincodiert wären.
- Testsprache soll aufgrund der Eignung für die Vektorisierung möglichst Englisch sein.

Daher wurde die Newsgruppe alt.support.cancer verwendet. Für deren Snapshot wurden jeweils entsprechende Outlierkandidaten nach dem in Kapitel 4.4 vorgestellten Kategorienmodell klassifiziert. Eine ausführliche Beschreibung dieser Testmengenklassifizierung ist im Umfang des YALE Outlier Plugin [105] enthalten.

6.2.2. alt.support.cancer Testmenge

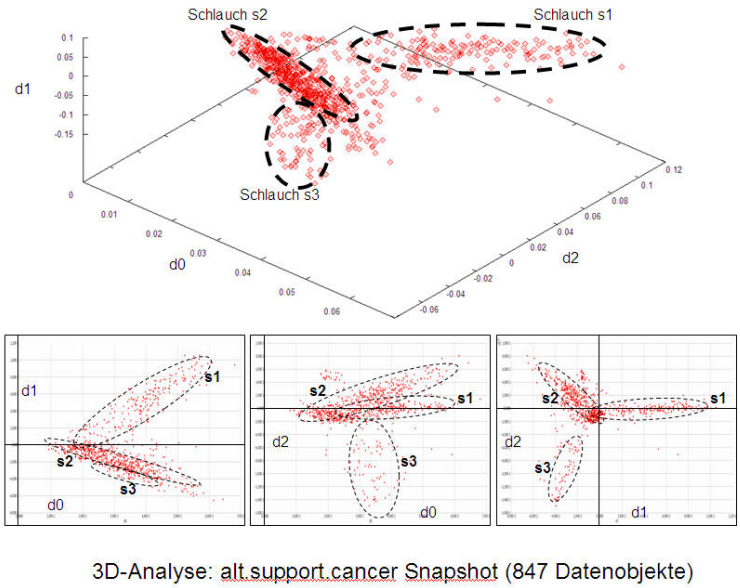
Für die vorliegende Testmenge wurde ein zusammen mit [105] ausgelieferter Snapshot der Newsgruppe alt.support.cancer in einem separaten Dateibaum mit einzelnen Textdateien pro Newsartikel ausgewertet. Nach Vektorisierung innerhalb YALE mittels WVTool wurde eine Objektmenge mit 847 Objekten und 2949 Dimensionen in einem Suchraum aufgespannt. Hierbei wurde ein Pruning der Terme außerhalb des vom WVTools Version 2.1 vorgegebenen Standardintervalls [4 ; 2000] bezogen auf deren Häufigkeit durchgeführt. Ab Version 2.2 gibt das WVTool standardmäßig dieses Intervall nicht mehr vor, sodass die Intervallgrenzen beim Laden von Experimenten manuell nachgestellt werden müssen, wenn die Experimente mit Version 2.1 abgespeichert wurden. Dieser Unterschied ist zu beachten, da die Vektorisierung sonst ca. 15.000 Dimensionen zum Ergebnis hätte.

Jedes Objekt im 2949-dimensionalen Datenraum repräsentiert einen Artikel des Snapshots der Newsgruppe. Um einen ersten Eindruck dieser hochdimensionalen Datenmenge zu gewinnen, wurde mittels Singular Value Decomposition eine Reduktion auf eine dreidimensionale Sicht durchgeführt. Abbildung 46 zeigt diese Sicht in einer dreidimensionalen Projektion und den drei zueinanderstehenden Achsen in zweidimensionalen Projektionen.

Beschreibung	**Bezeichner**	**Anzahl**	**Anteil**
Artikel	article	847	100,00%
SPAM Outlier	ngSPO	19	2,24%
Falsche Postings	gTO_WP	15	1,77%
Cross Postings	gTO_CP	25	2,95%
Einzelpostings	gDO_SP	28	3,31%
Diskussionsabweichler	gDO_TD	101	11,92%

Tabelle 1 - Ergebnisse der Vorkategorisierung im Überblick

Tabellenbeschreibung: Pro Kategorie wurde eine Anzahl von Artikeln als potentielle Outlier dieser Kategorie identifiziert und der Anteil an der Gesamtartikelzahl ausgewiesen.



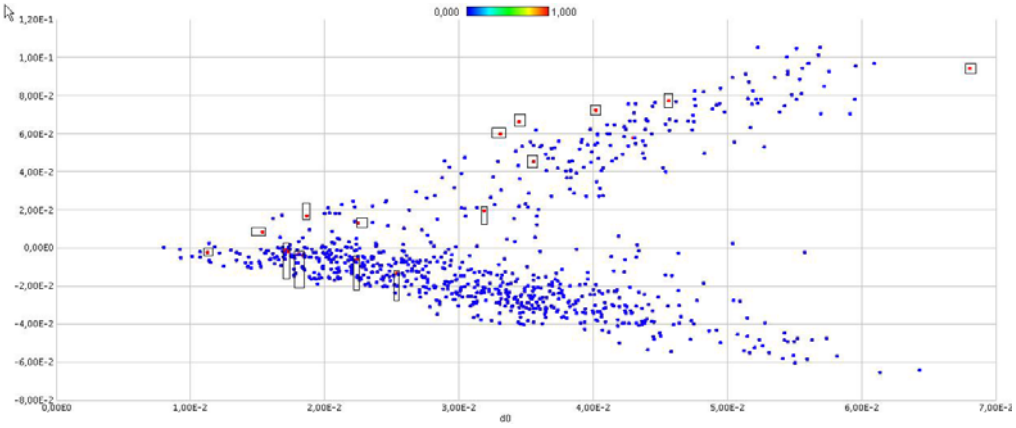
3D-Analyse: alt.support.cancer Snapshot (847 Datenobjekte)

Abbildung 46 - Testmengenanalyse in reduzierter Dimensionalität

Abbildungsbeschreibung: Es wird eine dreidimensionale Reduktion der 2.949-dimensionalen Menge visualisiert. Von besonderem Interesse ist die Frage, wo sich die vorkategorisierten Objekte befinden.

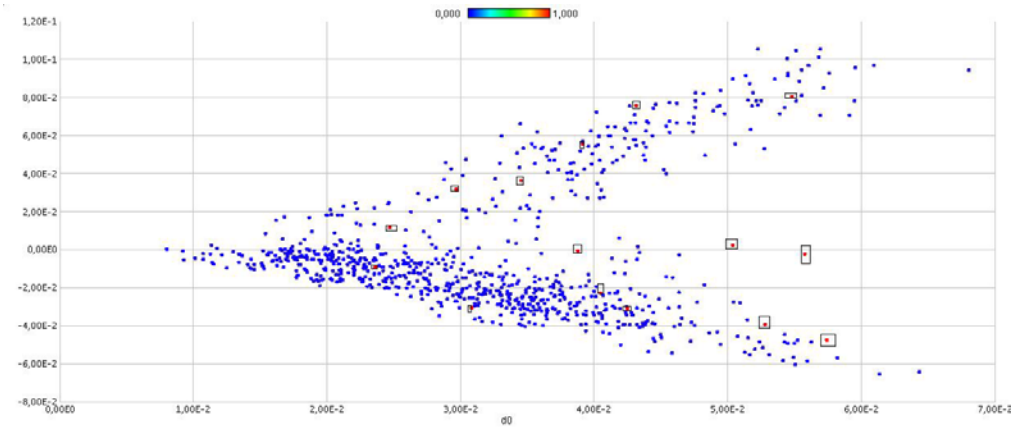
Hierbei ist gut zu sehen, dass sich die Mehrzahl der Objekte in drei Schläuchen befindet, sich also hier Ansammlungen von Vektoren der Einzeltexte bilden. Dabei darf die niedrigdimensionale Projektion nicht darüber hinwegtäuschen, dass natürlich nur ein stark vereinfachtes Bild der Ursprungsmenge dargestellt wird und daher auch ein Potential für Fehlinterpretationen besteht. Kernfrage im Rahmen der Klassifizierung potentieller Outlierkandidaten nach dem vorgestellten Schema war nun, wie sich diese in Zahlen darstellt. Die Ergebnisse sind in Tabelle 1 aufgeführt.

In den folgenden Abbildungen wird nun pro Kategorie grafisch in einer zweidimensionalen Reduktion der Datenmenge nach Singular Value Decomposition die Zahl der Outlierkandidaten durch Markierung ausgewiesen.



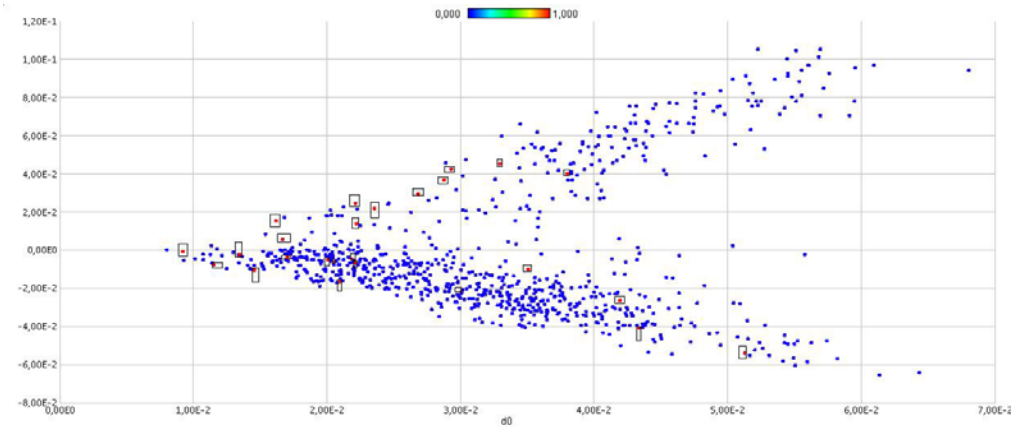
ngSPO Outlier Kandidaten

19 Objekte (2,24%)



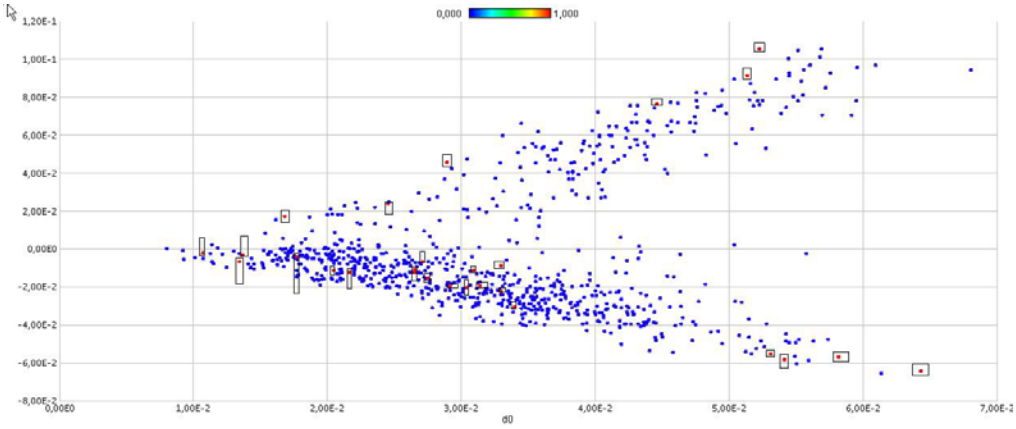
gTO_WP
Outlier
Kandidaten

15 Objekte
(1,77%)



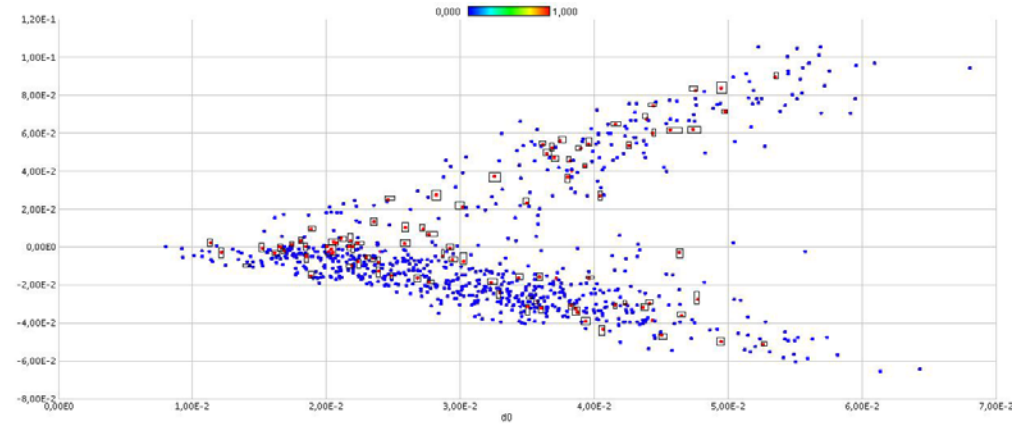
gTO_CP
Outlier
Kandidaten

25 Objekte
(2,95%)



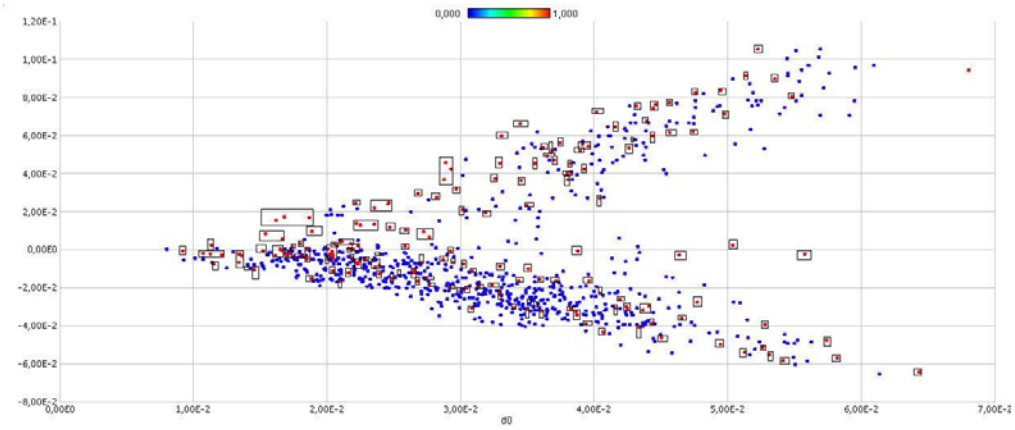
gDO_SP
Outlier
Kandidaten

28 Objekte
(3,31%)



gDO_TD
Outlier
Kandidaten

101 Objekte
(11,92%)



Alle Outlier Kandidaten markiert

In der grafischen Präsentation ist sehr deutlich zu sehen, dass manche Outlier Kandidaten durchaus bereits in der zweidimensionalen Reduktion aufgrund der Position der Textvektoren im Betrachtungsraum als zu erwartende Outlier in Frage kämen. Die experimentelle Untersuchung muss überprüfen, welche Verfahren Outlier finden und ob eine gänzliche oder zumindest partielle Übereinstimmung zwischen Kandidaten in den einzelnen Testkategorien und den identifizierten Outliern besteht.

6.3. Durchführung der Experimente für alt.support.cancer

Alle nachfolgend genannten Experimente beziehen sich auf die Testmenge von alt.support.cancer.

6.3.1. *D(k,n)* Experiment

***D(k,n)*-Experiment mit euklidischer Distanz**

Exemplarisch wird in diesem Kapitel die Beobachtung der Erkennung auf der Testmenge detaillierter eingeführt, bevor ein normativer Vergleich der Ergebnisse mittels Precision, Recall und F_Measure vorgenommen wird. In den anderen Experimentreihen mit alternativen Verfahren wird auf diese detaillierte Einführung dann verzichtet.

Die Analyse der Testdatenmenge von alt.support.cancer mit dem entfernungsbasierten Test zum *k*-ten nächsten Nachbarn als *D(k,n)*-Test wurde mit euklidischer Distanz auf der Ursprungsmenge mit 2.949 Dimensionen mit einem Wert für k=5, also zum 5-ten nächsten Nachbarn und einer erwarteten Zahl von Outliern von n=80 durchgeführt. Der *k*-Wert wurde gewählt, weil nach [13] die Qualität der Ergebnisse ab k=5 i.d.R. gut genug sind für eine entsprechende Outlier-Identifizierung und diese Qualität mit steigendem *k*-Wert nicht notwendigerweise signifikant ansteigt. Die Zahl für *n* wurde gewählt, um eine signifikante Menge an Outliern zu identifizieren und trotzdem vom Test vorerst nicht zu verlangen, alle 188 kategorisierten Outlier zu erkennen, sondern in etwa 10% der Gesamtdatenmenge unvoreingenommen als Outlier zu „erwarten“. Im Späteren wird untersucht, ob veränderte *n*-Werte auch Änderungen in der Effizienz der Identifizierung zur Folge haben. Die nachstehende Tabelle zeigt die Ergebnisse des $D_{k,n}(5,80)$ -Tests.

Kategorie	Erkennung	
	kategorisiert	erkannt
ngSPO	19	3
gTO_WP	15	1
gTO_CP	25	2
gDO_SP	28	11
gDO_TD	101	5
outlier	188	22
non_outlier	659	*58 (FP!)*

Tabelle 2 - *D(k,n)*-Verfahren mit k=5 und n=80 bei 2949 Dimensionen und euklidischer Distanz

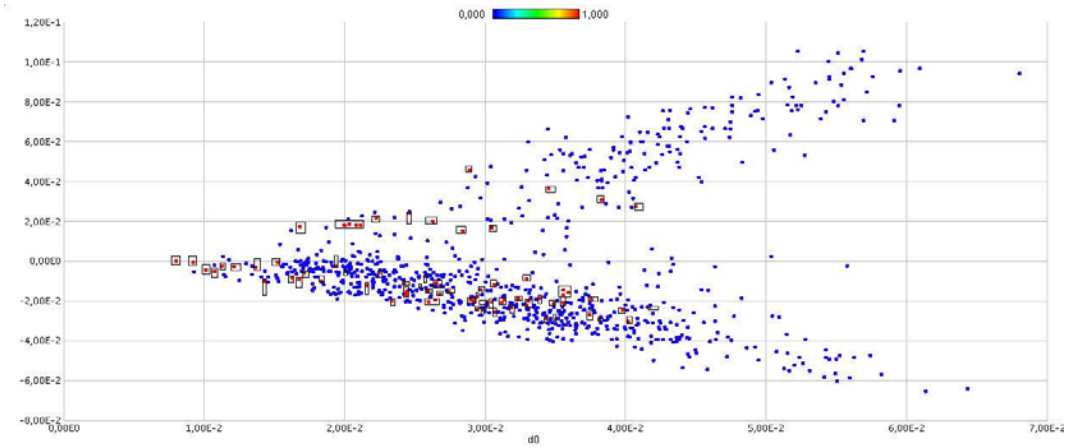


Abbildung 47 - *D(k,n)*-Test mit *k*=5 und *n*=80 in zweidimensionaler Reduktion dargestellt

Abbildungsbeschreibung: In der reduzierten Darstellung wird deutlich, dass eine Reihe von Objekten als Outlier auf der 2.949-dimensionalen Menge erkannt wurden, welche in der 2-dimensionalen Darstellung nicht intuitiv als solche gedeutet werden würden.

Abbildung 47 zeigt grafisch das Ergebnis der Outlier-Erkennung in einer zweidimensionalen Projektion (der mittels Singular Value Decomposition auf 3 Dimensionen reduzierten Datenmenge). Zumindest in der grafischen Repräsentation zeigt sich, dass viele der indentifizierten Outlier optisch nicht als solche Kandidaten zu erwarten wären, wenn von dem Abbild der dimensional reduzierten Menge ausgegangen wird. Im Späteren wird auch untersucht, wie sich die Ergebnisse der *D(k,n)*-Untersuchung auf einer dimensional stark reduzierten Datenmenge von denen der Ursprungsmenge unterscheiden. Die erkannten kategorisierten Objekte wurden jeweils farblich (rot) und durch Einkästelung gekennzeichnet.

Abbildung 48 stellt noch einmal übersichtlich die Erkennung pro Kategorie dar. Dabei sind in der obersten Reihe die nicht als Outlier kategorisierten Objekte dargestellt, in der zweiten Reihe die Diskussionsabweichler (gDO_TD), in der dritten die Einzelpostings (gDO_SP), in der vierten die Cross-Postings (gTO_CP) und darunter die Falsch-Postings (gTO_WP). In der untersten Zeile sind die SPAM-Nachrichten (ngSPO) verzeichnet. Die erkannten kategorisierten Objekte wurden auch hier jeweils farblich (rot) und durch Einkästelung gekennzeichnet.

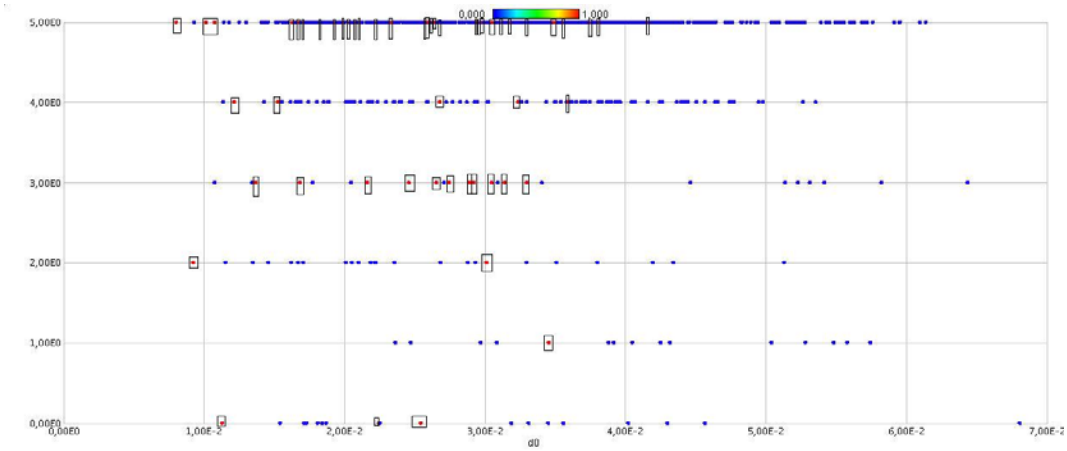


Abbildung 48 - *D(k,n)*-Test mit Visualisierung der positiv erkannten Kategorisierungen

Abbildungsbeschreibung: YALE unterstützt auch die Darstellung der Objekte in einer Dimension und anhand des entsprechenden Labels. Die unteren *y*-Skalenwerte (0 – 4) stellen die Outlierkategorien dar, der oberste (5) die Nicht-Outlier. So wird sehr gut sichtbar, welche Objekte korrekt bzw. inkorrekt erkannt wurden.

Eine Untersuchung der Auswirkung unterschiedlicher Werte für *n*, d.h. die Anzahl der erwarteten Outlier, zeigt mit einem Vergleich von *n*=20, *n*=80 und *n*=188, dass absolut gesehen die Anzahl der erkannten kategorisierten Outlier steigt. Allerdings verändert sich das Verhältnis von erkannten kategorisierten Outliern zu erkannten Outliern, die vorher nicht kategorisiert wurden, nicht wesentlich. Im Fazit führt also eine Wahl einer höheren Zahl von *n* absolut zu einem besseren Ergebnis der erkannten kategorisierten Objekte, wobei auch die Zahl der Erkennung von nicht kategorisierten Objekten gleichsam steigt.

Zusätzlich ist nun zu untersuchen, ob sich die Änderung des *k*-Wertes signifikant auf die Ergebnisse auswirkt. Dazu wurde eine weitere Experimentreihe mit den Werten *k*=5, *k*=10 und *k*=20 durchgeführt.

Die Ergebnisse zeigen deutlich, dass der Anteil der erkannten und vorher kategorisierten Outlier leicht steigt, wenn der *k*-Wert angehoben wird. Allerdings stützt die Untersuchung die Aussage von [13], dass die Erhöhung des *k*-Wertes ab einer gewissen Grenze keine hohen qualitativen Unterschiede zeigt. Wird nun die Erkenntnis, dass ein höherer *k*-Wert etwas bessere Ergebnisse zeigt, mit einem höheren gewählten Wert für *n*=188 kombiniert, so ergibt sich eine Verbesserung der Erkennung kategorisierter Kandidaten gegenüber z.B. einem Wertepaar *k*=5 und *n*=188.

Optimierung des *D(k,n)*-Experiments durch Dimensionsreduktion

Eine nahe liegende Frage ist nun, wie die Ergebnisse durch Vorverarbeitungsverfahren verbessert werden können. Kapitel 4.7 erschließt einige dieser Vorverarbeitungen. Aus der Erfahrung des Experiments lassen sich nun bereits zwei potentielle Wege ableiten, die Schwierigkeiten durch die Spärlichkeit der Urpsurngsmenge, welche vor allem in der hohen Zahl an Dimensionen begründet ist, abzumildern.

Zum einen kann die Anzahl der Dimensionen durch eine Reduktion der Attribute durch Beschneidung der Attributmenge anhand von Durchschnittsgewichten erreicht werden. Dadurch wird ein Teil der Ursprungsdaten und somit auch ein Teil der Ursprungsinformationen für die Auswertung erhalten. Gleichzeitig operiert die Gewichtung auf Term-Ebene, ist also vergleichsweise nah am Medium des „Textes“ und bezieht sich weniger auf die Positionierung von Objekten zueinander in einem vektorisierten Datenraum. Allerdings ist die Entscheidung zu anwendbaren Intervallgrenzen für die Beschneidung schwierig und eine unbewusste Verfälschung der Ursprungsmenge wahrscheinlich. Gleichzeitig lässt sich die Zahl der Attribute nicht soweit reduzieren, dass eine grafische Interpretation durch den Anwender ohne Zusatzoperationen möglich ist. Da im WVTool bereits ein Termhäufigkeitspruning stattfindet, wurde hier auf ein separates Beschneiden verzichtet.

Andererseits kann die Dimensionalität der Datenmenge mathematisch reduziert werden. Hierfür bietet sich z.B. das Singular Value Decomposition Verfahren an. Die resultierende Outlier-Analyse ist i.d.R. effizient, da sie auf einem für die meisten Standardverfahren gut darstellbaren niedrigdimensionalen Raum stattfindet. Zudem erlaubt die grafische Analyse durch den Anwender eine gute visuelle Überprüfung der Signifikanz und Plausibilität der Ergebnisse. Allerdings könnte die starke Reduktion der Dimensionen zu Fehlinterpretationen führen. Die Dimensionsreduktion mit SVD wurde hier durchgeführt.

Im Mittelpunkt der Anwendung von Dimensionsreduktion durch Singular Value Decomposition stehen zwei Fragen: (1) Lässt sich durch die Anwendung die Erkennung von Outliern im *D(k,n)*-Verfahren im Hinblick auf die Performance verbessern? Und (2) wirkt sich eine Reduzierung positiv auf die Anteile der erkannten kategorisierten Objekte aus, oder ist diese Auswirkung negativ oder neutral? Beispielhaft wird dies in einer dreidimensionalen SVD-Projektion der ursprünglichen Datenmenge untersucht.

Bezüglich der Performance des Verfahrens kann eine signifikante Verbesserung festgestellt werden, wie dies auch durch die Komplexität des *D(k,n)*-Verfahrens zu erwarten ist. Allerdings muss vorher eine Reduktion der Dimensionen durchgeführt werden, deren Aufwand sehr hoch ist. Im Praxisfall wird jedoch ein *D(k,n)*-Experiment auf der Originalmenge oft mit einer anschließenden Dimensionsreduktion nach dem SVD Ansatz verbunden sein, sodass dieser Aufwand für den Vergleich dort ggf. herangezogen werden muss. Wird somit der Aufwand einer wie auch immer gearteten Dimensionsreduktion aus der Betrachtung ausgespart, ergibt sich ein direkter Vergleich bezogen auf die Anzahl der Dimensionen der Testmenge *m*. Die vorliegende Implementierung des *D(k,n)*-Verfahrens hat eine Komplexität von $O(m \cdot n_X^{\,2}) + O(k \cdot n_X) + O(n_X)$. Da für die vorliegende Testmenge die Zahl der Dimensionen *m*=2949 dem ca. 3,48-fachen der Anzahl der Objekte $n_X = 847$ entspricht, kann durch Dimensionsreduktion die Komplexität von $O(3{,}48 \cdot n_X^{\,3}) + O(k \cdot n_X) + O(n_X)$ auf $O(10{,}44 \cdot n_X^{\,2}) + O(k \cdot n_X) + O(n_X)$ verringert werden, wenn die Anzahl der Dimensionen auf *m*=3 reduziert wird. Allgemein kann bei einem Wert $m > n_X$ die Komplexität um eine Potenz verringert werden. Wiederum verglichen mit der praktischen Anwendung ist jedoch der Aufwand für die Dimensionsreduktion hoch, sodass der Performancegewinn des *D(k,n)*-Anteils nicht stark ins Gewicht fällt.

Wichtig ist vor allem der Vergleich der Ergebnisse der Erkennung kategorisierter Objekte bei der Outlieridentifizierung in der dreidimensionalen Projektion mit den Ergebnissen bezogen auf die

Ursprungsmenge mit 2.949 Dimensionen. Im Fazit wird durch die Reduktion der Dimensionen eine Performance-Optimierung erreicht, welche aber zu Lasten der Erkennung kategorisierter Objekte als Outlier geht. Dies wird in der Auswertung der Experimente später im Detail ausgeführt.

Optimierung des *D(k,n)*-Experiments durch Einsatz der Kosinus-Distanz

Wird nun das *D(k,n)*-Experiment mit einer anderen Art der Distanz durchgeführt, ergibt sich das in der folgenden Tabelle dargestellte Ergebnis für *k*=20 und *n*=188.

Kategorie	Test-menge kat.	k_20_n_188							
		euklidische Distanz		quadratische Distanz		Kosinus-Distanz		Winkelmaß (rad)	
		erkannt	Recall	erkannt	Recall	erkannt	Recall	erkannt	Recall
ngSPO	19	10	52,63%	9	47,37%	10	52,63%	10	52,63%
gTO_WP	15	1	6,67%	1	6,67%	1	6,67%	1	6,67%
gTO_CP	25	11	44,00%	11	44,00%	11	44,00%	11	44,00%
gDO_SP	28	15	53,57%	16	57,14%	14	50,00%	15	53,57%
gDO_TD	101	20	19,80%	20	19,80%	19	18,81%	18	17,82%
outlier ges.	*188*	*57*	*30,32%*	*57*	*30,32%*	*55*	*29,26%*	*55*	*29,26%*

Tabelle 3 - Entfernungsmaße im Vergleich beim *D(k,n)*-Verfahren

Ausgewertet wurde eine Kombination der Werte *k* und *n* mit aus den vorhergegangenen Durchführungen bekannten verbesserten Ergebnissen, wobei nun die quadratische Distanz ausgewählt wurde, um wachsende Abstände zwischen Objekten stärker zu betonen, und des weiteren die Kosinusdistanz und zum Vergleich das Winkelmaß zwischen den Vektoren der Objekte als Abstandsfunktion eingesetzt wurde. Die Identifizierung von kategorisierten Objekten als Outlier ist bei Anwendung aller Distanzmaße in etwa gleich (d.h. bei Einsatz der Winkelmaße leicht geringer). Im Mittelpunkt der Vergleichsbetrachtung sollte also stehen, welche kategorisierten Objekte jeweils von den unterschiedlichen Anwendungen des *D(k,n)*-Verfahrens mit diversen Abstandsmaßen gleich oder unterschiedlich erkannt werden.

Erstaunlicherweise ergibt sich ein Bild, in welchem das *D(k,n)*-Verfahren mit allen eingesetzten Abstandsmaßen nahezu identische Outliergruppen ermittelt. Auch der direkte Vergleich der Liste von Objekten zeigt, dass bis auf vereinzelte Ausnahmen die gleichen Objekte als Outlier erkannt werden. Das in Abbildung 38 auf Seite 81 gezeigte Beispiel hätte auch bei dieser Testmenge stark abweichende Ergebnisse vermuten lassen. Die Tatsache jedoch, dass in anderen Testmengen erwartungsgemäße Unterschiede in den als Outlier identifizierten Objektmengen bestehen, sofern unterschiedliche Abstandsmaße angewendet werden, lässt einen Implementierungsfehler als Schluss nicht zu. Vielmehr könnte die Vermutung angestellt werden, dass durch die Spärlichkeit der Testmenge und die hohe Dimensionszahl gleiche Objekte als *k*-te nächste Nachbarn identifiziert werden und dabei die Wahl eines Abstandsmaßes wie gezeigt nur geringfügige Auswirkungen hat. Dabei wird zu prüfen sein, ob dies bei anderen Verfahren ähnliche Effekte zur Folge hat. Bei der Anwendung des ESOM Verfahrens decken sich die identifizierten Outlier bei der Anwendung von euklidischer Distanz und Kosinus-Distanz nur in einem sehr kleinen Bereich (vgl. auch Kapitel 6.3.4).

Auswertung mittels Precision, Recall, Fall Out und F_Measure

Die folgende Tabelle zeigt die Auswertung des *D(k,n)*-Experiments im Vergleich der ermittelten Werte für die Anzahl der Objekte in den Mengen TP, FN, TN und FP, sowie die Genauigkeit (Precision) und Vollständigkeit (Recall) der Ergebnisse und auch den Wert für Fall Out und F_Measure.

k	n	m	TP	FN	TN	FP	Precision	Recall	Fall Out	f_measure
5	20	2949	6	182	645	14	0,3000	0,0319	0,0212	0,0577
5	80	2949	22	166	601	58	0,2750	0,1170	0,0880	0,1642
5	188	2949	49	139	520	139	0,2606	0,2606	0,2109	0,2606
10	80	2949	23	165	602	57	0,2875	0,1223	0,0865	0,1716
20	80	2949	25	163	604	55	0,3125	0,1330	0,0835	0,1866
20	188	2949	57	131	528	131	0,3032	0,3032	0,1988	0,3032
5	20	3	3	185	642	17	0,1500	0,0160	0,0258	0,0288
5	80	3	18	170	597	62	0,2250	0,0957	0,0941	0,1343
5	188	3	38	150	509	150	0,2021	0,2021	0,2276	0,2021
10	80	3	16	172	595	64	0,2000	0,0851	0,0971	0,1194
20	80	3	8	180	587	72	0,1000	0,0426	0,1093	0,0597
20	188	3	29	159	500	159	0,1543	0,1543	0,2413	0,1543

Tabelle 4 - Auswertung *D(k,n)*-Verfahren / Precision und Recall

Tabellenbeschreibung: Die Tabelle zeigt die Auswertung des *D(k,n)*-Verfahrens für Precision und Recall für unterschiedliche Wertekombinationen von *k* und *n* bei voller Anzahl an Dimensionen der Ursprungsmenge (*m*=2949) und reduzierter Anzahl an Dimensionen (*m*=3). Die Ergebnisse für *m*=2949 sind für die euklidische Distanz und die Kosinusdistanz als Abstandsmaß gleich und daher nur einmal aufgeführt. Die Ergebnisse für *m*=3 wurden nur für die euklidische Distanz als Abstandsmaß ermittelt.

In Abbildung 49 wird gezeigt, dass eine Erhöhung des Wertes für *n* die Genauigkeit des Ergebnisses im hochdimensionalen Bereich leicht verringert, dafür aber die Vollständigkeit erhöht. Die Erhöhung des Wertes für *k* erhöht sowohl die Genauigkeit, als auch die Vollständigkeit. Im Rahmen der Anpassung von *k* und *n* kann gezeigt werden, dass das Wertepaar *k*=20 und *n*=188 einen Optimierungsweg zeigt, ohne das ein Anspruch auf einen Nachweis erfolgt, dass nicht weiter durch Wertekombinationen optimiert werden könnte.

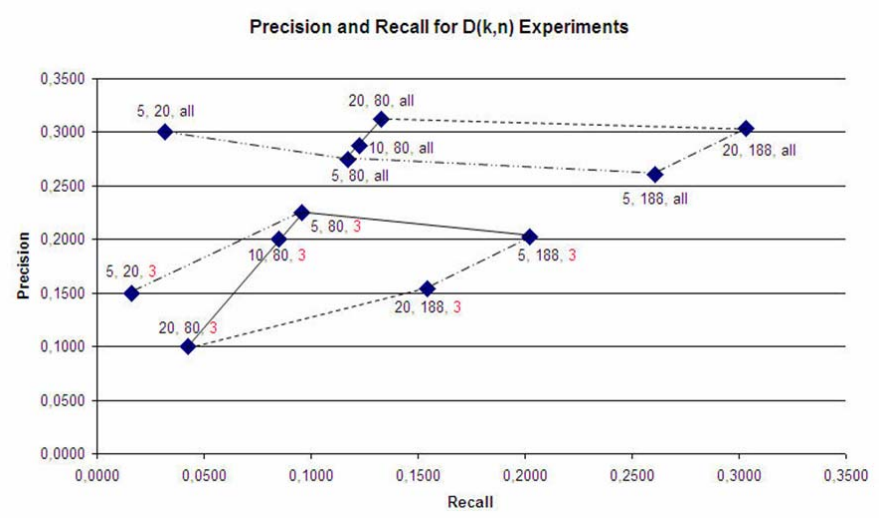


Abbildung 49 - *D(k,n)*-Auswertung für Precision und Recall

Abbildungsbeschreibung: Die Ergebnisse der unterschiedlichen Experimentreihen werden im Vergleich gezeigt und sind mit den Parametern *k,n,m* (*m*=2949 entspricht „all“) gekennzeichnet. Die Ursprungsmenge mit einer hohen Dimensionszahl steht für deutlich bessere Ergebnisse.

Der Abbildung ist zudem deutlich zu entnehmen, dass die Ergebnisse auf der in der Dimension reduzierten Menge insgesamt deutlich schlechter sind für alle betrachteten Wertekombinationen. Hier verschlechtert die Erhöhung des Wertes für *k* zudem das Ergebnis, während die Erhöhung des Wertes für *n* auch hier positive Auswirkung zeigt.

Abbildung 50 bestätigt noch einmal die Ergebnisse der Auswertung unterschiedlicher Wertekombinationen durch die grafische Analyse von F_Measure als Funktion des Wertes für *n* mit Bezug auf den Wert für *k* und die Anzahl *m* der Dimensionen des Suchraumes.

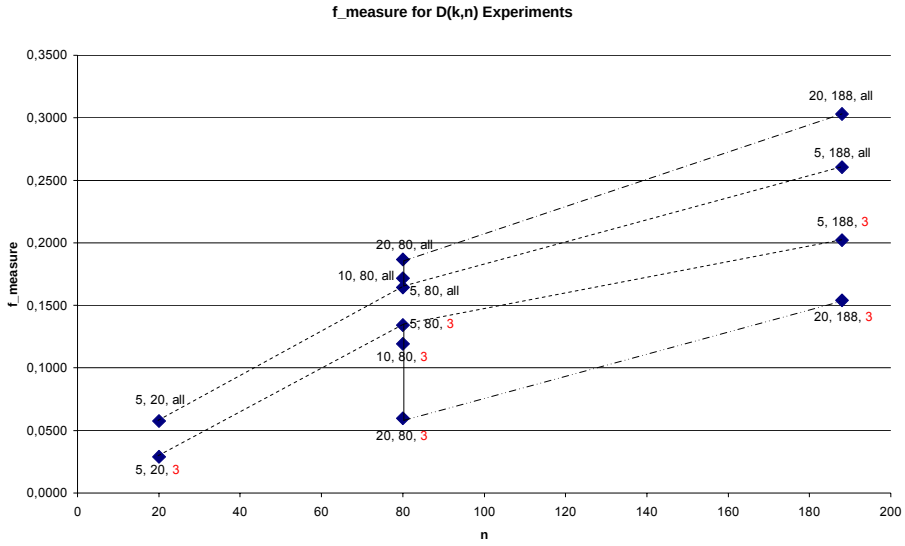


Abbildung 50 - Auswertung des *D(k,n)*-Experiments - F_Measure

Abbildungsbeschreibung: Die Ergebnisse der unterschiedlichen Experimentreihen werden für F_Measure im Vergleich gezeigt und sind mit den Parametern *k,n,m* (*m*=2949 entspricht „all“) gekennzeichnet.

Optimierung des *D(k,n)*-Verfahrens durch Textsplitting

Eine der im Kapitel 4.7 aufgeführten Vorverarbeitungsmöglichkeiten ist das Abschneiden der Headerinformationen der Newsartikel, um dadurch zum einen die Dimensionsmenge zu verringern, als auch die textuelle Analyse auf den Textkörper (Body) des Artikels zu beschränken. Um die Auswirkungen eines solchen Vorgehens auf das *D(k,n)*-Verfahren zu untersuchen, wurden bis auf die „Subject“-Zeile und die „From“-Zeile, welche den Autor des Artikels enthält, alle Headerzeilen aus den Artikeln entfernt. Die Ergebnismenge wurde erneut vektorisiert. Die resultierenden Textvektoren mit 2.555 Dimensionen wurden sodann den gleichen *D(k,n)*-Experimenten unterzogen und dies wurde zudem nach SVD-Reduktion auf eine dreidimensionale Menge und jeweils mit Einsatz der euklidischen und der Kosinusdistanz als Abstandsmaß wiederholt. Die folgende Tabelle führt die Ergebnisse aus.

	k	n	m	TP	FN	TN	FP	Precision	Recall	FallOut	f_measure
euklidische Distanz und Kosinusdistanz	5	20	2555	6	182	645	14	0,3000	0,0319	0,0212	0,0577
	5	80	2555	23	165	602	57	0,2875	0,1223	0,0865	0,1716
	5	188	2555	52	136	523	136	0,2766	0,2766	0,2064	0,2766
	10	80	2555	23	165	602	57	0,2875	0,1223	0,0865	0,1716
	20	80	2555	22	166	601	58	0,2750	0,1170	0,0880	0,1642
	20	188	2555	52	136	523	136	0,2766	0,2766	0,2064	0,2766
euklidische Distanz	5	20	3	0	188	639	20	0,0000	0,0000	0,0303	0,0000
	5	80	3	6	182	585	74	0,0750	0,0319	0,1123	0,0448
	5	188	3	23	165	494	165	0,1223	0,1223	0,2504	0,1223
	10	80	3	7	181	586	73	0,0875	0,0372	0,1108	0,0522
	20	80	3	5	584	75	183	0,0266	0,0085	0,7093	0,0129
	20	188	3	21	167	492	167	0,1117	0,1117	0,2534	0,1117
Kosinusdistanz	5	20	3	1	187	640	19	0,0500	0,0053	0,0288	0,0096
	5	80	3	15	173	594	65	0,1875	0,0798	0,0986	0,1119
	5	188	3	35	153	506	153	0,1862	0,1862	0,2322	0,1862
	10	80	3	12	176	591	68	0,1500	0,0638	0,1032	0,0896
	20	80	3	13	175	592	67	0,1625	0,0691	0,1017	0,0970
	20	188	3	34	154	505	154	0,1809	0,1809	0,2337	0,1809

Tabelle 5 - Auswertung *D(k,n)*-Verfahren nach Textsplitting

Tabellenbeschreibung: Die Tabelle zeigt die Auswertung des *D(k,n)*-Verfahrens für Precision und Recall für unterschiedliche Wertekombinationen von *k* und *n* bei voller Anzahl an Dimensionen der Ursprungsmenge nach Textsplitting (*m*=2555) und reduzierter Anzahl an Dimensionen (*m*=3). Die Ergebnisse für *m*=2555 sind für die euklidische Distanz und die Kosinusdistanz als Abstandsmaß gleich und daher nur einmal aufgeführt. Die Ergebnisse für *m*=3 wurden für die euklidische Distanz und die Kosinusdistanz als Abstandsmaß separat dargestellt.

Das Ergebnis zeigt in der grafischen Analyse, dass sich gegenüber der Betrachtung der Ursprungsmenge keinerlei Verbesserungen der Ergebnisse des *D(k,n)*-Verfahrens erreichen lassen.

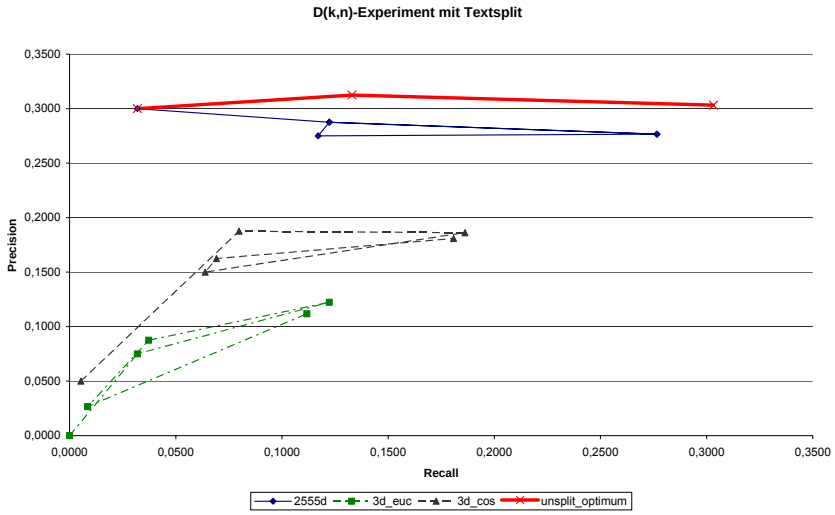


Abbildung 51 - *D(k,n)*-Experiment mit Textsplitting

Abbildungsbeschreibung: Das Textsplitting wirkt sich für die vorgegebene Testmenge beim *D(k,n)*-Verfahren negativ auf die Ergebnisse der Erkennung bzgl. der Vorkategorisierung aus.

Abbildung 51 zeigt die erreichbaren Kombinationen für Precision und Recall im Vergleich mit dem Optimum des Verfahrens auf der Ursprungsmenge. Hierbei wird ersichtlich, dass es immer günstigere Wertekombinationen für die Ursprungsmenge gibt. Damit erübrigt sich eine Analyse der f_measure Werte der Experimentreihen.

Ein weiteres interessantes Ergebnis ist hier jedoch, dass nach der Reduktion der Dimensionszahl auf drei Dimensionen der Einsatz der Kosinusdistanz als Abstandsmaß zu unterschiedlichen Ergebnissen führt. In der ersten Experimentreihe wurde das Distanzmaß nur auf der 2949-dimensionalen Menge geändert.

Zusammenfassung

Insgesamt kann festgestellt werden, dass für das *D(k,n)*-Experiment auf der Testmenge alt.support.cancer die Optimierung des Wertepaars für *k* und *n* Ergebnisverbesserungen zeigt. Eine Reduktion der Dimensionalität des Suchraumes verschlechtert die Ergebnisse der Outliererkennung, ein entsprechender Performancegewinn wird also mit negativen Auswirkungen auf die Ergebnisqualität erkauft. Der Einsatz unterschiedlicher Entfernungsmaße verändert die Ergebnisse der Outliererkennung mittels *D(k,n)*-Verfahren in den untersuchten hochdimensionalen Räumen nicht. Eine Optimierung durch Ausblenden der Headerinformationen brachte keine Ergebnisverbesserung.

6.3.2. *DB(p,D)*-Experiment

Das *DB(p,D)*-Experiment wurde unter Einsatz des entsprechenden YALE-Operators durchgeführt und dabei wurden Testreihen mit unterschiedlichen Parameterkombinationen von *D* und *p* durchgeführt, so wie dies auch von den Autoren des Verfahrens vorgeschlagen wird. Aufgrund der anzunehmenden spärlichen Verteilung der Objekte im Suchraum wurde von der durchschnittlichen Distanz aller Objekte ausgegangen und dieser Wert für *D* wurde um die Varianz bzw. Standardabweichung der Distanz varriert. Für jede dieser variierten Distanzwerte wurden verschiedene Werte für *p* untersucht. Aus den Ergebnissen der Untersuchung wurden jeweils grafische Analysen für die Genauigkeit (Precision) und Vollständigkeit (Recall) des Ergebnisses in Bezug auf die Vorkategorisierung und Erkennung (True Posivites, True Negatives, False Positives und False Negatives) abgeleitet. Dabei wurde angenommen, dass die Anzahl der identifizierten Outlier (OI) sinnvollerweise nicht oberhalb der Anzahl vorkategorisierter Outlier liegen soll, um eine Eingrenzung der Optimierung von Genauigkeit und Vollständigkeit zu erreichen.

***DB(p,D)*-Verfahren mit euklidischer Distanz**

In den folgenden Tabellen sind die Ergebnisse der Variierung der Werte für *p* und *D* ausgeführt, wobei das Experiment auf der 2949-dimensionalen Datenmenge durchgeführt wurde. Die erste Tabelle zeigt das Experiment für den Durchschnitt der Distanz aller Objekte.

D	p	TP	TN	FP	FN	Precision	Recall	FallOut	F_Measure	OI
1,3771	0,800	66	489	170	122	0,2797	0,3510	0,2580	0,3113	236
1,3771	0,850	39	561	98	149	0,2847	0,2074	0,1487	0,2400	137
1,3771	0,875	27	601	58	161	0,3176	0,1436	0,0880	0,1978	85
1,3771	0,900	18	630	29	170	0,3830	0,0957	0,0440	0,1531	47
1,3771	0,925	6	648	11	182	0,3529	0,0319	0,0166	0,0585	17
1,3771	0,950	2	656	3	186	0,4000	0,0106	0,0045	0,0207	5
1,3771	0,975	0	658	1	188	0,0000	0,0000	0,0015	0,0000	1

Tabelle 6 - Auswertung *DB(p,D)*-Verfahren mit *m*=2949, euklidischer Distanz und *D*=Ø

Die zweite Tabelle zeigt die Ergebnisse des Experiments für den Wert von *D* aus der Summe der durchschnittlichen Distanz der Objekte und der Varianz der Distanz zwischen den Objekten der Testmenge.

D	p	TP	TN	FP	FN	Precision	Recall	FallOut	F_Measure	OI
1,3806	0,800	48	523	136	140	0,2608	0,2553	0,2064	0,2580	184
1,3806	0,850	30	591	68	158	0,3061	0,1596	0,1032	0,2098	98
1,3806	0,875	21	619	40	167	0,3443	0,1117	0,0607	0,1687	61
1,3806	0,900	9	636	23	179	0,2813	0,0479	0,0349	0,0819	32
1,3806	0,925	2	655	4	186	0,3333	0,0106	0,0006	0,0205	6
1,3806	0,950	2	657	2	186	0,5000	0,0106	0,0030	0,0208	4
1,3806	0,975	0	658	1	188	0,0000	0,0000	0,0015	0,0000	1

Tabelle 7 - Auswertung *DB(p,D)*-Verfahren mit *m*=2949, euklidischer Distanz und *D*=Ø+σ^2

Die dritte Tabelle zeigt die Ergebnisse des Experiments für den Wert von *D* aus der Differenz der durchschnittlichen Distanz der Objekte und der Standardabweichung der Distanz zwischen den Objekten.

D	p	TP	TN	FP	FN	Precision	Recall	FallOut	F_Measure	OI
1,3173	0,800	179	20	639	9	0,2188	0,9521	0,9697	0,3558	818
1,3173	0,850	154	90	596	34	0,2130	0,8191	0,8634	0,3381	750
1,3173	0,875	142	126	533	46	0,2103	0,7553	0,8088	0,3290	675
1,3173	0,900	124	193	466	64	0,2102	0,6596	0,7071	0,3188	590
1,3173	0,925	104	313	346	84	0,2311	0,5532	0,5250	0,3260	450
1,3173	0,950	72	481	178	116	0,2880	0,3830	0,2701	0,3288	250
1,3173	0,975	40	584	75	148	0,3478	0,2128	0,1138	0,2640	115
1,3173	0,978	31	596	63	157	0,3300	0,1649	0,0956	0,2199	94
1,3173	0,980	27	609	50	161	0,3507	0,1436	0,0759	0,2038	77
1,3173	0,990	3	647	12	185	0,2000	0,0160	0,0182	0,0296	15

Tabelle 8 - Auswertung *DB(p,D)*-Verfahren mit *m*=2949, euklidischer Distanz und *D*=Ø-σ

Die vierte Tabelle zeigt die Ergebnisse des Experiments für den Wert von *D* aus der Differenz der durchschnittlichen Distanz der Objekte und der Varianz der Distanz zwischen den Objekten der Testmenge.

D	p	TP	TN	FP	FN	Precision	Recall	FallOut	F_Measure	OI
1,3735	0,800	75	455	204	113	0,2688	0,3989	0,3096	0,3212	279
1,3735	0,850	48	526	133	140	0,2652	0,2553	0,2018	0,2602	181
1,3735	0,875	35	576	83	153	0,2966	0,1862	0,1260	0,2288	118
1,3735	0,900	25	611	48	163	0,3425	0,1330	0,0728	0,1916	73
1,3735	0,906	22	621	38	166	0,3666	0,1170	0,0577	0,1774	60
1,3735	0,912	18	632	27	170	0,4000	0,0957	0,0409	0,1545	45
1,3735	0,925	9	645	14	179	0,3913	0,0479	0,0212	0,0854	23
1,3735	0,950	3	654	5	185	0,3750	0,0160	0,0075	0,0307	8
1,3735	0,975	0	657	2	188	0,0000	0,0000	0,0030	0,0000	2

Tabelle 9 - Auswertung *DB(p,D)*-Verfahren mit *m*=2949, euklidischer Distanz und *D*=Ø-σ^2

In der Abbildung 52 sind die Ergebnisse für die euklidische Distanz als Abstandsmaß bei 2949 Dimensionen noch einmal zusammengefasst.

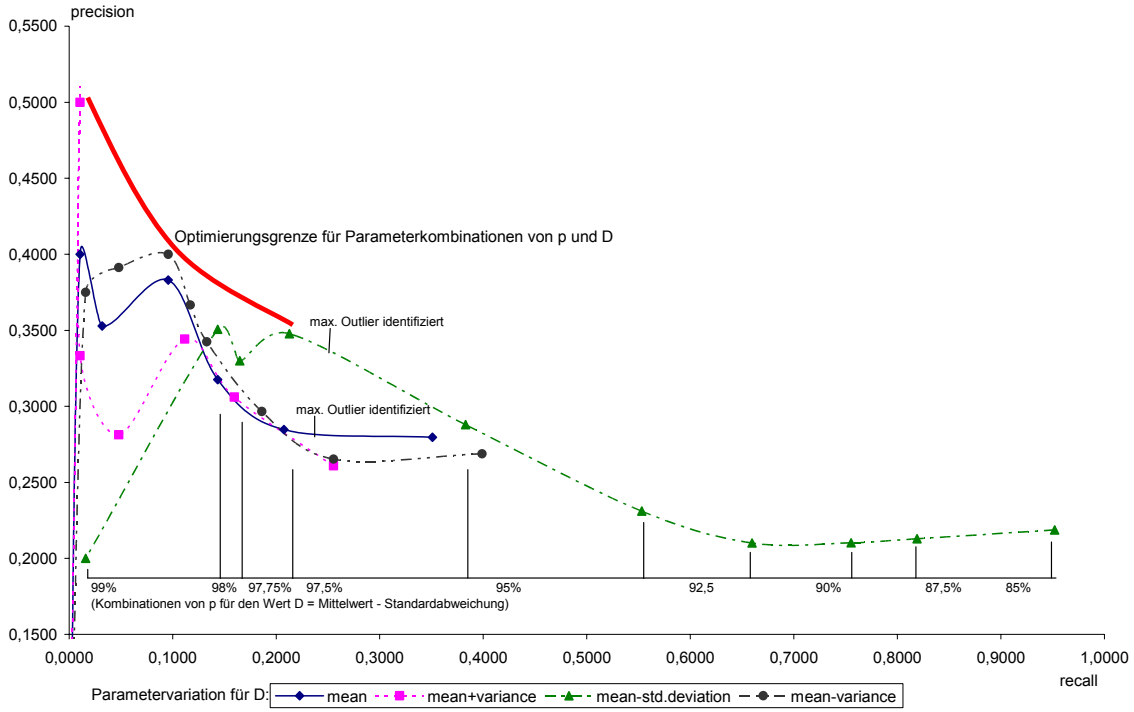


Abbildung 52 - *DB(p,D)*-Verfahren für euklidische Distanz bei voller Dimensionalität

> *Abbildungsbeschreibung:* Die Ergebnisse der Optimierung der Parameter *p* und *D* für das *DB(p,D)*-Verfahren bei euklidischer Distanz ergeben eine Optimierungsgrenze für die Erkennung der vorkategorisierten Objekte. Allerdings wurde auch vermerkt, ab wann mehr als die vorkategorisierten Objekte als Outlier erkannt werden, da ein darauf basierender hoher Recall-Wert nicht sinnvoll ist.

Die Ergebnisse zeigen, dass eine Variation der Parameter *p* und *D* eine Optimierung der Ergebnisse in Bezug auf die Vorkategorisierung und ausgerichtet auf die Genauigkeit (Precision) und Vollständigkeit (Recall) sowie deren ausgewogenes Verhältnis durchaus möglich ist. Dabei zeigt sich in der Zahl der Experimente aber auch eine Grenze der Optimierbarkeit (hier ohne Anspruch auf Vollständigkeit oder Beweis), die auf einen Kompromiss zwischen erzielbarer Genauigkeit und erzielbarer Vollständigkeit hinausläuft.

Gleichzeitig erkennt das Verfahren die vorkategorisierten Outlier im Vergleich zu anderen in dieser Arbeit eingesetzten Verfahren erstaunlich gut. Allerdings muss der Nutzer über die Werte von *p* und *D* einige Optimierungsarbeit leisten, wobei er i.d.R. nicht über die hier vorhandene Information der Vorkategorisierung, zumindest nicht in diesem Umfang, verfügen wird.

O.D.d.A erhöhen Werte für *D* oberhalb des Durchschnitts der Entfernungen zwischen den Objekten die Genauigkeit, solche unterhalb dieses Durchschnitts die Vollständigkeit, wobei in letzterem Fall meist auch das Verhältnis von Precision und Recall vorteilhafter ist.

Optimierung des *DB(p,D)*-Verfahrens durch Dimensionsreduktion

Um zu untersuchen, ob sich durch die Reduktion der Zahl der Dimensionen im Suchraum die Ergebnisse des *DB(p,D)*-Verfahrens optimieren lassen, wurde mittels Singular Value Decomposition die Zahl der Dimensionen auf m=3 reduziert. Die folgenden Tabellen zeigen die Ergebnisse der Experimente mit der euklidischen Distanz als statistisches Abstandsmaß. In der ersten Tabelle ist die durchschnittliche Distanz für *D* eingesetzt worden.

D	p	TP	TN	FP	FN	Precision	Recall	FallOut	F_Measure	OI
0,0587	0,400	66	426	233	122	0,2207	0,3511	0,3535	0,2710	299
0,0587	0,500	51	458	201	137	0,2024	0,2713	0,3050	0,2318	252
0,0587	0,600	40	479	180	148	0,1818	0,2127	0,2731	0,1960	220
0,0587	0,700	32	522	137	156	0,1893	0,1702	0,2079	0,1792	169
0,0587	0,750	25	540	119	163	0,1736	0,1330	0,1806	0,1506	144
0,0587	0,800	13	569	90	175	0,1262	0,0691	0,1366	0,0893	103
0,0587	0,850	3	602	57	185	0,0500	0,0159	0,0864	0,0241	60
0,0587	0,875	2	608	51	186	0,0377	0,0106	0,0774	0,0165	53
0,0587	0,900	1	612	47	187	0,0208	0,0053	0,0713	0,0084	48
0,0587	0,925	0	635	24	188	0,0000	0,0000	0,0364	0,0000	24

Tabelle 10 - Auswertung *DB(p,D)*-Verfahren mit *m*=3, euklidischer Distanz und *D*=Ø

In der zweiten Tabelle wurde *D* um den Wert der Varianz erhöht.

D	p	TP	TN	FP	FN	Precision	Recall	FallOut	F_Measure	OI
0,0602	0,400	60	432	227	128	0,2091	0,3191	0,3445	0,2526	287
0,0602	0,500	46	463	196	142	0,1901	0,2447	0,2974	0,2140	242
0,0602	0,600	37	489	170	151	0,1787	0,1968	0,2579	0,1873	207
0,0602	0,700	31	528	131	157	0,1914	0,1649	0,1988	0,1771	162
0,0602	0,750	24	544	115	164	0,1727	0,1277	0,1745	0,1468	139
0,0602	0,800	11	576	83	177	0,1170	0,0585	0,1259	0,0780	94
0,0602	0,850	3	605	54	185	0,0526	0,0160	0,0819	0,0245	57
0,0602	0,875	1	610	49	187	0,0200	0,0053	0,0744	0,0084	50
0,0602	0,900	0	612	47	188	0,0000	0,0000	0,0713	0,0000	47

Tabelle 11 - Auswertung *DB(p,D)*-Verfahren mit *m*=3, euklidischer Distanz und *D*=Ø+σ^2

In der dritten Tabelle wurde *D* um den Wert der Varianz vermindert.

D	p	TP	TN	FP	FN	Precision	Recall	FallOut	F_Measure	OI
0,0572	0,400	69	420	239	119	0,2240	0,3670	0,3627	0,2782	308
0,0572	0,500	55	445	214	133	0,2045	0,2926	0,3247	0,2407	269
0,0572	0,600	40	474	185	148	0,1778	0,2128	0,2807	0,1937	225
0,0572	0,700	34	512	147	154	0,1878	0,1809	0,2231	0,1843	181
0,0572	0,750	28	530	129	160	0,1783	0,1489	0,1958	0,1623	157
0,0572	0,800	15	568	91	173	0,1415	0,0798	0,1381	0,1020	106
0,0572	0,850	4	600	59	184	0,0635	0,0213	0,0895	0,0319	63
0,0572	0,875	2	608	51	186	0,0377	0,0106	0,0774	0,0166	53
0,0572	0,900	1	610	49	187	0,0200	0,0053	0,0744	0,0084	50

Tabelle 12 - Auswertung *DB(p,D)*-Verfahren mit *m*=3, euklidischer Distanz und *D*=Ø-σ^2

Auf eine stärkere Erhöhung oder Minderung von *D* durch Varrierung um die Standardabweichung wurde aufgrund der Erfahrungen der vorhergehenden Experimente verzichtet, da hierdurch keine zusätzliche Optimierung erreicht wird. Dies wurde durch Stichproben geprüft.

Die Abbildung 53 zeigt die Ergebnisse der drei Experimentreihen noch einmal in der grafischen Analyse.

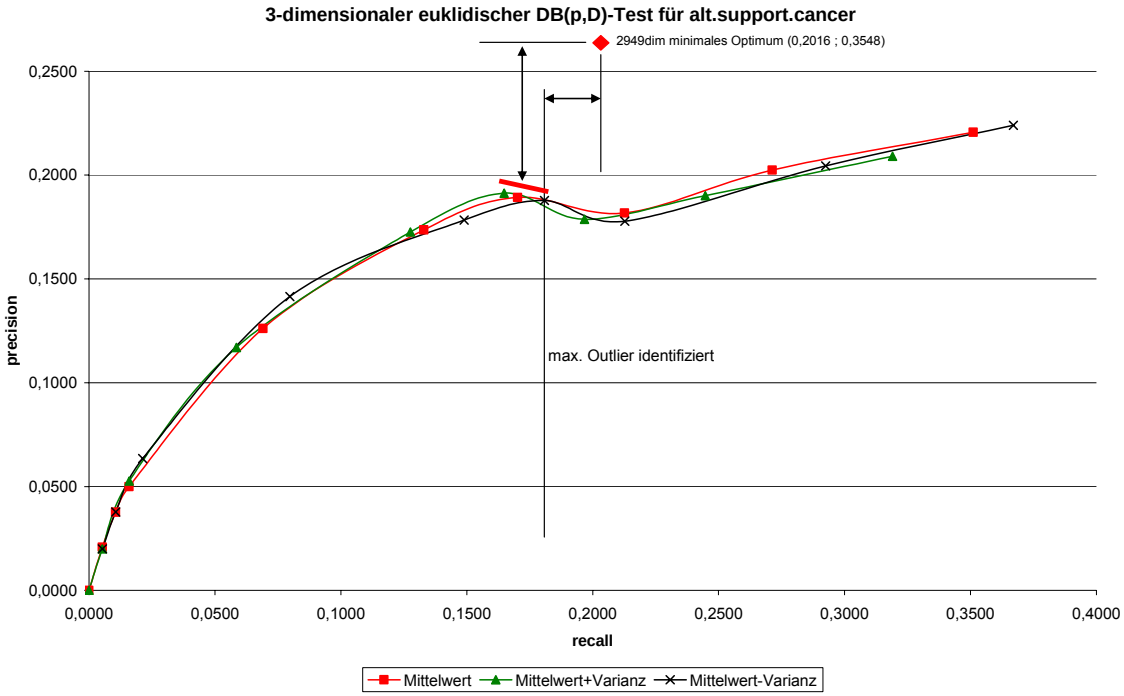


Abbildung 53 - *DB(p,D)*-Verfahren bei reduzierter Anzahl an Dimensionen

Abbildungsbeschreibung: Eine Dimensionsreduktion führt unabhängig von der Optimierung der Wertepaare für *p* und *D* zu wesentlich schlechteren Ergebnissen bei der Erkennung vorkategorisierter Objekte beim *DB(p,D)*-Verfahren.

Im Gegensatz zur vollen Anzahl an Dimensionen verändern sich die Ergebnisse nur undeutlich durch eine Variation von *D*, wobei eine Erhöhung von *p* sowohl die Genauigkeit als auch die Vollständigkeit steigert. Allerdings ist die Zahl maximaler Outlier in Bezug auf die Vorkategorisierung schnell erreicht.

Insgesamt verschlechtert die Reduzierung der Dimensionen die Ergebnisse bei entfernungsbasierten Verfahren, wenn die euklidische Distanz als Abstandsmaß eingesetzt wird. Sie liegen weit unter den nach Optimierung in der voll dimensionierten Menge möglichen Ergebnissen.

Die Reduzierung der Dimensionen führt allerdings zu einer starken Verbesserung der Performance des Verfahrens, weil sich die Komplexität um ggf. bis zu einer vollen Größenordnung verringert, sofern die Zahl der ursprünglichen Dimensionen im Vergleich zur Anzahl der Objekte gleich groß oder sogar größer ist. Dies ist bei der untersuchten Testmenge der Fall.

Optimierung des *DB(p,D)*-Verfahrens durch Einsatz der Kosinusdistanz

Weiterhin wurde untersucht, ob der Einsatz der Kosinusdistanz als alternatives Abstandsmaß die Ergebnisse des Verfahrens positiv beeinflusst. Die Tests wurden wiederum auf der Ursprungsmenge mit der vollen Zahl von 2949 Dimensionen durchgeführt. Die folgenden Tabellen zeigen die entsprechenden Ergebnisse. Die erste Tabelle zeigt die Ergebnisse für die durchschnittliche Distanz.

D	p	TP	TN	FP	FN	Precision	Recall	FallOut	F_Measure	OI
0,9500	0,750	85	442	217	103	0,2815	0,4521	0,3293	0,3469	302
0,9500	0,800	56	498	161	132	0,2581	0,2979	0,2443	0,2765	217
0,9500	0,850	36	576	83	152	0,3025	0,1915	0,1259	0,2345	119
0,9500	0,875	26	610	49	162	0,3467	0,1383	0,0744	0,1977	75
0,9500	0,900	13	633	26	175	0,3333	0,0691	0,0395	0,1145	39
0,9500	0,950	2	656	3	186	0,4000	0,0106	0,0046	0,0207	5
0,9500	0,975	0	658	1	188	0,0000	0,0000	0,0015	0,0000	1

Tabelle 13 - Auswertung *DB(p,D)*-Verfahren mit *m*=2949, Kosinusdistanz und *D*=Ø

In der zweiten Tabelle wurde *D* um den Wert der Varianz vermindert.

D	p	TP	TN	FP	FN	Precision	Recall	FallOut	F_Measure	OI
0,9447	0,850	46	534	125	142	0,2690	0,2447	0,1897	0,2563	171
0,9447	0,875	33	582	77	155	0,3000	0,1755	0,1168	0,2215	110
0,9447	0,900	24	617	42	164	0,3636	0,1277	0,0637	0,1890	66
0,9447	0,912	17	634	25	171	0,4048	0,0904	0,0379	0,1478	42
0,9447	0,925	9	646	13	179	0,4091	0,0479	0,0197	0,0857	22
0,9447	0,940	3	649	10	185	0,2308	0,0160	0,0152	0,0299	13
0,9447	0,950	3	655	4	185	0,4286	0,0160	0,0061	0,0308	7
0,9447	0,975	0	658	1	188	0,0000	0,0000	0,0015	0,0000	1

Tabelle 14 - Auswertung *DB(p,D)*-Verfahren mit *m*=2949, Kosinusdistanz und *D*=Ø-σ^2

In der dritten Tabelle wurde *D* um den Wert der Standardabweichung vermindert.

D	p	TP	TN	FP	FN	Precision	Recall	FallOut	F_Measure	OI
0,8773	0,900	114	243	416	74	0,2151	0,6064	0,6313	0,3175	530
0,8773	0,925	93	379	280	95	0,2493	0,4947	0,4249	0,3316	373
0,8773	0,950	64	519	140	124	0,3137	0,3404	0,2124	0,3265	204
0,8773	0,960	57	550	109	131	0,3434	0,3032	0,1654	0,3220	166
0,8773	0,968	45	576	83	143	0,3516	0,2394	0,1259	0,2848	128
0,8773	0,975	28	605	54	160	0,3415	0,1489	0,0819	0,2074	82
0,8773	0,990	3	651	8	185	0,2727	0,0160	0,0121	0,0302	11
0,8773	0,995	1	658	1	187	0,5000	0,0053	0,0015	0,0105	2

Tabelle 15 - Auswertung *DB(p,D)*-Verfahren mit *m*=2949, Kosinusdistanz und *D*=Ø-σ

Auf eine Erhöhung von *D* durch Variierung wurde aufgrund der Erfahrungen der vorhergehenden Experimente verzichtet, da hierdurch keine zusätzliche Optimierung erreicht wird. Dies wurde durch Stichproben geprüft. Die Abbildung 54 zeigt die Ergebnisse der drei Experimentreihen in der zu den vorhergehenden Experimenten vergleichbaren grafischen Analyse.

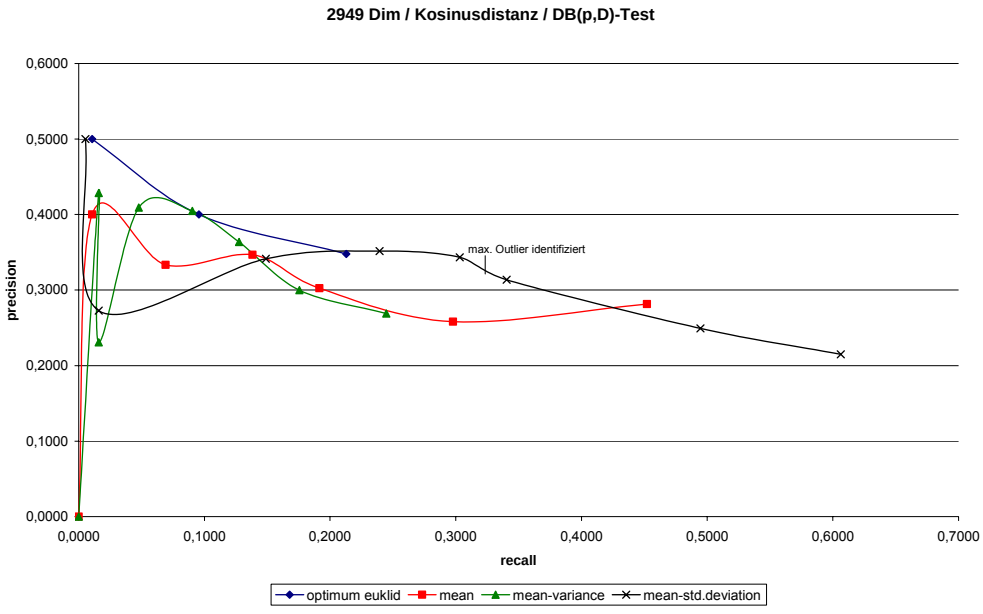


Abbildung 54 - *DB(p,D)*-Verfahren mit Kosinusdistanz als Abstandsmaß

Abbildungsbeschreibung: Der Einsatz der Kosinusdistanz führt zu vergleichbar guten Ergebnissen in der Erkennung vorkategorisierter Objekte, wobei im Bereich der Recall-Werte bei gleicher Precision Objekte vollständiger erkannt werden.

Diese zeigt für die Kosinusdistanz eine mit der euklidischen Distanz vergleichbare Güte der Ergebnisse, es wird jedoch keine signifikante Verbesserung erreicht. Zur Verdeutlichung ist die Optimierung der euklidischen Distanz zusätzlich in der Grafik verzeichnet.

Optimierung des *DB(p,D)*-Verfahrens durch Dimensionsreduktion bei Einsatz der Kosinusdistanz

Abschließend ist für das *DB(p,D)*-Verfahren zu klären, wie sich die Reduktion der Dimensionen bei Verwendung der Kosinusdistanz als Abstandsmaß auswirkt. Entsprechende Experimente führten zu den in den folgenden Tabellen aufgeführten Ergebnissen. Die erste Tabelle zeigt die Ergebnisse für die durchschnittliche Distanz.

D	p	TP	TN	FP	FN	Precision	Recall	FallOut	F_Measure	OI
0,5001	0,650	56	457	202	132	0,2171	0,2979	0,3065	0,2511	258
0,5001	0,700	43	540	119	145	0,2654	0,2287	0,1806	0,2457	162
0,5001	0,750	25	619	40	163	0,3846	0,1330	0,0607	0,1976	65
0,5001	0,775	13	641	18	175	0,4194	0,0691	0,0273	0,1187	31
0,5001	0,800	5	649	10	183	0,3333	0,0266	0,0152	0,0493	15
0,5001	0,850	1	658	1	187	0,5000	0,0053	0,0015	0,0105	2
0,5001	0,900	0	659	0	188	0,0000	0,0000	0,0000	0,0000	0

Tabelle 16 - Auswertung *DB(p,D)*-Verfahren mit *m*=3, Kosinusdistanz und *D*=Ø

In der zweiten Tabelle wurde *D* um den Wert der Varianz erhöht.

D	p	TP	TN	FP	FN	Precision	Recall	FallOut	F_Measure	OI
0,6532	0,500	54	443	216	134	0,2000	0,2872	0,3278	0,2358	270
0,6532	0,525	38	491	168	150	0,1845	0,2021	0,2549	0,1929	206
0,6532	0,550	34	540	119	154	0,2222	0,1809	0,1806	0,1994	153
0,6532	0,575	23	573	86	165	0,2110	0,1223	0,1305	0,1549	109
0,6532	0,600	18	604	55	170	0,2466	0,0957	0,0835	0,1379	73
0,6532	0,750	0	659	0	188	0,0000	0,0000	0,0000	0,0000	0

Tabelle 17 - Auswertung *DB(p,D)*-Verfahren mit *m*=3, Kosinusdistanz und *D*=Ø+σ^2

In der dritten Tabelle wurde *D* um den Wert der Varianz vermindert.

D	p	TP	TN	FP	FN	Precision	Recall	FallOut	F_Measure	OI
0,3488	0,750	66	466	193	122	0,2548	0,3511	0,2929	0,2953	259
0,3488	0,775	58	519	140	130	0,2929	0,3085	0,2124	0,3005	198
0,3488	0,788	53	543	116	135	0,3136	0,2819	0,1760	0,2969	169
0,3488	0,800	50	562	97	138	0,3401	0,2660	0,1472	0,2985	147
0,3488	0,825	41	579	80	147	0,3388	0,2181	0,1214	0,2654	121
0,3488	0,850	33	601	58	155	0,3626	0,1755	0,0880	0,2366	91
0,3488	0,875	25	624	35	163	0,4167	0,1330	0,0531	0,2016	60
0,3488	0,900	16	633	26	172	0,3810	0,0851	0,0395	0,1391	42
0,3488	0,925	10	645	14	178	0,4167	0,0532	0,0212	0,0943	24
0,3488	0,938	5	647	12	183	0,2941	0,0266	0,0182	0,0488	17
0,3488	0,950	3	650	9	185	0,2500	0,0160	0,0137	0,0300	12
0,3488	0,975	0	659	0	188	0,0000	0,0000	0,0000	0,0000	0

Tabelle 18 - Auswertung *DB(p,D)*-Verfahren mit *m*=3, Kosinusdistanz und *D*=Ø-σ^2

In der vierten Tabelle wurde *D* um den Wert der Standardabweichung vermindert.

D	p	TP	TN	FP	FN	Precision	Recall	FallOut	F_Measure	OI
0,1109	0,950	49	530	129	139	0,2753	0,2606	0,1958	0,2678	178
0,1109	0,960	47	549	110	141	0,2994	0,2500	0,1669	0,2725	157
0,1109	0,970	39	570	89	149	0,3047	0,2074	0,1351	0,2468	128
0,1109	0,973	27	593	66	161	0,2903	0,1436	0,1002	0,1922	93
0,1109	0,975	23	609	50	165	0,3151	0,1223	0,0759	0,1762	73
0,1109	0,990	16	633	26	172	0,3810	0,0851	0,0395	0,1391	42
0,1109	0,995	15	639	20	173	0,4286	0,0798	0,0303	0,1345	35
0,1109	0,999	10	646	13	178	0,4348	0,0532	0,0197	0,0948	23
0,1109	1,000	10	648	11	178	0,4762	0,0532	0,0167	0,0957	21

Tabelle 19 - Auswertung *DB(p,D)*-Verfahren mit *m*=3, Kosinusdistanz und *D*=Ø-*σ*

Die Abbildung 54 zeigt die Ergebnisse der drei Experimentreihen in der zu den vorhergehenden Experimenten vergleichbaren grafischen Analyse.

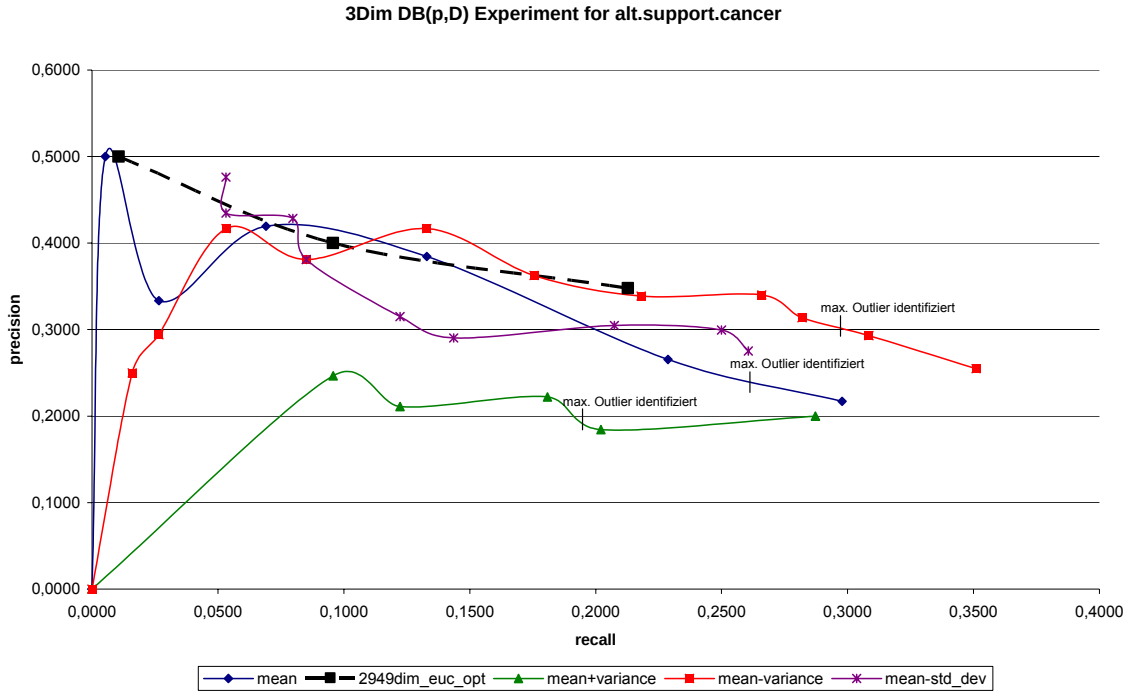


Abbildung 55 - *DB(p,D)*-Verfahren mit Kosinusdistanz und reduzierten Dimensionen

> *Abbildungsbeschreibung:* Der Einsatz der Kosinusdistanz bei reduzierter Dimensionszahl führt zu wesentlich besseren Ergebnissen, als der Einsatz der euklidischen Distanz nach Dimensionsreduktion. Allerdings zeigt der Vergleich mit den Textsplit-Experimentreihen, dass dies nicht zu verallgemeinern ist (vgl. auch Abbildung 60)

Zuallererst ist festzustellen, dass die Auswirkungen der Reduktion der Dimensionen stark von denen bei der euklidischen Distanz abweichen. Die Güte ist leicht besser und vergleichbar mit den Ergebnissen der Verfahrensanwendung auf eine Testmenge mit allen Dimensionen. Darüber hinaus wirkt sich, wie bereits beschrieben, der Performancegewinn positiv aus. Möglicherweise ändert die Reduktion durch das SVD-Verfahren die Abstände zwischen den Objekten stärker, als deren ursprünglichen Winkel zueinander, da sich auch die Charakteristik der Kurven für variierende *p*-Werte hier –im Gegensatz zur Reduktion mit euklidischer Distanz – nicht deutlich verändert.

Zusammenfassung der Untersuchungsergebnisse für das *DB(p,D)*-Verfahren

Insgesamt ist die Optimierung der Ergebnisse dieses Verfahrens durch geeignete Parameterwahl möglich und sinnvoll. Dies setzt jedoch die (teilweise) Kenntnis über eine Vorkategorisierung voraus und ist vergleichsweise aufwendig.

Die Reduktion der Anzahl an Dimensionen verschlechtert bei euklidischer Distanz die Ergebnisse.

Die Wahl eines alternativen Abstandsmaßes, hier der Kosinusdistanz, bringt keine signifikanten Änderungen. Allerdings kann bei Einsatz der Kosinusdistanz eine Reduktion der Anzahl an Dimensionen vorgenommen werden, ohne dass dies negative Auswirkungen auf die Ergebnisqualität zu haben scheint. Damit ergibt sich ein Pfad zur Optimierung der Performance, zumal im niedrigdimensionalen Bereich für das *DB(p,D)*-Verfahren mit dem Zellenalgoritmus eine weitere Steigerung der Performance möglich ist [6].

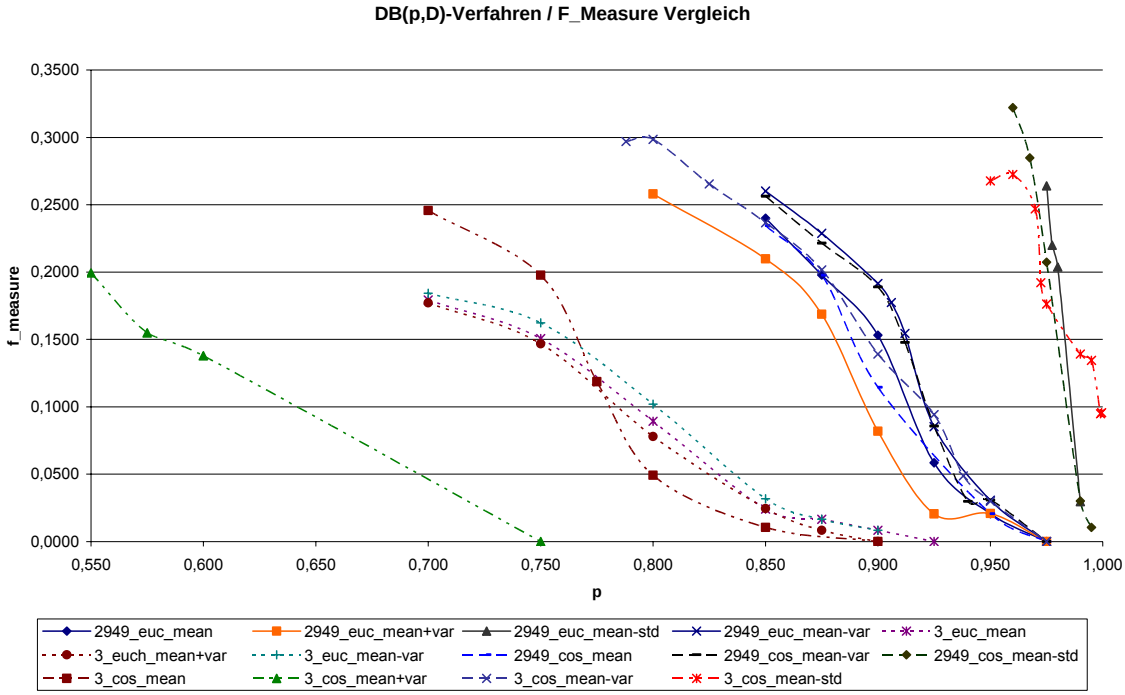


Abbildung 56 - *DB(p,D)*-Verfahren im F_measure Vergleich

Abbildungsbeschreibung: Durch die Auswahl einer Parameterkombination mit einem maximalen Wert für F_measure kann der Anwender eine optimale (gleich gewichtete) Balance zwischen genauer und vollständiger Erkennung erreichen. In der Grafik wurden Parameter, die mehr als die vorkategorisierten Outlier erkennen, nicht berücksichtigt.

Abbildung 56 zeigt den Vergleich der F_Measure Werte für alle Experimentreihen in Bezug zu den gewählten Werten für *p*, wobei die Datenreihe einer Linie jeweils einer fest gewählten Distanz *D* entspricht. Hierbei wurden jedoch nur Wertepaare berücksichtigt, die zur Erkennung der maximalen vorkategorisierten Outlieranzahl führten, da ansonsten der schnelle Anstieg des F_Measure Ergebnisses durch den hohen Wert der Vollständigkeit bei geringer Genauigkeit die Interpretation verfälschen würde.

Optimierung des *DB(p,D)*-Verfahrens durch Textsplitting

Um die Auswirkungen des bereits für das *D(k,n)*-Verfahren beschriebenen Textsplittings auf das *DB(p,D)*-Verfahren zu untersuchen, wurden bis auf die „Subject“-Zeile und die „From“-Zeile, welche den Autor des Artikels enthält, alle Headerzeilen aus den Artikeln entfernt. Die Ergebnismenge wurde erneut vektorisiert. Die resultierenden Textvektoren mit 2.555 Dimensionen wurden sodann den gleichen *D(p,D)*-Experimenten unterzogen, und dies wurde zudem nach SVD-Reduktion auf eine dreidimensionale Menge jeweils mit Einsatz der euklidischen Distanz und der Kosinusdistanz als Abstandsmaß wiederholt.

Abbildung 57 zeigt die Ergebnisse der Experimentreihen nach Optimierung der Parameterkombinationen für *D* und *p* bei voller Dimensionszahl (2555) bei Anwendung der euklidischen Distanz.

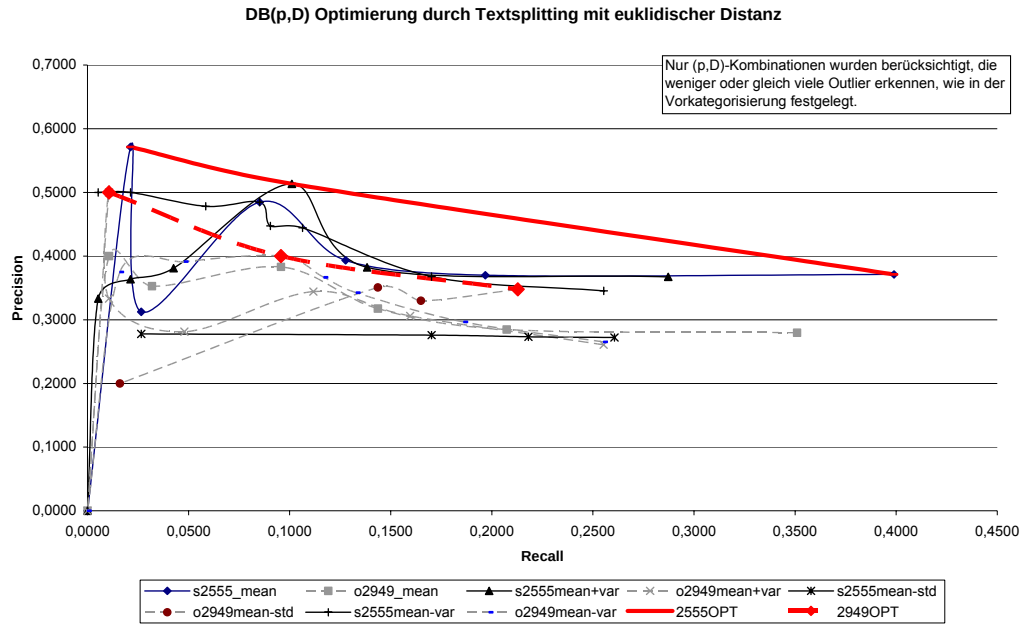


Abbildung 57 - *DB(p,D)*-Verfahren mit Textsplitting und euklidischer Distanz bei 2555 Dimensionen

Abbildungsbeschreibung: Bei euklidischer Distanz lässt sich die Optimierungsgrenze für die Parameterwahl noch einmal deutlich verbessern.

Hierbei ist zu erkennen, dass sich durch das Splitting für das *DB(p,D)*-Verfahren verbesserte Ergebnisse erzielen lassen. Dies zeigt sich insbesondere beim eingezeichneten Vergleich der optimalen Parameterkombinationen beider Anwendungen (mit und ohne Splitting). Den Einsatz der Kosinusdistanz als alternatives statistisches Maß zeigt im Ergebnis die Abbildung 58. Auch hier sind Verbesserungen im Vergleich zur Anwendung des Verfahrens ohne das Splitting sichtbar. Allerdings lässt sich die initiale Verbesserung mit der euklidischen Distanz als Abstandsmaß nicht signifikant steigern.

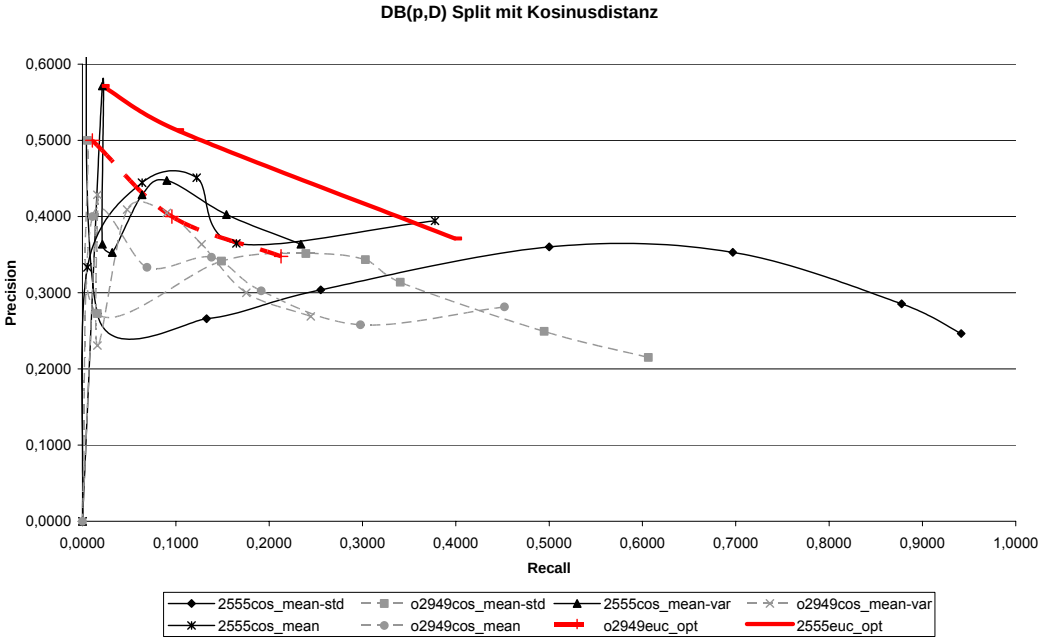


Abbildung 58 - *DB(p,D)*-Experiment mit Textsplitting und Kosinusdistanz bei 2555 Dimensionen

Abbildungsbeschreibung: Die Verbesserung durch das Textsplitting lässt sich mit Einsatz der Kosinusdistanz nicht nochmals signifikant steigern.

Abbildung 59 zeigt die Verfahrensergebnisse nach Anwendung des Textsplittings und nach einer Dimensionsreduktion der so erhaltenen Datenmenge mittels SVD. Hierbei wird deutlich, dass nicht nur die Ergebnisse der Auswertung deutlich schlechter sind, als ohne Textsplitting, sondern dass die Reduktion der Dimensionen zu einem signifikanten Qualitätsverlust führt.

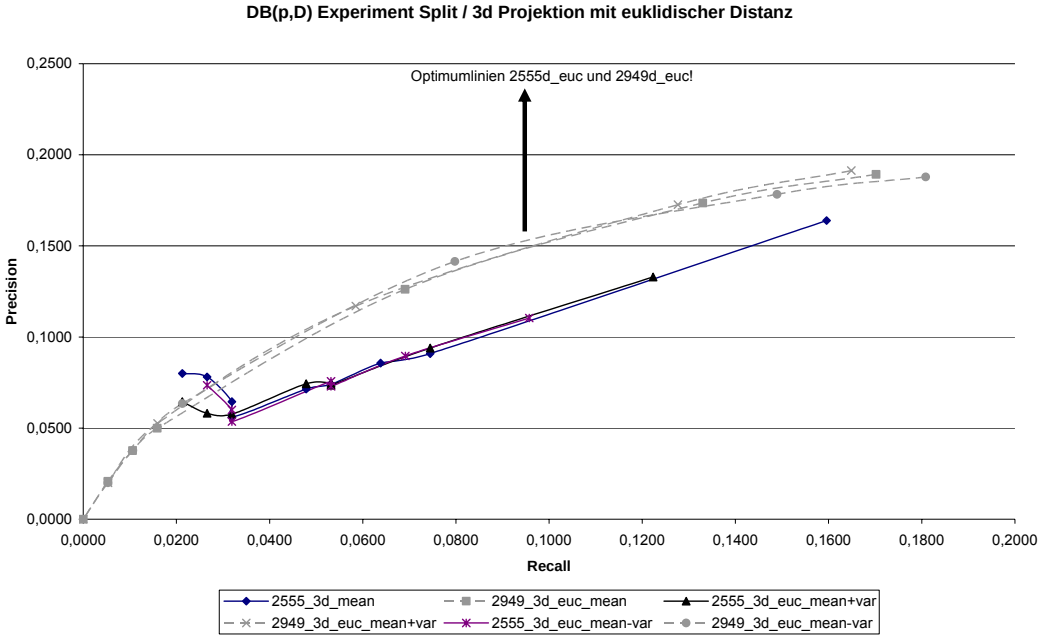


Abbildung 59 - *DB(p,D)*-Experiment mit Textsplitting, euklidischer Distanz und reduzierten Dimensionen

Abbildungsbeschreibung: Auch für die mittels Textsplitting behandelte Testmenge gilt, dass die Dimensionsreduktion zu deutlichen Ergebnisverschlechterungen bei der Erkennung vorkategorisierter Objekte führt.

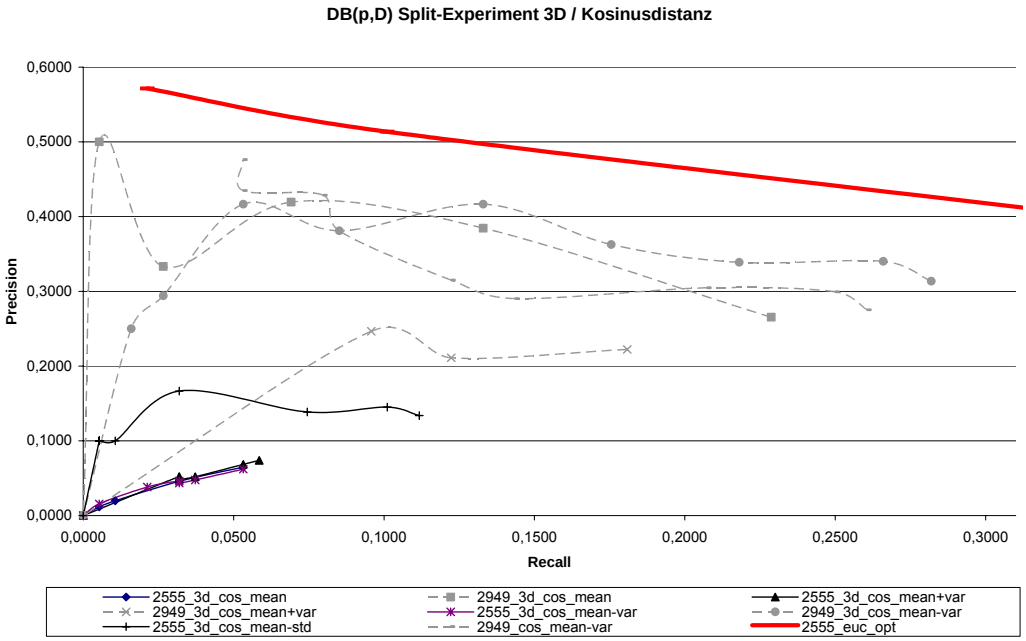


Abbildung 60 - *DB(p,D)*-Experiment mit Textsplitting und Kosinusdistanz bei reduzierten Dimensionen

Abbildungsbeschreibung: Während der Einsatz der Kosinusdistanz auf der dimensionsreduzierten Datenmenge vor dem Textsplitting zu vergleichsweise sehr guten Ergebnissen der Erkennung vorkategorisierter Objekte führte, ergibt das gleiche Experiment nach dem Textsplitting ein völlig anderes Ergebnis.

Abbildung 60 verdeutlicht eine sehr interessante und vor allem unerwartete Erkenntnis. Während die Anwendung der Kosinusdistanz auf der dimensionsreduzierten Datenmenge, welche aus der Ursprungsmenge ohne Textsplitting gewonnen wurde, trotz der Reduktion qualitativ sehr hochwertige und mit der volldimensionalen Menge durchaus vergleichbare Ergebnisse lieferte, fällt die Qualität der Ergebnisse in der reduzierten Menge nach Textsplitting sehr stark zurück. Insofern erscheint der Optimierungspfad durch Dimensionsreduktion bei Einsatz der Kosinusdistanz als nicht durchgängig belastbar, wobei jedoch eine formale Analyse zur abschließenden Betrachtung empirisch nicht vorweggenommen werden kann.

Insgesamt zeigt sich, dass eine Vorverarbeitung der Datenmenge mittels Textsplitting beim *DB(p,D)*-Verfahren durchaus zu verbesserten Ergebnissen bei der Erkennung vorkategorisierter Outlier führen kann. Dieser Effekt ist jedoch auf die nicht reduzierten Datenmengen beschränkt, wobei die Ergebnisse der unterschiedlichen eingesetzten Distanzmaße durchaus vergleichbar sind.

6.3.3. *LOF(MinPts)* Experiment

***LOF(MinPts)*-Experiment mit euklidischer Distanz**

Für die Analyse der Testdatenmenge von alt.support.cancer (vgl. Kapitel 6.2.2) mit der Betrachtung der lokalen Dichte der Objekte im Suchraum wurde der *LOF(MinPts)*-Operator des YALE Outlier-Plugins [105] eingesetzt. Dabei wurde ein *MinPts*-Intervall zwischen 10 und 20 angenommen. Die untere Grenze ergibt sich aus den starken statistischen Schwankungen des *LOF*-Wertes unterhalb eines *MinPts*-Wertes von 10 (vgl. Kapitel 2.5.1) und die obere Grenze aus der relativ spärlichen Besetzung des Datenraumes, die keine großen Clusteransammlungen vermuten lässt.

Zum besseren Vergleich der jeweiligen Ergebnisse wurde ein *LOF*-Experiment auf dem ursprünglichen Datenraum durchgeführt, der nach Anwendung des WVtools auf die Datenmenge erstellt wurde. Dieser umfasst 2.949 Dimensionen.

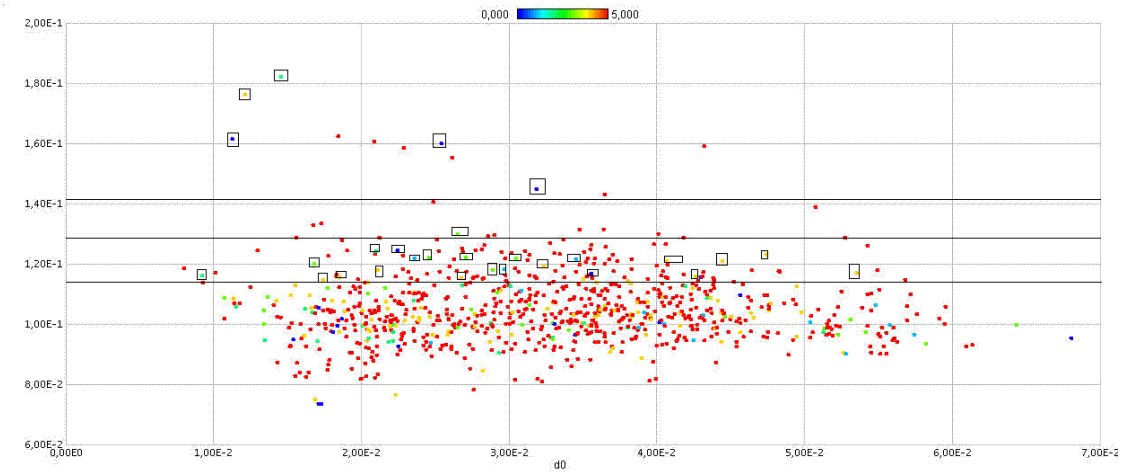


Abbildung 61 - *LOF*-Auswertung auf der Ursprungsmenge (2.949 Dimensionen)

Abbildungsbeschreibung: Auf der *y*-Achse sind die *LOF*-Werte der Objekte abgetragen und die ein-, zwei- und dreifache Standardabweichung vom Durchschnitt aller *LOF*-Werte aller Objekte durch Linien gekennzeichnet. Die breite Streuung der Dichtewerte lässt auf eine spärliche Besetzung des Datenraumes schließen.

Die Abbildung 61 zeigt die grafische Darstellung des Ergebnisses der LOF-Werte des Experimentes auf der Ursprungsmenge. Auf der *x*-Achse ist dabei die Dimension d_0 nach der Anwendung des SVD-Verfahrens zur Dimensionsreduktion abgetragen, wobei die *LOF*-Berechnung auf den 2.949 Ursprungsdimensionen stattfand. Auf der *y*-Achse ist der *LOF*-Wert abgetragen und die Farbe der Punkte zeigt die Kategorie, wobei alle nicht als Outlier kategorisierten Objekte der Testmenge rot dargestellt sind. Der durchschnittliche *LOF*-Wert aller Objeke beträgt 0,105 mit einer Standardabweichung von 0,012. Im Vergleich zu Kapitel 2.5, in welchem von einem *LOF*-Wert für Objekte tief innerhalb eines Clusters um den Wert 1 ausgegangen wird, zeigt die Auswertung des Experiments hier implementierungsabhängig abweichende Ergebnisse, die jedoch genauso ausdrucksfähig sind (siehe hierzu auch Kapitel 5.2.4). Die drei waagerechten Linien im Diagramm zeigen jeweils die einfache, zweifache und dreifache Standardabweichung des *LOF*-Durchschnitts über eben diesem Durchschnitt. Zusätzlich

wurden die Objekte oberhalb der Standardabweichungswerte, welche gleichzeitig in der Testmenge als Outlier-Kandidaten kategorisiert wurden und daher farbig abweichen, durch Markierungen hervorgehoben.

Die Darstellung deutet darauf hin, dass aufgrund der relativ gleichmäßig verteilten *LOF*-Werte die Ausgangsmenge im 2.949-dimensionalen Raum relativ spärlich besetzt ist. Diese Spärlichkeit erschwert eine Aussage über die Objekte auf Basis lokaler Dichten. Bereits in Kapitel 2.6 wurde darauf hingewiesen, dass in manchen Datenräumen die Dichteabweichungen so gering sind, dass eine Identifizierung anhand von Schwellwerten mit hinreichender Signifikanz nicht mehr möglich sein kann. Klar wird auf jeden Fall, dass nicht nur mindestens gleich viele Objekte ohne Kategorisierung in der Testmenge *LOF*-Werte oberhalb der Standardabweichungsgrenzen auftauchen, sondern dass sich auch eine hohe Zahl an kategorisierten Objekten dicht im Durchschnitt der *LOF*-Werte befindet.

Immerhin 28 kategorisierte Outlier-Kandidaten werden durch den Test erkannt, wobei sich von diesen 5 Objekte oberhalb der dreifachen Standardabweichung vom *LOF*-Durchschnitt befinden, ein Objekt oberhalb der zweifachen und 22 Objekte oberhalb der einfachen Standardabweichung von diesem Durchschnitt. Es werden also von 188 kategorisierten Outliern knapp 15% erkannt. Auf der anderen Seite werden jedoch auch 117 + 12 + 6 = 135 nicht kategorisierte Objekte innerhalb der 1-, 2- und 3-fachen Standardabweichung oberhalb des *LOF*-Durchschnittes identifiziert, was einem Anteil von fast 83% der entdeckten Outlier entspricht.

Im Weiteren wurde das Experiment mit einem *MinPts*-Intervall von 10 bis 40 und einem *MinPts*-Intervall von 20 bis 100 wiederholt. Das letztere Intervall wurde gewählt, um auch große Cluster bei der Betrachtung zuzulassen, obwohl deren Vorkommen nicht wahrscheinlich ist. Größere *MinPts*-Werte wurden nicht abgewandt, da keine zusätzlichen Erkenntnisse zu erwarten waren. Bei allen Experimenten zeigte sich, dass das Intervall von 10 bis 40 die gleichen Ergebnisse erzielte, wie das Intervall von 10 bis 20. Diese sind in der folgenden Tabelle noch einmal festgehalten:

LOF Schwelle	TP	TN	FP	FN	Precision	Recall	Fall Out	F_Measure
0,117	21	577	82	167	0,2039	0,1117	0,1244	0,1443
0,129	6	645	14	182	0,3000	0,0319	0,0212	0,0577
0,141	5	653	6	183	0,4545	0,0266	0,0091	0,0503

Tabelle 20 - Auswertung *LOF*-Verfahren mit *MinPts*=[10;20], *m*=2949 und euklidischer Distanz

Die Ergebnisse des *LOF*-Verfahrens für die euklidische Distanz auf der Ursprungsmenge sind insgesamt wesentlich schlechter, als die von reinen entfernungsbasierten Verfahren. Dies war aufgrund der Spärlichkeit der Testmenge auch so zu erwarten. *MinPts*-Intervalle mit geringen *MinPts*-Werten erreichen bei der „Selektion“ weniger Objekte als Outlier und durch einen hoch angesetzten *LOF*-Schwellwert (hier die dreifache Standardabweichung des *LOF*-Durchschnittes aller Objekte der Testmenge) eine höhere Genauigkeit, wobei die Vollständigkeit der Erkennung durch das Verfahren durchweg sehr gering ist. *MinPts*-Intervalle mit größeren *MinPts*-Werten erreichen insgesamt schlechtere Ergebnisse, wie aus der Tabelle für das *MinPts*-Intervall von 20 bis 100 zu entnehmen ist.

LOF Schwelle	TP	TN	FP	FN	Precision	Recall	Fall Out	F_Measure
0,056	23	570	89	165	0,2054	0,1223	0,1351	0,1533
0,06	11	630	29	177	0,2750	0,0585	0,0440	0,0965
0,064	5	643	16	183	0,2381	0,0266	0,0243	0,0479

Tabelle 21 - Auswertung *LOF*-Verfahren mit *MinPts*=[20;100], *m*=2949 und euklidischer Distanz

Ein wesentlicher Vorteil des *LOF*-Verfahens liegt darin, dass die einzelnen *LOF*-Werte der Objekte eine einfache Einordnung in der Gesamtmenge, z.B. anhand des Durchschnitts des *LOF* und der Abweichung hiervon, erlauben. Gleichzeitig ist keine echte Optimierung notwendig, sofern sinnvolle *MinPts*-Intervalle eingesetzt werden, da das Verfahren aus den Intervallen die jeweils maximalen *LOF*-Werte für jedes Objekt bereitstellt. Eine aufwendige Optimierung mit Interpretation der Auswirkung von Parameteränderungen entfällt.

Optimierung durch Dimensionsreduktion bei euklidischer Distanz

Um die Möglichkeiten einer Ergebnisoptimierung für das *LOF*-Verfahren zu untersuchen, wurde die Testmenge durch Singular Value Decomposition auf drei Dimensionen reduziert und die *LOF*-Experimente mit den genannten *MinPts*-Intervallen wiederholt.

Abbildung 62 zeigt die grafische Auswertung eines *LOF*-Experiments mit einem *MinPts*-Intervall von 10 bis 20 auf der alt.support.cancer. Testmenge, die mittels SVD auf drei Dimensionen reduziert wurde.

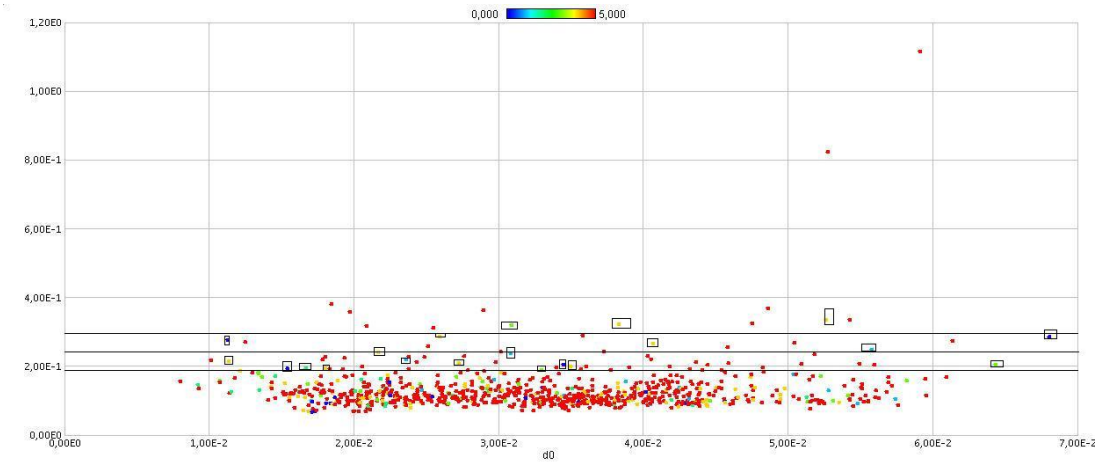


Abbildung 62 - *LOF* Analyse auf 3-dimensionaler Testmenge nach SVD

Abbildungsbeschreibung: Die auf drei Dimensionen reduzierte Datenmenge zeigt im Vergleich zur Abbildung 61 eine wesentlich geringere Streuung der Dichtewerte. Dies war bezogen auf Abbildung 46 auch zu erwarten.

Nach der Berechnung der *LOF*-Werte auf der auf drei Dimensionen reduzierten Datenmenge ergibt sich eine wesentlich bessere Sichtbarkeit abweichender Dichtewerte von potentiellen Outliern. Dies ist aus der grafischen Betrachtung der Datenmenge in drei Dimensionen in Abbildung 46 auch plausibel. Der durchschnittliche *LOF*-Wert aller Objekte der dimensionsreduzierten Testmenge beträgt im beschriebenen *MinPts*-Intervall 0,128 mit einer Standardabweichung von 0,06. In der Grafik sind wiederum die einfache, zweifache und dreifache Standardabweichung über dem Durchschnitt durch Linien gekennzeichnet. Von den kategorisierten Outliern werden entsprechend dieser Schwellwerte 12 + 5 + 3 = 20 Objekte erkannt. Des Weiteren werden 31 + 6 + 10 = 47 nicht kategorisierte Objekte als potentielle Outlier identifizierbar. Somit werden hier nur 10,6 % der kategorisierten 188 Outlier erkannt, allerdings mit einer geringeren Quote von 70% von Objekten, die als Outlier qualifizierbar sind, jedoch a priori keiner Kategorie zugeordnet wurden. Auf der anderen Seite werden im Bereich der deutlichen Outlier-Charakteristika nunmehr nur noch drei Objekte korrekt identifiziert, anstatt 5 auf der 2.949-dimensionalen Datenmenge. Oberhalb der dreifachen Standardabweichung vom Durchschnitts-*LOF*-Wert wurden da lediglich 6 nicht kategorisierte Objekte identifizierbar, was einem Verhältnis von 5:6 entspricht. Nach Reduktion ist oberhalb derselben Grenze ein Verhältnis von 3:10 zu beobachten. Die folgende Tabelle zeigt die Ergebnisse für das *MinPts*-Intervall von 10 bis 20.

LOF Schwelle	TP	TN	FP	FN	Precision	Recall	Fall Out	F_Measure
0,188	19	611	48	169	0,2836	0,1011	0,0728	0,1491
0,248	7	643	16	181	0,3043	0,0372	0,0024	0,0663
0,308	3	649	10	185	0,2308	0,0160	0,0152	0,0299

Tabelle 22 - Auswertung *LOF*-Verfahren mit *MinPts*=[10;20], *m*=3 und euklidischer Distanz

Die Dimensionsreduktion zeigt ein leicht besseres Verhältnis von Genauigkeit und Vollständigkeit in beiden Tests, wie auch die Tabelle mit den Ergebnissen des *MinPts*-Intervalls von 20 bis 100 zeigt.

LOF Schwelle	TP	TN	FP	FN	Precision	Recall	Fall Out	F_Measure
0,098	20	617	42	168	0,3226	0,1064	0,0637	0,1600
0,129	6	645	14	182	0,3000	0,0319	0,0212	0,0577
0,16	2	651	8	186	0,2000	0,0106	0,0121	0,0201

Tabelle 23 - Auswertung *LOF*-Verfahren mit *MinPts*=[20;100], *m*=3 und euklidischer Distanz

Die Annahme von *MinPts*-Intervallen mit größeren Werten für die obere Grenze von *MinPts* führt zu deutlich besseren Ergebnissen. Die Genauigkeit des Verfahrens bei hohen *LOF*-Schwellwerten erreicht allerdings nicht die gleiche Qualität, wie bei Tests über alle Dimensionen.

Optimierung des *LOF(MinPts)*-Experimentes durch Einsatz der Kosinus-Distanz

Im experimentellen Aufbau wurde weiterhin untersucht, ob durch den Einsatz der Kosinusdistanz als alternatives statistisches Abstandsmaß eine Optimierung der Ergebnisse erreichbar ist. Die Experimente wurden für die gleichen *MinPts*-Intervalle durchgeführt. Sie sind in den beiden nachstehenden Tabellen zusammengefasst.

LOF Schwelle	TP	TN	FP	FN	Precision	Recall	Fall Out	F_Measure
0,142	15	584	75	173	0,1667	0,0798	0,1138	0,1079
0,169	5	640	19	183	0,2083	0,0266	0,0289	0,0472
0,196	4	651	8	184	0,3333	0,0213	0,0121	0,0400

Tabelle 24 - Auswertung *LOF*-Verfahren mit *MinPts*=[10;20], *m*=2949 und Kosinusistanz

LOF Schwelle	TP	TN	FP	FN	Precision	Recall	Fall Out	F_Measure
0,067	18	601	58	170	0,2368	0,0957	0,0880	0,1364
0,077	11	637	22	177	0,3333	0,0585	0,0338	0,0995
0,087	5	646	13	183	0,2778	0,0266	0,0798	0,0486

Tabelle 25 – Auswertung *LOF*-Verfahren mit *MinPts*=[20;100], *m*=2949 und Kosinusdistanz

In dem Intervall mit kleinen *MinPts*-Werten sind die Ergebnisse in Bezug auf Genauigkeit und Vollständigkeit wesentlich schlechter, als beim Einsatz der euklidischen Distanz in vergleichbaren *MinPts*-Intervallen. Dagegen sind die Ergebnisse deutlich besser, wenn große *MinPts*-Werte zum Einsatz kommen. Grund für letzteres ist möglicherweise, dass bei spärlicher Verteilung die Betrachtung der Winkel zwischen den Objekten (welche ja Textvektoren repräsentieren) zu wesentlich besser sichtbaren großen Clustern führt, als die Betrachtung des Abstands zwischen den Objekten.

***LOF(MinPts)*-Experiment mit Dimensionsreduktion bei Einsatz der Kosinus-Distanz**

Verbleibend ist die Frage zu klären, wie sich eine Dimensionsreduktion auf die Ergebnisse auswirkt, wenn die Kosinusdistanz verwendet wird. Die folgenden Tabellen fassen die Ergebnisse der Experimente zusammen.

LOF Schwelle	TP	TN	FP	FN	Precision	Recall	Fall Out	F_Measure
0,397	16	626	33	172	0,3265	0,0851	0,0501	0,1350
0,6	5	642	17	183	0,2273	0,0266	0,0258	0,0476
0,803	4	648	11	184	0,2666	0,0210	0,0167	0,0389

Tabelle 26 - Auswertung *LOF*-Verfahren mit *MinPts*=[10;20], *m*=3 und Kosinusistanz

LOF Schwelle	TP	TN	FP	FN	Precision	Recall	Fall Out	F_Measure
0,227	19	608	51	169	0,2715	0,1011	0,0774	0,1473
0,341	6	638	21	182	0,2222	0,0319	0,0318	0,0558
0,455	3	649	10	185	0,2308	0,0159	0,0152	0,0298

Tabelle 27 - Auswertung *LOF*-Verfahren mit *MinPts*=[20;100], *m*=3 und Kosinusdistanz

Für kleine *MinPts*-Werte werden bei geringen *LOF*-Schwellwerten Ergebnisse mit im Vergleich zu anderen Konstellationen von Dimension, Distanzmaß und Schwellwert gutem Verhältnis zwischen Genauigkeit und Vollständigkeit erzielt. Sie sind jedoch etwas schlechter als beim Einsatz der euklidischen Distanz im in der Dimension reduzierten Suchraum der Testmenge. Die Annahme hoher *LOF*-Schwellwerte führ hier allerdings zu einer Verschlechterung der Ergebnisse.

Zusammenfassung der Ergebnisse der *LOF*-Experimente

Die Abbildung 63 fasst die Ergebnisse der *LOF*-Experimente noch einmal grafisch zusammen. Als Vergleich wurde die Optimierungsgrenze des *DB(p,D)*-Verfahrens für euklidische Distanz herangezogen.

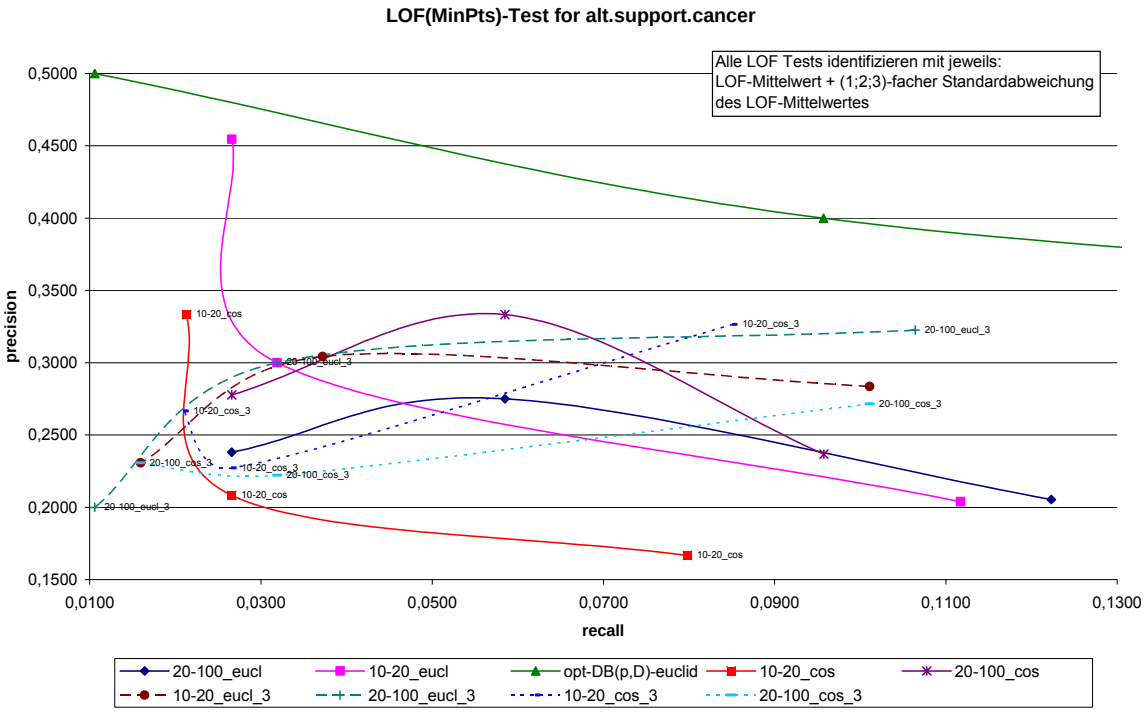


Abbildung 63 - *LOF*-Experiment Ergebnisse im Vergleich

Abbildungsbeschreibung: Die Ergebnisse der Kombinationen aus Distanzmaß, Dimensionszahl und *MinPts*-Intervall sind hier grafisch zusammengefasst, wobei die Optimierung des *DB(p,D)*-Verfahrens als Vergleich eingezeichnet ist.

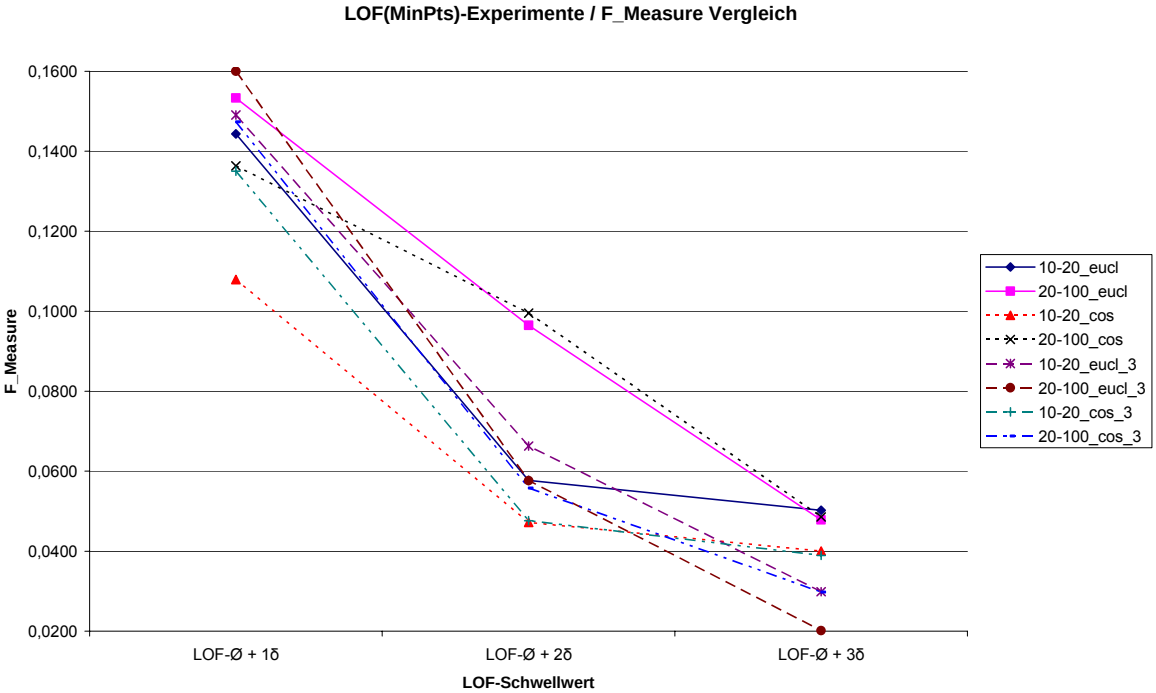


Abbildung 64 - *LOF*-Verfahren: F_Measure Vergleich

Abbildungsbeschreibung: Auch für das *LOF*-Verfahren wurden die F_Measure Werte verglichen (vgl. auch Abbildung 50).

Es ist festzustellen, dass die Messung lokaler Dichten insgesamt nicht die Qualität der rein entfernungsbasierten Verfahren erreichen kann. Dies ist für nahezu alle Bereiche der Fall, wobei der Vergleich mit anderen Verfahren nach deren Optimierung gezogen wird. Einer der deutlichen Vorteile des *LOF*-Verfahrens ist, dass über den *MinPts*-Wert eine Optimierung einfach möglich ist und ggf. auch ohne Kenntnis der vorkategorisierten Menge zu akzeptablen Ergebnissen führt. Dies ist bei rein entfernungsbasierten Verfahren mit Sicherheit schwerer zu erreichen. Die Reduzierung der Dimensionen verbessert die Vollständigkeit des Ergebnisses und auch das Verhältnis zwischen Genauigkeit und Vollständigkeit. Hier war bei reinen entfernungsbasierten Verfahren zumindest im Bereich der euklidischen Distanz ein anderer Trend ablesbar. Der Einsatz der Kosinusdistanz führte auch dort zu Ergebnisverbesserungen. Eine hohe Genauigkeit der Erkennung von vorkategorisierten Outliern ist beim *LOF*-Verfahren nur beim Einsatz hoher Schwellwerte gegeben. Bei diesen wird jedoch nur eine geringe Vollständigkeit erreicht. Unter Ausschöpfung aller Optimierungsmöglichkeiten über Distanzmaße und Dimensionskombinationen hinweg führt der Einsatz der Kosinusdistanz insgesamt beim *LOF*-Verfahren nicht zu sehr viel besseren Ergebnissen, als der Einsatz der euklidischen Distanz. Die Abbildung 64 zeigt abschließend noch einmal den Vergleich der Werte für f_measure der einzelnen Experimente mit *LOF*-Schwellwerten, die jeweils die einfache, zweifache und dreifache Standardabweichung über dem Durchschnitts-*LOF*-Wert aller Objekte in der Datenmenge liegen. Die besten Ergebnisse in dieser Betrachtung werden durch den Ansatz niedriger Schwellwerte und mit reduzierter Dimensionszahl in der Datenmenge unter Einsatz der euklidischen Distanz erreicht.

Optimierung des *LOF*-Verfahrens durch Textsplitting

Um die Auswirkungen des bereits für das *D(k,n)*- und *DB(p,D)*-Verfahren beschriebenen Textsplittings auf das *LOF(MinPts)*-Verfahren zu untersuchen, wurden bis auf die „Subject"-Zeile und die „From"-Zeile, welche den Autor des Artikels enthält, alle Headerzeilen aus den Artikeln entfernt. Die Ergebnismenge wurde erneut vektorisiert. Die resultierenden Textvektoren mit 2.555 Dimensionen wurden sodann den gleichen *LOF*-Experimenten unterzogen und dies wurde zudem nach SVD-Reduktion auf eine dreidimensionale Menge und jeweils mit Einsatz der euklidischen und der Kosinusdistanz als Abstandsmaß wiederholt.

Abbildung 65 zeigt zusammengefasst die Ergebnisse der Optimierung durch das Textsplitting. Die Verfahrensanwendung im hochdimensionalen Bereich erzielt nachweislich sehr viel bessere Ergebnisse, als die Anwendung auf einer in der Dimensionszahl stark reduzierten Datenmenge. Der Einsatz des Kosinus-Distanzmaßes führt zu Ergebnissen mit einer höheren Genauigkeit, während das euklidische Abstandsmaß zu höherer Vollständigkeit führt. In Kombination optimaler Wertepaare und Abstandsmaße ergibt sich die eingezeichnete Optimumlinie, welche im Bereiche der Genauigkeit sogar noch besser als das Optimum des *DB(p,D)*-Verfahrens mit euklidischer Distanz ist, jedoch nicht in vergleichbare Bereiche der Vollständigkeit vordringen kann. Im Vergleich zur nicht durch Textsplitting bearbeiteten Datenmenge führt das *LOF*-Verfahren mit Textsplitting zu deutlichen Ergebnisverbesserungen.

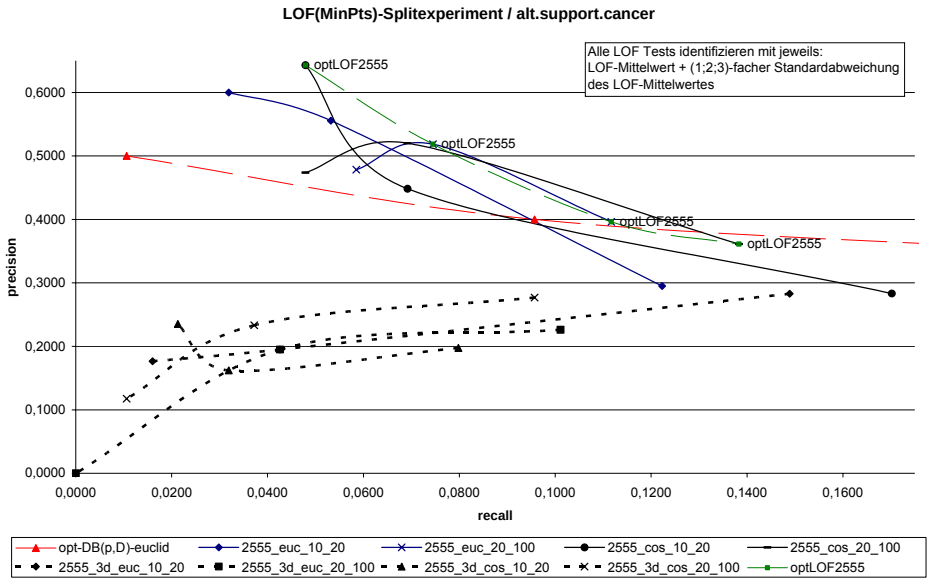


Abbildung 65 - *LOF*-Experiment mit Textsplitting

Abbildungsbeschreibung: Die Anwendung des Testsplittings zeigt verbesserte Ergebnisse vor allem im Bereich der Genauigkeit. Zum Vergleich wurde die Optimierungsgrenze des *DB(p,D)*-Verfahrens eingezeichnet.

6.3.4. ESOM Experiment

Für die Analyse der Testdatenmenge von alt.support.cancer (vgl. Kapitel 6.2.2) durch die DataBionic ESOM Tools [87] wurde zunächst in YALE mit dem WVtool die Datenmenge in Textvektoren umgewandelt. Dieses ExampleSet wurde sodann mit dem vom Outlier PlugIn bereitgestellten Export-Operator in das ESOM Format als Lerndatei (*.lrn) exportiert. Mit den ESOM Tools wurden auf dieser Lernmenge zwei Trainings in den Standard-Settings jeweils mit der Verwendung der euklidischen Distanz und der Kosinusdistanz durchgeführt. Im Ergebnis lagen zwei ESOM Karten vor, die nach potentiellen Outliern ausgewertet wurden. Dabei zeigte vor allem die Karte der euklidischen Distanz, dass die durch die *LOF*-Analyse der Ursprungsmenge erwartete spärliche Verteilung der Objekte im Datenraum auch hier gut zu sehen ist.

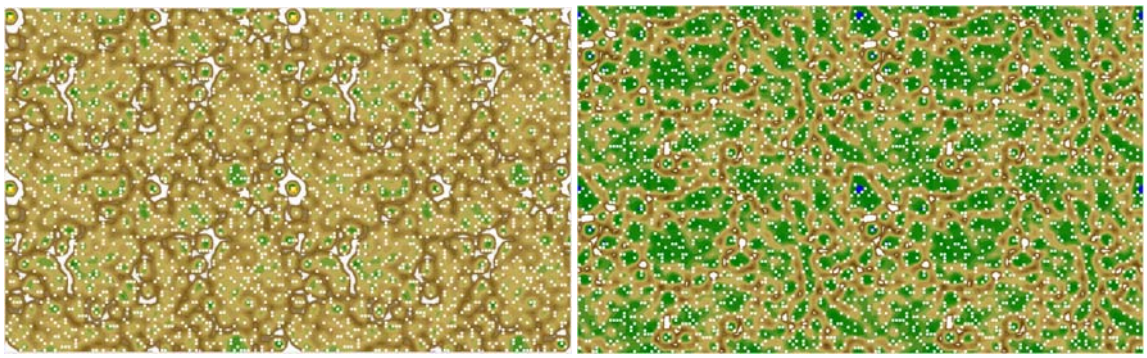


Abbildung 66 - ESOM Karten von alt.support.cancer (euklidisch (l), cosinus (r))

Abbildungsbeschreibung: Sowohl die euklidische als auch die Kosinusdistanz zeigen bei ihrem Einsatz durch das ESOM-Verfahren die spärliche Besetzung der Ursprungsdatenmenge sehr anschaulich. In der Abbildung wurde die vierfache Darstellung (tiled-Display) als grenzenlose Darstellung gewählt.

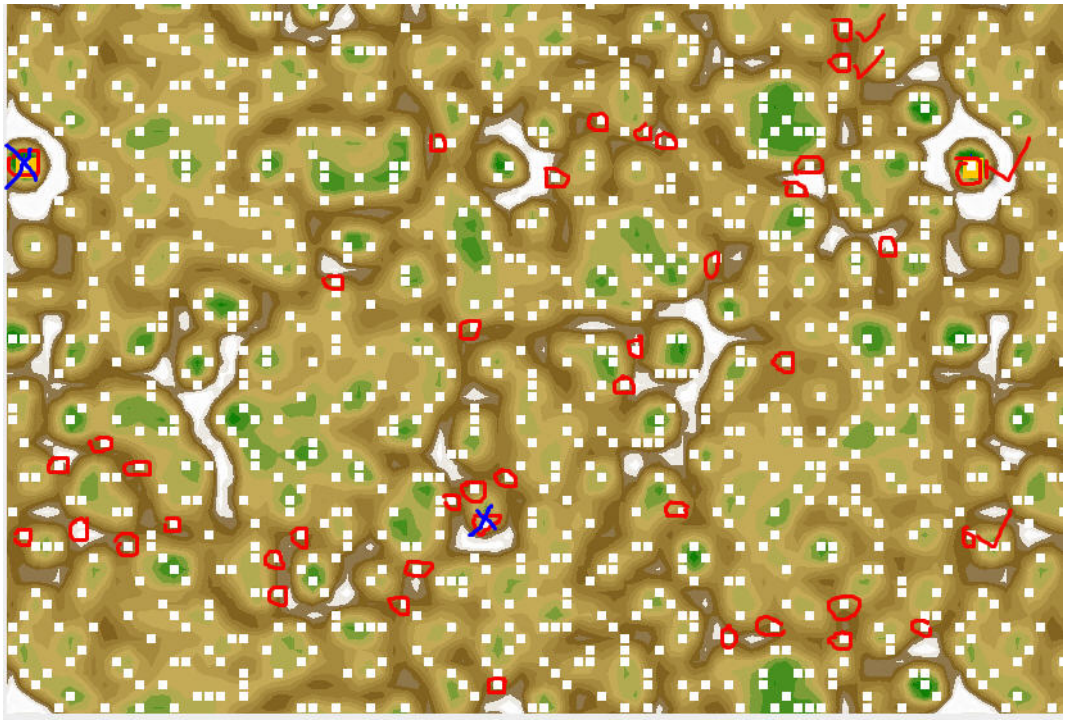


Abbildung 67 - ESOM Karte mit gekennzeichneten Outlier-Knoten bei euklidischen Distanz

Abbildungsbeschreibung: An Bergmassiven und –hängen liegende (oder von solchen eingeschlossene) Objekte wurden als Outlierkandidaten in der grafischen Auswertung gekennzeichnet. Dabei können Netzknoten mehrere Objekte enthalten. Die Streichung betrifft redundant gekennzeichnete Knoten, damit sichtbar wird, dass es sich um ein grenzenlos umlaufendes Display handelt.

Abbildung 66 zeigt links die Karte unter Anwendung der euklidischen Distanz als 4-fache Darstellung der Ursprungsmenge (Tiled Display [87]). Dabei ist bis auf wenige ausgeprägte Massive (weiß) zu erkennen, dass nur wenig aussagefähige Höhenunterschiede vorhanden sind, da deutliche große Täler mit hoher Ansammlungsdichte von Objekten fehlen. Die ESOM Karte in der gleichen Abbildung rechts unter Anwendung der Kosinusdistanz vermittelt auf den ersten Blick ein eindeutigeres Bild, aber auch hier sind sehr viele Gebirgsmassive vorhanden. Zum Vergleich zeigt Abbildung 43 eine Karte mit deutlicher Verteilung der Operatorentestmenge.

Für die Auswertung, d.h. das Finden von potentiellen Outliern, wurden nun die Objekt-Knoten identifiziert, welche sich am Rand von Bergmassiven befinden (vorzugsweise von diesen eingeschlossen bzw. an deren steileren Hängen gelegen). Dies erfordert eine manuelle Auswertung unter grafischer Interpretation. Die Abbildung 67 zeigt die 1-fache ESOM Karte für die euklidische Distanz mit farbigen Markierungen für potentielle Outlier-Knoten, die auch als Einkreisungen erkennbar sind. Abbildung 68 zeigt gleiches für die Kosinusdistanz.

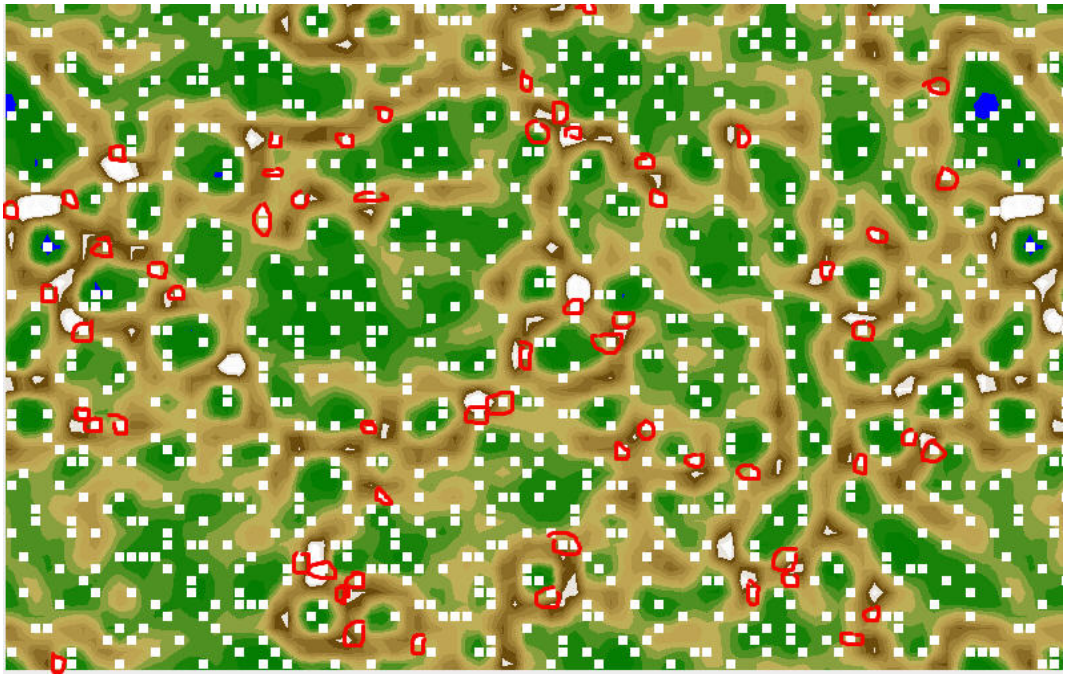


Abbildung 68 - ESOM Karte mit gekennzeichneten Outlier-Knoten bei Kosinusdistanz

Abbildungsbeschreibung: An Bergmassiven und –hängen liegende (oder von solchen eingeschlossene) Objekte wurden als Outlierkandidaten in der grafischen Auswertung gekennzeichnet. Dabei können Netzknoten mehrere Objekte enthalten. Die Kosinusdistanz zeigt zwar mehr Täler und damit stärkere Zusammenhänge von gewissen Objekten, allerdings gibt es auch weniger ausgeprägte Bergmassive zur Trennung unterschiedlicher Objekte.

Im Ergebnis zeigt die Auswertung, dass für die euklidische Karte 44 potentielle Outlier erkannt werden, für die Kosinusdistanz Karte werden 67 potentielle Outlier erkannt. Der Vergleich zwischen den Outlier-Kategorien der Testmenge und den erkannten Outliern ergibt für die euklidische Karte eine Übereinstimmung von 10 Outliern und für die Kosinus-Distanz-Karte eine Übereinstimmung von 18 Outliern. Zwar werden kategorisierte Outlier erkannt, die Fehlerquote gegenüber der Testkategorisierung liegt jedoch in beiden Karten bei mehr als 70%, d.h. die überwiegende Menge der erkannten potentiellen Outlier sind zumindest in der Testmenge nicht kategorisiert worden.

Es werden insgesamt 25 von 188 kategorisierten Outlierkandidaten durch eine kombinierte Anwendung der ESOM Analyse mittels euklidischem und Kosinusdistanz Training erfolgreich erkannt. Dabei ist Vorsicht anzuwenden, da die manuelle grafische Auswertung interpretationsfähig und fehlerbehaftet ist und ein hoher Anteil der erkannten potentiellen Outlier nicht den Kategorien zuzuordnen ist. Zudem wurde keine iterative Optimierung von Eingangswerten für das ESOM-Verfahren vorgenommen. Da jedoch die Analyse direkt auf der den Datenraum sehr spärlich besetzenden Ursprungsmenge durchgeführt wurde, sind die Ergebnisse besser, als allgemein zu erwarten wäre.

Eine interessante Grundfrage für zukünftige Betrachtungen ist z.B., warum die Anwendung der euklidischen Distanz und der Kosinusdistanz beim Training jeweils Outlier-Mengen identifiziert, die sich mit zwei „gleichen“ Outliern nur minimal überdecken, deren Ergebnisqualität aber ähnlich zu sein scheint. Weiterführend ist somit von Interesse, ob die Anwendung einer spezifischen Distanz Vorteile bringt, oder ob eine kombinierte Analyse mit verschiedenen Distanzen die Ergebnisse wesentlich verbessern würde. Für eine solche Analyse sind z.B. alle gezeigten Grafiken und auch das ESOM Datenset im Lieferumfang von [105] enthalten.

Von der Performance Analyse her ist eine ESOM Auswertung bei den gezeigten Testdatenmengen nicht für eine anwenderorientierte Identifizierung von Outliern geeignet. Die euklidische Berechnung brauchte auf einem Intel P5 3.0 Ghz Referenzsystem mit 1GB RAM eine Trainingszeit von mehr als 5 Stunden und bei Berechnung mit der Kosinusdistanz ca. 18 Stunden. Die grafische Auswertung mit einer Rückreferenzierung der erkannten Outlier-Knoten und ihrer Objekte ist ein zusätzlicher, nicht zu unterschätzender Zeitaufwand. Allerdings rechtfertigen die Erkenntnisse aus einer solchen Analyse die Betrachtung im Rahmen dieser Arbeit.

6.3.5. Anwendung von Autorenwissen

In den Abschnitten 4.8 und 5.2.9 wurde bereits ausgeführt, wie Autorenwissen prinzipiell zur Verbesserung der Ergebnisse von Outliererkennungsverfahren eingesetzt werden kann. Dabei sind die Einsatzmöglichkeiten – abhängig von der konkreten Zielrichtung – sehr vielfältig und die hier experimentell vorgestellten und nachvollzogenen Ansätze haben daher lediglich einen exemplarischen Charakter.

Zuerst ist ein Vergleich der Anwendung vorhandenen Autorenwissens im Vergleich mit der Anwendung von Outlier-Verfahren von Interesse. Dazu wird entsprechendes Autorenwissen benötigt. Im Test-SetUp wird dieses aus der vorkategorisierten Datenmenge erstellt. Hierzu wurde ein YALE-Experiment durchgeführt, welches aus der vorhandenen Datenmenge eine Autorendatei abgeleitete, die einen durchschnittlichen Erwartungswert für jeden Autor enthält, mit dem dieser Outlier verfasst. Dieser Erwartungswert ergibt sich wie bereits a.a.O. ausgeführt aus dem Verhältnis der vom Autor verfassten kategorisierten Outlier zu der Gesamtzahl der von diesem Autor verfassten Artikel in der Testmenge. Im Ergebnis wurden Erwartungswerte für alle der 143 Autoren, welche die 847 Artikel verfassten, ermittelt.

In einem ersten Experiment wurde das gewonnene Autorenwissen direkt auf die Testdatenmenge angewandt, wobei die Autorendatei jeweils zu 100%, 50% und 20% angewandt wurde, um zu simulieren, dass nicht immer das vollständige Wissen über alle Autoren vorhanden ist, wobei diese Simulation nicht die Möglichkeit beinhaltet, dass inkorrektes Wissen über einzelne Autoren vorhanden ist.

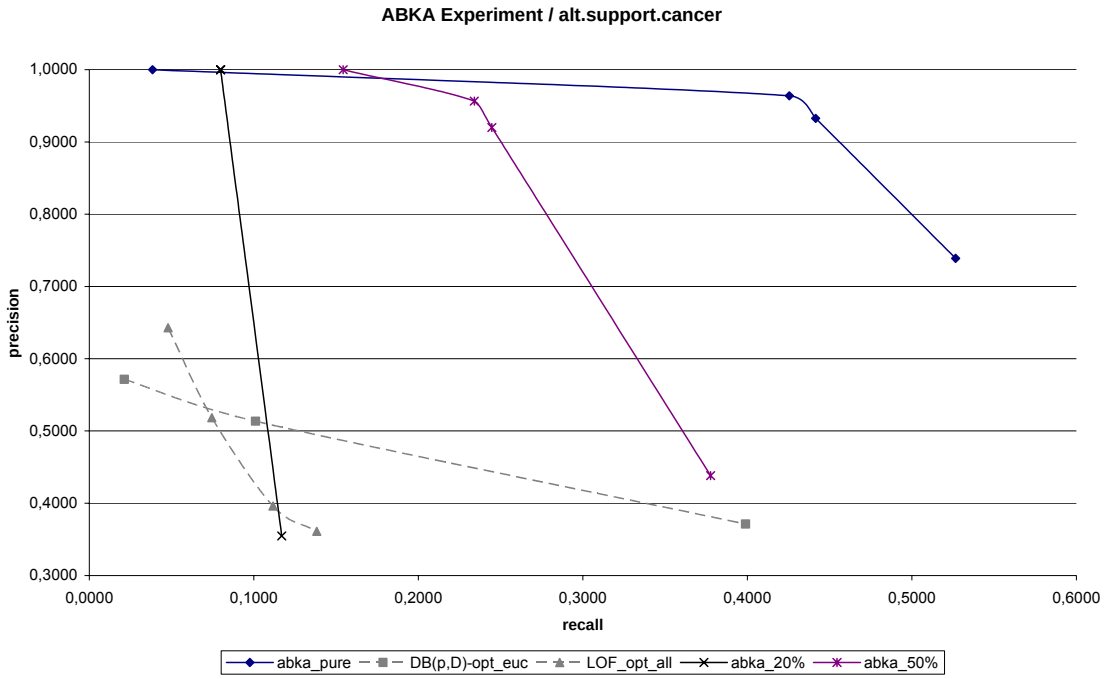
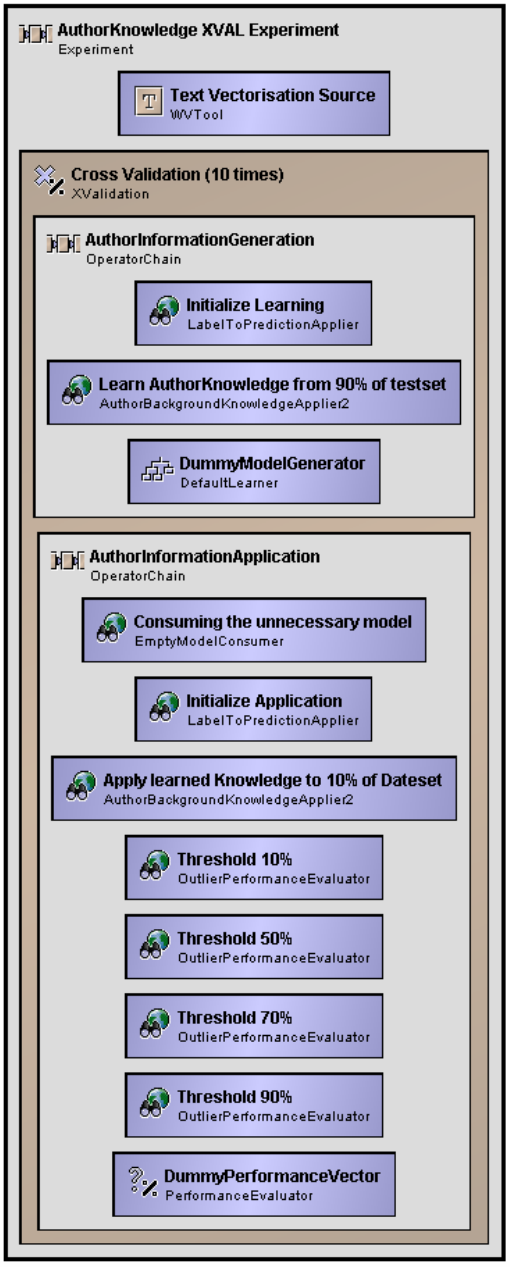


Abbildung 69 - Anwendung von Autorenwissen auf die Testdatenmenge

Abbildungsbeschreibung: In der Darstellung wird deutlich, dass direkt aus der Vorkategorisierung gewonnenes Hintergrundwissen diese auch direkt widerspiegelt.



TH	TP	TN	FP	FN	Precision	Recall	FallOut	F_Measure	OI
0,1	14	28	37	6	0,2745	0,7000	0,5692	0,3944	51
0,5	5	64	1	15	0,8333	0,2500	0,0154	0,3846	6
0,7	5	65	0	15	1,0000	0,2500	0,0000	0,4000	5
0,9	4	65	0	16	1,0000	0,2000	0,0000	0,3333	4
0,1	18	15	46	5	0,2813	0,7826	0,7541	0,4138	64
0,5	5	59	2	18	0,7143	0,2174	0,0328	0,3333	7
0,7	5	59	2	18	0,7143	0,2174	0,0328	0,3333	7
0,9	2	59	2	21	0,5000	0,0870	0,0328	0,1481	4
0,1	8	29	40	8	0,1667	0,5000	0,5797	0,2500	48
0,5	2	66	3	14	0,4000	0,1250	0,0435	0,1905	5
0,7	2	68	1	14	0,6667	0,1250	0,0145	0,2105	3
0,9	2	68	1	14	0,6667	0,1250	0,0145	0,2105	3
0,1	16	30	37	2	0,3019	0,8889	0,5522	0,4507	53
0,5	5	66	1	13	0,8333	0,2778	0,0149	0,4167	6
0,7	5	66	1	13	0,8333	0,2778	0,0149	0,4167	6
0,9	2	66	1	16	0,6667	0,1111	0,0149	0,1905	3
0,1	11	32	40	2	0,2157	0,8462	0,5556	0,3438	51
0,5	3	72	0	10	1,0000	0,2308	0,0000	0,3750	3
0,7	3	72	0	10	1,0000	0,2308	0,0000	0,3750	3
0,9	2	72	0	11	1,0000	0,1538	0,0000	0,2667	2
0,1	22	22	33	7	0,4000	0,7586	0,6000	0,5238	55
0,5	9	55	0	20	1,0000	0,3103	0,0000	0,4737	9
0,7	9	55	0	20	1,0000	0,3103	0,0000	0,4737	9
0,9	5	55	0	24	1,0000	0,1724	0,0000	0,2941	5
0,1	17	27	37	4	0,3148	0,8095	0,5781	0,4533	54
0,5	5	61	3	16	0,6250	0,2381	0,0469	0,3448	8
0,7	5	63	1	16	0,8333	0,2381	0,0156	0,3704	6
0,9	1	63	1	20	0,5000	0,0476	0,0156	0,0870	2
0,1	10	26	45	4	0,1818	0,7143	0,6338	0,2899	55
0,5	4	71	0	10	1,0000	0,2857	0,0000	0,4444	4
0,7	4	71	0	10	1,0000	0,2857	0,0000	0,4444	4
0,9	2	71	0	12	1,0000	0,1429	0,0000	0,2500	2
0,1	14	27	38	5	0,2692	0,7368	0,5846	0,3944	52
0,5	5	64	1	14	0,8333	0,2632	0,0154	0,4000	6
0,7	5	65	0	14	1,0000	0,2632	0,0000	0,4167	5
0,9	2	65	0	17	1,0000	0,1053	0,0000	0,1905	2
0,1	11	25	45	4	0,1964	0,7333	0,6429	0,3099	56
0,5	4	66	4	11	0,5000	0,2667	0,0571	0,3478	8
0,7	4	66	4	11	0,5000	0,2667	0,0571	0,3478	8
0,9	3	66	4	12	0,4286	0,2000	0,0571	0,2727	7
0,1	14	26	40	5	0,2602	0,7470	0,6050	0,3824	54
0,5	5	64	2	14	0,7739	0,2465	0,0226	0,3711	6
0,7	5	65	1	14	0,8548	0,2465	0,0135	0,3789	6
0,9	3	65	1	16	0,7762	0,1345	0,0135	0,2243	3

Tabelle 28 - Ergebnisse des Autorenwissen-Cross-Validation Experiments

Tabellenbeschreibung: In der Abbildung ist das YALE Experiment zur Cross-Validation grafisch dargestellt. In der nebenstehenden Tabelle sind die Ergebnisse der 10 Experimentreihen jeweils mit einem Threshold (TH) für den Outlierfaktor im Intervall [0;1] dargestellt, ab der ein Objekt als identifizierter Outlier gezählt wird. Die letzte farblich markierte Reihe beschreibt den Durchschnitt aller Experimente.

Abbildung 69 zeigt zusammenfassend die Ergebnisse in der Analyse und im Vergleich mit den optimalen Ergebnissen der Outlier-Verfahren (*LOF*-Optimum und *DB(p,D)*-Optimum). Es ist ablesbar, dass die reinen Outlierverfahren wesentlich schlechtere Ergebnisse liefern, wobei dies durch die Simulation nur teilweise vorhandenen Wissens um Autoren relativiert wird. Da das Autorenwissen nahezu vollständig die Vorkategorisierung repräsentiert und daher von einem hohen Qualitätsgrad der Ergebnisse einer Anwendung auf eben dieselbe Testmenge auszugehen ist, erscheinen die Outlierverfahren sogar in Bezug auf die Genauigkeit der Ergebnisidentifikation als sehr erfolgreich und zeigen lediglich in Bezug auf die Vollständigkeit Schwächen.

Cross-Validation Experiment für die Anwendung von Autorenwissen

Zur besseren Beurteilung der direkten Anwendung von Autorenwissen wurde in YALE ein Cross-Validation Experiment umgesetzt. Dabei wurde die Testdatenmente in jeweils 10 gleiche Teile aufgespaltet und aus einer Menge im Umfang von 9 von 10 Teilen jeweils Autorenwissen gewonnen und auf den 10-ten Teil angewendet, wobei dies rotierend für alle 10 Teile durchgeführt wurde. Dabei wurde die Testmengenteilung nach dem Zufallsprinzip vorgenommen, um Verfälschungen durch eine rein lineare Aufteilung zu vermeiden.

Die Ergebnisse des Experimentes und seine Aufstellung sind in Tabelle 28 im Detail aufgeführt. Dabei wurde für jede Experimentreihe auf einer 10%-tigen Teilmenge von 85 Elementen die Genauigkeit und Vollständigkeit der Erkennung gegenüber der Vorkategorisierung aufgelistet. Dabei ist zu beachten, dass die Vollständigkeitswerte natürlich nur für diese Teilmenge und nicht für die Gesamtmenge gelten.

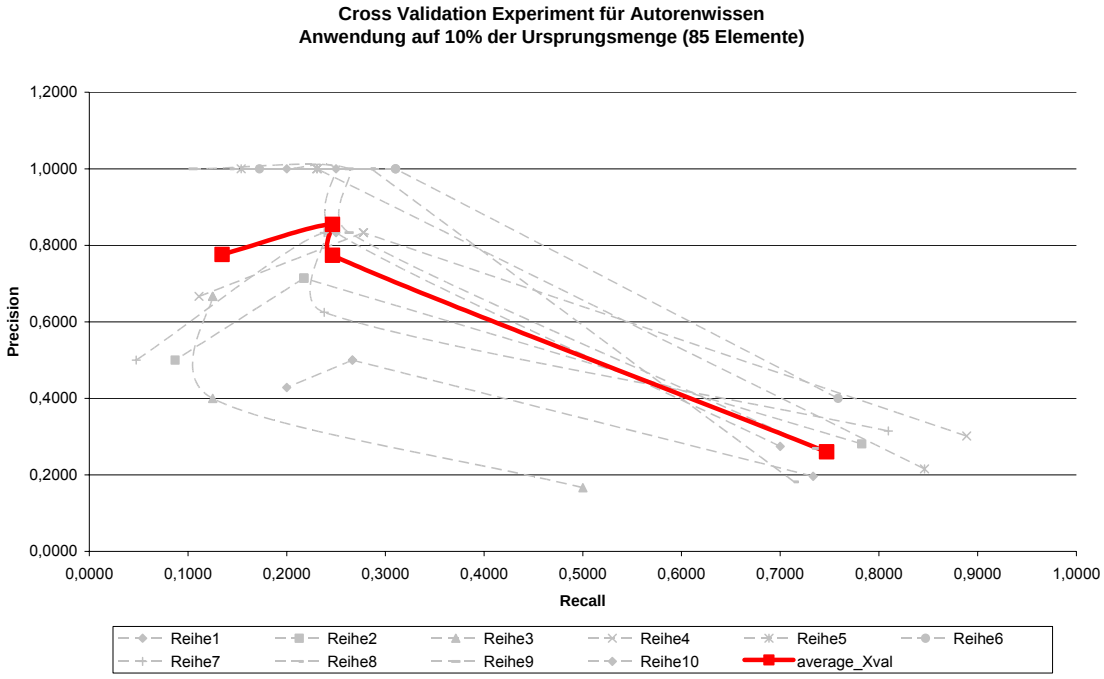


Abbildung 70 - Grafische Auswertung des Cross-Validation Experiments für Autorenwissen

> *Abbildungsbeschreibung:* Die Auswertung zeigt generell eine hohe Genauigkeit, wobei die Experimentreihen durchaus in den Trends schwanken. Auf einen Vergleich mit den Outliererkennungsverfahren wurde hier wegen der unterschiedlichen Basis für den Recall verzichtet.

Die in Abbildung 70 dargestellte Auswertung der Experimentreihen zeigt, dass auch bei Teilmengen als Lernbasis aus der Kategorisierung diese Wissensbasis ein robustes Werkzeug zur Erkennung von Outliern sein kann. Hierbei wird deutlich, dass die Anwendung eines Verfahrens, welches auf dem Mechanismus oder einer direkten Repräsentation von Mechanismen einer Anwendungsdomaine beruht, wie z.B. einer Vorkategorisierung, Outlier robust und zuverlässig erkennen kann. Allerdings ist zu prüfen, inwieweit neues Wissen, nach dem ein Anwender auch sucht, welches er jedoch nicht vorhersagen kann, von solchen Verfahrensansätzen nicht erkannt wird und damit verborgen bleibt.

Auf weitere Experimentreihen zur Verbindung von den herkömmlichen vorgestellten Verfahren und dem Verfahren mit Autorenwissen wurde verzichtet, weil ohne gründliche Analyse nur erwartet werden kann, dass sich die Ergebnisse der Erkennung vorkategorisierter Objekte ggf. noch steigern lässt. Eine Differenzierung des erkannten neuen Wissens könnte jedoch im Rahmen dieser Arbeit nicht umfänglich vorgenommen werden.

6.3.6. Vergleich der Erkennung von Kategorien

Um abzuklären, welche der Verfahren welche konkrete Kategorie der Outlier nach der Einteilung von Kapitel 4.4 besonders gut bzw. schlecht erkennen, wurde die Experimentreihe für alle Verfahren mit den optimierten Parameterkombinationen im Hinblick darauf ausgewertet. Die nachstehende Tabelle zeigt die entsprechenden Ergebnisse dieser Auswertung.

		ngSPO		gTO_WP		gTO_CP		gDO_SP		gDO_TD		FP
Verfahren	Wertepaare	Anzahl	Recall	Anzahl	Recall	Anzahl	Recall	Anzahl	Recall	Anzahl	Recall	
Vorkate-gorisie-rung	(entfällt)	19		15		25		28		101		0
ABKA	100%, t=0,3	17	89,47%	7	46,67%	20	80,00%	12	42,86%	43	42,57%	35
	100%, t=0,5	16	84,21%	5	33,33%	19	76,00%	11	39,29%	32	31,68%	6
	100%, t=0,7	16	84,21%	5	33,33%	17	68,00%	11	39,29%	31	30,69%	3
	100%, t=0,9	13	68,42%	4	26,67%	10	40,00%	11	39,29%	20	19,80%	0
LOF	2555, cos, 10_20	1	5,26%	2	13,33%	1	4,00%	2	7,14%	3	**2,97%**	5
	2555, euc, 20_100	1	5,26%	3	20,00%	1	4,00%	1	3,57%	8	7,92%	13
	2555, euc, 20_100	1	5,26%	3	20,00%	1	4,00%	5	17,86%	11	10,89%	32
	2555, cos, 20_100	3	15,79%	4	***26,67%***	2	8,00%	5	17,86%	12	11,88%	46
DB(p,D)	2555,euc,mean, p=0,95, D=1,3919	2	10,53%	0	**0,00%**	1	4,00%	1	3,57%	0	**0,00%**	3
	2555,euc,mean+var, p=0,875, D=1,394	3	15,79%	3	20,00%	2	8,00%	3	10,71%	8	7,92%	18
	2555,cos,mean, p=0,8, D=0,96977	5	26,32%	5	33,33%	15	60,00%	11	39,29%	35	34,65%	109
	2555,cos,mean-std, p=0,95, D=0,91737	14	73,68%	9	60,00%	21	***84,00%***	18	64,29%	69	68,32%	240
D(k,n)	2949, k=20, n=80	8	42,11%	0	**0,00%**	3	12,00%	7	25,00%	7	6,93%	55
	2949, k=20, n=188	10	52,63%	1	6,67%	11	44,00%	15	***53,57%***	20	19,80%	131

Tabelle 29 - Erkennung von Kategorien durch Outlier-Detection Verfahren

Tabellenbeschreibung: Für die unterschiedlichen Verfahren wird in der Tabelle aufgelistet, welche Performance diese bei der Erkennung von bestimmten Outlier-Kategorien gemäß den Definitionen der Vorkategorisierung zeigen. Dabei sind beste und schlechteste Werte jeweils hervorgehoben.

Dabei ist in der Tabelle grau hinterlegt, dass das *DB(p,D)*-Verfahren für eine akzeptable Vollständigkeit durchweg die besten Ergebnisse der Standardverfahren erzielt. Zudem erkennt dieses Verfahren bei großer Vollständigkeit die Cross-Posting-Outlier am besten. Bei einer hohen Genauigkeit erkennt dieses Verfahren am besten die SPAM-Outlier. Die Falschpostings und Diskussionsabweichler werden am schlechtesten erkannt. Das *D(k,n)*-Verfahren erzielt bei großer Vollständigkeit die besten Ergebnisse für Einzelposting-Outlier. Bei hoher Genauigkeit werden wiederum SPAM-Outlier am besten erkannt und Falschpostings am schlechtesten. Das *LOF(MinPts)*-Verfahren erkennt bei großer Vollständigkeit Falschpostings am besten, bei hoher Genauigkeit werden solche Falschpostings auch am besten erkannt, wobei Diskussionsabweichler am schlechtesten abschneiden. Die Ergebnisse für das Autorenwissen-Verfahren wurden nur zum Vergleich dargestellt, da diese Informationen direkt aus der Vorkategorisierung gewonnen wurden.

In der Zusammenfassung wird deutlich, dass bei einer Kalibrierung der Verfahren auf ein hohes Maß an Genauigkeit SPAM Outlier fast durchgängig am besten erkannt werden. Um jedoch zu beurteilen, warum die Verfahren spezifische Kategorien unter gewissen Parametereinstellungen besonders gut bzw. besonders schlecht erkennen, ist eine detaillierte Analyse der Mechanismen der einzelnen Kategorien im Zusammenhang mit der textuellen Ausprägung in den USENET News notwendig. In Verweis auf Kapitel 7 könnte dies der Gegenstand weitergehender Forschung sein.

6.4. Zusammenfassung der experimentellen Ergebnisse

Im Vergleich der Standardverfahren ergibt sich für das *DB(p,D)*-Verfahren die beste, auf die jeweiligen Anforderungen an Genauigkeit und Vollständigkeit ausgerichtete Optimierbarkeit, welche jedoch eine Vergleichsbasis durch eine Vorkategorisierung voraussetzt. Das *D(k,n)*-Verfahren erzielt gute Ergebnisse im Hinblick auf Genauigkeit und Vollständigkeit im Vergleich aller Verfahren, wobei wenige Variationen der Parameter *k* und *n* als Vorhersageabschätzung der Anzahl zu erwartender Outlier notwendig sind. Das *LOF(MinPts)*-Verfahren benötigt fast keine Optimierung, da nur *MinPts*-Intervalle anzugeben sind. Es zeigt im Verfahrensvergleich eine gute Genauigkeit, jedoch nur eine geringe Vollständigkeit. Der Grund hierfür ist vor allem in der spärlichen Besetzung des hochdimensionalen Raumes zu vermuten, durch die eine Dichtebetrachtung – vor allem eine lokale – gegenüber einem globalen Entfernungsmaß deutlich an Aussagekraft verliert. Insofern stützt dies die Aussage in der Literatur [68]. Zusammenfassend ist zu bemerken, dass alle ausgewählten Standardverfahren die vorkategorisierten Outlier nur in beschränktem Maße genau und vollständig erfassen, sofern dies auf Basis der Vorkategorisierung für diese spezifische Anwendungsdomaine beurteilt wird, was aber keiner formalen Einschätzung der Verfahren gleichkommt. Dies war wegen der schwierigen Begleitumstände und der hohen Erkennungshürden, welche durch die USENET News als Anwendungsbereich gesetzt werden, auch erwartbar.

Bei der Vorverarbeitung hat sich gezeigt, dass eine Reduktion der Dimensionen durch Singular Value Decomposition bis auf wenige Ausnahmen generell zu einer Verschlechterung der Erkennung im Vergleich zur Vorkategorisierung führt. Ein Optimierungsversuch durch Einsatz der Kosinusdistanz als Abstandsmaß führte auf den hochdimensionalen Mengen nicht zu signifikanten Unterschieden, zeigte aber leichte Verbesserungen. In gewissen Fällen erwies sich eine Kombination von Dimensionsreduktion und Kosinusdistanz als robust gegenüber einem Qualitätsverlust, die Betrachtung reicht jedoch nicht aus, daraus einen Grundsatz zu folgern. Das Textsplitting verbesserte die Ergebnisse vor allem beim *DB(p,D)*-Verfahren und beim *LOF*-Verfahren leicht in gewissen Konstellationen, verschlechterte diese jedoch beim *D(k,n)*-Verfahren. Somit ist die Reduktion der Basistexte durch das Abspalten der Headerinformationen von den Newsartikeln ein zusätzlicher Optimierungspfad.

Im Bereich der angepassten Verarbeitungsverfahren wurde die Einbeziehung von Autorenwissen als Alternative untersucht. Der reine Einsatz von Autorenwissen, welches aus der Vorkategorisierung gewonnen und welches mit einem hohen Grad an Vollständigkeit (d.h. ohne Lücken) eingesetzt wurde, zeigte hervorragende Ergebnisse. Dies war allerdings zu erwarten, da ein direkt aus der Vorkategorisierung gewonnenes Material diese natürlich auch direkt widerspiegeln muss. Sobald das Autorenwissen unvollständig eingesetzt wird, d.h. mit Lücken, sodass von der ursprünglichen Informationsmenge nur noch 20% vorhanden sind, fallen die Ergebnisse der Anwendung im Vergleich zu den anderen betrachteten Verfahren in deren Qualitätsbereiche zurück, sobald eine größere Vollständigkeit erwartet wird. Allein im Bereich der Genauigkeit bei sehr geringer Vollständigkeit zeigt das Autorenwissen-Verfahren bessere Ergebnisse. Dies ist vor allem auf den Einsatz von hohen Schwellwerten und den Gewinnungsprozess dieses Wissens zurückzuführen. Im Fazit spiegelt also das Autorenwissen, welches aus einer Kategorisierung gewonnen wurde, dieses auch am besten wider. Aber der Vergleich von Ergebnissen mit der Vorkategorisierung ist nur sehr bedingt eine Basis für eine Verfahrensbewertung.

Die Kombination von Autorenwissen und Standardverfahren ist in so vielfältiger Art und Weise möglich, dass eine empirische Betrachtung zum einen nur einen winzigen – und damit keineswegs repräsentativen – Ausschnitt zeigen könnte, zum anderen würde eine konsequente Verfolgung dieses Ansatzes den Umfang dieser Arbeit überschreiten. Sinn der Betrachtung in dieser Arbeit war vor allem der Vergleich mit Standardverfahren, nicht die Kombination. Daher wurde vor allem für eine weitergehende Betrachtung die Voraussetzung durch die Implementierung eines flexiblen Operators geschaffen und im Ausblick wird auf diese Möglichkeiten verwiesen.

Die Erkennungsquoten einzelner Kategorien zeigen je nach Ausrichtung des Verfahrens durch die entsprechende Parameterwahl bzw. auch pro Verfahren starke Abweichungen und daher kein durchgängiges Bild, zumal die Auswertung nur auf dem „pareto“ Anteil der optimalen Parameterkombinationen für alle Verfahren durchgeführt wurde.

7. Abschlussbetrachtung und Ausblick

„Dum ferrum candet, tundendum est.“

Outliererkennung in USENET Newsgruppen

Grundlegend ist zu erkennen, dass sich potentielle Outlierkandidaten in Newsgruppen auf einen den Teilnehmern bekannten Kontext beziehen. Dieser umfasst

- die Gesamtmenge oder eine Teilmenge der an der Diskussion teilnehmenden Autoren der Newsartikel,
- die übergeordneten, historisch noch bekannten und spezifisch in einem Thread behandelten Themen der Diskussion,
- das Wissen um Diskussionsstränge innerhalb und außerhalb des betrachteten Snapshots,
- teilweise komplexe persönliche Beziehungen zwischen den Autoren (inkl. Sympathien, Antipathien, Gruppenbildungen, etc.)
- gemeinsame oder verschiedenartige zugrunde liegende Interessen der Diskussionsteilnehmer
- andere Informationen von außerhalb der Newsgruppe oder des Snapshots oder basierend auf globalem oder aktuellem Allgemeinwissen, welches nicht explizit referenziert oder erwähnt wird

Rein textuelle Untersuchungen können diesen Kontext nur insoweit erfassen, als er sich in textuellen Unterschieden mit ausreichender Signifikanz in den Newsartikeln selbst manifestiert. Ist dies nicht der Fall, bleibt dem Erkennungsverfahren potentieller Outlier der Kontext zum größten Teil verborgen. Da bei der Vorkategorisierung von Objekten als Outlierkandidaten der Kontext aber zu einem gewissen Teil durch den Beobachter in der Entscheidungsfindung zur Selektierung einer bestimmten Kategorie für ein Objekt nachgebildet wird, weicht natürlich auch die Repräsentation der Vorauswahl stark von der auf die reinen Texte bezogenen Ergebnisse ab.

Daher handelt es sich bei der Anwendung von statistischen Verfahren auf die Texte in Newsgruppen um eine Reinanwendung. Möglicherweise ist die Ausblendung des Kontexts auch von Vorteil, um Wissen und Zusammenhänge aufzudecken, welche dem Nutzer ansonsten durch die Verfälschung aufgrund des Kontextwissens verborgen blieben. Verfahren zur Beurteilung der Qualität von Outliern stecken noch in der frühen theoretischen Betrachtung. Momentan beschränken sich die Autoren von Outlierverfahren auf die Prüfung der mathematischen Plausibilität des erkannten Outlier-Status von Objekten in verschiedenen Wissensdomainen. Zudem benötigen viele Verfahren eine Reihe von subjektiven Eingabeparametern, die Auswahl passender statistischer Maße oder auch eine subjektive Auswertung. Somit sind die Ergebnisse der Verfahrensanwendungen auf verschiedene Wissensdomainen keineswegs eindeutig.

Dies liegt auch in der Natur der Sache. Es ist für verschiedene Beobachter des gleichen Sachverhalts ggf. einfacher, sich über eine große Schnittmenge an Gemeinsamkeiten zu einigen, als über eine kleine Zahl von Ausnahmen, da schon allein die Erfolgsquote (im Sinne der Anzahl der positiv gleichsam eingeordneten Objekte im Gegensatz zu wenigen differenziert beurteilten Ausnahmen) im ersteren Fall höher ist. Der Erfolg von Outlier-Verfahren liegt somit derzeit vor allem in der unstrittigen Erkenntnis über extreme Ausnahmen, über welche Einigkeit erzielt werden kann. Hier spielt auch eine harte Kategorisierung eine wichtige Rolle. Als Beispiel sei die Erkennung von Kreditkartenbetrug angeführt. Für andere Anwendungsdomainen können Outlierverfahren eine Vorselektion von Wissen über Abweichungen jenseits definierter Schwellwerte geben. Meist ist dem Nutzer aber die Verantwortung über die endgültige Interpretation dadurch nicht abgenommen. Und diese ist selbst in überschaubaren Datenmengen nicht immer eindeutig. Für das in dieser Arbeit betrachtete Sachgebiet der USENET Newsgruppen ist z.B. eine Vorbewertung von Artikeln als Outlier möglich, sofern das Verfahren in der Lage ist, dem Anwender einen Wissenszuwachs in Form von erfolgreicher Kategorisierung von Artikeln (z.B. in einer Art Rating – ähnlich wie bei der Bewertung von SPAM) zu bieten. Folgende Fragen sollten vor allem in Bezug auf die Outliersuche in Texten, z.B. in USENET News, Thema der weitergehenden Forschung sein

Frage: Wie gut ist eine statistische Reintextanalyse ohne Kontextinformationen?

Frage: In welchem Umfang ist die Ergänzung von Kontextinformation sinnvoll und möglich?

Frage: Verbessern sich die Ergebnisse durch Hinzunahmen begrenzten Kontexts?

Frage: Wie Erfolg versprechend ist das Lernen von Zusammenhängen aufgrund gesammelter Informationen?

Allgemeiner Ausblick für anwendungsbezogene Outliererkennung

Die Betrachtung des Forschungsfeldes der Outliererkennung im theoretischen und auch im praktischen Bereich hat gezeigt, das nach wie vor eine Reihe von Aspekten ungeklärt ist. Auf der einen Seite gestaltet sich die Bewertung der Ergebnisse von Outlierverfahren als sehr schwierig, vor allem, wenn keine Vergleichsbasis durch eine Erwartungshaltung vorhanden ist. Eine formale Einschätzung von Verfahren und ihrer Ergebnisse ist u.a. aus diesem Grund bisher noch nicht erfolgreich vorgenommen worden. Auf der anderen Seite führt die Aufstellung einer Erwartungsbasis, wie z.B. in dieser Arbeit durch die Vorkategorisierung, zu dem Problem der Trennung zwischen erwartet erkanntem Wissen und neuem, unerwartet gelerntem Wissen. Ein strikter Vergleich mit der Vorkategorisierung blendet den neu gelernten Teil wohlmöglich vollständig als „Fehler“ aus. Dies entspricht sicher nicht der Realität, noch der Absicht des Anwenders auf der Suche nach sinnvollen und prägnanten Outliern. Es muss also sichergestellt werden, dass die Erwartung – und damit die daraus resultierende Eingrenzung – die Wissensentdeckung nicht behindert.

Dieser Zielkonflikt spiegelt sich bereits in der Definition von Outliern nach Hawkins wider. Diese – und damit alle auf ihr basierenden Verfahren – betrifft nur die Wirkung, d.h. den Verdacht aufgrund der Beobachtung des Verhaltens von Objekten. Sie betrifft nicht die Ursachen, d.h. die abweichenden Mechanismen selbst, aus denen die Verhaltensabweichung resultiert. Diese Mechanismen bleiben der Betrachtung vor allem aus zwei Gründen verborgen. Sie sind zum einen meist unbekannt oder zumindest nicht formal und vollständig beschrieben. Oft ist schon die Beschreibung des Normalverhaltens, also der Summe der Grundmechanismen ohne Abweichungen, entweder nicht vorhanden oder aber nicht formal und vollständig. Zum anderen sind die Mechanismen pro Anwendungsdomaine zuweilen sehr spezifisch. Eine auf die Ursachen ausgerichtete Betrachtung erschwert also die allgemeine Verwendung der entsprechenden Verfahren.

Sichtbar wird dies in den zwei aus der Verfahrenslistung sichtbaren Stoßrichtungen der Bemühungen der Outlierforschung. Die eine Richtung bemüht sich, die Wirkungserkennung zu optimieren und über Anwendungsdomainen hinweg zu verallgemeinern. Im Ergebnis werden Verfahren vorgestellt, welche möglichst breit anzuwenden sind, einen unifizierenden Charakter [3] haben und welche universell Abweichungen in verschiedenen Betrachtungsblickwinkeln ([4], [68]) erkennen können. Eine andere Richtung bemüht sich, die Ursache-Wirkungs-Kette möglichst gut nachzubilden, d.h. den Vorgang der Verhaltensabweichung nachzumodellieren, indem Normalverhalten als Basis zum Erkennen von abweichendem Verhalten nachgestellt wird ([5], [10], [14], [18], [19]). Auch hier wird darauf abgestellt, die Verfahren möglichst universell auszulegen.

Eine dritte Variante betrifft die Möglichkeit, die Outliererkennung auf eine spezielle Anwendungsdomaine auszurichten, indem die Verhaltensabweichung im Bezug zum Normalverhalten von diesem getrennt analysiert wird. Hier kommen z.B. paritätische Verfahren zum Einsatz ([28], [46], [11]). Interessant ist dieser generelle Ansatz vor allem deshalb, weil diese Grundidee dazu benutzt werden könnte, allgemeingültige Erkennungsverfahren mit solchen ursachenbezogenen Verfahren, welche versuchen, mittels einer Mechanismenanalyse Erwartungshaltungen als Vergleichsbasis bereitzustellen, zu verknüpfen, ohne dass es zu den vorher beschriebenen Nachteilen der Ausblendung der Wissenserkennung kommt. Wie genau dies geschehen kann, wird die weitere Entwicklung der Outlier-Forschung zeigen, welche mit Sicherheit noch am Anfang steht.

8. Literaturverzeichnis

„Scribendi recte sapere est et principium et fons“

[1] Van Rijsbergen, C. J., *Information Retrieval*, 2nd edition, Dept. of Computer Science, University of Glasgow, 1979

[2] Gerard Salton, *Automated Text Processing*, Addison Wesley, 1989

[3] Knorr, Edwin M., Ng, Raymond T., *Algorithms for Mining Distance-Based Outliers in Large Datasets*, Proc. 24th VLDB, 1998

[4] Breuning, S., Kriegel, H.P., Ng, Raymond T., Sandner, J., *LOF: Indentifying Density-Based Local Outliers*, ACM SIGMOD Int. Conf. on Management of Data, Dallas, TX, 2000

[5] W. Jin, A. K. Tung, and J. Han, *Mining top-n local outliers in large databases*, In Proc. of KDD'2001, pages 293–298, 2001.

[6] Edwin M. Knorr, Raymond T. Ng , *A Unified Approach for Mining Outliers*, Proc. 7th CASCON, pages 236-248, 1997, Anmerkung: Dies ist eine erweiterte Version von [7].

[7] Edwin M. Knorr, Raymond T. Ng, *A unified notion of outliers: Properties and computation*, Proc. KDD, pages 219-222, 1997

[8] Edwin M. Knorr, Raymond T. Ng, Vladimir Tucakov, *Distance-based outliers: algorithms and applications*, The VLDB Journal — The International Journal on Very Large Data Bases, Volume 8 , Issue 3-4, pp: 237 – 253, 2000, ISSN:1066-8888

[9] Stephen D. Bay, Mark Schwabacher, *Mining distance-based outliers in near linear time with randomization and a simple pruning rule*, Conference on Knowledge Discovery in Data, Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp: 29 – 38, 2003, ISBN:1-58113-737-0

[10] Zengyou He, Xiaofei Xu, Shengchun Deng, *Discovering cluster-based local outliers*, Pattern Recognition Letters, Volume 24 , Issue 9-10, Pages: 1641 - 1650, June 2003, Elsevier, ISSN:0167-8655

[11] Z. He, S. Deng, X. Xu, *Outlier Detection integrating semantic knowledge*, In: Proc. Of the 3rd International Conference on Web-Age Information Management, Beijing, China, pp. 126-131, 2002

[12] George Kollios, Dimitrios Gunopulos, Nick Koudas, Stefan Berchtold, *Efficient Biased Sampling for Approximate Clustering and Outlier Detection in Large Data Sets*, IEEE Transactions on Knowledge and Data Engineering, Volume 15 , Issue 5, Pages: 1170 – 1187, September 2003, ISSN:1041-4347

[13] S. Ramaswamy, R. Rastogi, and K. Shim, *Efficient algorithms for mining outliers from large data sets*, In Proc. of SIGMOD'2000, pages 427–438, 2000.

[14] Simon Hawkins, Hongxing He, Graham Williams and Rohan Baxter, *Outlier Detection Using Replicator Neural Networks*, CSIRO Mathematical and Information Sciences

[15] Li Wei, Weining Qian, Aoying Zhou, Wen Jin, Jeffrey X. Yu, *HOT: Hypergraph-based outlier Test for Categorical Data*, citeseer.ist.psu.edu/644023.html

[16] Amitabh Chaudhary and Alexander S. Szalay and Andrew W. Moore, *Very Fast Outlier Detection in Large Multidimensional Data Sets*, citeseer.ist.psu.edu/695046.html

[17] Jörg Pawlitschko, *On the distribution of a test statistic for outlier detection in exponential samples*, Dept. of Statistics, University of Dortmund

[18] J. Tang, Z. Chen, A. Fu, D. Cheung, *A Robust Outlier Detection Scheme in Large Data Sets*, PAKDD, 2002. http://citeseer.ist.psu.edu/tang01robust.html

[19] Zhixiang Chen, Ada Wai-Chee Fu, Jian Tang, *On Complementarity of Cluster and Outlier Detection Schemes*, Dept. of Computer Science University of Texas-Pan American, and Dept. of Computer Science and Engineering, Chinese University of Hong Kong, citeseer.ist.psu.edu/610797.html

[20] Aoying Zhou, Weining Qian, Hailei Qian, Jin Wen, Shuigeng Zhou, Ye Fan, *A Hybrid Approach to Clustering in Very Large Databases*, Lecture Notes In Computer Science; Vol. 2035, Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Pages: 519 – 524, 2001, ISBN:3-540-41910-1

[21] Dimitris Glotsos, Jussi Tohka, Jori Soukka and Ulla Ruotsalainen, *A New Approach to Robust Clustering by Density Estimation in an Autocorrelation Derived Feature Space*, Proceedings of the 6th Nordic Signal Processing Symposium - NORSIG 2004, June 9 - 11, 2004, Espoo, Finland

[22] Edwin M. Knorr and Raymond T. Ng, *Extraction of spatial proximity patterns by concept generalization*, Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD), 347-350, 1996

[23] G.J. McLachlan and D. Peel, *Robust cluster analysis via mixtures of multivariate t-distributions*, In A. Amin, D. Dori, P. Pudil, and H. Freeman, editors, Advances in Pattern Recognition, number 1451 in Lecture Notes in Computer Science, pages 658--665. Springer, 1998

[24] M. T. Gallegos, *Robust Clustering under general normal assumptions*, Technical Report MIP-0103, Fakultät für Mathematik und Informatik, Universität Passau

[25] M. T. Gallegos, A *Robust Method for Clustering Analysis*, Technical Report MIP-0013, Fakultät für Mathematik und Informatik, Universität Passau, Oktober 2000

[26] Y. Huang, H. Xiong, S. Shekhar, and J. Pei., *Mining Confident Co-location Rules without a Support Threshold*, 18th ACM Symp. on Applied Computing, 2003

[27] S. Shekhar, P. Zhang, Y. Huang, R. R. Vatsavai, *Trends in Spatial Data Mining*, citeseer.ist.psu.edu/694073.html, Book Chapter in Data Mining: Next Generation Challenges and Future Directions, AAAI/MIT Press, 2003

[28] S. Shekhar, C.-T. Lu, and P. Zhang, *A Unified Approach to Detecting Spatial Outliers,* GeoInformatica 7:2, 139-166, Kluwer Academic Publishers, The Netherlands, 2003

[29] S. Shekhar, C.-T. Lu, and P. Zhang, *Detecting graph-based spatial outliers: Algorithms and applications (a summary of results)*, In Proc. of KDD'2001, 2001.

[30] Wei Wang, Jiong Yang, Richard Muntz, *An Approach to Active Spatial Data Mining Based on Statistical Information*, IEEE Transactions on Knowledge and Data Engineering, Volume 12 Issue 5, , September 2000

[31] D. Boley and M. Gini and K. Hastings and B. Mobasher and J. Moore, *A Client-Side Agent for Document Categorization and Exploration*, Journal of Internet Research, Vol. 8, No. 5, 1998.

[32] Ying Liu, Alan P. Sprague, Elliot Lefkowitz, *Network flow for outlier detection*, Proceedings of the 42nd annual ACM Southeast regional conference table of contents, Pages: 402 – 103, 2004, ISBN:1-58113-870-9

[33] V. Barnett, T. Lewis, *Outliers in Statistical Data*, John Wiley and Sons, New York, 1994

[34] K. Yamanishi, J. Takeuchi, G. Williams, *Online Unsupervised Outlier Detection using finite mixtures with discounting learning algorithms*, In: Proc. of KDD 2000, Boston, MA, USA, pp.: 320-325, 2000

[35] K. Yamanishi, J. Takeuchi, *Discovering Outlier Filtering rules from unlabled data combining a supervised learner with an unsupervised learner*, In: Proc. of KDD 2001, pp.: 389 – 394, 2001

[36] I. Ruts and P. Rousseeuw, *Computing depth contours of bivariate point clouds*, Journal of Computational Statistics and data Analysis, 23:153–168, 1996.

[37] T. Johnson, I. Kwok, R. Ng, *Fast Computation of 2-Dimensional Depth Contours*, In Proc. 4th KDD Conference, New York, AAAI Press, pp. 224 – 228, 1998

[38] F. Preparata, M. Shamos, *Computational Geometry: An Introduction*, Springer, 1998

[39] J.W. Tukey, *Mathematics and the picturing of data*, Proc. Int. Congress Math., Vancouver, Vol. 2, pp. 523 – 531, 1975

[40] J.W. Tukey, *Exploratory data analysis*, Addison Wesley, Reading, MA, USA, 1977

[41] C. G. Small, *A survey of multidimensional medians*, Int. Stat. Rev. 58, pp. 263 – 277, 1990

[42] A. Niinimaa, *Bivariate generalizations of the median*, Technical Report, University of Oulu, Finland, 1993

[43] D.L. Donoho, *Breakdown properties of multivariate location estimators*, Ph.D Qualifying Paper, Harvard University, 1982

[44] D.L. Donoho, M. Gasko, *Breakdown properties of location estimates based on halfspace depth and projected outlyingness*, Ann. Statist., pp. 1803 – 1827, 1992

[45] Hawkins, *Outliers in Statistical Data*, Chapman & Hall, 1980

[46] T. Cheng, Z. Li, A *Multiscale Approach to detect spatial-temporal Outliers*, Department of Land Surveying and Geo-Informatics, Hong Kong Polytechnic University, 2004

[47] D. Freedman, R. Pisani, R. Purves, *Statistics*, W.W. Norton, New York, 1978

[48] Hartung, Joachim, Elpelt, Bärbel, *Multivariate Statistik: Lehr- und Handbuch der angewandten Statistik*, Oldenbourg (München, Wien), 6. Auflage, 1999, ISBN 3-486-25287-9

[49] N. Draper, H. Smith, *Applied Regression Analysis*, John Wiley & Sons, 1966

[50] P. J. Rousseeuw, A. M. Leroy, *Robust Regression and Outlier Detection*, John Wiley & Sons, 1987

[51] Edwin M. Knorr, Raymond T. Ng, *UO(p,D)-Outlier: A unified notion of outliers*, Unpublished Manuskript, Dept. of Computer Science, University of British Columbia, 1997

[52] U. Fayyad, G. Piatetsky-Shapiro and P. Smyth, *Knowledge Discovery and Data Mining: Towards a Unifying Framework*, Proc. 2nd Int. Conf. on KDD, Portland, OR, pp. 82-88, 1996

[53] S. Berchthold, D. A. Keim, H.-P. Kriegel, *The X-tree: An Index Structure for High Dimensional Data*, 22nd Conf. on VLDB, Bombay, India, pp. 28-39, 1996

[54] R. Weber, H.-J. Schek, S. Blott, *A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces*, Proc. Conf. VLDB, New York, pp. 194-205, 1998

[55] E. M. Knorr, R. T. Ng, *Finding Intensional Knowledge of Distance-based Outliers*, Proc. 25th Int. Conf. on VLDB, Edinburgh, Scotland, pp. 211-222, 1999.

[56] M. F. Jiang, S. S. Tseng, C. M. Su, *Two-phase Clustering process for Outlier Detection*, Pattern Recognition Letters, Vol. 22 (6/7), pp. 691-700, 2001

[57] Z. He, X. Xu, S. Deng, J. Z. Huang, *Clustering Categorical Data Streams*, http://arxiv.org/abs/cs/0412058, To Appear in Journal of Computational Methods on Science and Engineering(JCMSE), 2003

[58] Z. He, X. Xu, S. Deng, *Sqeezer: an efficient algorithm for clusterung categorical data*, Journal of Computer Science and Technology, Vol. 17 (5), pp. 611-624, 2002

[59] B. Liu, W. Hsu, Y. Ma, *Integrating classification and association rule mining*, Proc. KDD, pp. 80-86, New York, 1998

[60] T. Zhang, R. Ramakrishnan, M. Livny, *BIRCH: an efficient data clustering method for very large databases*, In Proc. ACM-SIGMOD Int. Conf. Management of Data, Montreal, Canada, pp. 103-114, June, 1996

[61] N. Roussopoulos, S. Kelley, F. Vincent, *Nearest neighbour queries*, Proc. ACM SIGMOD, San Jose, California, pp. 71-79, 1988

[62] N. Beckmann, H-P. Kriegel, R. Schneider, B. Seeger, *The R*-Tree: an efficient and robust access method for points and rectangles*, In Proc. ACM SIGMOD, Atlantic City, NJ, pp. 322-331, May, 1990

[63] S. Ramaswamy, R. Rastogi, and K. Shim, *Efficient algorithms for mining outliers from large data sets*, Technical Report, Bell Laboratories, Murray Hill, 1998

[64] N. Cressie, *Statistics for Spatial Data*, Revised Edition, New York, Wiley, 1993

[65] L. Anselin, *Exploratory Spatial Data Analysis and Geographic Information Systems*, In: M. Painho, *New Tools for Spatial Analysis*, pp. 45-54, 1994

[66] L. Anselin, *Local Indicators of Spatial Association: LISA*, Geographical Analysis, 27 (2), pp. 93 – 115, 1995

[67] X. Yao, *Research issues in spatio-temporal data mining*, Whitepaper submitted to UCGIS workshop on Geospatial Visualization and Knowledge Discovery, Lansdowne, VA, November, 2003.

[68] C. Aggarwal and P. Yu., *Outlier detection for high dimensional data*, In Proc. Of SIGMOD'2001, pages 37–47, 2001.

[69] K. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft, *When is Nearest Neighbors Meaningful?*, ICDT Conference Proceedings, 1999

[70] R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan, *Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications*, ACM SIGMOD Conference Proceedings, 1998

[71] K. Chakrabarti, S. Mehrotra, *Local Dimensionality Reduction: A New Approach to Indexing High Dimensional Spaces*, VLDB Conference Proceedings, 2000

[72] A. Hinneburg, C. C. Aggarwal, D. A. Keim, *What is the nearest neighbor in high dimensional spaces?*, VLDB Conference Proceedings, 2000

[73] A. Arning, R. Agrawal, P. Raghavan, *A linear method for deviation detection in large databases*, KDD Conference Proceedings, 1995

[74] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975

[75] C. Darwin, *The Origin of the Species by Natural Selection*, Published 1859

[76] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, *Optimization by simulated annealing*, Science (220) (4589): pp. 671-680, 1983

[77] K. A. De Jong, *Analysis of the Behaviour of a Class of Genetic Adaptive Systems*, Ph. D. dissertation, University of Michigan, Ann Arbor, MI, 1975

[78] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, MA, 1989

[79] C. C. Aggarwal, J. B. Orlin, P. Tai, *Optimized Crossover for the Independent Set Problem*, Operations Research 45(2), März, 1997

[80] M. Horton, A. Adams, *RFC 1036 – Standard for Interchange of USENET Messages*, IETF Network Working Group, Request for Comments: 1036, Internet Engineering Task Force , December 1987

[81] Brian Kantor, Phil Lapsley, *RFC 977 – Network News Transfer Protocol, a Proposed Standard for the Stream-Based Transmission of News*, IETF Network Working Group, Request for Comments: 977, Internet Engineering Task Force, February 1986

[82] Henry Spencer, David Lawrence, *Managing Usenet*, O'Reilly. First Edition, ISBN: 1-56592-198-4, 1997

[83] Mark Harrison, *The USENET Handbook – A User's Guide to Netnews*, O'Reilly, First Edition, ISBN: 1-56592-101-1, 2005

[84] Paul Wood, *Saving yourself from external Spamnation*, A MessageLabs Whitepaper, MessageLabs Ltd, September 2004

[85] David H. Crocker, *RFC 822 – Standard for the format of ARPA Internet text messages*, IETF Request for Comments: 822, Internet Engineering Task Force, August 1982

[86] P. Faltstrom, P. Hoffman, A. Costello, *RFC 3490 – Internationalizing Domain Names in Applications (IDNA)*, IETF Network Working Group, Request for Comments: 3490, Internet Engineering Task Force, März 2003

[87] Ultsch, A., Moerchen, F.: *ESOM-Maps: tools for clustering, visualization, and classification with Emergent SOM*, Technical Report, Dept. of Mathematics and Computer Science, University of Marburg, Germany, No. 46, 2005

[88] Kohonen, T.: *Self-Organizing Maps*, Springer, NY, 1995

[89] Ultsch, A.: *Data Mining and Knowledge Discovery with Emergent Self-Organizing Feature Maps for Multivariate Time Series*, In Kohonen Maps, (1999) , pp. 33-46

[90] Ultsch, A.: *U*C: Self-organizied Clustering with Emergent Feature Map*, In Proceedings Lernen, Wissensentdeckung und Adaptivität (LWA/FGML 2005), Saarbrücken, Germany, (2005), pp. 240-246

[91] Ultsch, A., Moutarde, F.: *U*F Clustering: a new performant Cluster-mining method based on segmentation of Self-Organizing Maps*, In Proceedings Workshop on Self-Organizing Maps (WSOM 2005), Paris, France, (2005), pp. 25-32

[92] Mierswa, Ingo and Klinkberg, Ralf and Fischer, Simon and Ritthoff, Oliver. *A Flexible Platform for Knowledge Discovery Experiments: YALE -- Yet Another Learning Environment*. In LLWA 03 - Tagungsband der GI-Workshop-Woche Lernen - Lehren - Wissen - Adaptivität, 2003.

[93] Fischer, Simon and Klinkenberg, Ralf and Mierswa, Ingo and Ritthoff, Oliver. *Yale: Yet Another Learning Environment -- Tutorial. No. CI-136/02*, Collaborative Research Center 531, University of Dortmund, Dortmund, Germany, 2002.

[94] Ritthoff, Oliver and Klinkenberg, Ralf and Fischer, Simon and Mierswa, Ingo and Felske, Sven. *Yale: Yet Another Machine Learning Environment*. In Klinkenberg, Ralf and Rüping, Stefan and Fick, Andreas and Henze, Nicola and Herzog, Christian and Molitor, Ralf and Schröder, Olaf (editors), LLWA 01 -- Tagungsband der GI-Workshop-Woche Lernen -- Lehren -- Wissen -- Adaptivität, No. Nr. 763, Seiten 84--92, Dortmund, Germany, 2001.

[95] Fischer, Simon, Mierswa, Ingo, *The YALE GUI Manual*, Chair of Artificial Intelligence, Department of Computer Science, University of Dortmund, SourceForge.net, October 2004

[96] Fischer, Simon, Klinkenberg, Ralf, Mierswa, Ingo, Ritthoff, Oliver, *YALE 2.4.1: Yet Another Learning Environment Tutorial (User Guide, Operator Reference, Developer Tutorial)*, Chair of Artificial Intelligence, Department of Computer Science, University of Dortmund, 2001-2004

[97] Wurst, Michael, *The YALE Cluster Plugin (User Guide, Operator Reference, Developer Tutorial)*, Chair of Artificial Intelligence, Department of Computer Science, University of Dortmund, 2004

[98] Mierswa, Ingo, *Value Series Preprocessing with YALE (User Guide, Operator Reference, Developer Tutorial)*, Chair of Artificial Intelligence, Department of Computer Science, University of Dortmund, 2004

[99] Wurst, Michael, Mierswa, Ingo, Fischer, Simon, *The YALE Word Vector Plugin (User Guide, Operator Reference, Developer Tutorial)*, Chair of Artificial Intelligence, Department of Computer Science, University of Dortmund, 2004

[100] M. Ester, H.-P. Kriegel, J, Sandner, X. Xu, *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, (DBSCAN), Proc 2nd. Int. Conf. On Knowledge Discovery and Data Mining, Portland, OR, AAAI Press, 1996, pp. 226-231

[101] J.B. Lovins, *Development of a Stemming Algorithm*, In Mechanical Translation and Computational Linguistics, 11(1-2), 11-31, 1968

[102] M. F. Porter, *An algorithm for suffix stripping*, In Readings in information retrieval, pages 313--316, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc., 1997

[103] G. Salton and C. Buckley, *Term-weighting approaches in automatic text retrieval*, Information Processing Management, 24(5):513--523, 1988

[104] G. Furnas, S. Deerwester, S. Dumais, T. Landauer, R. Harshman, L. Streeter and K. Lochbaum, *Information retrieval using a singular value decomposition model of latent semantic structure*, in The 11th International Conference on Research and Development in Information Retrieval, Grenoble, France: ACM Press, 1988, pp. 465--480.

[105] S. Deutsch, *The YALE Outlier Plugin (User Guide, Operator Reference, Developer Tutorial)*, Chair of Artificial Intelligence, Department of Computer Science, University of Dortmund, 2006

[106] Makhoul, John; Francis Kubala; Richard Schwartz; Ralph Weischedel: *Performance measures for information extraction.* In: Proceedings of DARPA Broadcast News Workshop, Herndon, VA, February 1999.

[107] Baeza-Yates, R.; Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. New York: ACM Press, Addison-Wesley. Seiten 75 ff.

[108] Z.He, X. Xu, S. Deng, *An Optimization Model for Outlier Detection in Categorical Data*, http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0503081, 2005

[109] Z.He, X. Xu, S. Deng, *A Unified Subspace Outlier Ensemble Framework for Outlier Detection in High Dimensional Spaces*, http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0505060, 2005

[110] Z.He, X. Xu, S. Deng, J.Z. Huang, *FP-Outlier: Frequent Pattern Based Outlier Detection*, Computer Science and Information Systems, Volume 02 , Issue 01 (June 2005) ISSN:1820-0214, ComSIS Consortium, 2005

[111] IDEA Softwarebeschreibung, http://www.gdpdu-portal.com/IDEA/IDEA_Allgemein.htm, 2006