Client-side Web Mining for Community Formation in Peer-To-Peer Environments

Kun Liu, Kanishka Bhaduri, Kamalika Das, Phuong Nguyen, Hillol Kargupta (2006)

Hendrik Ewerlin

TU Dortmund

15.01.2008

Schnellstart

Worum geht's?

Gegeben

Ein Peer-To-Peer Netzwerk mit irgendeiner Verbindungsstruktur.

Aufgabe

Bilde Communities von Peers, die ähnliche Interessen haben!

Zusätzliche Anforderungen

- ...ohne dass irgendjemand die Interessen des anderen erfährt.
- ...sodass eine hierarchische Struktur entsteht.

Schnellstart

Worum geht's?

Offene Fragen

1. Wieso P2P-Communities?
2. Was soll das heißen - „Ähnliche Interessen“?
3. Wie funktioniert die Formierung der Community?
   - Geheimhaltung?
   - Hierarchie?

Übersicht

Übersicht

Was ist Peer-To-Peer?

Peer-To-Peer ist ganz sicher nicht...

Client-Server

- Klare Aufteilung der Teilnehmer in Clients und Server
- Server bietet dem Client einen Dienst an (Asymmetrische Beziehung)
- Beispiele: Webserver, Mailserver, Vorlesung, Kaffeeautomat, ...
- +: Intuitive Sichtweise, einfache Protokolle
- –: Server ist Single Point of Failure und durch DOS-Attacken verwundbar
- –: Server wird bei aufwändigen Anwendungen möglicherweise zum Bottleneck (Speicher / Traffic / Rechenleistung)

Was ist Peer-To-Peer?

Peer-To-Peer (P2P)

- Beziehung zwischen gleichartigen Nutzern (Peers)
- Dezentrale und selbstorganisierende Netzwerke mit Overlay Routing
- Hohe Fluktuation, keine garantierte Verfügbarkeit
- +: Gigantische Mengen an verfügbaren Ressourcen (Speicher / Traffic / Rechenleistung)
- +: Lastverteilung, kein Single Point of Failure, keine Bottlenecks
- –: Aufwändige Protokolle und Algorithmen

Klassische P2P-Netzwerke

Klassische P2P-Netzwerke

1. Pseudo-P2P
   - Napstar (1999)
   - Zentralisierte Indexverzeichnisse
2. Chaos-P2P
   - Gnutella (2000)
   - Network Flooding
3. Strukturiertes P2P
   - CAN, Chord, Pastry, Distance-Halving, Skip-Net, ... (ab 2001)
   - Document Routing; Auffinden in $O(\log n)$ Hops

Mehr dazu: Online-Vorlesung „Peer-To-Peer Netzwerke“, Prof. Christian Schindelhauer, Uni Freiburg (SS 2006)

Wieso Communities?

Nachteil

- Peers haben von sich aus schon spezialisierte Interessen und Informationsbedürfnisse.
- Diese werden in klassischen Ansätzen nicht genug berücksichtigt.

Idee zur Verbesserung: P2P-Communities

- Peers mit Gemeinsamkeiten gruppieren
- Spezielle Anfragen zuerst an die Community stellen
- +: Schnelleres Auffinden von gesuchten Inhalten

Also:

Peers mit Gemeinsamkeiten gruppieren!

Übersicht

Interessen und Ähnlichkeit

Interessen können...

- explizit erfragt werden
- implizit ermittelt werden (z.B: durch Betrachtung angefragter Internetseiten)

Je nach Anwendungsfall sind verschiedene Repräsentationen denkbar.

Ansätze für P2P-Communities

Ansätze für P2P-Communities

1. Linkanalyse
   - „Self-Organization and Identification of Web Communities“, G.W. Flake, S. Lawrence, C. Lee Giles, Frans M. Coetzee (2002)
2. Vertrauen
   - „Trust-based Community Formation in Peer-To-Peer File Sharing Networks“, Yao Wang (2004)
3. Ontologieüberdeckung
   - „Semantic self-formation of communities of peers“, S. Castano, S. Montanelli (2005)
4. Attributähnlichkeit
   - „Efficient discovery of implicitly formed P2P-Communities“, M.S. Khambatti, K.D. Ryu, P. Dasgupta (2002)

Linkanalyse

Linkanalyse

- Betrachtung von Peers und deren Links untereinander
- Community $\hat{=}$ Teilmenge der Peers, sodass für jedes Mitglied gilt: Die Mehrzahl seiner Links führt zu Mitgliedern
- Identifikation ist für gegebene Startknoten durch einen einfachen Max-Flow-Algorithmus möglich.
- $+$: Vollständig unabhängig von Inhalt
- $-$: Erfordert Offenlegung der Links

Vertrauen & Reputation

Vertrauen & Reputation

- Vertrauen $\hat{=}$
  - Erwartung von A an zukünftiges Verhalten von B
  - basierend auf vergangenden Erfahrungen mit B
  - im Bezug auf ein bestimmtes Themengebiet
- Reputation $\hat{=}$ Vertrauen vieler Peers in B im Bezug auf ein Themengebiet

Vertrauen & Reputation

Trust-based Routing

- Problem bei Network Flooding: Man flutet auch Bereiche des Netzwerkes, wo die Peers keine Ahnung vom Thema haben.
  $\Rightarrow$ großer Aufwand
- Schlauer wäre: Nur die Bereiche fluten, wo die Peers Ahnung haben.
- Algorithmische Umsetzung:
  - Antworten auf Suchanfragen werden evaluiert:
  - Gute Antworten wecken Vertrauen; schlechte schmälern es
  - Beim nächsten Mal fragt man zuerst die Peers, denen man vertraut
  - Nur wenn das Suchbedürfnis nicht gestillt ist, fährt man fort
- $+$: Enorme Verringerung der Anfragekomplexität
- $+$: Tiefere Suche möglich

Vertrauen & Reputation

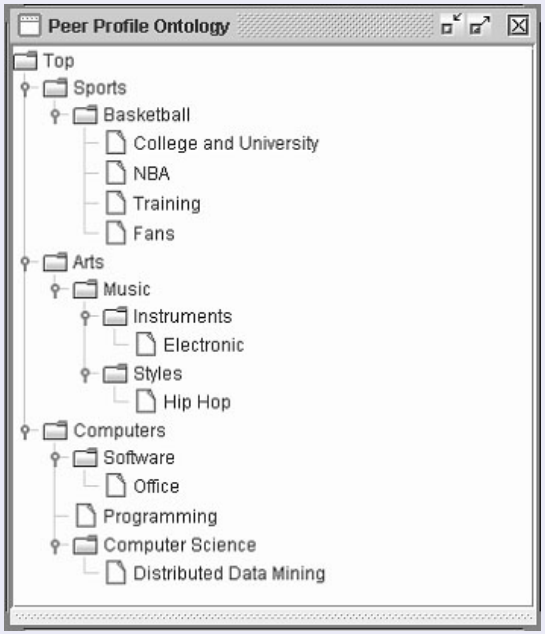
Trust-based Communities

- In themenbasierten Communities werden Dokumente zentral gesammelt und von Mitgliedern bewertet.
- Peers, deren Dokumente gut bewertet wurden, bekommen Vertrauen.
- Reputation hilft, auch neue Dokumente der Peers angemessen einzuordnen.
- +: Gute Inhalte werden gefördert.
- +: Störenfriede werden isoliert.

Ontologieüberdeckung

Ontologieüberdeckung

- Interessen liegen als Ontologien vor.
- Ein „Semantic Matchmaker“ legt die Ontologien von Peers übereinander und vergleicht sie auf Ähnlichkeit, indem er
  - Synonyme angemessen behandelt
  - den Pfad der Konzepte als Kontext berücksichtigt
  - schließlich Schnittmengen aufzeigt
- +: Semantisch reiche Profile
- –: Ungenau, mögliche Fehleinschätzung
- –: Offenlegung von Interessen erforderlich



Attributähnlichkeit

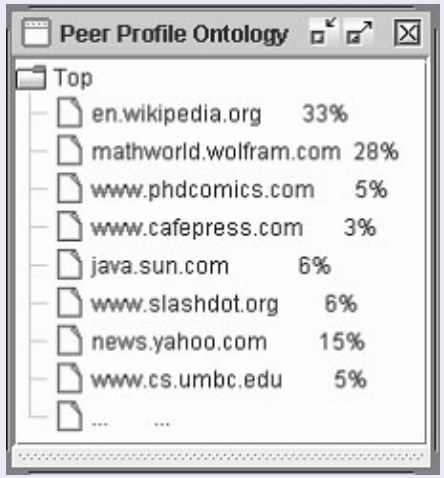
Attributähnlichkeit - Mengen

- Grundmenge von Attributen A vorgegeben
- Interessen sind Teilmengen dieser Grundmenge ($I_p \subseteq A$)
- Ähnlichkeitstest: Schnittmenge der Interessen bilden
- $+$: Einfach
- $+$: Keine Unsicherheit durch vorgegebene Grundmenge
- $-$: Müßig in Verbindung mit Benutzereingabe

Attributähnlichkeit

Attributähnlichkeit - Vektoren

- Grundmenge von Attributen A vorgegeben
- Interessen eines Peers sind Vektoren; Komponenten entsprechen den Attributen ($I_p : A \mapsto \mathbb{Z}_\mu$)
- $+$: Gewichtung möglich
- Ähnlichkeitsbegriff 1: Abstand im Interessenraum
- Ähnlichkeitsbegriff 2: Skalarprodukt



Attributähnlichkeit

Skalarprodukt

Für das Skalarprodukt von zwei Vektoren gilt:

$$\vec{x} \cdot \vec{y} = \sum_{i=1}^{n} x_i \cdot y_i$$

$$\vec{x} \cdot \vec{y} = |\vec{x}| \cdot |\vec{y}| \cdot \cos \alpha$$

α ist der von $\vec{x}$ und $\vec{y}$ eingeschlossene Winkel.
Für normierte Vektoren ($|\vec{x}| = |\vec{y}| = 1$) ist $\alpha = \arccos \vec{x} \cdot \vec{y}$.
$\Rightarrow$ Das Skalarprodukt ist stark mit der Cosinus Ähnlichkeit verwandt.

Übersicht

Formierung der Community

Gegeben

- Ein Peer-To-Peer Netzwerk mit irgendeiner Verbindungsstruktur.
- Interessensvektoren (Ähnlichkeit: Skalarprodukt)

Aufgabe

- Bilde Communities von Peers, die ähnliche Interessen haben!

Zusätzliche Anforderungen

- ...ohne dass irgendjemand die Interessen des anderen erfährt.
- ...sodass eine hierarchische Struktur entsteht.

Und so geht's

Und so geht's

- Der Ansatz basiert auf Network Flooding:
  - Es gibt einen Community Initiator.
  - Der kontaktiert seine d-Nachbarschaft,
  - bestimmt alle Ähnlichkeitswerte (Skalarprodukt)
  - und lädt alle ein, deren Ähnlichkeit größer als eine Grenze G ist.
- Also alles ganz einfach!

Die Frage ist nur...

1. Wie bestimmt man unter Geheimhaltung ein Skalarprodukt?
2. Wie ähnlich ist ähnlich genug? $G = ...$

Privates Skalarprodukt

Problem

Gegeben: Profilvektoren von zwei Peers.
Aufgabe: Bestimme das Skalarprodukt der Profilvektoren ohne dass ein Peer den Profilvektor des anderen erfährt.

Dafür brauchen wir etwas Cryptographie...

Privates Skalarprodukt

Public-key Cryptosysteme

$\hat{=}$ Algorithmen für Key **G**eneration, **E**ncryption und **D**ecryption.

- G generiert Schlüsselpaare $(priv, pub)$
- $E_{pub}(m, r)$ codiert eine Nachricht m
- $D_{priv}(c)$ decodiert einen Ciphertext
- Außerdem gilt:
  - $\forall (priv, pub) \in G : \forall m, r \in \mathbb{Z}_\mu : D_{priv}(E_{pub}(m, r)) = m$
  - G, E, D in Polynomialzeit berechenbar
  - Ohne $priv$ keine Decodierung in Polynomialzeit
  - Kein Erraten von $priv$ in Polynomialzeit

Privates Skalarprodukt

Homomorphe Public-key Cryptosysteme

$\hat{=}$ Public-key Cryptosysteme + Rechenoperationen auf Klartext durch Manipulation von Ciphertext:

$$\forall m_1, m_2, r_1, r_2 \in \mathbb{Z}_\mu :$$
$$D_{priv}(E_{pub}(m_1, r_1)E_{pub}(m_2, r_2) \mod \mu^2) = m_1 + m_2 \mod \mu$$
$$D_{priv}(E_{pub}(m_1, r_1)^{m_2} \mod \mu^2) = m_1 m_2 \mod \mu$$

Sowas gibt's.

Privates Skalarprodukt

Algorithmus

1. A...
   - kennt $(a_1, \ldots, a_n)$, $(priv, pub)$
   - berechnet $(E(a_1), \ldots, E(a_n))$ durch Codieren und verschickt es
2. B...
   - kennt $(b_1, \ldots, b_n)$
   - berechnet $(E(a_1 \cdot b_1), \ldots, E(a_n \cdot b_n))$ durch Potenzieren
   - berechnet $E(\sum_{i=1}^{n} a_i \cdot b_i)$ durch Multiplizieren und verschickt es
3. A...
   - berechnet $\sum_{i=1}^{n} a_i \cdot b_i \mod \mu$ durch Decodieren

Wie ähnlich ist ähnlich genug?

Zwischenergebnis

- Wir können Ähnlichkeit zu anderen Peers berechnen. (Und das auch noch so richtig vertraulich.)
- Communitybildung ist einfach: Wir laden alle d-Nachbarn ein, deren Ähnlichkeit größer als G ist.

Problem

Wie ähnlich ist ähnlich genug? $G = ...$

Wie ähnlich ist ähnlich genug?

Schlechte Ideen

1. Wähle G fest, irgendwie.
   Wie viele Peers sind dann ähnlicher als G? 50%, 80%, 0%, 100%?
   Wir wissen nicht, wie die Ähnlichkeit im Netzwerk verteilt ist!
2. Wir haben doch unsere d-Nachbarn. Wähle G so, dass es die d-Nachbarn geeignet teilt. Niemand sagt, dass unsere Nachbarn repräsentativ für das ganze Netzwerk sind!
3. Befrage alle Peers und wähle G passend.
   Lieber nicht!

Fazit

Wir brauchen einen schlaues Verfahren mit

1. beherrschbarem Aufwand
2. globaler Gütegarantie für G

Wie ähnlich ist ähnlich genug?

$$|x_1| < |x_2| < | \quad \dots \quad | < |x_i| < | \quad \dots \quad | < |x_{n-1}| < |x_n|$$

Gedankenexperiment (1)

- Angenommen es gibt n Peers im Netzwerk.
- Deren Ähnlichkeitswerte $x_1, ..., x_n$ sind vollständig bekannt und aufsteigend sortiert $(x_1 < ... < x_n)$

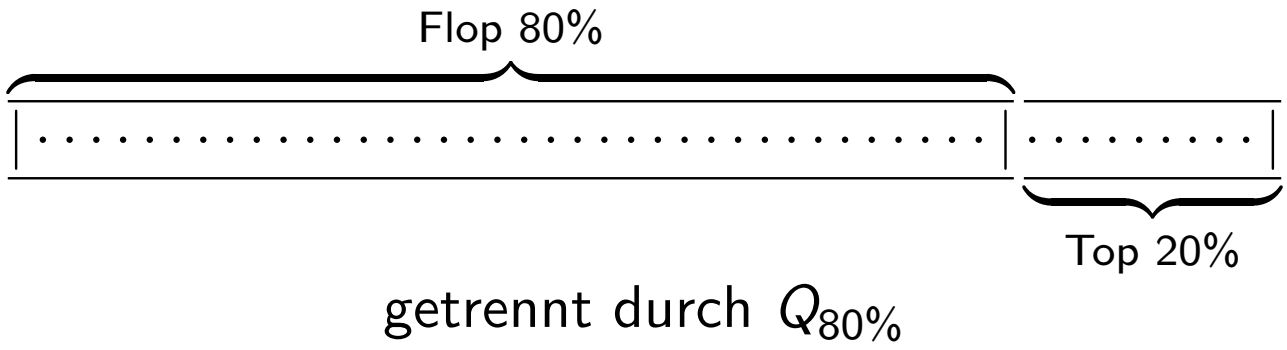
Wie ähnlich ist ähnlich genug?

$$\overbrace{\underbrace{|x_1| < |x_2| < | \quad \cdots \quad | < |x_i|}_{=:G} < | \quad \cdots \quad | < |x_{n-1}| < |x_n|}^{i \text{ von } n \text{ Peers mit } x \leq G}$$

Gedankenexperiment (2)

- Angenommen x_i wird als Grenzwert G genommen.
- Dann gibt es genau i Peers mit $x \leq G$.
- Bei zufällig gleichverteilter Wahl erwischt man einen solchen Peer mit Wahrscheinlichkeit $p := \frac{i}{n}$.

Wie ähnlich ist ähnlich genug?



Gedankenexperiment (3) - Definition p-Quantil

- Eine Grenze mit dieser Eigenschaft heißt p-Quantil Q_p:
  $Pr(x \leq Q_p) = p$
  (x = Ähnlichkeit eines zufällig gleichverteilt gewählten Peers)
- Solche Grenzen zu bestimmen ist interessant. Angenommen, wir kennen $Q_{80\%}$. Dann können wir bestimmen, ob ein Peer zu den Top 20% ähnlichsten Peers im Netzwerk gehört.

Grenzwert durch Stichprobe

Jetzt: Die Wunderwaffe

Algorithmus zum Bestimmen einer Grenze G, die mit einer Wahrscheinlichkeit von mindestens q das p-Quantil übertrifft.

Algorithmus

- Wähle zufällig gleichverteilt $N := N(p, q)$ Peers aus dem Netzwerk.
- Bestimme deren Ähnlichkeiten x_i.
- Setze $G := \max\{x_i\}$.

Grenzwert durch Stichprobe

Analyse

Wie groß ist die Wahrscheinlichkeit, dass diese Grenze größer als Q_p ist?

$$\begin{aligned} Pr(G > Q_p) &= 1 - Pr(G \leq Q_p) \\ &= 1 - Pr(\max\{x_i\} \leq Q_p) \\ &= 1 - Pr(\text{Alle } x_i \leq Q_p) \\ &= 1 - p^N > q \end{aligned}$$

Wie wählt man N für gegebenes p, q?

$$N(p,q) := \left\lceil \frac{\log(1-q)}{\log(p)} \right\rceil$$

Grenzwert durch Stichprobe

Zahlenbeispiele (1)

Wir wollen eine Grenze G, die mit einer Wahrscheinlichkeit von mindestens q das p-Quantil übertrifft. Wie viele Peers fragen wir?

p	q	$N(p,q)$
80%	80%	8
85%	80%	10
90%	80%	16
95%	80%	32
80%	85%	9
85%	85%	12
90%	85%	19
95%	85%	37

p	q	$N(p,q)$
80%	90%	11
85%	90%	15
90%	90%	22
95%	90%	45
80%	95%	14
85%	95%	19
90%	95%	29
95%	95%	59

Grenzwert durch Stichprobe

Zahlenbeispiele (2)

Wir haben N Peers befragt. Wie sicher können wir sein, dass wir ein bestimmtes p-Quantil übertreffen?

p	N	$q(p, N)$
70%	5	83%
80%	5	67%
90%	5	41%
95%	5	23%
70%	10	97%
80%	10	89%
90%	10	65%
95%	10	40%

p	N	$q(p, N)$
70%	15	100%
80%	15	96%
90%	15	79%
95%	15	54%
70%	20	100%
80%	20	99%
90%	20	88%
95%	20	64%

Grenzwert durch Stichprobe

Fazit

Der Algorithmus „Grenzwert durch Stichprobe“ liefert uns eine gute Grenze G mit

1. beherrschbarem Aufwand (sogar unabhängig von der Netzwerkgröße!)
2. einer globalen Gütegarantie

...vorausgesetzt: Wir können zufällig gleichverteilt Peers im Netzwerk kontaktieren.

Zufällige Peers im Netzwerk

Problem

Wie findet man mit uniformer Verteilung zufällige Peers im Netzwerk?

Lösung

Random Walk!

Grundidee

- Initiator erzeugt Random Walk Paket mit Kontaktdaten und Lebensdauer.
- Solange die Lebensdauer nicht abgelaufen ist, wird das Paket an einen zufälligen Nachbarn weitergeleitet.
- Der Peer, der das abgelaufene Paket erhält, meldet sich beim Initiator.

Random Walks

Wahl des nächsten Peers

Entscheidene Frage

Wie wählt man den nächsten Peer?

Metropolis-Hastings Random Walk

p_{ij} sei die Wahrscheinlichkeit, dass Peer i das Paket an j sendet. d_i sei der Grad des Knotens i. N_i sei die Nachbarschaftsmenge von Peer i.

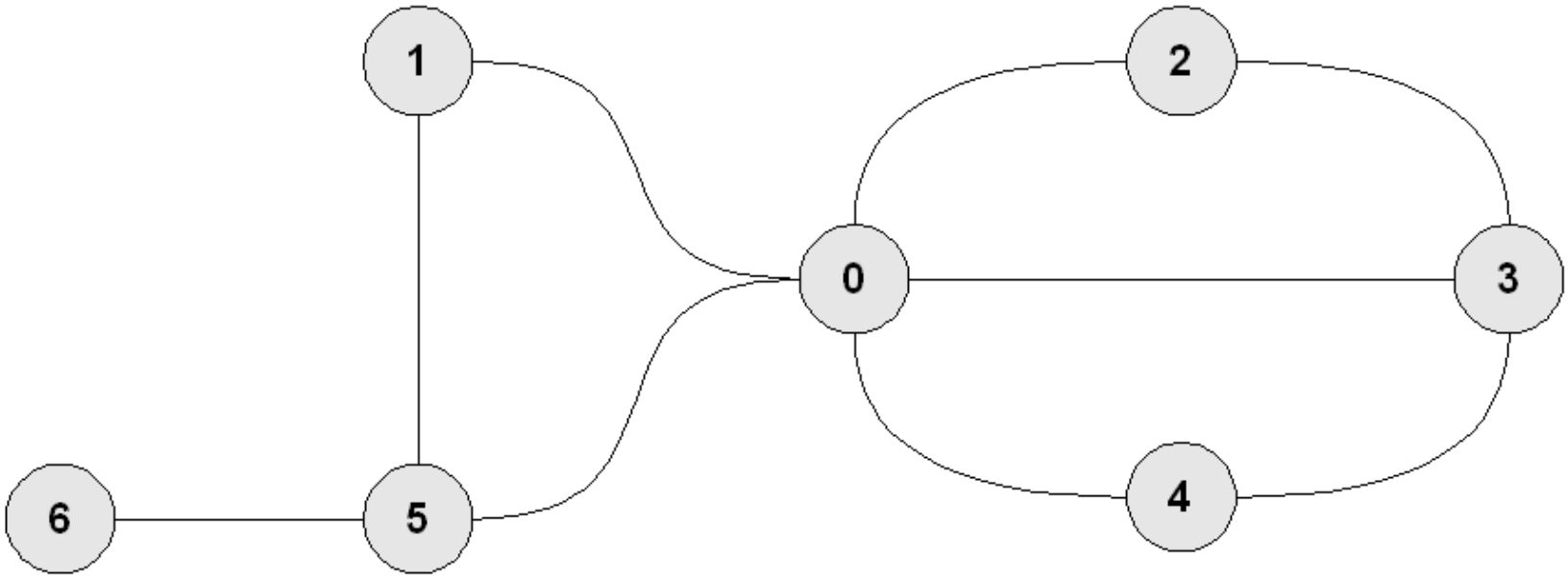
$$p_{ij} := \begin{cases} 0 & j \notin N_i, i \neq j \\ \frac{1}{\max(d_i, d_j)} & j \in N_i, i \neq j \\ 1 - \sum_{j \in N_i} p_{ij} & i = j \end{cases}$$

führt für Wege der Länge $O(\log n)$ zu uniformer Verteilung.

Random Walks

Beispiel (1)

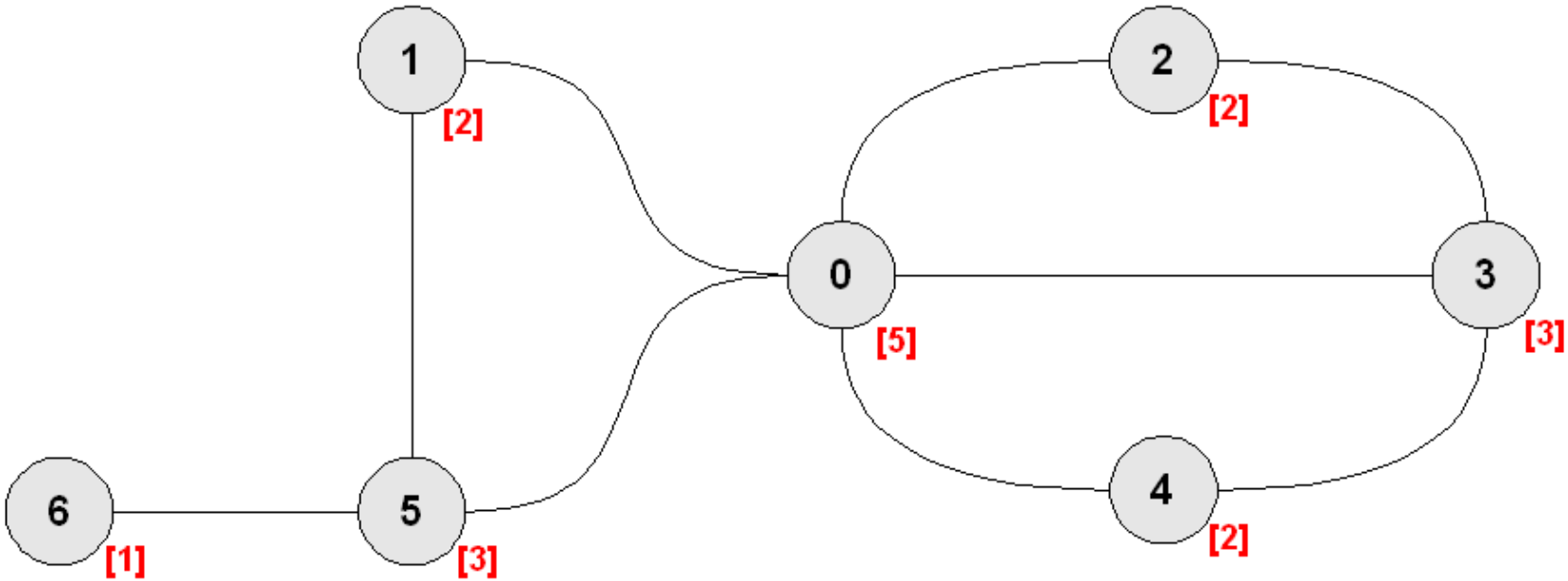
Man betrachte das folgende Netzwerk:



Random Walks

Beispiel (2)

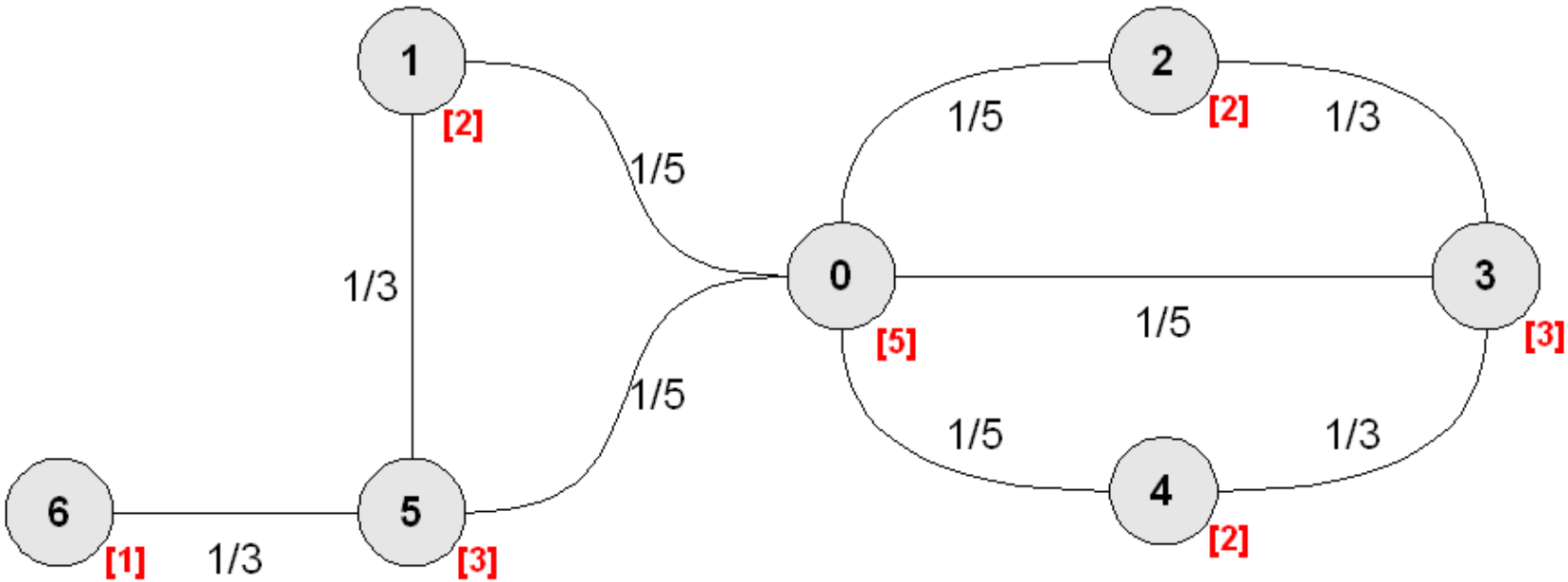
Knoten mit Knotengrad beschriften:



Random Walks

Beispiel (3)

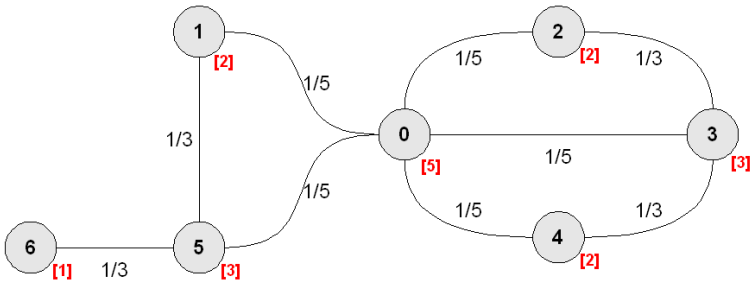
Kanten mit 1/(Größter anliegender Knotengrad) beschriften:



Random Walks

Beispiel (4)

Ergänzen der jeweiligen Gegenwahrscheinlichkeiten auf der Diagonalen ergibt die vollständige Transitionsmatrix.



$$\begin{pmatrix} 0 & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & 0 \\ \frac{1}{5} & \frac{7}{15} & 0 & 0 & 0 & \frac{1}{3} & 0 \\ \frac{1}{5} & 0 & \frac{7}{15} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{5} & 0 & \frac{1}{3} & \frac{2}{15} & \frac{1}{3} & 0 & 0 \\ \frac{1}{5} & 0 & 0 & \frac{1}{3} & \frac{7}{15} & 0 & 0 \\ \frac{1}{5} & \frac{1}{3} & 0 & 0 & 0 & \frac{2}{15} & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{2}{3} \end{pmatrix}$$

Random Walks

Beispiel (5)

Was passiert für längere Pfade?

$$M^{32} = \begin{pmatrix} 14\% & 14\% & 14\% & 14\% & 14\% & 14\% & 14\% \\ 14\% & 14\% & 14\% & 14\% & 14\% & 14\% & 14\% \\ 14\% & 14\% & 14\% & 14\% & 14\% & 14\% & 14\% \\ 14\% & 14\% & 14\% & 14\% & 14\% & 14\% & 14\% \\ 14\% & 14\% & 14\% & 14\% & 14\% & 14\% & 14\% \\ 14\% & 14\% & 14\% & 14\% & 14\% & 14\% & 14\% \\ 14\% & 14\% & 14\% & 14\% & 14\% & 14\% & 14\% \end{pmatrix}$$

Man sieht: Diese Initialisierung führt für hinreichend lange Wege zu uniformer Verteilung. (Animation: siehe Fassung mit Übergängen)

Ablauf: Formierung der Community

1. Größe der Stichprobe berechnen
   - $N := N(p, q) = \left\lceil \frac{\log(1-q)}{\log(p)} \right\rceil$
2. Stichprobe auswerten
   - N zufällige Peers im Netzwerk kontaktieren (RandomWalk)
   - Ähnlichkeiten bestimmen, Maximum zur Akzeptanzgrenze G erklären
3. Mitglieder identifizieren
   - Fluten der d-Nachbarschaft und Kontaktaufnahme
   - Massenhafte Ähnlichkeitsbestimmung
4. Einladen
   - Einladen aller Peers, die das Ähnlichkeitsmaß G überschreiten.
5. Erweitern
   - Gegenseitiger Community-Austausch
   - Ähnlichkeit ist - wenigstens ein bisschen - transitiv.

Message Complexity (1)

1. Größe der Stichprobe berechnen
   - N Randomwalks der Länge λ, 64 Bits pro Paket
   - $64 * \lambda * N$
2. Stichprobe auswerten
   - N Peers in der Stichprobe
   - Vektor hat d Komponenten, jeder Ciphertext benötigt $2 * |\mu|$ Bits
   - $2 * |\mu| * (d + 1) * N$
3. Mitglieder identifizieren
   - Fluten des Netzwerks: M Peers erhalten $\alpha \geq 1$ Pakete, 64 Bit Daten
   - $64 * \alpha * M$
   - Ähnlichkeitsbestimmung mit M Peers
   - $2 * |\mu| * (d + 1) * M$

Message Complexity (2)

4. Einladen
   - Einladen von O Peers, P nehmen an
   - $8 * (O + P)$ Bits
5. Erweitern
   - Gegenseitiger Community-Austausch mit P Peers, β durchschnittliche Anzahl von Peers in erweiterten Communities
   - $64 * (P^2 + P * \beta)$

Message Complexity (3)

Überblick

Aktion	Formel (Bit)	Byte	kByte
Random Walk	$64 * \lambda * N$	12320	12
Skalarprodukt	$2 * \|\mu\| * (d+1) * (N+M)$	513044	501
Netzwerk fluten	$64 * \alpha * M$	16000	15
Einladung	$8 * (O+P)$	100	0
Austausch	$64 * (P^2 + P * \beta)$	32000	31
Insgesamt	$\sum \ldots$	573464	560

Rechenbeispiel

$p = q = 90\% \Rightarrow N = 22$
$\lambda = 70$
$|\mu| = 8,\ d = 250$

$M = 1000, \alpha = 2$
$O = P = 50, \beta = 30$

Experimentelle Ergebnisse

Schwellwert & Quantil

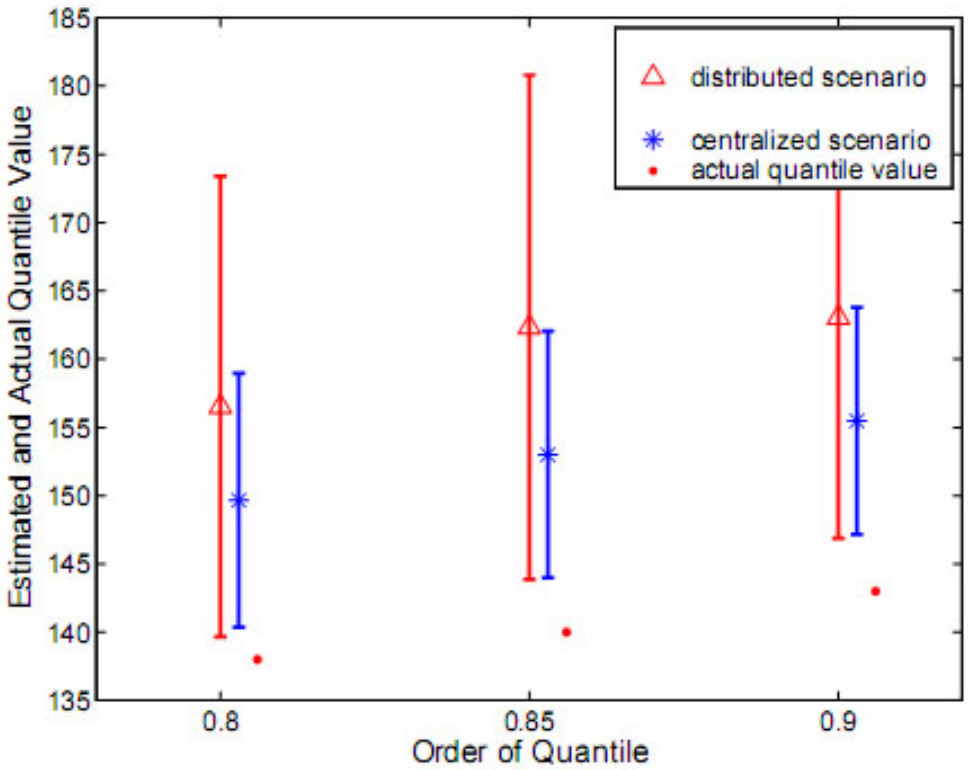


Figure 5: Estimated and actual quantile value w.r.t. the order of quantile. The results are an average of 100 independent runs.

Experimentelle Ergebnisse

Zeitaufwand & Privates Skalarprodukt

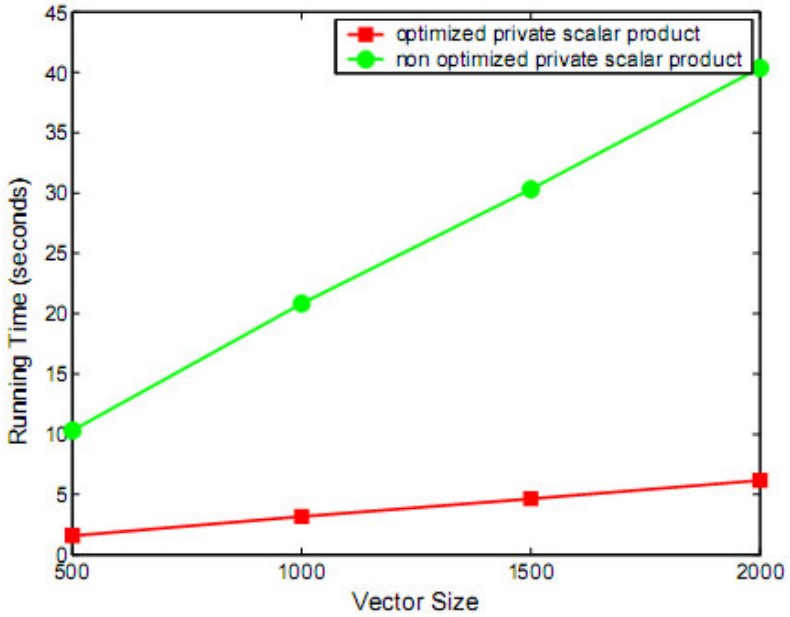


Figure 7: Time required to compute private scalar product with varying dimension of the vector. The results are an average of 10 independent runs.

TTL	Ave Num of Community Members	Time (in secs)
3	3	55.00
4	8	77.50
8	13	173.00

Table 2: Average number of community members found by the initiator without community expansion.

TTL	Ave Num of Community Members	Time (in secs)
3	7	59.00
4	12	82.50
8	17	179.00

Table 3: Average number of community members found by the initiator with community expansion.

Übersicht

Zusammenfassung

Zusammenfassung

1. Peer-To-Peer Netzwerke
   - Client-Server VS Peer-To-Peer
   - 3 Arten von Peer-To-Peer Netzwerken
2. Peer-To-Peer Communities
   - Verbesserte Suchzeit durch Gruppierung von Peers mit Gemeinsamkeiten
     - Linkanalyse
     - Vertrauen
     - Ontologieüberdeckung
     - Attributähnlichkeit
3. Peer-To-Peer Community nach Liu et al. (2006)
   - basierend auf Profilvektoren, Skalarprodukt
   - Fluten des Netzwerkes zur Identifikation von Mitgliedern
   - Highlight 1: Private Skalarproduktberechnung
   - Highlight 2: Zufallsstichprobe zur Einschätzung des Grenzwertes

Zusammenfassung

Und: Echte Lebenshilfe!

- Angenommen, du warst in 22 Restaurants.
- Danach findest du ein Restaurant, in dem es dir noch besser schmeckt.
- Dann kannst du ab heute mit 90%iger Sicherheit allen erzählen, dass es sich um eines der Top-10% Restaurants weltweit handelt!
- Immer vorausgesetzt: Du wählst deine Restaurants zufällig gleichverteilt. ;-)
- Danke für die Aufmerksamkeit!