

Übungen zur Vorlesung **Maschinelles Lernen**

Wintersemester 2008/2009

Allgemeines

Begleitend zur Vorlesung findet eine Übung statt. Die Teilnahme an der Übung, sowie die Bearbeitung der Übungsblätter ist Voraussetzung für einen Scheinerwerb (s.u.).

- Termin: Donnerstags 14 - 16 Uhr in Raum 113/GB IV, Campus Süd
- Anmeldung zur Übungsgruppe in der ersten Vorlesung.
- Ausgabe der Übungsblätter: Dienstags in der Vorlesung bzw. im Netz

Erster Termin In der ersten Übung am Donnerstag, dem 16.10. findet von 14 - 16 Uhr eine kurze Vorstellung der Software zur Bearbeitung der Übungen statt.

Abgabemodalitäten

Die Abgabe der bearbeiteten Übungsblätter ist bis zum Dienstagabend der folgenden Woche, 23.59 Uhr per eMail an `christian.bockermann@cs.uni-dortmund.de` zu schicken. Bei Abgaben, die Implementierungen enthalten, schicken Sie bitte neben dem Quellcode zusätzlich das von der Umgebung erstellte Plugin mit (siehe unten).

Die Abgabe kann in *einmalig zu Beginn festgelegten Gruppen* zu maximal 3 Personen erfolgen. Auf der Gruppenabgabe bitte sämtliche Namen vermerken!

Scheinerwerb

Jede Abgabe wird mit bis zu 10 Punkten bewertet. Die Bedingungen zum Scheinerwerb sind

- mindestens 80% der erreichbaren Punkte

Die Notenvergabe für den Schein erfolgt nach erreichter Punktzahl bei den Abgaben.

Begleitunterlagen und Übungsblätter

Die Folien zur Vorlesung, sowie die Übungsblätter sind online verfügbar unter

<http://www-ai.cs.uni-dortmund.de/LEHRE/VORLESUNGEN/WS09/MLV/>

Die RapidMiner Lernumgebung

Eine Vielzahl der Algorithmen und Vorverarbeitungsschritte, die innerhalb der Vorlesung vorgestellt werden, sind in dem Java-basierten Tool **RapidMiner** integriert. **RapidMiner** wurde als Open-Source Projekt am Lehrstuhl 8 entwickelt.

Mit Hilfe von **RapidMiner** haben Sie die Möglichkeit, eigene Daten-Experimente zur Klassifikation, dem Clustering und andere Lernaufgaben selbst durchzuführen. Darüber hinaus ist ein Bestandteil der Übung die Implementierung vorgestellter Algorithmen innerhalb des **RapidMiner**-Frameworks (siehe "Implementierungsaufgaben").

Implementierungsaufgaben

Die Übungen zur Vorlesung beziehen eine Reihe von Implementierungen der vorgestellten Algorithmen ein. Durch die Implementierung werden sie feststellen, welche Probleme der Algorithmen auf großen Datenmengen erst haben.

Die Algorithmen werden innerhalb des **RapidMiner** ¹ Data-Mining Tools implementiert und können in Form eines Plugins in das Tool integriert werden. Dazu werden wir zu den jeweiligen Aufgaben Java-Interfaces bereitstellen, die Sie zur Lösung der Aufgabe implementieren sollen.

Für die Implementierung benötigen Sie eine Entwicklungsumgebung für Java, einen Client für das Subversion Versionsmanagement und das **ant**-BUILTtool.

Mit der Eclipse-IDE² und dem zugehörigen **subclipse**-Plugin³ bekommen Sie eine kostenlose und sehr leistungsfähige Programmierumgebung, die zudem auch in vielen Unternehmen als Entwicklungsumgebung dient.

Aufgaben im SVN-Repository

Die für die Aufgaben benötigten Klassen und Interfaces werden mit Herausgabe des Übungsblattes in das Subversion-Repository integriert, so dass Sie auf diese einfach zugreifen können, indem Sie einmalig das Repository aus-checken und dann zu gegebener Zeit jeweils aktualisieren.

Die URL für das Repository ist:

`https://kissen.cs.uni-dortmund.de/svn/mlv-uebung`

Mit Hilfe dieser URL läßt sich ein *Eclipse*-Projekt aus-checken, innerhalb dessen Sie die Lösungen der Aufgaben implementieren können.

¹RapidMiner ist ein am LS8 entwickeltes Open-Source Data-Mining Tool. Es ist über die Vorlesungsseite bzw. unter <http://rapid-i.com> erhältlich.

²Eclipse ist verfügbar unter <http://www.eclipse.org>.

³Das Plugin, bzw. die Eclipse-Update URL finden Sie unter <http://subclipse.tigris.org>.

Check-Out “von Hand” Mit Hilfe der Eclipse-Umgebung, dem subclipse-Plugin und der obigen Repository-URL können Sie alle Operationen auf dem Repository über Eclipse anstoßen. Nachfolgend wird der Umgang mit dem Repository mit Hilfe des `svn`-Kommandos beschrieben.

Mit dem Befehl

```
svn co https://kissen.cs.uni-dortund.de/svn/mlv-uebung
```

wird die aktuelle Version der Übungsumgebung in das lokale Verzeichnis `mlv-uebung` geladen. Um nach erstmaligem Check-Out die lokalen Dateien zu aktualisieren (z.B. um die nächsten Übungsaufgaben zu erhalten), ist der `update`-Befehl erforderlich, den Sie von ihrem lokalen `mlv-uebung`-Verzeichnis aufrufen:

```
svn update
```

Struktur der Übungsumgebung

Sobald Sie die Umgebung für die Übungsaufgaben (`mlv-uebung`) mittels SVN heruntergeladen haben, steht Ihnen ein Java-Projekt mit der folgenden Verzeichnisstruktur zur Verfügung:

```
mlv-uebung/  
mlv-uebung/exercises  
mlv-uebung/exercises/com/rapidminer/exercises/KNN.java  
...  
mlv-uebung/solutions  
mlv-uebung/lib/rapidminer.jar  
mlv-uebung/build.properties  
mlv-uebung/build.xml
```

Sowohl das `exercises` als auch das `solutions` Verzeichnis sind Quellcode-Verzeichnisse. Das Verzeichnis `exercises` enthält die für die Aufgaben benötigten Klassen und Interfaces. Im Verzeichnis `solutions` sollen Sie Ihre Lösungen der Aufgaben erstellen. In diesem Verzeichnis werden keine Dateien von uns bereitgestellt, so dass darin erstellte Dateien auch nicht überschrieben werden.

Erstellen des Plugins Die `build.xml`-Datei enthält die Anweisungen um mit Hilfe des `ant`-Tools aus allen Übungen und den von Ihnen erstellten Lösungen ein RapidMiner-Plugin zu erzeugen. Wenn Sie `ant` innerhalb des Verzeichnisses `mlv-uebung` aufrufen, wird das Plugin erstellt:

```
[bockerma@kieker ~]$ cd mlv-uebung  
[bockerma@kieker mlv-uebung]$ ant  
Buildfile: build.xml
```

```
init:
  [mkdir] Created dir: /home/bockerma/mlv-uebung/build

compile:
  [javac] Compiling 2 source files to /home/bockerma/mlv-uebung/build

plugin:
  [jar] Updating jar: /home/bockerma/mlv-uebung/RapidMiner-Exercises-4.1.jar

BUILD SUCCESSFUL
Total time: 0 seconds
```

Verwendung des Plugins Damit die Klassen aus dem Plugin innerhalb von RapidMiner genutzt werden können, muss das entstandene Jar-Archiv zur RapidMiner-Installation hinzugefügt werden. Dazu wird das Archiv einfach in das Verzeichnis `lib/plugins/` innerhalb des RapidMiner-Verzeichnisses kopiert.

Auch dies lässt sich mit Hilfe des `ant`-Tools vornehmen. In der Datei `build.properties` muss dazu lediglich der Parameter `rapidminer.home` auf das Verzeichnis der lokalen RapidMiner-Installation gesetzt werden. Danach führt der Aufruf von

```
ant install
```

im Verzeichnis `mlv-uebung` dazu, dass das Plugin direkt an die richtige Stelle kopiert wird.

Beim nächsten Start von RapidMiner sollte das Plugin geladen werden und die zusätzlichen Operatoren zur Verfügung stehen.