

AI Tools



YALE Yet Another Learning Environment

Arndt Faulhaber

14.12.2006

**Seminar: AI Tools (WS 2006/07)
Universität des Saarlandes**

Betreuer:

Michael Feld

Übersicht



- **Teil 1:**

Vergleich:

WEKA : YALE

- **Teil 2:**

Anwendung:

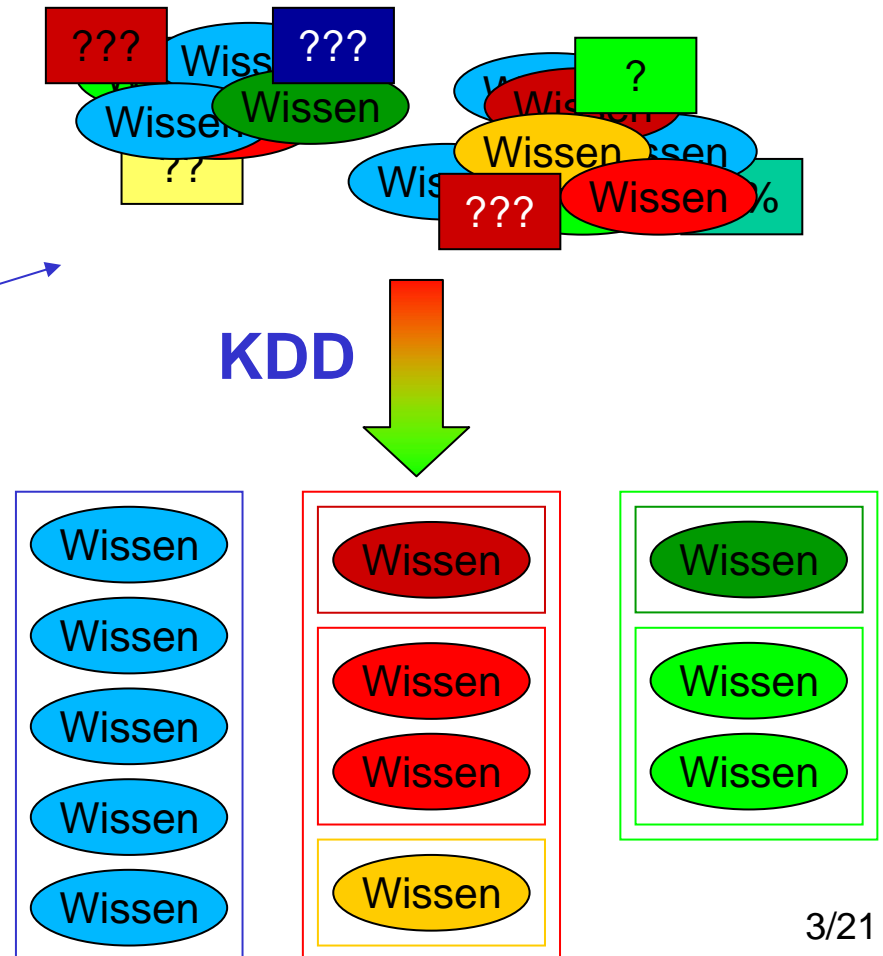
YALE im Java-Projekt

- **Zusammenfassung, Diskussion**

Motivation



- Entwicklung: Computer und Netzwerke
 - Ungeheure Informationsmengen (z.B. WWW)
 - Babylonisches Gewirr an Repräsentationen
- Erschließen von Wissen aus Daten
 - Datenmenge manuell nicht auswertbar
 - => Automatisiert: Knowledge Discovery and Data Mining (KDD)



Motivation



- **Beispiel 1:**
Zuordnen von Dokumenten zu Themen
 - Klassisches IR-Problem
 - **Jedoch:**
Dokumente meist „kontaminiert“ durch Tags, Formatierungen etc.**=> Vorverarbeitung**
- **Beispiel 2:**
Datenbank mit Informationen
 - Daten strukturiert
 - **Jedoch:**
Möglicherweise nur ein Teilstring relevant**=> Vorverarbeitung**

1

```
<section> Polluted Text </section>  
</subsection> More polluted Text  
</subsection>
```

Wortsuche:

```
{section(2), subsection(2),  
  more(1), polluted(2), text(2)}
```

Ziel: keine Tags!

```
{more(1), polluted(2), text(2)}
```

2

Inhalt DB:

```
„Adenauer-Str. 34“
```

Ziel:

```
„Adenauer-Str.“
```

YALE



- Framework für KDD / Maschinelles Lernen
- Start: 2001, Universität Dortmund, am Lehrstuhl für KI
- 2002 erste Version
- Seit 2004 auf SourceForge (OpenSource)
- Java Bibliothek, GPL
- Yet Another...
 - Keine passenden, frei verfügbaren Bibliotheken für Maschinelles Lernen (ML) / KDD
 - Vorverarbeitung (zusätzliche Skripte nötig)
 - Variierbare Ansätze (z.B. Auswahl eines besten Klassifizierers)
 - Einfaches Variieren von verwendeten Datensätzen (z.B. für Cross-Validation)
 - Flexible Attributerstellung und –auswahl

ML-Werkzeuge



WEKA : YALE

ML-Werkzeuge: Übersicht Aufbau



WEKA

- Lineare Abarbeitung des Experiments:
 - Daten laden
 - Klassifizieren
 - (Clustern)
 - Validieren (k-fold X)
 - Visualisieren

YALE

- Projekte als Operatorenbäume (beliebig große Unterbäume)
 - Kombination von vorhandenen Operatoren ergibt große Menge von Metaoperatoren

ML-Werkzeuge: Übersicht Vorverarbeitung



WEKA

- Begrenzte Möglichkeit zur Daten-
vorverarbeitung
 - Datenquellen
 - CSV, DB, ARFF
 - Binary, C45
 - Filter für
Attributen/Instanzen
 - Ansonsten selbst
programmierte
Vorverarbeitung

YALE

- Vorverarbeitung auch ein
Operatorbaum
 - Div. Datenquellen
 - Parsen (per Regexp)
 - Weitere Vorverarbeitung
über Filter (ggf. Plugins)
 - Ähnlich wie in WEKA
 - Fouriertransformation
 - Attribute/Instanzen:
verschiede Kriterien (z.B.
Attributwerte)

ML-Werkzeuge: Übersicht Diverses



WEKA

- Teilweise mehrfache Kopien der Daten im Speicher

YALE

- Kein Kopieren von Daten im Speicher
- Verschiedene Views der Daten
- Plugin-Architektur

ML-Werkzeuge: Vorverarbeitung I



- Auswahl der Quelle
- Modifikation der Daten
 - Parsen (reguläre Ausdrücke)
- Überarbeiten der Attribute
- Repräsentation der Daten

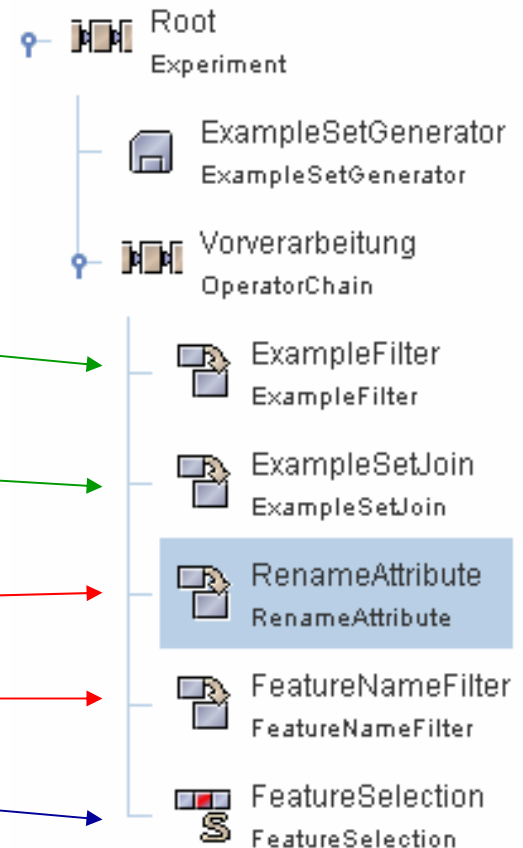
Key	Value
configure_operator	Start configuration wizard...
attributes	C:\Programme\YALE\yale-3.4\ Edit ...
sample_ratio	1.0
sample_size	-1
datamanagement	double_array
column_separators	,\s* \s*\s+
comment_chars	#
decimal_point_character	.
use_quotes	<input type="checkbox"/>
permutate	<input type="checkbox"/>
local_random_seed	-1

ML-Werkzeuge: Vorverarbeitung II



- Einige Vorverarbeitungs-Beispiele:

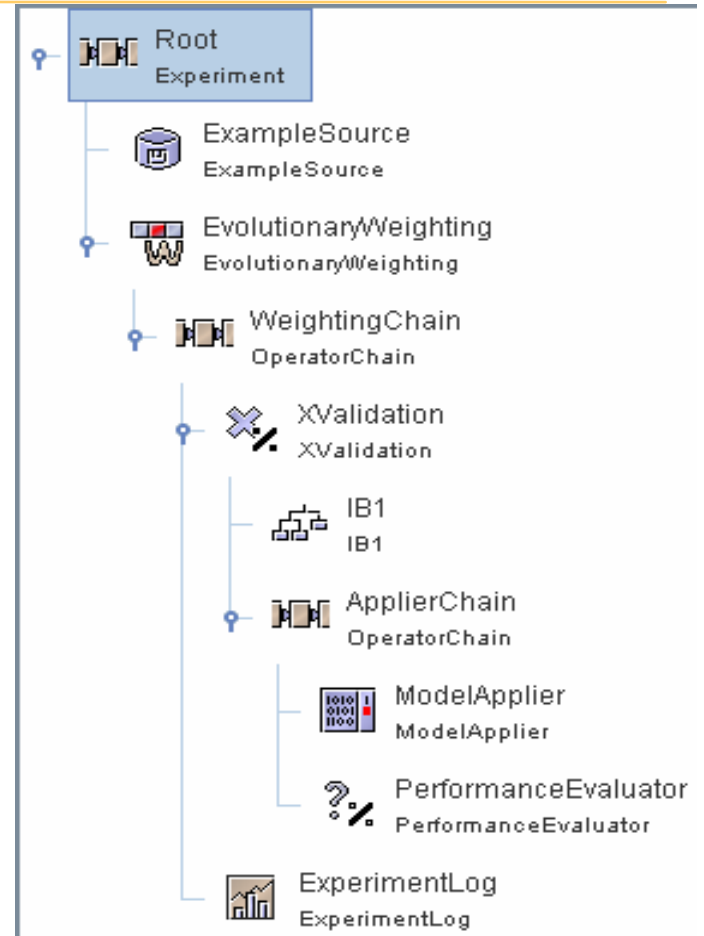
- Instanzfilter
- JOIN von Instanzmengen
- Attribute umbenennen
- Attributfilter
- Attributauswahl



YALE: Operatorbäume



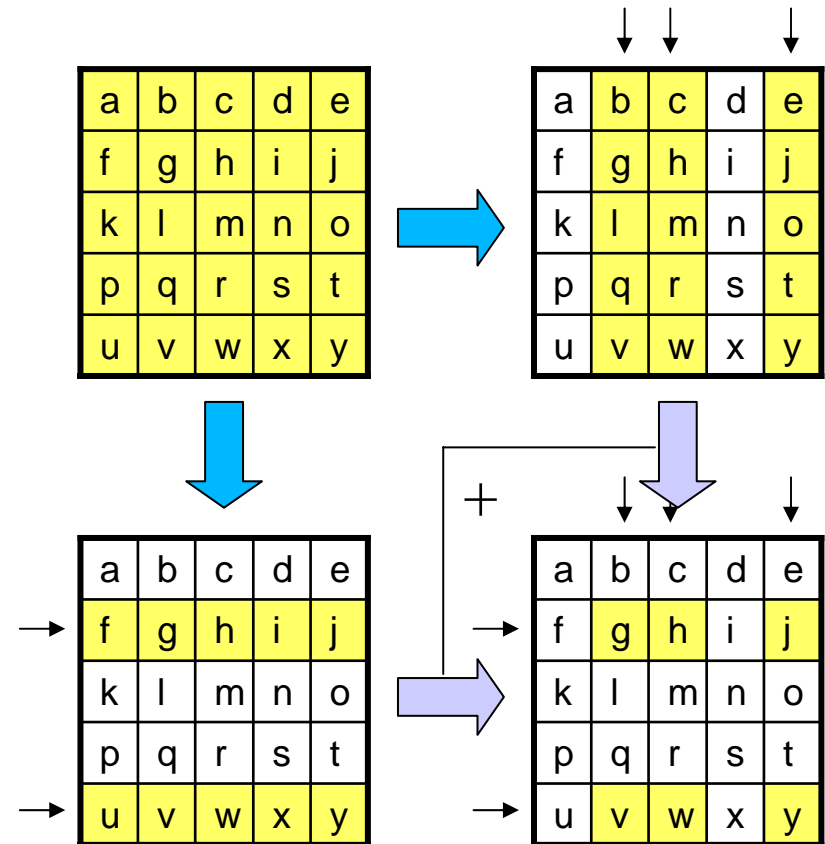
- Mächtigkeit
 - Kombination verschiedener Schritte
- Beispiel
 - Lade Daten
 - Attributgewichtung
 - Gewichte Attribute durch genetisches Verfahren
 - Validiere/bewerte Ergebnisse
 - Protokolliere Gewichtung



YALE: Views



- Verschiedene Sichten auf Daten
 - Intern: DataTable-Klasse
- Vermeidet kopieren von Daten (Bsp: Crossvalidation)



YALE: Plugins



- Verfügbare Plugins
 - Clustering
 - WordVectorTool (NLP)
 - SeriesData (Audio-Daten)
 - DataStream
 - DistributedDataMinig
 - Optimieren von Parametern (Netzwerkkosten etc. für verteilte Anwendung)
- Einfügen von Erweiterungen über Wrapper
 - Bsp: mySVM (C basiert)

Anwendung: YALE im Java-Projekt



YALE
Rapid Development...

Anwendung: Beispiel



- ML basierte Anwendung entwickeln
 - Gegeben
 - Trainingsdaten zum Trainieren eines Lernalgorithmus
 - Testdaten zur Evaluation
- Ziele
 - Einfache, schnelle Entwicklung (YALE-GUI)
 - Leichte Erweiterbarkeit
 - Kein Neukompilieren, wenn sich ML-Teil ändert
 - Trennung von Applikation und ML-Teil

Anwendung: Beispiel : Ansatz

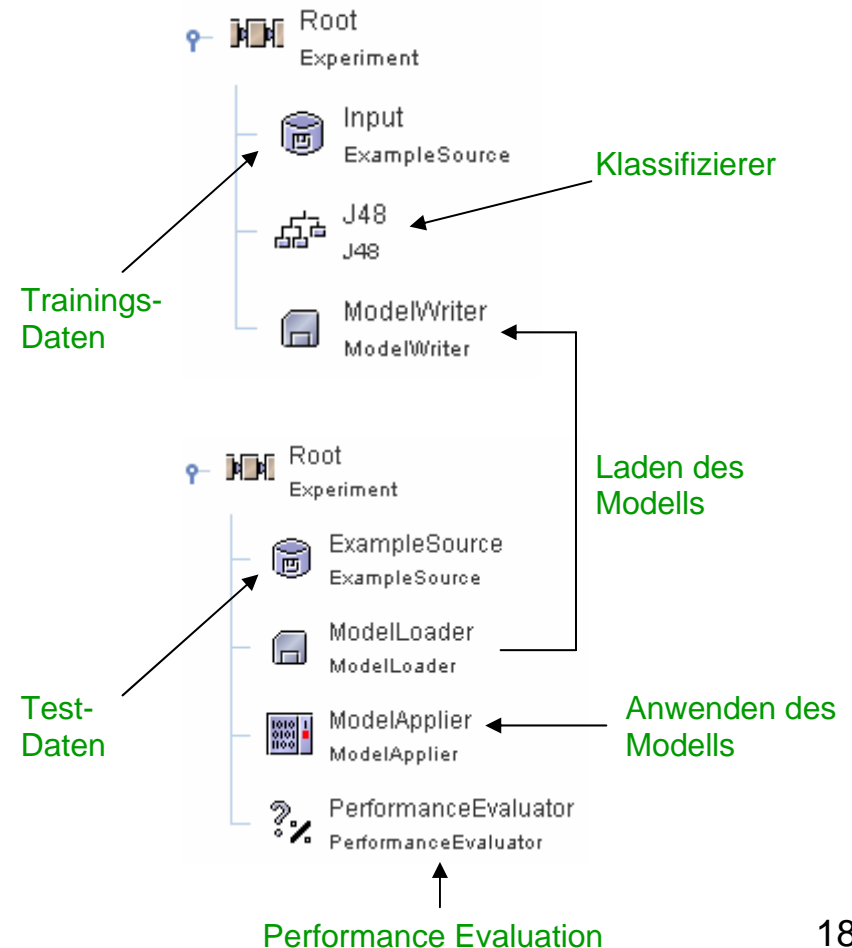


- YALE-Anteil
 - Modell trainieren
 - Klassifikationsexperiment
 - Möglich: Übergabe von Daten zur Klassifikation direkt in Applikation
- Java-Anteil
 - a) Einbindung des trainierten Modells in Java
 - Statisches Klassifizierungs-Modell
 - b) Einbindung des Operatorbaums
 - Anpassung des Modells möglich
 - Steuerung der Modellgenerierung durch Applikation

Anwendung: Beispiel : Experimente



- Trainieren eines Modells
- Verwenden des Modells zur Klassifizierung



Anwendung: Beispiel : In Java



```
// Initialization of YALE
System.setProperty("yale.home",
    "C:\\Programme\\YALE\\yale-3.4");
Yale.init();
```

```
// create Experiment
File expFile =
    new File("C:/afa/yale-test/MyExperiment.xml");
Experiment e = new Experiment(expFile);
```

```
// start Experiment
IOContainer res = e.run();
```

```
// Load Classification Experiment
expFile = new File(
    "C:/afa/yale-test/MyClassification.xml");
e = new Experiment(expFile);
```

```
// start Experiment
IOContainer result = e.run();
```

- Yale initialisieren
- Laden des Trainings-Experiments
- Verwenden des Klassifizierungs-Experiments

Zusammenfassung



- YALE integriert WEKA
- Mächtigkeit durch Zusammenschalten von Operatoren in Operatorbäume
- Einfache Integration in Java-Applikationen
- „Rapid Development“:
Anpassen der Experimente im YALE-GUI ohne die Java-Applikation ändern zu müssen

Bibliographie



- [1]: O. Ritthoff, R. Klinkenberg, S. Fischer, I. Mierswa und S. Felske: „YALE: Yet Another Machine Learning Environment“, In Klinkenberg, Ralf und Rüping, Stefan und Fick, Andreas und Henze, Nicola und Herzog, Christian und Molitor, Ralf und Schröder, Olaf (Editoren), LLWA 01 - Tagungsband der GI-Workshop-Woche Lernen - Lehren - Wissen - Adaptivität, Nr. 763, Seiten 84-92, Dortmund, Deutschland, 2001.
- [2]: YALE Homepage:
<http://rapid-i.com/>
Stand: Dez. 2006
- [3]: YALE Tutorial:
http://rapid-i.com/component/option,com_weblinks/catid,24/Itemid,60
Stand: Dez. 2006
- [4]: WEKA Homepage:
<http://www.cs.waikato.ac.nz/ml/weka/>
Stand: Dez. 2006

Demo



- YALE-Gui (+XML Dateien)
 - Trainings-Experiment
 - Klassifikations-Experiment
- Eclipse / Java
 - Ytest
 - Extraktion der Klassifizierten Daten