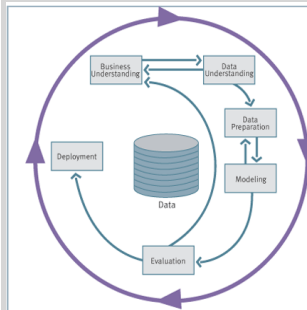


Data Mining mit RapidMiner



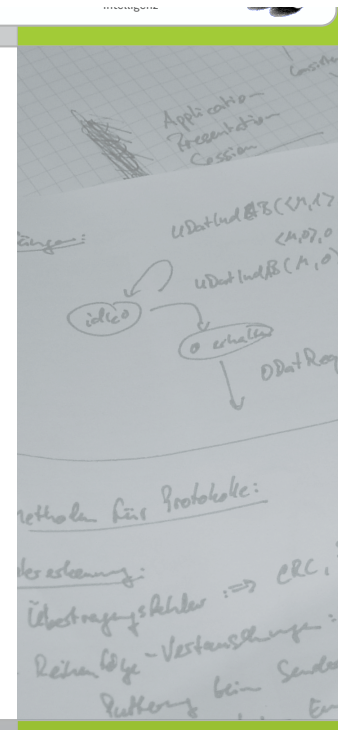
Anforderungen

- Einfache wiederverwendbare Spezifikation von DM-Prozessen
- Austauschbarkeit von Lern-Verfahren, insbesondere:
 - Durchführen von Verfahrensvergleichen
- Kombination/Verschachtelung von Verfahren
- Verfahren zur Merkmalsauswahl und -generierung

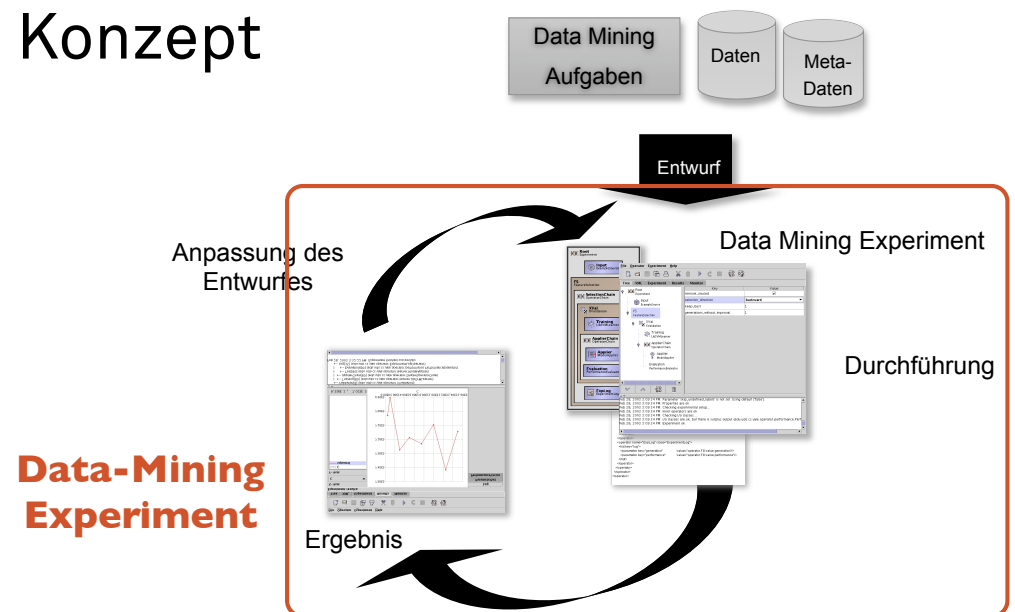


Motivation

- CRISP: DM-Prozess besteht aus unterschiedlichen Teilaufgaben
- Datenvorverarbeitung spielt wichtige Rolle im DM-Prozess
- Systematische Evaluationen erfordern flexible und strukturierte Experimentierumgebung
- Ggf. periodische Wiederholungen von Analysen notwendig



Konzept



Data-Mining Experiment

Data Mining Aufgaben

- Klassifikation/Regression
- Transduktion
- Merkmalsextraktion
- Merkmalsgenerierung
- Merkmalsselektion
- Concept Drift
- Zeitreihenanalyse
- Text-Mining

RapidMiner

- Modellierung von DM-Prozessen als Abfolge von Operatoren (Ketten)
- Verschachtelung von Operatoren
- Transparente/effiziente Datenhaltung
- Leichte Erweiterbarkeit
- GUI-Modus/Batch-Modus
- Einbindung externer Programme (z.B. Weka, SVM-Implementierungen)

Integrierte Operatoren

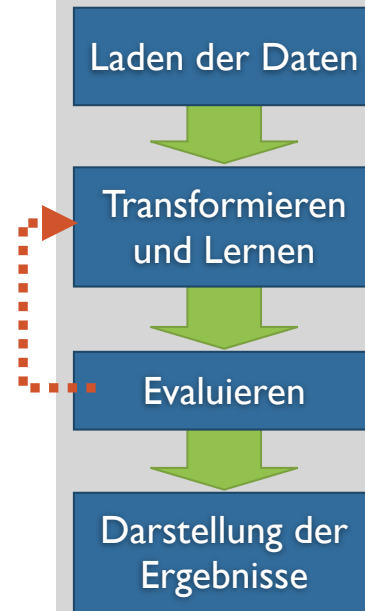
- Operatoren zur Ein-/Ausgabe
- Datenvorverarbeitung
- Zahlreiche Lernverfahren (Weka-Lerner, Clustering, ...)
- Performanzbewertung von Lernverfahren
- Verwaltung/Ausgabe von Lernergebnissen

Information

- Open-Source (GPL-Lizenz)
- Erfolgreiche Anwendung auf unterschiedliche Lernaufgaben
- Weltweite Verbreitung (Anwender und Forscher in über 30 Ländern)
- Dokumentation/Download/uvn unter <http://rapid-i.com>

DM-Experiment

- Laden der Daten
 - Datenbank, Datei
- Transformieren und Lernen
 - Fehlende Werte? Normierung? Klassifikation? Clustering?
- Optimierung:
 - Verfahrensauswahl, Parameter
- Ausgabe der Ergebnisse
 - Performanz, Regeln, Cluster

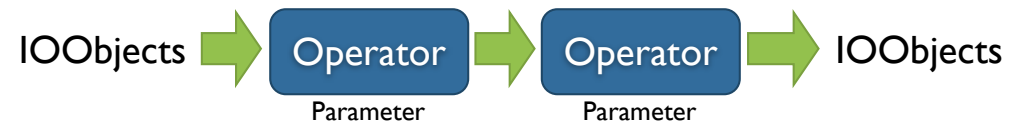


Operator/OperatorChain

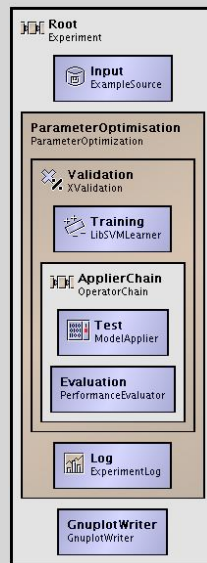
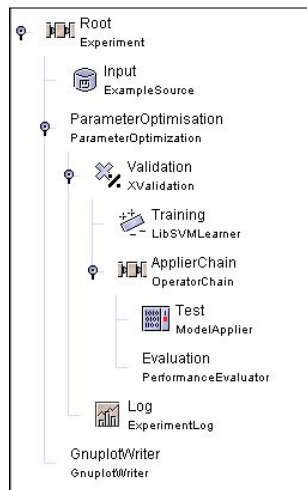
- Operator: Typ, Name, Eingabe, Ausgabe, Parameter



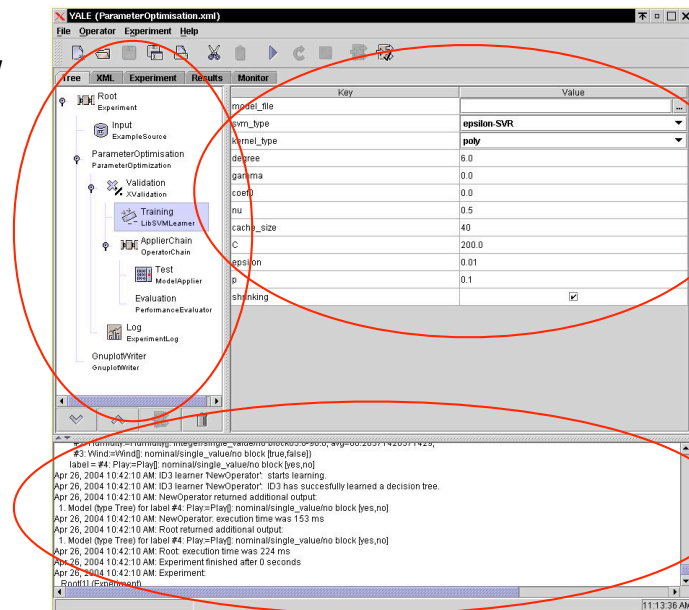
- OperatorChain: Verkettung von Operatoren



Beispiel: Operatorbaum



Aufbau/ Ablauf



Parameter

Logfenster

IObject

- Objekte, die zwischen Operatoren ausgetauscht werden
- Beispiele:
 - ExampleSet (eine Menge von Daten)
 - Model (gelerntes Model)
 - PerformanceVector (Menge von Leistungsmaßen)
 - Ähnlichkeit
 - Merkmalsgewichte
 - ...

ExampleSet (IObject)

- Beschreibung der Attribute (Metadaten):
 - Name
 - Skala: nominal, integer, real, ...
 - Einheit
 - Typ: Einzelwert, Zeitreihe, ...
 - Position (Spalte) in der Datendatei
- Sicht auf Daten

ExampleSet (IObject)

- Spezielle Attribute:
 - Label
 - Predicted label
 - Id
 - Cluster
- Beliebig erweiterbar...

ExampleSet (IObject)

File Table				
golf.data (1)	golf.data (2)	golf.data (3)	golf.data (4)	golf.data (5)
Outlook	Temperature	Humidity	Wind	Play
attribute ▾	attribute ▾	attribute ▾	attribute ▾	label ▾
[unit]	[unit]	[unit]	[unit]	[unit]
nominal ▾	integer ▾	integer ▾	nominal ▾	nominal ▾
single... ▾	single... ▾	single... ▾	single... ▾	single... ▾
sunny	85.0	85.0	false	no
sunny	80.0	90.0	true	no
overcast	83.0	78.0	false	yes
rain	70.0	96.0	false	yes
rain	68.0	80.0	false	yes
rain	65.0	70.0	true	no
overcast	64.0	65.0	true	yes
sunny	72.0	95.0	false	no
sunny	69.0	70.0	false	yes

ExampleSource (Operator)

- Input: -
- Output: ExampleSet
- Parameter: Attributdatei, Datendatei, Sampling, ...

The screenshot shows the WEKA GUI with the 'Monitor' tab selected. On the left, a tree view shows the hierarchy: Root (Experiment) -> Input (ExampleSource) -> NewOperator (DecisionTreeLearner). The 'Monitor' table on the right displays the configuration for the 'ExampleSource' operator.

Key	Value
attributes	data/golf.xml <input type="button" value="Edit"/> ...
sample_size	-1
datamanagement	double_array
separator_chars	::
ignore_chars	
comment_chars	#

At the bottom of the window, a status bar reads: "Apr 26, 2004 11:15:45 AM: ParameterOptimisation returned additional output:"

IOObjects/Resultate

The screenshot shows the WEKA GUI with the 'Monitor' tab selected. The 'Experiment results' area displays a decision tree for the 'golf' dataset. The tree structure is as follows:

```
graph TD
    Root[Wind] -- true --> Outlook1[Outlook]
    Root -- false --> Temp1[Temperature]
    Outlook1 -- rain --> No1[no]
    Outlook1 -- overcast --> Yes1[yes]
    Outlook1 -- sunny --> Temp2[Temperature]
    Temp2 -- <=75.0 --> Yes2[yes]
    Temp2 -- >75.0 --> No2[no]
    Temp1 -- <=83.0 --> Yes3[yes]
    Temp1 -- >83.0 --> No3[no]
```

The bottom of the window shows a log of events:

```
Apr 26, 2004 10:42:10 AM: Root returned additional output:
1. Model type Tree for label #4. Play=Play: nominal/single_value/no block [yes,no]
Apr 26, 2004 10:42:10 AM: Root: execution time was 224 ms
Apr 26, 2004 10:42:10 AM: Experiment finished after 0 seconds
Apr 26, 2004 10:42:10 AM: Experiment:
Root[1] (Experiment)
+- Input[1] (ExampleSource)
+- NewOperator[1] (DecisionTreeLearner)
```

The status bar at the bottom right shows the time: "11:12:20 AM".