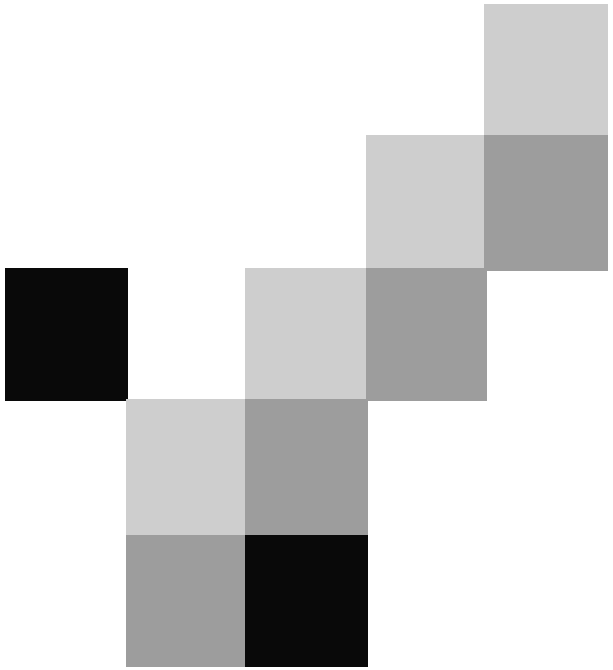


Seminar: Business Intelligence

–

Teil II: Data Mining und Discovery

Sprachen und Systemarchitekturen

Jörg Ramser

23.01.2004

Übersicht

- Motivation
- Data-Mining-Primitive
- Data-Mining-Anfragesprachen
  - DMQL
  - OLE DB for DM
- Architekturen von Data-Mining-Systemen
  - Klassifizierung
  - Beispiel
- Zusammenfassung

Motivation Data-Mining-Sprachen

- Das Finden *aller* in den Daten enthaltenen Muster macht *keinen* Sinn
  - Zahl der Muster steigt exponentiell mit Datenbankgröße
    - Performanzprobleme
    - die meisten Muster für Anwender uninteressant

- Anwender hat aber eine Vorstellung von dem was für ihn interessant ist
  - ⇨ Data-Mining als interaktiver Vorgang

⇨ Sprachen zur Kommunikation mit dem Data-Mining-System

Data-Mining-Primitive

- Data-Mining-Anfragen bestehen aus 5 Primitiven
  - Auswahl relevanter Daten
  - Auswahl der zu erkennenden Wissensart
  - Definition von Hintergrundwissen
  - Definition von Relevanzmaßen
  - Präsentation/Visualisierung der Ergebnisse

Auswahl der relevanten Daten

- nur Teil der DB für Anwender von Interesse
  - Auswahl der relevanten Tupel
  - Auswahl der relevanten Attribute/Dimensionen
- Anwender hat allerdings nur grobe Vorstellung
  - Daten mit starker semantischer Beziehung zu relevanten Daten werden häufig übersehen
  - ⇨ Unterstützung bei der Auswahl durch System
- Relationale DB: relationale Anfrage
  - Selektion, Projektion, Join, Aggregation…
  - ⇨ Minable Views

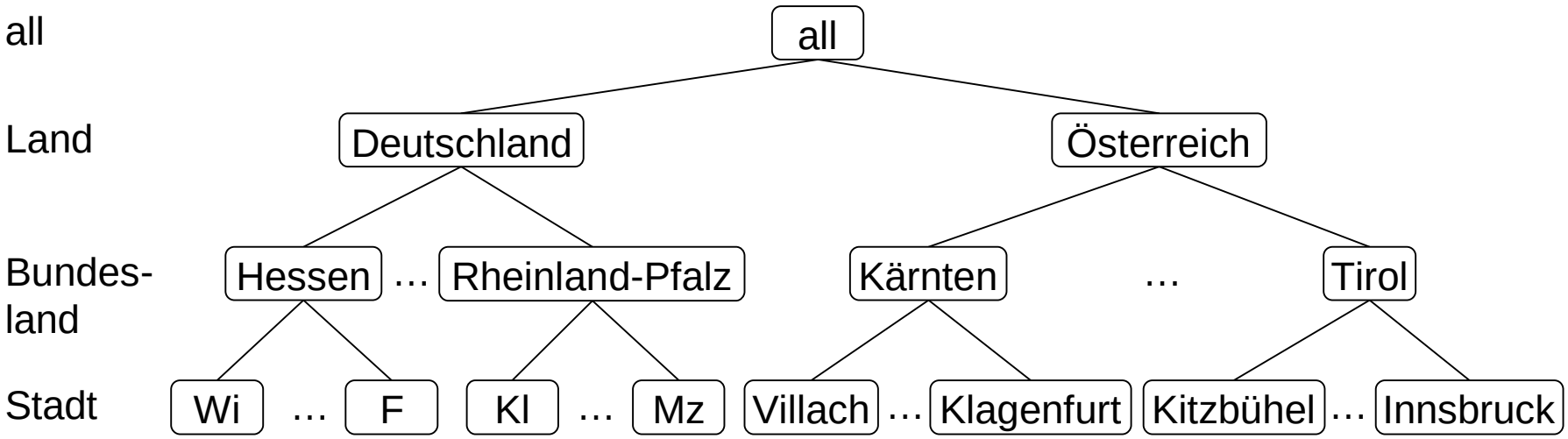
Auswahl der zu erkennenden Wissensart

Ziel: weitere Reduktion der uninteressanten Muster

- Arten
  - Begriffsbeschreibungen
    - Datencharakterisierung
    - Datendifferierung
  - Assoziationsregeln
  - Klassifikation
  - Vorhersage
  - Clusterbildung
  - Zeitliche Entwicklungsanalyse
- Zusätzlich Metamuster

Hintergrundwissen

- *Allgemein:* Wissen über die Anwendungsdomäne
  - Unterstützt das DM-System bei der Suche nach Mustern
  - Hilft bei der Bewertung der gefundenen Muster
- *Hier:* Begriffshierarchien
  - Entdecken von Wissen auf verschiedenen Abstraktionsebenen
  - erleichtern das Verständnis



Relevanzmaße für Ergebnisse

Ziel: weitere Reduktion der uninteressanten Muster

- Typen
  - Nützlichkeit (support)
  - Zuverlässigkeit (confidence)
  - Einfachheit
    - Mustergröße
    - Regellänge
  - Neuheit
- Schwellenwerte
  - Muster, die Schwellenwerte nicht erreichen werden nicht angezeigt

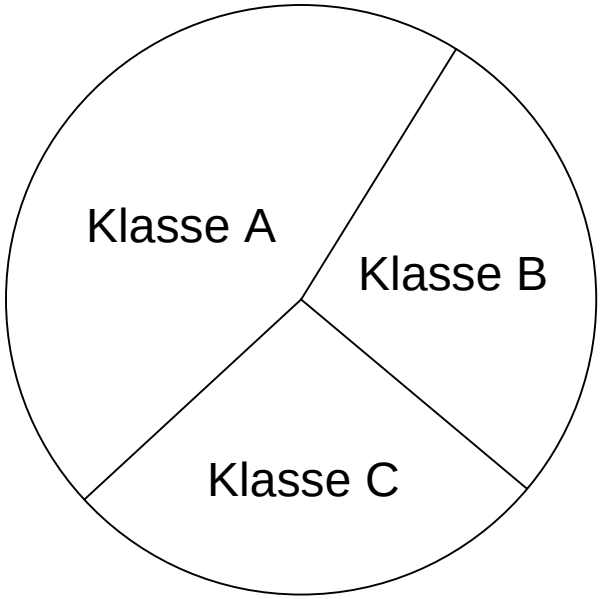
Präsentation/Visualisierung der Ergebnisse (I)

- verschiedene Darstellungsmöglichkeiten für verschiedene Anwendergruppen
  - Regeln
  - (Kreuz-) Tabellen
  - Kuchen- und Balkendiagramme
  - Entscheidungsbäume
  - Würfel

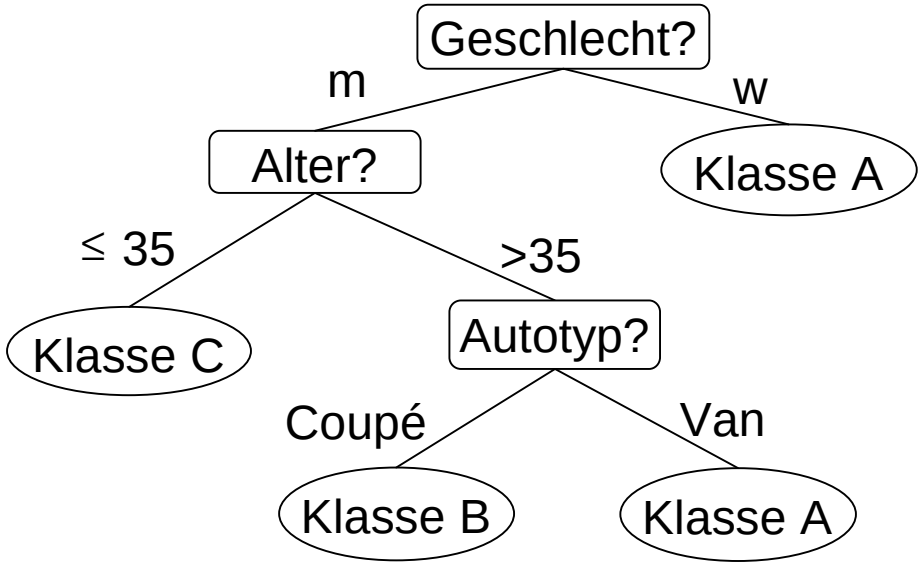
- Unterstützung von OLAP-Operationen
  - Drill-Down, Roll-Up
  - Hinzufügen/Entfernen von Attributen und Dimensionen

  $\Rightarrow$ Betrachtung des Wissens auf verschiedenen Abstraktionsebenen

Präsentation/Visualisierung der Ergebnisse (II)



Kuchendiagramm



Entscheidungsbaum

Geschlecht(X,“w“) => Klasse(X,“A“)
Geschlecht(X,“m“) ∧ Alter(X,≤“35“) => Klasse(X,“C“)
Geschlecht(X,“m“) ∧ Alter(X,“>35“) ∧ Autotyp(X,“Coupé“) => Klasse(X,“B“)
Geschlecht(X,“m“) ∧ Alter(X,“>35“) ∧ Autotyp(X,“Van“) => Klasse(X,“A“)

Regeln

Übersicht

- Data-Mining-Primitive
- Data-Mining-Anfragesprachen
  - DMQL
  - OLE DB for DM
- Architekturen von Data-Mining-Systemen
  - Klassifizierung
  - Beispiel
- Zusammenfassung

DMQL

- vorgeschlagen von Han, Fu… (1996)
- Einsatz in DBMiner
  - kein Zugriff auf DMQL
- SQL-ähnliche Syntax
- basiert direkt auf den 5 Primitiven

Auswahl relevanter Daten

- praktisch identische Syntax mit SQL

```
use database Uni_DB
in relevance to S.Name, S.Geschlecht, S.Hauptfach, S.Geburtsort,
        S.Geburtsdatum, S.Wohnort, S.Notendurchschnitt, S.Semester
from Studenten S
where S.status <> ’diplomiert’ and S.semester > 13
```

Auswahl der zu erkennenden Wissensart

- **Datencharakterisierung**

  mine characteristics **as** Langzeitstudenten
  analyze count%

- **Datendifferenzierung**

  mine comparison
  for Langzeitstudenten **where** S.status <> ’diplomiert’ **and** S.semester > 13
  versus Nicht_Langzeitstudenten **where**
  S.status = ’diplomiert’ **or** S.semester <= 13
  analyze count

- **Klassifikation**

  mine classification **as** Risikoklassifizierung
  analyze Versicherungsrisiko

Modellierung von Begriffshierarchien

- Definition von Begriffshierarchien
  - **define hierarchy** hierarchie_ort **on** ort **as**
    [stadt, bundesland, land]
- Einbindung von Begriffshierarchien
  - **uses hierarchy** hierarchie_ort **for** ort

Definition von Relevanzmaßen

- with-Statement erlaubt Setzen von Relevanzmaßen und Schwellenwerten

- Beispiele

  - **with** support **threshold** = 0,05
  - **with** confidence **threshold** = 0,70

Ergebnispräsentation/-visualisierung

- Auswahl der Darstellungsform
  - **display as** decisiontree
- dynamisches Betrachten der Ergebnisse auf verschiedenen Abstraktionsebenen
  - **roll up on** Ort
  - **drill down on** Ort
  - **add** Stadt
  - **drop** Stadt

Übersicht

- Motivation
- Data-Mining-Primitive
- Data-Mining-Anfragesprachen
  - DMQL
  - OLE DB for DM
- Architekturen von Data-Mining-Systemen
  - Klassifizierung
  - Beispiel
- Zusammenfassung

OLE DB for Data Mining

- **Ziele:**
  - Einbinden von Data-Mining-Algorithmen verschiedener Hersteller (provider) in verschiedene Anwendungsprogramme (consumer)
  - Schaffung eines Industriestandards
- vorgeschlagen von Microsoft (2000)
- unterstützt im Microsoft SQL Server 2000
- SQL-ähnliche Syntax
- zentrales Objekt: Data-Mining-Model (DMM)
  - enthält Regeln, Klassifikationen, Formeln zur Repräsentation des Wissens
  - unterstützt Vorhersage fehlender Werte
  - spezieller Tabellentyp (geschachtelte Tabellen)

Spezifikation von Data-Mining-Modellen

```
CREATE MINING MODEL [Studiengang_Vorhersage]
(
    [Matrikelnummer]              LONG      KEY,
    [Geschlecht]                  TEXT      DISCRETE,
    [Semester]                    LONG      DISCRETE
    [Studiengang]                 TEXT      PREDICT,
    [Studiengang Probability]     DOUBLE    PROBABILITY of [Studiengang]
    [Teilnahme]                   TABLE
    (
        [Veranstaltung]           TEXT      KEY,
        [Note]                    DOUBLE    CONTINOUS,
        [VeranstaltungsTyp]       TEXT      DISCRETE
    )
)
USING [Decision Trees]
```

Bereitstellung von Trainingsdaten

```
INSERT INTO [Studiengang_Vorhersage]
(   [Matrikelnummer], [Geschlecht], [Studiengang],
    [Studiengang Probability], [VeranstaltungsTeilnahme])

SHAPE
{   SELECT [Matrikelnummer], [Geschlecht], [Studiengang]
    FROM [Studenten]}

    APPEND
    ({      SELECT [Matr_Nr], [Veranstaltung], [Note]
            FROM [Teilnahme]}
            RELATE [Matrikelnummer] TO [Matr_Nr]})
    AS [VeranstaltungsTeilnahme]
```

Durchsuchen des Data-Mining-Model

- DMM enthält Regeln, Formeln, Klassifikationen, Verteilungen, Knoten …
  - Ergebnis des DM-Algorithmus
- **SELECT** * **FROM** [Studiengang_Vorhersage].CONTENT
  - liefert den Inhalt in Form eines Baumes zurück
- Regeln als XML-Strings in den einzelnen Knoten gespeichert
  - XML-String ermöglicht die Wissensdarstellung auf verschiedene Arten (5. Primitive)

Anfragen zur Vorhersage

- auf Basis eines trainierten DMM
- Anfrage mittels SELECT-FROM-ON-WHERE
  - SELECT
    - zur Auswahl der anzuzeigenden Spalten
  - FROM
    - Quelldaten: Verknüpfung der aktuellen Daten (mit den fehlenden Werten) mit dem DMM mittels **Prediction-Join**
  - ON
    - durch ‘and‘ verknüpfte Join-Bedingungen
  - WHERE
    - zur Angabe von Relevanzmaßen mit Schwellenwerten (4. Primitive)

Anfragen zur Vorhersage (Beispiel)

```
SELECT
    T1.[Matrikelnummer], M1.[Studiengang]

FROM
    [Studiengang_Vorhersage] AS M1 PREDICTION JOIN
    (
        SELECT [Matrikelnummer], [Studiengang],
                [Semester], [Geschlecht]
        FROM [Studenten]
    ) AS T1

ON
    M1.[Geschlecht] = T1.[Geschlecht] AND
    M1.[Semester] = T1.[Semester]

WHERE PredictProbability(M1.Studiengang) > 0.8
```

DMQL vs. OLE DB for DM

	DMQL	OLE DB for DM
Vorteile	▪ unterstützt alle fünf Primitive ▪ orientiert sich am SQL-Standard	▪ unterstützt bis auf die Definition von Hintergrundwissen alle Primitive ▪ Auswahl beliebiger Data-Mining-Algorithmen verschiedener Hersteller ▪ orientiert sich am SQL-Standard
Nachteile	▪ es existiert kein System, das den Zugriff auf und die Nutzung von DMQL zulässt ▪ ermöglicht nicht die Auswahl beliebiger Data-Mining-Algorithmen verschiedener Hersteller	▪ unterstützt nicht die Definition von Hintergrundwissen

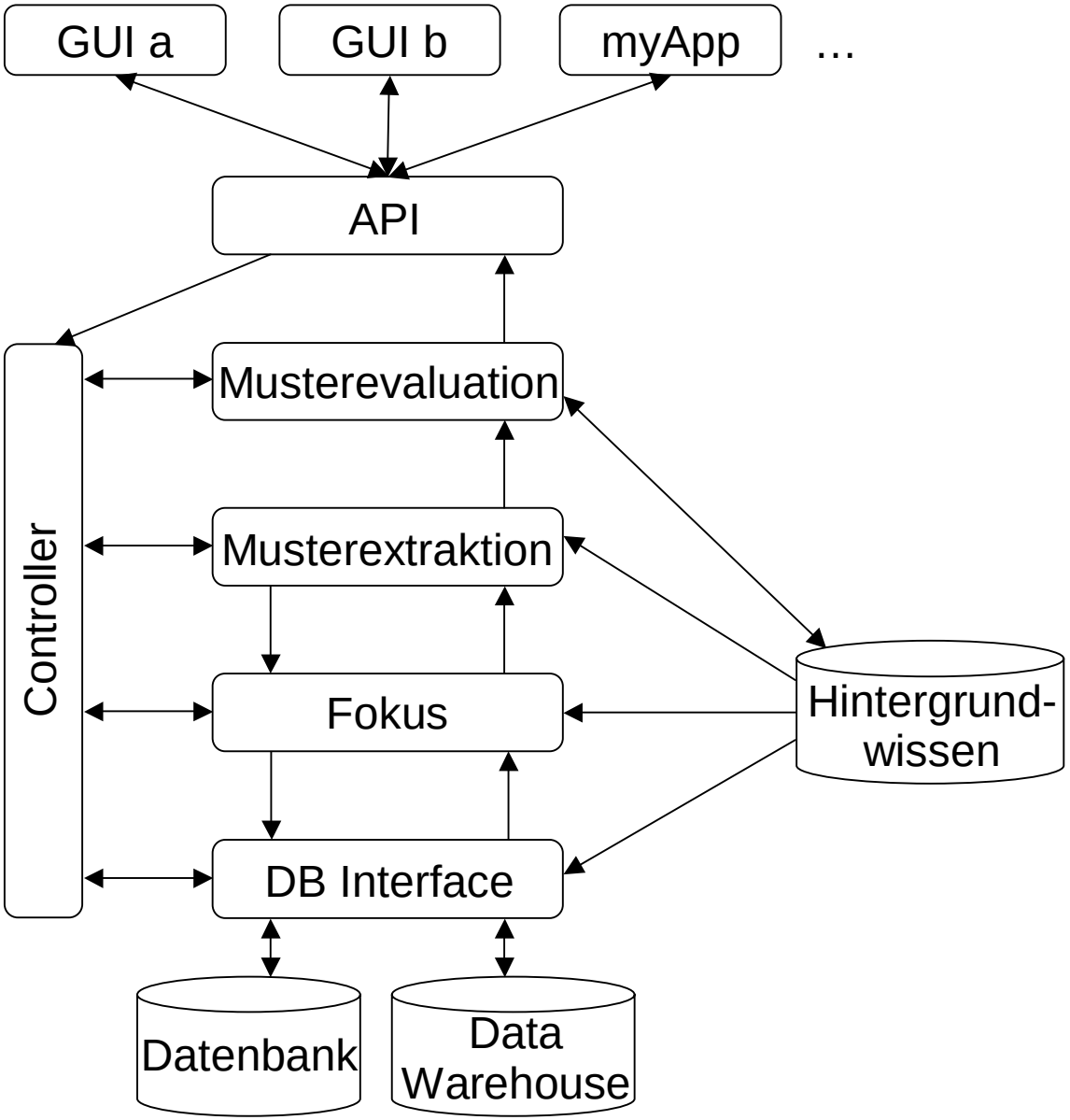
Übersicht

- Motivation
- Data-Mining-Primitive
- Data-Mining-Anfragesprachen
  - DMQL
  - OLE DB for DM
- Architekturen von Data-Mining-Systemen
  - Klassifizierung
  - Beispiel
- Zusammenfassung

Klassifizierung von DM-Architekturen

- Klassifizierung der DM-Architekturen anhand des Integrationsgrades in DB/DW-Systeme
  - **Keine Kopplung (no coupling)**
    - keinerlei Integration
  - **Lose Kopplung (loose coupling)**
    - Laden/Speichern der Daten aus/in DB/DW-System
    - Verarbeiten der Daten vollkommen unabhängig von DB/DW
  - **Mittelstarke Kopplung (semitight coupling)**
    - DM-System nutzt wichtige Funktionalitäten der DB/DW-Systeme (Indexierung, Mehrwege-Joins, Sortierung, Aggregation…)
  - **Starke Kopplung (tight coupling)**
    - komplett in DB/DW-System integriert
    - DM funktionale Komponente des DB/DW-Systems

Beispiel für starke Kopplung



Zusammenfassung

- Motivation
- Data-Mining-Primitive
- Data-Mining-Anfragesprachen
  - DMQL
  - OLE DB for DM
- Architekturen von Data-Mining-Systemen
  - Klassifizierung
  - Beispiel
- Zusammenfassung