PER - Models

Seminar in machine learning WT 09/10

Agenda

- Motivation for the PER-Model

- What is the PER-Model?

- Entity-Relationship-Model

- Extending ER-Model to PER-Model

- A look on DAPER-Model

- DAPER-Model vs. Plate Model

- DAPER-Model vs. PRM

- Extensions to PER-Model and possible future works

- Conclusion
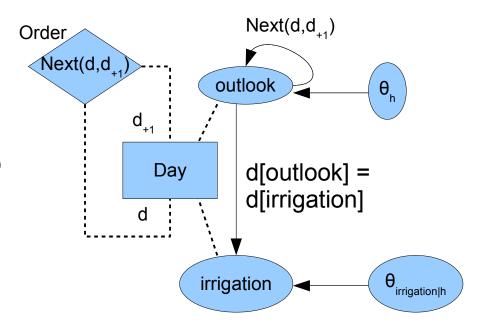
- References

Motivation for the PER-Model

- Datasets grow and grow...

- Todays datasets are not flat anymore
  - e.g. consist relationships or probabilistic dependencies

- Already existing graphical representations get complicated to read and have to be extended to describe data

- Entity Relationship Model is a already well defined with many features
  - But not really designed for data representation
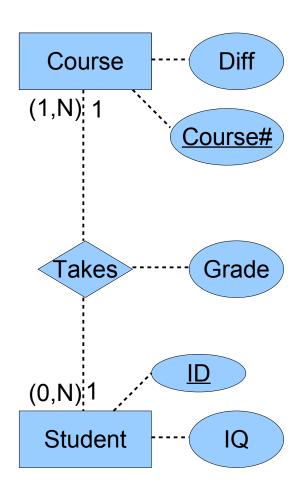
What is the PER-Model?

- Probabilistic Entity Relationship Model (PER-Model)

- A graphical model for describing relational data

- Extends the Entity Relationship Model (ER-Model)

- Adds probabilistic distributions and constraints

Entity Relationship Model

- Designed for graphical diagram the structure of databases through design process

- Shows the structure of a possible database without using an underlying dataset

- Uses the concept of Entities and their relationships

Course

Diff

(1,N) 1

Course#

Takes

Grade

(0,N) 1

ID

Student

IQ

Entity Relationship Model

- graph with 3 sorts of classes:

- entity classes $\qquad E_i \in \mathscr{E}$

- relationship classes $\quad R_i \in \mathscr{R}$

- attribute classes $\qquad A_i \in \mathscr{A}$

Entity Relationship Model
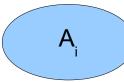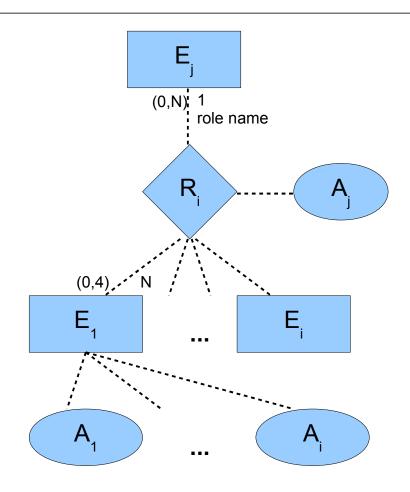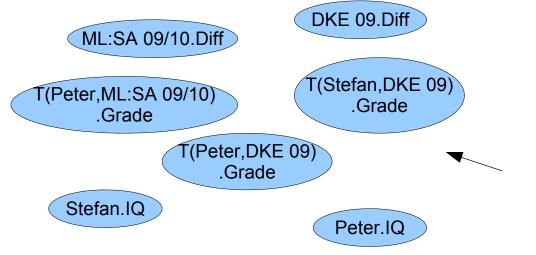
- corresponding classes are connected by dashed edges

- entities participating in a relationship can have a:

  - complexity e.g. (0,4)

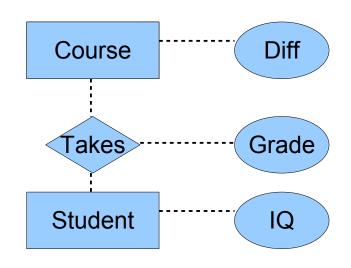 - cardinality e.g. 1

  - role names father

Entity Relationship Model

- Every ER-Model need a skeleton for creating a ground-graph
- Ground graph shows dependencies of attributes

Course — Diff

Takes — Grade

Student — IQ

ML:SA 09/10.Diff

DKE 09.Diff

T(Peter,ML:SA 09/10).Grade

T(Stefan,DKE 09).Grade

T(Peter,DKE 09).Grade

Stefan.IQ

Peter.IQ

Takes	
Student	**Course**
Peter	ML:SA 09/10
Peter	DKE 09
Stefan	DKE 09

Course
ML:SA 09/10
DKE 09

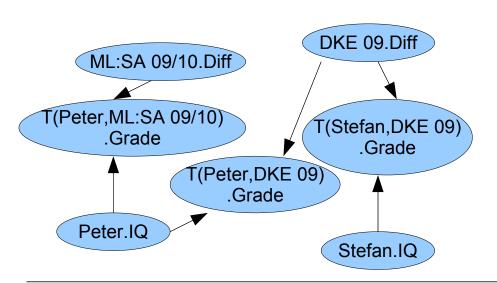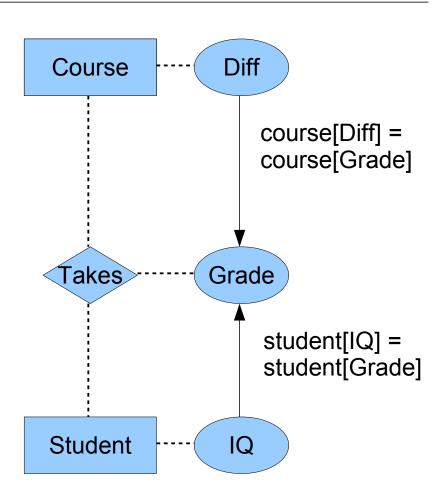| Student |
|---|
| Peter |
| Stefan |

Extending ER-Model to PER-Model

- For representing probabilistic data we need also:
  - Dependencies between attributes
  - Constraints for dependencies
  - A possibility of modeling probabilistic distributions

- This is **not** (or only partially) supported by the ER-Model

- PER-Model add these features and more...

Extending ER-Model to PER-Model

- Depedencies of attribute classes are drawn as edges
- First-order expression could added to dependencies

Course - - - - Diff

course[Diff] = course[Grade]

Takes - - - - Grade

student[IQ] = student[Grade]

Student - - - - IQ

ML:SA 09/10.Diff

DKE 09.Diff

T(Peter,ML:SA 09/10) .Grade

T(Stefan,DKE 09) .Grade

T(Peter,DKE 09) .Grade

Peter.IQ

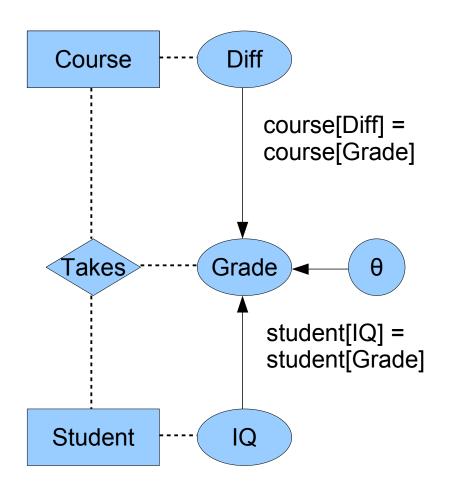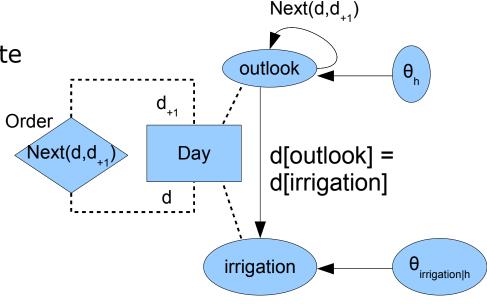Stefan.IQ

Extending ER-Model to PER-Model

- Probabilistic distributions are modeled by an attribute
  - e.g. $p(Takes.Grade) \cong \theta$

- Attribute corresponds to a global entity class
  - Global class is **not** drawn
  - Global class has only **one** entity

Course — Diff

course[Diff] = course[Grade]

Takes — Grade ← θ

student[IQ] = student[Grade]

Student — IQ

Extending ER-Model to PER-Model

- Example of Hidden Markov Models in PER-Model
  - Outlook is hidden attribute
  - Irrigation is non-hidden attribute
  - restriction to relationship class $Next(d,d_{+1})$ to be order on set of days

Next(d,d_{+1})

outlook

θ_h

Order

Next(d,d_{+1})

d_{+1}

Day

d

d[outlook] = d[irrigation]

irrigation

$\theta_{irrigation|h}$
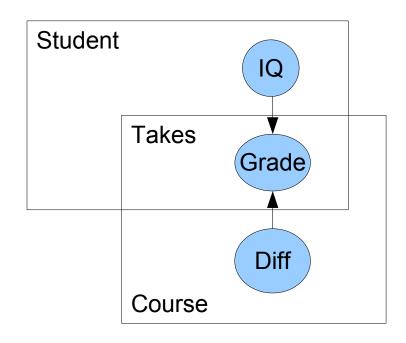
A look on DAPER-Model

- Directed Acyclic PER-Model

- Shown examples all are DAPER-Models

- PER-Model also allows undireted depencies and cycles
  - Cycles ≠ self-relationship
  - Cycle = projected pairwise self-relationship

- Why use DAPER-Models?

- DAPER-Models are easier to understand and nearer to already existing models like PRMs and Plate Models

DAPER-Model vs. Plate Model

- Plate Models do not self-relationships

- Dependencies can not point to an attribute outside the plate

- Depencies are only directed

- dependencies correspond to a single entity
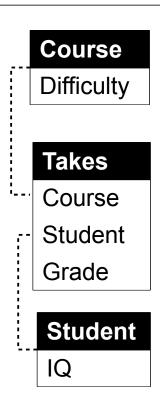
- No constraints

DAPER-Model vs. PRM

- PRMs use tables for both relationship classes and entity classes

- Dependencies are listed seperately

Course
| Difficulty |

Takes
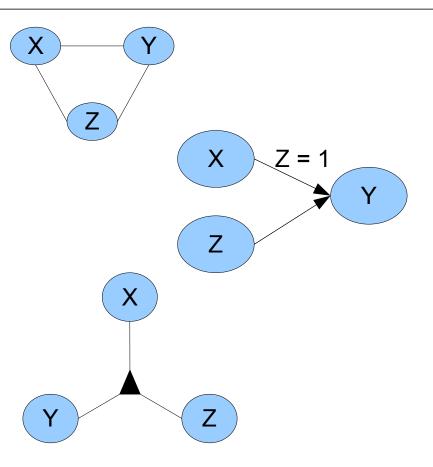| Course |
| Student |
| Grade |

Student
| IQ |

Takes.Course.Difficulty → Takes.Grade
Takes.Student.IQ → Takes.Grade

Extensions to PER-Model and possible future works

- PER-Models can expand to other graphical models than DAGs
  - e.g. contingent DAGs using contingent DAPER-Models
  - e.g. undirected graphical (UG) model with undirected PER-Models
  - e.g. hierachical log-linear graphical (HLLG) Models with hierachical log-linear entity relationship (Helper) models
  - etc.

Conclusion

- PER-Models are an good way to represent probabilistic data structures

- Fast to learn if ER-Model is common
  - ER-Model is widely spread

- Handling of probabilistic distributions maybe is not intuitive as it could be

- PER has wide scope for extensions

References

- For PER-Model:
  - *Introduction to Statistical Relational Learning*, pages 201-238, Lise Getoor and Ben Taskar, 2007
- For ER-Model:
  - Fundamentals of Database Systems, 5th Edition, Ramez Elmasri and Shamkant B. Navathe, 2007
- *Paper of readings in Data Knowledge Engeneering summer term 2009,* Prof. Alejandro Buchmann, Prof. Johannes Fürnkranz and Dr. Ilia Petrov, 2009
  http://www.dvs.tu-darmstadt.de/teaching/dke/2009/

picture credits frontpage: http://blogs.ubc.ca/dean/2008/02/multitasking-an-information-skill-we-teach/