Introduction

Learning Bayesian Networks With Hidden Variables for User Modeling

Barbara Großmann-Hutter, Anthony Jameson, and Frank Wittig
Department of Computer Science, University of Saarbrücken, Germany
http://w5.cs.uni-sb.de/~ready/ (Slides, etc.)

Overview

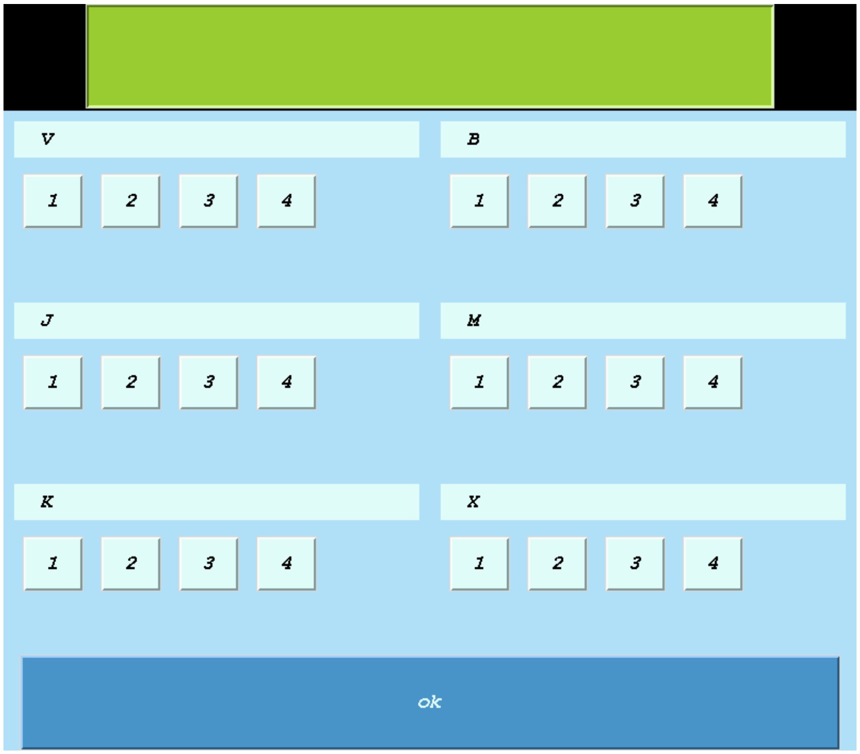
1. Example domain and experiment
2. Modeling the results by learning Bayes nets
   - Proposal 1 and issues
   - Proposal 2 and issues
   - ...

1. Conclusions
2. (Optional:) *Why* learn about users−in−general?

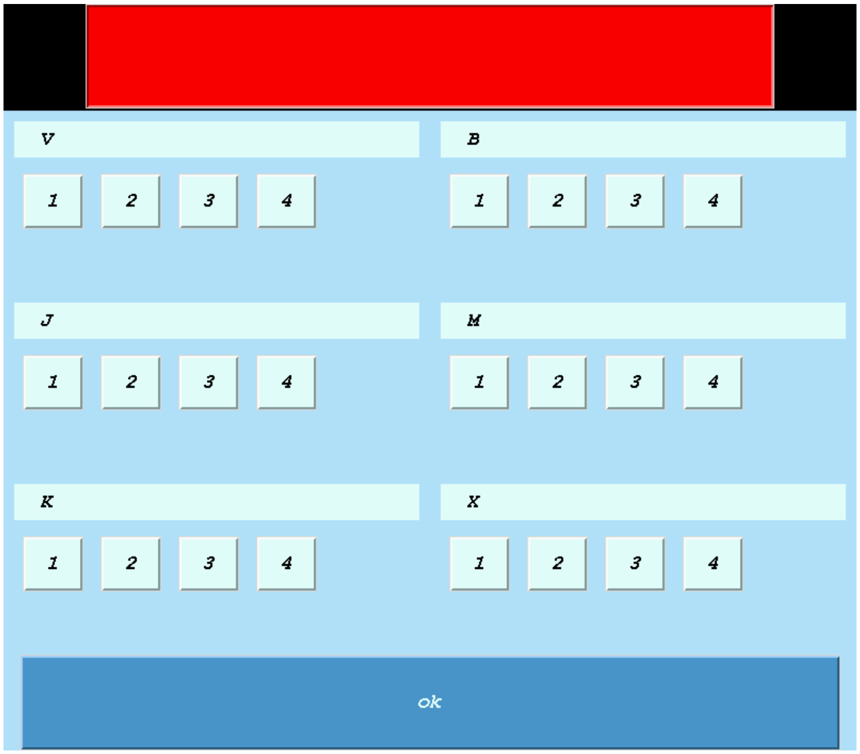
Table of Contents

Experiment: Method

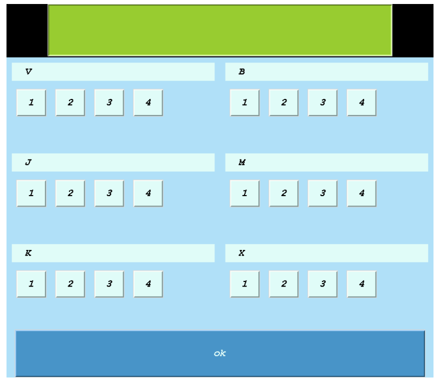
Experimental Setup (1) 3



Experimental Setup (2) 4



Stepwise vs. Bundled Instructions
5



Stepwise:

$\mathcal{S}$: Set *X* to *3*
$\mathcal{B}$: ... OK
$\mathcal{S}$: Set *M* to *1*
$\mathcal{B}$: ... OK
$\mathcal{S}$: Set *V* to *4*
$\mathcal{B}$: ... Done

Bundled:

$\mathcal{S}$: Set *X* to *3*, set *M* to *1*, set *V* to *4*
$\mathcal{B}$: ... ... Done

Variables in Experiment
6

Independent variables

1. Presentation mode
   - *Stepwise* vs. *bundled*
2. Number of steps in task
   - 2, 3 or 4 steps
3. Distraction by secondary task
   - No secondary task vs. monitor the flashing lights

Dependent variables (selection)

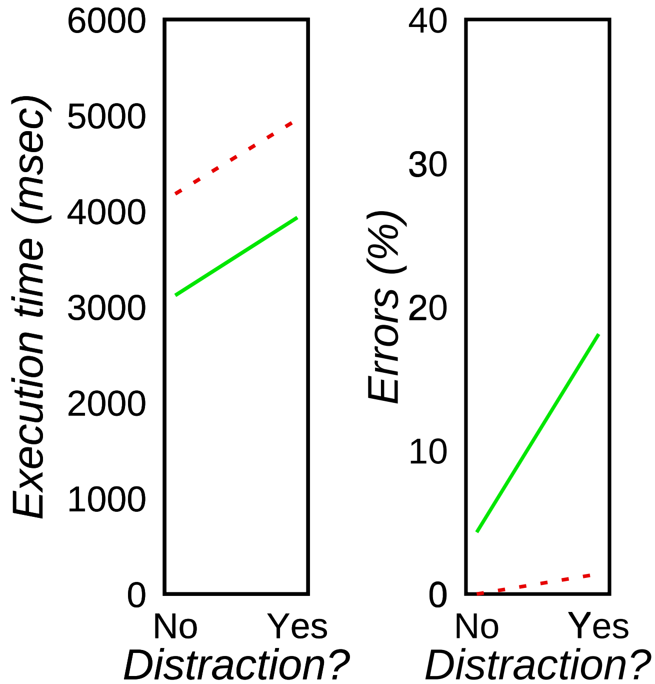
1. Total time to execute an instruction sequence
   Including "OK"s, etc.
2. Error in main task
   - Buttons not pressed, or wrongly pressed

Experiment: Results

Main Results (1)

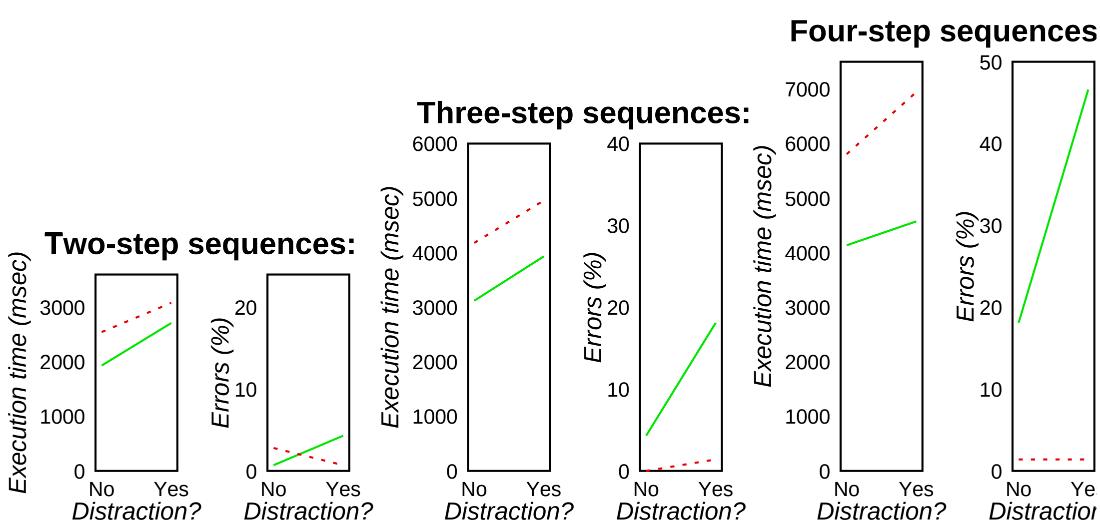
7

Three-step sequences:



Main Results (2)

8



Learning Bayes Nets

1: Modeling Only Observable Variables (1) 9

Definition

- Structure is specified on the basis of theoretical considerations
  This holds for all nets discussed here
- Only observable variables of experiment are included in network

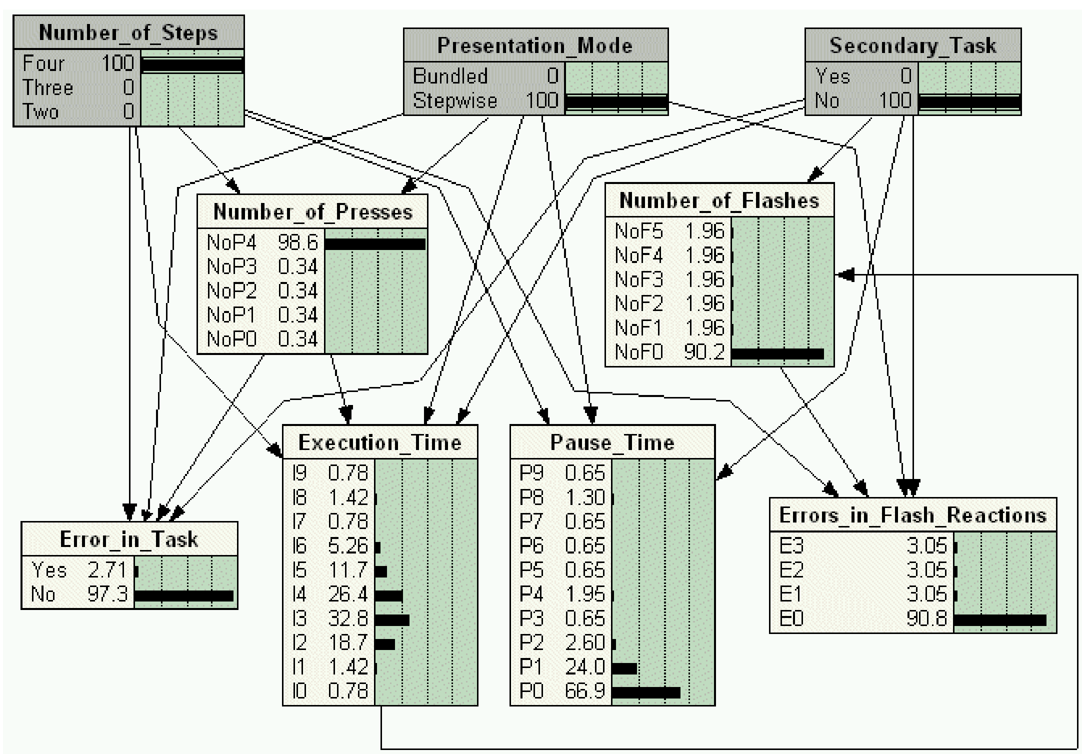
Positive points

- Learning can be done straightforwardly with many BN tools
- Learning is very fast (e.g., < 1 sec)

Negative points

- Little theoretical interpretability
- Relatively inefficient evaluation
  Too many parents per node
- Doesn't take into account systematic individual differences

1: Modeling Only Observable Variables (2) 10



2: Hidden Theoretical Variable (1)

Definition

- Hidden variable "Working Memory Load" added
  Basis:
  - Psychological theory
  - Previous experimental results
- Learning with Russell et al.'s APN algorithm
  ⇒ Gradient descent

Positive points

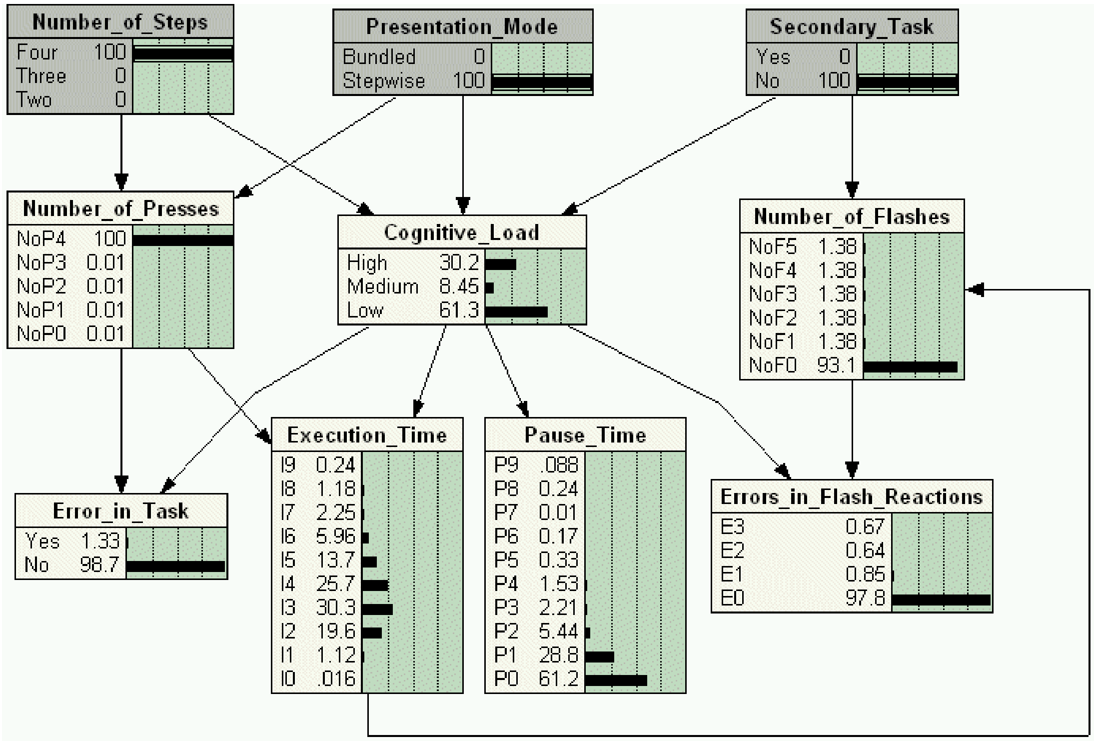
- Better theoretical interpretability
  ⇒ Easier to leverage existing psychological knowledge
  ⇒ Possible to add or replace variables without relearning everything from scratch
- Relatively efficient evaluation

2: Hidden Theoretical Variable (2)

Negative points

- Learning times several orders of magnitude greater (hours or nights)
  Note: Partly due to current limitations of Netica, soon to be removed
- Some aspects of CPTs involving the hidden variable are implausible
  E.g., strangely nonmonotonic relationships
- Individual differences are still not taken into account

2: Hidden Theoretical Variable (3)



3: Modeling Individual Differences (1)

Procedure

Add to each observation in the dataset a new observable feature:

"Overall average execution time of the user in question"

Distinction

- Variables that are naturally observable in an application setting
- Variables that can be made observable in an experimental setting

  How to do this:

  Exploit possibilities for measuring and controlling variables

  Ensure an appropriate number of observations from each subject and/or in each condition

3: Modeling Individual Differences (2)

Positive points

- Accuracy of learned net is greater

  Here: 50% (vs. 44%) accurate prediction of $\mathcal{U}$'s execution time in training set

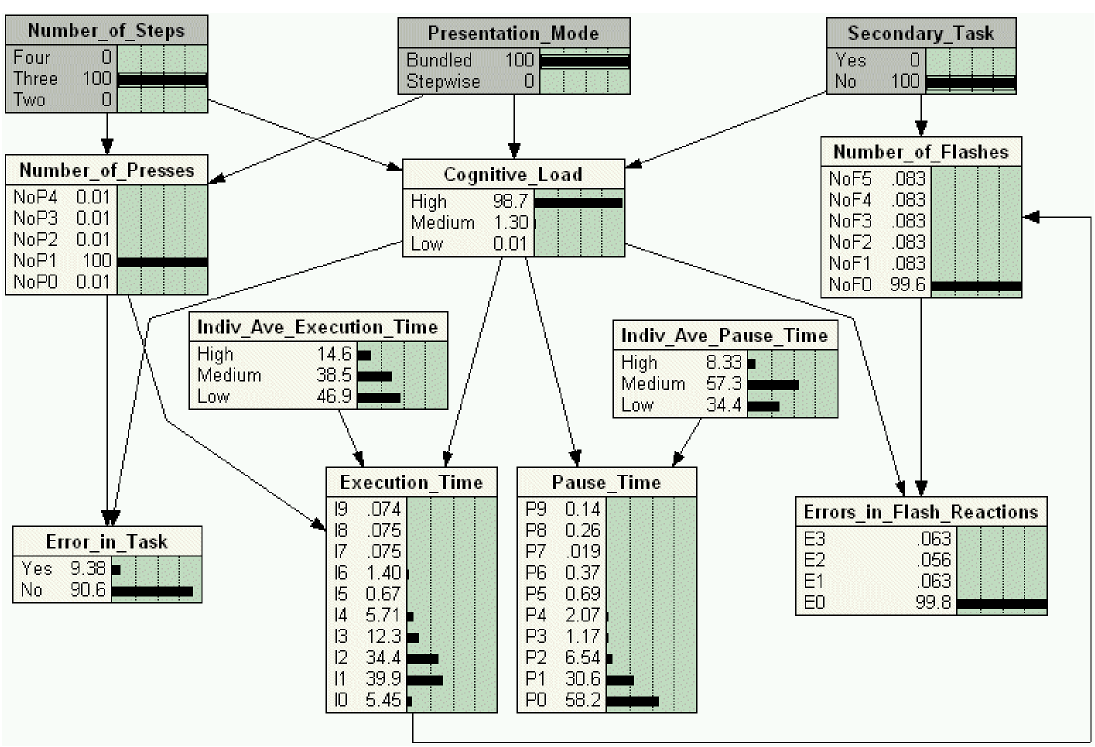
    (Not in itself surprising or significant)

- When the individual-speed variable can be assessed (with uncertainty) in an application situation, prediction accuracy will be improved

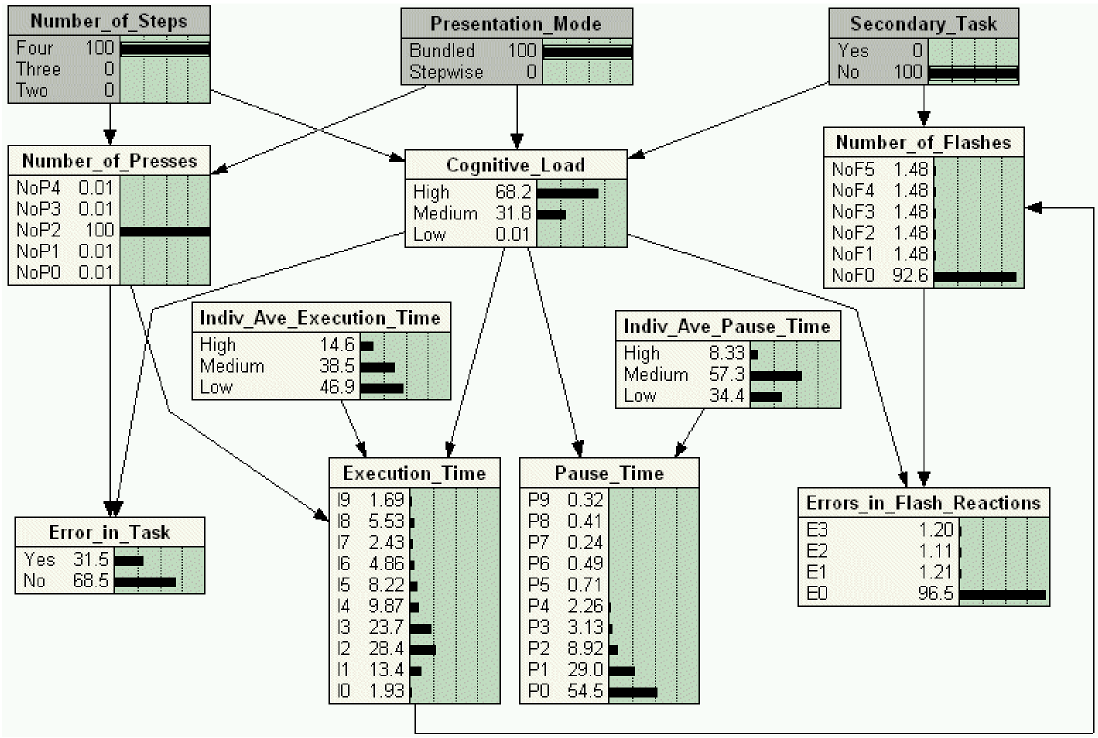
Negative points

- CPTs are still sometimes implausible

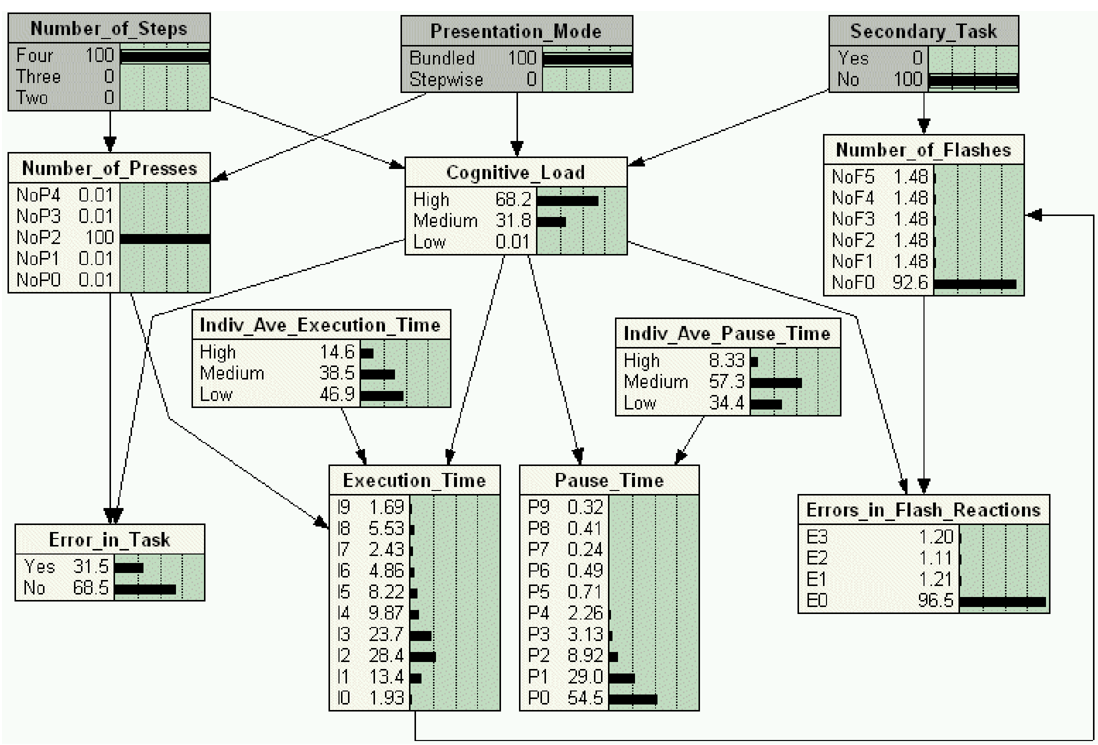
3: Modeling Individual Differences (3)



3: Modeling Individual Differences (4)



3: Modeling Individual Differences (5)



4: Constraining the Nature of Relationships (1) 19

Basic idea

- Formulate theoretically motivated qualitative constraints
  E.g., "More steps ⇒ Higher WM load"
- Ensure that only networks that (almost) satisfy these constraints can be learned

Procedure

1. Translate qualitative formulations of constraints into quantitative inequalities concerning conditional probabilities
   See Druzdzel & van der Gaag (UAI95)
2. Define a corresponding penalty term for nets that violate a constraint
3. Factor in the penalty term when determining the next step in the gradient descent
4. (Strategy tried up to now:)
   Give the penalty term less weight as the search proceeds
   Motivation: Otherwise it might take forever to find a solution

4: Constraining the Nature of Relationships (2) 20

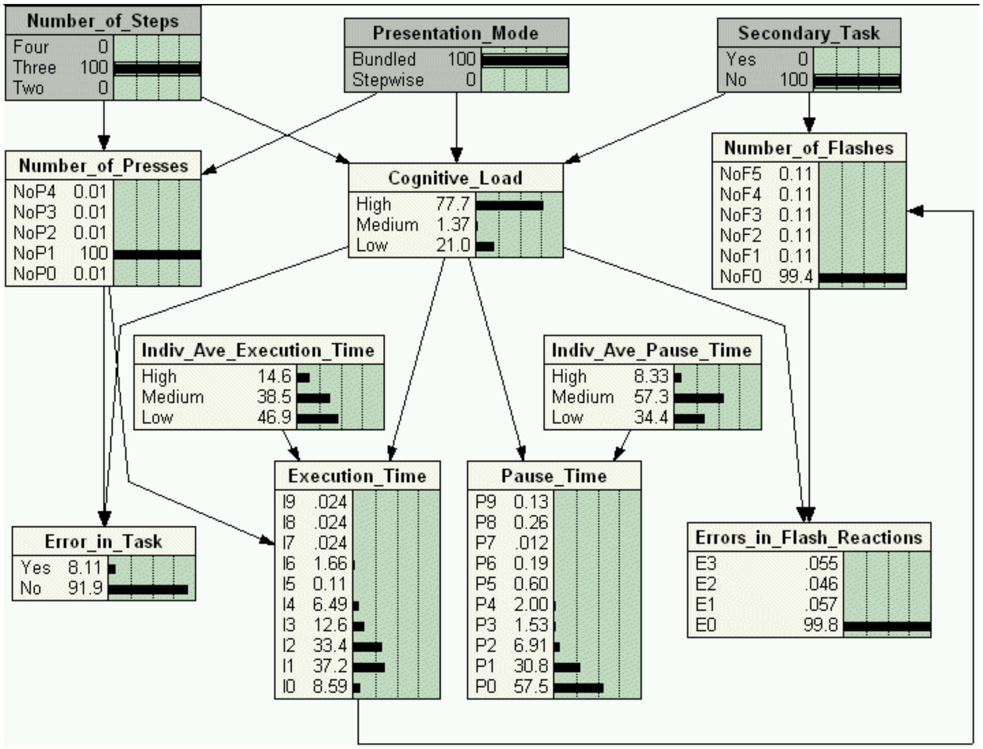
Positive points

- The learned nets do satisfy the constraints better

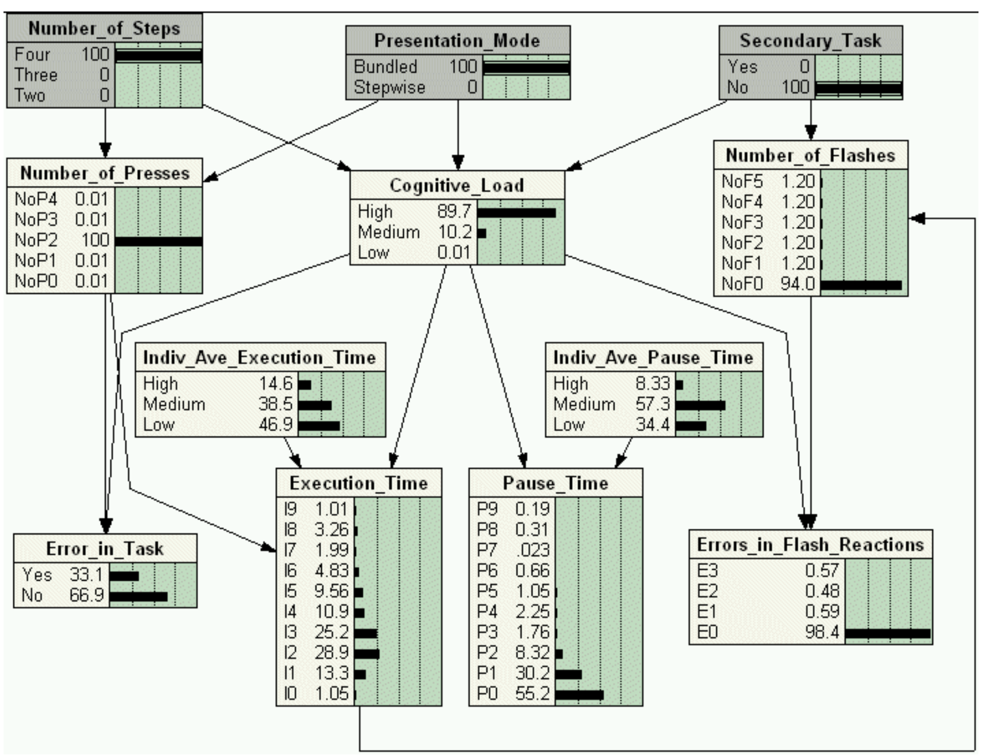
Negative points

- There are still some constraint violations

4: Constraining the Nature of Relationships (3)21



4: Constraining the Nature of Relationships (4)22



5: Choosing Learning Methods Flexibly (1)

Basic idea

Each CPT can be seen as a learning problem with its own specific features

So why not choose the most suitable learning technique for each CPT (cf. Musick, KDD96)?

Example:

If you think that *A* and *B* have a linear influence on *C*, use linear regression to estimate the parameters

Simple application here

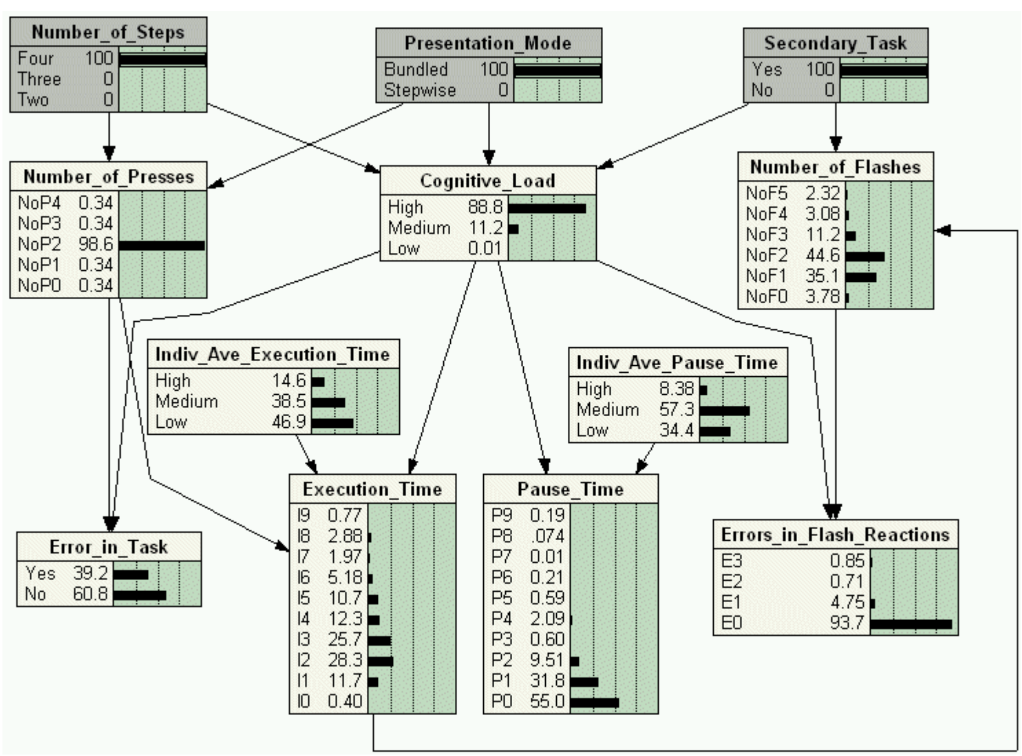
1. For CPTs that involve only observable variables, use simple methods
2. Then fix these CPTs before starting to use gradient descent

Positive points

- Saves a lot of learning time

  Here: about 1/3
- Perhaps better prediction of extreme observations?

5: Choosing Learning Methods Flexibly (2)



Conclusions

What have we done?

- First(?) example of learning an BN with a hidden variable for user modeling
- Example of using BN learning to explain results of a psychological experiment
- Identification of several problems that seem especially important for BN learning in this context
- Outline of briefly tested possible solutions to these problems

What do we have to do now?

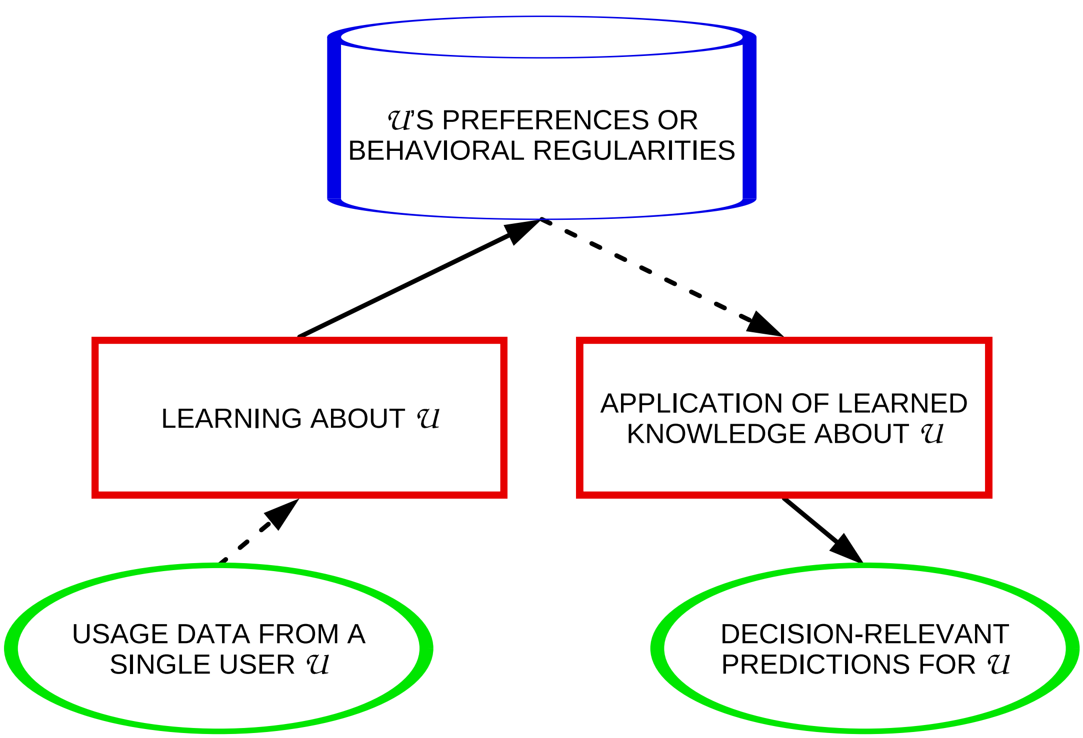
- Investigate possible answers more thoroughly
- In particular perform thorough and systematic evaluations
- Look into further issues of this sort

  E.g., What is the best criterion here for evaluating a learned net?

  Should it be evaluated in terms of success at the particular tasks for which the net is to be used?

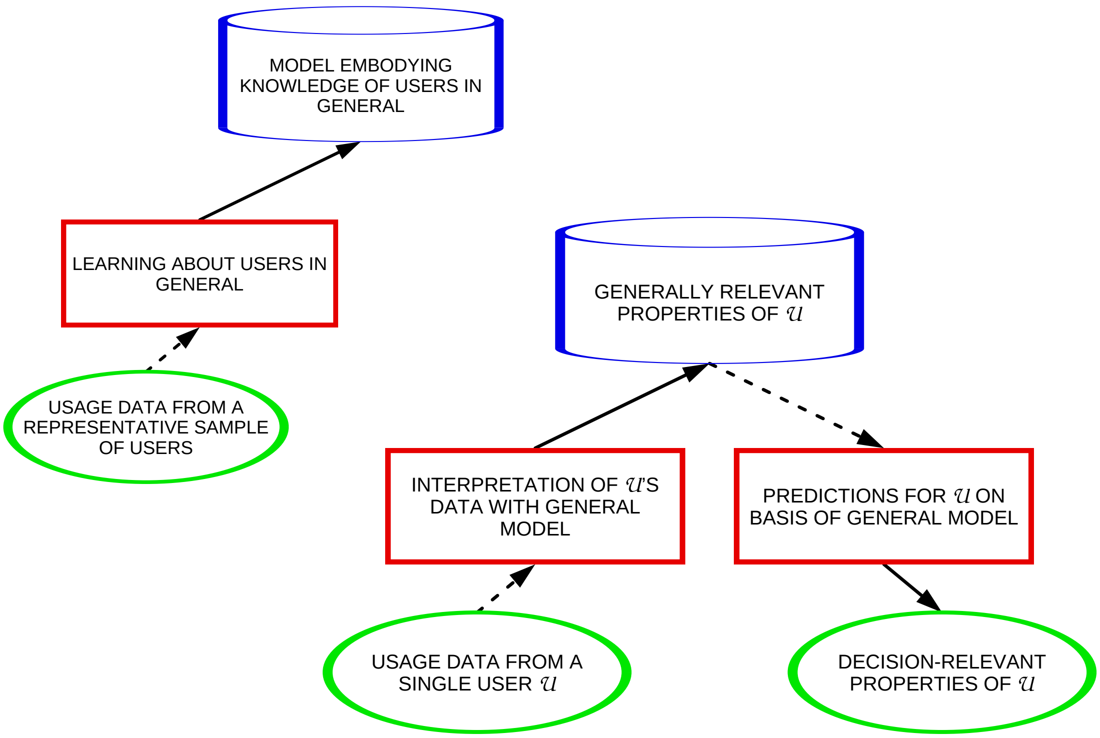
  Cf. Greiner et al. (UAI97); Kontkanen et al. (UAI99)

Why Learn About Users-in-General?
Learning About Individual Users



Learning About Users in General

27



Which Approach to Use?

28

When to learn for users in general?

- Useful generalizations can be made about all users
- These generalizations are not obvious but must be learned from data
- Only limited data is available about any given user

When to learn for each individual user?

- There are few nontrivial generalizations
- Individual users differ not only in details but in their overall structure, strategies, etc.
- A reasonably large about of data is available for each user