Dynamic Bayesian Networks for Student Modeling

TODO: Read Paper, summarize and write down questions
Next TODO: Copy method of paper and do BKT and build Factorization according to theory in paper

High-level summary:

Comparing the performance of 3 models:

1. BKT_c

- Use method from paper: [2108301741L Individualized Bayesian Knowledge Tracing Models]

2. Performance Factor Analysis: PFA

- Use method from paper: [2110171427L Performance Factor Analysis - A New Alternative to Knowledge Tracing]
- Logistic Regression Model, models the probability of solving a task t correctly

$$p_{st} = (1 + \exp(-(\theta_s + \sum q_{kt}(\beta_k + \gamma_k \cdot S_{sk} + \rho_k \cdot F_{sk}))))^{-1}.$$

3. Dynamic Bayesian Network

- Use method from paper: [2110171455L Beyond Knowledge Tracing Modeling Skill Topologies with Bayesian Networks] and builds on [2110171505L Computational Education Using Latent Structured Prediction]

- Define domain-specific DBN's modeling skill hierarchies for 5 large-scale data sets

Bayesian Knowledge Tracing can only represent one skill at a time. Dynamic Bayesian Network (DBN) can represent **multiple skills jointly within one model**. The most popular algorithms to estimate student knowledge performance are performance factor analysis (PFA) and Bayesian Knowledge Tracing (BKT).

This paper contributes to 1.) The research of modeling (skill hierarchies) and 2.) The research of assessment (of student models).

Details Summary:

Background: Student Models

Latent Factor Models

- applies logistic regression
- Assumption: the probability of a correct response can be represented by a mathematical function of student and skill parameters. Binary task outcome = Bernoulli distribution

There are 2 types:

1. Additive Factor Model (AFM)

2. Performance Factor Analysis (PFA)

Bayesian Knowledge Tracing

- Special case of DBN (Dynamic Bayesian Networks) and HMM
- **Latent Variable** = Student knowledge about a specific skill, binary
- **Observed Variable** = Student answers (correct/incorrect)

The state of the latent variable is inferred based on the observed variables. There are 2 probabilities in a HMM: Emission probabilities and transition probabilities (Also see A friendly intro to HMM).

- **Transition Probability:** Transitioning from unknown to known state
- **Emission Probability:** P(Guess) and P(Slip)

We employ one BKT per skill. The learning task is to estimate the parameters (P(Guess), P(Slip), P(L_0), P(Forget), P(L)) given some observations.
Given a sequence of observations

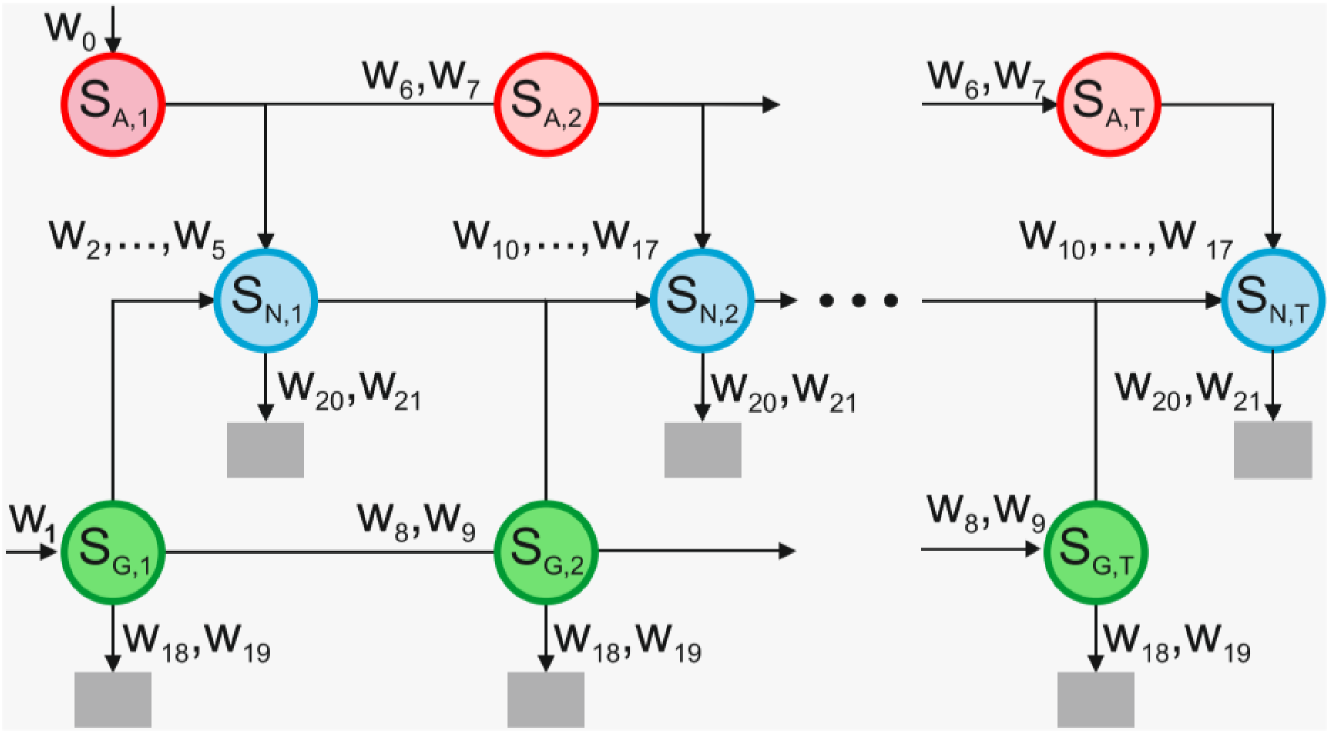
$y_m = (y_{m,1}, \ldots y_{m,T})$ with $y_{m,t} \in \{0,1\}$ and time $t \in \{1,\ldots,T\}$ for the m-th student with $m \in \{1,\ldots,M\}$ what are the parameters $\theta = \{p_0, p_L, p_F, p_S, p_G\}$ that maximize the likelihood $\prod_m p(\mathbf{y_m} \mid \theta)$

This task is solved with

1. **Brute-force grid search:** 2109091054L Contextual Slip and Prediction of Student Performance after Use of an Intelligent Tutor
2. **Gradient Descent:** 2108301741L Individualized Bayesian Knowledge Tracing Models
3. **Expectation Maximization:** 2109091055L Intelligent Tutoring Systems_A Bayes Net Toolkit for Student Modeling in Intelligent Tutoring Systems

Methods: Dynamic Bayesian Networks

We add: Dependencies between the different skills. If S_A and S_B are conditionally dependant, then S_A is a perequisite for mastering S_B



The figure shows a DBN unrolled over T time steps. The circular nodes represent binary skill variables. The rectangular nodes represent observable variables (binary student answer of incorrect/correct).

Probabilistic Notation

H : Unobserved Variables/Latent Space
y : Observed Variables
m : student
Example

- Student m solves a task associated with skill $S_{G,1}$ in first time step correctly: $O_{G,1} = 1$.
- $y_m = O_{G,1}$
- $h_m = S_{A,1}, S_{N,1}, O_{N,1}, S_{A,1}$ (all other variables)

When learning we want to find the parameters θ that maximize the likelihood of the observed data $\bigcup_m y_m$ with $y_m = (y_{m,1}, \ldots, y_{m,T})$, a sequence of T binary answers from student m
Log-likelihood of a DBN:
$L(\theta) = \sum_m ln(\sum_{h_m} p(y_m, h_m|\theta)$ where:
$p(y_m, h_m|\theta) = \prod p(X_{m,i} = x_{m.i}|pa(X_{m,i}) = x_{m,pa}(X_{m,i}) = \prod p_{ij_{m,i}k_{m,i}})$

Log-Linear Models

An alternative to formulate the log-likelihood of a DBN is using a log-linear model which is more flexible and used.
To reformulate we map the latent space H and the observed space x to an F-dimensional feature vector: $H \rightarrow \mathbb{R}^F$.
Reformulated log-likelihood:
$L(w) = \sum_m ln(\sum_{h_m} exp(w^T \phi(y_m, h_m) - ln(Z)))$
where:

- Z = normalizing constant
- w = weights of the model

Optimization

- Using log-likelihood formulation
- Algorithm from Paper: Computational Education using Latent Structured Prediction
- Direct application on DBN Model, approach was extended by including constraints on parameters

Specification

The figure shows 3 skills. S_N, S_G have tasks associated with them. S_A cannot be observed.
We set $F = 22$ weights that can be associated with a parameter set θ

$\simeq$ = proportioned to

Slip and Guess Probabilities

O_N= task associated with skill S_N
$w_2 0 \simeq p(O_N = 0|S_N = 0) = 1 - P_G$ and $w_2 1 \simeq p(O_N = 0|S_N = 1) = P_S$

Learning and Forgetting

$w_6 \simeq p(S_{A,t} = 0|S_{A,t-1} = 0) = 1 - p_L$ and
$w_7 \simeq p(S_{A,t} = 0|S_{A,t-1} = 1) = P_F$, the probability of forgetting

Skills dependencies

S_A and S_G are prerequisites for knowing S_N.

The probability that S_N is mastered in t depends on 1.) state of skill S_N in the previous step and 2.) on the states of S_A and S_G in the current time step

$w_{10} \simeq p(S_{N,t} = 0 | S_{N,t-1} = 0, S_{A,t} = 0, S_{G,t} = 0) = 1 - p_{L0}$ where p_{L0} = probability that the student **learns S_N despite not knowing S_A and S_G** and

$w_{17} \simeq (S_{N,t} = 0 | S_{N,t-1} = 1, S_{A,t} = 1, S_{G,t} = 1) = p_{F1}$, the probability of **forgetting a previously learnt skill**

$w_l \simeq 1 - p_{LM}$ if $l \in (11, 12, 13)$ and $w_l \simeq 1 - p_{FM}$ if $l \in (14, 15, 16)$

where p_{LM} is the **probability that the student learns S_N given that he knows at least one of the precursor skills of S_N**

and

p_{FM} is the **probability that the student forgets the previously known skill S_N when either S_A or S_G or none of them are known.**

.... SOME MISSING

Overview of Parameters θ

p_0

p_G = probability to guess

p_L = probability of learning

p_F = probability to forgetting a skill

p_{L0} = probability to learn a skill despite not knowing the other prerequisite skills

p_{F1} = probability of forgetting a previously learned skill

p_{LM} = probability to learn S_N given that the student knows at least one of the precursor skills of S_N

p_{FM} = probability that the student forgets the previously known skill $S_N$when either S_A or A_G or none of them are known

p_{P0} = the probability of knowing a skill despite having mastered only part of the prerequisite skills

p_{P1} = probability of failing a skill given that all precursor skills have been mastered already

Experimental Setup

- Learn parameters on training set
- Evaluate on test set
- Measure evaluation with student-stratified 10-fold cross validation

BKT_C

They used the method described in the paper [2108301741L Individualized Bayesian Knowledge Tracing Models] which

- applies skill-specific parameters
- uses gradient descent for optimization
- Set Probability of forget to 0
- Bound P_s and P_G by 0.3 (so call it BKT_C)

Generate 10 different models using different start values
Include default values from the other paper and 9 randomly generated sets of start values fulfilling the constraints
Select model with best prediction accuracy with root mean squared error on the test set comparison

- Train parameters of latent factors models with lme4 package of R *(Question: Alternative for python?)*
- For AFM and PFM, for the unseen students in the test sets, the student parameter (student proficiency θ_p) was set to the mean of the trained student parameters
- Use constrained latent structured prediction to learn the parameters of the DBN
- For interpretable parameters, we constraint the parameter set θ:
  - C_1: $p_D \leq 0.3$ for $D \in (G, S, L, F, L0, F1)$, guessing, slipping, learning and forgetting
    - The constraints on θ can be directly turned into constraints on w

Zotero Local library, Cloud library

Tags: #BayesianKnowledgeTracing #Bayes #dynamicprogramming #DynamicBayesianNetworks #Bayesian_Student_Modelling #BKT #DBN #HMM

More research needed:

- student-stratified 10-fold cross validation
- Additive Factor Model (AFM)
- Performance Factor Analysis (PFA)
- Constrained latent structured prediction ??