Knowledge based systems
Analysing the eligibility of a person for higher education using a Bayesian network

Lisa Mischer & Frank Steiler
Interactive and knowledge based systems (T2INF4307)
DHBW Stuttgart
Contact: it12147@lehre.dhbw-stuttgart.de

January 11, 2015

Contents

Chapter 1

Introduction

Within knowledge based systems, logic is a commonly used way to represent connections between data and expressions. Unfortunately logic can not handle uncertainty or unprecise data. Bayesian Networks have been developed to takle this problem, by representing knowledge as a set of variables and their dependencies within a directed acyclic graph. [**Reichardt:2014aa**]

A probabilistic network is using conditional probabilities between the nodes of the graph and inferences to calculate the probability of symptoms and/or causes. There are three types of inferences that are occurring in the network and enable the functionality of the graph: diagnostic, causal and inter-causal inference.

A Bayesian Network is defined by a set of edges (nodes) connected through vertices within a graph: $D = (V, E)$. Every node has finit set of mutually exclusive states. On top of that the network is quantifying the dependencies within a separated conditional probability table (CPT) for each node. [**Vomlel:2005aa**]

Concluding to create and then use a Bayesian Network, a user has to create the correct graph first and then determine all values for the CPT. A correct network can either be created by an expert, by using data mining techniques to find connections between entities or through machine learning.

By adding observations for a specific case to the network, the network is updating beliefs about other variables. Furthermore the probability of a certain event or state can be predicted by observing other events or states. Therefore it can support decision making and has numerous applications, like the diagnosis of diseases, automatic troubleshooting or education testing. [**Vomlel:2005aa**]

Chapter 2

Preparation of data

Bayesian networks need discrete variables to work with. Before the test data could be used, it had to be transformed. Most important was to discretise all continuous variables. Additionally, there was missing information in some cases. In the cleaned data set, we took not specified as a possible value for the variable into account. Furthermore, the data should be harmonized, e.g. standardised identifiers, consistent ranges for grades, etc.
These preparations are necessary to train the Bayesian network, as well as to analyse a certain case. Only cases which are prepared in the same way can be evaluated with our result network.

Chapter 3

Structure of the network

To find a structure for the network, we took a look at the information which we had in our data set. We considered which connections had to be definitely there, e.g. Mathematics', German's and Physic's grade had to influence the numerus clausus. Furthermore, we took it for granted that Mathematics influences OLT-Math, and same for German which influences OLT-German. Mathematics influences moreover Physics, since a basic understanding of formulas and variables is essential for Physic. To determine whether or not the Bayesian network is suited to give a recommendation for the course, we used the error rate as an indicator. The generation of the final network was an iterative process consisting of making changes of the arcs. After each change we determined the new error rate and checked if it was below the old one. If it was, then the change was saved, otherwise another change was tested.

The best error rate arrived at 12%. Figure 1 Finales Netz on the next page shows our final Bayesian network. Based on this network the recommendations are given.

Since we only used a limited version of Netica, it was necessary to limit our data set to 15 variables. We left out the state, even though it improved the error rate. Taking only 5 of 16 states into account would have been possible, but 16 different parameter values were too much to handle for Netica. Since possible students all over Germany are taken into account, it was not reasonable to work just with 5 parameter values, so we decided to leave this value out of consideration.

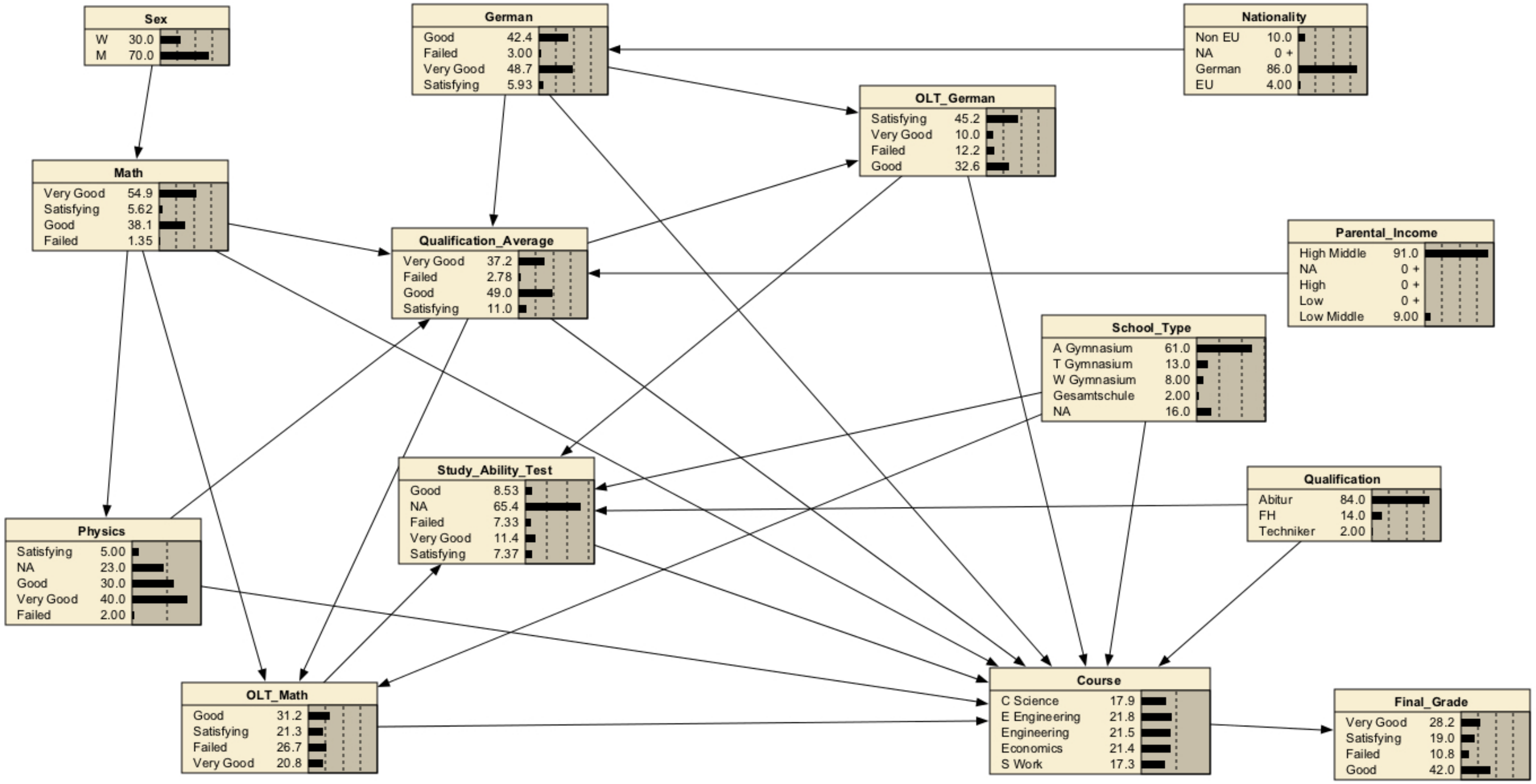

Figure 1: Finales Netz

Chapter 4

Learning conditional probability tables (CPT)

To create the network we used Netica. After we made the first draw for a network and all nodes were connected in a plausible way, the CPTs needed to be calculated. Netica offers three different functions to learn the CPTs from data. After testing each of the learning algorithms, we could achieve the best results with the Expectation Maximization Algorithm.

"Briefly, E[xpectation] M[aximization] learning repeatedly takes a Bayes net and uses it to find a better one by doing an expectation (E) step followed by a maximization (M) step. In the E step, it uses regular Bayes net inference with the existing Bayes net to compute the expected value of all the missing data, and then the M step finds the maximum likelihood Bayes net given the now extended data". [**Netica:Online**]
Other learning algorithms offered were counting, which is the simplest and fastest one, and gradient descent. Gradient Descent is faster than Expectation Maximization, whereas the latter one is more robust. Both perform better than counting, if there are missing values.
Figure 2 Parallele Koordinaten on the next page shows all trainings data in a parallel coordinate system. All these cases were taken into the calculation of the CPTs.

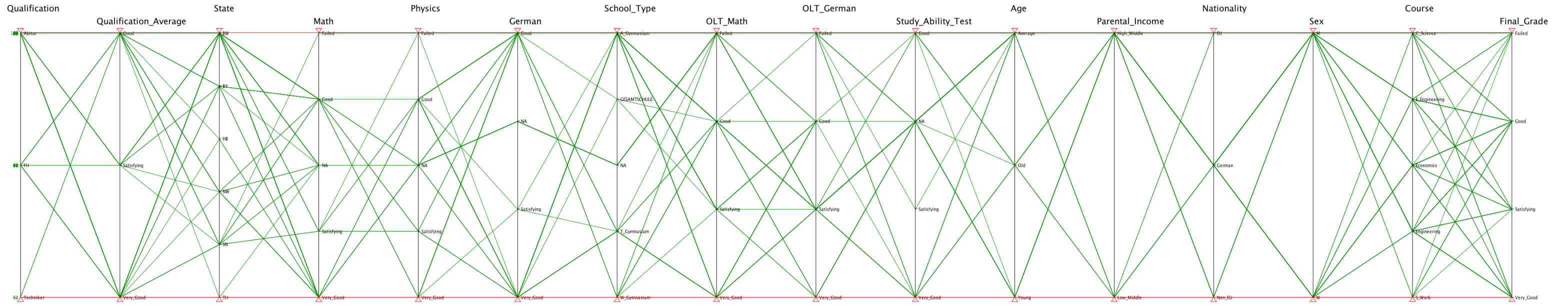

Figure 2: Parallele Koordinaten

Chapter 5

Implementation - WhatToStudy

Concrete implementation of the program, functionalities and documentation of implementation

Listings

Appendices