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 12, 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 imprecise data. Bayesian Networks have been developed to tackle 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 finite 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 and machine learning to specify the CPT values.

By adding observations for a specific case to the network, it 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

As part of our exercise we received a set of data that was supposed to help us derive the structure of the network. To be able to use the stated cases we first had to analyse and harmonise the provided data.

These preparation included the transformation of continuous variables into discrete ones, to enable their usage within the network. The variables we had to adjust were all provided grades, the test results of the online tests and study ability test, as well as the parental income.

Furthermore we chose a general syntax for the naming of values, to have standardised identifiers and consistent ranges. E.g. we defined a not available value as "NA", where the source used different words in different entities, like "keine", "n.a." and many more. We did not need to do this necessarily, but it simplified the work with the network.

A detailed description about the conversation of these values can be found within the documentation of the implementation in chapter 5 on page 9, more specific in section 5.3 on page 9.

These preparations were 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, but the implementation is able to read a cleaned

Chapter 3

Structure of the network

To find the best structure of the network, we took a look at the provided information, trying to find connections between columns of the data set.

Therefore we plotted the data within a parallel coordinate system, shown in appendix B on page 13. We used this representation to derive connections by reducing the data on a single axis to a smaller range and hoping to observe a similar behaviour on another axis. By revealing such a influence we concluded a direct connection between the entities.

On top of that we thought about logical connections, e.g. the mathematics', German's and physic's grade had to influence the qualification average. Furthermore, we took it for granted that the mathematics grade influences the results of the math online test, as well as the German grade influences the results of the German online test. Moreover the mathematics results influence the physics grade, since a basic understanding of mathematics is essential for physics.

To determine whether or not our Bayesian network is suited to predict the final grade of a specific student, we tested it using the provided data set. This test tried to predict the final grade of a person, using all available information from the data excluding the actual course and the actual final grade. By comparing the predicted and the actual result, we received an error rate for our network. We used this error rate to compare different versions of the network.

The generation of the final network was an iterative process, changing the arcs, CPTs and/or entities within each step. After changing the structure, we determined the new error rate and compared it to the previous one. By

using this process we were able to improve the overall performance of the network.

At the end we were able to achieved an error rate of 11%, predicting the course taken, and 4%, predicting the final grade of a student. Appendix A on page 12 is showing our final Bayesian network. The implementation documented in chapter 5 on page 9 is using this network to recommend for or against studying a specific course.

The tool we used to specify our Bayesian network is called Netica, unfortunately we only had access to a limited version of this program. Concluding, it was necessary to limit our data set to 15 entities. Therefore we had to leave out at least one piece of information and take a slightly more imprecise result into account. We chose to ignore the information about the state, even though it improved the error rate significantly, when taking only 5 of 16 states into account. Unfortunately it is necessary to accept all 16 available states, since all of them could be chosen. This resulted in an entity with 16 different parameter values, which were too much to handle for Netica. Since it was not reasonable to allow only 5 parameter values, we decided to leave this entity out of consideration. Furthermore we left out the income of the parents and the nationality, since these did not provide any gain in information to the network.

Chapter 4

Learning conditional probability tables (CPT)

After creating a draft of the network layout, containing all plausible node connections, we hat to quantify the dependencies between nodes within the conditional probability tables (CPT). Since we were not able to consult an expert about this problem we chose to use machine learning algorithms to generate the tables.

Fortunately the tool we used to specify the network offered a set of machine learning algorithms to generate the CPTs from a data set. These include a 'counting algorithm', an 'expectation-maximization (EM) algorithm' and a 'gradient descent algorithm'.

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". [**Corp.:2010aa**]

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.

Appendix B on page 13 shows all trainings data in a parallel coordinate system. All these cases were taken into the calculation of the CPTs.

Chapter 5

Implementation - WhatToStudy

5.1 General usage

5.2 Operation modes

5.2.1 Interactive mode

5.2.2 Evaluation mode

5.2.3 Draw mode

5.2.4 Learning mode

5.2.5 Testing mode

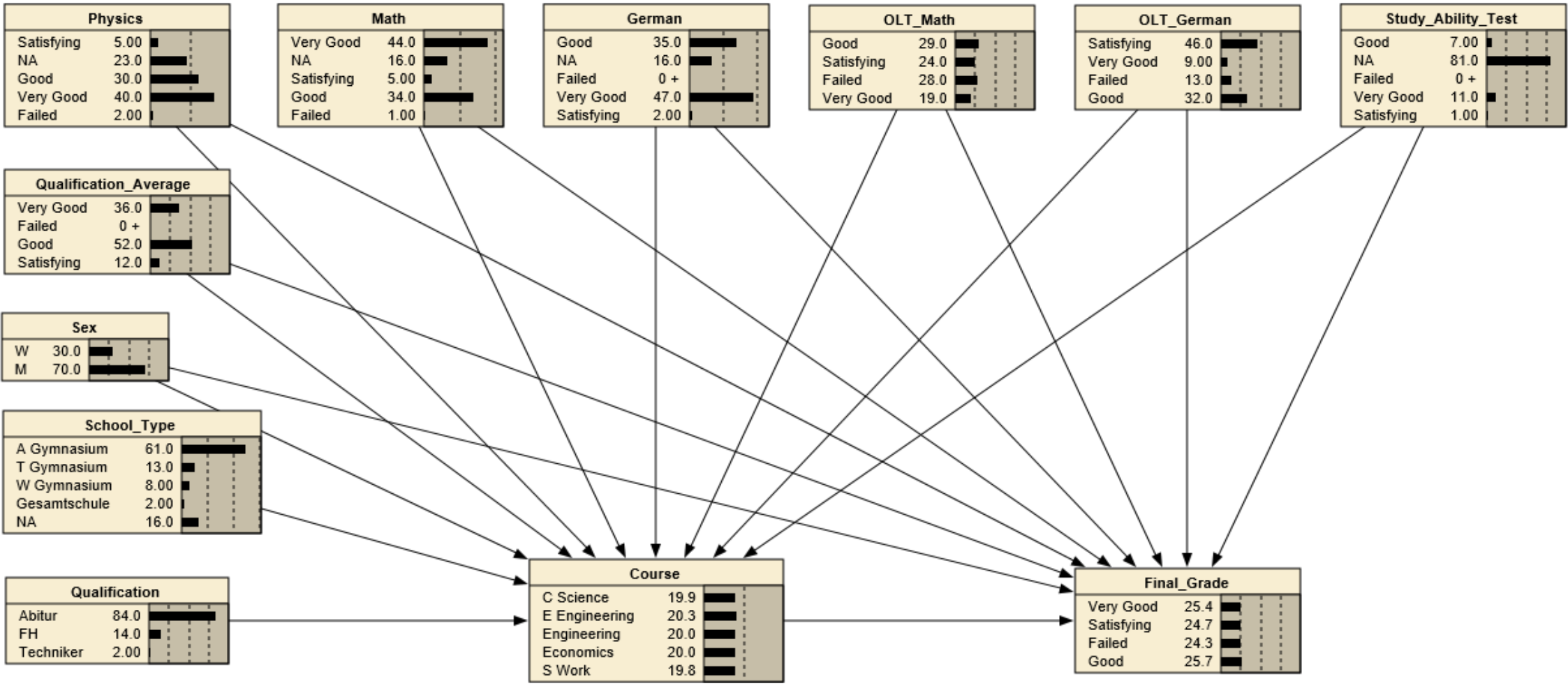
5.2.6 Help mode

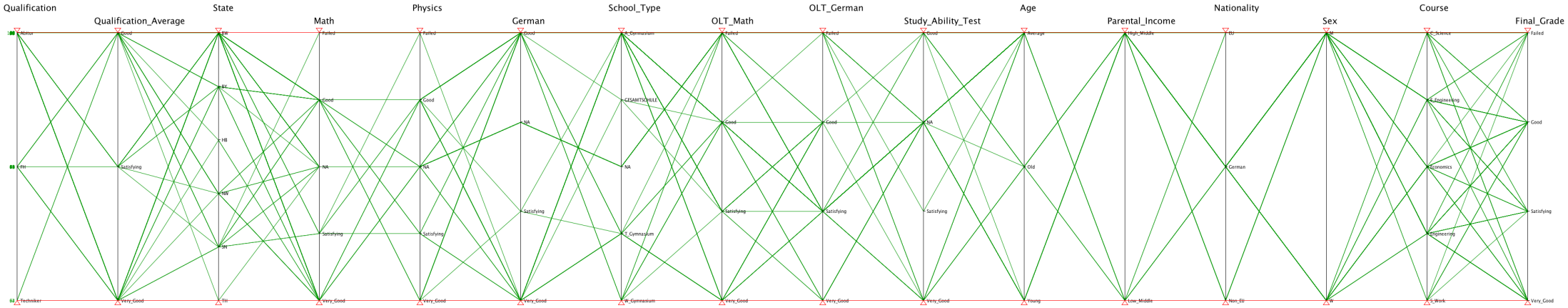
5.2.7 Version printing mode

5.3 Case file handling

Listings

Appendices


Physics
Satisfying 5.00
NA 23.0
Good 30.0
Very Good 40.0
Failed 2.00
Math
Very Good 44.0
NA 16.0
Satisfying 5.00
Good 34.0
Failed 1.00
German
Good 35.0
NA 16.0
Failed 0 +
Very Good 47.0
Satisfying 2.00
OLT_Math
Good 29.0
Satisfying 24.0
Failed 28.0
Very Good 19.0
OLT_German
Satisfying 46.0
Very Good 9.00
Failed 13.0
Good 32.0
Study_Ability_Test
Good 7.00
NA 81.0
Failed 0 +
Very Good 11.0
Satisfying 1.00
Qualification_Average
Very Good 36.0
Failed 0 +
Good 52.0
Satisfying 12.0
Sex
W 30.0
M 70.0
School_Type
A Gymnasium 61.0
T Gymnasium 13.0
W Gymnasium 8.00
Gesamtschule 2.00
NA 16.0
Qualification
Abitur 84.0
FH 14.0
Techniker 2.00
Course
C Science 19.9
E Engineering 20.3
Engineering 20.0
Economics 20.0
S Work 19.8
Final_Grade
Very Good 25.4
Satisfying 24.7
Failed 24.3
Good 25.7


Qualification
Qualification_Average
State
Math
Physics
German
School_Type
OLT_Math
OLT_German
Study_Ability_Test
Age
Parental_Income
Nationality
Sex
Course
Final_Grade
Abitur
FH
Technker
Good
Satisfying
Very_Good
BW
BY
HB
NI
SN
TH
Failed
Good
NA
Satisfying
Very_Good
Failed
Good
NA
Satisfying
Very_Good
G_Gymnasium
GESAMTSCHULE
NA
T_Gymnasium
W_Gymnasium
Failed
Good
NA
Satisfying
Very_Good
Failed
Good
NA
Satisfying
Very_Good
Good
NA
Satisfying
Very_Good
Average
Old
Young
High_Middle
Low_Middle
EU
German
Non_EU
M
W
C_Science
E_Engineering
Economics
Engineering
C_Work
Failed
Good
Satisfying
Very_Good