A Bayesian Network Approach to Improving Student Performance for College Admission

By Bogdan Tesileanu

Introduction

Learning institutions are everywhere, and can be as diverse as their students in terms of practices and support. For this reason, understanding the situations and motivations of each student allows the school and teachers of that school to best prepare students for the future ahead. As a result, schools must keep in mind the differences among students and their own economic situation at home in order to better understand their chances of a college admission.

In an attempt to predict student admittance with incomplete information, we will look at different factors that influence students' performance using various random variables by creating a Bayesian Network (BN). A BN is a probabilistic model that uses a directed graph to infer outcomes based on incomplete information. The nodes on the graph contain a set of probability distributions and the edges represent the influence each node has on other nodes. A Bayesian Network framework was chosen as opposed to others because it works well with incomplete information and is especially flexible when inferencing, having the ability to infer the values of all variables using the underlying algorithm.

The paper will contain 5 parts, the first of which will be *Background*. This section will provide relevant information about the framework being used and will go into some detail about the education variables that were chosen. The *Methodology* section will go in depth into the application, pseudocode, and any technical aspects involved with the program. Implementation of the program and outputs of running codes to discover the effects of different factors on college acceptance will be done in the *Results* section. Finally, the *Conclusion* will contain all key takeaways, biases, and any improvements of the application that may be possible in the future.

Background

Education is of utmost importance in most countries, and thus has been widely studied. While some of the research happened across the seas, it is all relevant in terms of what boosts performance and what doesn't. Author Jennifer Barry highlights the importance of a student's socioeconomic status, finding that it was the biggest factor out of all that were studied. A lower status implies more stress and conflict at home, as well as less access to resources. Moreover, author Orhan Kara reported that school absences correlated negatively at about 22% with grades. Once students get behind, they are stuck in a vicious cycle of making up work while more work is accumulating. Similarly, parent income was also positively correlated in this case. A higher income yields better resources, and more resources means the student is able to gather and absorb all the information they need more efficiently. More recently, Realyvásquez-Vargas decided to do the same investigations under the COVID-19 pandemic. He focused on factors that affected student learning at home. These included lighting, noise, and temperature. These factors were chosen since they differ from a regular school-setting classroom to one's bedroom. It was found that lighting and temperature had the biggest impact on concentration and focus, followed by noise. In a classroom, such variables are held constant, while at home (especially with poorer families), many of these conditions may be less than ideal. Last but not least, Kawtar Tani conducted a study that, like Kara's, focused on student absenteeism. According to the study, students in higher level classes had better attendance record than those in lower, more basic classes, and that increase meant better grades. The study also found that an increased number of dependents at home was correlated to an increase in grades. This effect could be attributed to an increased sense of responsibility at home, and a desire to succeed to help the family or significant other. There are many other factors that go into student grades and performance, but only a few were highlighted here and will be used throughout the paper and for the Bayesian Network application.

A Bayesian network is a probabilistic graphical model that uses a directed acyclic graph (DAG). The variables are represented by nodes and the causal dependence of each node on the rest is illustrated by a directed arch. Cyclic relationships are not allowed in this type of graph. A set of conditional probability distributions is also defined for the nodes that define the strength of the relationship between them. In other words, for every possible outcome in a parent node, that influence on a child node is calculated.

A Bayesian Network relies on Bayes Theorem of conditional probability, but in order to work, it requires random variables, conditional relationships, and probability distributions. To implement our model, we will create these random variables with conditional relationships among them. The mathematical relationships will be dictated by Bayes Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

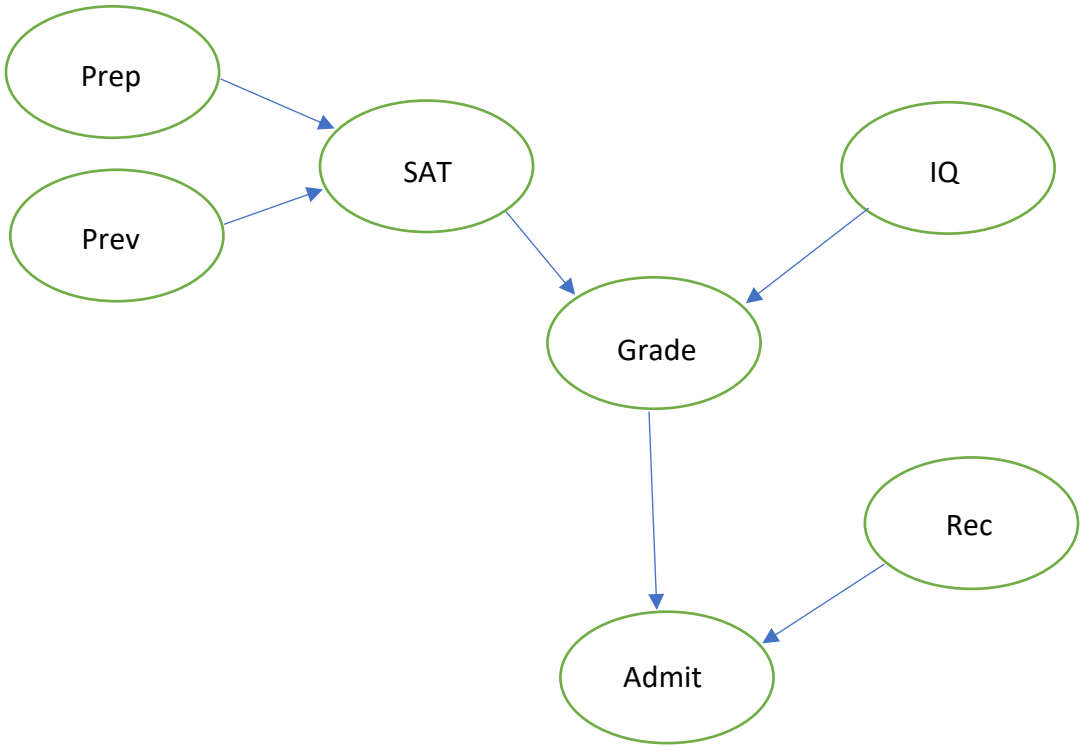
In other words, the posterior probability distribution *P(A|B)* is equal to the likelihood of an event *P(B|A)* multiplied by its prior probabilities *P(A)*, normalized over a marginalization factor *P(B)*. The algorithm computes this outcome each time the program is run, and updates the probabilities accordingly. The theorem has many other applications, but this program will be a small glimpse into its usability.

Methodology

To create a Bayesian Network for this task, we will use Jupyter Notebooks and the "pomegranate" package (version 0.14.6). The package allows the user to define a BN or to create one by learning from the data. In this case, we will define our own. The goal is to find out whether a student will be admitted to college or not, therefore we will use the following variables:

Prep: whether or not a student took a preparatory SAT course
Prev: previous SAT scores
SAT: SAT score
IQ: Level of IQ
Grade: Overall Grade/GPA in school
Rec: the number of recommendation letters on file
Admission: whether or not the student was admitted to college

Creating a DAG with the above variables leads to the following:



As illustrated, there are 7 variables that will be included in the network. The first is **Prep**, which is whether or not the student has taken any kind of preparatory SAT scores. These are sometimes offered by the school, or sometimes privately. They do cost a decent amount of money, so not all students choose to attend one, even though they are known to be beneficial. Next, the **Prev** variable describes whether the student had high or low previous SAT scores. Generally, students take the SAT two or three times to try to achieve the best score possible. However, the scores are usually within one or two standard deviations of each other. The **SAT** variable describes the highest score the student received on his or her test. Test scores can be sent singly or as a combination if multiple are taken, and thus the highest score in each of the three sections is usually counted. **IQ** is the fourth variable in the network and is one that doesn't usually change with any outside influences. There have been multiple studies that have tried to teach people techniques and strategies but have failed when it comes to raising IQ. The **Grade** variable is affected by both SAT and IQ, assuming that all three of those factors are positively correlated (i.e. if IQ and SAT score is high, Grades are probably also high). It is assumed that someone with a high IQ and high SAT score will also do well in their classes since it will be easier for them to learn. **Rec** refers to the number of recommendation letters each student submitted on their college application. Usually, the more recommendations the better, but the content is also crucial in the decision. Finally, **Admit** is the variable of interest. It describes the likelihood that the student will have a high or low chance of admission. Realistically, several more factors would be included in the decision-making process, but for the purpose of this paper, we will restrict the variables to the above seven.

Python 3 and *pomegranate* make it easy to code the above network. Below is the pseudocode:

```
for all variables:
        initialize and give name
        if root node:
                assign probability
        if not root node:
                assign conditional probability
for each arch:
        connect from variable A to variable B

initialize the network
predict using the network
```

The random variables "**Prep**", '**Prev**", "**IQ**", and "**Rec**" will all have discrete probability distributions since they are root nodes. The rest of the variables have been assigned conditional probabilities given their parents, all of which can be found in the Appendix. At the moment of this paper, pomegranate does not support continuous prior probabilities, therefore only discrete probabilities have been used. All probabilities can be found in the Figures section (page 12).

Github Repository for working program:

https://github.com/tesileanu/745/blob/3668b23b7c53b44a0913abdc3efad30d6c9c534d/ProjectCode.ipynb

Results

Using Jupyter notbooks and the pomegranate package, we are able to make some predictions with the limited amount of information we have.

Example 1:

If someone does a prep course, does "good" on previous SATs, and has a high IQ, then they are more likely to have 1 recommendation letter and a 50% chance of being admitted to college (details in Appendix). There is also 80% belief that the student achieved a "high" score on the SAT. This is backed up by the "givens" that the student previously took a prep course and achieved high marks on the same test. This information only provides a subtle glimpse into the student's admission decision, giving it only a 50% belief, since the recommendation letters are still unknown.

Example 2:

Next, suppose we only know that the student in question has a high IQ, but we have no other information. Assuming nothing else about the other parameters, the grade of the student and the admittance probability will increase slightly, intuitively speaking. The code yields modest increases in both, specifically a 66% chance of having a "High" grade and a 58% chance of having a "High" possibility of admittance.

Example 3:

Often, students can be exceedingly smart, but perform poorly on standardized tests. We will simulate this scenario with a student having a "High" IQ but having a "low" SAT score. Before, we reach the admittance numbers, we will discuss the rest of the variables. With the above conditions, there is only a 13% chance that the student ever took a preparatory class. Likewise, there is a 65% chance that the student scored "Bad" on previous SATs. Nevertheless, the contradicting parameters that were assigned in this scenario will still yield a 50% chance of a "High" grade, and therefore a 50% chance of college admittance.

Example 3 (extended)

If we include the number of recommendation letters the student received in the above example, having only one will decrease their chance by about 10%, but having 3 would raise it by 30%. Having two would have no effect.

The extended Example 3 is a clear indication of how this system should be used, if implemented in the real world. Each student could have their own profile and their counselors or advisors would be able to make small changes to their grade, SAT scores, or recommendation letters to see which change would increase their admittance rate the most. For instance, instead of retaking a class to get a higher grade or GPA, a student could get a more efficient increase in admittance probability by inquiring about a third recommendation letter. Nevertheless, both actions should be advised, but sometimes there are time constraints and only certain changes can be made.

Conclusion

In summary, a general trend has revealed itself that a student's previous achievements will directly impact their admission. The Bayesian model aided the discussion by predicting the outcomes with the limited amount of information available. Though the conditional probabilities and discrete probabilities are realistic, they are not accurate or up to date and therefore should not be regarded as such. Nevertheless, the model proves the usefulness of such a program for schools around the country, and the effectiveness of Bayes Nets in the real world. Regardless of the causation for these differences in test scores, schools should attempt to minimize the economic effects on student learning through different enrichment programs.

The author of the paper is a teacher himself, and therefore may have slight biases towards the processes and scenarios seen on a day-to-day basis. For instance, the importance of recommendation letters may be overstated in the model, or the economic status of students may be even more impactful than considered (such as those who can't take SAT prep classes because they can't afford them vs. those who won't take them out of indifference).

In the future, more accurate data can be used to assign the probabilities found in individual counties, cities, or schools. The data can be adjusted with more variables as long as there exists enough information to do so. As a prototype, the current program work to demonstrate the use of a Bayesian Network and may in the future be used to improve college admissions efficiently.

References

Barry, Jennifer. “THE EFFECT OF SOCIO-ECONOMIC STATUS ON ACADEMIC ACHIEVEMENT.” Https://Soar.wichita.edu/, Dec. 2006, soar.wichita.edu/bitstream/handle/10057/616/t06115.pdf?sequence=3&isAllowed=y.

Horny, Michal. “Bayesian Networks” BU.edu, Boston University, 18 Apr. 2014, https://www.bu.edu/sph/files/2014/05/bayesian-networks-final.pdf.

Kara, Orhan, et al. “Factors Affecting Students Grades In Principles Of Economics.” American Journal of Business Education (AJBE), vol. 2, no. 7, 2009, pp. 25–34., doi:10.19030/ajbe.v2i7.4581.

Realyvásquez-Vargas, Arturo, et al. “The Impact of Environmental Factors on Academic Performance of University Students Taking Online Classes during the COVID-19 Pandemic in Mexico.” Sustainability, vol.12, no. 21, 2020, p. 9194., doi:10.3390/su12219194.

Tani, Kawtar, et al. “Evaluation of Factors Affecting Students' Performance in Tertiary Education.” Journal of Pedagogical Research, vol. 3, no. 2, 2019, pp. 1–10., doi:10.33902/jpr.2019252504.

Figures

Probability charts:

Prep	
Yes	25%
No	75%

Prev	
Good	50%
Bad	50%

IQ	
High	50%
Low	50%

Rec	
1	50%
2	33.3%
3	16.7%

SAT			
Prep	Prev	SAT Outcome	Probability
Yes	Good	High	80%
Yes	Good	Low	20%
Yes	Bad	High	60%
Yes	Bad	Low	40%
No	Good	High	50%
No	Good	Low	50%
No	Bad	High	10%
No	Bad	Low	90%

Grade			
SAT	IQ	Grade Outcome	Probability
High	High	High	90%
High	High	Low	10%
High	Low	High	60%
High	Low	Low	40%
Low	High	High	50%
Low	High	Low	50%
Low	Low	High	10%
Low	Low	Low	90%

Admit			
Grade	Rec	Admit Outcome	Probability
High	1	High	70%
High	1	Low	30%
High	2	High	80%
High	2	Low	20%
High	3	High	90%
High	3	Low	10%
Low	1	High	10%
Low	1	Low	90%
Low	2	High	20%
Low	2	Low	80%
Low	3	High	70%
Low	3	Low	30%

Appendix

Project Code:

```
import math
from pomegranate import *
import matplotlib
import graphviz

prep = DiscreteDistribution({'Yes':1/4, 'No':3/4})
prev = DiscreteDistribution({'Good':1/2, 'Bad':1/2})
iq = DiscreteDistribution({'High':1/2, 'Low':1/2})
rec = DiscreteDistribution({'1':1/2, '2':1/3, '3':1/6})

sat = ConditionalProbabilityTable(
   [['Yes', 'Good', 'High', 0.8],
   [ 'Yes', 'Good', 'Low', 0.2],
   [ 'Yes', 'Bad', 'High', 0.6],
   [ 'Yes', 'Bad', 'Low', 0.4],
   [ 'No', 'Good', 'High', 0.5],
   [ 'No', 'Good', 'Low', 0.5],
   [ 'No', 'Bad', 'High', 0.1],
   [ 'No', 'Bad', 'Low', 0.9]], [prep, prev])

grade = ConditionalProbabilityTable(
   [['High', 'High', 'High', 0.9],
   [ 'High', 'High', 'Low', 0.1],
   [ 'High', 'Low', 'High', 0.6],
   [ 'High', 'Low', 'Low', 0.4],
   [ 'Low', 'High', 'High', 0.5],
   [ 'Low', 'High', 'Low', 0.5],
   [ 'Low', 'Low', 'High', 0.1],
   [ 'Low', 'Low', 'Low', 0.9]], [sat, iq])

admit = ConditionalProbabilityTable(
   [['High', '1', 'High', 0.7],
    ['High', '1', 'Low', 0.3],
    ['High', '2', 'High', 0.8],
    ['High', '2', 'Low', 0.2],
    ['High', '3', 'High', 0.9],
    ['High', '3', 'Low', 0.1],
    ['Low', '1', 'High', 0.1],
    ['Low', '1', 'Low', 0.9],
    ['Low', '2', 'High', 0.2],
```

```
    ['Low', '2', 'Low', 0.8],
    ['Low', '3', 'High', 0.7],
    ['Low', '3', 'Low', 0.3]], [grade, rec])

d1 = State(prep, name="prep")
d2 = State(prev, name="prev")
d3 = State(sat, name="sat")
d4 = State(iq, name="iq")
d5 = State(grade, name="grade")
d6 = State(rec, name="rec")
d7 = State(admit, name="admit")

network = BayesianNetwork("Student Performance Analysis with Bayesian Networks")
network.add_states(d1, d2, d3, d4, d5, d6, d7)
network.add_edge(d1, d3)
network.add_edge(d2, d3)
network.add_edge(d4, d5)
network.add_edge(d3, d5)
network.add_edge(d5, d7)
network.add_edge(d6, d7)
network.bake()

beliefs = network.predict_proba({
   'prep': 'Yes',
   'prev': 'Good',
   'iq': 'high',
})

print("".join( "{}t{}".format( state.name, str(belief) ) for state, belief in zip( network.states,
beliefs )))

preptYesprevtGoodsatt{
   "class" : "Distribution",
   "dtype" : "str",
   "name" : "DiscreteDistribution",
   "parameters" : [
      {
         "Low" : 0.20000000000000032,
         "High" : 0.7999999999999998
      }
   ],
   "frozen" : false
}iqthighgradet{
   "class" : "Distribution",
```

```
   "dtype" : "str",
   "name" : "DiscreteDistribution",
   "parameters" : [
      {
         "Low" : 0.5000000000000001,
         "High" : 0.4999999999999999
      }
   ],
   "frozen" : false
}rect{
   "class" : "Distribution",
   "dtype" : "str",
   "name" : "DiscreteDistribution",
   "parameters" : [
      {
         "1" : 0.4999999999999998,
         "2" : 0.3333333333333333,
         "3" : 0.16666666666666685
      }
   ],
   "frozen" : false
}admitt{
   "class" : "Distribution",
   "dtype" : "str",
   "name" : "DiscreteDistribution",
   "parameters" : [
      {
         "Low" : 0.5,
         "High" : 0.5
      }
   ],
   "frozen" : false
}
```