Naïve Bayes Classifier Model

Cristina Freitas Bazzano
Roger Williams University - Computer Science Major
cfreitasbazzano666@g.rwu.edu

HISTORY

Bayes' theorem is named after Rev. Thomas Bayes (1701–1761), who first provided an equation that allows new evidence to update beliefs.

It was further developed by Pierre-Simon Laplace, who first published the modern formulation in his 1812 *Théorie analytique des probabilités.*

MOTIVATION

Despite the fact that the far-reaching independence assumptions are often inaccurate, the naive Bayes classifier has several properties that make it surprisingly useful in practice.

In particular, the decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one-dimensional distribution.

This helps alleviate problems stemming from the curse of dimensionality, such as the need for data sets that scale exponentially with the number of features.

Faculty Advisor

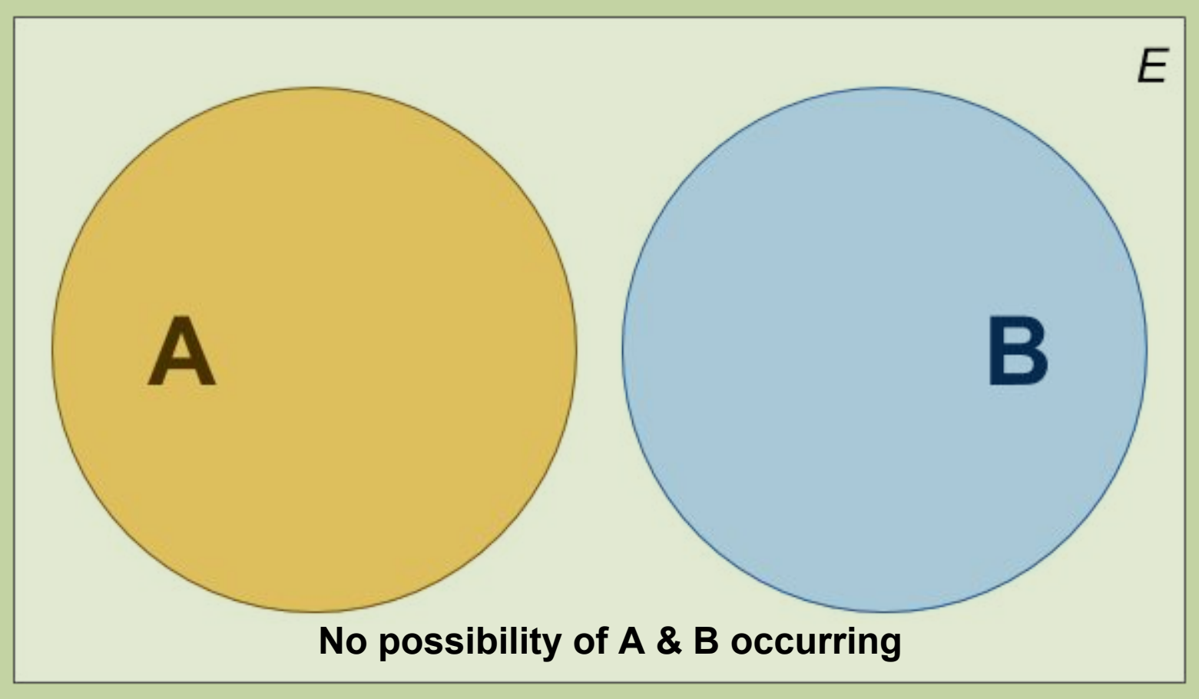
Dr. Robert Jacobson
Assistant Professor of Mathematics
rjacobson@rwu.edu

INTRODUCTION

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set.

This probabilistic classifiers are based on the Bayes Theorem assumption that the value of a particular feature is independent of the value of any other feature, given the class variable.

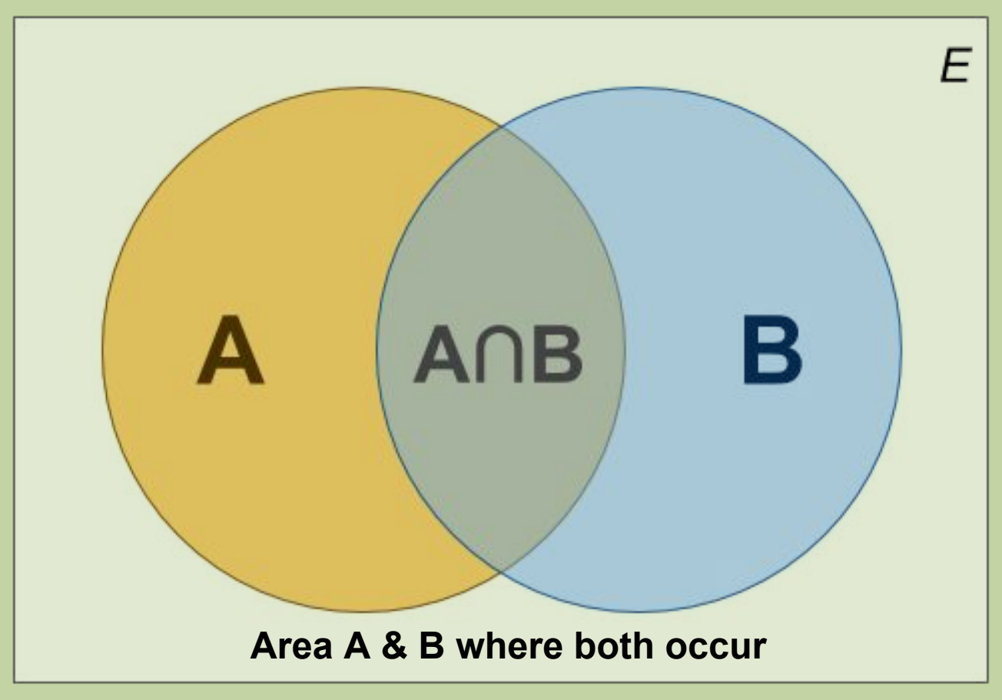
Independent Events



E = Event Space
A, B = Event Sets

Figure 1

Overlapping Event Sets



E = Event Space
A, B = Event Sets

Figure 2

METHODS AND MATERIALS

In probability theory and statistics, Bayes' theorem (alternatively Bayes' law or Bayes' rule) describes the probability of an event, based on conditions that might be related to the event.

When applied, the Bayesian probability interpretation expresses how a subjective degree of belief should rationally change to account for an evidence, this is Bayesian inference.

Bayes' theorem is stated mathematically as the following equation:

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

Given:

$$P(A \cap B) = P(B|A)\cdot P(A) = P(A|B)\cdot P(B)$$

$$P(A|B) = Condicional\ Probability\ of\ Event\ A\ given\ that\ B\ has\ happened$$

$$P(A) = Probability of Event A$$

$$P(B) = Probability of Event B$$

This model can be used to solve the following problem:

Assume we have observed occurrences of A & B in an event set. Then we want to know given B (Data), what is the probability that A (Hypothesis) has occurred.

PROBLEM

In our analysis we are interested in the final grade of a student in Calculus I, and we know the student's major, course placed in (College Algebra, Precalculus or directly in Calculus I), grade in this course and the period his going to take Calculus I.

Assuming that the final grade is related to this previous placements, then, using Bayes' theorem, informations about the student's background can be used to more accurately assess the probability that they will have a good grade in Calculus I.

Mathematically, we are trying to solve the following equation:

$$P(Y \mid Course, Course_{GPA}, Major, Period) = \frac{P(Y)\cdot P(Course|Y)\cdot P(Course_{GPA}|Y)\cdot P(Major|Y)\cdot P(Period|Y)}{evidence}$$

$$evidence = \sum_i^{A:F} P(Y_i)\cdot P(Course \mid Y_i)\cdot P(Course_{GPA} \mid Y_i)\cdot P(Major \mid Y_i)\cdot P(Period \mid Y_i)$$

$$Y = \{A \quad A- \quad B+ \quad B \quad B- \quad C+ \quad C \quad C- \quad D+ \quad D \quad D- \quad F\}$$

The independence of the naive Bayesian classifier for this model is represented by the following belief network (Figure 3), where the features are the nodes, the target variable (the classification) has no parents, and the classification is the only parent of each input feature.

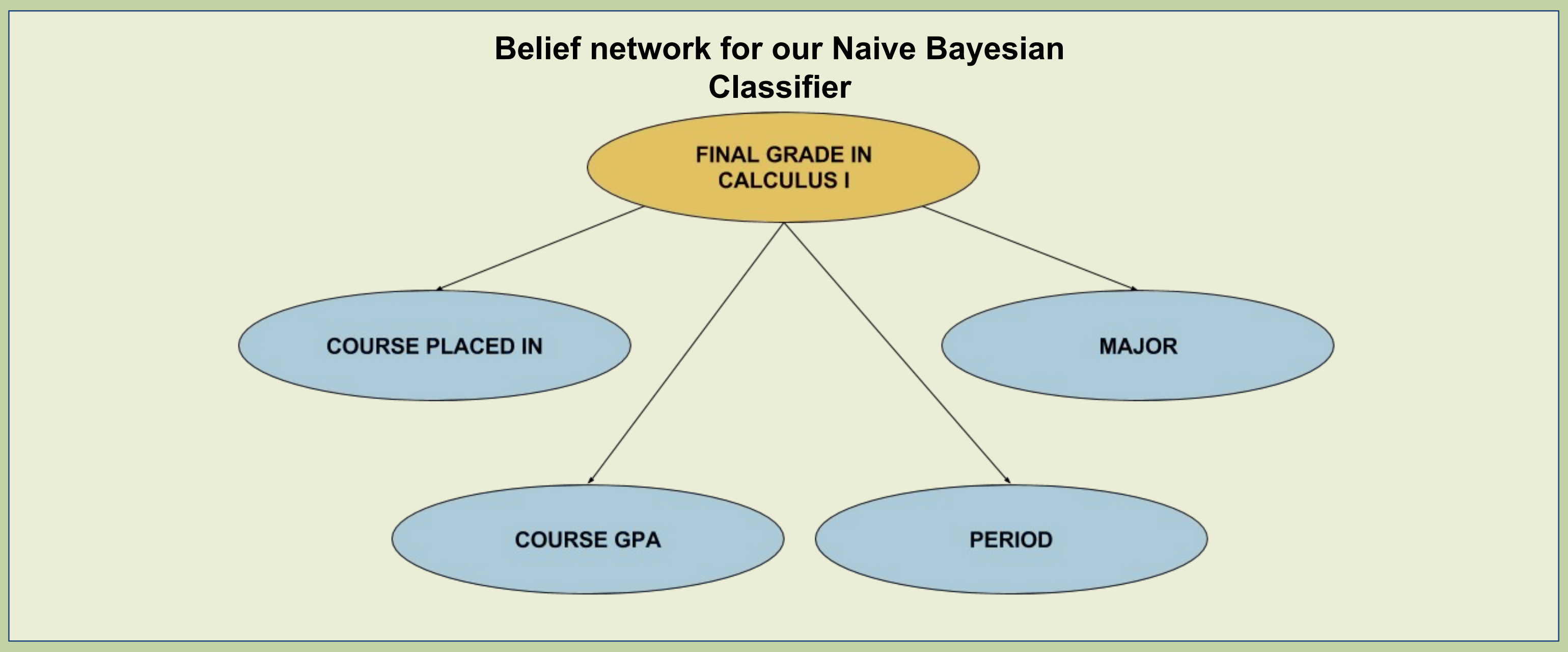

Belief network for our Naive Bayesian Classifier
FINAL GRADE IN CALCULUS I
COURSE PLACED IN
MAJOR
COURSE GPA
PERIOD

Figure 3

HOW DOES IT WORK

What we are trying to find out is the class of a new example. In a Bayesian classifier, the learning agent builds a probabilistic model of the features and uses that model in the classification.

To build our model we used a training data with a normal distribution over the features, similar to how you would find in the real world.

A sample of the our training data is shown in Figure 4.

Training Data Sample

Index	Math213	Math213Semester	Major	CoursePlaced	Course_GPA
1	C-	1	Engineering	CalcI	P
3	A-	1	Biology	CalcI	P
4	A	1	Biology	CalcI	P
5	B	1	Business	CalcI	P
6	C	3	Security Assurance Studies	CalcI	P
7	D+	1	Construction Management	CalcI	P
8	D	3	Biology	CollegeAlg	B-
9	C+	3	Computer Science	CalcI	P
10	F	2	Architecture	Precal	A-

Table 1

With this training data we are able to calculate all the conditional probabilities of the features given their class P(B|A). With this result and the probability of each class P(A) and all the features P(B) we can classify each student by outputting the probability of each class given it's features P(A|B) and selecting the class with the highest probability.

Test Data Sample

Index	Math213Semester	Major	CoursePlaced	Course_GPA	FinalGrade
1	3	Mathematics	CollegeAlg	B+	C-
2	1	Engineering	CalcI	P	A
3	1	Biology	CalcI	P	A
4	3	Engineering	CollegeAlg	A	B
5	3	Undeclared Liberal Arts	CollegeAlg	B-	C-
6	1	Engineering	CalcI	P	A
7	2	Marine Biology	Precal	A	C
8	1	Engineering	CalcI	P	A
9	1	Biology	CalcI	P	A
10	3	Engineering	CollegeAlg	A	B

Table 2

SMOOTHING

If a given class and feature value never occur together in the training data, then the frequency-based probability estimate will be zero.

This is problematic because it will wipe out all information in the other probabilities when they are multiplied. Therefore, it is often desirable to incorporate a small-sample correction, called pseudocount, in all probability estimates such that no probability is ever set to be exactly zero. This way of regularizing naive Bayes is called Laplace smoothing when the pseudocount is one for the features and #Y (number of classes) for the classes.

RESULTS

We generated some random fake data to test our built model. Our test data is a combination of the possible features, representing new students that will take Calculus I.

A sample of our test data with the classification for each test example is shown in *Table 2*. The Final Grade is the output of our model.

Our prediction has an average accuracy of 65% with no class error. When admitting one class error this accuracy increases to 70%. The model has an average class error of 3.5 classes.

When doing this prediction randomly you would have 8% of chance to get the right final grade of a student between the 12 possibilities. That shows how good our model can increase this prediction.

CONCLUSIONS

The naive Bayesian classifier we built works well with our independence assumption. The results we collected showed us that the class we used is a good predictor of our features.

After analysing our results, we notice that students that were placed directly in Calculus I are more likely to pass the course than students placed in more basic courses first. This is a reasonable result, since students placed directly in Calculus I must have a stronger mathematical bases and can handle more easily the advanced mathematical problems.

We could also infer that the final grade in Calculus I of students placed in one of the basic mathematical courses first is usually smaller than the grade that the student got in this basic course.

In a future project, this model could be used to analyse how effectively the pre courses of Calculus I are. Depending on the results of a classification using this model, the minimum grade to pass this basic courses could be rethought.

REFERENCES

- Poole, David; Mackworth, Alan (2010). Artificial Inteligence Foundations of Computational Agents. http://artint.info/html/ArtInt_181.html
- Ng, Andrew. Machine Learning Lecture Notes CS229. http://cs229.stanford.edu/notes/cs229-notes2.pdf

Roger Williams University