Cameron Zurmuhl

CS200

Professor Dahl

24 March 2018

How Bayesian Classifier Algorithms Influence Predication with Uncertainty

Modern data expansion has pushed the need for more advanced analytical tools that computers provide. The exponential increase in information repositories has led to unorganized data collection (Mishan, et al. 59), and this lack of structure creates inefficiencies in human analytics. In addition to the time deficiency in having a human mine through a massive, messy data set, humans are noted to have biases in decision-making when the data is presented in a specific order or recency, which can lead to incorrect results (Sarker & Sriram, 1458). To approach the problems that technology users and businesses need solutions for, it is favorable to turn to probability for making decisions with uncertainty. Using algorithms rooted in a probability calculus for predictions is beneficial because of their disregard for recency or order in the data presented. Plus, humans still have control of the decision-making process by specifying a probability threshold for an outcome. For example, a manager can claim that only if an outcome is defined with 97% probability will he act. Because of their ease of implementation, expandability, and interoperability, Bayesian classifying algorithms have significantly influenced how analysts have worked with data to make predictions with uncertainty.

A problem in data science is prediction and classification of objects given a set of attributes, or evidences. Businesses may want to know the probability of a certain customer clicking on their advertisement given that she is female, lives in Santa Barbara, and has a college degree. Naturally, it is a problem rooted in conditional probability. The Naïve Bayes classifier algorithm is a machine learning algorithm that predicts the unknown class of a target variable. Given training data, the algorithm can predict whether the person will click on the advertisement, given her three attributes.

The algorithm builds on Bayes' Rule from probability (Rosen, 470): suppose that E is an event from a sample space S and that $C_1, C_2,..,C_n$ are mutually exclusive events such that $\bigcup_{i=1}^{n} C_i = S$. Assume that $P(E) \neq 0$ and $P(C_j) \neq 0$ for $j = 1, 2, \ldots, n$. Then:

$$P(C_j \mid E) = \frac{P(E \mid C_j)P(C_j)}{\sum_{j=1}^{n} P(E \mid C_j)P(C_j)} \tag{1}$$

$P(C_j \mid E)$ is known as the posterior probability, or the probability that given certain evidence (one or more attributes), the target variable belongs to class C_j. $P(E \mid C_j)$ is the likelihood of seeing evidence E given class C_j. $P(C_j)$ is the prior probability of class C_j, absent conditions, and $P(E)$ is the likelihood of evidence E. $P(E)$ is calculated by using the Law of Total Probability in the denominator of equation 1. If the evidence E is partitioned among n classes, then the probability of that evidence is the sum of the probabilities of the evidence given each class[1]. The algorithm therefore classifies the target variable from the maximum obtained posterior probability for all C_j, and all probabilities except for the posterior are estimated from training data. There are advantages and disadvantages to using this algorithm. The advantages include simplicity, efficiency, and sufficient performance in real-world applications. However, it is called naïve because of an inaccurate assumption of independence among evidences given a classification, potentially leading to false probabilities. This paper will first present an argument for using conditional probability in the real world, then explore the Naïve Bayes classifier's applications in areas such as spam filtering, predicting bank failures, and business intelligence.

I. An Argument for Conditional Probability

Before exploring the applications of the Naïve Bayes classifier algorithm, I would like to briefly argue why we should consider using automated conditional probability analysis over human instinct in some real-world situations. Consider the Monty Hall dilemma, a problem named after the 1970s game show host, as presented in Henk Tijms' *Understanding Probability* (2012):

[1] It should be noted that $P(E)$ is often impractical to calculate in the real world. The Naïve Bayes algorithm is not an exact copy of Bayes' rule, and uses just the numerator to compare outcomes. The inclusion of $P(E)$ in the denominator is used to normalize numerical results to probabilities. The equations in this paper will include the denominator for consistency in discussion of probability, but the actual algorithm does not need to normalize to probabilities for correct results (Sahami, Mishan et al., 62).

> The contestant in a television game show must choose between three doors. An expensive automobile awaits the contestant behind one of the three doors, and joke prizes await him behind the other two. The contestant must try to pick the door leading to the automobile. He chooses a door randomly, appealing to Lady Luck. Then, as promised beforehand, the host opens one of the other two doors concealing one of the joke prizes. With two doors remaining unopened, the host now asks the contestant whether he wants to remain with his choice of door, or whether he wishes to switch to the other remaining door. (Tijms, 213)

The ultimate question becomes should the contestant change doors at the end of the game or remain with his first choice. It initially appears irrelevant to change doors; there seems to be a 50% chance of getting the automobile at the end of the game. However, the game show host guarantees to display the contents of an incorrect door at the end of the game instead of a random door. This increases the probability of obtaining the car when switching doors to $\frac{2}{3}$. Consider the following chance tree for game (Tijms, 215):

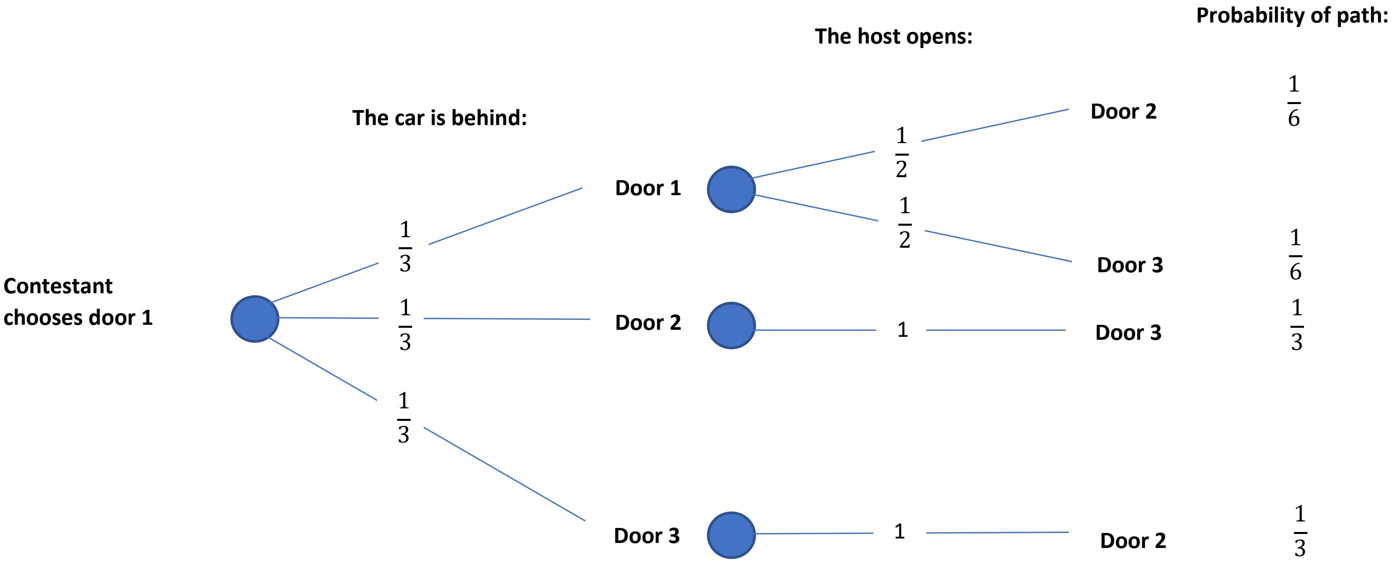

Figure 1: Chance tree for the Monty Hall dilemma

After building the tree, it is clear to see that the probability of obtaining the automobile given the contestant switches doors at the end of the game is two-thirds, and there is a one-third chance that he loses the car if he switches. When American columnist Marilyn vos Savant wrote the solution to the problem in her weekly column in the Sunday *Parade* magazine, she received thousands of letters in response (Tijms, 214). About ninety-percent of the letters, which included professional mathematicians from around the globe, debated the correctness of her analysis and favored the fifty-fifty probability at the end of the game. This shows that sometimes human instinct and calculation miss the full picture of the problem and is better left to automatic computation. Of course, the Monty Hall dilemma is relatively simple and only has four outcomes given a starting door. A clever analysist could figure out the correct probabilities but imagine when the outcomes grow exponentially with modern data sets, and authority figures expect the most accurate calculations under stress. Under such conditions, algorithms and computers must be employed.

II. The Effect of the Naïve Bayes Classifier on Spam Mail

An important data source that average people use daily is an email inbox. Since email is cheap and one of the fastest, most economical forms of communication available (Sahami), it has grown extremely popular for both everyday conversations and business use. Unfortunately, the rise in email popularity also influenced the rise in spammers' and direct marketers' sending junk mail (Sahami). Some of these unsolicited messages contain off-color material that is offensive to users and have potential to morally corrupt minors. Furthermore, junk mail wastes memory space and time and becomes a nuisance. The solution is to invent a system that classifies an email as either junk or legitimate and sorts it appropriately. One approach is to manually comb through an inbox, deleting any unwanted mail. Slightly better is to use commercial software that allows the user to hand-build a set of rules for the system to detect junk mail (Sahami). There are multiple flaws with this model, as it assumes people are technically-oriented and have the time to implement such a rule set. The best approach is to use a classifier to directly process the data in the inbox using probability calculus from training data. The abundance of today's training data warrants the implementation of the classifier algorithm over manual filtering.

The Naïve Bayes classifier transcends a basic text analysis of emails that often results in a strict class decision of either junk or legitimate. Although classification is the goal of any classifying algorithm, a Naïve Bayes approach incorporates various attributes of spam to generate a confidence range and utility model to avoid rigid binary decisions (Sahami). The utility model is derived as follows: with situational problems such as spam filtering, there are four options. An outcome can be a true positive, true negative, false positive, or false negative.

By probability, all four options are bound to occur some percentage of the time. The questions become value-based: is it more harmful to have a false positive or a false negative? Whatever a person may choose as their preference, one can then program an appropriate posterior probability threshold that classifies the email. Taking a false negative as a more harmful impact, we could observe the highest posterior probabilities which account for the high cost of a false negative and the moderate cost of a false positive. The analysis would lead to an optimal level of decisions for a spam filter (Sahami). The user would only have to quantify their utility preferences for implementation's sake, such as deeming a 99.9% posterior probability high enough for classifying mail as junk.

There are numerous attributes one could associate with spam: a lack of external attachments, an abundance of non-alphabetic characters in the subject field, or even bright flashy colors in the message itself. The relationship between an attribute and its class is modelled by a directed acyclic graph (Sahami):

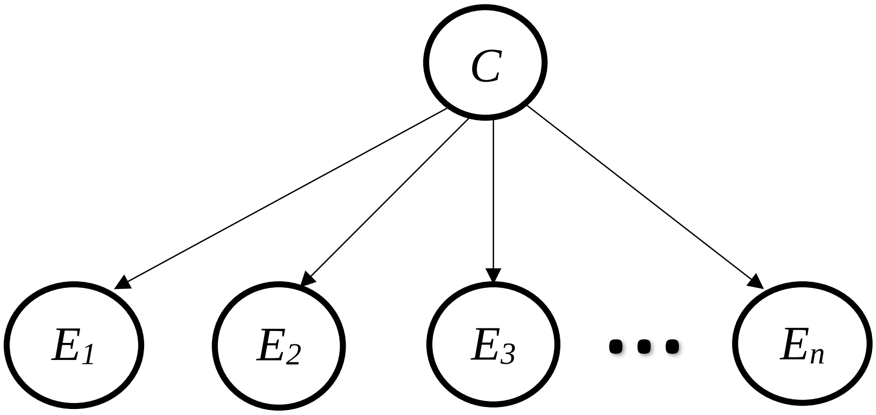

Figure 2: A DAG that models the relationship between class and attributes

This graph is called a "Bayesian network" (Sahami). "In such a graph, each random variable E_i is denoted by a node. A directed edge between two nodes indicates a probabilistic dependency from the parent node to that of the child" (Sahami). Therefore, the relationship is unidirectional: the probability of a class depends on its attributes. The Naïve Bayes algorithm applies this model to a classification problem, and to describe a probability distribution satisfying the dependency assumptions, "each node E_i in the network is associated with a conditional probability table, which specifies the distribution over *E_i* given any possible assignment of values to its parents" (Sahami). The probabilities in the table are determined from training data the analyst provides.

The graph is acyclic because it analyzes a unidirectional probability dependence from class to attribute and assumes that each E_i is independent. This assumption gives the name "naïve". For equation 1 to be multiplicative among evidences, we assume the attributes of the class are independent, such that

$$P(E = e \mid C = c_j) = \prod_i P(E_i = e_i \mid C = c_j). \tag{2}$$

Combining equations 1 and 2 give the fully generic form of Bayes' Theorem, assuming independence among evidences:

$$\mathrm{P}(C = c_j \mid E = e) = \frac{P(C = c_j) \prod_i P(E_i = e_i \mid C = c_j)}{\sum_{j=1}^{n} [P(C = c_j) \prod_i P(E_i = e_i \mid C = c_j)]}. \tag{3}$$

This is where there are criticisms of the algorithm. For example, there may be evidence of a dependent association between emails having flashy colors and having a plethora of non-alphabetic characters. Despite this large assumption, it turns out it is not the worst to make (Hand and Yu, 394-395). For one, "its intrinsic simplicity means low variance in its probability estimates" (Hand and Yu). Furthermore, the fact of handpicking attributes tends to reduce interdependence. Despite assuming independence, the algorithm works well and is beneficial for data like email where users can specify attributes, which decreases interdependence (Liu, et al). The next section will describe an alternative to assuming independence, but there are tradeoffs for creating more complicated classifiers. With a 99.9% threshold for classifying messages as junk, the following matrix was generated from Sahami's study involving a real-world email inbox using a Naïve Bayes classifier:

	Classified Junk	Classified Legitimate	Total
Actually Junk	36 (92.0% precision)	9	45
Actually Legitimate	3	174 (95.0% precision)	177
Total	39	183	222

Figure 3: Confusion matrix for Sahami's real usage scenario

As the chart demonstrates, the actual ratios of false negatives and false positives are satisfyingly low. Again, the Law of Large Numbers dictates that the probability that these percentages will deviate from their averages will go to zero, but not that the proportions themselves will go to zero (Devore & Berk, 304). There will always be some degree of variability.

The Naïve Bayes algorithm makes an impact on modern data classification because it is simple and reliable. Above all, it's a powerful classifier that allows automatic computation of probability ranges for various user defined attributes, which distinguishes it from other rigid, binary classifiers. The ability to define a probability threshold to control for false positives or negatives is also highly attractive. In addition, it is also easily expandable, for an analyst simply adds wanted attributes to their sample space, which expands the computation incrementally by a multiplication without re-examining the entire training data set. It is therefore clear that the algorithm has a positive influence on a user's experience with spam mail by significantly decreasing the time spent filtering mailboxes. The algorithm graciously circumvents the need for generating a strict rule set for data, which increases classification accuracy. Although there is a large assumption in independence, the negative impact of that assumption is hardly shown due to decrease variability in handpicked attributes.

III. The Effect on Business and Banking

Another application of the Naïve Bayes classifier is in the field of Business Intelligence. With the improvement of electronic commerce and high frequency trading, classifying algorithms are useful in recognizing consumer and firm behavior (Mishin, et al). An interesting industry that benefits from these algorithms is banking, particularly when predicting bank failures.

Current empirical evidence suggests that human auditors' decision processes often violate probability rules (Sarkar & Sriram, 1457). In addition, human auditors are prone to order and recency bias; the order of evidence or the recency of evidence presented may have an influence on an auditor's decision rather than the evidence itself. Since bank failures strain Federal Deposit Insurance Corporation's resources as well as undermine investor confidence, it is important to accurately predicting bank health with data (Sarkar & Sriram, 1457). If bank insolvency is detected early, regulators can intervene and help mitigate the negative effects failure has on average citizens.

Although institutional failures can ultimately be lead to managerial decisions, "such decisions are not easily observable at a point in time" (Sakar & Sriram, 1458). A way to classify banks is to track recent financial ratios as evidences for a healthy or an at-risk bank. Since auditors use probability measures to determine sufficiency of audit evidence, it is not incorrect to consider financial ratios for prediction in a Bayesian network (Sakar & Sriram, 1458). A mathematical model that represents the same uncertainty that auditors use is preferred because often auditors have difficulty in specifying consistent prior distributions in data, and auditors' predictions are often mis-calibrated (Sakar & Sriram, 1458). With proper training data, an algorithm could overcome an auditor's limitations and handle the data more precisely without bias.

The processes of building a Bayesian classifier in context does not digress too far from how an auditor decides the status of a firm. The auditor starts with an initial belief regarding a firm's status in the form of a prior probability distribution. The auditor then starts collecting various evidence for his hypothesis. The auditor then evaluates the posterior probability distributions after evaluating evidence and makes a judgement about the firm. To compensate for the variability in auditors' prior probability distributions, a computer could estimate such distributions by calculating the class of a bank given prior evidence. For example, "if there were ten historical cases with the same attribute values, and nine of them resulted in the outcome c_j, then the probability estimate for P($C = c_j \mid E = e$) would be 0.9" (Sakar & Sriram, 1461). Although the method seems to be computationally straight forward, imagine the difficulty in implementing the frequency tables when many predictive attributes are considered (the computation grows at a rate of $2^{\text{number of attributes}}$ because either the attribute belongs to the class, or not). This increases the need for more data, for the number of observed outcomes decreases as the number of considered attributes increases (Sakar & Sriram, 1461).

To avoid the large data requirement in analyzing outcomes, we again assume independence among the attributes. The reasoning is that "the attribute values that help predict the [class] are viewed as realizations that reflect the [class]. Therefore, if the [class] was known in advance, then the expectations of the values for each attribute should depend only on the [class], and not on other attribute values" (Sakar & Sriram, 1461). Although the assumption eases the way analysts work with the data, it is not always accurate in context. If we were to analyze financial ratios as an indicator of bank failure, there may be evident dependencies: The Return on Assets (ROA) and Return on Equity (ROE) both reflect the profitability of a bank and may be dependent variables (Sarkar & Sriram, 1462). To account for such dependency, we can calculate the mutual information of each attribute and repackage the attributes into clusters. "The induced probability model then consists of composite attributes corresponding to the final set of clusters" (Sarkar & Sriram, 1464). The mutual information measure is calculated as follows (Sarkar & Sriram, 1464): given class *C* and evidences *E_1...E_r* we have

$$\mathrm{I}(C;\ E_1;\ldots;E_r) = \sum_{C,\ E_1,\ldots,E_r} \mathrm{P}(C,E_1,\ldots,E_r)\ \mathrm{x}\log\frac{\mathrm{P}(C,E_1,\ldots,E_r)}{\mathrm{P}(C)P(E_1)\ldots\mathrm{P}(E_r)}. \tag{4}$$

The more dependence evidences *E_1…E_r* share, the higher the mutual information across the variables. When determining the appropriate clusters of evidences with highest mutual information, it is important to iteratively merge the clusters to avoid an exponential upper bound in the running time it takes for the algorithm to complete (Sarkar & Sriram, 1464).

Although clustering is a more correct approach than assuming independence, there are costs and benefits. Of course, more data is needed to support clustering, and there is a higher deficiency in the classifier when data is missing. Interestingly, studies have shown that the Naïve Bayes algorithm is the least sensitive to missing data because of its assumption in independence (Peng, et al). Sarkar and Sriram's study analyzed fourteen financial ratios across the years 1986-1988 for several banks and used both the naïve classifier and clustered classifier. When comparing the clustered Bayesian classifier to the naïve classifier, the naïve classifier seemed to discriminate more distinctly between healthy and distressed firms, whereas the clustered classifier approximated the true underlying uncertainty in the distributions perfectly (Sarkar & Sriram, 1472). The results show that both classifiers are valid and yield results that are reliable.

Using a Bayesian classifier algorithm (rather clustered or naïve) has important implications for the data banks provide to their auditors. The fact that these algorithms can provide reliable probabilities can help auditors derive posterior probabilities quicker and with more confidence for decision-making. These classifiers help firms make informed predictions about their client base and business model and help improve profit margins without bias (assuming correct training data). Although auditors may use other class evidence than financial ratios in their calculus, they can easily include such evidence in the Bayesian network to expand the model (Sarkar & Sriram, 1472). Bayesian classifier algorithms therefore have a positive influence in helping key decision-makers work with data by building accurate prediction models.

IV. Conclusion

The evolution of modern data repositories calls for automated ways to make predictions from data with a degree of uncertainty. Given the current expanding data sets, it is advantageous to use algorithms to correctly identify probability distributions in the data. Often human intuition has lack of foresight or is susceptible to biases, as shown in the Monty Hall dilemma as well as auditor behavior. Bayesian classifiers are intuitively straightforward and efficient algorithms for making inferences for the unknown class of target variables given evidences. This paper focused largely on the Naïve Bayes classifier algorithm, which assumes independence among attributes given a class to calculate posterior probabilities. Although the assumption is slightly incorrect, the mentioned studies noted the classifier’s strong ability to determine spam mail and bank health with a small fraction of misclassification. If attribute dependence was pivotal to incorporate in an analysis, one could iteratively cluster the data in according to the highest calculated mutual information among attributes. Although clustering identifies the underlying probability distribution most accurately, there is a cost of increased data requirements and needed computation. Also, the assumption of independence is not totally incorrect because it is noted that interdependence is decreased when attributes are handpicked, which is a common practice. In a world of endless data and uncertainty, valuable money and time is at stake. Customers want a faster, more pristine user experience, and businesses want reliable intelligence to improve profit margins. The Naïve Bayes classifier algorithm transcends the way humans interact with data to make predictions more accurate, meaningful, and reliable.

Works Cited

Devore, Jay L., and Kenneth N. Berk. “Modern Mathematical Statistics with Applications.” *Springer Science and Business Media, LLC*, 2012.

Hand, David J., and Keming Yu. “Idiot's Bayes—not so stupid after all?” *International Statistical Review* 69.3 (2001): 385-398.

Kenneth H. Rosen. 2002. “Discrete Mathematics and its Applications” (5th ed.). *McGraw-Hill Higher Education.*

Liu, Peng, Lei Lei, and Naijun Wu. “A quantitative study of the effect of missing data in classifiers.” *Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on*. IEEE, 2005.

Mishan, Mohd Taufik, et al. “An analysis on business intelligence predicting business profitability model using Naive Bayes neural network algorithm.” *System Engineering and Technology (ICSET), 2017 7th IEEE International Conference on*. IEEE, 2017.

Sahami, Mehran, et al. “A Bayesian approach to filtering junk e-mail.” *Learning for Text Categorization: Papers from the 1998 workshop*. Vol. 62. 1998.

Sarkar, Sumit, and Ram S. Sriram. “Bayesian models for early warning of bank failures.” *Management Science* 47.11 (2001): 1457-1475.

Tijms, Henk. “Understanding probability.” *Cambridge University Press*, 2012.