

Survey on Naive Bayes Algorithm

K.V.Lakshmi¹,N. Sowjanya Kumari²

^{1,2}Asst. Proff, Dept. of CSE,WISTM Engineering college(India)

ABSTRACT

Naive Bayes is a classification algorithm which is based on Bayes theorem with strong and naive independence assumptions. It simplifies learning by assuming that features are independent of given class. This surveys about naive Bayes algorithm, which describes its concept, hidden naive Bayes, text classification, traditional naive Bayes and machine learning. Also represents augmented naive Bayes by examples. And at the end some applications of naive Bayes and its advantages and disadvantages has discussed for a better understanding of the algorithm.

1.INTRODUCTION

Naïve Bayes is a subset of Bayesian decision theory. It's called naive because the formulation makes some naïve assumptions.It is a popular classification method in addition to conditional independence, overfitting, and Bayesian methods.

In the face of the simplicity of Naive Bayes, it can classify documents surprisingly well. Instinctively a potential justification for the conditional independence assumption is that if the document is about politics, this is a good evidence of the kinds of other words found in the document. Naive Bayes is a reasonable classifier in this sense and has minimal storage and fast training, it is applied to time-storage critical applications, such as automatically classifying web pages into types and spam filtering.

Considering a set of objects, each of which belongs to a known class, and each of which has a known vector of variables, the aim is to create a rule which enables to allocate future objects to a class, given just the vectors of variables marking out the future objects. These problems are known as supervised classification problems, are worldwide, and most of the methods for constructing such rules have been developed. It is very easy to establish, and no need any complicated repetitive parameter estimation schemes. This means it should be applied to huge data sets. It is easy to interpret, so unskilled users in classifier technology can make out the reason for it is making the classification it makes. Finally, it often does surprisingly well it should not be the best possible classifier in any particular application, but it can usually be relied on to be robust and to do well.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

In a classification problem, our hypothesis (h) may be the class to assign for a new data instance (d).One of the easiest ways of selecting the most probable hypothesis given the data that we have that we can use as our prior

knowledge about the problem. Bayes' Theorem provides a way that we can calculate the probability of a hypothesis given our prior knowledge.

Bayes' Theorem is stated as:

$$P(h/d) = (P(d/h) * P(h)) / p(d)$$

Where

- **P(h|d)** is the probability of hypothesis h given the data d. This is called the posterior probability.
- **P(d|h)** is the probability of data d given that the hypothesis h was true.
- **P(h)** is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h.
- **P(d)** is the probability of the data (regardless of the hypothesis).

You can see that we are interested in calculating the posterior probability of P(h|d) from the prior probability p(h) with P(D) and P(d|h).

After calculating the posterior probability for a number of different hypotheses, you can select the hypothesis with the highest probability. This is the maximum probable hypothesis and may formally be called the maximum a posteriori (MAP) hypothesis.

This can be written as:

$$MAP(h) = \max(P(h/d))$$

or

$$MAP(h) = \max((P(d/h) * P(h)) / P(d))$$

or

$$MAP(h) = \max(P(d/h) * P(h))$$

The P(d) is a normalizing term which allows us to calculate the probability. We can drop it when we are interested in the most probable hypothesis as it is constant and only used to normalize.

Back to classification, if we have an even number of instances in each class in our training data, then the probability of each class (e.g. P(h)) will be equal. Again, this would be a constant term in our equation and we could drop it so that we end up with:

$$MAP(h) = \max(P(d/h))$$

II. REPRESENTATION USED BY NAIVE BAYES MODELS

The representation for naive Bayes is probabilities. A list of probabilities are stored to file for a learned naive Bayes model. This includes:

1. Class Probabilities: The probabilities of each class in the training dataset.

2. Conditional Probabilities: The conditional probabilities of each input value given each class value.

III. LEARN A NAIVE BAYES MODEL FROM DATA

Learning a naive Bayes model from your training data is fast. Training is fast because only the probability of each class and the probability of each class given different input (x) values need to be calculated. No coefficients need to be fitted by optimization procedures.

IV. CALCULATING CLASS PROBABILITIES

The class probabilities are simply the frequency of instances that belong to each class divided by the total number of instances.

For example in a binary classification the probability of an instance belonging to class 1 would be calculated as:

$$P(\text{class}=1) = \text{count}(\text{class}=1) / (\text{count}(\text{class}=0) + \text{count}(\text{class}=1))$$

In the simplest case each class would have the probability of 0.5 or 50% for a binary classification problem with the same number of instances in each class.

V. CALCULATING CONDITIONAL PROBABILITIES

The conditional probabilities are the frequency of each attribute value for a given class value divided by the frequency of instances with that class value.

For example, if a “*weather*” attribute had the values “*sunny*” and “*rainy*” and the class attribute had the class values “*go-out*” and “*stay-home*”, then the conditional probabilities of each weather value for each class value could be calculated as:

- $P(\text{weather}=\text{sunny}|\text{class}=\text{go-out}) = \text{count}(\text{instances with weather}=\text{sunny and class}=\text{go-out}) / \text{count}(\text{instances with class}=\text{go-out})$
- $P(\text{weather}=\text{sunny}|\text{class}=\text{stay-home}) = \text{count}(\text{instances with weather}=\text{sunny and class}=\text{stay-home}) / \text{count}(\text{instances with class}=\text{stay-home})$
- $P(\text{weather}=\text{rainy}|\text{class}=\text{go-out}) = \text{count}(\text{instances with weather}=\text{rainy and class}=\text{go-out}) / \text{count}(\text{instances with class}=\text{go-out})$
- $P(\text{weather}=\text{rainy}|\text{class}=\text{stay-home}) = \text{count}(\text{instances with weather}=\text{rainy and class}=\text{stay-home}) / \text{count}(\text{instances with class}=\text{stay-home})$

VI. MAKE PREDICTIONS WITH A NAIVE BAYES MODEL

Given a naive Bayes model, you can make predictions for new data using Bayes theorem.

$$\text{MAP}(h) = \max(P(d|h) * P(h))$$

Using our example above, if we had a new instance with the *weather* of *sunny*, we can calculate:

$$\begin{aligned} \text{go-out} &= P(\text{weather}=\text{sunny}|\text{class}=\text{go-out}) * P(\text{class}=\text{go-out}) \\ \text{stay-home} &= P(\text{weather}=\text{sunny}|\text{class}=\text{stay-home}) * P(\text{class}=\text{stay-home}) \end{aligned}$$

We can choose the class that has the largest calculated value. We can turn these values into probabilities by normalizing them as follows:

$$P(\text{go-out}|\text{weather}=\text{sunny}) = \frac{\text{go-out}}{(\text{go-out} + \text{stay-home})}$$
$$P(\text{stay-home}|\text{weather}=\text{sunny}) = \frac{\text{stay-home}}{(\text{go-out} + \text{stay-home})}$$

If we had more input variables we could extend the above example. For example, pretend we have a “car” attribute with the values “working” and “broken“. We can multiply this probability into the equation.

For example below is the calculation for the “go-out” class label with the addition of the car input variable set to “working”:

$$\text{go-out} = P(\text{weather}=\text{sunny}|\text{class}=\text{go-out}) * P(\text{car}=\text{working}|\text{class}=\text{go-out}) * P(\text{class}=\text{go-out})$$

VII.GAUSSIAN NAIVE BAYES

Naive Bayes can be extended to real-valued attributes, most commonly by assuming a Gaussian distribution.

This extension of naive Bayes is called Gaussian Naive Bayes. Other functions can be used to estimate the distribution of the data, but the Gaussian (or Normal distribution) is the easiest to work with because you only need to estimate the mean and the standard deviation from your training data.

VIII.REPRESENTATION FOR GAUSSIAN NAIVE BAYES

Above, we calculated the probabilities for input values for each class using a frequency. With real-valued inputs, we can calculate the mean and standard deviation of input values (x) for each class to summarize the distribution.

This means that in addition to the probabilities for each class, we must also store the mean and standard deviations for each input variable for each class.

IX.LEARN A GAUSSIAN NAIVE BAYES MODEL FROM DATA

This is as simple as calculating the mean and standard deviation values of each input variable (x) for each class value.

$$\text{mean}(x) = 1/n * \text{sum}(x)$$

Where n is the number of instances and x are the values for an input variable in your training data.

We can calculate the standard deviation using the following equation:

$$\text{standard deviation}(x) = \text{sqrt}(1/n * \text{sum}(xi-\text{mean}(x))^2)$$

This is the square root of the average squared difference of each value of x from the mean value of x, where n is the number of instances, sqrt() is the square root function, sum() is the sum function, xi is a specific value of the x variable for the ith instance and mean(x) is described above, and ^2 is the square.

X.MAKE PREDICTIONS WITH A GAUSSIAN NAIVE BAYES MODEL

Probabilities of new x values are calculated using the Gaussian Probability Density Function (PDF).When making predictions these parameters can be plugged into the Gaussian PDF with a new input for the variable, and in return the Gaussian PDF will provide an estimate of the probability of that new input value for that class.

$$\text{pdf}(x, \text{mean}, \text{sd}) = (1 / (\text{sqrt}(2 * \text{PI}) * \text{sd})) * \text{exp}(-((x-\text{mean})^2)/(2*\text{sd}^2))$$

Where pdf(x) is the Gaussian PDF, sqrt() is the square root, mean and sd are the mean and standard deviation calculated above, PI is the numerical constant, exp() is the numerical constant e or Euler's number raised to power and x is the input value for the input variable.

We can then plug in the probabilities into the equation above to make predictions with real-valued inputs.

For example, adapting one of the above calculations with numerical values for weather and car:

$$\text{go-out} = P(\text{pdf(weather)}|\text{class=go-out}) * P(\text{pdf(car)}|\text{class=go-out}) * P(\text{class=go-out})$$

XI.HOW NAIVE BAYES ALGORITHM WORKS?

Let's understand it using an example. Below I have a training data set of weather and corresponding target variable 'Play' (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition. Let's follow the below steps to perform it.

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table			
Weather	No	Yes	
Overcast		4	=4/14 0.29
Rainy	3	2	=5/14 0.36
Sunny	2	3	=5/14 0.36
All	5	9	
	=5/14	=9/14	
	0.36	0.64	

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Problem: Players will play if weather is sunny. Is this statement is correct?

We can solve it using above discussed method of posterior probability.

$$P(\text{Yes} | \text{Sunny}) = P(\text{Sunny} | \text{Yes}) * P(\text{Yes}) / P(\text{Sunny})$$

Here we have $P(\text{Sunny} | \text{Yes}) = 3/9 = 0.33$, $P(\text{Sunny}) = 5/14 = 0.36$, $P(\text{Yes}) = 9/14 = 0.64$

Now, $P(\text{Yes} | \text{Sunny}) = 0.33 * 0.64 / 0.36 = 0.60$, which has higher probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

Naive Bayes Applications

The Naive Bayes algorithm is used in multiple real-life scenarios such as:

1. Text classification: It is used as a probabilistic learning method for text classification. The Naive Bayes classifier is one of the most successful known algorithms when it comes to the classification of text documents, i.e., whether a text document belongs to one or more categories (classes).

2. Spam filtration: It is an example of text classification. This has become a popular mechanism to distinguish spam email from legitimate email. Several modern email services implement Bayesian spam filtering. Many server-side email filters, such as DSPAM, Spam Bayes, Spam Assassin, Bog filter, and ASSP, use this technique.

3. Sentiment Analysis: It can be used to analyze the tone of tweets, comments, and reviews—whether they are negative, positive or neutral.

4. Recommendation System: The Naive Bayes algorithm in combination with collaborative filtering is used to build hybrid recommendation systems which help in predicting if a user would like a given resource or not.

Naive Bayes Advantages And Disadvantages

Advantages:

- It is easy and fast to predict class of test data set. It also perform well in multi class prediction
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- It performs well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

Disadvantages:

- One of the problems of Naïve Bayes is known as the "Zero Conditional Probability Problem." This problem wipes out all the information in other probabilities too. There are several sample correction techniques to fix this problem such as "Laplacian Correction.
- Another disadvantage is the very strong assumption of independence class features that it makes. It is near to impossible to find such data sets in real life.

S.NO	ALGORITHM	FEATURES	LIMITATIONS
1	CH 3 Algorithm	<ul style="list-style-type: none"> • Good. Suitable can be easily interpreted. • Easy to implement. • Can use both discrete and continuous values. • Data will store. 	<ul style="list-style-type: none"> • Small variation in data can lead to different decision trees. • Does not work very well on a small training data set. • Overfitting.
2	ID3 Algorithm	<ul style="list-style-type: none"> • It produces the more accurate result than the CH 3 Algorithm. • Decision tree is increase and space consumption is reduced. 	<ul style="list-style-type: none"> • Requires large working time. • Sometimes it may produce very long rules which are very hard to process. • Requires large amount of memory to store tree.
3	K-Nearest Neighbor Algorithm	<ul style="list-style-type: none"> • Classes used are the locally available. • Does not cut off the learning process. • Sometimes it is failure with regard to some training data. • Not suited for unbalanced classes. 	<ul style="list-style-type: none"> • Time to find the nearest neighbor is a large training data set can be enormous. • It is sensitive to noisy or irrelevant attributes. • Performance of algorithm depends on the number of instances used.
4	Naive Bayes Algorithm	<ul style="list-style-type: none"> • Simple to implement. • Used Computational efficiency and consistency rate. • It produces accurate results for most of the classification and prediction problems. 	<ul style="list-style-type: none"> • The precision of algorithm decreases if the amount of data is small. • For obtaining good results it requires a very large number of records.
5	Support Vector Machine Algorithm	<ul style="list-style-type: none"> • High accuracy. • Work well even if data is not linearly separable in the high feature space. 	<ul style="list-style-type: none"> • Speed and size requirements both in training and testing is more. • High computational and memory demands. requirements for classification is more space.
6	Artificial Neural Network Algorithm	<ul style="list-style-type: none"> • It is not to use with the parameters to adjust. • A neural network, inputs and outputting is not needed. • Easy to implement. • Applicable in a wide range of problems in real life. 	<ul style="list-style-type: none"> • Requires high processing time if neural network is large. • Difficult to know how many neurons and layers are necessary. • Learning can be slow.

XII.CONCLUSION

From the result of comparing all the algorithms we have observed that the time taken to build any model is less when using Naive Bayes and correctly classified instances are more. We applied this algorithm in data set using Gaussian Naïve and we did the prediction and accuracy is also greater in Naive Bayes Hence it is concluded that naïve Bayes performs well when compared to remaining algorithms.

REFERENCES

- [1] Liangxiao Jiang, Harry Zhang, and ZhihuaCai, —A Novel Bayes Model, Hidden Naive Bayes|| , IEEE Transaction on Knowledge and Data Engineering, Vol 21, No. 10, pp. 1361 – 1371
- [2] S. O. N. A. Taheri, "Learning the naive Bayes classifier with optimization models,|| vol. 23, no. 4, pp. 787–795, 2013.
- [3] J. Ren, S. D. Lee, X. Chen, B. Kao, R. Cheng, and D. Cheung, —Naive Bayes Classification of Uncertain Data,|| no. 60703110.
- [4] I. Rish, "An Empirical Study of the Naïve Bayes Classifier," no. January 2014.
- [5] W. Zhang and F. Gao, —Procedia Engineering An Improvement to Naive Bayes for Text Classification,|| vol. 15, pp. 2160–2164, 2011.
- [6] SiddharthBanga, SakshamMongia, VaibhavTiwari, Mrs. SunitaDhotre, —Regression and Augmentation Analytics on Earth’s Surface Temperature|| , IJCST, 2017.
- [7] Xindong Wu, Vipin Kumar, J. Ross Quinlan, JoydeepGhosh, Qiang Yang, Hiroshi Motoda, Geoffrey, J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, Dan Steinberg, —Top 10 algorithms in data mining|| , Springer-Verlag London, 2007
- [8] Arthur Chol, NazgolTavabi, Adnan Darwiche, —Structured Features in Naive Bayes Classification|| , Association for the Advancement of Artificial Intelligence, 2016.
- [9] Harry Zhang, —The Optimality of Naive Bayes|| , American Association for Artificial Intelligence, 2004.
- [10] ToonCalders, SiccoVerwer, —Three naive Bayes approaches for discrimination-free classification|| , Data Min Knowl Disk, 2010.