e-PGPathshala

Subject : Computer Science

Paper: Machine Learning

Module: Naïve Bayes Classification

Module No: CS/ML/30

Quadrant I – e-text

Welcome to the e-PG Pathshala Lecture Series on Machine Learning. In this module we will discuss a fairly simple but powerful Classification based on Bayes Theorem – Naïve Bayes.

Learning Objectives:

The learning objectives of this module are as follows:

- To understand Probabilistic Classification
- To explain Naïve Bayes Classifier
- To understand some applications of Naïve Bayes Classifier

30.1 Classification problem

Let us again look at the classification problem, a supervised learning problem. The training data consists of examples of the form (d,h(d)) where d is the data object to classify (inputs) and h(d) is the correct class information for d. The goal of classification is to find the class $h(d_{new})$ of $d_{new,}$ an hitherto unseen data object (Figure 30.1).

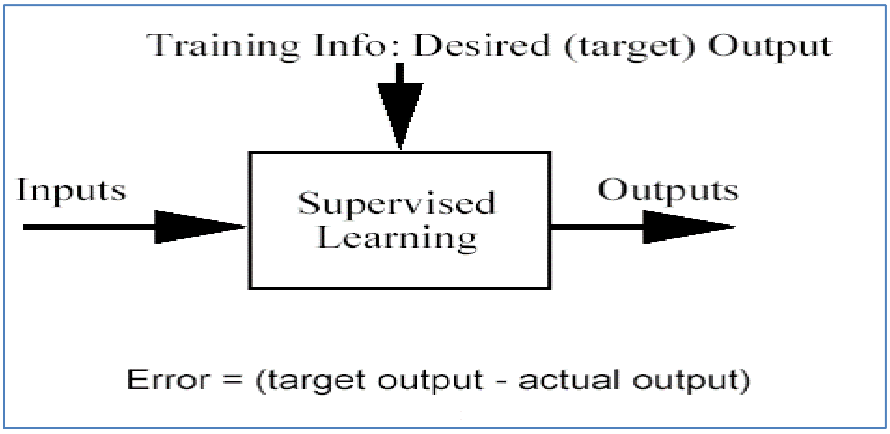

Training Info: Desired (target) Output
Inputs
Supervised Learning
Outputs
Error = (target output - actual output)

Figure 30.1 Supervised Learning

30.1.1 Methods to Create Classifiers

There are three methods to establish a classifier as given below:

a) **Model a classification rule directly -** Examples: k-NN, decision trees, perceptron, SVM
b) **Model the probability of class memberships given input data -** Example: perceptron with the cross-entropy cost
c) **Make a probabilistic model of data within each class -** Examples: **naive Bayes**, model based classifiers

Of these methods a) and b) are examples of discriminative classification and *c*) is an example of generative classification. Moreover *b*) and *c*) are both examples of probabilistic classification.

30.2 Probabilistic Classification

Generally, in classification the goal is to predict the value of a class c given the value of a input feature vector x. From a probabilistic perspective, the goal is to find the conditional distribution p(c|x).There are two types of probabilistic models of classification, the discriminative model and the generative model.

30.2.1 Discriminative model

The most common approach to probabilistic classification is to represent the conditional distribution using a parametric model, and then to determine the parameters using a training set consisting of pairs <x_n, c_n> of input vectors along with their corresponding target output vectors. In other words discriminative classifiers model the posterior p(c|x) directly which can then be used to make predictions of c for new values of x (Figure 30.2). Here for example the classes can be C_1=benign mole or C_2 = cancer which can be modelled given the respective data points. This is known as a discriminative approach, since the conditional distribution discriminates directly between the different values of c.

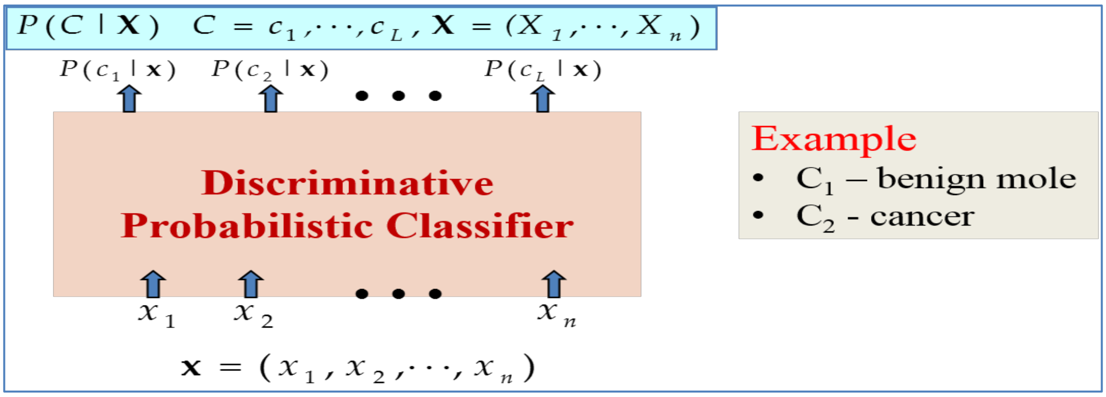

$P(C \mid \mathbf{X}) \quad C = c_1, \cdots, c_L, \mathbf{X} = (X_1, \cdots, X_n)$
$P(c_1 \mid \mathbf{x})$
$P(c_2 \mid \mathbf{x})$
$P(c_L \mid \mathbf{x})$
Discriminative Probabilistic Classifier
x_1
x_2
x_n
$\mathbf{x} = (x_1, x_2, \cdots, x_n)$
Example
• C_1 – benign mole
• C_2 - cancer

Figure 30.2 Discriminative Model

30.2.2 Generative model

The generative model is a model for randomly generating observable data values, typically given some hidden parameters. It specifies a joint probability distribution over observation and label sequences. Generative models are used for modelling data directly that is modelling observations drawn from a probability density function). This approach to probabilistic classification finds the joint distribution p(x, c), expressed for instance as a parametric model, and then subsequently use this joint distribution to evaluate the conditional p(c|x) in order to make predictions of c for new values of x by application of Bayes theorem. This is known as a generative approach since by sampling from the joint distribution it is possible to generate synthetic examples of the feature vector x (Figure 30.3). Here for the vector of random variables we learn the probability for each class c given the input. In practice, the generalization performance of generative models is often found to be poorer than that of discriminative models due to differences between the model and the true distribution of the data.

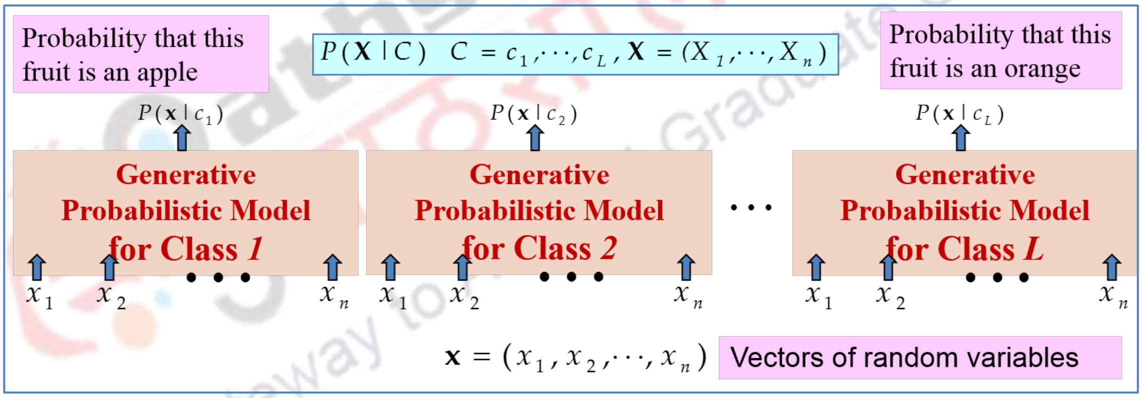

Probability that this fruit is an apple
$P(\mathbf{X}|C) \quad C = c_1,\cdots,c_L, \mathbf{X} = (X_1,\cdots,X_n)$
Probability that this fruit is an orange
$P(\mathbf{x}|c_1)$
$P(\mathbf{x}|c_2)$
$P(\mathbf{x}|c_L)$
Generative Probabilistic Model for Class 1
Generative Probabilistic Model for Class 2
Generative Probabilistic Model for Class L
x_1 x_2 • • • x_n
x_1 x_2 • • • x_n
x_1 x_2 • • • x_n
• • •
$\mathbf{x} = (x_1, x_2,\cdots, x_n)$
Vectors of random variables

Figure 30.3 Generative Model

30.3 Naïve Bayes

Naïve Bayes is a type of generative model. Here we try to determine the conditional probability P(C|X) that is the probability of a class given a set or bag of features. This probability can be determined by finding the likelihood of the input features given the class and the prior probability of the class. Here the difficulty lies in learning the joint probability of the features given the class.

$$P(C|\mathbf{X}) \propto P(\mathbf{X}|C)P(C) = P(X_1,\cdots,X_n|C)P(C)$$

An important assumption that Naïve Bayes method makes is that **all input attributes are conditionally independent.** In other words assume that the joint probability can be found by finding the individual probability of each feature given the class (Figure 30.4).

$$
\begin{aligned}
P(X_1, X_2, \cdots, X_n \mid C) &= P(X_1 \mid X_2, \cdots, X_n, C) P(X_2, \cdots, X_n \mid C) \\
&= P(X_1 \mid C) P(X_2, \cdots, X_n \mid C) \\
&= P(X_1 \mid C) P(X_2 \mid C) \cdots P(X_n \mid C)
\end{aligned}
$$

$$\mathbf{x} = (x_1, x_2, \cdots, x_n)$$

Product of individual probabilities

Figure 30.4 Naïve Bayes Assumption

This is an important special, simplification of a Bayes optimal classifier, where the hypothesis is the classification and all attributes are conditionally independent given the class (Figure 30.5)

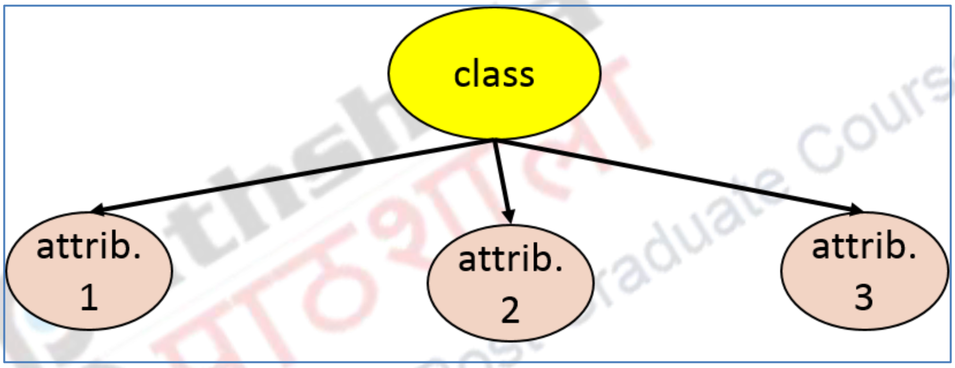


Figure 30.5 Independence of Attributes

The setting for the Naïve Bayes classifier is where a set of training examples is provided, and a new instance is presented, described by the tuple of attribute values ***(a$_l$, a$_2$ ...a$_n$)***. The job of the learner is to predict the target value (classification), for this new instance.

Naïve Bayes is one of the most practical Bayes learning methods. The naive Bayes classifier applies to learning tasks where each instance *x* is described by a conjunction of attribute values and where the target function f (x) can take on any value from some finite set V. Naïve Bayes method is generally used when there is a moderate or large training set available and when attributes that describe instances are conditionally independent given the classification. Some successful applications of Naïve Bayes method are medical diagnosis and classifying text documents.

30.3 Naïve Bayes Classifier: Assumptions

As already discussed for the Naïve Bayes classifier we need to determine two probabilities that is prior probability of the class and the likelihood of the data with respect to the class.

The prior probability of the class $\boldsymbol{P(c_j)}$ **c**an be estimated from the frequency of classes in the training examples.

However the likelihood or the joint probability of the features of the input data given the class $\boldsymbol{P(x_1,x_2,\ldots,x_n|c_j)}$ is of the order of $O(|X|^n \bullet |C|)$ and can only be estimated if a very, very large number of training examples was available. This is where the naivety of Naïve Bayes method comes into the picture that is the **conditional independence assumption** where we assume that the probability of observing the conjunction of attributes is equal to the product of the individual probabilities.

Let us first formulate Bayesian Learning. Let each instance *x* of a training set *D* be a conjunction of *n* attribute values $<a_1,a_2,..,a_n>$ and let *f(x),* the target function, be such that $f(x) \in V$, a finite set. According to the Bayesian approach using Maximum A Posteriori (MAP) we can specify this as follows:

$$\begin{aligned} v_{MAP} &= \operatorname{argmax}_{v_j \in V} P(v_j|a_1,a_2,..,a_n) \\ &= \operatorname{argmax}_{v_j \in V} [P(a_1,a_2,..,a_n|v_j)\, P(v_j)/P(a_1,a_2,..,a_n)] \\ &= \operatorname{argmax}_{v_j \in V} [P(a_1,a_2,..,a_n|v_j)\, P(v_j)] \end{aligned}$$

Here we try to find a value $\boldsymbol{v_j}$ that maximizes the probability given the attribute values. Applying Bayes Theorem this is finding the $\boldsymbol{v_j}$ that maximizes the product of the likelihood $\boldsymbol{P(a_1,a_2,..,a_n|v_j)}$ and the prior probability $\boldsymbol{P(v_j)}$.

Now according to **Naïve Bayesian Approach, w**e assume that the attribute values are conditionally independent so that $\boldsymbol{P(a_1,a_2,..,a_n|v_j) = \prod_i P(a_1|v_j)}$. This means that too large a data set is not required.

Therefore ***<u>Naïve Bayes Classifier:</u>*** $\boldsymbol{v_{NB} = \operatorname{argmax}_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)}$

Naïve assumption of attribute independence for a set of k features is as given below:

$$P(x_1,\ldots,x_k|C) = P(x_1|C)\cdot\ldots\cdot P(x_k|C)$$

Now if the If i-th attribute is categorical then $P(x_i|C)$ is estimated as the relative freq of samples having value x_i. However if the i-th attribute is continuous then $P(x_i|C)$ is estimated through a Gaussian density function. It is computationally easy in both cases.

30.5 Naïve Bayes Algorithm

Naïve Bayes Algorithm (considering discrete input attributes) has two phases

1. **Learning Phase**: Given a training set **S**,

Learning is easy, just create probability tables.

```
For each target value of $c_i$ $(c_i = c_1, \cdots, c_L)$
    $\hat{P}(C = c_i) \leftarrow$ estimate $P(C = c_i)$ with examples in $\mathbf{S}$;
    For every attribute value $x_{jk}$ of each attribute $X_j$ $(j = 1, \cdots, n; k = 1, \cdots, N_j)$
        $\hat{P}(X_j = x_{jk} | C = c_i) \leftarrow$ estimate $P(X_j = x_{jk} | C = c_i)$ with examples in $\mathbf{S}$;
```

2. **Test Phase**: Given an unknown instance X_j, N_j X L

Look up tables to assign the label *c** to **X'** if X' = (a_1', a_2', ……a_n')

Classification is easy, just multiply probabilities

$$[\hat{P}(a_1' | c^*) \cdots \hat{P}(a_n' | c^*)]\hat{P}(c^*) > [\hat{P}(a_1' | c) \cdots \hat{P}(a_n' | c)]\hat{P}(c), \quad c \neq c^*, c = c_1, \cdots, c_L$$

Naive Bayes Classifier

In general we assume that target function is *f: X* ➔ *V*, where each instance *x* is described by attributes $a_1, a_2 .. a_n$

The most probable value of *f(x)* is:

$$v_{MAP} = \underset{v_j \in V}{\arg\max} P(v_j | a_1, a_2 \ldots a_n)$$

$$v_{MAP} = \underset{v_j \in V}{\arg\max} \frac{P(a_1, a_2 \ldots a_n | v_j) P(v_j)}{P(a_1, a_2 \ldots a_n)}$$

$$= \underset{v_j \in V}{\arg\max} P(a_1, a_2 \ldots a_n | v_j) P(v_j)$$

In the case of Naive Bayes Algorithm

For each target value V_j

$\hat{P}(v_j)$ ← estimate $P(v_j)$

For each attribute value a_i of each attribute a

$$\hat{P}(a_i|v_j) \leftarrow \text{estimate } P(a_i|v_j)$$

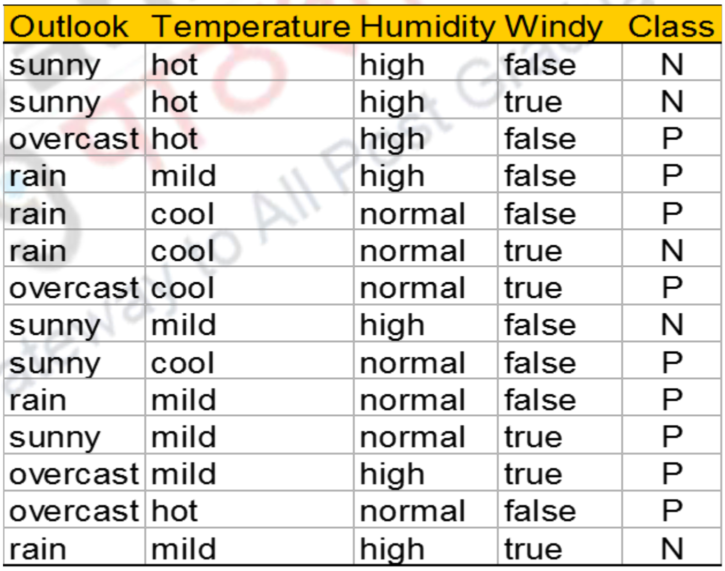
$$v_{NB} = \arg\max_{v_j \in V} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i|v_j)$$

Here we use the Naive Bayes assumption:

$$P(a_1, a_2 \ldots a_n|v_j) = \prod_i P(a_i|v_j)$$

which gives

$$\text{Naive Bayes classifier: } v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$$



Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Figure 30.6 Playing Tennis Example

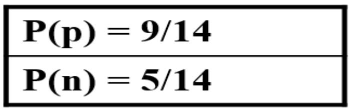
30.6 Characteristics of Naïve Bayes

Therefore Naïve Bayes classifier only requires the estimation of the prior probabilities $P(C_K)$ (where k is the given number of classes), and p (where p is the number of attributes) conditional probabilities for each class. We will be able to answer full set of queries across classes and features. Empirical evidence shows that Naïve Bayes classifiers work remarkable well. It has been that the use of a more complex full Bayes (belief) network provides only limited improvements in classification performance.

30.7 Example 1 – Playing Tennis

Given a training set (Figure 30.6), we can compute probabilities from the data.

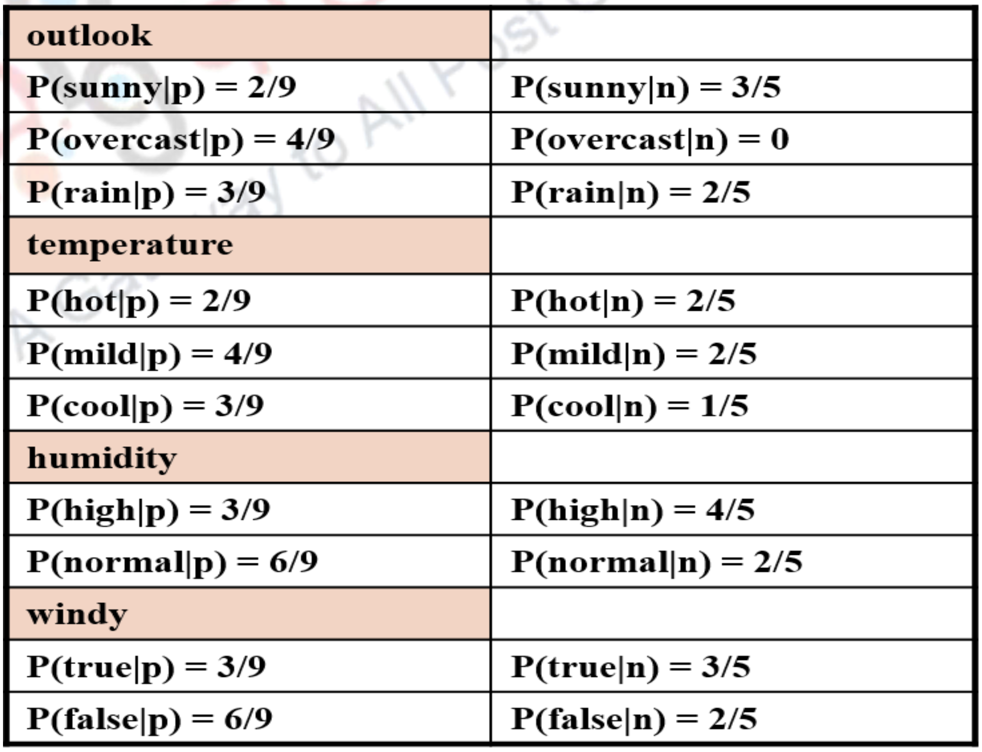
Training Phase: First we compute the probabilities of Playing tennis (positive) and not Playing Tennis (negative) as P(p) and P(n) respectively (Figure 30.7).



P(p) = 9/14
P(n) = 5/14

Figure 30.7 Probability of Playing tennis and not Playing Tennis

Probabilities of each of the attributes is then calculated as given below:
We need to estimate P(x_i|C) where x_i is each value for each attribute given C which is either positive (p) or negative (n) class for Play Tennis. For example consider the attribute Outlook has value sunny for 2 of the 9 positive samples (2/9) and for 3 of the 5 negative samples (3/5). We calculate such probabilities for every other attribute value for attribute Outlook and similarly for each value of each of the other attributes Temperature, Humidity and Windy (Figure 30.8)

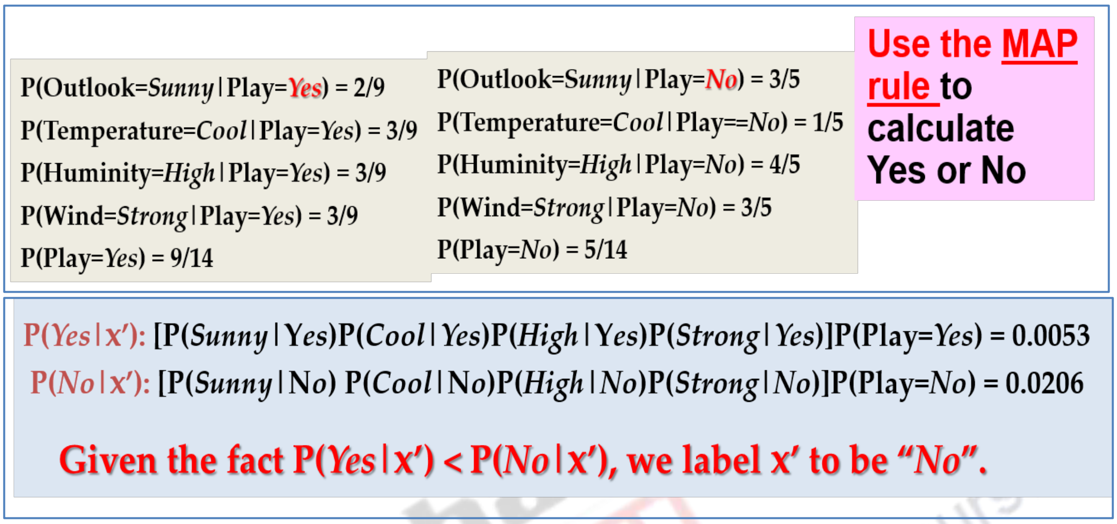


outlook	
P(sunny|p) = 2/9	P(sunny|n) = 3/5
P(overcast|p) = 4/9	P(overcast|n) = 0
P(rain|p) = 3/9	P(rain|n) = 2/5
temperature	
P(hot|p) = 2/9	P(hot|n) = 2/5
P(mild|p) = 4/9	P(mild|n) = 2/5
P(cool|p) = 3/9	P(cool|n) = 1/5
humidity	
P(high|p) = 3/9	P(high|n) = 4/5
P(normal|p) = 6/9	P(normal|n) = 2/5
windy	
P(true|p) = 3/9	P(true|n) = 3/5
P(false|p) = 6/9	P(false|n) = 2/5

Figure 30.8 Conditional Probabilities of each Attribute

Test Phase: Given a new instance **x'** of variable values, we need to calculate the probability of either Playing Tennis or not Playing Tennis. Now assume we are given the values of the four attributes as shown below:

x'=(Outlook=*Sunny,* Temperature=*Cool,* Humidity=*High,* Wind=*Strong*)



P(Outlook=*Sunny*|Play=*Yes*) = 2/9
P(Temperature=*Cool*|Play=*Yes*) = 3/9
P(Huminity=*High*|Play=*Yes*) = 3/9
P(Wind=*Strong*|Play=*Yes*) = 3/9
P(Play=*Yes*) = 9/14

P(Outlook=*Sunny*|Play=*No*) = 3/5
P(Temperature=*Cool*|Play==*No*) = 1/5
P(Huminity=*High*|Play=*No*) = 4/5
P(Wind=*Strong*|Play=*No*) = 3/5
P(Play=*No*) = 5/14

Use the MAP rule to calculate Yes or No

P(*Yes*|x'): [P(*Sunny*|*Yes*)P(*Cool*|*Yes*)P(*High*|*Yes*)P(*Strong*|*Yes*)]P(Play=*Yes*) = 0.0053
P(*No*|x'): [P(*Sunny*|No) P(*Cool*|No)P(*High*|*No*)P(*Strong*|*No*)]P(Play=*No*) = 0.0206

Given the fact P(*Yes*|x') < P(*No*|x'), we label x' to be "*No*".

Figure 30.9 Calculating whether Tennis will be played given the new instance

Here we consider the probability of Outlook being Sunny and look up the probability of Play Tennis and not Play tennis. Similarly we look up the probabilities for Temperature =Cool, Humidity = High and Wind = Strong. Now we can calculate the probability of Playing tennis and not Playing Tennis given the new instance **x'** by finding the product of each of the probabilities obtained for the value of each variable. Now we find that P(Yes/ x') = 0.0053 and P(No/ x') = 0.0206. Since P(No/ x') is greater we can conclude that the new instance **x'** will be labelled as No.

30.8 Example 2 – Text Classification

We learn from examples which articles are of interest. The attributes in this case are the words. Since we are talking about the Naïve Bayes model, this just means that we have a random sequence model within each class. It has been found that NB classifiers are one of the most effective classifiers for this task.

30.8.1 Definition of Text Classification

First let us define the Text Classification problem. Here a document d is a data point in multi-dimensional space X, whose dimension is based on the number of unique words in the corpus. C is the set of categories we want the documents to be classified under. D is the training set of labelled documents <d,c>. The learning algorithm needs to find a mapping γ that X → C (Figure 30.10).

$d \in \mathrm{X}$ the document d in the multi-dimensional space X

$\mathbf{C} = \{c_1, c_2, \ldots, c_J\}$ a set of classes (categories, or labels)

D the training set of labeled documents $\langle d, c \rangle$

$\langle d, c \rangle \in \mathrm{X} \times \mathbf{C}$

Target function: $\gamma : \mathrm{X} \rightarrow \mathbf{C}$

Learning algorithm: $\Gamma(\mathrm{D}) = \gamma$

$\langle d, c \rangle = \langle$"Beijing joins the World Trade Organization", China$\rangle$

$\gamma(d) = c$ $\gamma(d) =$ China

Figure 30.10 Definition of Text Classification

30.8.2 Uses of Text Classification

Text classification is the task of classifying text documents to multiple classes. This has many applications as listed below:

- Is this mail spam?
- Is this article from comp.ai or misc.piano?
- Is this article likely to be relevant to user X?
- Is this page likely to lead me to pages relevant to my topic? (as in topic-specific crawling), etc.

30.8.3 Naïve Bayes Classifier and Text Classification

30.8.3.1 Document Representation

Naïve Bayes Classifier has been applied widely for text classification tasks. The question to be answered is how to represent text documents as feature vectors? The vector space representation is one where each document is represented by a n dimensional vector where n is the number of unique words (after removing all stop words) of all the documents in the document corpus (the collection unique words is called vocabulary).There are many vector space variants. The vector space model that is widely used is the one where each document is represented in each dimension by the frequency of occurrences of the words. One variant of the vector representation is the binary version each dimension is 1 if the word is present in the document, 0 otherwise. A problem with vector space representation is that the vectors are likely to be as large as the size of the vocabulary. Then "feature selection" techniques are used to select only a subset of words as features. One simple feature selection method is to limit the vocabulary by having words with frequency of occurrence that are above a threshold. In another variation of the vector space model we have the unigram model where the document is represented as a vector of positions with values being the words.

30.8.3.2 Naïve Bayes Algorithm –Testing Phase

During the testing phase, given a test document X *where* n *is* the number of word occurrences in X, *we need to r*eturn the category of the document. Here we find the category C using Maximum Likelihood that is the C that gives maximum probability with product of likelihood of n attributes given Ci and the prior probability of Ci. Here we assume that a_j is the word occurring at the the jth position in X. The probabilities of each of the attributes is assumed to be independent according to the Naïve Bayes assumption.

$$\underset{c_i \in C}{\operatorname{argmax}} \quad P(c_i) \prod_{j=1}^{n} P(a_j \mid c_i)$$

30.8.3.3 Text Naïve Bayes Algorithm – Training Phase

Let V be the vocabulary of all words in the documents in D. For each category $c_i \in C$. Let D_i be the subset of documents in D in category c_i

The prior Probability is calculated as follows $P(c_i) = |D_i| / |D|$

Let T_i be the concatenation of all the documents in $D_{i.}$ Let n_i be the total number of word occurrences in $T_{i.}$

For The Likelihood for each word $a_j \in V$ *is calculated as follows:*

Let n_{ij} be the number of occurrences of a_j in T_i

Let $P(a_i \mid c_i) = (n_{ij} + 1) / (n_i + |V|)$

30.9 Learning the Model

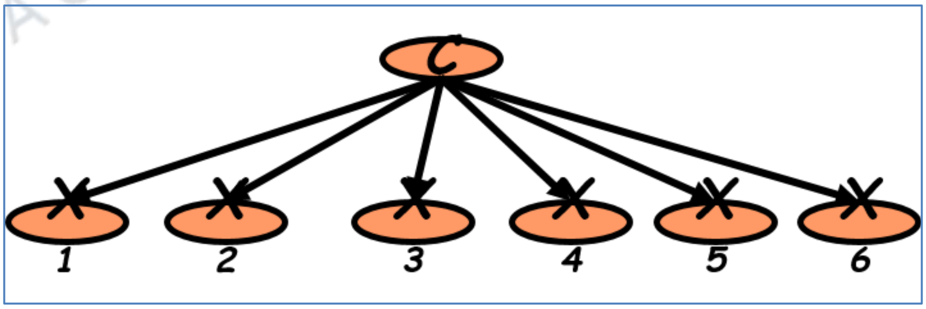


Figure 30.11 Naïve Bayes Model

Let us assume that the Naïve Bayes Model is represented as in Figure 30.11 where C is the class which is dependent on 6 attributes. The common practice used to find the maximum likelihood is to simply use the frequencies in the data. The two terms of Maximum Likelihood, prior probability and likelihood are calculated as given below. We explain with document classification as example. The prior probability P(c$_j$) of a class c$_j$ is calculated as the number of documents in the labelled corpus with category C=c$_j$ divided by the total number of documents. The likelihood of the each attribute or feature (word) xi given class c$_j$ is calculated as the number of documents of class c$_j$ having that attribute divided by the total number of documents of class c$_{j.}$ We find the posterior probability for each class in this way.

$$\hat{P}(c_j) = \frac{N(C = c_j)}{N}$$

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j)}{N(C = c_j)}$$

Figure 30.12 Calculating Prior Probability and Likelihood

In order to reemphasize the naïve Bayes assumption, we use the example of classification of Flu given the 5 symptoms runnynose, sinus, cough, fever, muscle-ache (Figure 30.13). We do not find the probability of all symptoms occurring together but assume that the symptoms are independent and hence calculate the likelihood of each symptom given the class independently.

$$P(X_1, \ldots, X_5 \mid C) = P(X_1 \mid C) \cdot P(X_2 \mid C) \cdot \cdots \cdot P(X_5 \mid C)$$

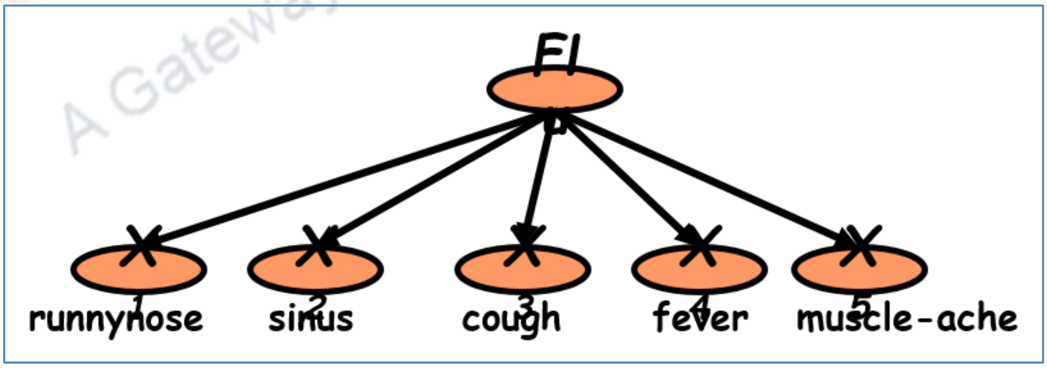


Figure 30.13 Flu and its Symptoms

$$\hat{P}(X_5 = t \mid C = nf) = \frac{N(X_5 = t, C = nf)}{N(C = nf)} = 0$$

Now the issue is what if we have seen no training cases where patient had no flu and muscle aches? In other words certain cases are zero because these cases not available in training set. Zero probabilities cannot be conditioned away, no matter the other evidence therefore we need to do some method of smoothing. We will explain this in the next section.

30.9.1 Naive Bayes - to Classify Text: Basic method

Let us assume that the documents are represented by attributes that are text positions, where the values are words. This is as given below.

$$c_{NB} - \operatorname*{argmax}_{c_j \in C} P(c_j) \prod_i P(x_i \mid c_j)$$

$$- \operatorname*{argmax}_{c_j \in C} P(c_j) P(x_1 - \text{"our"} \mid c_j) \cdots P(x_n - \text{"text"} \mid c_j)$$

However there are too many possibilities. Now let us assume that the classification is *independent* of the positions of the words, now we need to use some parameters for each position and hence we come to what is called the “Bag of words” model.

30.10 Example 3 –Text Classification

Let us take a toy example of text classification to explain Naïve Bayes method. Let us assume we are given N=4 documents which are classified into two classes: “China”, “not China”. We are also know the vocabulary V = {Beijing, Chinese, Japan, Macao, Tokyo}. We need to classify the test document given.

	docID		c = China?
Training set	1	Chinese Beijing Chinese	Yes
	2	Chinese Chinese Shangai	Yes
	3	Chinese Macao	Yes
	4	Tokyo Japan Chinese	No
Test set	5	Chinese Chinese Chinese Tokyo Japan	?

Figure 30.14 Example for Text Classification

The prior probability of class China and not China is given below:

$$\hat{P}(c) = 3/4 \quad \hat{P}(\overline{c}) = 1/4$$

We need nonzero probabilities for all words, even for words that don't exist. So we just count every word one time more than it actually occurs (a way of smoothing). Since we are only concerned with relative probabilities, this inaccuracy should be of no concern. Therefore the way to calculate likelihood for the example is as follows (Figure 30.15):

$$P(word|C) = \frac{count(word|C) + 1}{count(C) + V}$$

(*V* is the total vocabulary, so that our probabilities sum to 1.)

Figure 30.15 Calculating Likelihood

Now we need to carry out estimation to find the likelihood and use this to do the classification of the test document. We find the likelihood of each word (Chinese, Tokyo, Japan) in the test document for class Chinese and class not Chinese. Then we find the posterior probability of class Chinese and class not Chinese as given in Figure 30.16. We find that the probability for Class Chinese is higher and so that is the class of the test document.

Estimation

$$\hat{P}(\text{Chinese}|c) = (5+1)/(8+6) = 3/7$$

$$\hat{P}(\text{Tokyo}\,|\,c) = \hat{P}(\text{Japan}\,|\,c) = (0+1)/(8+6) = 1/14$$

$$\hat{P}(\text{Chinese}\;|\,\bar{c}) = (1+1)/(3+6) = 2/9$$

$$\hat{P}(\text{Tokyo}|\,c) = \hat{P}(\text{Japan}|\,c) = (1+1)/(3+6) = 2/9$$

Classification

$$P(c\,|\,d) \propto P(c) \prod_{1 \le k \le n_d} P(t_k\,|\,c)$$

$$P(c\,|\,d_5) \propto 3/4 \cdot (3/7)^3 \cdot 1/14 \cdot 1/14 \approx 0.0003$$

$$P(\bar{c}\,|\,d_5) \propto 1/4 \cdot (2/9)^3 \cdot 2/9 \cdot 2/9 \approx 0.0001$$

Figure 30.16 Estimation and Classification for the Example

30.11 Naïve Bayes Classifier - Comments

The foremost reason for the wide use of Naïve Bayes classifier is that it is easy to implement. The major advantage of Naïve Bayes classifier is that it has good learning speed as well as classification speed. It has modest space storage requirements. In addition incrementality is supported. Recommendations can re-done as more attribute values of the new item become known. Another advantage is that it seems to work very well in many scenarios. Naïve Bayes Classifier has much wider range of applicability than previously thought, this despite using the independence assumption.

However classification accuracy is different from probability estimate accuracy. We make the assumption of class conditional independence and therefore there is loss of accuracy because practically, dependencies exist among variables. As an example in hospitals: we can have patients who have profiles based on variables such as age, family history etc , who are associated with symptoms: such as fever, cough etc., and can have diseases such as lung cancer, diabetes etc and some of these variables are not independent. Dependencies among these variables cannot be modeled by Naïve Bayesian Classifier, We will tackle these dependencies later using Bayesian Belief Networks.

30.12 Issues Relevant to Naïve Bayes

1. **Violation of Independence Assumption**

For many real world tasks, events are correlated, but nevertheless, naïve Bayes works surprisingly well anyway.

$$P(X_1,\cdots,X_n|C) \neq P(X_1|C)\cdots P(X_n|C)$$

2. **Zero conditional probability Problem**

Such problem exists when no example contains the attribute value

$$\hat{P}(x_1|c_i)\cdots\hat{P}(a_{jk}|c_i)\cdots\hat{P}(x_n|c_i) = 0$$

In this circumstance,

$$P(X_1,\cdots,X_n|C) \neq P(X_1|C)\cdots P(X_n|C)$$

during test. One solution to the issue is to estimate conditional probabilities with virtual examples. We have estimated probabilities by the fraction of times the event is observed to n_c occur over the total number of opportunities *n. However* this provides poor estimates when n_c is very small. What happens if none of the training instances with target value v_j have attribute value a_i? In other words n_c is 0.

When n_c is very small we calculate likelihood as follows :

$$\hat{P}(a_i|v_j) = \frac{n_c + mp}{n + m}$$

Here n is number of training examples for which $v=v_j$, n_c number of examples for which $v=v_j$ and $a=a_i$, *p* is **prior** estimate and *m* is weight given to prior (i.e. number of ``virtual'' examples) .

$$v_{NB} =_{v_j \in V} P(v_j)\prod_i \hat{P}(a_i|v_j)$$

Summary

- Explained the concept of Probabilistic Classification
- Described Naïve Bayes Classifier
- Discussed some applications of Naïve Bayes Classifier