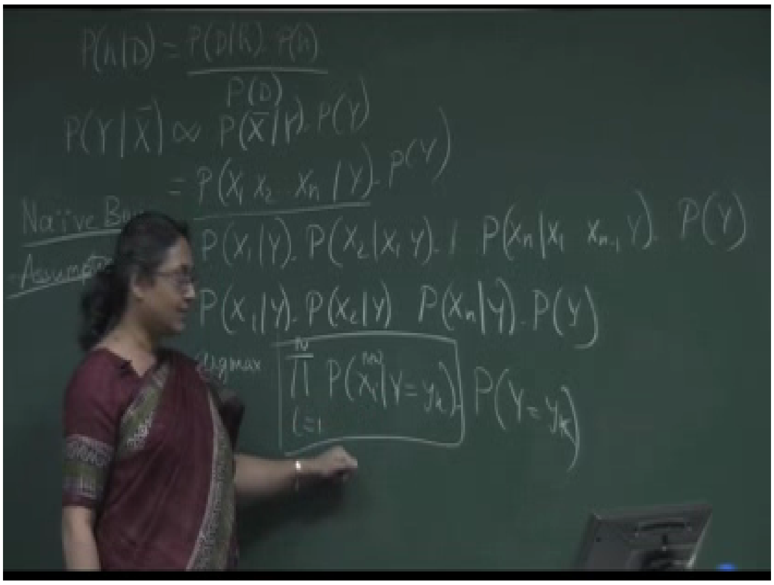
Introduction to Machine Learning

Prof. Sudeshna Sarkar
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 17
Naive Bayes

Good morning. Today, we will talk about part c of the module on Bayesian learning. Today's topic is Naive Bayes. In the last class, we looked at the Bayes theorem.

(Refer Slide Time: 00:31)



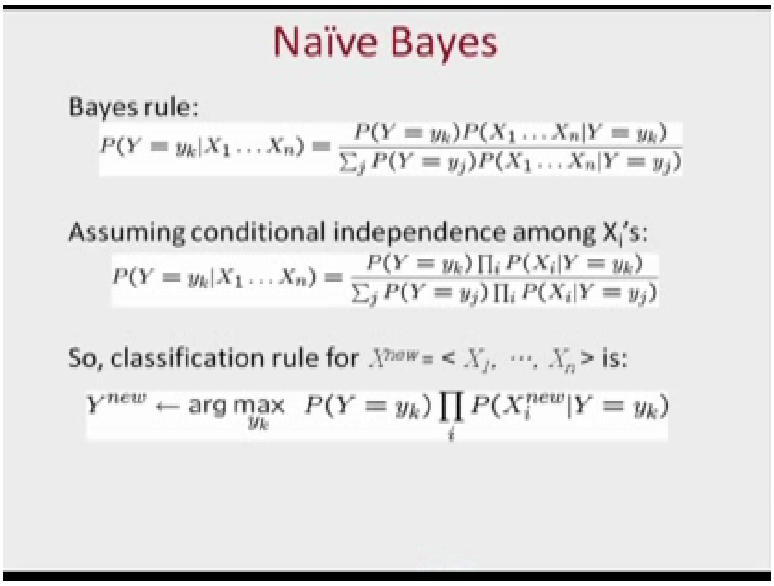
To recapitulate, it says the posterior probability of a hypothesis given the data is given by probability of D given h times prior probability of the hypothesis divided by the likelihood of the data. So, if you are trying to find out, let us apply Bayes theorem to classification. You want to find out the classification Y given the input X. So, if you apply Bayes theorem, probability Y by X is proportional to – as we have seen that, for different hypothesis, the likelihood of the data is identical. So, we need not consider it. So, we can consider the probability Y by Y given X. So, it is proportional to probability of X given Y times probability of Y; that is now X is the input instance and it can be a vector of features.

So, we can write it as probability X 1, X 2, X small n; if small n is the number of features by Y times P Y. Now, this probability X 1, X 2, X n given Y is a joint probability. And, joint probability is difficult to learn and represent, because if there are n features; even if the features are Boolean, there are 2 to the power n possible combinations of the features. And, you have to store the probability values corresponding to all of them. And, this is an intractable problem.

Now, in Naive Bayes, which we will talk about today, we make a simplifying assumption. The assumption that we make is that, individual X i's are independent given Y. In general, we can write this part as probability X 1 given Y times probability X 2 given X 1 Y times probability X n given X 1, X 2, X n minus 1 Y times probability of Y.

Now, in the Naive Bayes assumption, we say that, probability of X I given X j Y is equal to probability X i given Y; or X i and X j are independent given Y. And, based on that assumption, we can rewrite this as probability of X 1 given Y times probability X 2 given Y times probability of X n given Y times probability of Y. This is based on the Naive Bayes assumption. So, we are assuming conditional independence among the individual attributes X 1, X 2, X n. And based on this, we can do the classification. So, we are assuming all the input features are conditionally independent; and this can be computed as probability X 1 given Y, X 2 given Y, X n given Y, etcetera.

(Refer Slide Time: 04:47)



So, if you look at the slide, from Bayes rule, if the probability of Y taking a particular value y k given the value of the input features X 1, X 2, X n is given by probability Y equal to y k times probability X 1, X 2, X n given Y equal to y k divided by – this is the denominator, which is independent of Y equal to y k.

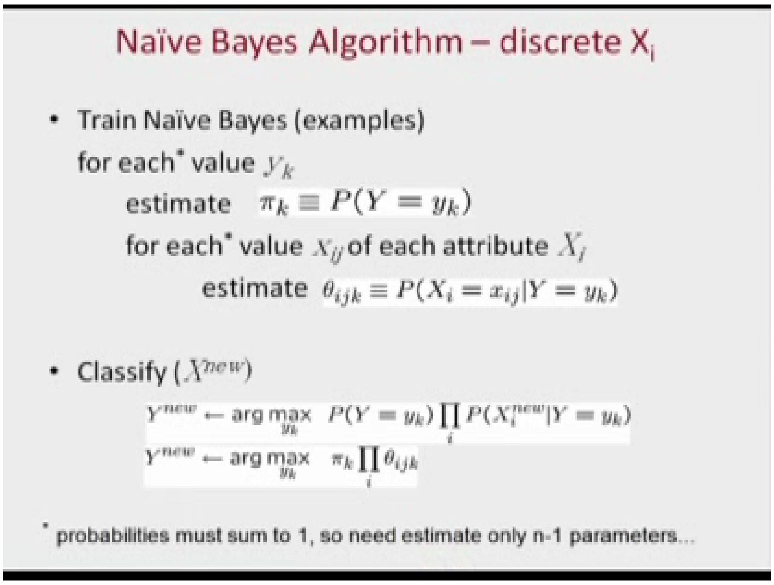
Assuming conditional independence, we get it as probability Y equal to y k times product of probability X i given Y equal to y k. And, there is a denominator. And based on this, we can get a classifier. The classifier says that, given a new example, the classification Y new is that y k for which this quantity is maximized; that is, the product over all the training, all the attributes, probability X i given Y of the training example of the new instance probability X i new given Y, given Y equal to y k times probability of the prior probability Y equal to y k. So, these times this is maximum. So, you want to take that classification for which the prior probability of Y equal y k times this product is maximum.

Now, if we look at the individual probabilities that we require to compute this, what do we notice is that, for each value of suppose Y takes two values: plus and minus; we need to know for all such cases, we need to know a probability of Y equal to true, probability of Y equal to false. And, for each feature X i, we need to know probability of X i given Y equal to plus; probability of X i given Y equal to minus. And, X i can have different values.

Suppose X i has 3 values, for each of the values of X i, we have to estimate probability X i equal to value 1 given Y equal to plus; probability X i equal to value 2 given Y equal to plus; probability X i equal to value 3 given Y equal to plus. Similarly, probability X i equal to value 1 given Y equal to minus and so on. So, what we have is that are the number of probabilities that we required to calculate.

Let us assume that, X 1, X 2, X n and Y are binary attributes. For a Y, we require two values; actually, one of them will suffice that; then we can get for if each X i has two values; so, for each X i will require X i equal to true for each value of Y. If we know X i equal to true, we can also get X i equal to false. So, we have X i equal to true given Y equal to true; X i equal to true given Y equal to false. So, two values for each X i. So, total 2 n plus two values – 2 n plus 1 value will suffice to represent these probabilities, which is very much possible. And, this is a simple; this gives us a simple algorithm for classification called Naive Bayes.

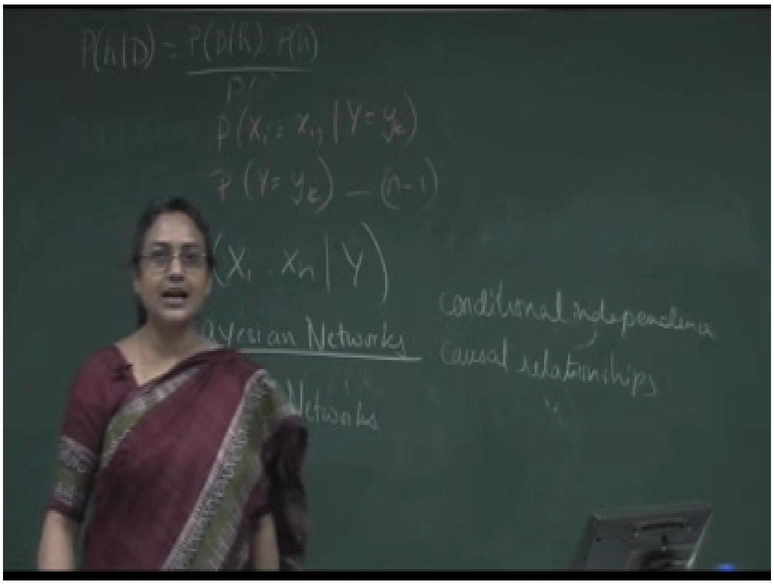
(Refer Slide Time: 08:48)



Now, let us see what is the resulting Naive Bayes algorithm? When you have discrete values of X; for which you can look at the slide, which gives the outline of the Naive Bayes algorithm. This is a very simple algorithm. So, when we train Naive Bayes, we take the training set; and for each value y k; suppose there are n values of y k, we need to estimate only n minus 1 parameters, because y k equal to 1, y k equal to 2 from y k equal to 3, you know the probabilities of some of the probabilities is 1.

So, you need to estimate only n minus 1 of the values. Anyway for each value y k, we will estimate pi k as the probability of Y equal to y k. This is the prior probability. How do we estimate that? Suppose we are given 100 training examples, and y k has three values: v 1, v 2, v 3, y k equal to v 1 for 70 of the examples; v 2 for 20 of the examples; v 3 for 10 of the examples; then, probability of Y equal to y k could be estimated to be 70 by 100; for value 2, 20 by 100; for value 3, 10 by 100 or some other estimate measure which we will again talk about.

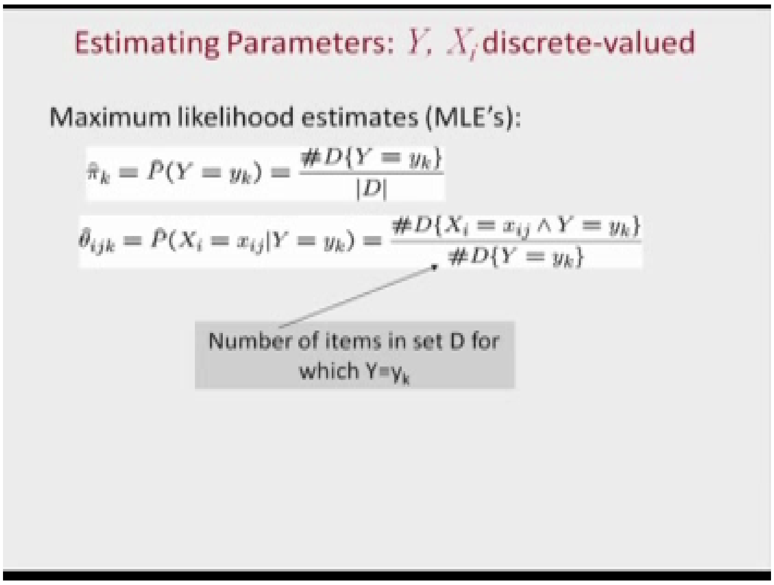
(Refer Slide Time: 10:28)



Now, for each value x i j of each attribute X i for the j-th instance for each attribute X i for each x i j, we will estimate theta i j k as the parameter that, probability that X equal to x i j given Y equal to y k. So, if y k takes n different values; then, this will require n minus 1 estimate. If each x takes k values, we require how many estimates? We require 2 into n estimates, so 3 minus 1 into n. So, small x i j is the different values that attribute x i can take. So, these probabilities we need to estimate. Now, based on this estimate, we can have the class.

So, in the training phase, we learnt this estimates from the training examples. And after we have learnt the estimate, you look at the slide again; we can classify a new instance X new as its class Y new is that y k for which this expression is maximized; probability Y equal to y k times product over i; probability X i new given Y equal to y k as we have seen. In terms of simplified values of the parameters that we have written, this is given by this. So this is the Naive Bayes algorithm for the case, where all the attributes are discrete valued or nominal valued.

(Refer Slide Time: 12:36)

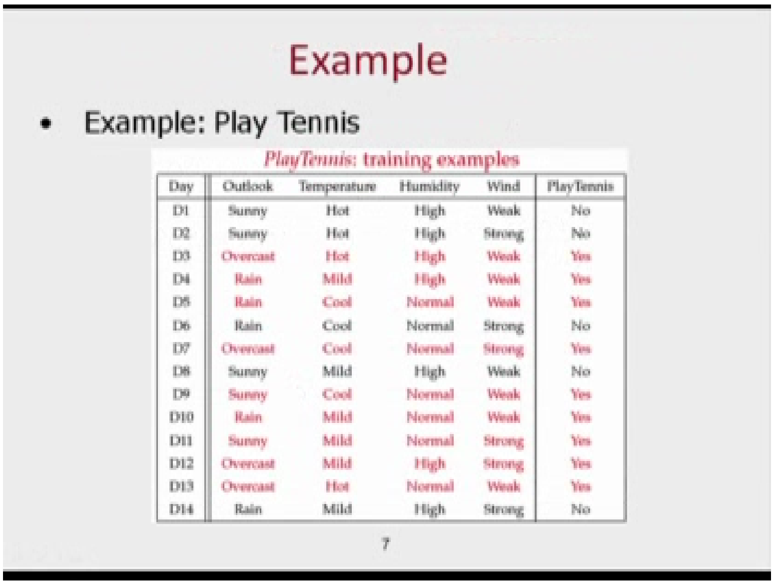


Before we proceed, let us look at an example and one more thing that I forgot to tell you is that, when we estimate these parameters – probability of X i given Y or probability Y, we may sometimes come across a situation, where in the training example, the count for computing the probability is 0. Then, we have a problem. You see in the classification, we have a product that we are computing. We are taking the product of this probability and the product of these probabilities.

Now, if we have insufficient training instances, there may be a case, where probability X i equal to X i j given Y equal to y k; you know there is no training example for a particular y k for which X i is a particular value of X i j. So, this value if we do Naive estimation of the probabilities by frequency counting, this probability will become 0. And, if one probability term becomes 0, the entire product becomes 0. In order to avoid that, we need to do something called smoothing in order to avoid such situations. And so, what we do is that, when we do the estimating of the different parameters; for example, when we try to estimate pi k; for pi k, we look at the number of times y equal to y k divided by total number of data instances.

For theta i j k estimate, which is probability X i equal to small x i j given Y equal to y k, we count the number of instances for which capital X i equal to small x i j and Y equal to y k divided by number of instances for which Y equal to y k. This is the simple formula for maximum likelihood estimation. And in this case there it is possible that, especially in computing theta i j k, sometimes we will get the numerator as 0.In order to avoid that, we introduce smoothing, where we initialize some small probability to each of these values. And we will see in a later slide that, we can add plus 1 to each of the numerator and compensate it by some value added to the denominator.

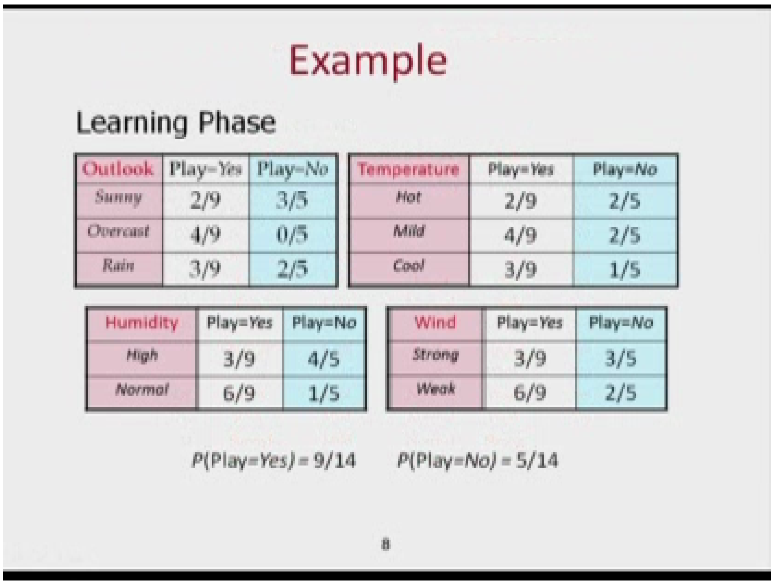
(Refer Slide Time: 15:17)



Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

But, before that, let us look at an example. This is an example taken from a Mitchell's book on Machine Learning; where, we have a description of different days. And the attributes are outlook, temperature, humidity and wind. These are the climate attributes of different days. And, the target attribute is whether it is a good day for playing tennis.

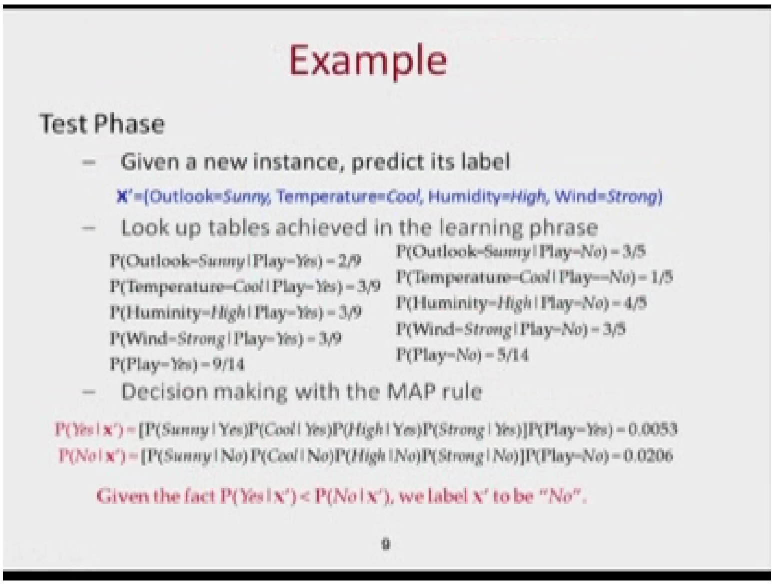
(Refer Slide Time: 15:42)



Given this training example, if you apply Naive Bayes to it, in the training phase, you will output the probabilities. So, if outlook is sunny, play equal to yes – given outlook is sunny is 2 by 9; play equal to - given outlook is sunny is 3 by 5; play equal to yes – given outlook is overcast is 4 by 9; play equal to no - given outlook over cast is 0 by 5 and so on. These are the values that we get by doing the maximum likelihood instance; estimation from the data.

These are the prior probabilities for playing tennis and for not playing tennis. And, these are the values of theta i j k. So, these can be estimated using the previous maximum likelihood estimate formula that we have seen. And this is how we get these values. Now, this is what happens in the training phase.

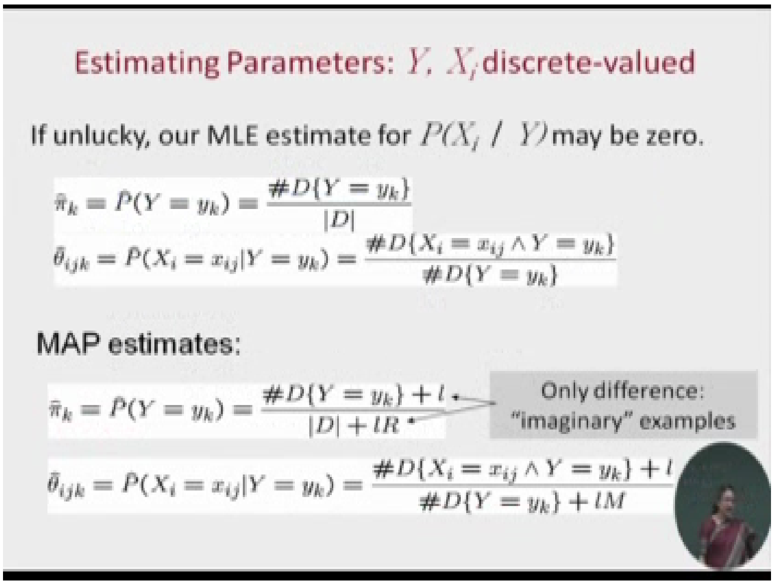
(Refer Slide Time: 16:42)



In the test phase, you have given a new instance and you have to predict its (Refer Time: 16:52) For example, suppose the new instance is outlook is sunny, temperature equal to cool, humidity is high and wind is strong. And based on this probability values that we have seen in the previous page, we can do the decision with the MAP rule.

And we find out that, probability yes given x prime turns out to be 0.0053; probability no given x prime is 0.0206. And, because probability of yes given x prime is less than probability no given x prime, we label x prime to be no. So this is a simple application of Naive Bayes; it is an extremely simple algorithm. We look at the training set. You estimate; do a MLE estimate of the different parameters; then given the test set, we apply that formula.

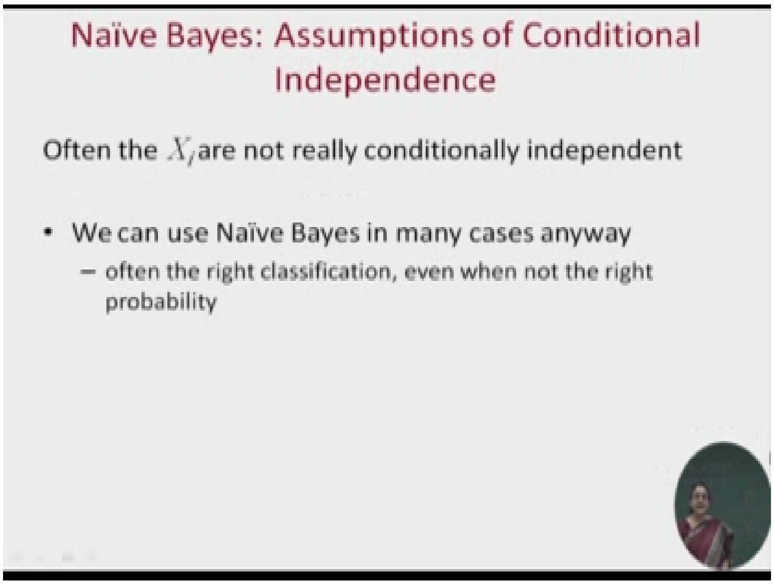
(Refer Slide Time: 17:46)



Now, as I mentioned that, if you are unlucky, the estimate for probability X i given Y may be zero, because there may be that, some particular attribute value is not represented for a particular class, because we do not have sufficient training example. To alleviate the fact, we can use smoothing. There are many approaches for smoothing including many sophisticated approaches, but we will introduce only a simplest approach for smoothing.

What we do is that, for every probability estimates that we do, we add some number; that number could be 1 or could be a fraction l, which corresponds to some imaginary instances, because we are adding a small positive value to the numerator. We must compensate by adding l into R to the denominator, where R is the number of possible values of y k, so that the sum of the pi k's become remain 1. Similarly, to estimate theta i j k, we can add l here. And in the denominator we must compensate by adding l M, so that the sum of theta i j k over a particular value of i j will be equal to 1. So this is smoothing, which we can apply in order to alleviate the problem due to zero probability.

(Refer Slide Time: 19:26)

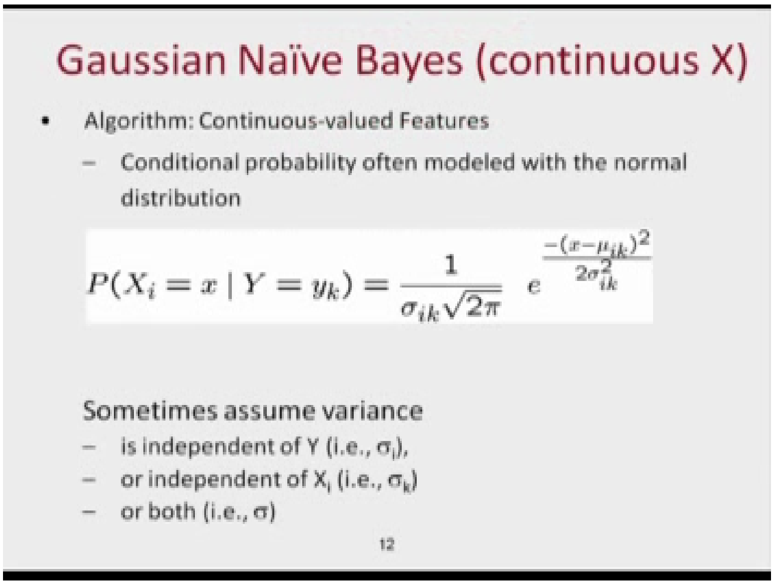


Now, one important assumption that we made in Naive Bayes is that, the X i's are conditionally independent given Y, but this is not really a valid assumption. And, it often does not hold. We can often use the right classification but, even if this assumption does not always hold, Naive Bayes is surprisingly quite effective; given its simplicity, it surprisingly quite effective in many number of cases.

And, often it turns out that, even if the assumption is not valid, Naive Bayes gives the correct classification, because Naive Bayes we are not really using this assumption to find the exact probability, but to choose between the different possible classes. And, in that way, Naive Bayes works quite well in many cases. For example, in text classification, Naive Bayes is a very standard algorithm, which is applied and does surprisingly well it is fast; and because even when the assumptions are not right, it gives the right example.

Now, we will look at the case, where the input attributes are continuous value. We have so far seen that, the both the input attribute and the output attribute are discrete value. What if the input attribute is continuous value? If the input attributes are continuous valued, we can assume that, the conditional probability of that attribute can be modeled by a Gaussian. And based on that, we can have Gaussian Naive Bayes.

(Refer Slide Time: 21:13)

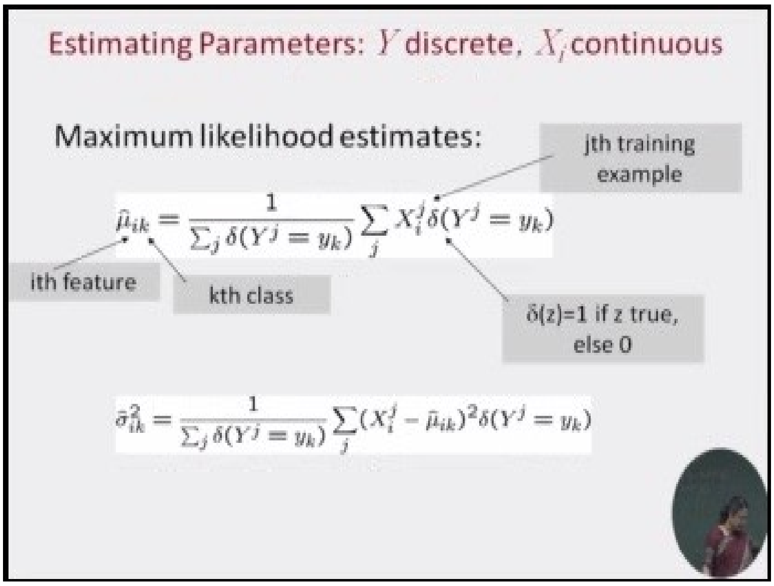


For this, I will request you to look at this slide. Suppose you have continuous valued features, then you can model the conditional probability – probability X i equal to x given Y equal to y k as a normal distribution or Gaussian distribution, which is given by this standard formula 1 by root over 2 pi sigma square e to the power minus x minus mu i k by whole square divided by 2 sigma square. Sometimes we may assume that, this variance – the sigma square term here is independent of Y or independent of X i or both. We can assume that this is same for all X i or Y or you can assume that they are same for all Y i and so on. This makes the model have less number of parameters if you wish. But, under this assumption we can have the Gaussian Naive Bayes algorithm.

In the Gaussian Naive Bayes algorithm, in the training phase, we look at the training dataset. And from the training dataset, we estimate pi k as before; pi k is probability Y equal to y k. The prior probability of the different classes this is estimated as before, but for each attribute X i we estimate the mu i k and sigma i k.

For each X i for a particular y k, in order to find probability X i given Y, we estimate mu and sigma from the data. And after we have done this estimate, in the testing phase, we can classify the new instance X new as Y new is that y k for which this is the standard formula for Naive Bayes. Here probability Y equal to y k was estimated as pi k. And, for probability X i new given Y equal to y k, we use a normal distribution over X i new mu sigma; where, mu sigma were the parameters found in the training phase. So, based on that, we can apply the Gaussian Naive Bayes algorithm.

(Refer Slide Time: 23:49)

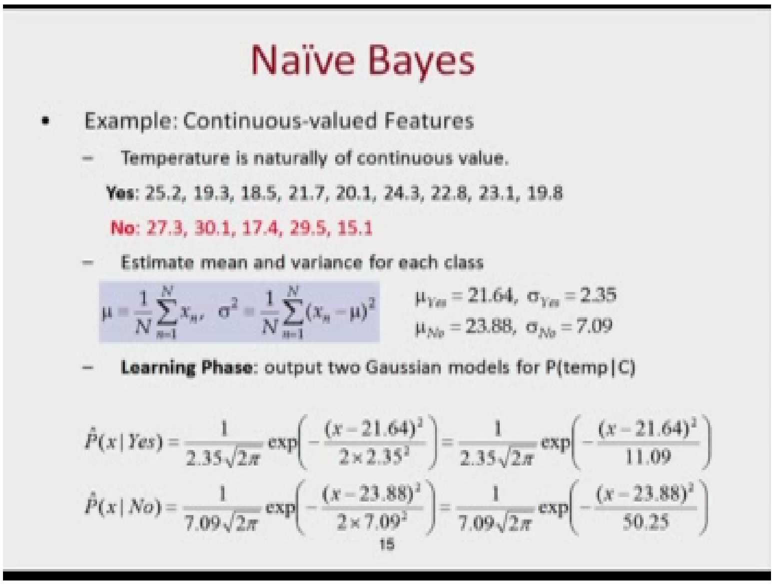


Now, in the Gaussian Naive Bayes, is used for continuous X and so this is used particularly in case, where X is – X is continuous and Y is discrete. And, the maximum likelihood estimate as we have said, the estimate of mu 1 sigma is given by – mu is given by estimate of the mean of the sample. This is the standard way of doing maximum likelihood estimate. And sigma is also obtained by the standard deviation of the sample as is given by this slide. So, to conclude Naive Bayes, is a very simple algorithm, which makes the Naive as the assumption that the different attributes X i and X j are independent given the value of the class.

This assumption is not always realistic, but it simplifies our computations greatly; and in many cases, the resulting algorithm is quite good even though it is so simple. Even though the independence assumption is not always satisfied in practice, as attributes are often correlated, we are get quite good results.

But, we cannot always apply Naive Bayes. And as we have seen, we cannot do the full joint distribution. Probability X 1 X n given Y, it is not tractable to really do this. And to alleviate this, we study Bayesian networks. In Bayesian networks, we can strike a balance; we need not make full independence assumptions or full dependent assumptions, rather we denote the causal relationships and conditional independence.

(Refer Slide Time: 25:52)



The specific conditional independence of the different attributes; and in belief networks, we also denote causal relationships. So, we show the actual relations and actual independences between the attributes. And based on this we can get different learning algorithms, which I have do not make as Naive assumptions as Naive Bayes, but can capture the relationships in the domain.

And it is an advanced topic, and we have different types of the Bayesian networks; we have belief networks also called Directed graphical model. We also have another type of networks – Bayesian networks, which are called Undirected graphical models. And these can capture different relationships. But today we finish this topic.

Thank you very much.