Course Name - Recommender Systems
Professor Name - Prof. Mamata Jenamani
Department Name - Industrial and Systems Engineering
Institute Name - Indian Institute of Technology Kharagpur
Week - 06
Lecture - 27

Lecture 27: Naïve Bayes classifier for content based recommendation

Hello everyone. We continue our discussion on Content Based Recommender System. In this regard, we have started talking about various methodologies that can be used in the context of content based recommendation. So, last class we discussed about decision trees. Today, we are going to talk about net-based classifier. To remind you, why are we using classifiers? Because, now we have in case of content based recommender system, we are dealing with a classification problem where the class attribute or the response variable is derived from the user's feedback, that is your rating matrix and all the item details are obtained from the item description.

By item description, we mean it can be explicitly described, it can be extrinsic or it can be intrinsic. Intrinsic in the sense, it has to be discovered from the item. And in this context, we saw how we can get those features from text data. So, similarly with other kind of items also we can get it.

So, now the major problem with the decision tree kind of setting was that it was not very scalable when the number of features are very high. And moreover, it was actually partitioning the data into subsets at each level. That is why it was very restrictive. So, today we are going to talk about the net-based classifier. Net-based classifier is one of the simplest of the Bayesian classifiers.

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

So, all these Bayesian classifiers are statistical classifiers and they can predict class membership probabilities. And looking at this probability, we can say that to which class a particular tuple belongs. Now, all these Bayesian classifiers basically they are based on Bayes theorem. Now, when we talk about the application of such algorithms in the case of recommender system, it is understood that Bayesian classifiers give reasonably high accuracy and speed when applied to large databases, which is the case in case of recommendation system. Now, net-based classifier is one of the simplest in among the Bayesian classifiers.

And this classifier actually uses a property called class conditional independence. And because of this class conditional independence, this the computations are simplified. And

in this sense, this is also called net-based. As we go ahead, we are going to see what this class conditional independence is. Here, Bayes theorem is the basis for all these Bayesian classifiers.

Director	Lang	award	type	Likes?
Dir1	English	no	Drama	no
Dir1	English	no	Scifi	no
Dir2	English	no	Drama	yes
Dir3	Hindi	no	Drama	yes
Dir3	Other	yes	Drama	yes
Dir3	Other	yes	Scifi	no
Dir2	Other	yes	Scifi	yes
Dir1	Hindi	no	Drama	no
Dir1	Other	yes	Drama	yes
Dir3	Hindi	yes	Drama	yes
Dir1	Hindi	yes	Scifi	yes
Dir2	Hindi	no	Scifi	yes
Dir2	English	yes	Drama	yes
Dir3	Hindi	no	Scifi	no

- Suppose there is a new movie with the following details: (Dir1, Hindi, Yes, Drama)
- We have to predict the class variable, whether he would like the movie or not.

$$P(Likes = Yes|X) = \frac{P(X|C)P(C)}{P(X)}$$

$$P(Likes = No|X) = \frac{P(X|C)P(C)}{P(X)}$$

$P(X)$ is common in both and can be ignored

And as we can see here, just have a review of this Bayes rule. As we can see here, here we have probability of a class given a tuple, X is the tuple. And probability of class given a tuple can be determined from probability of the tuple given a class and probability of class and probability of tuple. So here, these two are the prior probability or a priori values and these two are a posteriori values. So, let us try figuring out how to get all this in the context of a given dataset.

Director	Lang	award	type	Likes?
Dir1	English	no	Drama	no
Dir1	English	no	Scifi	no
Dir2	English	no	Drama	yes
Dir3	Hindi	no	Drama	yes
Dir3	Other	yes	Drama	yes
Dir3	Other	yes	Scifi	no
Dir2	Other	yes	Scifi	yes
Dir1	Hindi	no	Drama	no
Dir1	Other	yes	Drama	yes
Dir3	Hindi	yes	Drama	yes
Dir1	Hindi	yes	Scifi	yes
Dir2	Hindi	no	Scifi	yes
Dir2	English	yes	Drama	yes
Dir3	Hindi	no	Scifi	no

- *X*=(Dir1, Hindi, Yes, Drama)
- We have to find the posteriori probability P(C|X)
- C, i.e., *Likes?* can take value Yes/No

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

To remind you how our dataset is organized, dataset will be organized like this. There will be X, a set of features which we call X here. And there will be a class variable or a response variable and there will be a number of records. So, the problem in case of a recommender system is we have to build a Bayesian classifier given this dataset. So, this is the example pattern or the training pattern.

How do you get this example or training pattern? You get it from the item, this you are getting from the items, these you are getting from the items, these are item features and this is the class variable. So, which you obtain from the rating matrix or some kind of feedback matrix. It can be explicit or it can be implicit. Now, in this setting what we are supposed to determine? We are supposed to determine when a new item comes, new item. It will have again feature 1 to feature p.

If there are n attributes, i.e., $X = (X_1, ..X_n)$

$$P(X \mid C) = P(X_1 \mid C)P(X_2 \mid C)..P(X_n \mid C)$$

$$P(X \mid C) = P(Director = Dir1 \mid C) \times P(Lang = Hindi \mid C) \times P(Award = Yes \mid C) \times P(Type = Drama \mid C)$$

So, given this set of features, given this set of features there will be respective values. Given this set of features what is going to be the class? Whether it belongs to class 1 or it belongs to class 2 and so on. So, which means this will be known and this we have to determine for every available class level. So, this side, this value, this value and this value must be known. So, the conditional probability that probability of X given C must be known, conditional probability of C must be known, sorry, the probability of C must be known, prior probability and prior probability of X must be known.

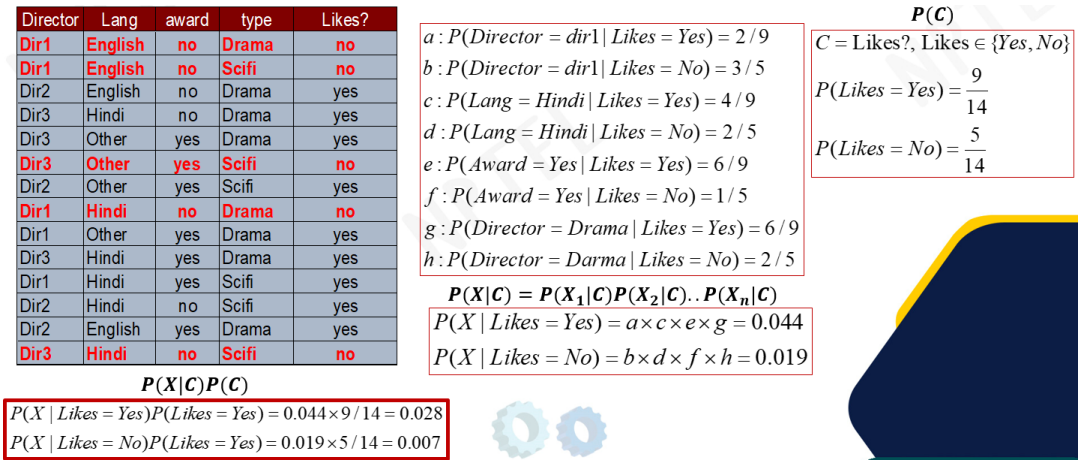
So, let us go ahead. So, this is the example that we are, we also considered in last couple of classes. Now, in this example, this is our class variable and these first four are the features. So, suppose one new data comes, director is one, Hindi, movie is Hindi, then award, it has got award and it is a type called drama, it is called genre called drama. So, given this new movie, whether you are going to recommend this to the user or not, that is the problem.

So, now user, what you are supposed to determine, whether user will like it or user will not like it. So, for this purpose, we are supposed to taking this is C, C, this is C, find out C given X. So, for yes, we have to find out certain probability, for likes equal to no, we have to find out certain probability. Out of these two, whichever is the highest, we will be treating that this particular feature vector belongs to that class. So, if yes is high, we will recommend it, if no, yes, your likes is no, I mean if the no is high, then we will not recommend it.

Now, come to the computation of X. Now, what is X? X now is a not a single value, it is multiple values. Now, if they are multiple, it may be possible that these features are dependent on each other. So, if they depend on each other, then finding this conditional

probability becomes difficult. Now, if we assume that all these attributes are independent of each other, then this computation becomes easy.

They become just the multiplication of individual attributes given C. So, which means now, what was our data set? We had a movie with director 1, language was Hindi, it got award and it was a drama type. So, probability of and assuming that they are independent of each other, all these features, this can be computed, probability of director given C, probability of language given Hindi, sorry, given C, probability of language equal to Hindi given C, probability of award equal to yes given C and so on. So, now the question is how many C's we have in this setting in the example, the last example. This is our class variable C.



So, C can take two values, C can be either yes or it can be no. It can be yes or it can be no. So, for both yes and no, we will compute these values and find out. So, this computation is as follows. So, now look at this computation. What was our example? Our example was, our example was, example was director is 1, language is Hindi and award and drama. Director is 1, language is Hindi, award is, it has got award and it is a drama type. So, what are we going to determine? We are going to determine probability of X given C. So, this C can be yes. So, for yes, 1 we have to compute.

So, when we compute yes, we have to find out probability of X1 given C into probability of X2 given C, probability of X3 given C and probability of X4 given C. So, here this is X1, this is X2, this is X3, this is X4. So, X1 given likes equal to yes, X1 given likes equal to no and so on. So, which means now you have found out for all the positive values, these are when likes equal to yes, this is the value for the first attribute, second attribute, third attribute and fourth attribute.

So, this is A, C, E, G. So, A, C, E, G are getting multiplied and this is the value. Similarly, for likes equal to no, these values will be multiplied and you get this value. So, next is we are supposed to determine. So, one thing I forgot to tell you, in this setting

when you find out this like equal to yes, I mean the likes equal to yes given X and likes equal to no given X, in both what is common is this one. So, we may think of not considering it further.

So, we do not have to really compute these two. So, when we do not compute these two, what is to be compared? This is to be compared and this is to be compared. So, here C is yes, here C is no. So, now when C is yes, C equal to yes, this is the value. How do you get this value? This we have computed last time also, last class also. Out of total 14 records, 9 records are yes. So, it is 9 by 14, 5 records are no. So, it is 5 by 14. So, this you computed. So, now probability of X given C into probability of C.

So, this you have computed probability of X given C and now you multiply both. So, this is for C equal to yes, this is for C equal to no. So, where from this value is coming? It is coming from this value, this particular value is coming from here, this particular value is coming from here and you multiply it with this one and this one. This one is coming from here, where? So, this is coming from here, this is coming from here.

So, now you got these two. So, given this record new movie where director is director 1, language is Hindi, award is yes and there is some issue here. So, this is not the director again, this is type. This is a typo. So, this is type. So, type is drama. Then whether the movie should be recommended? Yes, it should be recommended because this value is high. When you compare this two, this value is high. So, this should be recommended. Now, we have another situation here. Now, suppose we have a new movie with a new director.

Director is for and is director 4, its language is Hindi, it has got award and it is a drama type. However, so far there is no evidence that there is somebody called director 4. So, because that is not there in the database, both these values are going to be 0 and because of our class conditional independence and while finding out this joint probability values, we are getting 0. Why? Because this is 0, this is 0. Now, how to best deal with this situation? Now, to deal with this situation, we use some concept called Laplacian correction.

So, to understand what is Laplacian correction, let us extend this example. Now, suppose the class like equal to yes in some training dataset D and there are 1000 tuples of this type in the dataset. Now, out of this 1000 with director 1, you have this many, with director, you have 0 tuples with let us say director 1, 990 tuple with director 2 and 10 tuples with director 3. Now, what is the probability of this? Probability of this is 0, 0.99 and 0.010. Now, without Laplacian correction, we saw that we are getting 0 probability when we consider some tuple where one of the attributes is not existing in the current dataset. There is no evidence in the current, it is not visible currently. So, now what do

you do? You pretend, you assume that we have one more tuple. So, already one more tuple of each.

So, already you had 1000 tuples. So, assuming that there is one more tuple on director 1 and one more tuple on director 2 and 3 respectively, we now get 100, 100 and 3 tuples. Now, because we have assumed that there is a tuple called for director 1, the probability for director 1 is now 1 upon 1003. So, also for director 2 and director 3. So, even if there is no evidence, now because of our assumption, now look, we have given equal importance to all the three directors. So, accordingly we have added one 1 record in the name of director 1, 2 and 3 respectively.

1. Let *D* be a training set of tuples of attributes and a class variable. Each tuple is represented by an *n*-dimensional attribute vector, ***X*=(** $X_1, X_2, \ldots, X_n$), depicting *n* measurements made on the tuple from *n* attributes. Suppose that there are *m* classes, $C_1, C_2, .., C_m$. The naïve Bayesian classifier predicts that tuple ***X*** belongs to the class C_i if and only if

$$P(C_i \mid X) > P(C_j \mid X) \text{ for } 1 \le j \le m,\ j \ne i.$$

2. The above is called *maximum posteriori hypothesis*. By Bayes' theorem

$$P(C_i \mid X) = \frac{P(X \mid C_i)P(C_i)}{P(X)}$$

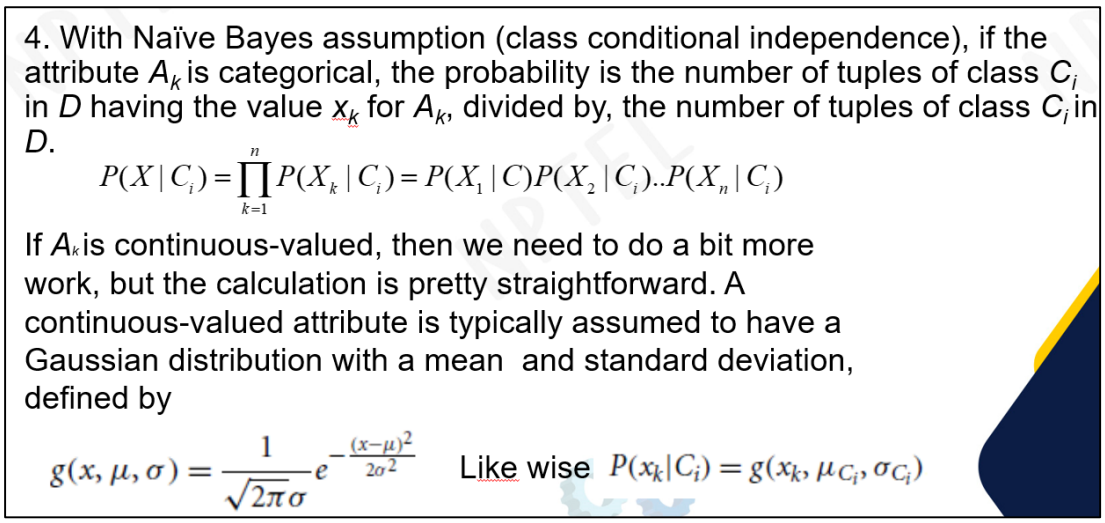
3. Since the denominator is common, only the numerator can be used for comparison.

So, therefore, recomputing this, we get these values. Now, if you compare this value with original value as what? 0.990. This is not much difference. What was the original value in this case? It was 0.010. This is also not much difference. But this one which was 0, now has got some value. So, however small it may be, you will not land up in a situation like this. This situation is not good because this person, there is evidence that this person likes in the movies. He likes movies with hours.

He likes drama type of movies. Now, because director 4 is not there, will you not recommend him? So, now we have a correction because of which with some less probability, we will recommend it. But it is, it will at least come in the recommendation list. Now, let us just have a look at once again the Naive Bayes classifier complete algorithm. We have seen it through an example. So, as I go ahead, I will be, I do not have to explain much because you have already seen this.

Now, let D be the training set of topples of attributes and class variable. This attributes are drawn from item description and class variable is the liking or disliking of the user. Now, each topple is represented by an n dimensional attribute vector x1 to xn depicting n

measures made on the topple from n attributes. Suppose there are m classes. In our case, how many attributes were there? 4 attributes.



How many class levels were there? 2 class levels. But it can have multiple class levels as well. So, now the Naive Bayes classifier predicts that topple x belongs to class ci if and only if this probability is higher than the probability of the rest. So, that all this j from 1 to m which does not include i should have lesser probability compared to this. So, higher, so whichever has the highest probability will be predicting that as our recommended class.

So, with certain high probability. So, in our example, what was that recommended class? Our recommended class was yes. So, this was our recommended class. Now, this above is called maximum posteriori hypothesis by Bayes theorem. We know that this is true since denominator is common only the numerators can be used for comparison.

This is also we have seen. Now, this is the Naive Bayes assumption. With Naive Bayes assumption, the class conditional probability, class conditional independence if the attribute ak is categorical, the probability is the number of topples of class ci in D having value xk of ak divided by the number of topples in ci in D. So, this is total number of that variable divided by c.

So, this is what we did here. This is what we did here. For example, in case of director 1, it was director 1. How many times the director 1 was there? Director 1 was here, here, here and here. In 4 places, in 5 places in fact. So, out of this 5 places, in how many places it was yes and how many places it was no. So, director was no here, he was no here, he was no here and there was a yes here.

So, that is what it says that you have to find out how many number of director 1 divided by the here one i is also missing. So, divided by the ci then how many number of in how

many number of places yes is there. So, now, if ak is continuous valued, then we need to do. Now, this is what the case of discrete. Now, if the ak is continuous valued which is quite possible because some of the attributes in our case can be continuous as well.

For example, suppose you would have given the given the rating average rating given by I mean the total average rating for the movie. So, that would have been let us say 8.2, 7.5 that would have been continuous. In that case what we have to do? We have to think that this is being drawn from a Gaussian distribution with min mu and standard deviation sigma and we can use the pdf of that distribution as a measure.

So, in our case we can find out the average values for mu ci and sigma for ci and with respect to that particular attribute we can find out. So, in our case we had this c to be discrete, but it can be continuous it says that. Now, what are the issues with Naive Bayes classifier in the context of recommendation system? Its performance is unsatisfactory when the documents in the training set have different lengths thus resulting in a rough parameter estimation. Now, handling rare categories if fewer documents are available from a particular category probability is going to be very low. Now, these conditions frequently occur in user profiling task where no assumption can be made on length of training documents and where obtaining an appropriate set of negative example is problematic.

Because most of the time either the person will be saying giving some rating likes or dislikes like or in a Likert scale, but saying dislikes specifically in a Likert scale is also rare to get, but it cannot be 0. So, these are the references that we are we have done and specifically we have taken example from here. And this is one of the very early adoption of such a system for news recommender system and if you look at today also Naive Bayes is still a popular algorithm. So, these are my conclusions. Naive Bayes is a simple yet powerful method for recommendation generation.

It works on the principle of Bayes rule and simplifies the computation under class conditional independence. Laplacian corrections are often applied to deal with the situations when zero probability occur due to lack of training example. With this we finish this lecture. Thank you.