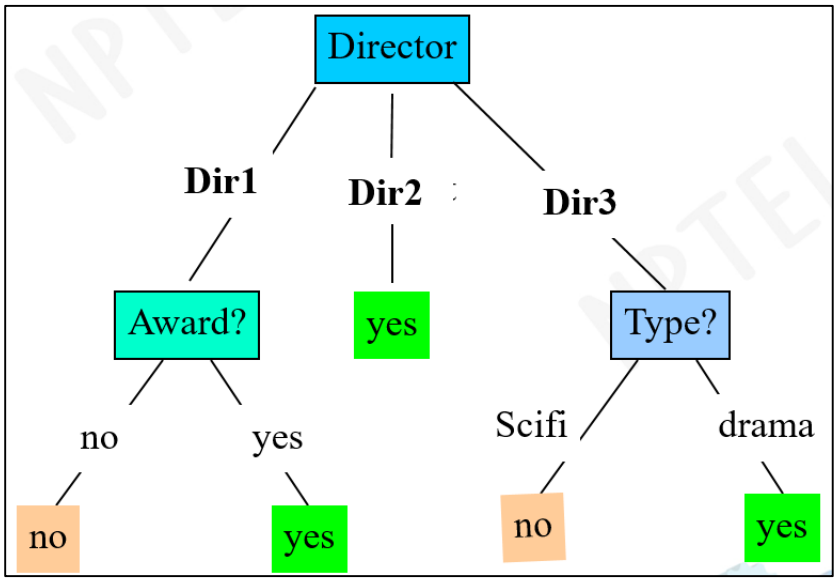
Course Name - Recommender Systems
Professor Name - Prof. Mamata Jenamani
Department Name - Industrial and Systems Engineering
Institute Name - Indian Institute of Technology Kharagpur
Week - 06
Lecture - 26

Lecture 26: Decision Trees for content based recommendation

Hello everyone. Welcome to the 6th module. We have already started content based recommender system from 5th module onward. And here specifically we are going to talk about various methods which are used in the case of content based recommendation approach. So, in this particular lecture we are going to see how to use decision tree for content based recommendation. The concept of decision tree we have seen in many earlier lectures as well.

So, in today's lecture we are particularly going to build a decision tree from our data set. So, this decision tree induction is a learning algorithm for creating the decision tree where the attributes are all discretized are discrete or discretized and class variable is also having discrete values. So, when you build a tree you make a structure where there will be root node there will be branches and so on. The root and the internal nodes are called non leaf which are called non leaf nodes they basically test various attributes and as you branch as you branch based on certain criteria you end finally, in a leaf node or the terminal node.



So, if you follow the path starting from the root and depending on the root variable take a branch then you can reach the leaf node. And when a new data set comes once you create the tree when a new data set comes you start testing the attributes from which is

there in the root node and you continue. So, this is one example which we already saw this is a decision tree pertaining to this particular data set. Now there are four features and one response variable or one dependent variable. So, these are the four features and this is the response variable.

	Director	Lang	award	type	Likes?
1	Dir1	English	no	Drama	no
2	Dir1	English	no	Scifi	no
3	Dir2	English	no	Drama	yes
4	Dir3	Hindi	no	Drama	yes
5	Dir3	Other	yes	Drama	yes
6	Dir3	Other	yes	Scifi	no
7	Dir2	Other	yes	Scifi	yes
8	Dir1	Hindi	no	Drama	no
9	Dir1	Other	yes	Drama	yes
10	Dir3	Hindi	yes	Drama	yes
11	Dir1	Hindi	yes	Scifi	yes
12	Dir2	Hindi	no	Scifi	yes
13	Dir2	English	yes	Drama	yes
14	Dir3	Hindi	no	Scifi	no

This is the response variable. So, now the question is why director has to be there at the root? Why not language, award or type? It appears that this attribute language is not appearing here at all. So, on what basis all these decisions are made? Now coming to these attribute selection criteria we already have known about few attribute selection criteria and few such we are going to here for making this split. So, this is the algorithm for decision tree induction. You construct the tree in a top down recursive divide and conquer manner at start all the training examples are at the root attributes are categorical.

As I told if it is continuous you have to discretize them in advance. Now example patterns are partitioned recursively based on the selected attribute. The test attributes are selected basis on the basis of certain heuristic or statistical measure. Now when do you stop? After all the samples are exhausted and you have made the tree there is no remaining attribute to be considered then you stop. So, this is what I was trying to make you stop when all these examples belong to this class no all the examples under this belong to yes and so on that is your stopping criteria.

And you can also stop these are the three different situations all the samples of a given node belong to the same class. There is no sample less left that is second and sometimes there is no attribute remaining to explore further you are exhausted with all the attributes.

In that case the majority voting is employed. Majority voting in the sense out of all the in that particular set depending on how many samples variables of which category are there whatever is the majority you employ it as your class. Information gain as an attribute selection measure we have already seen that we were using to find out the best set of features for any model.

$$Gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459.$$

$$Gini_{lang \in Hindi, other} =$$
$$= \frac{10}{14} Gini(D_1) + \frac{4}{14} Gini(D_2)$$
$$= \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right)$$
$$= 0.443$$
$$= Gini_{lang \in English}$$

So, now here particularly in decision tree information gain Gini index and other in criteria they are used to decide the node and the splitting of the node splitting of the variables based on the node. Second criteria could be your Gini index we have last class we have talked lot of things about them. So, I am not going to discuss more ok. So, now coming to this particular decision tree we have already discussed how to compute this information gain and we how to compute this information needed based on this entropy. So, what we do is we calculate the entropy with respect to D first as usual 9 are the yes 5 are the no and the formula is sum of minus p i log of p i over all the labels.

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

So, you got this now you got 3 partitions with respect to the variable director what are the 3 partitions with because director has 3 values director 1 director 2 and director 3 with

respect to 3. See look how many rows are with director 1 2 plus 3 5 director 2 4 rows director 4 5 rows and so on. Now how many yes and how many no also has been tabulated here. Based on that following this formula for each of this partition you calculate this entropy value. As you calculate this entropy value the total information content these are the probability of getting this fraction this is the probability of getting this fraction is the probability of getting this fraction and you multiply it with respect to your information content.

- Class P: Likes = "yes" (9 tuples)
- Class N: Likes= "no" (5 tuples)

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

Director	p_i	n_i	$I(p_i, n_i)$
Dir1	2	3	0.971
Dir2	4	0	0
Dir3	3	2	0.971

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

$$Info_{dir}(D) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2) = 0.69$$

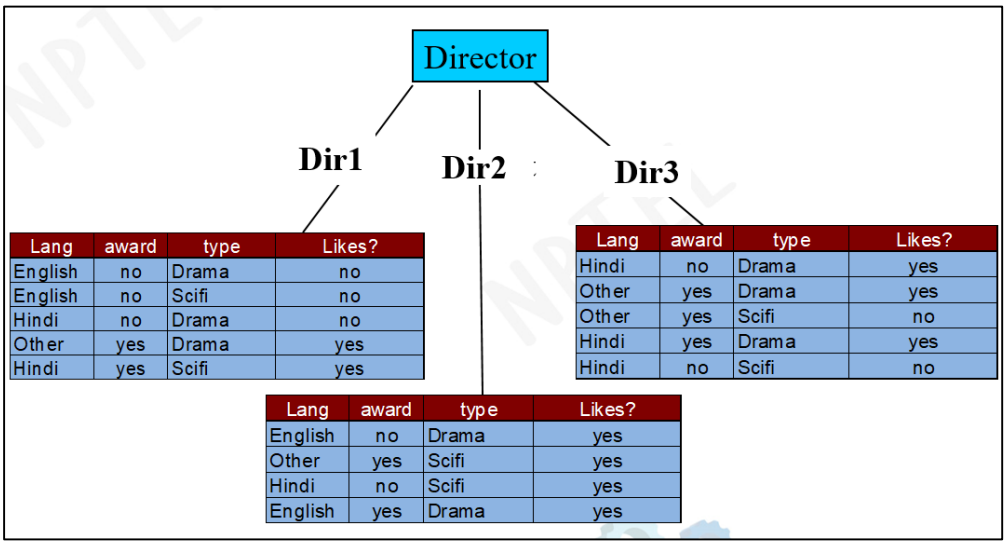
$$Gain(dir) = Info(D) - Info_{dir}(D) = 0.246$$

$$Gain(language) = 0.029$$

$$Gain(award) = 0.151$$

$$Gain(type) = 0.048$$

And finally, with this process if you take this information D minus with respect to the director you get some gain. So, gain here is 0.246 gain here is 0.151. So, if we would like to compare where it is the best here it is the best. So, that is how in the tree this becomes the root. So, this has the highest information gain. Now with making this as the root what happened we have now partitioned the data set. In the first partition we have kept all the entries with respect to director 1 here all the entries with respect to director 2 and director 3. Now look at the entries related to director 2 here all the elements are yes.

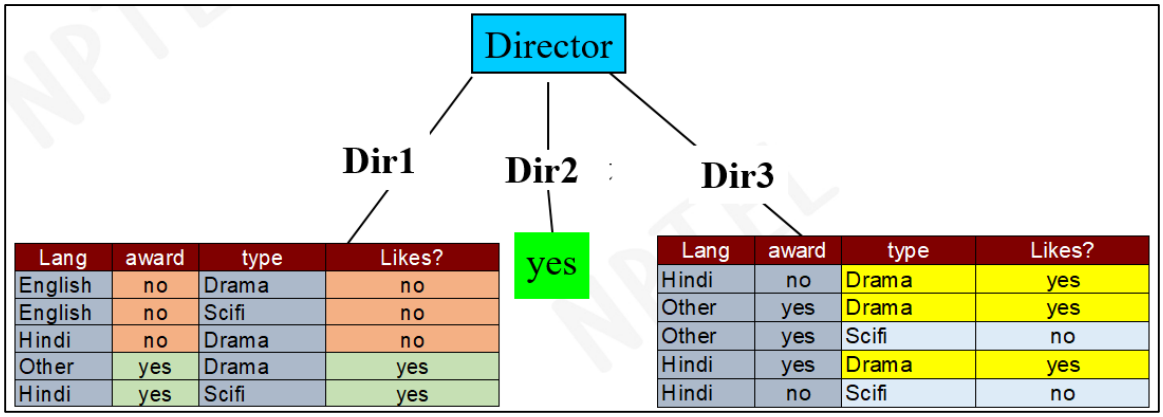


Lang	award	type	Likes?
English	no	Drama	no
English	no	Scifi	no
Hindi	no	Drama	no
Other	yes	Drama	yes
Hindi	yes	Scifi	yes

Lang	award	type	Likes?
Hindi	no	Drama	yes
Other	yes	Drama	yes
Other	yes	Scifi	no
Hindi	yes	Drama	yes
Hindi	no	Scifi	no

Lang	award	type	Likes?
English	no	Drama	yes
Other	yes	Scifi	yes
Hindi	no	Scifi	yes
English	yes	Drama	yes

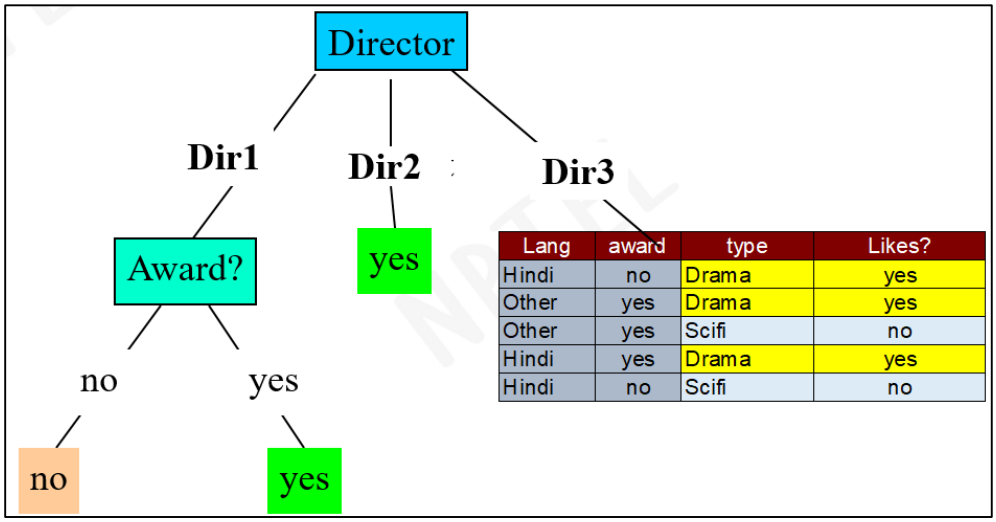
So, we do not have to test further because if the director is director 2 the person is going to like the movie. To remind you how did we construct how did we arrive at this kind of data set. In the typical recommender system setting you find out the item features then from the rating table you find out whether the person likes the item or not that is how you constructed a data set a complete data set which had all these 3 3 group of rows together. Now with respect to director 2 this person has seen all the movies. So, in future if a movie comes with respect to director 2 it is going to be recommended.



Lang	award	type	Likes?
English	no	Drama	no
English	no	Scifi	no
Hindi	no	Drama	no
Other	yes	Drama	yes
Hindi	yes	Scifi	yes

Lang	award	type	Likes?
Hindi	no	Drama	yes
Other	yes	Drama	yes
Other	yes	Scifi	no
Hindi	yes	Drama	yes
Hindi	no	Scifi	no

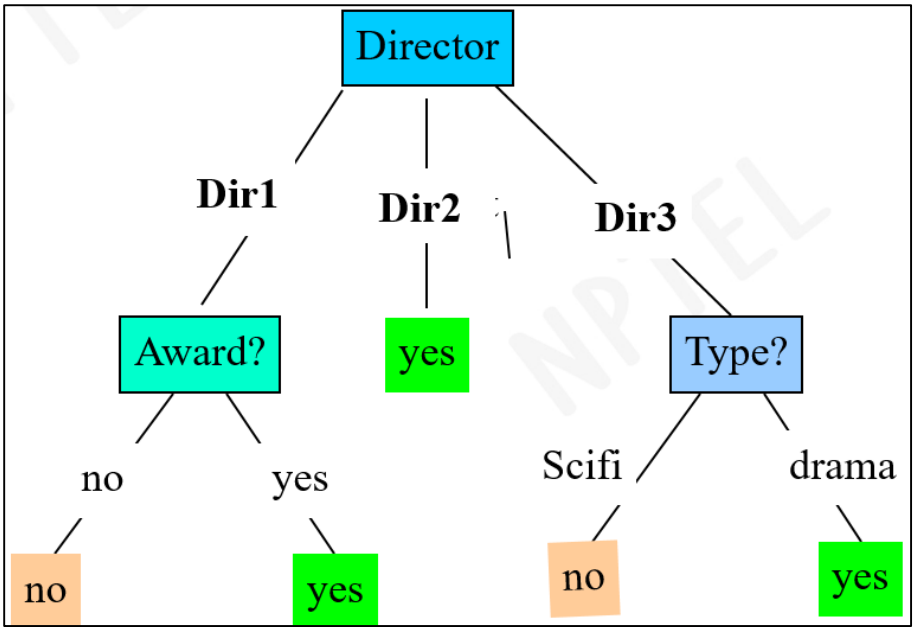
So, moving ahead here there is no need to go further. So, this is our stopping criteria is met. Now with respect to this one we are again going to find out the information information again ok. So, now which one we have now 3 attributes language, type and hour. Now which with respect to which attribute that has to find out we have to find out with this small data set we get information again with respect to these 3 attributes.



Lang	award	type	Likes?
Hindi	no	Drama	yes
Other	yes	Drama	yes
Other	yes	Scifi	no
Hindi	yes	Drama	yes
Hindi	no	Scifi	no

After this we are not going to do the computation now it turns out that hour gives the highest attribute highest information again. Now this hour when the hour is no likes is also no when hour is yes likes is also yes. So, our stopping criteria is met. So, this part of

the tree is over. So, here again we have to test with language hour and type and it turns out that type keeps the highest again.



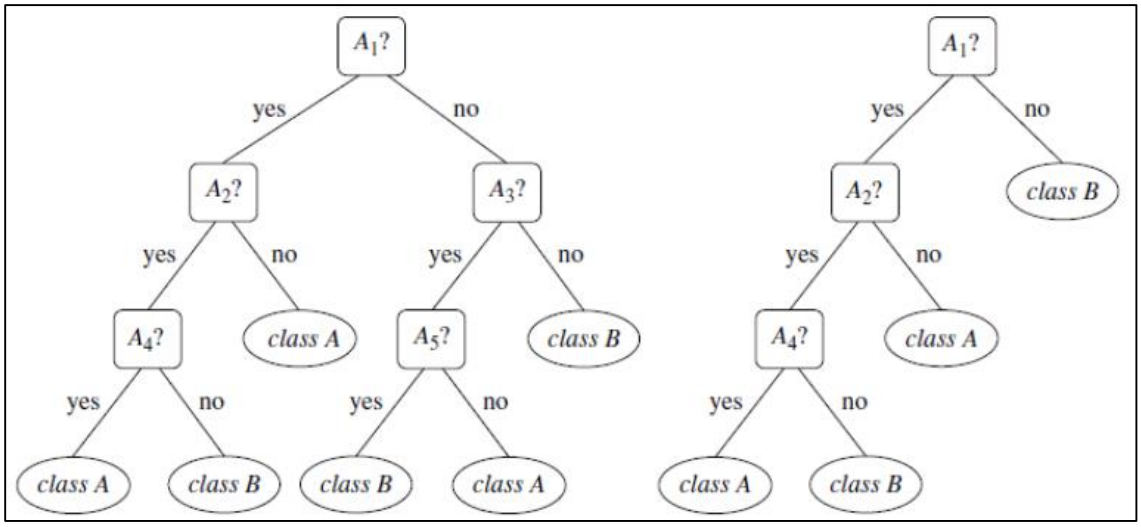
So, and we can see when it is drama type the person likes the movie otherwise he does not. So, our full tree is met ok. So, now what are the problems in decision tree induction? If the dimensions are very high and sample size is low then you are not going to get a good tree. Typically, the ratio of the sample size with respect to dimensionality must be as large as possible. If there are noises outliers and so on you may also get erroneous result.

So, in that context you need to carry out something called tree pruning that we are going to see shortly. Now scalability is another issue because as you saw to find out information gain make the partition you have to consider all the elements in the data set together all the rows in the data set together. So, if the data set is very very large then you cannot feed the entire thing into the main memory. So, as a result computational for computational I mean the what will happen you have to have algorithms which will be keeping something in the secondary memory then taking it back and this back and forth will be increasing your computational time. So, now for removing these noises and outliers something called tree pruning is carried out.

So, when you build the decision tree and there are anomalies it will actually over feed the data it will not only feed the actual data it will try feeding the outliers as well. So, under such circumstances what do you do? You when the tree is growing starting from the node it is growing either you stop it in between do not let it grow or you make it grow then using certain criteria you try cutting it. So, this is what I was trying to say let us say this is a fully grown tree. Now, suppose we decide to cut it here after this A 3. So, if we

stop it here this is our leaf node which leaf node I will adopt the highest occurring leaf node.

So, either I put it class B or I put a distribution like 2 by 3 times I am going to get class B and 1 by 3 times I am going to get class A either you put some kind of distribution actually counting the records or you simply put it as class B. So, that by chance if something is erroneously detected as class A it is eliminated. So, this is from a fully grown tree I have chopped off. Similarly starting from root node when I move ahead I may decide to stop here. So, if I decide to stop here again I will follow the same approach what will be the class level here class A is appearing a number of times.



So, my level will be A or I can also have a distribution where A is occurring with probability 2 by 3 and B is occurring with probability 1 by 3. I can also extract rules from a decision tree how do I extract the rules to extract the rules if you remember while talking about the rule bases rule usually has 2 parts antecedent and consequence. So, this part before the leaf node makes the antecedent and this is your consequence. So, accordingly the antecedent how many rules will have that that many branches this is my first rule. So, this if I follow this branch I get my first rule if the director is dir 1 and award is no then likes will be no if director dir is director is dir 1 award is yes then like is yes.

So, likewise I can now extract 5 rules. So, when we this is a very small example, but when we have a large data set with many variables we are going to get a very big tree. So, in that case we are supposed to get large number of rules. So, out of those rules how many to select. So, while talking about rule based recommender system we will also discuss in this line.

So, these are the disadvantages of decision tree induction in recommender system. This is particularly not suitable for high dimensional data such as text making a tree is difficult

and it experimental it has been shown that it gives poor performance. So, some of the very early applications they show good utilization of decision tree and moreover this is a very very explainable method. So, therefore, as a first method we have discussed about it, but let me tell you this is not a practical method as when we deal with a very large data set. We can use them to get the rules and use the rules after removing some of the redundant and not very useful rule and make the system faster.

So, it has been observed that even rule based classifiers can provide superior result because they do not assume strict partition of the feature space. So, what do we mean by strict partition? Because you saw that as we move ahead we started with the entire data set and each side of the tree the data set was distributed. So, we have to follow only one path. So, if any observation comes which is not following any of this path this is discarded straight away. So, such things can be avoided if we go for a rule based approach.

So, with this we stop these are our references and this is how we conclude. Decision tree one of the simplest choice for content based recommender system if the number of features are less. Many early recommendation implementation use decision tree methods, but they are not widely adopted due to scalability issues. However, they are very explainable. Thank you.