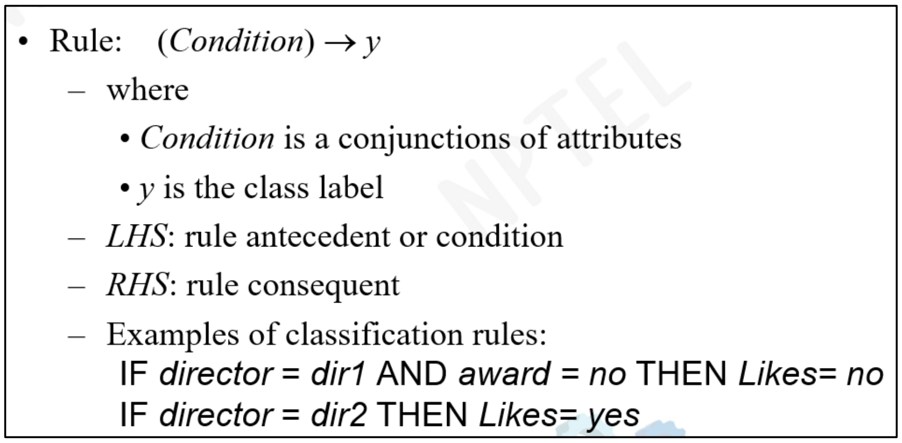
Course Name - Recommender Systems
Professor Name - Prof. Mamata Jenamani
Department Name - Industrial and Systems Engineering
Institute Name - Indian Institute of Technology Kharagpur
Week - 06
Lecture - 29

Lecture 29: Rule based classification

So, in this lecture, hello everyone. In this lecture, we are going to talk about rule based classification in the context of content based recommender system. And with this, we continue to see what a rule based classifier is. In case of a rule based classifier, what we do? We classify the records using a set of if then else rules. While talking about rule based classification, already in many settings, in fact, let me tell you in the context of collaborative filtering also, we saw one rule based classifier. What was it? It was association rule mining based classifier.

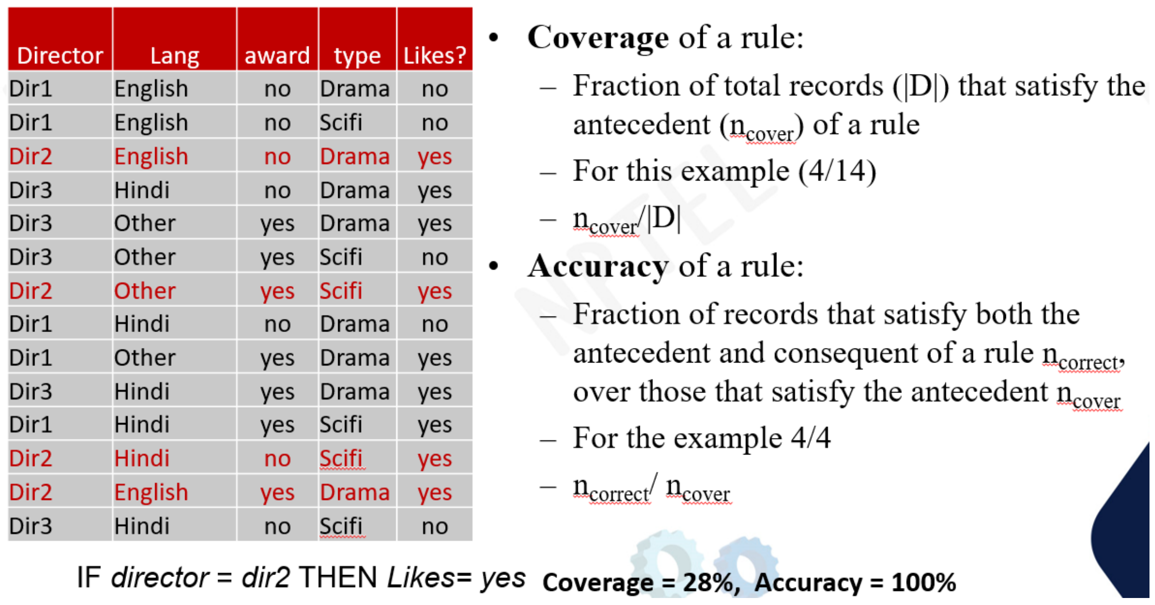
In that association rule mining based classifier, it was a different setting where we were not using item features, rather we are taking the data from the rating matrix itself. And we were finding out which items are generally bought together. And looking at how they looking at these groups, we try to find out certain frequent patterns. And from those frequent frequent patterns, which were satisfying certain criteria, we tried creating certain rules.


- Rule: $(Condition) \rightarrow y$
  - where
    - *Condition* is a conjunctions of attributes
    - *y* is the class label
  - *LHS*: rule antecedent or condition
  - *RHS*: rule consequent
  - Examples of classification rules:
    IF *director* = *dir1* AND *award* = *no* THEN *Likes*= *no*
    IF *director* = *dir2* THEN *Likes*= *yes*

So, that was also a kind of rule based classifier. But today, we are going to talk about rule based classifier in the context of content based recommendation, where this antecedent values will be coming from the item description. So, this antecedent or this conditions will be coming from the item description and this value y and this value y is the response variable. So, this will be from the item feature and this is user feedback. Now, when it comes to condition, this condition is it can be complex, there can be and condition, there can be or condition and so on.

y is the class level. So, this is antecedent, LHS is antecedent, RSS is consequences all these things you already know. And with respect to the data set that we have been considering so far, this could have been one of the rules. If director is DIR 1, award is no, then likes. In fact, these rules we discovered from the decision tree.

Now, rules can have two important criteria, coverage and accuracy. So, the coverage is the fraction of total records that satisfy the antecedent of a rule. Now suppose this is our rule, if director is DIR 2, then likes equal to yes. Now in how many places DIR 2 is there? 4 places 1, 2, 3, 4. So, how many topples this rule covers? This is the total number of topples that is D, this is number of total number and out of this 4.



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

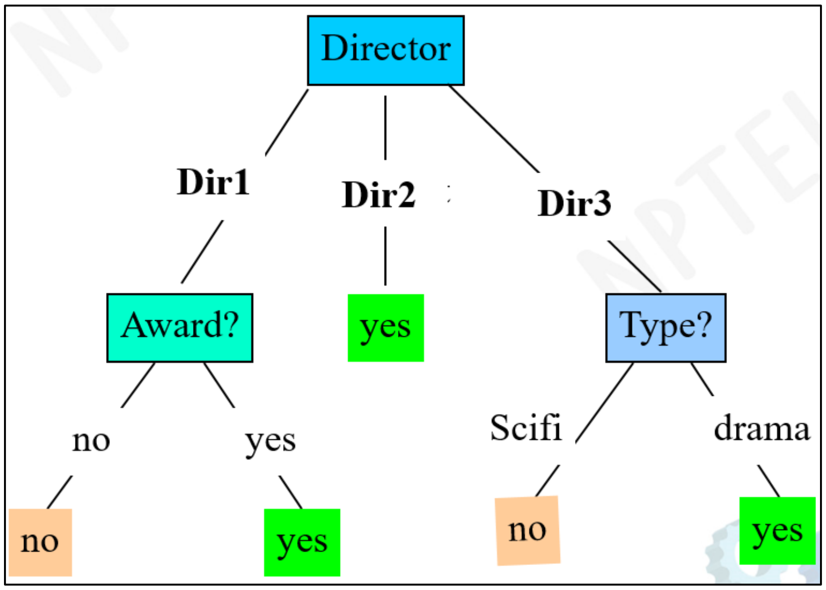
- **Coverage** of a rule:
  - Fraction of total records (|D|) that satisfy the antecedent (n_{cover}) of a rule
  - For this example (4/14)
  - $n_{cover}/|D|$
- **Accuracy** of a rule:
  - Fraction of records that satisfy both the antecedent and consequent of a rule $n_{correct}$, over those that satisfy the antecedent n_{cover}
  - For the example 4/4
  - $n_{correct}/ n_{cover}$

IF *director = dir2* THEN *Likes= yes* **Coverage = 28%, Accuracy = 100%**

So, 4 records out of 14 are for director 2. Now come to the accuracy of a rule. Now fraction of the records that satisfy both antecedent and consequence of a rule over those that satisfy the antecedent. So now here our consequence is likes equal to yes. Now out of those 4 records in how many likes equal to yes? In all 4 likes equal to yes? So, we have accuracy is 100 percent.

So, even if it is a very good, now see we are going to talk about the rule selection criteria little late, but try to understand that accuracy is very high, but it is not even covering 50 percent of the its coverage is not that good, maybe it is not a if we have you may have better rules which has higher accuracy and higher coverage as well. Moving ahead as I told you we have seen how to generate rules from decision trees. So, in this case this decision tree we have already constructed in in one of the lecture. We are right now not concerned about how to get this tree and in fact, from this tree generating the rules also we discussed. So, how many rules? Depending on how many leaf nodes how do we reach reach the leaf node? Starting from here to here to here one rule, from here to here another rule, here third rule, fourth rule.

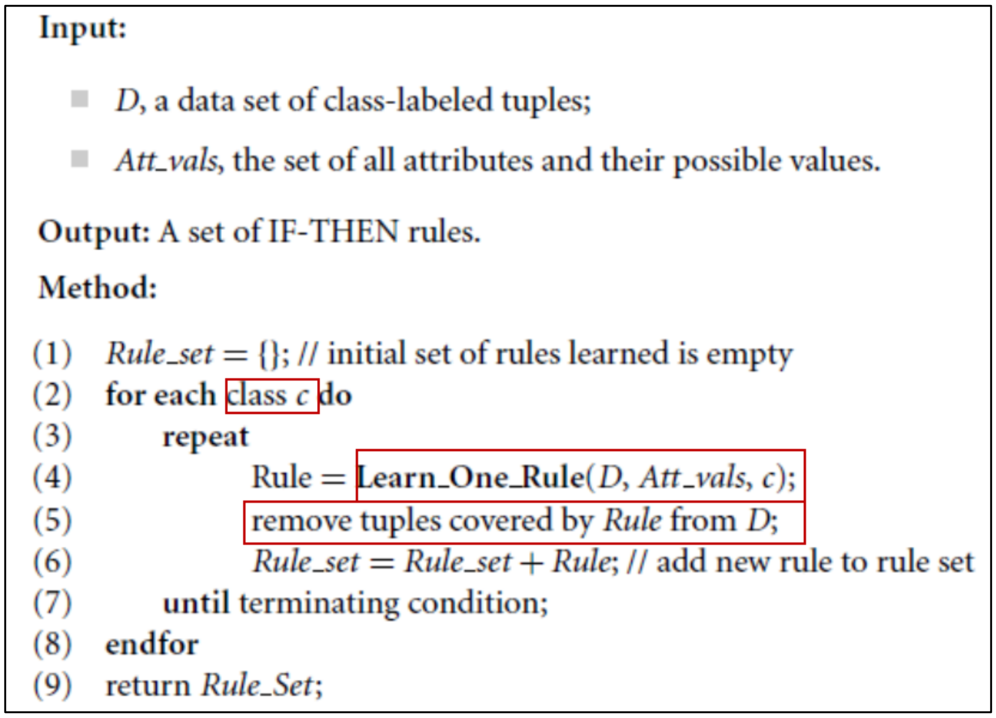
So, total 5 leaf nodes are there and we got 5 rules. So, one rule for each leaf antecedent contains the conditions for every node on the path from the root to the leaf. Hence consequent is the class assigned by the leaf straightforward this is a straightforward method, but the rule might be overly complex. So, if you cover all the paths this is a small tree, but the tree can be very depth can be very high in that case the rule may become very complex. What is the rule here? If director is dir 1, if awardee is no, do not recommend or he is not going to like the item.



So, these rules also I did not write, but these rules you can also write down. So, these rules we have already seen last class last rule I mean the in the class when we were discussing about decision tree. However, the rules that are getting discovered from the decision tree first of all require the decision tree to be constructed, then we know that there are certain issues with the decision tree. What is the issue? It partitions the data and because of this partition you may lose some of the variables. For example, here in this data set we had 4 variables like language was 1, but language is actually missing here this variable is now gone.

We cannot have any rule corresponding to that language, but it may so happen that the user is likely to like the likely to like the Hindi language movies. There are other issue as well the rules may become very complex. So, to avoid such situations we can also directly generate the rule from the data set and the algorithms which use which directly discover rules from the data set are called sequential covering algorithms. So, there are in fact, many sequential covering algorithms we are going to discuss the simplest of them. Now, what it says if then rules can be extracted directly from the training data without having to generate the decision tree first.

The idea here is that the rules are learnt sequentially one at a time where each rule for a given class will ideally cover many of the classes topples. The rules are learnt one at a time each time a rule is learnt the topple covered by the rule are removed and this process continues. So, now, this sequential learning of rules that is learning one at a time is contrast to the decision tree. In contrast to decision tree based rule generation why so? Because here one rule at a time we are learning, but in case of decision tree first we construct the decision tree then we learn all the rules at a time. So, the learning is actually simultaneous whereas, the learning is sequential in this context.



Input:

- *D*, a data set of class-labeled tuples;
- *Att_vals*, the set of all attributes and their possible values.

Output: A set of IF-THEN rules.

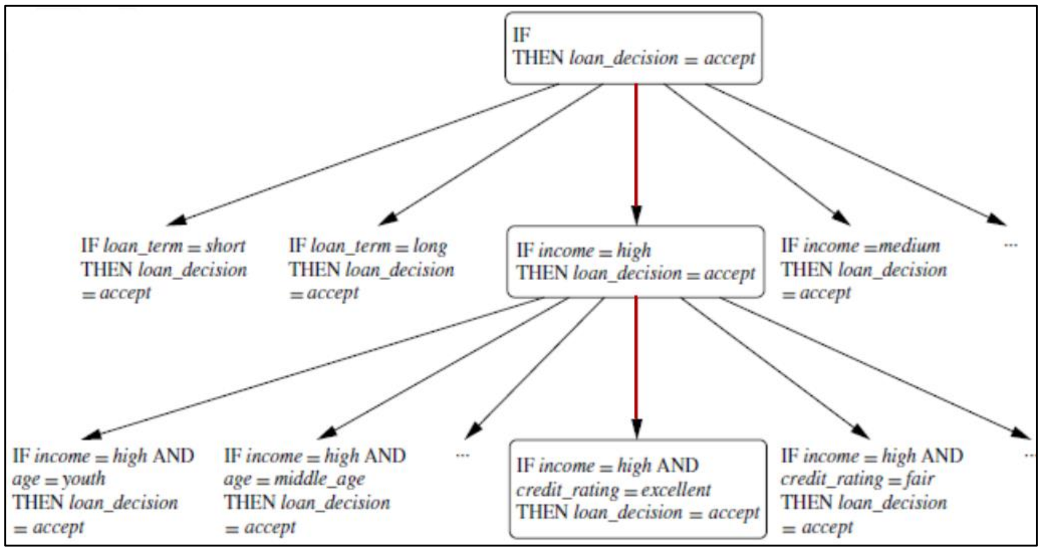
Method:

```
(1) Rule_set = {}; // initial set of rules learned is empty
(2) for each class c do
(3)     repeat
(4)         Rule = Learn_One_Rule(D, Att_vals, c);
(5)         remove tuples covered by Rule from D;
(6)         Rule_set = Rule_set + Rule; // add new rule to rule set
(7)     until terminating condition;
(8) endfor
(9) return Rule_Set;
```

So, this is a very primitive sequential covering algorithm, but this can tell a lot of things about the sequential rule generation. So, input to this algorithm is the data set of class labeled topples. So, what is the data set if class labeled topples? This entire data set with the features and response variable. Then attribute values are the set of all attributes and their possible values. So, what are the attribute and values? In our case attribute is director.

What are the values under this attribute? dir1, dir2, dir3. Attribute is language. What are the values under this? English, Hindi and others. So, those things are the input. Now output is of course, a set of rules. You start with an empty set of rules. You start with an empty set of rules. Start with an empty set of rules and take one class at a time and learn. So, in context of our data set how many classes are there? Two? Two? Yes, and no? So, start with yes, discover the rules, then repeat the whole process for class equal to no. Then one rule at a time you learn.

After you learn the rule, how we learn the rule that we are going to see shortly. Then you remove the topples which are covered by d. Then you add this rule to this data set. Now, how do we learn the rules? The rules are grown once again in a tree kind of manner, but in a general to specific manner. We can think of this as a beam search where we start off with an empty rule and then gradually keep appending attributes to test it.


IF
THEN loan_decision = accept
IF loan_term = short THEN loan_decision = accept
IF loan_term = long THEN loan_decision = accept
IF income = high THEN loan_decision = accept
IF income = medium THEN loan_decision = accept
...
IF income = high AND age = youth THEN loan_decision = accept
IF income = high AND age = middle_age THEN loan_decision = accept
...
IF income = high AND credit_rating = excellent THEN loan_decision = accept
IF income = high AND credit_rating = fair THEN loan_decision = accept
...

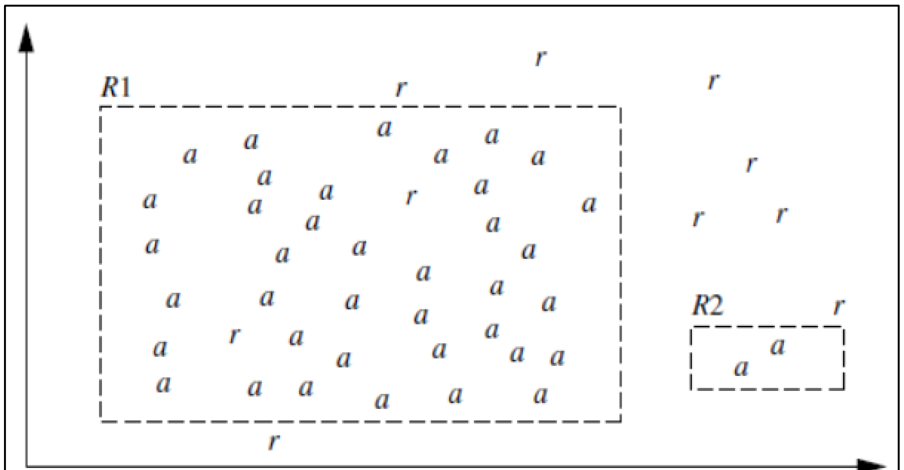
We append by adding the attribute test as a logical conjunct to the existing condition of the rule. So, this is a new example let us say we are considering. Suppose D is an training set where each attribute each record has the attributes like applicants age, income, education level, residence, credit rating and so on to recommend whether he will be getting loan or not. So going sequentially first you start with the decision. So the first make this antecedent part blank.

This part is there is nothing under if. Start with the then condition. Then condition is accept. So as per the algorithm start with the class. So you have the rule where there is no antecedent part. Then you keep adding the antecedent. So depending on how many variables you have how many attributes have that many you can have loan term equal to short then loan decision accept loan term equal to long. So in our case in our example if we go ahead with our example what will you do? First you will be constructing the rule without antecedent considering this. Then let us say you will be considering the attribute first attribute that is director. So you will have three branches over here director 1 if director equal to dir 1 then likes equal to yes.

Here dir equal to 2 like equal to yes. So this way you will be generating large number of rules. So this is what it says. Now suppose learn one value what is this learn one value business? Learn one rule at a time. So for each value sorry learn one rule at a time so for learn one rule finds that the attribute test income equal to high best improves the accuracy of the empty rule.

We pretend it to be the condition so that now this part which was empty is now added with income equal to high. So now what we mean to say you can check with everything but you have to have some kind of rule selection measure which say if you add this as your antecedent with the predesigned consequence which is equal to actually the class level. What is your performance measurement criteria? So we have already seen two things one is coverage and accuracy. So at least for them we will be checking and if it is satisfied if it is a good one we keep otherwise we remove we do not consider this further. Now suppose in this context this turns out to be good.

Then along with this what all can be added we have added income equal to high there are other variables as for example A is equal to youth, A is equal to middle age, credit rating excellent, credit rating fair and so on. Out of that again we find out taking these two things together we have good number of records where accept is equal to sorry loan decision equal to accept. So we keep so now the question is how do we measure the rule quality? Here we are showing two rules R1 and R2 and there are two classes that response variable A and R this small a and small r. Now suppose rule A covers this many number of topples through this dotted line represents rule A's coverage and this dotted line shows rule 2's coverage. Now if you look at R1, R1 is actually covering 40 topples.



"*a*" to represent the tuples of class "*accept*" and "*r*" for the tuples of class "*reject*."

So coverage wise it is very good. But here out of so many it is covering only let us say 2. Here it is 40 here it is 2. So coverage wise this is good. Now come to the accuracy. Now coming to accuracy here it has cover 2 and both in this case it has correctly identified. So the accuracy is 100 percent. But if we come here there are two places which we have this value R. So within this rule we have two such patterns where R is equal to where the class is something other than A. So which means now out of 40 only 38 are correctly classified. So your accuracy here 38 by 40 that is almost 95 percent. So which means now with accuracy and coverage because we do not need this rule.

This rule is giving very few is based on very few example patterns. So we have to decide which while deciding which rule to keep which rule to purge either of them alone cannot decide. So accuracy and coverage alone individually may not be good measures for evaluating rule quality. So there is a need for a measure that integrates aspects of accuracy and coverage together. There are in fact many such and one of these you already know. So entropy may be used as one of the method. Now suppose for one particular rule you have a data a portion of the data D is covered and with respect to this small part of the data you find out the entropy. Entropy about entropy about its formula etc. we already know.

So we have to find out entropy. So find out entropy with respect to rule 1 entropy with respect to rule 2 and higher the entropy better is the rule. There is one more called the foil gain. It looks at the positive and negative topples covered by a particular rule and it goes by this formula and it favors the rules that have high accuracy and cover many positive topple at the same time. So these are the references that we have used for rule based classifier and specifically this PPT also helped me a lot. So these are my concluding remarks on this lecture.

$$FOIL\_Gain = pos' \times \left(\log_2 \frac{pos'}{pos' + neg'} - \log_2 \frac{pos}{pos + neg}\right)$$

Rule based classifiers classify records by using a collection of if then else rules. Rules can be generated from decision tree or directly from the data set using some sequential algorithm. Here also I told you we can also generate the rules from the association rule mining. However, that was also a rule based system but that was in the context of collaborative filtering where we are using only the rating matrix. But here we had a situation in which we had to use information from two sources.

One is from the item side; one is from the rating side or the feedback side. So our choice for such situations was to generate the decision rules directly from the decision tree or directly from the data. So here in this process we may end up generating very large number of rules. So therefore, we have to purge, we have to remove many of the rules. So for this purpose we are supposed to find out the rule quality.

See this rule quality based on only accuracy and coverage is not sufficient. So therefore, we have to find out certain measure which can give better insights about the rule quality which may even combine this accuracy and average concepts. So therefore, in this context we also saw some two such matrix. Thank you.