NPTEL

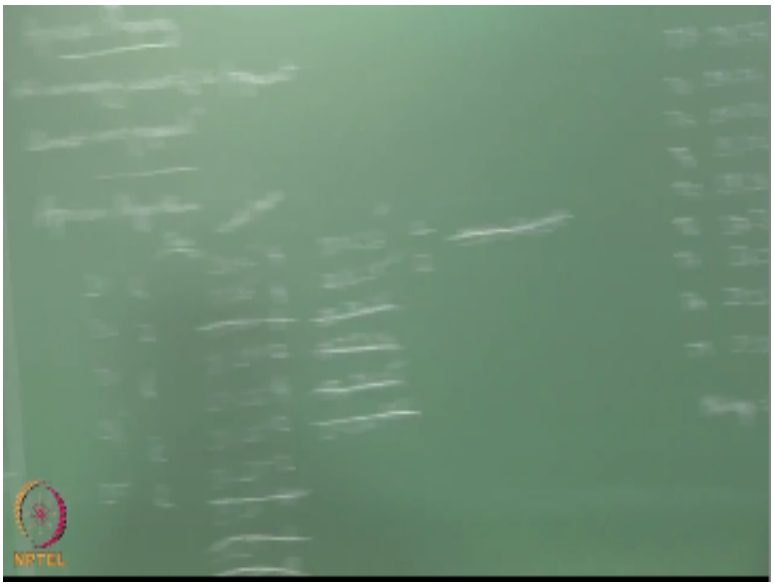
NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture-82
The Apriori Property

Prof.Balaram Ravichandran
Computer Sicence and Engineering
Indian Institute of Technology Madras

(Refer Slide Time: 00:19)



Great, so all of this is fine so now we will move to how do you find the frequent item sets so we will make use of something called the a priori property right so what is a priori property very simple so you use this idea for pruning the candidates that you will generate well you want to count right so most frequent pattern mining algorithms go like this right so you start off with a database of transactions right and then you generate candidate item sets.

Okay these are all the possible item sets that could be frequent okay then you go count them right and then based on the counting you can prune away some of those right if you are doing this blindly suppose I give you a universe of five elements how many item sets can you generate. right so I can generate two to the power of 5-1 candidates then I have to go count all the two to

the power of 5-1 candidates I have to count right if five is fine what if you are Eva's on at2 to the power of 10 million - 1right.

So that is not going to work right you need some very strong way of pruning these things right in fact whatever I am going to talk to you about today will not work if you are Amazon you need even stronger ways of pruning things there are other techniques for doing it so based on this a priori property rights so people propose something called the right so a priori algorithm is not the only one that uses a priori property a lot of frequent pattern mining algorithms used a priori property right but there is a specific algorithm called the a priori algorithm right.

So what I am going to do I am going to talk to you about two different algorithms for doing this I will just do this by illustration just start take a database of transactions and then I will walk you through the steps for doing this doing accounting so I just write down the database here and may I am going to say that I require main Sub Pop winds up of two.

So what I will do with a priori the algorithm proceeds in passes okay in pass one what will I do I find out the prequel I am frequency of all one item sets right I find out the frequency of all one item sets so what will I do here okay right so we do this is the counts right now what we do is I do any kind of pruning I want so I throw away all the one item sets which have below the minimum support threshold right so I throw away I six i7 all that right.

So none of this will get thrown away right so this will all be there now what I do is for people who know databases are people doing databases right I do a self join right and generate candidates for people who do not know cells join I basically extend all these patterns by one all possible extensions right so why is join and interesting your thing because join has been highly optimized by the database community.

So if I just say go ahead and do a joint okay I can compute it much faster than saying that okay go do a sequential scan of this and for extend each pattern by one right sand I am assuming that everything is commutative so I do not have once I do I1 I 2 I do not have to do I to I 1because they are all sets these are all the candidates for the phase two right so what I do now is I can do a pruning what is the pulling I will do.

So remember I am going to use the apiary property so this has to be frequent right all that subsets have to be frequent so what are the subsets of this well I one right to it turns out that

since I generated this join from onetime set table which are all frequent so all of the subsets will be frequent so in this case I do not have to do any pruning right I will just count so the counts for these.

Ok these are the cons for this now I can do running so anything that is not frequent electrode suit other things that are not frequent that that so all these have gone right so what I have done here I have done a count and then I did a prude right now what do I do again whatever is left I do a I know ourselves join our for non CH people I extend it by one more right provided I 1 and I 2 I mean so the first elements are common right.

So for example I can do a I 1 I 1I 2 I 3 right so I cannot do I 1 I 2 I for I can extend I 1 I 2 by adding I for because I do not neither have I do not have an I one I for right why do I have even if I have stuck it off no I need the first elements to be common when I do cells joint I need the first elements to be common so if I want to do an I one I to I for I need I 1 Ifor here so they do not have that I cannot do the I one I to I for joint.

Ok so this mean additional join gives me an additional pruning and taking advantage of my set property that the order does not matter so I am doing an additional floating so likewise I can do I 1 I 2 I p I 1 I 3 I 5 then and I can do I can do these 6things I1 I 2 I 3 I 1 I 2 We- 5 1 I 3 2i 3i for its basic light 2i3i for i2 i3 i5 I to i-4 i-5 these are the six elements I will get after I do the joint right now even before I count I can do some pruning how.

So I look at this right so can I prove this no I cannot prove this icon from this but Ione i3 i5 I can prove why because the subset it is not frequent right even before I do the counting I can do the pruning this is where I use the a priori property right what about i2 i3i-4 can you prove I 34 is gone right so I can prove that what about this again I3fi is gone I can prove what about this I for if is gone I can put.

So all of this I prune even before I count right so now I only have two item sets of size3 2 3 item sets that I have to actually go and do the counting for will there be a join after this case I could do a joint after this and the joint will bei1 i2 i3 i5 first two will have to be common so I can do a joint so after this will do a joint which will be i1 i2 i3i5.

And will this be frequent our friend i3 i5 comes to our rescue right so this will not be frequent so I am done right so what are all the frequent itemsets1 2 3 okay so what is the big drawback with

apriority algorithm he said you do generate lot of candidates right and then you keep ruining them but even here even though you pruned a lot of the candidates without counting but you did end up generating a lot of unnecessary candidates right and then you have to go back and verify the a priori property for them and then prune them that is one drawback and the second drawback is in every face you scan the data all over again and you do you do the counting.

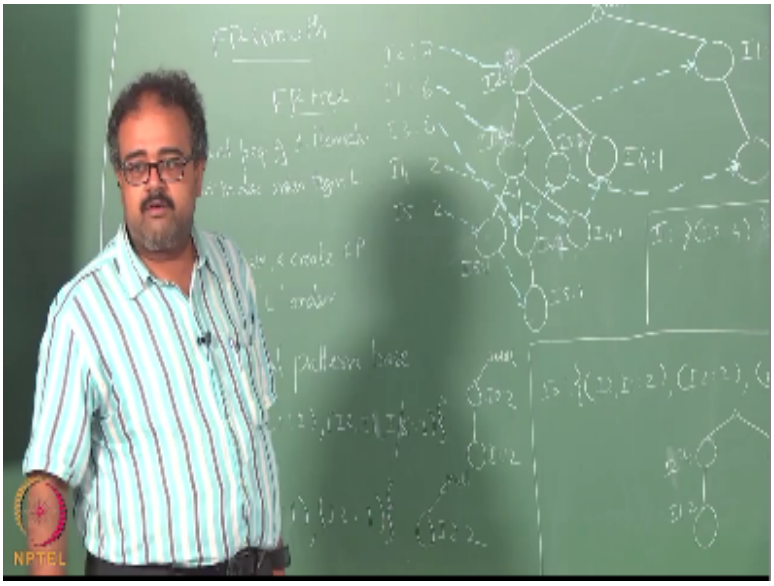
Right so when you counted this okay you do not somehow save the information for generating this count as well right so there are lot of newer frequent pattern mining algorithms that work on very large data sets which they try to do it with a single pass through the data they keep ancillary data structures around okay they do a single pass through the data and they are able to count for all the frequency you would think that you need at least two passes one pass to generate the candidates and one pass to do the counting.

There are ways of doing it without actually doing that a single pass algorithms are available but I am going to talk to you about a two pass algorithms now right which is efficient but everybody gets a player algorithm right it is fairly easy right great so let us do them to pass algorithm look at two pass algorithms we call FP growth in ad about F P stands for frequent pattern growth right FP grow than.

So it tries to avoid candidate generation right price to avoid unnecessary candidate generation and minimize the number of passes you go over the data right how does this accomplish this they accomplish this by creating a data structure called the FP three right so once you created FB three once you create at the FP tree you make many passes over the FE tree but it is a somewhat of a compact tree you know means like a lot of balance streets more like a kind of a tie like data structure if people know that but you do not have to know that right.

So you just go or the trees it is much more compact representation than going over the table all over again right so you know once you construct the tree the tree can stay in memory and you essentially go over that tree again and again right so let us do this step by step will construct the FP tree first again then from there Petrie will start generating this okay so what we do first phase is you go over the database once core the database once and then you count.

(Refer Slide Time: 16:12)



What do you think you are going to count the frequency of all the one item sets that you have to do there is no other go right it count right and you sort this by descending order so let us call this is this order as some L ok so the L is this ordering now what I have to do is go into each transaction okay and reorder the items by L right.

So what I do is okay you guys can help me out here so that will become right so whatever this fine this fine what about this has to change that is fine that is fine that is fine this has to change anything else has to change so three and five or ok and this again has to change right so I do not have to do a separate pass over this ok I can do this reordering as I do my second pass in the first pass through the data I County onetime sets second pass through the data I create the FB tree in the L order so L order is essentially each transaction.

I will do this now forever transaction I create a path in the tree right so I will start off with the route as some null okay nothing so no items that the first transaction is what so I will label this as i2 i1 Wi-Fi right it is the count how many times have seen this once what is the next transaction I to I for right so I start off with null set then I add I to then I had I for right so this is essentially that.

Okay what is the next transaction I too anything else of course a lot more I too I won so I to comes for I one becomes too I for becomes one next one is I 1 i3 again no not again the first time so what do I do right and then what we get I 2 I 3 so you can see that I can revote ordering

whenever I read the transaction I did not have done it in the first pass right so I can do this then where are we t7 is right andt8 is I to I won i3 i5then the last 29 I to again then right.

So we have our fp3 right so we have a ref Petrie so there is only one thing that is needed to complete this so for ease of navigation so I am going to have pointers from this table right so the where does I to start here okay likewise where this I one start here okay it is rests there is a second entry of I 1 this will connected then what about I three so I three is also connected like that then I for my god it starts looking very scary rate this is why I wanted the ok.

That is arrive III so essentially our constructed this fp3 by doing a seconds can over the data and if you think. About it so if you will take any path down this three okay the prefix of the path that is the things that come at the beginning of the path will tend to before frequent and the things that come at the end of the path is a whole idea of behind what we have done is this FP three things that come at the beginning of a path if I take a path.

From the node to them from the root to the leaf that thinks that come at the beginning they tend to be more frequent than things that come at the end so now what we are going to do is we're going to work from bottom up right and try to generate an auxiliary data structure for FP tree right from which we will generate the frequent patterns.

We can just read off the frequent patterns from this auxiliary data structure so what is what is it that we do we generate something called a conditional there is something called a conditional pattern base so we typically do this in the reverse order of our table here so I will take i-5right so I look at all the paths that contain I right and take the prefixes of that prefix of i-5 I will take all the paths where I 5 occurs and I take the prefix of that.

Right how many pass the sift occur here so I to I 1 I 5 right so that is one thing so I will just take the prefix so I 2 comma if won and it occurs once and then anywhere else if to I 1 I 3 I philet this I 2 I 3 I won sorry I one sorry to I 1 I 3 there is a conditional pattern base for I five so if took the things that k right now what I will do is I will assume that these are my only transactions and I will create an f e 3well.

I will assume that these are the only transactions I have and I will create an FP three but before I do that I will ignore all the entries in this conditional base which appear only once all the items one item sets that appear only once will ignore them right so I three appears only once the whole

thing right I one appears twice I to appear twice but I three appears only once right so I will ignore the occurrence of I three.

Now I will try to create a FB tree with the remaining two transactions so what are the transactions I to i1 andi2 i1 so my FB tree will look like right so I look at the path of a specific frequency so the path here I 1I 2 I one path has a frequency of two and I know that it is followed by five because that is how I selected these things right so the frequency of i2 i1 I five will be 2.

We already counted that we countered that by doing multiple passes of the data so now not only will this give me this if to I 1 I 5 it will give me any frequent pattern in which is a part of regardless of how large the items at this so it will give me not only the three item sets frequent item sets in which five is part of it will also give me the frequent to item sets in which if phi is part of right.

So then I can say that yeah so how do I read the two item sets that I phi is front of just take any prefix in this path and or any combination in this path right which has only one thing right so not only is i2 i1 I five frequent I won I 5 is frequent I to is also frequent and the frequency is too right so that is basically done so I have counted all the frequent item sets with effect likewise I can do this for.

So what are the frequent what are the what is the conditional pattern-based for I for so this is called the conditional FB tree ok this is the conditional pattern base and the HP tree I construct from the conditional pattern base is called the conditional fp3 and from the conditional tree I basically can read off the frequent patterns ok so what is the conditional pattern base for I for yes we have to look at wherever I for oversee that is why you need this data structure right go here follow that ok I for occurred here follow that back so I have I to I 1 I for occurring once so Ito I one occurring once.

Again follow that there okay now what can I do I can ignore Ione right under construct my FP three uses even simpler than this conditional fee 3 is essentially so the frequent patterns are I to I for right frequent patterns are I to I for basically no done all the candidates yeah so the nice thing is now I can ignore I for and I five right now I want to go to I three right so if you look at it so there is something beyond I three but I do not have to worry about it if this was part of a frequent pattern over already caught it.

When I went back for a correct so when I start now when I go to i3 and I can construct in the conditional pattern base for I three I have only have to look at the prefix above I three not what comes after I three so if there was anything it should have been part of that should have been captured already if it has it been capture I do not worry about it and that is where we go from I 52 I 2 in this case right.

So I three what is the base so start off here so it will be Three I 1 I 2 right so essentially I to come up I won and the count is two because I three-count was too right so the count should be two so far we have only had account of one okay but here the count will be to in anywhere else I three occurs here so that is basically I to anyone else I three occurs here how is the conditional FP tree look for this look like for this right so again I can read the frequent patterns of this.

So what is it so I to i1 I 3 with the frequency of two right I3i to with a frequency of for I 1 I 3with the frequency of two here two here so I 1 I 3 has a frequency of for i2 i3has a frequency of four I to I 1 I 3 has a frequency of two I can just read it off the tree okay so we can check whether that is correct so I 1 I 2 I 3 as a frequency of for I 1 I 2 as a frequency sorry too I 1 I 2 as frequency of for I 1 I 3 sorry this one is not done I 1 I 3 as a frequency of for i2 i3 has a frequency of four.

Right so whatever we counted but all the frequent item sets that contain i three or done in one shorter likewise you have to do one for I to know I am sorry I won so what will be I 13 what is the conditional pattern base for I will don't have to worry about that way because I know the frequency of one item set I know the frequency of i1 is already six I know that so basically this is the only thing I will get and so the tree will look like for what four times right so this path has been taken four times that's what this tells me so that means I to I one should have appeared four times so the prefix I to ending with the suffix I.

One would have appeared four times that is basically what the conditional pattern base tells me so the conditional pattern base fori1 is this I to colon for and the conditional if we try is just one note there is null and then I to colon for and the frequent patterns they yeah so the frequency of i1 i2 is for and that is what we get here I 1 I 2 is for we get that right and of course the 11 item set frequency is already given to you by the table do I need to do the conditional base.

For I to know does not matter right because I have taken all the other items so I do I do not have to actually the process separately right so what is the nice thing about this algorithm is that a I did not have to do a generation of a lot of candidates and then prune things down right so I had a way of traversing this three that just gave me the frequent item sets right plus one thing second thing is I just did two passes over the data all subsequent passes were done on the tree right on the tree is somewhat compact assuming patterns actually repeat right mean .

If the patterns do not repeat at all and every transaction is a unique subset then you are doomed wait so you will get a very large tree I mean that'd be slightly compact but still it will still be a large tree right but since patterns repeat typically right so you are same it is any questions on how this happens right so first you do one pass through the data right count the frequency of one item sets right sort them they do a second pass through the data construct the FP tree right put in all this navigation links.

Right and then for each item one item set construct conditional pattern base and then construct a conditional fp3 and from that you can read off all the frequent item sets that contain that item make sense any questions on that good point so I have done I have done this I fire right so I basically I computed the conditional pattern base right so if you look at a three right it occurs only once in the entire conditional pattern base right.

So I can pull it off so now my conditional pattern base will actually become I to if 1 colon 1 and then I to I 1 colon 1 so I will just create my three based on that sorry that that is decided by means of yeah let us decide it by min self so whichever occurs less than min sup number of times in this will remove it because it cannot figure in a frequent pattern so that the one okay you're confused about the one is it not need some in some anything address less than winds up because missteps is to hear.

So it just becomes one so anything that occurs less than windsurf number of times so again we can rework the whole thing setting winds up to three right and then you would see an interesting things right. So I mean all of this will go there will be no frequent pattern of length three right so both of this T I fey for will gorightI for I Phi 5know we'll all go I forth yeah that is correct Rick, so I for fail all go there will be no pattern set feature I for a nice five in fact they will go from here this cable itself.

Because they occur only twice so you only have three entries so we start off with a frequency of three you only have three entries in this table and so you are everything becomes simpler right okay that is a good point so if you if you start off with only three entries in this table when you collect the FP tree itself so you leave out I 5 and I for from the transactions there so these entries will not even be made this idea for entries will not even be made at what you will do is if some something get dropped out.

Because the one item set is not frequent right I will delete them from the transaction order also so I have sorted them in the decreasing order of this table right and whatever is below the threshold I just deleted so t1 will become I to I 1 wait and t2 will become just I to, so that will not even figures the I pit and then the FP tree construction ok any questions I bet any applies to Nationals yes there are algorithms that use hashing and in fact take it back.

So the even the algorithms abuse hashing actually give you the right count but there are algorithms such that give you an approximation there are so if so many efficient algorithms that give you exact counts nowadays that use hashing lets the I do not know if you should be using approximation but there are other approximation algorithms for this see the more interesting research question to ask now is what happens if I am not just counting elements from not just counting subsets if I got them counting data with my additional structure in it wait so that is a more interesting question to ask.

IIT Madras Production

Funded by
Department of the higher education
Ministry of the human resource department
Government of India

www.nptel.ac.in