NPTEL

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

Lecture-73
The BIRCH Algorithm

Prof. Balaraman Ravindran
Computer Science and Engineering
Indian Institute of Technology Madras

So the idea is I want to cluster really large I want a cluster really large data sets right very large data sets and the way I am going to do it is the following right so I am going to do a very rough clustering at the beginning right I am going to do a rough clustering at the beginning where I am going to produce really tight clusters right I am going to produce really tight clusters right but many of them and i produce a lot of small, small clusters.

So essentially taking the suppose I have a million data points will reduce it to some 100,000data points but each one of them will be very, very small diameter clusters right I produce these things and then what do I do I take one representative point from each cluster and then I try to do my hierarchical clustering on that so initially I do something that is little fast end the fast and dirty right.

So I might get the correct clusters I might get a little bit off here and there but, but then I do a second pass through the data and I will fix everything right so the way I do this is follows I do one pass through the data produce some rough clusters right and then I take the centroids of all these clusters right and then I do a proper clustering algorithm we only work with the centroids I do not worry about the million data points I reduced it to10,000 centroids 100,000 centroids.

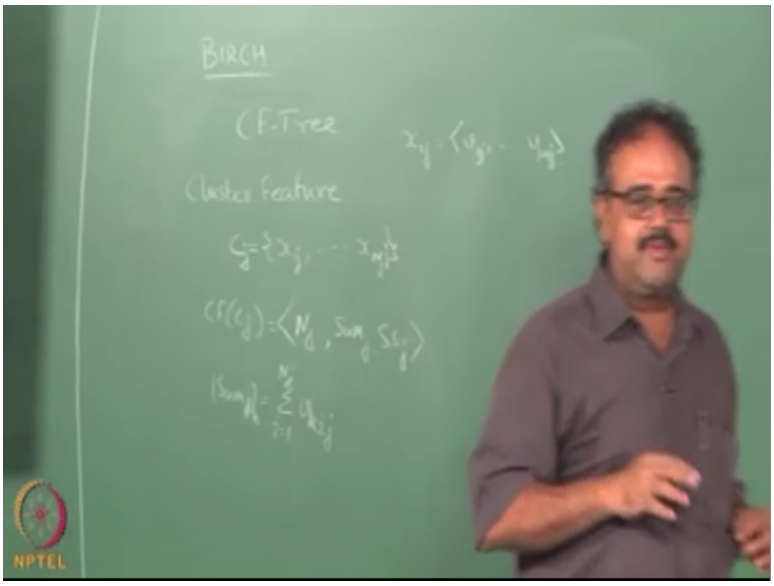
Whatever I only work with these 10,000data points so I know I know small enough that I can fit it in memory right memory is so large nowadays you can fit a million data points in memory but let us scale it appropriately right you have a billion data points and you are notable to fit all of that in memory then reduce it to something small that you can fit in memory right so now what you do after you have done the clustering right.

You will have a new set of centroids for all your clusters right now what you do run through the data once more and I sane each data point to the closest centroid so essentially you are making two passes through the data once to produce clusters which are very tight right and then you take the representative points and then you do whatever you are you can do an iterative clustering algorithm on top of that so once you are finished with that so what you do you go back and reassign the data points to the nearest centroid.

So the two passes through the data so that is one of the things that they should think about when you are working with very, very large data sets how do you minimize the number of passes through data right so if you can do it with one pass through the data grade if you cannot we try to keep the number of passes minimal right if think of k-means rated very plain the k-means how many passes are you making through the data as many iterations.

As there are so every time you do an iteration you actually need access to all the data points because I need to compute the distance of each and every data point to the centroid so I need to read the data all over again right so with a very, very large data just going through that might be a plain if I cans tore everything in memory right so if I want to read the data set all over again from the disk then it becomes a plain.

(Refer Slide Time: 03:26)



So we need to have something better than that right so what they came up with is a data structure called the C F tree came up with a data structure called these a of tree where C F stands for clustering feature it is C F stands for clustering feature or cluster feature right so what is the cluster feature it is a three tuple.

Suppose I have a cluster right that consists of some endpoints like right so I will denote the cluster by see if the cluster C consists of some end points I will denote it as x_1 to x_n right in fact I should do something more than this I should say probably cluster j consists of cluster j consists of Nj points which I will denote as x_{1j} to x_{nJJ} so it is consists of NJ point so this is the crystal so the clustering feature corresponding to the cluster J will be number of points in the cluster okay.

And the sum of the individual coordinates of the data points suppose my x_{1j} is consists of some b_1 up till Vp right will be too many indices but so it is a point my x_{1J} is a point in a P dimensional space right so I essentially each one of them is a point in P dimensional space right so this will be essentially so my ∑ is call the sum j.

And the I will come to that in a minute so some j sum j again a vector wait some j is a vector and okay so some gate coordinate of some j essentially the sum of the gate coordinate of all the data points some of the coordinates in individual coordinates some j is a vector sum of the vectors here that is my data points as vector concerns and some of the vectors the, the reason I did not do that as some of the vectors is sometimes people have been confused in the past.

When I read some of the vectors they end up adding all the coordinates I did not I do not want the ∑ also to run over k right so basically if you think of proper vector addition it is just the sum of all the data points right but this notation is convenient for us when I want to write they will take the square of the individual coordinates and add them up so these are this is what I call the clustering feature.

And the claim that they make in the birch paper is that this clustering feature is sufficient for you to do some form of hierarchical clustering without actually not knowing the identity of the data points and suppose I give you the clustering features of two different clusters right is there any way you can compute the distance between the two clusters I give you the clustering feature of two different clusters.

I give you see FJ and CF I right can you compute the distance between cluster I and cluster J I give you the sums the squared sums as well as the number of data points so if I divide the sum by the number of data points I get the centroid right I can go ahead and compute the distance between the centroids that gives me one way of measuring distance between clusters that I do not need to know the individual data points all I need to know is the sum of three vectors that is one thing it said anything else I can do I am sorry I have some square also we can use that I can do the radius right.

So I am going to leave that tears of homework okay show me that you can compute the radius in terms of some and SS right so I can I can actually compute the radius of the merge cluster also right based on the sum, sum of the squared sums of the coordinates right so these are the things that I can do I can even do the diameter ok so I will leave you towards that out so you can do this you can do the centroid distance you can do the radius.

And you can do the diameter distances you can't do a single link complete link ok so that is gone right but remember what you are doing at this stage is trying to the initial stage with the CF three is trying to find small clusters the Custer's of small diameter so that is what you are interested but I am not interested in doing single link or complete link at this point I what I am trying to do when I use the CF trip in a bill.

The CF try I will tell you how to build the CF three a minute so what I am interested at this change is only in data reduction I mainly interested in doing data reduction I am not really

interested in actually finding the final clusters right so it is fine so we can use either the centroid measure or the diameter or the radius and then keep some kind of a threshold right so what do I do now is I start off from the root ok so the root is going to have so one cluster right.

So you going to have a clustering feature that is going to be there so as, as and when the data points come in right I start updating the clustering feature at the root right the first data point comes in the clustering feature will be1 comma the data point comma the squares of the coordinates of the data point that will be the first one okay then the next, next data point comes in then what do I do 2,the sums of the two vectors right plus the sums of the squares of the coordinates.

But I do this only if the two points are close together and if they are too far away if the diameter is very far-right very large then I do not merge them together I actually make it into a different cluster and I actually do the clustering feature there right so I keep doing this right I keep going until I come to some point where I have too many too many clusters that I have thrown it at this point right.

So essentially I start breaking the breaking the cluster into two breaking this leaf this node into two so some clusters will go here right and some clusters will go here right and what do I do in this place I insert a new clustering feature that summarizes all these data points enter a new clustering feature that summarizes all these data points so why am I doing this so when a new data point comes in right.

I look at these two clusters and figure out which of those two clusters is closer to this data point ok and then root it down the trees ER right so if I keep all the clusters at one level then every time a data point comes in I will be doing a very large linear scan to find out which cluster the data point should belong to does it make sense and suppose I suppose I start off with one cluster then I produce ten clusters.

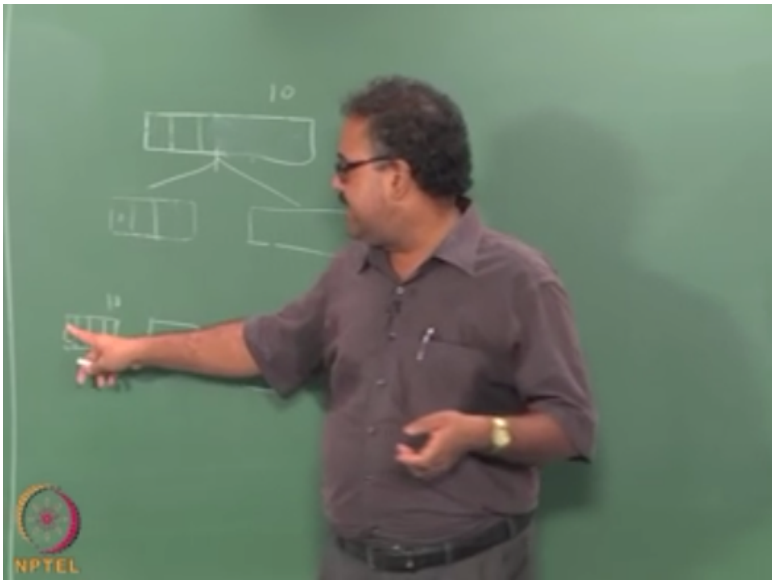
Now a new data point comes in I have to check against all the ten clusters to see which is the best cluster to put the data pointed right now what I am trying to do is organize this in a more efficient manner so that instead of taking order of 10 right it is going to take some kind of a logarithmic factor here when I have to make two comparisons will take order of two now depending on how many I put in here.

So I suppose I divided into two and then put five here five here and we will take order of six right but then it is a tree becomes deeper then you can see the savings right the savings will be tremendous so it is you have some kind of a logarithmic compression so that is essentially what we do here yeah okay so initially what I do is I have 11 node okay.

So I start keeping track of all my clusters here right suppose I reached in clusters let us say 10 is the maximum it I reach 10 clusters a new data point comes in I want to create 11 the cluster sorry various factors it depends on what is your memory size right so and in fact you have to go even further down actually so it depends on your page size not just on in memory size right want your each leaf to span multiple pages right because a single axis could actually lead to more page faults right.

So you want to see if axis one entry in a leaf I would really like the entire leave to come into the and stay there so there are all these kinds of consideration when you start talking about really large data right you have to start going into the system level of things right so you have to know how things are stored you have to know how things are accessed right, right so here let us say I pick 10 for somehow right I said 10 is the number I am going to store and what are these numbers.

(Refer Slide Time: 16:05)



These are essentially three things like this right each one of these only three things like this at some point I say okay even if have a 4k be page even like, like 100 clusters I should run out of

space right I will say no, no I want to break it right so then what I do is I split this and I put 10 here and I say another 10 here I am sorry put a 5 here put another 5 here right and then I will start off with two entries here right instead of having 10 entries it all go down I will get two entries here right.

And each one of them will be so the first entry will be summarizing the lab the first child the second entry will be summarizing the second child right now a new data points come in I can try to shove them into this right and then at some point it will come to a point where this thing gets filled up then I will split it into two right so I can do one of two things I can actually push it further down right or you can split it into two and add another entry.

There which is better not by this like 10 rate I mean I have 10 I have space to store 10 CF features there right so I could I could either make it broad branching at all I could make it deep which is better rod it better be bleep deep is better draw this better so if you choose to go broad when do you go deep then what gets filled up when this gets filled up and then you only push it down from there is it no, no yes right that is ready to do it work done.

But do not doubt yourself what you said was correct okay I just wanted to see how sheer you are up your answer so essentially you keep broadening it right and this gets filled up then again you split that everything goes down one level okay so keep doing this and until you have gone through all your data points so that the small point here is to notice I am not doing a numeric example and if people are interested in a numeric example of how birch works.

You can refer to the birch paper so we will put up the birch paper online right we will write yes we will put up the best paper online okay so what we what will happen is suppose a data point comes very, very beginning right so it will get associated let us say with this cluster so and for that cluster I compute the clustering feature added to the clustering feature in that cluster right but then later on as I keep going down more and more data points will come here.

And some more data points will come here and so on so forth it is likely that if you if I try to insert the data point again into this class three right it may go into some other cluster because things move so there is this significant order effect right so sometimes what people do with the birch is after they grow on the tree to some point they stop okay and then they try to do some kind of rebalancing okay that is just an audition optimization step.

You need not do the rebalancing step you could just grow the CF tree in one shot right more often than not it will work so the end of the clustering what you will have is you will have a lot of right lot of leaves at the bottom and each one of them containing say ten clustering features right and these are your final clusters that you want reservation or these your final clusters know right.

You remember I told you the first phase when I am constructing the clustering features the sea of tree all I am interested in finding on tight clusters that I can then use as representatives for doing a further clustering right so what do I do now is go into each of these clustering feature compute the centroid I can do that right divide some by NJ so corresponding to each entry in each one of the leaves can get one data point which is the centroid right.

And then I go start with these centroids and do a clustering all over again here I can do whatever I want I can do single link I can do a complete link whatever so because I reduce the number of data points is something small and manageable right I can do whatever clustering method I want on these data points right and then what I do is the final centroids I get after I do that clustering I go back and look at the data all over again right.

So I can the data once when I build the CF three right now is can the data again after I finish my clustering ok and then assign the data to the nearest centroids you just need a new point in you can just as any recording video now I need to cluster the whole data right I need to know what cluster you belong to right so I need to know that right so like I said I cannot trust the identity here.

So initially I said okay this clustering feature got the data point I now so the same clustering feature would have percolated down to the leaf but I cannot say that okay this clustering feature whatever cluster it goes to that is the cluster that the data point belongs to because that would have drifted away this is the centroid could have drifted due to the later data points.

So I cannot do that is not the, the, the, the mapping between the, the CF the clustering feature and the data point is not, not static it would have changed or drifted away. Therefore I have to do the second round of assignment people understand birch say I really like birch you know because it is a very simple algorithm and it also allows you to do handle very large volumes of data I usually ask people the implement bursts in one of the programming assignments okay.

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in