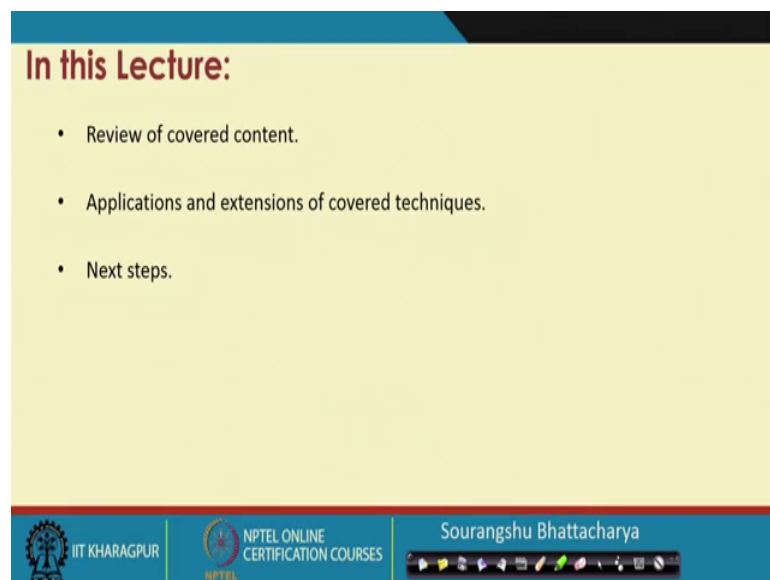


Scalable Data Science
Prof. Sourangshu Bhattacharya
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 24
Conclusion

Hello all welcome to the 24th and last lecture of NPTEL course on scalable data science. So, today we are going to conclude the course so, we are not going to learn anything new, but we are just going to review.

(Refer Slide Time: 00:39)



In this Lecture:

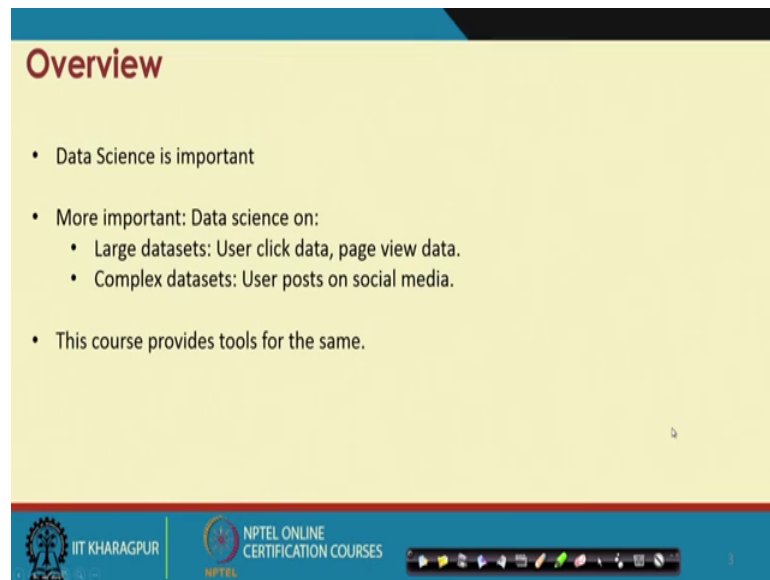
- Review of covered content.
- Applications and extensions of covered techniques.
- Next steps.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Sourangshu Bhattacharya

So, so we are going to review the content that we have covered in this course, then we are going to discuss some applications in the context of each of the things are each of the topics that, you have covered in this course and also the extensions to the power techniques. And then, we are also going to discuss some of the next steps, as you all are probably aware that this area is very new area.

So, much of the subjects covered in this course have been developed in the last 10 to 20 years and many of the topics are still under development and many of the topics are still finding newer and newer applications. So, many of the topics covered in this course are kind of actively under research so, we will see some of the next steps, in some of the cases that that the community is taking ok.

(Refer Slide Time: 01:43)



The slide is titled "Overview" in a red font. It contains three main bullet points. The first is "Data Science is important". The second is "More important: Data science on:", which has two sub-bullets: "Large datasets: User click data, page view data." and "Complex datasets: User posts on social media.". The third main bullet point is "This course provides tools for the same.". At the bottom of the slide, there are logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES, along with a navigation bar.

- Data Science is important
- More important: Data science on:
 - Large datasets: User click data, page view data.
 - Complex datasets: User posts on social media.
- This course provides tools for the same.

So, as we all know data science is an important subject in today's world, because all the decisions, all the actions everything is now becoming data driven. And we are sort of in digital or data driven age and also you know everything is connected so, that also helps in doing data driven decision making.

So, but what is now become more important is that the size of data sets have become much larger, again due to advances in computing technology, advances in sensor technology, networking etcetera. So, earlier maybe there was people use to deal with kilo bytes are at the maximum megabytes of data sets even, you know 10 or 15 years back people use to typically deal with these sizes of data sets.

But, now it is very common to deal with gigabytes of data sets ok. So, it is now become very important to be able to do all the data science that, you want to do on very large data set. Another aspect of modern data sets is kind of complex data sets ok.

So, for example consider the data set of user post on social media. So, so you know earlier the data sets used to be somewhat structured data sets right so, every for every data point there are these fields. So, for example, for every user there may be a post and there maybe users age etcetera, but now considered a user post let us say on Facebook ok. So, the post may have some text, but the post may also have something like a timestamp ok. So, and then the posty may also have some images and the post may be directed at few of the friends of this user ok.

So, all these things are very different types of data, so, the first type is the unstructured textual data that people are putting ok, where you have to use some kind of natural language processing techniques, or similar techniques to process the data.

Then there is the image so, images is another kind of data which is again some kind of unstructured data, but you need to use a different set of techniques something like and image processing techniques for analyzing that kind of data. Timestamp is a single number for each post, but it induces a dependence on the data, so, for example a post which has a larger timestamp than another post came after that post.

So, this kind of a sequential nature of data appears, because of the timestamp also how close, how much time after a particular post did another post appear ok. And finally, the post maybe the so, this the previous type of data is called temporal or sequential data ok.

And finally, you know people may referred their post to some of their friends or they may mention some of their friends are depending on the platform, you know they may quote some other post from some other friends ok. And this is a very different kind of data, because this not only tells you about the person who is posting, but also the people the other people who made have may have made their own post and so, on and so, forth ok.

So, all this is making the data sets more complex and interconnected. So, there is no you know the data sets are becoming multimodal so, there is image text then the data sets are interconnected. So, posts of one user is connected to post of another user, but all this things are coming up.

So, both these aspects actually calls for a new set of tools so, you cannot so, you can only use the old set of tools to process this data to some extent, but if you want to take full advantage of this kind of data ok, which are large and very complex data sets. Then you have to learn new set of tools and in this course what we have tried to do and people have realize this and people have developed these new set of tools over the past let us say 10 or 20 years so, past few decades and this set of so, in this course what we strive to do is to provide you and introduction with you know this set of tools ok.

(Refer Slide Time: 07:23)

Streaming Algorithms

- Computational setting:
 - Data arrives one by one.
 - Network, user input, Other devices, sensors, etc.
 - You are allowed to store only a “summary” of data.
 - Original data stream is too large.
 - Answer a “query” at any time.
 - Such as how many distinct elements have passed the stream.
 - Approximate answer is sufficient.
 - It is known that you cannot provide exact answer.
 - Exact answer is not important.

The slide also features a video inset of a man in a green and white striped shirt speaking. At the bottom, there are logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES.

So, the first class of tools that we talk about is the stream processing or sketching ok. So, the setting is here is that data arrives one by one ok, so, it could be because the data is coming over network, it could be because the data is a user input process. So, the data user is actually inputting the data one by one in a sequential manner something like user is posting his messages one after the other. It could be because the data is coming from other devices maybe you are you know now you are in a IOT world.

So, you are in a interconnect you are in a world where devices are interconnected and each of device these devices are running their own algorithms and so, you know you might be writing a program for your fridge, which is connected to many other devices like a cell phone your maybe power supply and so, on and so forth.

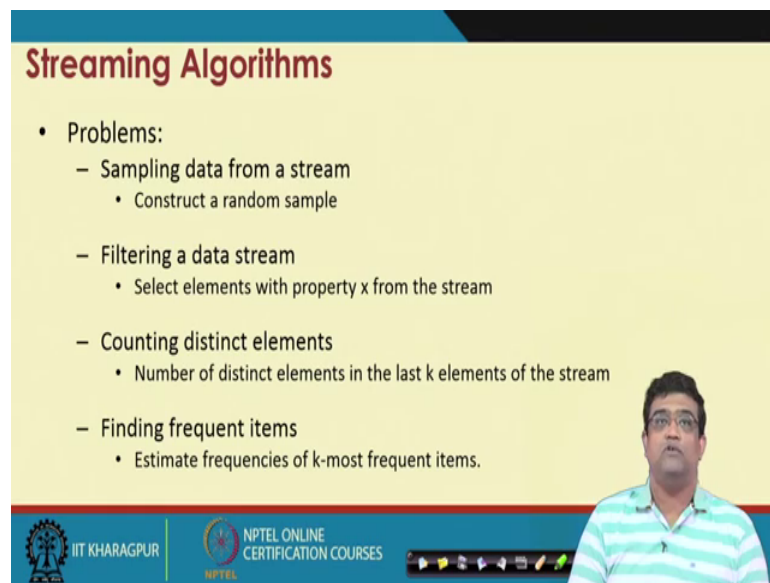
And those devices may be sending some data to your fridge ok. And then there are sensors of course, there are many a times data comes in a streaming manner from very sensors so, these are all sources of streaming data ok. So, in this setting basically you are allowed to store only a summary of the data. So, this is because mostly because the device on which you are processing the data a is not suffice does not a sufficient memory to store all the data. So, it is too large the data is too large and here, the task is to answer a query at any point in time.

So, the query could be for example, how many distinct elements have passed the stream. So, and the characteristic of this kind of algorithms is that are typically you know an

approximate answer to the query is sufficient you do not have to answer the query exactly you can give an approximate answers.

So, for example you may not so, one cases that you are also there could be two reasons one is that, you need not provide exact answer see only want rough estimate, or it could be known that with certain memory constraints you cannot provide the exact answer. So, both these are reasons for going for an approximate answer ok.

(Refer Slide Time: 10:07)



The slide is titled "Streaming Algorithms" and lists four main problem categories:

- Problems:
 - Sampling data from a stream
 - Construct a random sample
 - Filtering a data stream
 - Select elements with property x from the stream
 - Counting distinct elements
 - Number of distinct elements in the last k elements of the stream
 - Finding frequent items
 - Estimate frequencies of k -most frequent items.

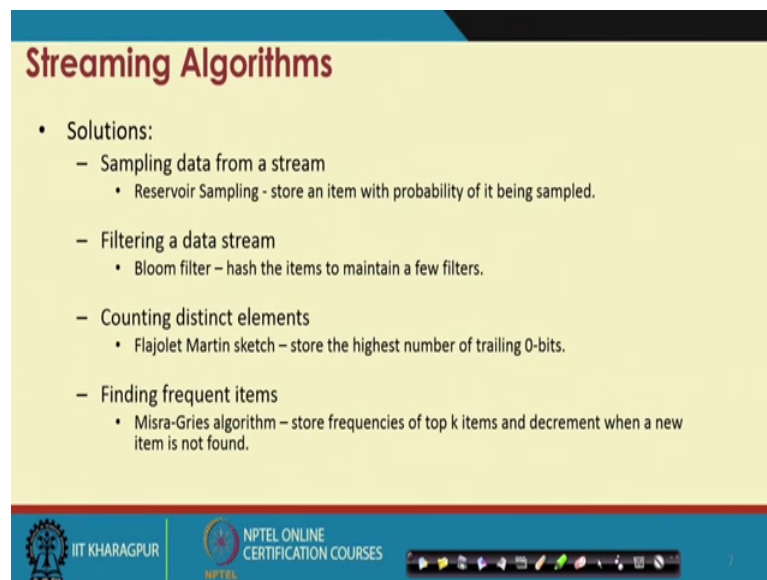
The slide also features the IIT KHARAGPUR logo, NPTEL ONLINE CERTIFICATION COURSES logo, and a video player interface with a presenter's video feed.

So, what are the queries are the problems? So, the first set of problems is that that we have discussed is sampling the data from a stream. So, the idea is that there are different types of items which are coming in a stream so, and you have to construct a random sample from that. So, for example there are you know IPs arriving at a router and you have to construct a random sample of IPs that have come to this router. Now, this set of IP, should follow the same distribution as the original distribution in the entire stream so, that is the problem. The second type of problem is filtering a data stream.

So, here you may have to select so, your data stream may have many different types of elements or many elements having many different properties. So, you have to only select those elements which have a certain property say x ok. So, maybe you are getting numbers in a stream and you have to you know select numbers which have appeared at least once before or something like that ok. .

The second the third type of problem is counting distinct elements, so, basically again your elements are arriving in a stream and you want to know how many different elements have come. So, so this is the problem and the final problem is finding frequent item so, you may have to estimate frequencies of the k most frequent items.

(Refer Slide Time: 12:10)



Streaming Algorithms

- Solutions:
 - Sampling data from a stream
 - Reservoir Sampling - store an item with probability of it being sampled.
 - Filtering a data stream
 - Bloom filter – hash the items to maintain a few filters.
 - Counting distinct elements
 - Flajolet Martin sketch – store the highest number of trailing 0-bits.
 - Finding frequent items
 - Misra-Gries algorithm – store frequencies of top k items and decrement when a new item is not found.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, these are the problems that we have dealt with and what are the solutions that we have discussed.

So, for the sampling problem we have described the reservoir sampling algorithm, basically the idea is that when the items arrived in a stream, you store an item with probability of it being sampled. So, you update based on the number of items you have already seen, you update the probability of retaining this item so, that is the trick is reservoir sampling. Similarly filtering a data stream the idea is that. So, so that so the technique that we have described is the bloom filter.

So, the basic idea is that you hash the items to you so you hash the items and then each item gets hashed to a particular bucket. And then you maintain the filters on this hash buckets rather than items themselves, because the number of items can be very large ok.

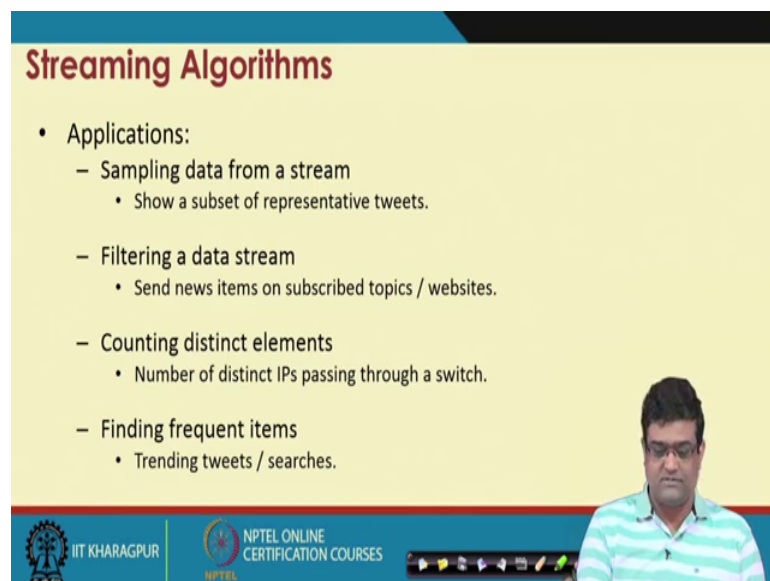
So, instead of maintaining the filters that whether a particular item will should be included or not, you meant on the item you maintain the filter on the hashed value of the item. And because there may be many collisions so, you use many hash functions instead

of one ok so, that is a bloom filter technique. The third problem is the counting distinct items and, you have you have learnt many things for this the one of the techniques you have learnt is the Flajolet Martin sketch, which basically stores the highest number of trailing 0s.

And from that it can give you an estimate of how many distinct items have so, it so, highest number of trailing 0s bits in a hash of the items ok. So, each item comes to hash it can you store the highest number of trailing 0 bits and from that you can get an estimate of how many distinct items have passed through the stream. And the final problem that we have describe this finding frequent items and, again there is a very rich literature on this and we have also described many algorithm.

So, one of them is for example, the Misra Gries algorithm, where you basically store the frequencies of are you strive to store the frequencies of top k items. And whenever new item is not found, whenever you see a new item, you know and you do not find it in the top k list, you reduce the count of this new of the items in the current list. And when the item count of existing item become 0, then that position become free ok. So, you essentially a sort of charge all the items for you know being in the top k list, but not appearing in the stream ok.

(Refer Slide Time: 15:47)



Streaming Algorithms

- Applications:
 - Sampling data from a stream
 - Show a subset of representative tweets.
 - Filtering a data stream
 - Send news items on subscribed topics / websites.
 - Counting distinct elements
 - Number of distinct IPs passing through a switch.
 - Finding frequent items
 - Trending tweets / searches.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, what are the applications of this right so, for example, take the first problem right so, sampling data from a stream so consider the problem of showing a subset of representative tweets right.

So, you have your tweeter timeline and tweeter wants to show you some set of tweets, but not the entire set of most recent tweets ok. So, it only wants to show subset of tweets, but, then it wants to show them you know in a manner such that the fo[r]- the distribution of the tweets that appear on your timeline is sort of representative of the distribution of the entire set of tweets that appear on your timeline ok.

So, this is the problem of reservoir sampling so, you want sample some representative tweets from your actual time line to show on your screen ok. Now, the second problem is filtering a data stream right so consider the stream of all new so, many news agencies all over the world you know polling in which many news say you consider for example, a news aggregator like Google are Yahoo are somebody and may all the news agencies all over the world, they are polling in with news articles. .

Now, you have let us say subscribe to certain kinds of news articles right. So, you want sports, you want you are maybe music, but you do not want politics or something like that ok. So, how do you know so, and every user has mentioned this kind of preferences for their.

So, the problem is to actually send a news items only on the subscribe topics to each user. So, you have to filter the originals large stream of news item, you know in a personalized manner based on the subscribe topics or maybe subscribe websites with this is the problem of you know so, you need to solve the problem of filtering a data stream in this case.

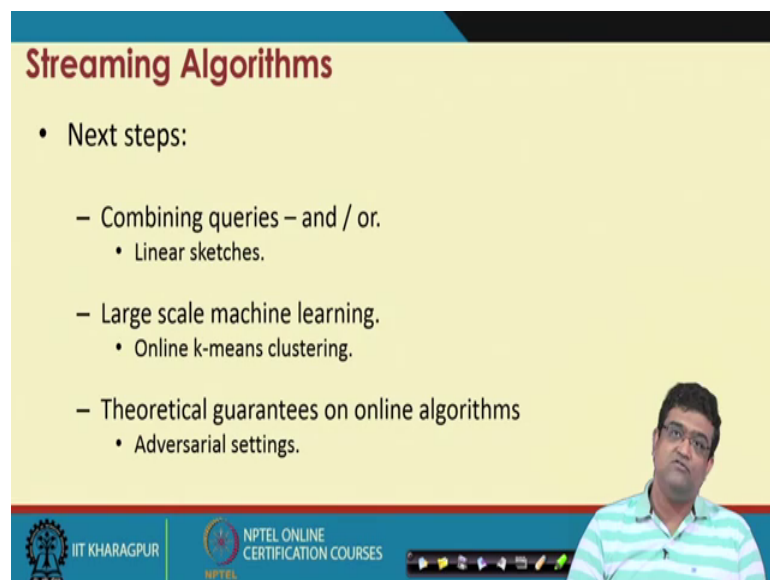
The third problem is that of counting distinct item so, for example, switch you know it sees many IP addresses, it may want to access it in it is own importance are how many different IP is are how many different computers are routing their packets through this particular switch it is so, high.

So that means, it wants to know whether it is a it is a very important central half position or is it somewhere in the periphery of the network so, that you know it only gets a fuse

are packets from a few IPs ok. And so, this is this is the problem of counting the number of distinct items that pass through that switch ok.

Similarly so, the final problem is that are finding frequent item. So, for example, you know you want to show trending tweets or trending searches so, you know tweets get re-tweeted so, every tweet has a re-tweet count and searches get saying search gets repeated many time by many different people. So, certain tweets and certain searches are trending that is these are beings tweeted or search very frequently, let us say in the past 5 minutes or past half an hour are so, and you want to find this right.

(Refer Slide Time: 19:51)



Streaming Algorithms

- Next steps:
 - Combining queries – and / or.
 - Linear sketches.
 - Large scale machine learning.
 - Online k-means clustering.
 - Theoretical guarantees on online algorithms
 - Adversarial settings.

The slide also features a video feed of a presenter in the bottom right corner and logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES at the bottom.

So, this is the and you may want to find top 10 search tweets are top 10 search searches and this is the problem of finding frequent items. . So, so this community is also trying to extend itself. So, one of the direction it is trying to extend itself is combining search queries ok.

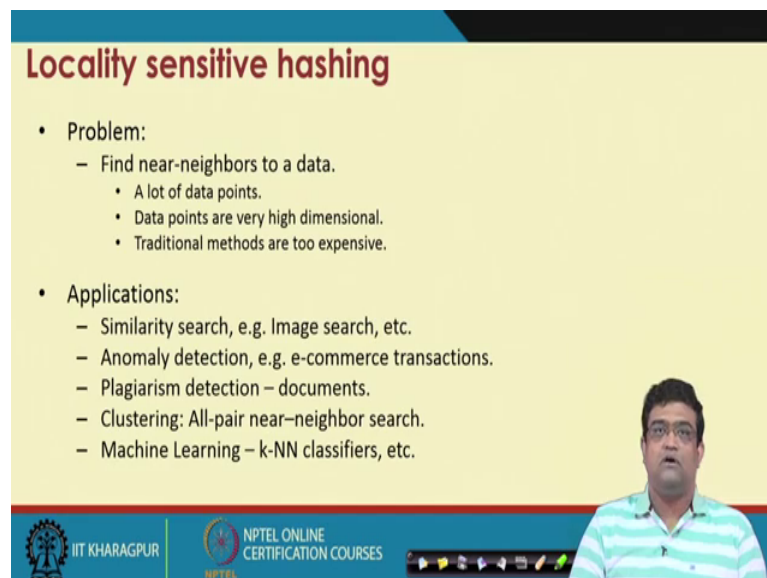
So, for example, you know instead of saying that you know I want to search for a particular I want to filter on a particular topic you may say I want to filter on a combination of topics or you know, one can ask a combination of queries ok. And so, basically people are looking at linear sketches, where you know it is known that you can combine two different sketch linear sketches to form a sketch of for the combined quarry ok.

So, these are some of the things people I have done in the past ok, another place where this online setting is or which is the term that machine learning people used for the streaming setting is in large scale machine learning, where are you know many a times data is you know there is a lot of data hours for some reason the data is coming in an online manner or a streaming manner. And you want to train machine learning models on this kind of data.

So, this is a very active area of research in machine learning and some of the ideas are borrowed from this streaming algorithm settings. And also there are some theoretical guarantees, which are given for as many online algorithms whose background basically originates in the techniques that we have described here. And one special technique here is an adversarial setting, where you know an adversary make give you know certain data or certain information in a streaming manner you know you know one after the other manner and you have to optimize your own objective.

And the adversary is trying to make you not optimize your objective ok. So, all these things are very also or these setting are a very high areas of research and important areas of research.

(Refer Slide Time: 22:25)



Locality sensitive hashing

- Problem:
 - Find near-neighbors to a data.
 - A lot of data points.
 - Data points are very high dimensional.
 - Traditional methods are too expensive.
- Applications:
 - Similarity search, e.g. Image search, etc.
 - Anomaly detection, e.g. e-commerce transactions.
 - Plagiarism detection – documents.
 - Clustering: All-pair near-neighbor search.
 - Machine Learning – k-NN classifiers, etc.

The slide is part of an NPTEL presentation. It features the IIT Kharagpur logo and the NPTEL Online Certification Courses logo at the bottom. A small video inset shows a man in a striped shirt speaking.

So, the second broad class of techniques that I am that, we have described is the locality sensitive hashing class of techniques and also some kind of dimensionality reduction techniques come together. So, what is the problem of locality sensitive hashing so,

basically you have to find near neighbors to a particular data point ok. So, this could happen because, there are lot of data points so the yeah so, the problem is challenging maybe because there are lot of data points, or maybe each of these data points are very high dimensional and calculating the similarity itself is very difficult. And in both of these cases the simple naive algorithms are very expensive ok.

So, the application can include for example, similarity search so, you could do a image search, now nobody wants to retrieve the exact same image ok, that is instead what people want to do is they want to retrieve similar images to a particular input image ok. And this is an example of similarity search then people want to go anomaly detection.

So, for example, you have a certain transaction and you want to quickly know, whether it is an anomalous transaction are not. So, want to find similar transactions to this particular transaction very quickly, so if you do not find any similar transaction, you can think that it is an anomalous transaction.

Another application could be plagiarism detection so, either it could be in some search some sort of documents, papers or it could be in websites, or other content where you want to know given a piece of content, whether this content is copied from somewhere or part of this content is copied from somewhere or not ok. So, in this case you want to basically search for similar content among the other content that you may have and you may want to detect that, whether this content is plagiarized or not, another application is for example, clustering where for many clustering algorithms. So, you may have to do all pair nearest neighbor search so, for every they point in a data set, you have to find the nearest points among the other points ok.

So, example is for example, k means algorithm ok so, you have to find the nearest points slow given point. And again you can use something like a locality sensitive hashing if your data set itself is too large. And then of course, there are applications in machine learning like k nearest neighbor classifiers.

(Refer Slide Time: 25:27)

Locality sensitive hashing

- Ideas:
 - Hashing is for exact search.
 - Utilize collisions - similarity.
 - Family of hash functions.
 - For various notions of similarity:
 - Jaccard, L_2 , Hamming, etc.
 - Amplification
 - Many hash functions.
 - And-ing makes probabilities lower.
 - Or-ing makes many high probabilities higher.
 - Adjust to get the perfect threshold.

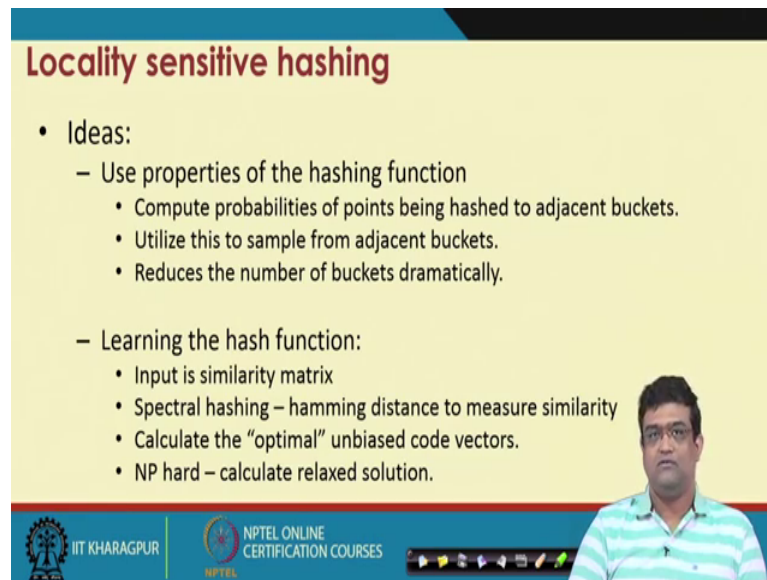
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, what is the idea the idea is that you know for exact search a many a times use hashing right. So, if you want to find all the entries in a list with the same key value, you just hash the keys and you create a hash lists and then you do an exact search ok. So, so, the idea is locality sensitive hashing is simple can you utilize collisions to measure similarity ok.

And you do that by defining a family of hash functions rather than a single hash function ok. So, this can be done for many similarity measures also we have described Jaccard milite t L_2 similarity etcetera. And the key idea here is that you can use many such hash functions from the family of hash functions.

So, what happens is when you do so, you do two types of operation you do either and-ing or or-ing operations the and-ing operations makes the probabilities lower for, but then it also make the probability is lower for nearby objects or similar objects the probability of it them being retrieved also become lower. So, then you do a or-ing operation to make the probabilities of nearby objects higher. So, you can adjust the ratio of the number of and-ing and or-ing operation to get a kind of a threshold at which you want to stop. So, you can then only retrieve objects with certain similarity threshold ok.

(Refer Slide Time: 27:21)



Locality sensitive hashing

- Ideas:
 - Use properties of the hashing function
 - Compute probabilities of points being hashed to adjacent buckets.
 - Utilize this to sample from adjacent buckets.
 - Reduces the number of buckets dramatically.
 - Learning the hash function:
 - Input is similarity matrix
 - Spectral hashing – hamming distance to measure similarity
 - Calculate the “optimal” unbiased code vectors.
 - NP hard – calculate relaxed solution.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

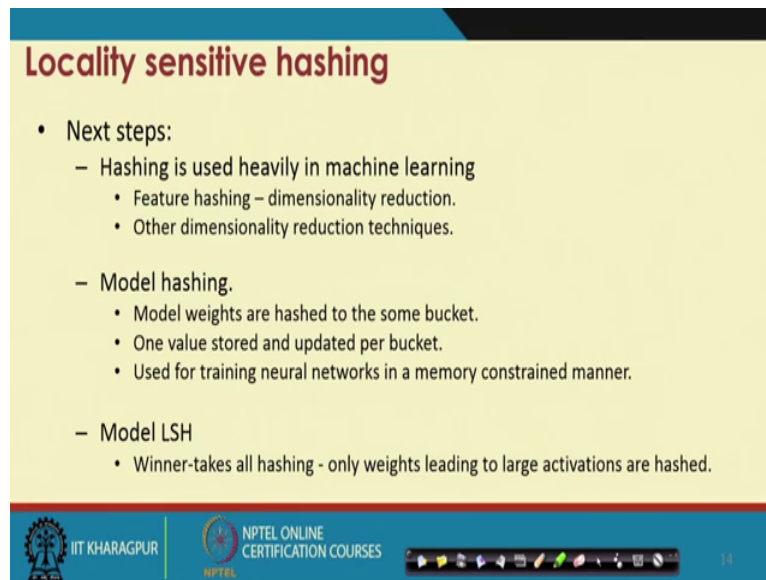
So, next improvement on this was to use what was the multiprogramming hashing problem is to use basically properties of the hashing function. So, what you do is so, the problem with LSH, so, as we all understand LSH is a scheme for you know space time trade off.

So, you create large you create a large data structure, typically it is quite large and, then you instead of do linear time search over the points you can do sub linear time search ok. So, here the idea is that can you reduce the size of the data structure by computing probabilities of points being hashed into nearest near buckets. And then utilize this probability to sample points from adjacent hash buckets as well ok. So, this reduces the basically the number of hash buckets are the number of hashing functions that you have to use dramatically ok.

And the final improvement that we saw is can we learned the hashing function itself. So, for example, if the input is similarity matrix and you do not have data point, but instead you have a similarity matrix, can you calculate code vectors for this hashing points such that the hamming distance between this quotes actually gives you the similarity. And then you can use the you know the LSH scheme for over hamming distance to sample or to find the nearest neighbor efficiently for this similarity matrix ok.

And yeah we so, the basically the spectral hashing techniques calculates relax solution to this problem.

(Refer Slide Time: 29:27)



Locality sensitive hashing

- Next steps:
 - Hashing is used heavily in machine learning
 - Feature hashing – dimensionality reduction.
 - Other dimensionality reduction techniques.
 - Model hashing.
 - Model weights are hashed to the some bucket.
 - One value stored and updated per bucket.
 - Used for training neural networks in a memory constrained manner.
 - Model LSH
 - Winner-takes all hashing - only weights leading to large activations are hashed.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | 14

So, so what are the next step so, the next so, firstly this hashing trick has been used heavily in machine learning. So, one of the ways it has been used is in the feature hashing which has been used for dimensionality reduction. So, you can hash many features into one bucket ok.

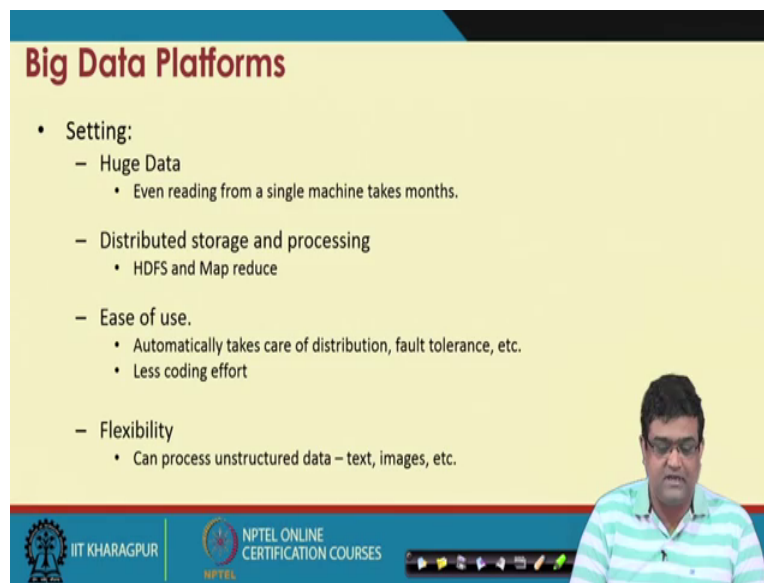
And this may cause the collision of features, but hopefully the collision stone result in much degradation of performance. So if you have high dimensional data this is one way of reducing the dimensionality, for you know doing machine learning effectively on high dimensional data's. The second trick is that of model hashing ok so, with the advent of deep learning and all these techniques.

Now, one of the things that has happened is model sizes have become very large ok. So, so how to reduce so, but you may want to run these models for example, on a cell phone which has low memory are you may even want to train some of these models on low memory ok. So, the solution to this is to hash the model parameters so, if you have lets a large number of model rates, you can hash this model rates into some hash buckets. So, smaller number of hash buckets and then store a one value per bucket ok.

And then you can so, basically you can train for example, neural networks in the memory constrained manner, using this data structure of hash model rather than the entire original model ok. And an extensional that can be the winner takes all hash, which basically updates weights of parameters which lead to large activations of neurons.

So, instead of you know so, this allows basically instead of hashing weights randomly, this allows in some sense weights which cause large or large activations of certain neurons to be preferably hash together ok. So, these are some of the recent developments in this particular area.

(Refer Slide Time: 32:17)



Big Data Platforms

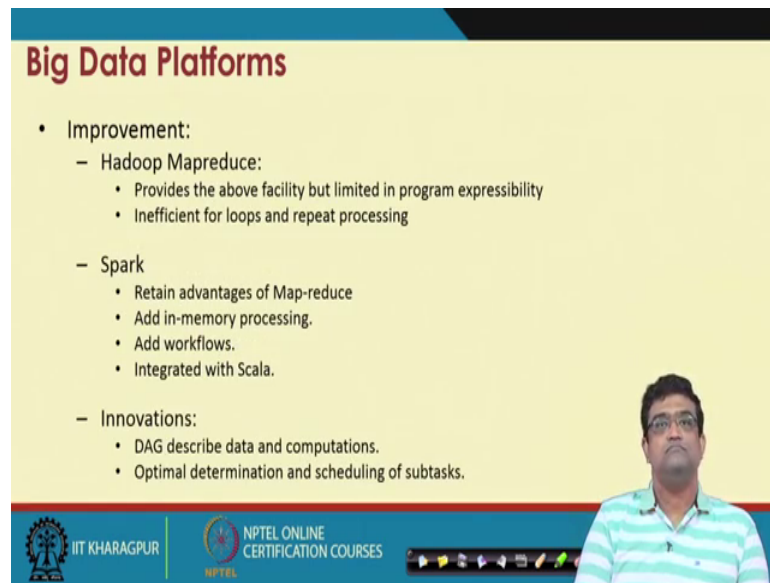
- Setting:
 - Huge Data
 - Even reading from a single machine takes months.
 - Distributed storage and processing
 - HDFS and Map reduce
 - Ease of use.
 - Automatically takes care of distribution, fault tolerance, etc.
 - Less coding effort
 - Flexibility
 - Can process unstructured data – text, images, etc.

The slide also features a video inset of a man in a striped shirt speaking, and logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES at the bottom.

The third a for a broad area, that we have covered is the big data platforms area and the setting is of course, that you have huge data you have you have to do a distributed storage and processing and we have described the hadoop distributed file system for distributed storage. And the map reduce framework for distributed processing and basically the motivation behind this kind of frame work is that first the programmer should find it easy to use.

So, for example, programmer does not have to you know take care of how to distribute the data, or how to distribute the tasks and the programmer should not have to take care of for example, fault tolerance and think like that. And also the programmer should not write all the distribution code like you know. So, there should be a less coding effort ok and so, at the same time the framework where design with flexibility in mind that, it can you know process unstructured data like text, images etcetera.

(Refer Slide Time: 33:27)



Big Data Platforms

- Improvement:
 - Hadoop Mapreduce:
 - Provides the above facility but limited in program expressibility
 - Inefficient for loops and repeat processing
 - Spark
 - Retain advantages of Map-reduce
 - Add in-memory processing.
 - Add workflows.
 - Integrated with Scala.
 - Innovations:
 - DAG describe data and computations.
 - Optimal determination and scheduling of subtasks.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Now so, how do so, this was the first set of big data platforms and these were designed about maybe 10 years back now, more than someone more than 10 years back 15 years back. And also there problem was that they had a limited expressibility, because the map reduce framework could express only certain types of programs and they were in efficient for certain types of processing so for example, iterative and interactive applications use to runs slower on map reduce ok.

So, these problems were basically allocated by another platforms spark, which was designed about 7 years back and became matured about 5 years back. And then so, the retain advantages of map reduce they do not lose anything from map reduce, but they add in memory processing they add workflows and the integrate in into scalar ok.

And so, this provides with hugely more expressive platform, for expressing your big data computation and the key innovations there were basically directed a cyclic graph to describe both data and computations and basically, spark could optimally determine you know the both the so, what subtask to perform and how to schedule them ok.

(Refer Slide Time: 35:18)

Big Data Platforms

- Next steps:
 - The concept of DAG processing with states:
 - Tensorflow – loose fault tolerance
 - Adding specific features:
 - Stream processing.
 - SQL type queries
 - Data frames and optimization.
 - Machine Learning algorithms:
 - Distributed optimization.
 - Recursion – decision trees.
 - Matrix multiplication, etc.

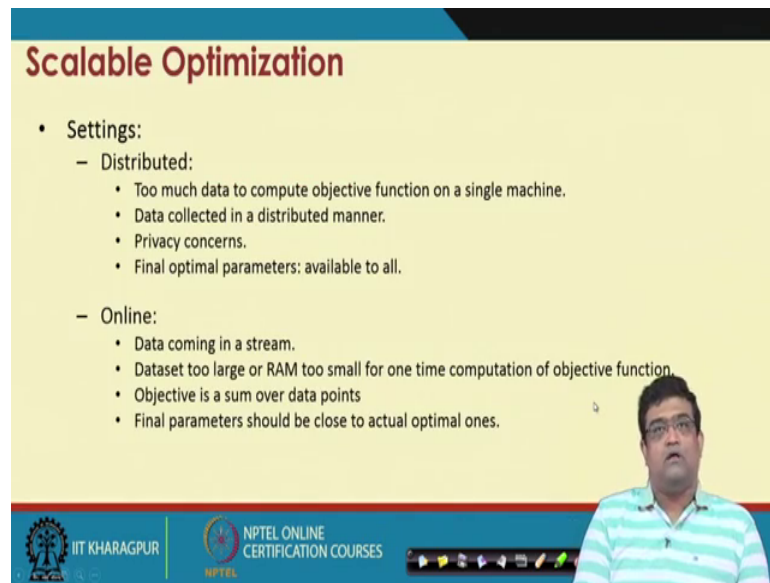
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

So, this concept of dag processing has now cotton a lot and many of the high level system for example, I will describe one system which is the tensor flow ok. So, this system is become very popular for deep learning and this system also uses the same concept of using that to express processing, you know also sometimes for bulk transformations to express processing.

So; however, tensor flow for example, uses states for each node and hence it loses the fault tolerance, but works very well in practice ok. And then the spark itself has added many specific features something like stream processing, SQL type queries data frames so, basically for optimize processing or vectorial data and also some amount of machine learning libraries. And the other important area of research has been how to put many of the machine learning algorithms into the spark framework.

So, for example, how can one do distributed optimization on spark, how can one write recursive programs in spark for example, can be trained decision trees using spark etcetera and some linear algebra applications like matrix multiplication matrix inversion etcetera.

(Refer Slide Time: 36:55)



Scalable Optimization

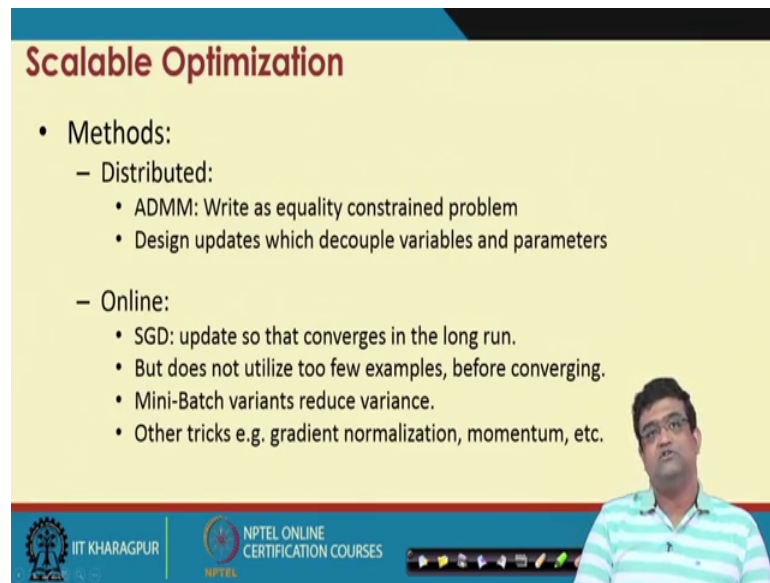
- Settings:
 - Distributed:
 - Too much data to compute objective function on a single machine.
 - Data collected in a distributed manner.
 - Privacy concerns.
 - Final optimal parameters: available to all.
 - Online:
 - Data coming in a stream.
 - Dataset too large or RAM too small for one time computation of objective function.
 - Objective is a sum over data points
 - Final parameters should be close to actual optimal ones.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

The fourth area that we describe is that have scalable optimization and, we have mainly describe two types of scalable optimization techniques one is distributed, where which are useful when there is too much later to compute the objectives in a single machine and data is a or data is collected in a distributed manner or there are some privacy concerned concerns ok.

And the final parameters should be available to on to all. The second setting is the online optimization, where the data required for completing the objective function is arriving in a stream again could be because of any number of reasons write memory is too small and the objective function is some over certain data points ok. And again the final parameters should be of the optimization algorithms should be close to actual parameter.

(Refer Slide Time: 37:55)



Scalable Optimization

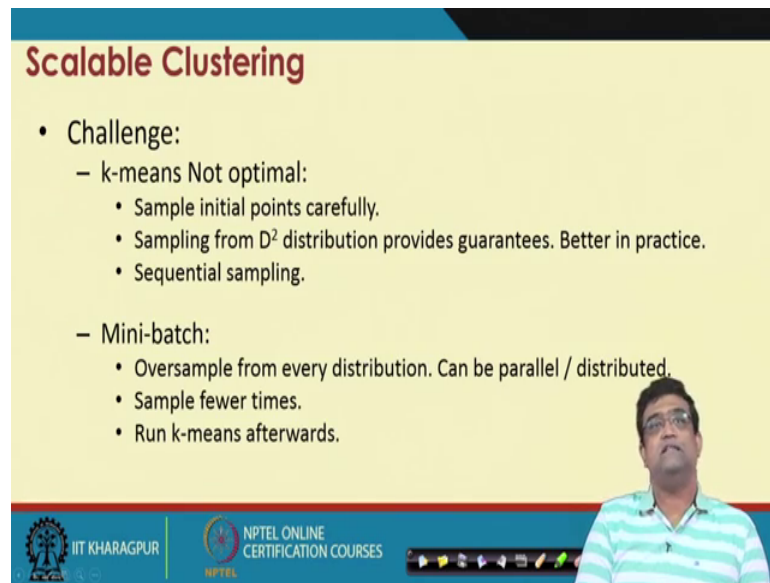
- Methods:
 - Distributed:
 - ADMM: Write as equality constrained problem
 - Design updates which decouple variables and parameters
 - Online:
 - SGD: update so that converges in the long run.
 - But does not utilize too few examples, before converging.
 - Mini-Batch variants reduce variance.
 - Other tricks e.g. gradient normalization, momentum, etc.

The slide includes logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES, and a video inset of a man in a striped shirt.

So, for the distributed setting we have describe the ADMM method basically the key idea, here is that you write the optimization problem as a equality constraint problem with many variables for one corresponding to every in distributed note. And then you design updates which decouples the variables and the so, it actually decouple just the variables are the parameters.

So, decouples the variables so, the updates two variables are not coupled with one another ok. So, this is the key idea the on the in the online settings, we have describe various stochastic gradient descent algorithms. And we have so, the key idea here is to you know to design the updates such that they converge in the long run, but do not take two few iterations to not even explore the space in a good enough manner ok.

(Refer Slide Time: 39:14)



Scalable Clustering

- Challenge:
 - k-means Not optimal:
 - Sample initial points carefully.
 - Sampling from D^2 distribution provides guarantees. Better in practice.
 - Sequential sampling.
 - Mini-batch:
 - Oversample from every distribution. Can be parallel / distributed.
 - Sample fewer times.
 - Run k-means afterwards.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

And there are many advance mentioned extensions like mini batch clearly normalization momentum etcetera, which we have discussed briefly. In the last technique that area that we have describe is scalable clustering and here, the challenge was that you know the original k means clustering technique was not optimal.

So, there was so, first to describe the k means plus plus which basically provides a sequential sampling algorithm, which guarantees approximately that the solution that you get is someone close to optimal, but the problem is that this is a sequential sampling technique.

So, next we describe mini batch kind of variant where we over sampling every distribution. So, that basically we can do it with fewer number of iterations so, original k means plus plus would require k iterations to sample the k initial point, but the new update would require much fewer iterations. So, with that we are we conclude this course and thank you all very much for your attention.

Thank you.