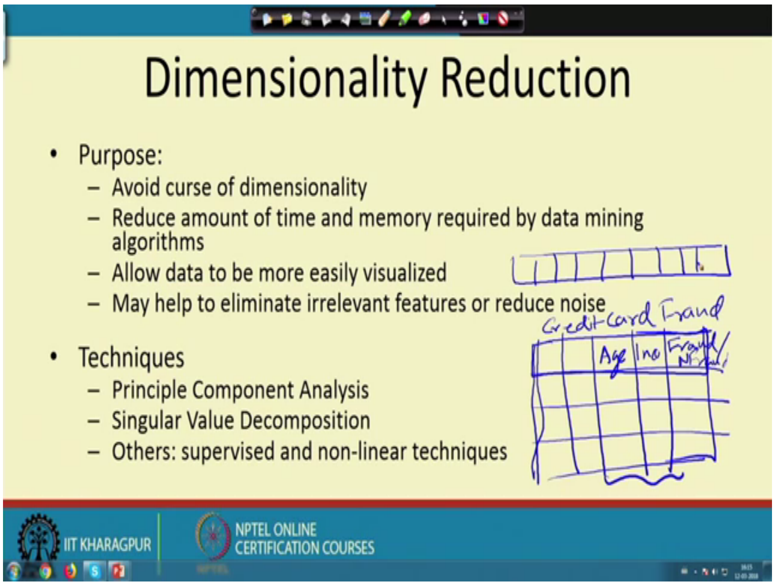
**Data Mining**
**Prof. Pabitra Mitra**
**Department of Computer Science & Engineering**
**Indian Institute of Technology, Kharagpur**

# Lecture - 41
# Dimensionality Reduction – I

We will in this lecture discuss a topic which is important across all the determining algorithms we discussed previously, weight, clustering, equation, classification. The talks here that we will talk about is dimensionality reduction.

(Refer Slide Time: 00:39)



## Dimensionality Reduction

- Purpose:
  - Avoid curse of dimensionality
  - Reduce amount of time and memory required by data mining algorithms
  - Allow data to be more easily visualized
  - May help to eliminate irrelevant features or reduce noise
- Techniques
  - Principle Component Analysis
  - Singular Value Decomposition
  - Others: supervised and non-linear techniques

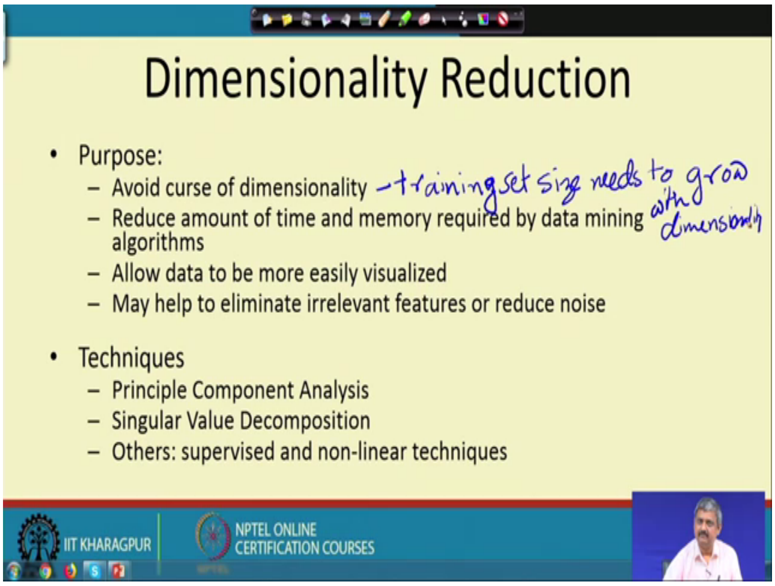
That means, you have a number of attributes in your data often a large number of attributes, but the question is often all these attributes are not really required or important for the data mining task. So, what we do in dimensionality reduction is to reduce the number of attributes that we actually need.

So, for example, suppose I have a table the data table the data table like this, say a credit card fraud detection data table and we want to decide on the classification problem whether fraud or not fraud, not fraud ok. And maybe there are a lot of attributes maybe income, age, maybe many many other attributes. So, suppose I decide that I do not take all of them I take only a few of the attributes then I am in effect doing it dimensionality reduction.

Why it is dimension reduction? Because each of the data points each row in this table each row in this table you can consider as a point in some space as a vector and the number of attributes you have is the dimension of that vector dimension of that point. So, what we are doing in dimensionality reduction is to reduce that dimension and this is done for the following purposes.

First thing is that it removes something called card sub dimensionality; that means, as your dimensions increase, as you need more and more attributes in your data you need more and more training samples to train your algorithms. More you more you increase the dimension more you need training samples ok. So, naturally in the previous discussions it was evident that it is dependent on the, so this problem is the curse of dimensionality.

(Refer Slide Time: 03:17)



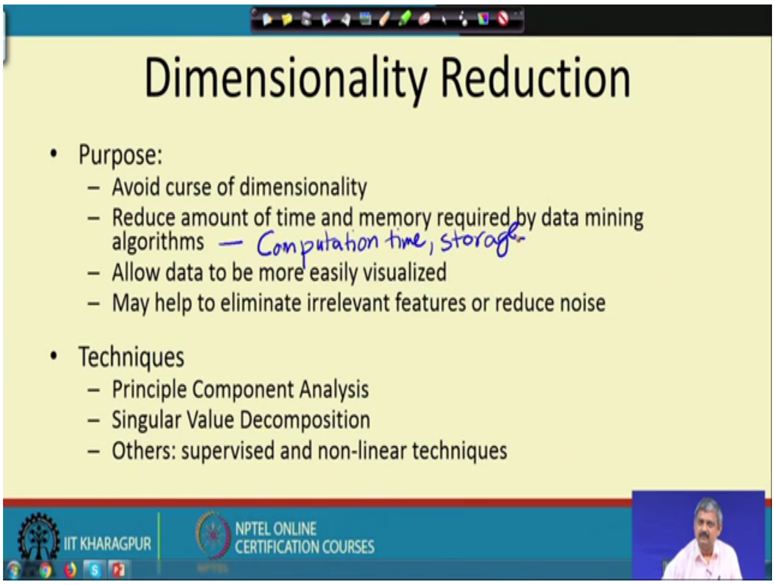
## Dimensionality Reduction

- Purpose:
  - Avoid curse of dimensionality – training set size needs to grow with dimensionality
  - Reduce amount of time and memory required by data mining algorithms
  - Allow data to be more easily visualized
  - May help to eliminate irrelevant features or reduce noise
- Techniques
  - Principle Component Analysis
  - Singular Value Decomposition
  - Others: supervised and non-linear techniques

It is more the dimension more training set and it grows exponentially; that means, if the dimensionality goes up by 1, you need square of the number of training set on it goes up rapidly ok. So, that is the first motivation.

Second motivation of course, if you increase dimensionality your computation time will increase your storage will increase ok.

(Refer Slide Time: 04:13)



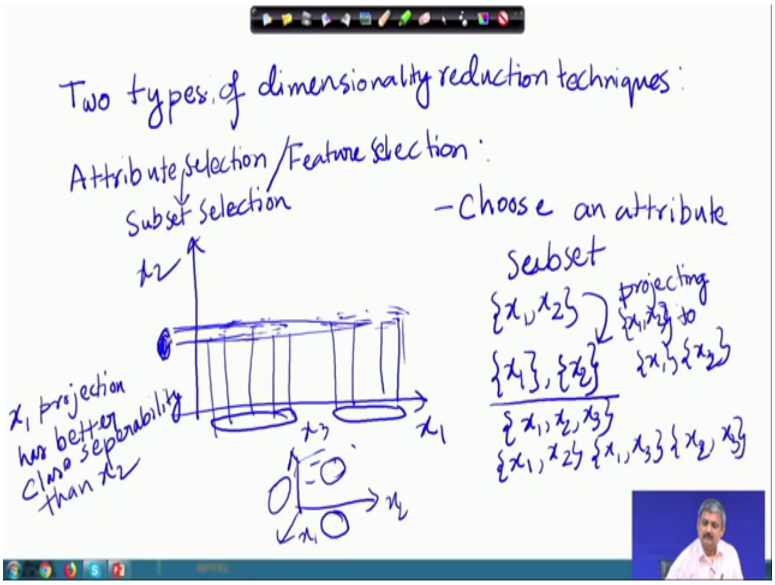
# Dimensionality Reduction

- Purpose:
  - Avoid curse of dimensionality
  - Reduce amount of time and memory required by data mining algorithms — Computation time, storage
  - Allow data to be more easily visualized
  - May help to eliminate irrelevant features or reduce noise
- Techniques
  - Principle Component Analysis
  - Singular Value Decomposition
  - Others: supervised and non-linear techniques

So, you want to reduce dimensionality to improve upon computation time, and storage that is the second motivation.

The third motivation is that you can more the dimension and more difficult it is to interpret the data to visualize the data ok. So, we actually we human beings cannot visualize beyond 2 or 3 dimension and also this dimension reduction will help us to eliminate the noisy features, the features which are not relevant noisy features. So, in this lecture we will discuss some of the techniques for dimension reduction, but let me first explain you what it actually means.

(Refer Slide Time: 05:24)



Two types of dimensionality reduction techniques:

Attribute selection / Feature selection:

Subset selection

- Choose an attribute subset

$x_2$
$x_1$
$x_3$
$x_2$
$x_1$
$x_1$ projection has better class seperability than $x_2$
$\{x_1, x_2\}$
$\{x_1\}, \{x_2\}$
projecting $\{x_1, x_2\}$ to $\{x_1\}\{x_2\}$
$\{x_1, x_2, x_3\}$
$\{x_1, x_2\}\ \{x_1, x_3\}\ \{x_2, x_3\}$

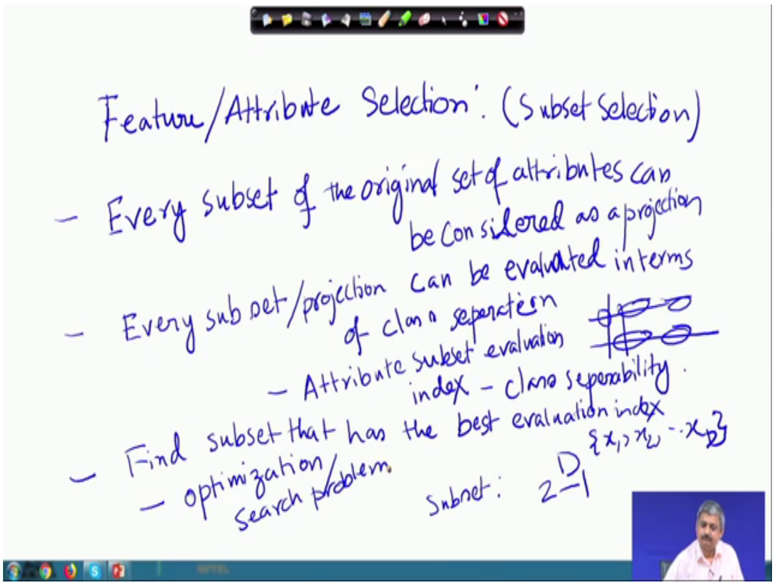
There are two types of techniques one is known as attribute selection or feature selection. What we do in this approach, let me take an example. Suppose I have two attributes and maybe my data points look like this. So, in my attribute selection I have to choose one out of this two features. So, I have to choose a n attribute subset. So, if my original set of attributes at x 1 and x 2 I have to choose either x 1 or x 2 if the subset this attribute station actually means attribute of subset selection ok.

So, in case of 3 attributes if I had 3 attributes I could have chosen only x 1 x 2 or I could have chosen x 1 and x 3 or I could have chosen x 2 and x 3, if I had to choose 2 out of 3 ok. So, what does this choosing means? See for example, in this case, so out of x 1 x 2 if I choose only x 1 it actually means projecting each and every point looking at their x 1 coordinates only. So, basically it means that projecting x 1 x 2 plane points to x 1 or x 2.

Similarly, in this case there if there are 3 thing, x 3 if I choose two it is a projection of a 3 dimensional point either to this plane this plane or this plane ok. So, attribute selection is equivalent to collection. But which of these two projection you will take? You look at the projection if your objective is to classify the examples into these two classes then maybe the x projection is better x for x and x because if we project x 1 the x 1 projection is better because if we project to x 2 everything gets mapped to this set of points. So, I can write I can write than x 2 ok. So, this projection has more separated classes than this projection. So, naturally x 1 is a better choice.

This principle is generalized in attributes selection algorithm how is generalized, it is generalized.

(Refer Slide Time: 10:20)



Feature/Attribute Selection: (Subset Selection)

- Every subset of the original set of attributes can be considered as a projection
- Every subset/projection can be evaluated in terms of class separation
  - Attribute subset evaluation index - class separability.
- Find subset that has the best evaluation index
  - optimization/search problem.

$\{x_1, x_2, \ldots x_D\}$

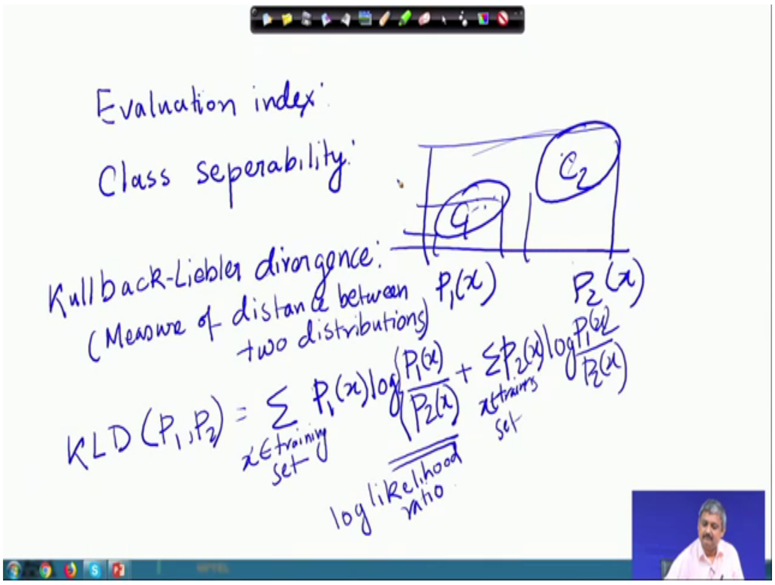
Subset: $2^D - 1$

It is generalised in the following way. So, every projection that we get every subset that we get you can evaluate it you can give it a score in terms of class separation or any other criteria ok. So, for example, here it is better than this ok, this projection is better than this projection.

So, these I have to define something called attribute subset evaluation index ok, absolute which may take a form of a class separability. And then what we do? Find the subset that has the best evaluation index ok. So, in this case there are two subsets only x 1 and x 2 in general if we have say x 1 x 2 up to x D. How many subsets are possible? 2 to the power D different subsets are possible ok. If you have originally D features capital D features to depart D different subsets are possible D minus one actually.

So, each of these subsets I find a evaluation index classify ability and choose the best. So, this is a optimization or a search problem find the best subset ok. So, what I will quickly do is to discuss one evaluation index and few such optimization or such techniques.

(Refer Slide Time: 14:45)



Evaluation index:

Class seperability:

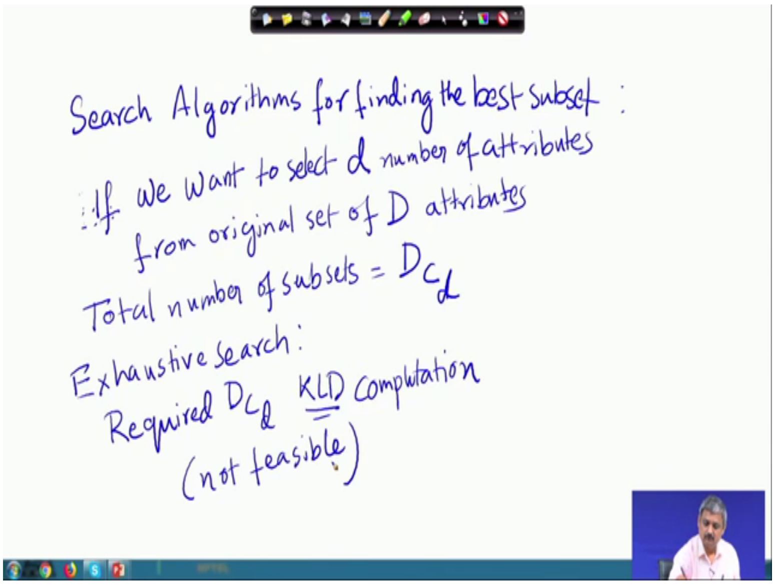
Kullback-Liebler divergence:
(Measure of distance between two distributions)

$$KLD(P_1, P_2) = \sum_{x \in \text{training set}} P_1(x) \log \underbrace{\left(\frac{P_1(x)}{P_2(x)}\right)}_{\text{log likelihood ratio}} + \sum_{x \in \text{training set}} P_2(x) \log \left(\frac{P_1(x)}{P_2(x)}\right)$$

You want to find how much separated two classes are. If we know the probability distribution of the first class let me call it p 1 x and the probability distribution of the second class p 2 x I can define a measure called distributions. This is one particular form of KL dimensions ok. So, what you do? At every training point you find out how much separability class 1 gives, how much separability class 2 gives, take their likelihood ratio.

So, this quantity is ratio multiplied by their probabilities add up overall (Refer Time: 17:49). So, this projection if we consider x to be this x 1 we will have a higher KLD than this projection ok. There are other measures also I am just giving an example.

(Refer Slide Time: 18:11)



Search Algorithms for finding the best subset :

If we want to select $d$ number of attributes from original set of $D$ attributes

Total number of subsets $= {}^{D}C_{d}$

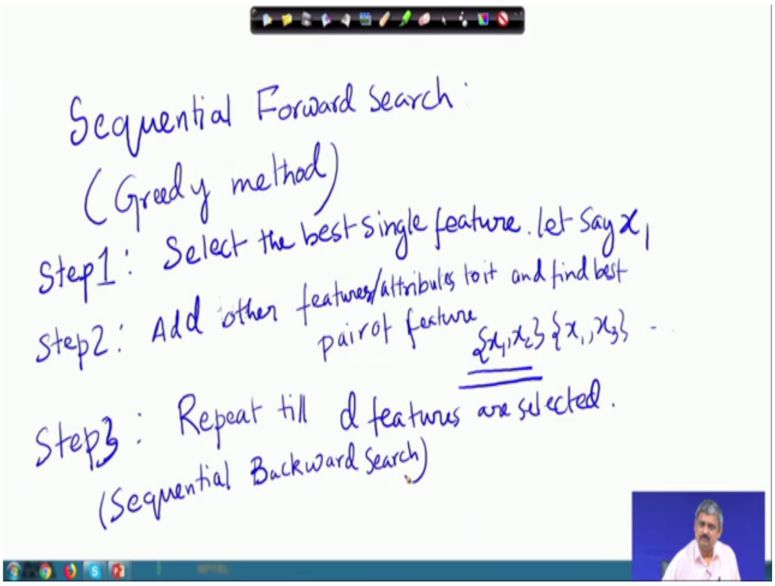
Exhaustive search :

Required ${}^{D}C_{d}$ $\underline{\underline{\text{KLD}}}$ computation

(not feasible)

You have so many subsets do not that for every subset you have to find KL divergence or any other evaluation measure need not be KLD. This is not feasible ok, too many computations.

(Refer Slide Time: 20:30)



Sequential Forward Search :

(Greedy method)

Step1 : Select the best single feature. let say $x_1$

Step2 : Add other features/attributes to it and find best pair of feature $\underline{\underline{\{x_1, x_2\}}}$ $\{x_1, x_3\}$ -

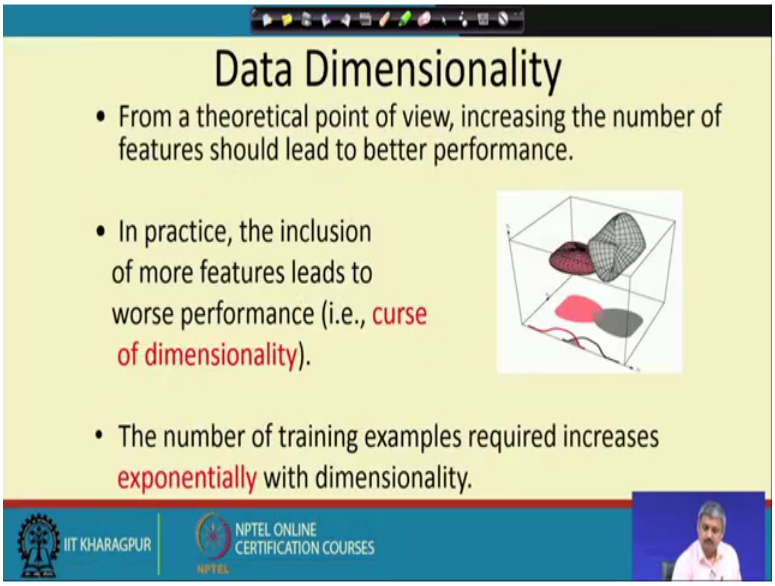
Step3 : Repeat till $d$ features are selected.

(Sequential Backward Search)

So, there is another method greedy method feature ok. So, x 1 you keep you add x 2, you add x 3 and so on and select which is the best KL divergence again selected ok. So, this is a greedy method not optimal, but often it works.

The opposite of this is ok. So, there you take all the features and remove the best remove the worst again remove 2 keep the rest, remove 2 ok, so the reverse of this ok. So, these are the two methods. There are other methods like beam search, sequential floating forward search, branch and mounds search, which are used. So, this is a very important step this is a very time consuming, but important step, there are different ways of doing this ok.

(Refer Slide Time: 24:00)

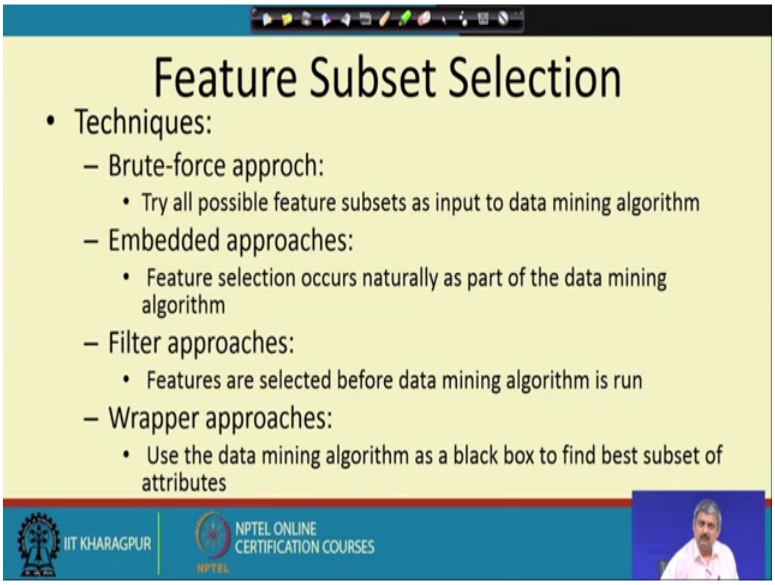


## Data Dimensionality

- From a theoretical point of view, increasing the number of features should lead to better performance.
- In practice, the inclusion of more features leads to worse performance (i.e., curse of dimensionality).
- The number of training examples required increases exponentially with dimensionality.

Wait, I will come back to this ok. So, this is what I have told. You have to select these are there approaches brute force.

(Refer Slide Time: 24:22)



# Feature Subset Selection

- Techniques:
  - Brute-force approch:
    - Try all possible feature subsets as input to data mining algorithm
  - Embedded approaches:
    - Feature selection occurs naturally as part of the data mining algorithm
  - Filter approaches:
    - Features are selected before data mining algorithm is run
  - Wrapper approaches:
    - Use the data mining algorithm as a black box to find best subset of attributes

There are again two approaches if their evaluation measure is tuned to the data mining algorithm. It is called a filter embedded it is not it is called a filter and wrapper two approaches e e e e. If you use the data mining algorithm it is the wrapper if you do not use it is called a filter sometimes the algorithm itself does the embedded dimensional reduction ok. So, this is the technique this is feature selection.

In our next talk we will discuss another group of algorithms called feature extraction algorithms ok, in our next talk.

Thank you for this talk.