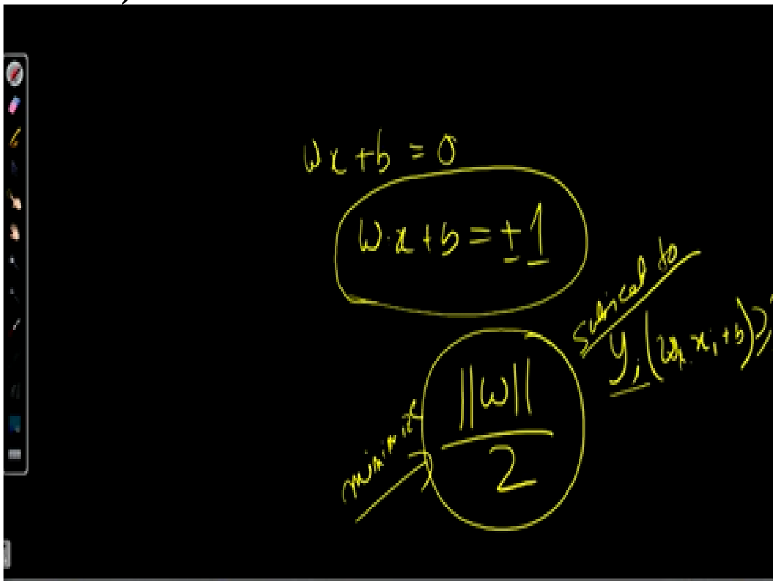
Statistical Learning for Reliability Analysis

Prof. Monalisa Sarma
Subir Chowdhury School of Quality and Reliability
Indian Institute of Technology, Kharagpur

Lecture - 59
SVM (Part - VI)

Hello guys, so at last we have come to the end of this lecture. This is the last but one lecture of this course and in today's class we will wrap up our discussion on SVM. And then our last lecture is a tutorial class.

(Refer Slide Time: 00:44)



So, now before going ahead with this lecture today let us just take a quick recap what we have learned in this last four five lectures on SVM. So, basically, we are doing SVM for a classification purpose we were talking of SVM for a binary classification problem (()) **(00:59)** datas which are linearly separable. Then we went on to find out the equation of the hyperplane. So, what was the equation of the hyperplane.

So, basically it was W . x + b = 0 this is my equation of the hyperplane. Now I we have already discussed this is W . x this is a dot product. So, we have already discussed that this expression if we divide by any constant value, we will get the then the hyperplane will that way we will get will be getting infinite hyperplane but all with the same slope and the same intercept. So, now to just to avoid this confusion we went ahead.

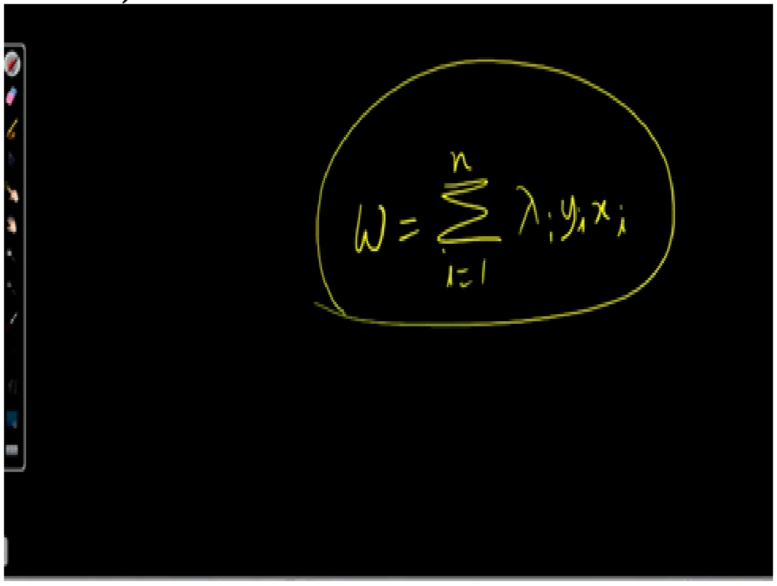
And we found out that W x + b then W x + b = + - 1 which is nothing but two pipeline basically two line parallel to the hyper plane we can say it is a there is a parallel boundary we can say. So, now and then this was our equation of the hyperplane these are just two parallel hyperplane. Now our task was to find out the maximum margin hyperplane. So, in optimization task in the classification in SVM our main objective was to find out the maximum margin hyperplane.

The hyperplane that will give us the maximum margin is not it that was our objective. So, with that objective then we went on to see that this is very much an optimization problem where we have used this the way we try to optimize this function let me write it here we try to optimize this is not it. And subject to there was a constraint subject to the constraint what was the constraint w i y i w i and w i . x i + b this is greater equals to 1 subject to this constraint.

So, this is we have formulated this computation of maximum margin hyperplane. Our task is to compute the maximum margin hyperplane. So, computation of Maximum hyper margin hyperplane we have formulated it as an optimization problem so what is the optimization, we need to minimize this thing minimize. We need to minimize this subject to what subject to y i then to w i . x i + b which is greater equals to one y i is the different classes.

Different means there will be two class. So, subject to this we have to minimize this and then we went ahead and to use lagrangian multiplier for optimizing this problem is not it for optimizing this formulation.

(Refer Slide Time: 03:46)



$$W = \sum_{i=1}^{n} \lambda_i y_i x_i$$

So, like now than what we have done. So, we have while trying to while using lagrangian multiplier we have introduced a dummy variable the dummy variable was λ remember. So, now then after we had formulated the lagrangian then we have differentiated with respect to the input variable that is x i x i of all we using all the attributes. So, we have differentiate the lagrangian with respect to that as well as we have differentiated the lagrangian with respect to the λ.

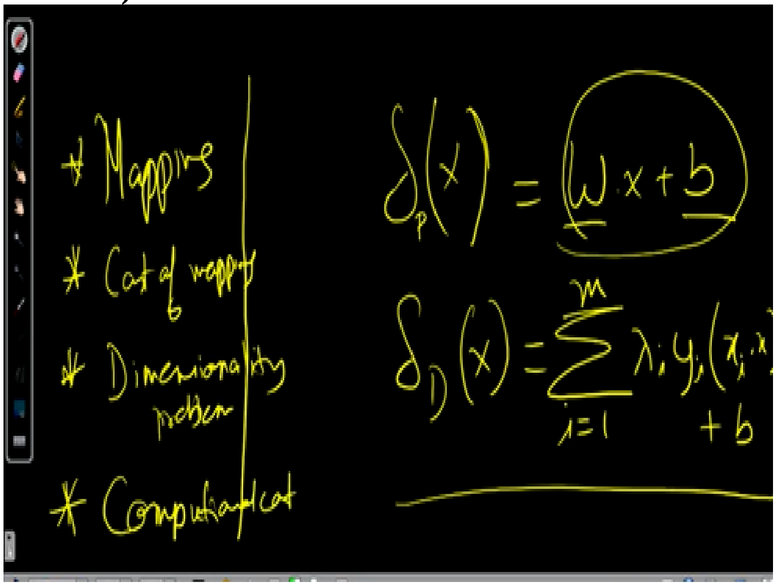
Then we got different functions for that we try to find out the solution using any of the equation solving method. And then we found out the different values of the w n b so, while trying to do that we found out our W what is our W is equals to we found out our W = i = 1 to n λ i w i x i is not it. This is this was our W so n is the total support vectors. Now what happens once this is then again till this it was ok for us if we have data which was linearly separable.

Now if the data’s are not linearly separable then what we have seen there are one way is to use the if some of the data not be very few data’s are not linearly separable then we could use the soft margin. And then if the soft margin is not applicable then we went ahead to use the transformation method like trans we have transformed the data from one space from one input space basically from the original space.

And we have transformed the data to some other space of a higher dimensional by doing what we could solve the non-linearity of the data. If the data is non-linear so what happens what we have done this data we have transformed into some other space and once we have transformed it into some other space then we have used the same what to say same optimization problem and then we have solve it using lagrangian.

But there what we have said instead of this initially the lagrangian form which you have used we have called it as a primal form that is the initial form. Then we have changed a bit and we are calling it a dual form.

(Refer Slide Time: 06:07)


* Mappings
* Cost of mapping
* Dimensionality problem
* Computational cost
$\delta_p(x) = w \cdot x + b$
$\delta_D(x) = \sum_{i=1}^{m} \lambda_i y_i (x_i \cdot x) + b$

So, dual what is the dual form of the lagrangian let me before going to the dual form now using the initial primal form of the lagrangian when we had to classify a data suppose we are classifying a test data x so, I am using it as to find out the classification updates test data we had to this is my step. So, from optimizing the lagrangian I got the value of w I got a value of b then using this w . x + b I could find out where my x data belongs to which class.

So, this was till then. Now if the data is not linear then we have transformed and we have used the primal I mean so sorry the dual form of this, when we have used a dual form what was my classifier now. So, let me call it as this let me call it that primeval this may call is a dual form what was my dual formula classifier. Now during my duel from the classify and water equivalent just that it is i = 1 to m then λ i y i x i . x + b this was my dual form of the classifier.

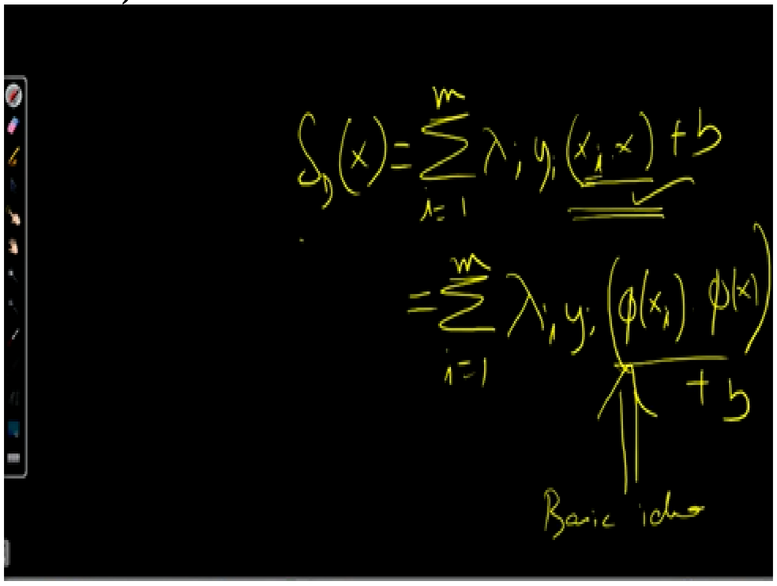
And then the optimization function also change it was a big function I am not writing it again you can see it in my slide. So, this was the dual form of the classifier. So, updating was till here but then we had some other problem. When we have transformed the data from one input dimensional space to some other dimensional space which is usually higher than this it is never lesser at sometimes, we are in the same dimensional but most of the time we go today higher dimension.

So, when we do this transformation there are some issues with this. What are the issues the different issues? One of the issues is mapping. How do we do the mapping? That is how because for each monomial we will select a different mapping function or combination of two three monomial I will select a mapping function. So, how we will do the mapping this is one problem. Second problem is the cost of mapping.

Then the third my third problem is the dimensionality problem. Dimensionality problem usually SVM does not suffer from the curse of dimensionality but and when it goes to a very high and high dimensional space computation it becomes very computational internship. Then it may suffer from dimensionality problem as well. The dimensional problem as well as computation cost dimensionally problem means computation cost will increase both are related actually as well as computational cost.

So, these were the issues of the transformation of data from a lower dimensional space to a higher dimensional space to make the data linearly separable. Now then so mathematician have found a very intelligent way of dealing with this so, where we do not at all have to do the mapping without doing the mapping how we can go about it. So, what was the way?

(Refer Slide Time: 09:12)



$$S_D(x) = \sum_{i=1}^{m} \lambda_i y_i \underline{(x_i \cdot x)} + b$$

$$= \sum_{i=1}^{m} \lambda_i y_i (\phi(x_i) \cdot \phi(x)) + b$$

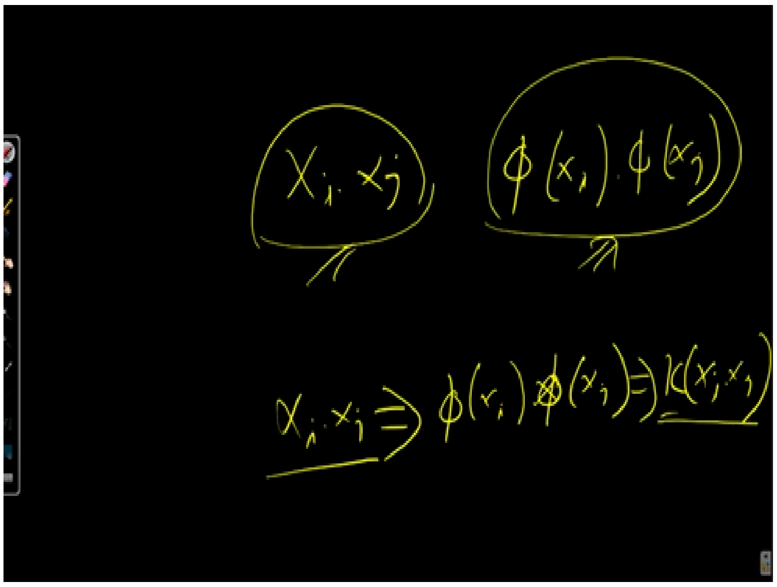
Basic idea

So, what we have done here? Then next what we tried doing is that like so what was our what to say classify function let me write my classifier function again. Classify function in the dual form so what was my classifier function? My classifier function was λ i y i x i . x + d i = 1 to m this is my classifier function and a dual form. So, now this is in this classifier function what we see this is the dot product between the this is one test data for each days that I will see a dot product of this x is a test data and this is my training data.

So, with the test data is a dot product of each and every training data. So, this is what when I try to find a class of x this is what we do dot product of this. Now when we; talk of dot product. What is dot product? Dot product actually we try to find out the similarity. When we find out a dot product between two vectors so if we try to find out the similarity is not it. The similarity of the two vectors x i . x so, now when we have transformed it into a difference space.

So, this equation again and in the transform thing dimensional transforms dimensional space I can write this φ of x i this is the transform data and this is φ of x + b just I have transformed this is not it. So, here this is the; if now the condition is if the x i . x gives a similarity measure. Similarly, φ i x φ x i . φ x is also the similarity measure between φ x i and φ x this is the basic idea between behind a kernel trick, this is the basic idea. This is the basic idea behind a kernel trick.

(Refer Slide Time: 11:26)


$X_i \cdot x_j$
$\phi(x_i) \cdot \phi(x_j)$
$X_i \cdot x_j \Rightarrow \phi(x_i)\phi(x_j) \Rightarrow K(x_i, x_j)$

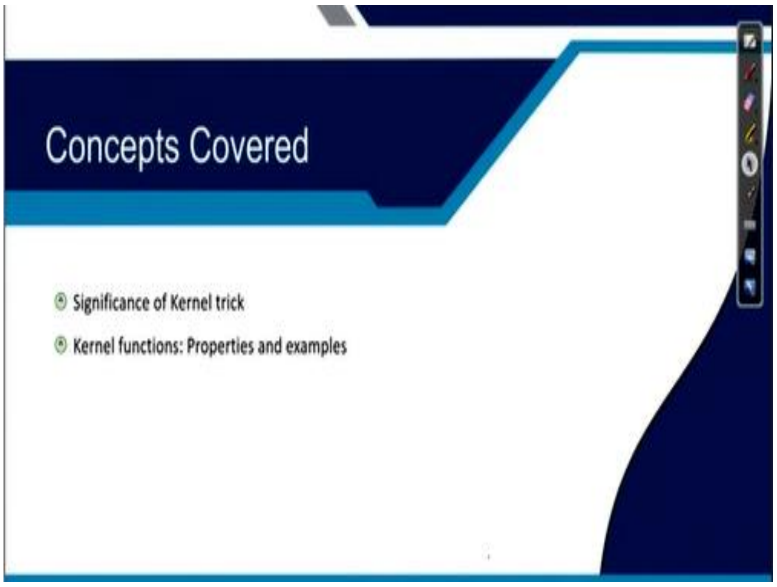
So, what we went on to do is that so, now first thing is that when we have seen this X i X i . X j and φ of X i . φ of X j so this gives the similarity between these two data and these two vectors this gives the similarity between these two vectors the what is φ X i is a transform vector of X i φ X i is the transform vector of X j. Now this also gives similarity this also gives similarity. So, we were wondering is there any correlation between these two.

And then we went on to find that this two are correlated φ of X i . φ of X j is very much correlated to X i . X j so, because then we can do, we could write that X i . X j implies to φ of X i . φ of X j implies to K of X i . X j, what does this mean. This kernel functions physically implies similarity this kernel function K it physically implies similarity in the transform space consider using the input space.

So, I am repeating again what does this kernel function it implies similarity in the transform states φ of X i and φ of X is a similarity in the transform space. So, K of X i it gives the similarity in the transform space considering the input space taking the input space. We have taken the taken the data of the input the dimensional in the input space where it is whatever dimensional it is whatever two dimensional three dimension it is there have taken that data.

And now we do not literally have to go to the transform states without going I can use a kernel function where I can see the similarity of the transform space data taking the input of the input dimensional input space. So, this is the now what is this K function? K function is such a function we are taking the input space data I can tell that similarity of the transform space data. Now what is this K function. So, essentially, we need to find this K function. So, that is what that is where we have stopped in my last lecture.

(Refer Slide Time: 13:54)

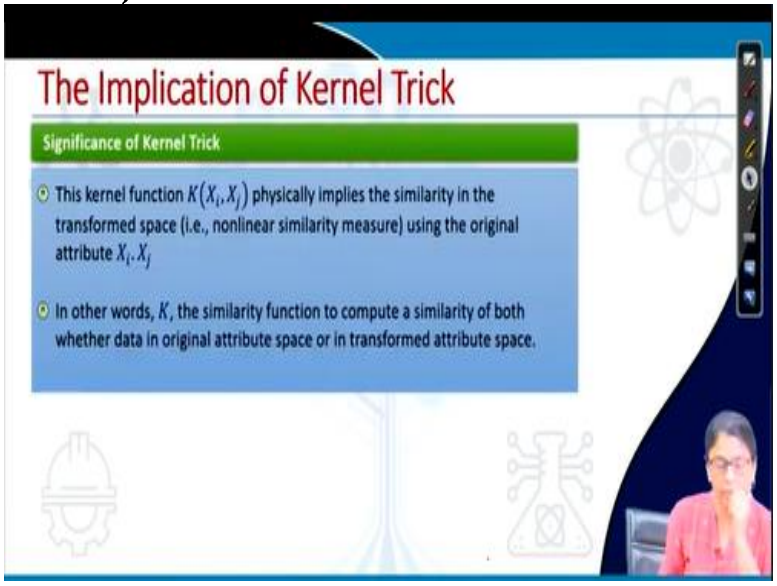


Concepts Covered

- Significance of Kernel trick
- Kernel functions: Properties and examples

So, now starting from that so, what we will see is that the significance of the kernel trick how this we could use this kernel tree and we will also see different properties and some examples of kernel function.

(Refer Slide Time: 14:06)



The Implication of Kernel Trick

Significance of Kernel Trick

- This kernel function $K(X_i, X_j)$ physically implies the similarity in the transformed space (i.e., nonlinear similarity measure) using the original attribute $X_i . X_j$
- In other words, K, the similarity function to compute a similarity of both whether data in original attribute space or in transformed attribute space.

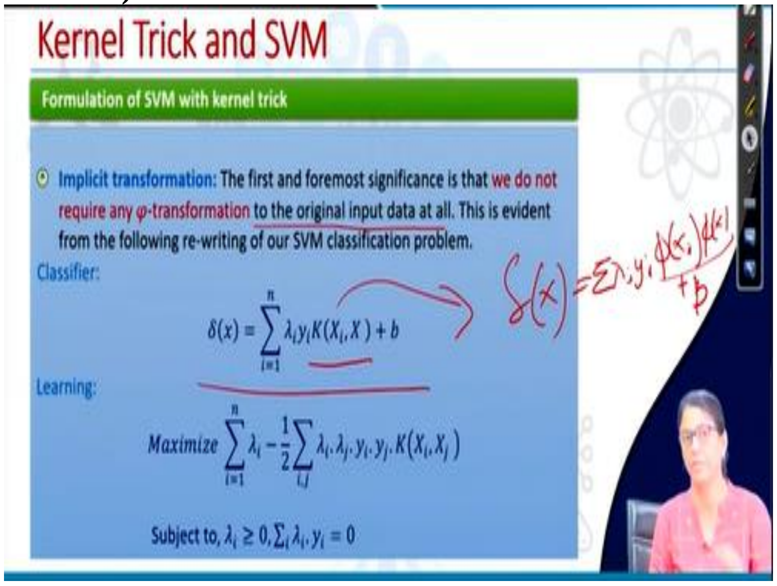
So, basically what is the significance of kernel trick which I have already mentioned let us read it out the kernel function K of X i . X j physically implies the similarity in the transform space. What does K X i X j thus? It implies the similarity in the transform space that is the non-linear similarity measure using the original attribute. What is using? It is using the original data original X i X j like in my last lecture I have given you an example two fruits in the Indian soil.

If I know the similarity between these two, similarly, if I have some function which can tell the similarity between two fruits and say UK soil, I do not need to go there I do not need to know what I do I can just tell the similarity between two fruits. If I have such function which can tell the similarity in the transform space there my transform space is a UK soil. So, if I can tell the similarity in the transforms is taking the using the original attribute using the original attribute.

So, if I can know that; then I do not need to transform it my requirement is that why I am transforming a data so that I can the non-linearity I can make it linear. So, and actually what I am doing when I am trying to do the classification, I am basically trying to find out a similarity between two factors is not it what are this input inputs are nothing but the different vectors. So, when I am trying to classify whether it test data falls in this or this.

I am trying to find out how much less data is similar to which the different two types of data. So, I am just finding the similarity so that is it. In other words, K the similarity function to compute a similarity of both whether data in the original attribute space or in that transform attribute space.

(Refer Slide Time: 16:01)



Kernel Trick and SVM

Formulation of SVM with kernel trick

- Implicit transformation: The first and foremost significance is that we do not require any φ-transformation to the original input data at all. This is evident from the following re-writing of our SVM classification problem.

Classifier:

$$\delta(x) = \sum_{i=1}^{n} \lambda_i y_i K(X_i, X) + b$$

$$\delta(x) = \Sigma \lambda_i y_i \phi(x_i)\phi(x) + b$$

Learning:

$$\text{Maximize} \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i . \lambda_j . y_i . y_j . K(X_i, X_j)$$

Subject to, $\lambda_i \geq 0, \sum_i \lambda_i . y_i = 0$

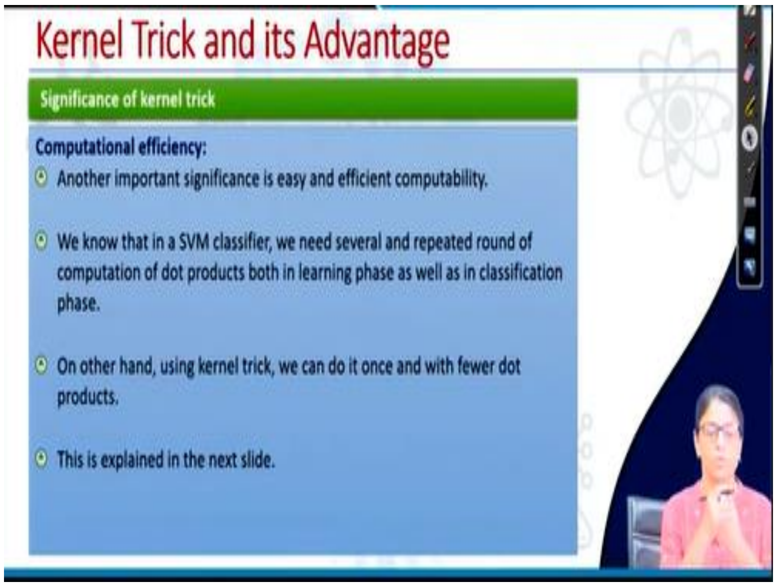
So, the first and foremost significance is that we do not require any φ transformation. We have such a function which can give the similarity and my transform space taking the data from the input space original input space. So, we do not require any φ transformation to the original input data at all. So, now what is my classifier? This is my classifier. Earlier classify λ i y i φ of X i φ of X is not it.

This I was writing as λ of X; the X is the test data λ i y i φ of x i φ of X + b so φ of X i and φ of X means transforms this. Now I do not need this transform what is that instead of this I can write K of X i, X the similarity between these two factor K I have X i and X and subjected so this is my classifier now. And what is the learning what else I need to maximize this we have already seen the dual form of the lagrangian.

Now in the last lecture we have seen. So, I need to maximize this subject to this so now my problem has become this if my data is linear fine if my data is non-linear so I will use this as my what to say optimization problem maximize this and subject to this. If when I will do this, I will be able to find out the different values of the lambdas different values of the basically I need to find out the values of λ only.

Once I find out the values here only λ is in the primal form, I had I required W I require to find out W I require to find out B in this form I only need to find out the values of λ. So, that is sufficient for me to find out the expression of the hyper plane. And once I know the value of Λ then I can use the classifier to find out the class of a test data.

(Refer Slide Time: 18:09)



Kernel Trick and its Advantage

Significance of kernel trick

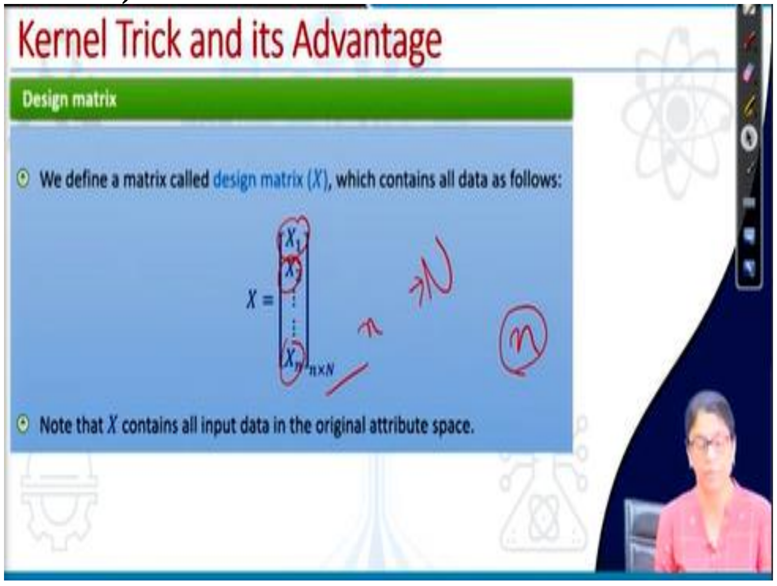
Computational efficiency:

- Another important significance is easy and efficient computability.
- We know that in a SVM classifier, we need several and repeated round of computation of dot products both in learning phase as well as in classification phase.
- On other hand, using kernel trick, we can do it once and with fewer dot products.
- This is explained in the next slide.

So, now what is the significance? One important significance it is easy and efficient computability. So, this first of all this is we do not have this already we have learned a last lecture we do not need W we do not need B just we need λ. We know that in SVM classifier we need several and repeated round of computation of dot product both in the learning phase as well as classical classification phase.

So, dot product X i . X X i . X j so, we need to do repeated round of this dot product both as in the learning phase as well as the and classification phase. But on the other hand, using the kernel trick we can do it once and with fewer dot product we will see it how.

(Refer Slide Time: 18:59)



Kernel Trick and its Advantage

Design matrix

- We define a matrix called design matrix (X), which contains all data as follows:

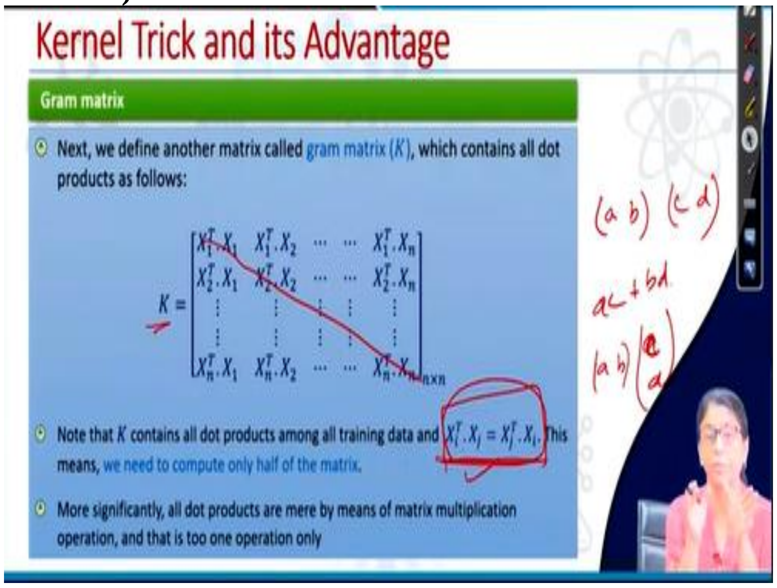
$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ \vdots \\ X_n \end{bmatrix}_{n \times N}$$

- Note that X contains all input data in the original attribute space.

So, we define a matrix called we are calling it a design matrix which contains the data as follows. So, this is my design matrix where X 1 to X n these are the input data. So, total my total how many input data I have that means I can say it is the training data that is n training data or how many observation I have along with the class level that I call I may call it instead of trading data I can call it observation also along with the class level.

So, I have total n data and it is in how much dimensional say it is an N dimensional that means each X 1 is a vector of N dimensional. Each input data this is one input data this is one input data so total there are small n data and each data each dimension of data is capital N so; I am defining a design matrix where my design matrix is small n cross begin. And so, where X contains all input data in the original input space.

(Refer Slide Time: 19:58)



Kernel Trick and its Advantage

Gram matrix

- Next, we define another matrix called gram matrix (K), which contains all dot products as follows:

$$K = \begin{bmatrix} X_1^T.X_1 & X_1^T.X_2 & \cdots & \cdots & X_1^T.X_n \\ X_2^T.X_1 & X_2^T.X_2 & \cdots & \cdots & X_2^T.X_n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ X_n^T.X_1 & X_n^T.X_2 & \cdots & \cdots & X_n^T.X_n \end{bmatrix}_{n\times n}$$

- Note that K contains all dot products among all training data and $X_i^T.X_j = X_j^T.X_i$. This means, we need to compute only half of the matrix.
- More significantly, all dot products are mere by means of matrix multiplication operation, and that is too one operation only

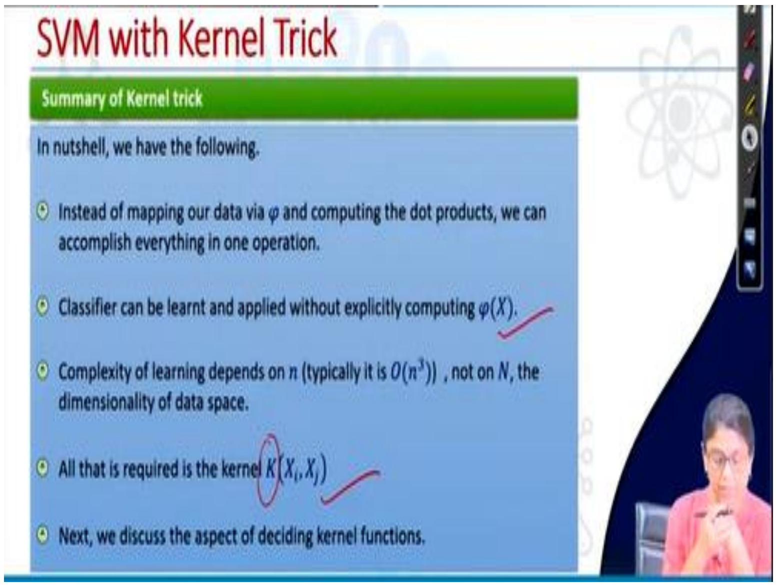
Now what I will do I will define another matrix which I call a gram matrix this basically this is K what K we are talking about which contains all dotproducts as follows. We are calculating all the dotproducts what is dot product X so, when they suppose these two vectors a b two vectors say c d how do we do the dotproducts what do we do the dot product you know this so, it is ac + bd is not it. So, now here if I write in this way, we get the same thing was not it.

So, that is all that is what we are trying we are finding out a dot product between all this input observation so this is how we are doing X 1 T X 1 X 1 T X 2 similar this one is all with X 1 then the second row with X 2 X 2 T in X 1 X 2 T X 2. So, the third row with X 3 the third input data that way we find all the dot product. Now if we see here the note that K contains all that purpose among all training data that means among all observation.

And note that this is definitely X i transpose into X i X j = X j transpose into X i this is obviously is not it, this is an obvious thing. So, when this is true that means, we do not need to compute the whole thing only if we compute half of it if you compute this, this value we will get because this is this, this is equal to this we are computing this and we are storing it. We are calling this as a gram matrix. This means we need to compute only half of the matrix.

More significantly all dotproducts are made by means of matrix multiplication operation and that is to one operation only. So, now this is how we have computed all the dotproducts and essentially, we try to find a similarity between these two and we have kept it.

(Refer Slide Time: 22:13)


SVM with Kernel Trick

Summary of Kernel trick

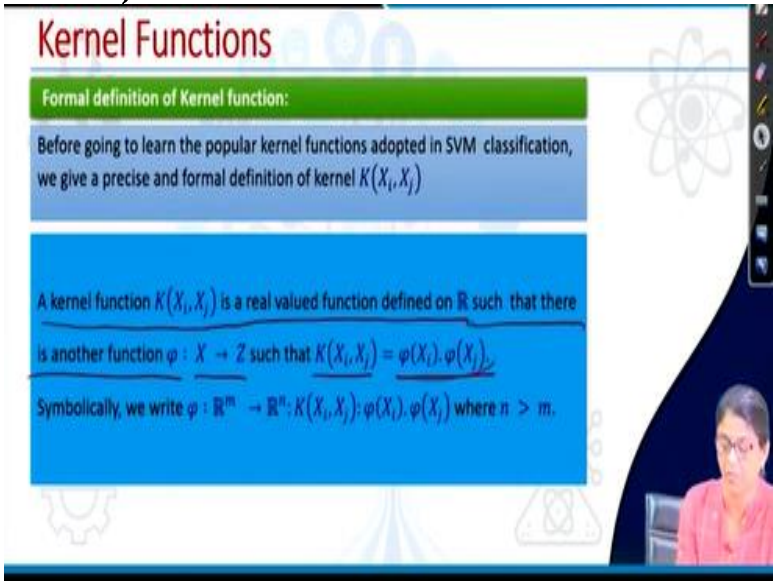
In nutshell, we have the following.

- Instead of mapping our data via φ and computing the dot products, we can accomplish everything in one operation.
- Classifier can be learnt and applied without explicitly computing $\varphi(X)$.
- Complexity of learning depends on n (typically it is $O(n^3)$), not on N, the dimensionality of data space.
- All that is required is the kernel $K(X_i, X_j)$
- Next, we discuss the aspect of deciding kernel functions.

So, in nutshell we have the following, instead of mapping our data via φ and computing the dotproducts we can accomplish everything in one operation. Classifier can be learnt and applied without explicitly computing φ of X and here the complexity of learning depends on the number of input instance number of observation number of training data. It does not depends on the dimensionality because it is a simple matrix and **(()) (22:40).**

It does not depends on the dimensionality it just depends on how many input data we have. So, it depends on N typically order of n to the power 3 and all that is required is the kernel all that is required is that we need to do this the risk K of X i X j is it simple exercise is a simple dot product or whatever what we need to do we need to find this kernel function. So, next we discuss the aspect of deciding kernel function.

(Refer Slide Time: 23:10)


Kernel Functions

Formal definition of Kernel function:

Before going to learn the popular kernel functions adopted in SVM classification, we give a precise and formal definition of kernel $K(X_i, X_j)$

A kernel function $K(X_i, X_j)$ is a real valued function defined on $\mathbb{R}$ such that there is another function $\varphi : X \rightarrow Z$ such that $K(X_i, X_j) = \varphi(X_i).\varphi(X_j)$.

Symbolically, we write $\varphi : \mathbb{R}^m \rightarrow \mathbb{R}^n : K(X_i, X_j) : \varphi(X_i).\varphi(X_j)$ where $n > m$.

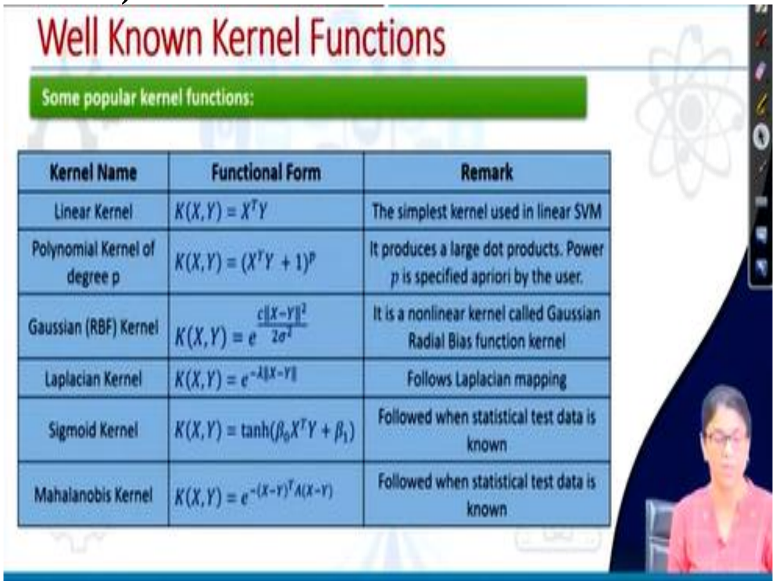
So, before going to learn the thing different type of kernel function let us formally define kernel function. This is the formal definition you see the formal definition of the kernel function. A kernel function K of X i, X j is a real valued function defined on R definitely it is a real valued function is not it defined on R such that such that there is another function φ. So, which such as φ of X gives Z such that K of X i X j = φ of X i . φ of X j.

So, this is the definition of kernel function if I formally define it. Please go to it again a kernel function K X i X j is a real valued function defined on R such that there is another function φ such that X gives Z φ of X gives that such that K of X i X j = φ of X i X j. K of X i X j physically implies that it gives a similarity in the transform stress that is it gives the similarity between φ of X i and φ of X j, what is φ of X i and φ of X j?

Φ of X i and φ of X j is the vectors in the transformation the input vector centre transform space. So, K of X i as this gives the similarity in this transform space how does it give the similarity for similarity it takes the input attributes in the input space it takes the vectors in the input space. So, symbolically we can write R of m, R m it is transforms transform to R n where n is greater than m.

Dimensionality n is dimensionality of m where here dimensionality is m it gets transformed to n dimensionality where n is greater than n and K of X i X j = φ of X i . φ of X j this is the formal definition of kernel function. So, now I think you can understand you understood what is kernel function and we have defined a formally same thing just a formal definition.

(Refer Slide Time: 25:20)



Well Known Kernel Functions

Some popular kernel functions:

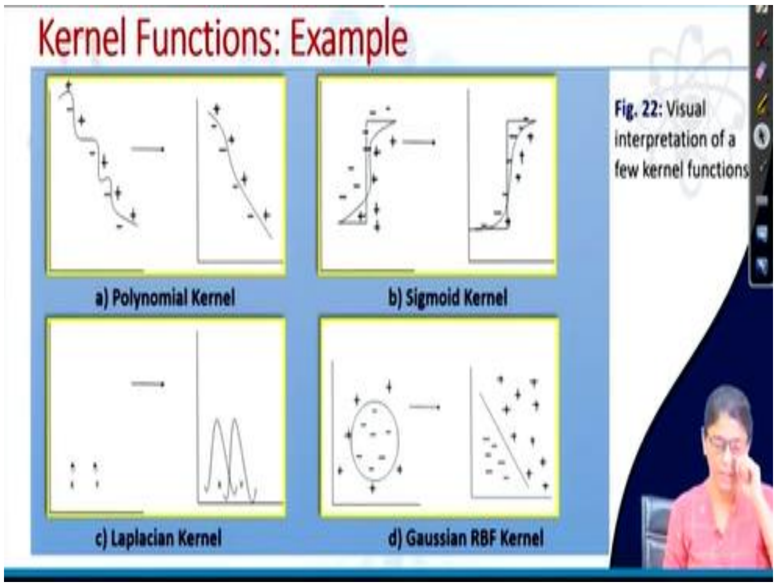
Kernel Name	Functional Form	Remark
Linear Kernel	$K(X,Y) = X^T Y$	The simplest kernel used in linear SVM
Polynomial Kernel of degree p	$K(X,Y) = (X^T Y + 1)^p$	It produces a large dot products. Power p is specified apriori by the user.
Gaussian (RBF) Kernel	$K(X,Y) = e^{\frac{c\|X-Y\|^2}{2\sigma^2}}$	It is a nonlinear kernel called Gaussian Radial Bias function kernel
Laplacian Kernel	$K(X,Y) = e^{-\lambda\|X-Y\|}$	Follows Laplacian mapping
Sigmoid Kernel	$K(X,Y) = \tanh(\beta_0 X^T Y + \beta_1)$	Followed when statistical test data is known
Mahalanobis Kernel	$K(X,Y) = e^{-(X-Y)^T A(X-Y)}$	Followed when statistical test data is known

So, these are some popular kernel function. So, what we have just shown this was this kernel function this is a popular kernel function. So, we can use any of the kernel function now in a given set of data we do not know which kernel function will work properly. So, it will just we will try each and every kernel function if we already notice that of this type and accordingly, we will select a kernel function which is suitable for that data.

But usually, it is very difficult to know data of which type we have our training data wafer observation but from the observation it is very difficult to make out it the data is of what type. Data is a linearly separable written a nonlinear separable or what type it follows which distribution nothing so, better find out try with all sort of kernel and then the kernel fish gives the best result that we will be using for our and for SVM of that particular data process.

So, these are the different I am not going to the details of this so, these are the different kernel functions this is a functional form and it is wave what is used it is given here. You can go the each of the for learning each of the kernel function if you can just Google it and you can learn each and every kernel function.

(Refer Slide Time: 26:35)



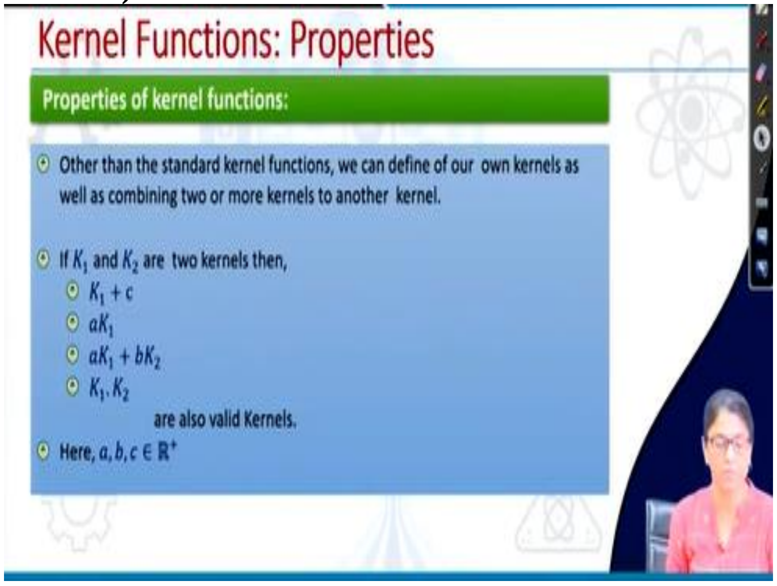
Kernel Functions: Example

Fig. 22: Visual interpretation of a few kernel functions

So, and like here is the inner figure I have just shown what is what effect we get if we use the polynomial kernel. So, this is the polynomial kernel this p value, p value is again we can you can take any value you can take p 1 to any value which will have to do it and see which value will give the better result. So, in polynomial kernel this was initially our data was in this form now in the transform space if we see we got the data in this now which can be easily linearly separable.

Similarly, if I use the sigmoid kernel if data is in this form my data got data has transformed to this space which is again linearly separable. Similarly, this this these are the different you can see this data if it is totally non-linear data if I use the Gaussian rbf 10 **(()) (27:21)** and radial based function kernel so it my data gets transformed this.

(Refer Slide Time: 27:32)



Kernel Functions: Properties

Properties of kernel functions:

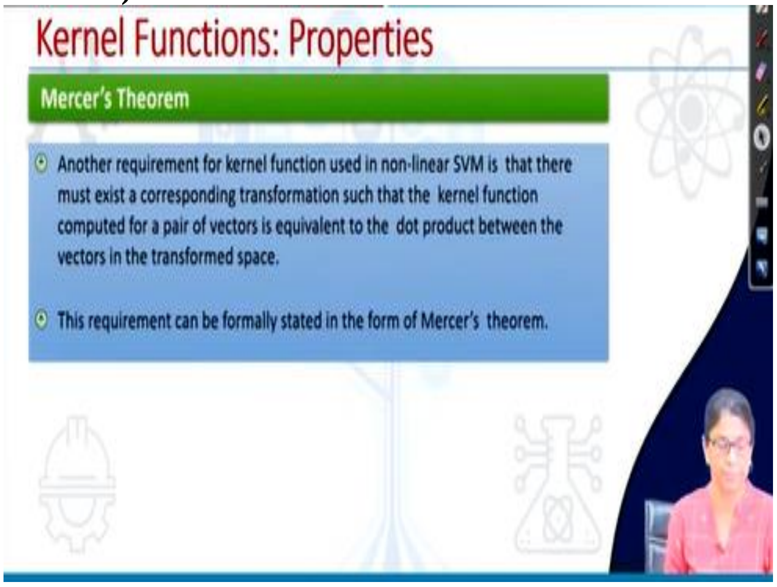
- Other than the standard kernel functions, we can define of our own kernels as well as combining two or more kernels to another kernel.
- If K_1 and K_2 are two kernels then,
  - $K_1 + c$
  - aK_1
  - $aK_1 + bK_2$
  - $K_1 . K_2$

  are also valid Kernels.
- Here, $a, b, c \in \mathbb{R}^+$

So, some properties of kernel function. So, other than the standard kernel function we can define of our own kernel as well as combining two or more kernels to another kernel. These are the standard kernel function we can define our own kernel function also. But our kernel function has to satisfy certain criteria just arbitrarily we cannot define it has to satisfy certain criteria and that way we can define our cone kernel function.

Moreover, given a kernel function by joining two or more kernel we can have another kernel we do not know which kernel will work best for a type for class of for a what to say category for a group of data. So, if K 1 and K 2 are kernel then this also can be a kernel this also this also different combination where a b c is nothing as just positive real numbers. These are all these are valid kernels we can have some of this kernel in this way by using different constant a b c which belongs to the real number and positive space.

(Refer Slide Time: 28:29)



Kernel Functions: Properties

Mercer's Theorem

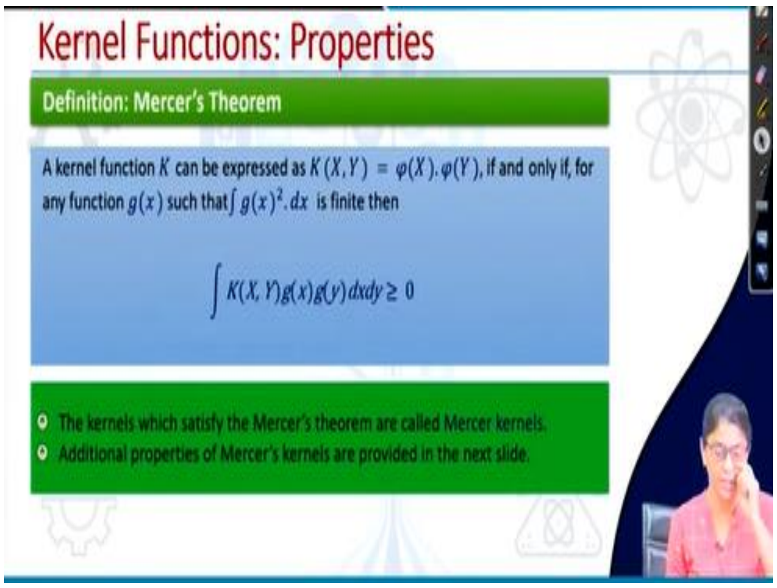
- Another requirement for kernel function used in non-linear SVM is that there must exist a corresponding transformation such that the kernel function computed for a pair of vectors is equivalent to the dot product between the vectors in the transformed space.
- This requirement can be formally stated in the form of Mercer's theorem.

So, in kernel function used in non-linear SVM one of the requirements is that there must exist which I already talked basically. There must exist a corresponding transformation such that the kernel function computed for a pair of vectors is equivalent to the dot product between the vectors in the transform space. This is the main requirement of the kernel function for classification purpose what we need for classification purpose.

What is the main requirement of kernel function is that there must exist so we have what we have seen and we are repeating it again and again there must exist a corresponding transformation such that the kernel function completed for a pair of vectors in the non-transform space will give us the similarity in the transform space. Such that a kernel function computed for a pair of vectors is equivalent mark this term is equivalent to the dot product between the vectors in the transform space.

This requirement can be formally stated in the form of Mercer's theorem we have a theorem called Mercer's theorem.

(Refer Slide Time: 29:53)



Kernel Functions: Properties

Definition: Mercer's Theorem

A kernel function K can be expressed as $K(X,Y) = \varphi(X).\varphi(Y)$, if and only if, for any function $g(x)$ such that $\int g(x)^2.dx$ is finite then

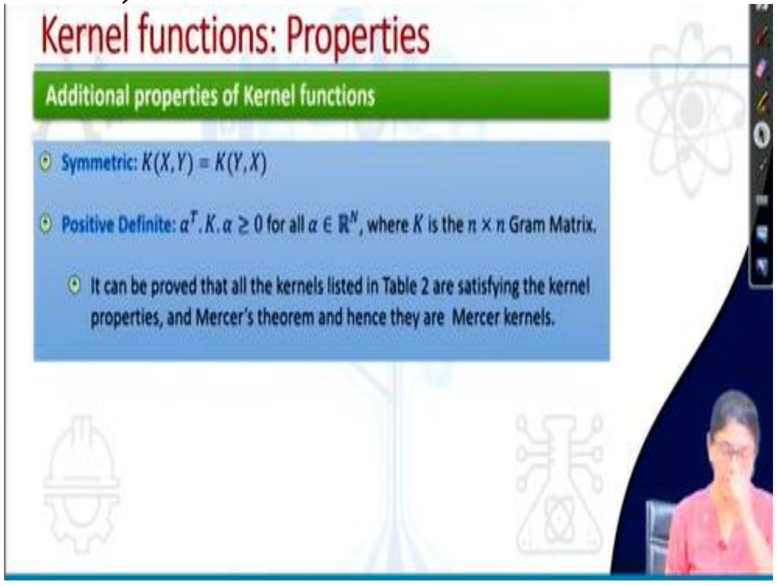
$$\int K(X,Y)g(x)g(y)dxdy \geq 0$$

- The kernels which satisfy the Mercer's theorem are called Mercer kernels.
- Additional properties of Mercer's kernels are provided in the next slide.

So, what does martial theorem says if kernel function K can be expressed as this if and only for any function g x any function g x you can normally is for any function g x such that integration of g x ^{2} dx is finite you can select any function g x but that it should satisfy this should be finite and this should be greater equals to zero. Then we as here we have seen that we can define our own kernels we can very well define our own kernel.

But our kernel function should satisfy this it should satisfy the Mercer's theorem what is this Mercers' theorem this is the Mercer's theorem it should satisfy this. The kernel's theorem which satisfy the Mercer's theorems are called Mercer kernel.

(Refer Slide Time: 30:55)



Kernel functions: Properties

Additional properties of Kernel functions

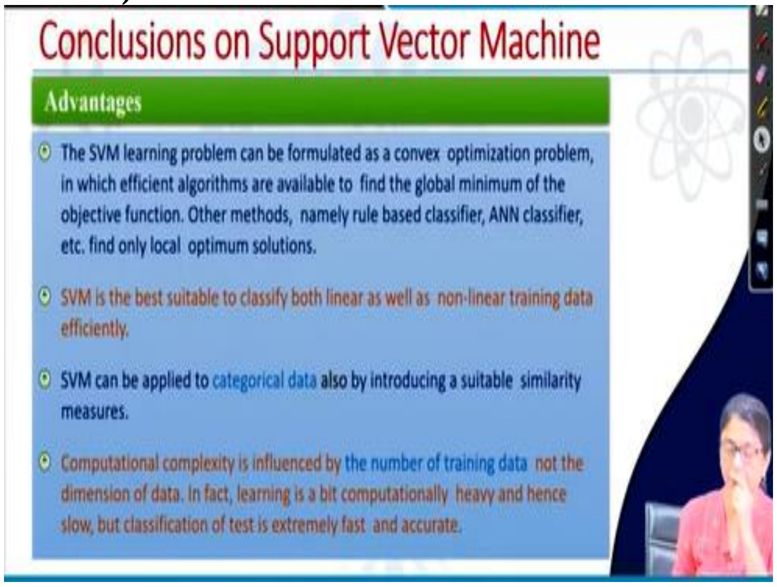
- Symmetric: $K(X,Y) = K(Y,X)$
- Positive Definite: $\alpha^T.K.\alpha \geq 0$ for all $\alpha \in \mathbb{R}^N$, where K is the $n \times n$ Gram Matrix.
  - It can be proved that all the kernels listed in Table 2 are satisfying the kernel properties, and Mercer's theorem and hence they are Mercer kernels.

There are some additional properties which will be seen here the kernel one thing is our kernel function should be symmetric that is the reason why we have computed half of the matrix within compute the other half. If you compute the half, we can get the other half. So, in the example what we have seen. So, that is one kernel function simple dot product so it has to be symmetry and it should be positive definite.

What does positive definite means? A to the power T K . α should be greater equals to 0 where α is this so, α ^{T} . K . α is greater equals to 0 where α belongs to this where K is a n cross n gram matrix. It can be proved that all the kernels that we have listed in table 2 this is the table two this is the table two oh sorry I did not mark it here table. So, anyway this is the table two are satisfying the kernel properties.

And Mercer's theorem enhance their Mercer's kernel all the kernel function that we have mentioned all those are called Mercer's kernel because they are satisfying the kernel properties water is properties these two properties and as well as the Mercer's theorem.

(Refer Slide Time: 32:24)



Conclusions on Support Vector Machine

Advantages

- The SVM learning problem can be formulated as a convex optimization problem, in which efficient algorithms are available to find the global minimum of the objective function. Other methods, namely rule based classifier, ANN classifier, etc. find only local optimum solutions.
- SVM is the best suitable to classify both linear as well as non-linear training data efficiently.
- SVM can be applied to categorical data also by introducing a suitable similarity measures.
- Computational complexity is influenced by the number of training data not the dimension of data. In fact, learning is a bit computationally heavy and hence slow, but classification of test is extremely fast and accurate.

So, now in coming to the end of it so what is the advantage of support vector machine. The support vector machine learning problem can be formulated as a convex optimization problem. We have seen it can be formulated as a convex optimization problem which we have solved it using then like then Lagrangian multiplier in which efficient algorithms are available. One of the efficient algorithm is like the Lagrangian multiplier to find the global minimum of the objective function.

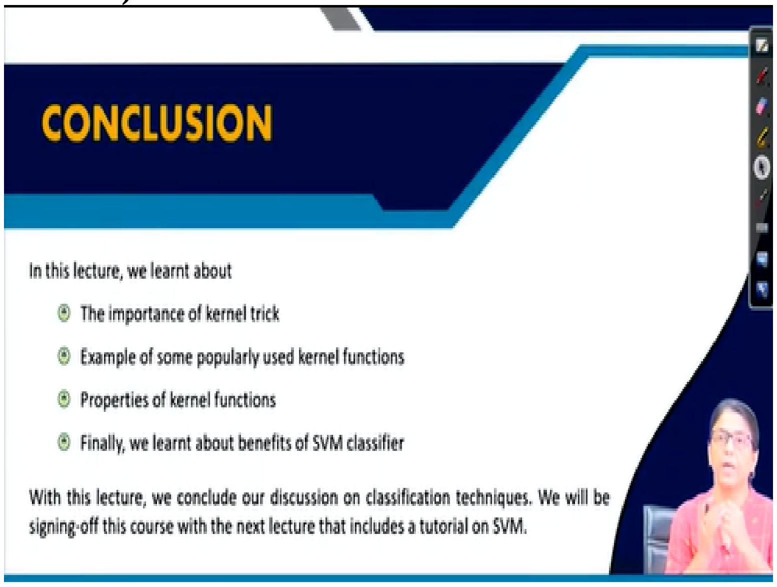
Other methods namely rule based classifier ANN classify etcetera finally local optimum solution. This things ANN classifier rule based classifier it finds optimal solution when we try to find out the classified machine SVM that function. So, it using this convex optimization problems and what we get we get the global optimum. Then SVM is best suitable for classify both linear as well as non-linear training data efficiently.

Linear training data we have already seen non-linear trending determines using this kernel function it does it efficiently. SVM can be applied to categorical data also by introducing a suitable to similarity measures because category data we do not have those as such we do not have a value. So, we can introduce a suitability measures and accordingly it can be used for classification. Computational complexity is influenced by the number of training data we have already seen.

Computation complexities very much depend on the input space it does not depend on the dimension of the space not the dimension of the data. In fact, learning is a bit computationally heavy because computational heavy means we do not know which kernel function will be best suitable. So, and during the learning function we may have to try different learning function and if we have to use the polynomial kernel.

Then we have to use maybe we have to use on different values of p and find out which is best suitable and hence slow. So, learning is based a bit slow but once learning is done classification of text is extremely fast and accurate. So, these are the advantages of SVM.

(Refer Slide Time: 34:13)



CONCLUSION

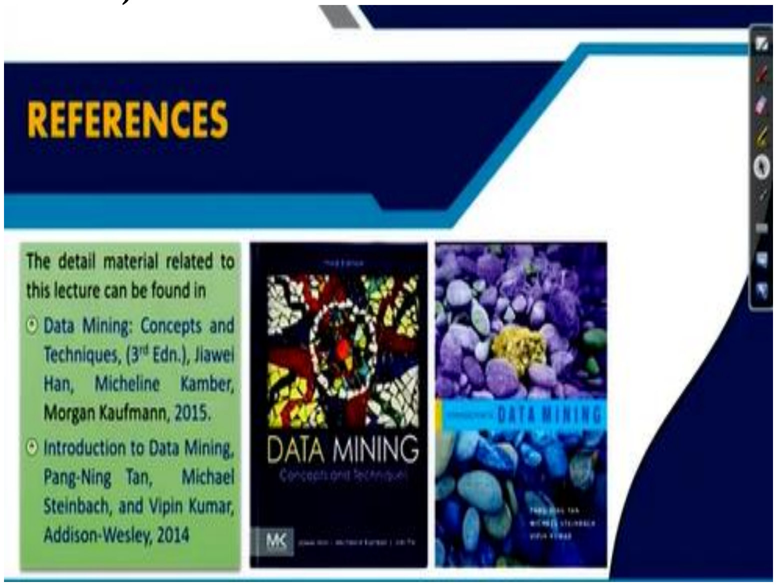
In this lecture, we learnt about

- The importance of kernel trick
- Example of some popularly used kernel functions
- Properties of kernel functions
- Finally, we learnt about benefits of SVM classifier

With this lecture, we conclude our discussion on classification techniques. We will be signing-off this course with the next lecture that includes a tutorial on SVM.

With this we come to the end of this lecture today. We have learned about the importance of the kernel trick example of some popularly used kernel function we have also seen the properties of kernel function and finally we have also mentioned the benefits of the SVM classifier. So, with this as I told this is the last lecture of this course means from the discussion on point of view and will be signing up this course with a tutorial class and my next lecture.

(Refer Slide Time: 34:44)



REFERENCES

The detail material related to this lecture can be found in

- Data Mining: Concepts and Techniques, (3rd Edn.), Jiawei Han, Micheline Kamber, Morgan Kaufmann, 2015.
- Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, Addison-Wesley, 2014

DATA MINING
Concepts and Techniques
MK
DATA MINING

Thank you, guys thank, you.