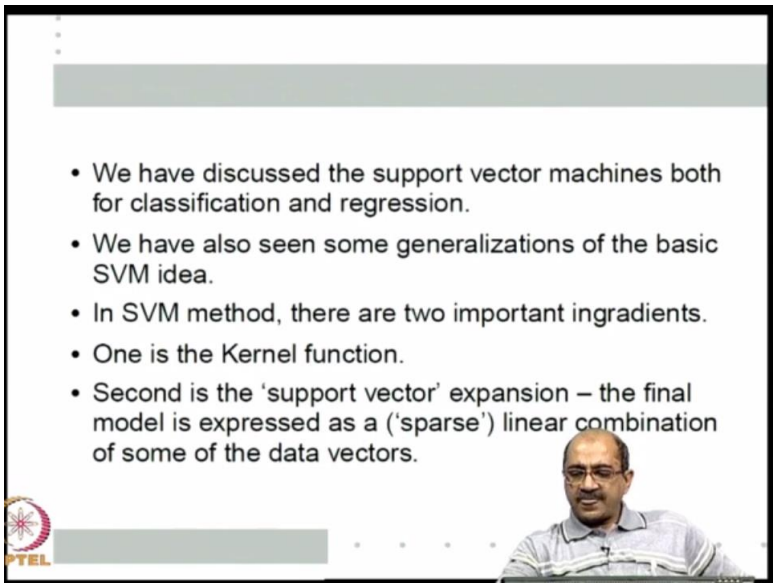
Pattern Recognition
Prof. P. S. Sastry
Department of Electronics and Communication Engineering
Indian Institute of Science, Bangalore

Lecture - 37
Positive Definite kernels; RKHS; Representer Theorem

(Refer Slide Time: 00:33)



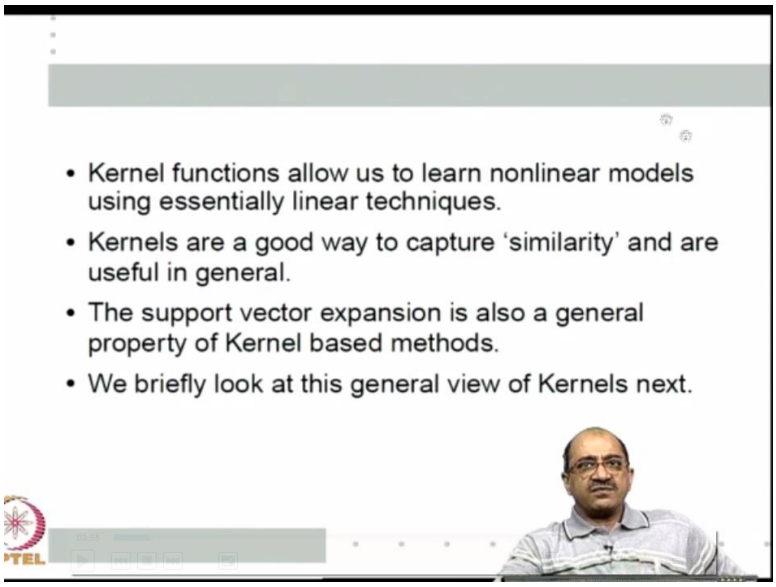
Hello and welcome to this next lecture in the pattern recognition course. We have been looking at support vector machines idea. This lecture, we will close this; we will look at kernels in general. Just to recall, we have looked at the support vector machines both for classification and regression. The basics of electro machine that glance optical hyperplane for a two class classification, and also seen how we can do support regression using epsilon in sensitive loss function. And we also seen some generalization of the basic SVM idea. For example, look at the new SVM the way to add b square to the primal objective functions of that the dual becomes very simple to solve. So, which is called successive over relaxation SVM.

So, we have seen some of the generalization of the basic SVM idea. So, to ultimately take some very broad look at what the SVM idea is added, there essentially two important ingredients in the SVM idea. So, if I am looking at only classification, then they told you the, the power of the algorithm comes from the fact that we are learning optimal hyperplane, which is what allows us to learn a classifier with very low true risk. And by using kernels to do inner products, we are able to learn a nonlinear classifier with

this thing, but if I look at everything all the SVM type ideas. So, essentially kernels is one important idea. And equally important idea, which is what allowed us to use the kernel idea in the first place is that the final function admits a representation in terms of kernel functions.

So, we seen for example, the in the SVM the final w is a linear combination of data vectors. And hence the final classification function of the discriminant function can be expressed as the linear combination of kernels of kernel function values. So, we can call this as support vector expansion. So, representing your final classifier say finite classifier function f X where sin of f X will be your classifier, representing that function f X as a nice kernel expansion. So, we will call that support vector expansion, so these two are the two ingredients that allowed us to solve nonlinear problem using linear techniques.

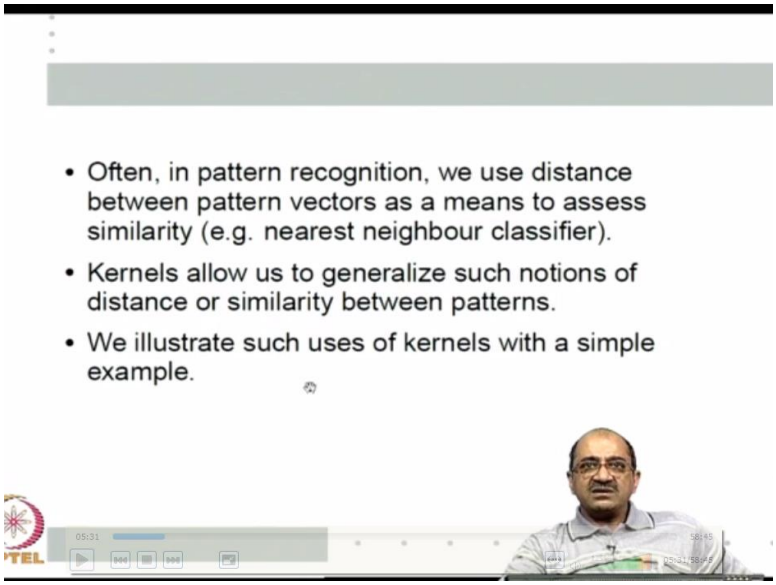
(Refer Slide Time: 03:15)



Now, both these are much general, much more general than what we have seen in the simple SVM method. Kernel functions in general allow us to learn nonlinear models using linear techniques and also a very good way to capture similarity that is useful in a general way. So, kernel functions can be thought of as capturing similarity, and dissimilarity between feature vectors. In the same way support vector expansion is also a very general property of kernel based methods. So, what we will do in this class is to look at this general overview of kernels. So, what I will do is I will first give you a very brief interaction or brief idea of what we mean by kernels at good way to capture

similarity. And then we look at positive different kernels in a little more detail to understand the kernel idea in a slightly more general fraction then what you seen so far.

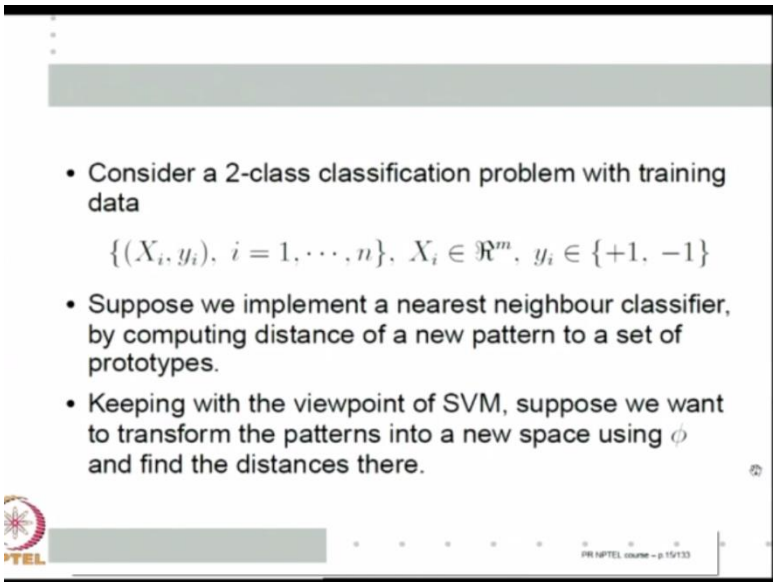
(Refer Slide Time: 04:16)



In pattern recognition, we always use distance as a means to access similarity that is one of the ways which you can do this. You know at the beginning of the course, we looked at nearest neighbor classifier. Nearest neighbor classifier I stored some prototypes and when you give me new pattern I find the distance the equilibrium distance of the new pattern to each of the prototypes whichever prototype is closest to I will put that in the class, we have seen that as an interesting very simple classifier. And if I have sufficiently many prototypes it is worst case probability of error is at worst twice that of the basic error.

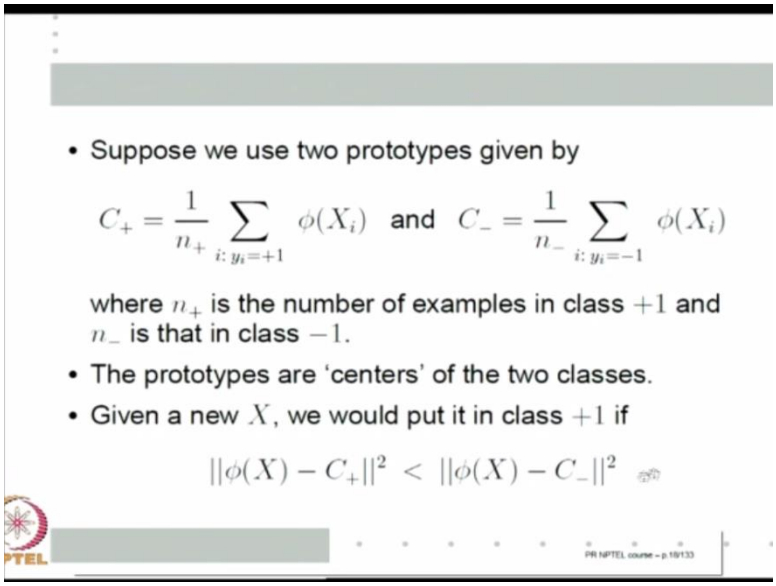
So, in that sense by simple classifier it does remarkably well. Now, kernel allow us to generalize such notions. So, the basic idea there is 2 patterns are closed to each other in a distance sense then they are also similar patterns. So, the question is what kind of distance is a good distance? So, we may have some similarity function, which we can think of as a kernel. And then we can use the kernel in place of the usual distance. We will, we will look at the very simple example to see how kernels can be used in a nearest neighbor classifier.

(Refer Slide Time: 05:38)



So, for the example let us take as usual two class classification problem, let us say X i y i X that is X 1 Y 1 X x n y n are the samples and the classes are plus 1 minus 1 and the feature vectors are in R m. Now, suppose we implement in a nearest neighbor classifier by computing distance from a set of prototypes. And the whole idea is that what are these SVM idea I have amount to that we do not do it is the original feature space. But we map the feature vectors to some other high dimensional space and do distance as this. So, the same way we using some phi to map X s and we are actually finding Euclidian distance R distance using a inner product in the (()) space of phi.

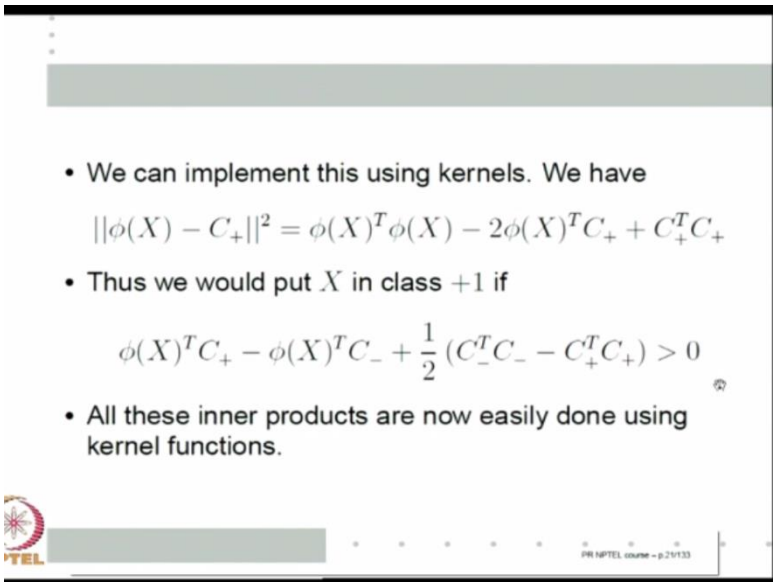
(Refer Slide Time: 06:30)



So, let us say we are using 2 prototypes, we call them as C plus and C minus, C plus is 1 by n plus summation of phi X i where i is such that y i is plus that it is nothing. So, n plus is the number of examples of class plus 1 similarly, n minus is the number of examples of class minus 1. So, this is nothing but the average of all the patterns of class one, not in the original space. But in the real space of phi because I have mapped the original feature vectors using phi to some other space, in that space C plus is the centre of all the class plus 1 examples.

Similarly, C minus is the centre of all the class minus 1 example. So, these are the 2 centers of the 2 classes. So, we use them as the prototypes, the idea is that if you give me a nu X i will go to the reign space of phi. So, I will find the distance between phi X and C c plus this is phi X minus C plus norm square. If that is less than phi X minus C minus norm square then I put it in class plus 1 other wise I will put it in class minus 1 this. So, this is a simple nearest neighbor classifier, but distance has not found in the original feature space but in the phi space.

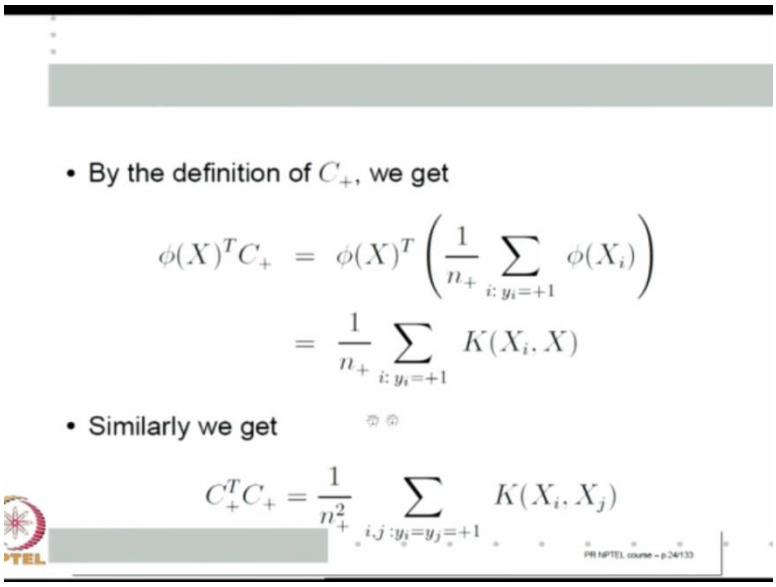
(Refer Slide Time: 07:46)



Now, we can do all this using kernels as follows, so basically we need to look at this phi X minus C plus whole square that is the distance. By expanding phi X minus C plus whole square becomes phi X transpose phi X minus 2 phi X transpose C plus C plus transpose C plus. Now, I want phi X minus C plus whole square greater than phi X minus C minus whole square. So, do this simple algebra this constant term cancel both

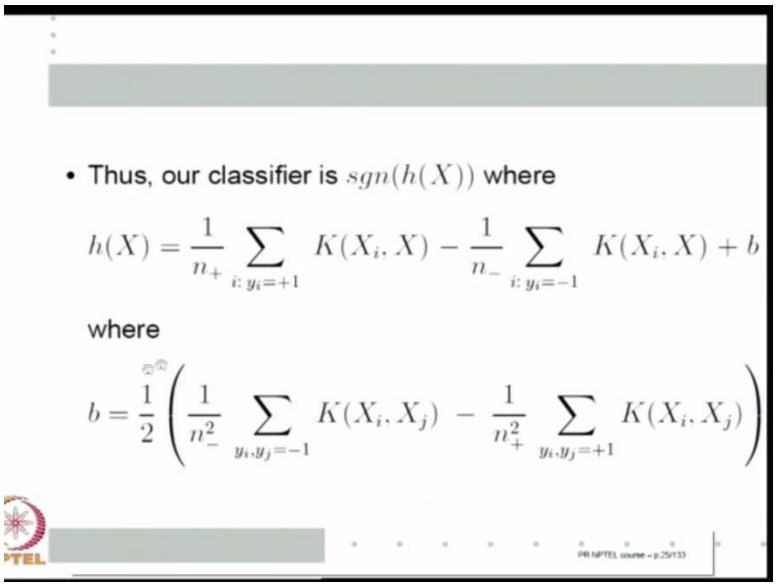
sides. So, finally, it means we will put X in class plus 1 if this quantity is greater than 0, phi X transpose C plus minus phi X transpose C minus plus some constant which is C C C minus transpose C minus, minus C plus transpose C plus. So, basically I have to do all these inner products to be able to implement my nearest neighbor classifier by first transforming all the patterns using the function phi. But the idea is that all these inner products can now be kernelised.

(Refer Slide Time: 08:45)



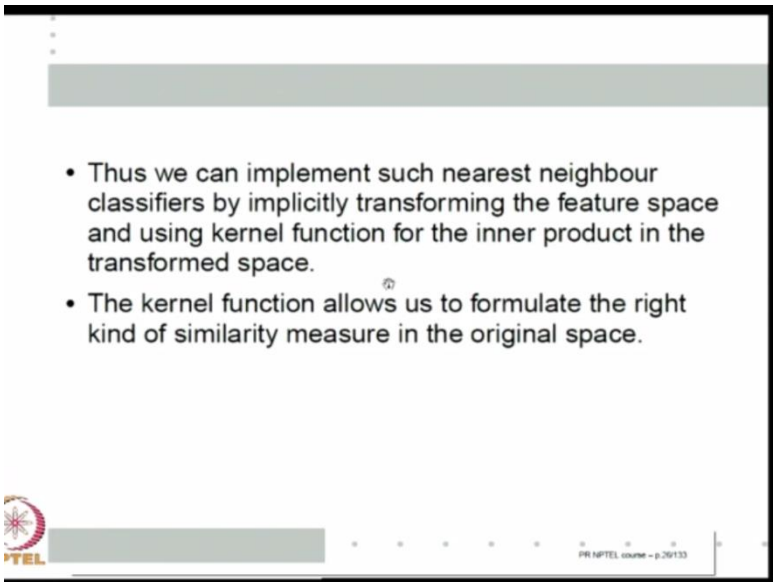
Very easy to see that they can be kernelised, what is phi X transpose C plus? Phi X transpose, this is my C plus. Now, push phi X inside then it becomes phi X transpose phi X i which is nothing but K X i X. In a similar way, what is my C plus? My C plus is this I want C plus transpose C plus, what I will ultimately I get is phi X i transpose phi X j for all i j pairs at both y i and y j are plus 1 and phi X i transpose phi X j is K X i X j. So, I get C plus transpose C plus like this similarly, I can write C minus transpose C minus.

(Refer Slide Time: 09:25)

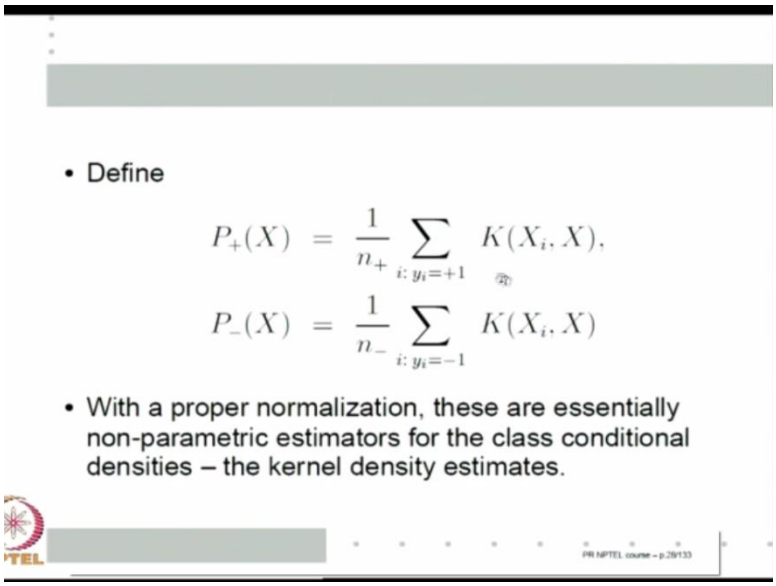


Which means finally, my nearest neighbor classifier is sin of h (X) where h (X) can be obtained like this. This is phi X transpose C plus this is phi X transpose C minus and the constant b is this plus C plus transpose C plus C minus transpose C minus. So, everything can be kernelised. So, what it meant is that I can for example, use phi to transform my features into some other high dimensional space and finding the actual distances there. So, if I have a suitable kernel then I can actually implement nearest neighbor classifier by implicitly transforming features into high dimensional space. Now, another way of looking at it is essentially the kernel captures. So, if I think phi X transpose whatever norm of phi X minus phi y is a good matrix for distance between pattern vectors X and y then correspondingly K X y is a good similarity measure.

(Refer Slide Time: 10:30)



(Refer Slide Time: 10:55)

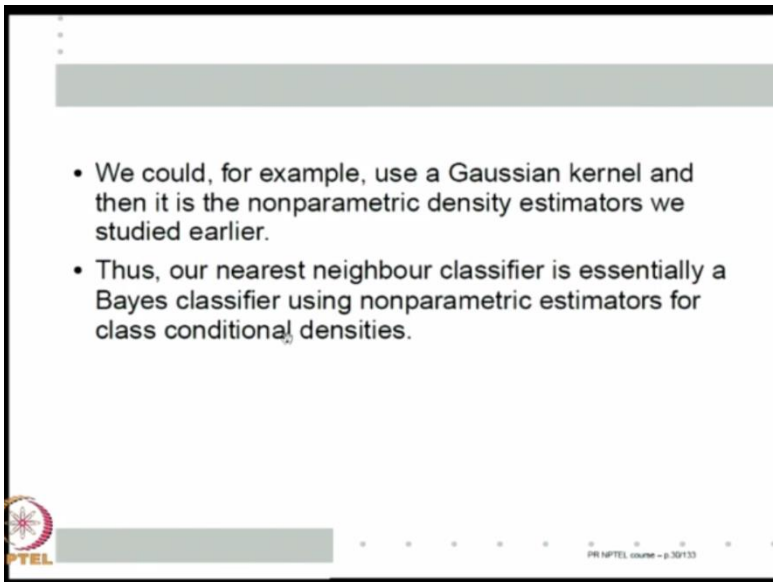


So, that is we can implement nearest neighbor classifiers by implicitly transform the feature space. And the kernel function allows us to formulate the kind of similarity measure in the original space. For example, if I am using a Gaussian kernel so, which means, that I am measure similarity over some spatial extent by (( )), my sigma parameter and so on. Another way of looking at is suppose I take this what do we call phi X transpose C plus phi X transpose C minus I call them P plus X and P minus X. Now, these are very familiar expressions, this is summed over all the examples of 1 class some kernel functions. So, there is some function defined here it is value at X is summed over

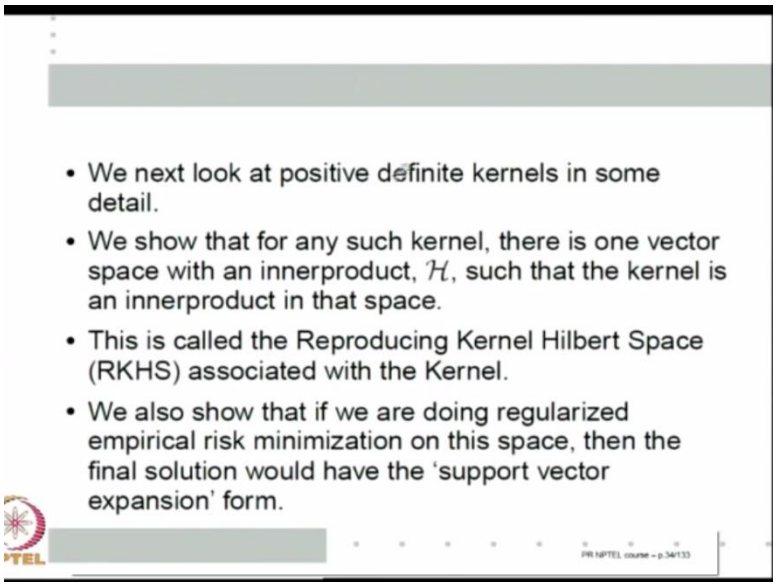
all examples of class 1, the kernel functions at each of the example points this X is same as this X that is the argument.

So, if for example, this was a a a proper window function; this is nothing but a non parametric estimate of density. For example, if I take this to be Gaussian that means I am putting a Gaussian centered at each of the X i's and some sum up and sum up all their contribution, this X that is my estimated class conditional density at X. So, with a proper normalization these are nothing but non parametric estimate for class conditional densities the kernel density estimates. So, we have seen the kernel density estimates and my final classifier is if P plus X minus P minus X is greater than something.

(Refer Slide Time: 12:16)



(Refer Slide Time: 12:52)



So, so all it means is that no these are nothing but non parametric density estimators that we have seen earlier these are the kernel density estimators. And in that sense what I call the nearest neighbor classifier using kernels is like a Bayes classifier, using a non parametric density estimator for class conditional densities using the kernel a popularly normalized kernel with the density estimate. So, in this sense once again you can see kernels give us some kind of a similarity metric all. Now, let us, let us look at f few more general theoretical details of kernels. We defined positive definite kernels earlier we, we actually looked at two different kinds of definition of kernels.

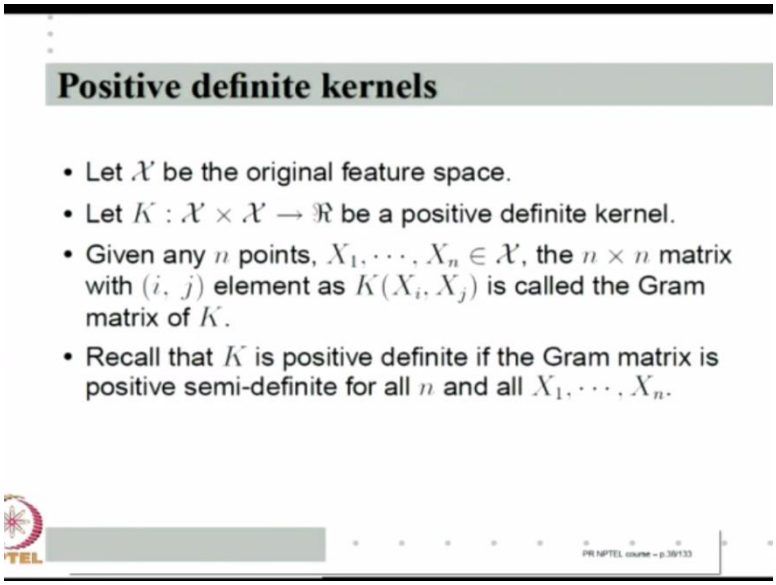
One of positive definite kernels, other kernel that satisfy Mercer's theorem, we just that anything that satisfies Mercer's theorem is such that there is always exist a phi and K X comma X prime will be phi X transpose phi X prime. We did it say, we did not prove this theorem, but for positive definite kernels in this class, we will show that there are always exist such a space and we actually construct this space. So, we look at positive definite kernel in some detail, because these are one of the most important kernel pattern recognition today. So, we show that for any such kernel there is one vector space, we call that a H script H which has an inner product on it.

So, a a vector space endure with inner product that we can construct such that the kernel is an inner product in that space. So, this particular space is called the reproducing kernel Hilbert space RKHS associated with this kernel K. And we will we will actually show

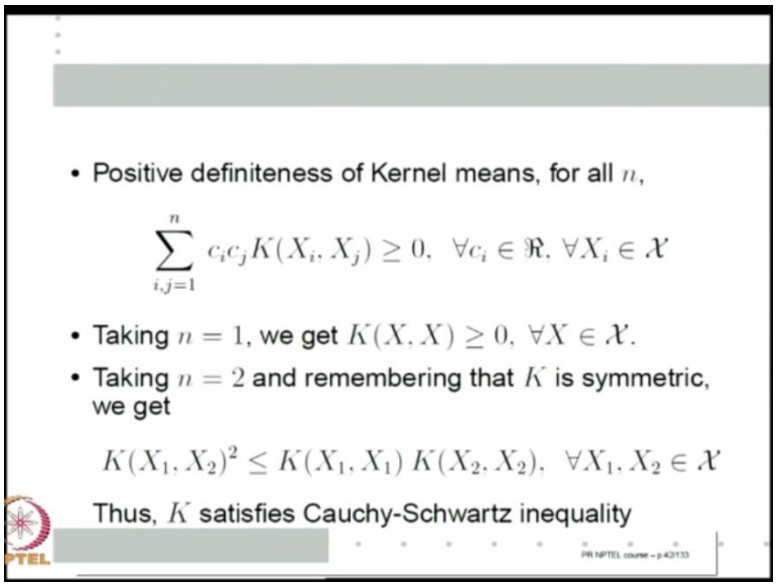
how to construct this space. And then we also show that if you are doing regularized empirical risk minimization under almost any loss function. Then the final solution would have a support vector expansion form. The, the, the rest of this lecture assumes some level of mathematical sophistication.

Specifically I am going to assume that everybody knows general vector space as norms basis orthogonal vectors, orthogonal complements of subspaces and so on and so forth. So, if any of you do not any of vector spaces may be it will be a little difficult to further part of the lecture. The, the, and the other, other hand the rest of this lecture is independent of the course. So, if you do not understand rest of the lecture can just skipped it.

(Refer Slide Time: 15:04)



(Refer Slide Time: 15:58)



So, let us start looking at positive definite kernels as earlier, let X be the original feature pace this script X. And K is a positive definite kernel that is K is a symmetric function on X class X real value symmetric function X class X which, which satisfies some conditions just recall that given any n points. Let us say X 1 given any n and n points X 1 X 2 X n in my feature space. The n by n matrix whose i jth element is K X i X j is called the Gram matrix of K. So, if I take all the X i X j's and arrange them as n by n matrix would be a symmetric matrix and recall that K is a positive definite kernel. If the gram matrix is positive semi definite that is it is quadratic form is always greater than or equal to 0 for all n and all points X 1 to X n, this is what we defined earlier.

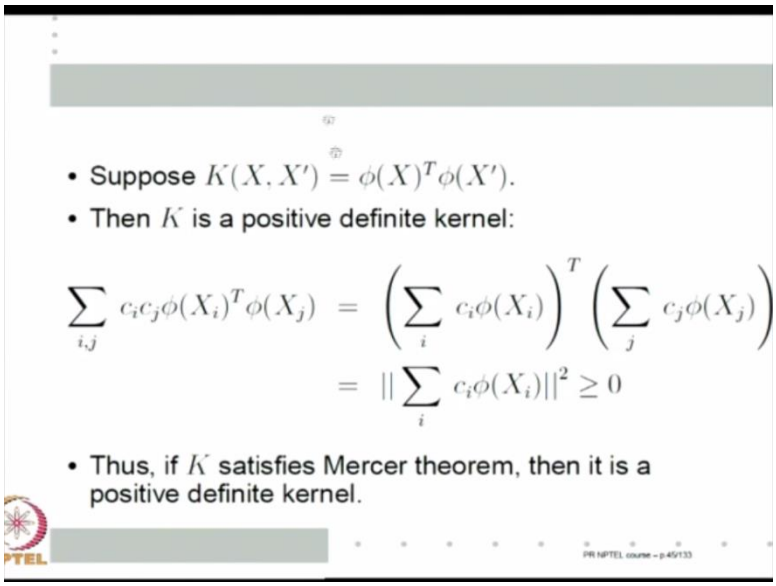
So, specifically because the quadratic to be greater than equal to 0, a positive definite kernel if K is a positive definite kernel, the, for every n and every X 1 to X n, the summation over i n j going from 1 to n C i C j K X i X j. This is the quadratic form of the matrix is greater than equal to 0 for all scalars X i. So, given any scalars even C 1 C 2 C n and any element from my feature space X 1 X 2 X n this double summation i j going from 1 to n C i C j X i X j has to be greater than or equal to 0 if K has to be a positive definite kernel. For this talk, we are going to confine ourselves to all our scalars being rear even though everything we say can be extended to these scalars coming from the complex field, we will restrict ourselves to scalars being all real numbers.

So, what does this mean? Once again this is nothing but the Gram matrix n by n Gram matrix is positive if I take n is equal to 1 of course the matrix has only one element. So, which means K X comma X should be greater than equal to 0 positive. Similarly, different if I take n is equal to 2 it will be a 2 by 2 matrix. The first row being K X 1 X 1 X K X 1 X 2; second row will be K X 2 comma X 1 K X 2 comma X 2. We want that to be positive semi definite, which for example, mean that the determinant has to be greater than or equal to 0 if n is equal to 2. And I take the determinant, determinant will be K the main diagonal is K X 1 X 1 into K X 2 X 2 of diagonal is X 1 X 2 into X 1, but K is symmetric.

So, the determinant has to be positive so, K X 1 X 2 whole square should be less than equal to K X 1 X 1 into K X 2 X 2, this is true for every pair of features X 1 X 2. So, if K is a positive definite kernel then in particular it satisfies this given any 2 feature vectors X 1 X 2 K of X 1 comma X 2 whole square is less than or equal to K of X 1 comma X 1 into K of X 2 comma X 2. I hope all of you recognize this structure of this inequality this is nothing but Cauchy Schwartz inequality, because we think of K as a inner product if K was an indeed an inner product. Obviously, phi X 1 transpose phi X 2 whole square is less than or equal to phi X 1 transpose phi X 1 to phi X 2 transpose phi X 2.

But we have not yet shown, that K is actually inner product, we just defined that a symmetric function with positive definite kernel if it satisfies this. But it satisfying this means that this kind of Cauchy Schwartz inequality is satisfied by the kernel. Of course, the Cauchy Schwartz inequality does not mean that there is a space in, in and a function phi is at K X comma X 1 comma X 2 is phi X 1 transpose phi X 2. But certainly the function K by virtue of it being a positive definite kernel satisfies the Cauchy Schwartz inequality. This is going to be very important for us alter on in the in, in studying positive definite kernels.

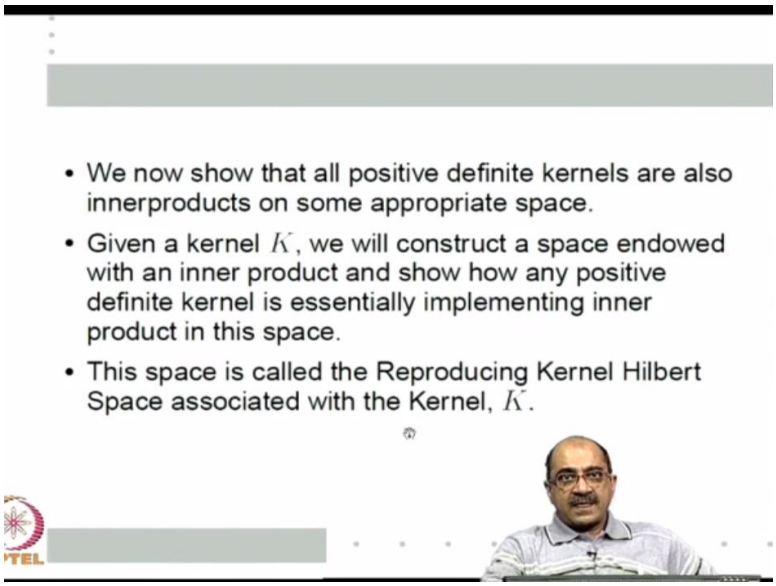
(Refer Slide Time: 19:15)



If in fact, the kernel is obtained as a inner product in some other case that is if script X can be mapped to some other space using phi and the real space of phi has an inner product here I represent it the transpose. So, in, in, in fact, if K X coma X prime is phi X transpose phi X prime then that such a K is certainly positive definite why, because for K to be positive definite kernel. I want to show that this C i C j K X X i X j should be greater than equal to 0 K X i X j is nothing but phi X i transpose phi X j.

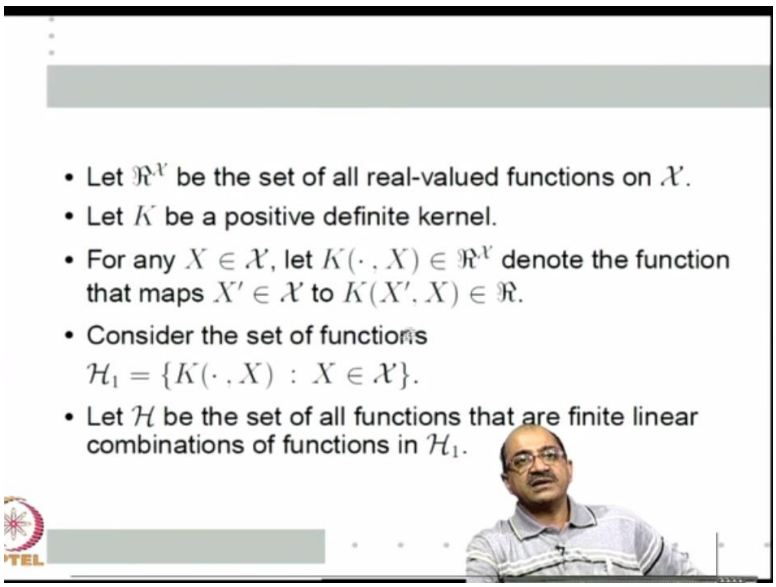
So, I need to show C i C j phi X i phi X j transpose phi X i transpose phi X j is greater than or equal to 0. So, this can this can always be written as summation over i C i phi X i transpose summation over j C j phi X j these two are same element. So, this is nothing but norm of summation over i C i phi X i norm square of that. So, this is always positive. So, if in fact, K happens to be an inner product then K will always be positive definite. But we only ask this for positive definiteness and this is of course, much easier to verify then your Mercer's theorem. So, where if K satisfies Mercer's theorem K satisfies Mercer's theorem. Then we know that there is some phi and K can be written like this. So, if K satisfies Mercer's theorem then it is a positive definite kernel.

(Refer Slide Time: 20:49)



Now, we show that all positive definite kernels are also inner products on some appropriate space as I said we show this by saying given a kernel K we will construct a space endowed with the inner product. And show how positive definite kernel is essentially implementing inner product in this space, as I said this space is call the reproducing kernel Hilbert space.

(Refer Slide Time: 21:13)



So, to start on this let R superscript script X, this one let this be set of all real valued functions on my feature space that is if I have any function g that maps my feature space

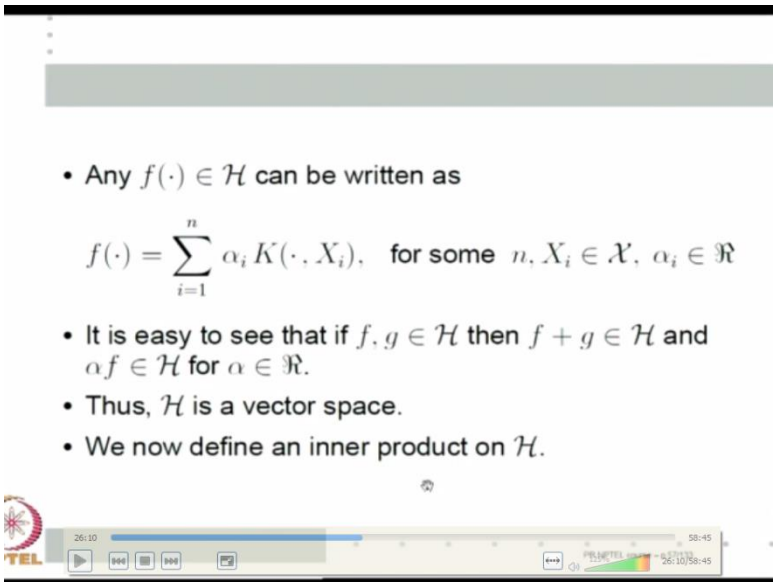
to real numbers then that g will be an element of this set. So, this set consists of all possible real valued functions on my feature space X. Now, K be the given positive definite kernel. So, given any element in my script X any feature vector X, let us represent by K dot comma X. This is general representation for functions the dot is where the functions argument is.

So, essentially this is a function because you put anything in place of this dot you get a real number. And think that you can put in place of this dot or elements of script X. So, K dot comma X is nothing but a function that maps script X to real numbers. So, this belongs to R superscript X. So, K dot X is some real valued function. Let us say this denotes the function that maps any X prime to K X prime comma X. So, that is almost so simply dot is the notation for the dummy argument of the function.

So, K dot comma X is the real valued function on X whose value at an argument X prime is K of X prime comma X. Now, consider the set of functions H 1 which consist of all such K dot X‘s for R X for every X in my feature space actually we can think of K dot X at the kernel functions centered at X. It is actually this norm is actually function its value at any argument value is the value of the kernel function X prime comma X so, this we will call this a kernel centered X.

So, H 1 is the set of all the set of kernels centered at all possible feature vectors all possible elements of X. Let us script H be the set of function that are finite linear combinations of functions in H, we will you take H 1 and make finite linear combinations. Let us say alpha 1 K dot X 1 alpha 2 K dot X 2 like that so all possible finite linear combinations of functions H 1. If you take all possible finite linear combinations of functions of H 1, let us call that set of functions as script H. So, script H is the set of all functions that are finite linear combinations of functions in H 1.

(Refer Slide Time: 23:53)



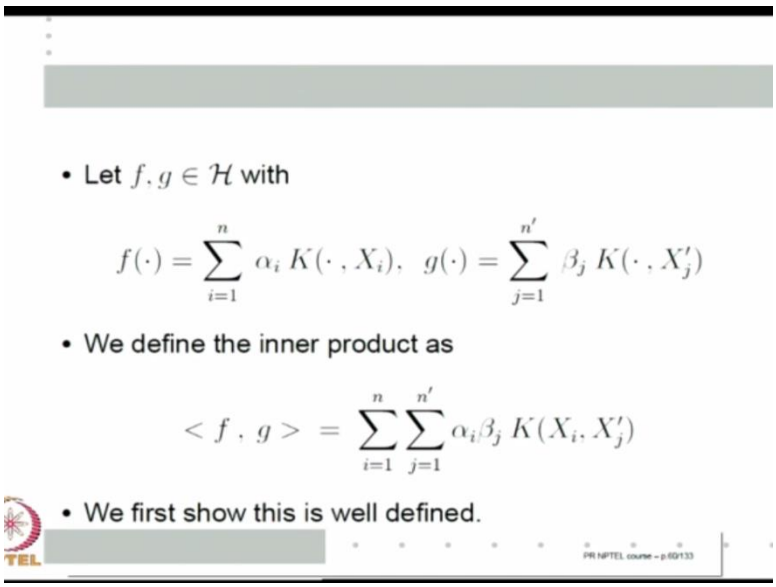
So, any f in H would be a linear combinations of functions from H 1 and the only functions in H 1 are K dot X type where X is an element of script X. So, any f in my script H can be written as alpha i K dot X i for some n. So, it is a finite linear combination so, we do not know how many term they will be, but there will be some in (( )) such that there will be n terms in the linear combination.

So, f will be i is equal to 1 to n for some n of alpha i K dot X i for some real numbers alpha i and some X i. So, every function in this script H can be written like this. For some X 1 to X n elements of script X some alpha 1 to alpha n real numbers. And some n f can be written as i is equal to 1 to f in alpha i K X which means if I have 2 functions f and g in H is very easy to say f plus g is also in h. Because f plus g would once again be some finite linear combination of K dot X‘s. And similarly, alpha times f will also be in H for any real number alpha which means H is a vector space over the field of real s, because the rest of the vector space axioms are easy to verify, essentially we have a vector addition and scalar multiplication while define.

So, this, this set H know is a vector, vector space under the usual addition of functions and scalar multiplication of functions a vector space over the field of real numbers. As I said everything as a in this lecture can be extended to the field of complex numbers with minor modifications, but for simplicity we will stick to all scalars being layer. So, what we showed is that if we take finite linear combinations of kernels entire that any points in

X, if we call that as the space H that H is the is a vector space under the usual function addition and scalar multiplication. So, now what we are going to do is that we define an inner product on a space H inner product. And then show that essentially there can be a phi that maps script X to this H. So, that K X comma X prime will be phi X phi X prime inner product that we are going to define.

(Refer Slide Time: 26:21)



So, first I have to define an inner product for any two elements of H. So, let us take any 2 elements of H and g, every element of H is a finite linear combination of kernel functions entered at X i. So, f is some summation i is equal to 1 to n alpha K dot X i alpha i K dot X i for some real alpha i some elements X i. So, g is beta j K dot X j prime so, X 1 X 2 X n X 1 prime X 2 prime X n prime. These are all arbitrary I took n and n prime also different, because the 2 functions may have different number of terms in there linear when they represented as linear combinations of K's.

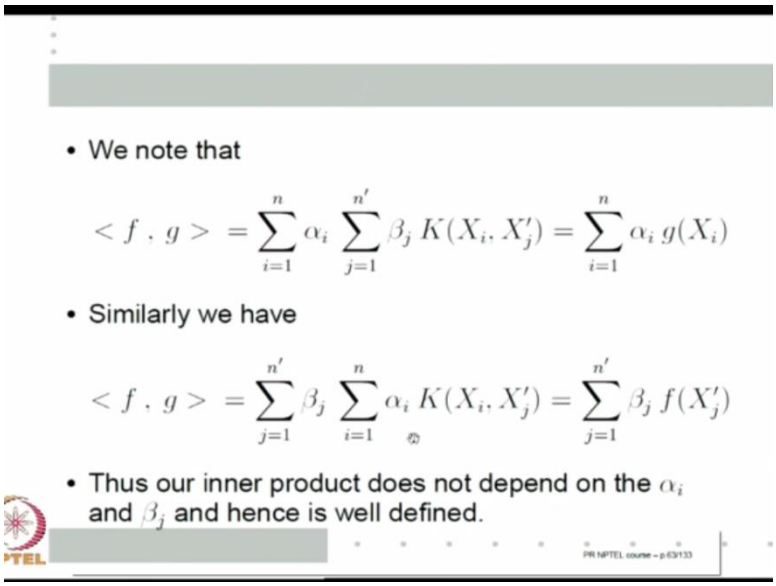
So, given 2 functions f and j like this, we will define inner product I am going to represent inner product like this in general the standard notation for inner product between the 2 angular brackets f comma g within the angular bracket. So, this is the inner product of f and g if f is this and g is this. We will represent the inner product of f comma g as i is equal to 1 to n j is equal to 1 to n prime summation alpha i beta j K of X i comma X j prime.

So, if f is this and g is this then the inner product is defined to be this. Of course, we have to show that this is an inner product it has satisfy all the properties of inner product before we can go for there to show that this is an inner product. We have to first show that this is well defined, what do you mean by well defined when I say elements of H R finite linear combinations of elements of H 1 which are nothing but kernel functions centered at point x.

So, this function f given different arguments say f of X is alpha i K i X comma X i. Now, a given a function a of H there may be more than one way of writing it as a linear combination of some kernel function when we say we all possible linear combinations there is no guarantee that 2 different linear combinations do not give you the same function which on other words a given function may not be representable uniquely as a linear combination which means these alpha i's are not (( )).

The function of course, is (( )) because function is an element in H, but the same function may be representable with some alpha i primes of and some K y i's who knows. So, essentially what we have to show is that it is not dependent on this alpha i's and beta j's, because they can be different kinds of representations. So, the, the inner product should not be dependent on the expansion coefficients. So, first you have to show that it does not depend on the expansion coefficients.

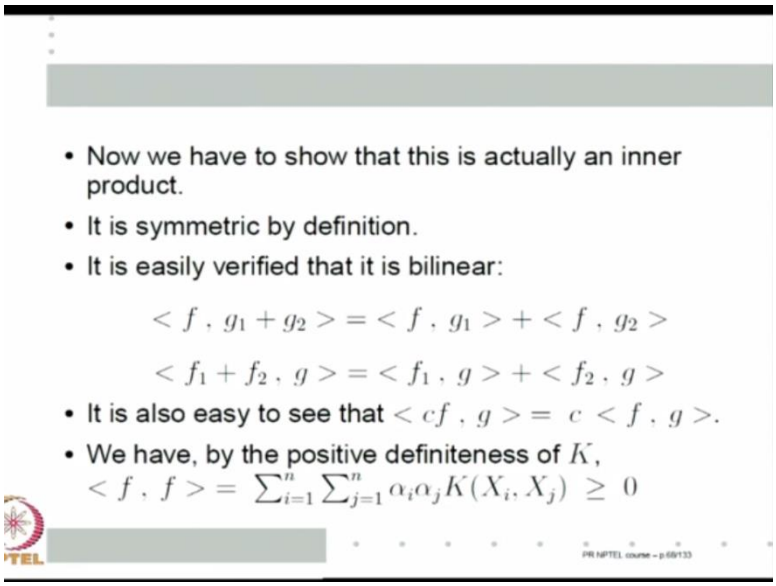
(Refer Slide Time: 29:21)



The other product as we defined I can take alpha i outside, what I have is j is equal to 1 to n prime beta j K X i X j prime g dot is j is equal to 1 to n prime beta is a K dot X j prime. So, g of X i will be beta j X i X j prime. So, if I take alpha i out, what I have is beta j K X i X j prime which is nothing but g f X i. So, I can write the inner product as i is equal to 1 to n alpha i g of X I, which means while it depends on the function g it does not depend on either beta j or X j prime i g can be written in different ways. It does not matter as long is the same function it only depends on the values of g at X i. So, i g has a different expansion it does not matter. Similarly, by taking beta j by interchanging the 2 summations and taking beta j out I can write as j is equal to 1 to n prime beta j i is equal to 1 to n alpha i K i X i X j prime, what is this? f dot is this.

So, f of X j prime will be i is equal to 1 to n alpha i K i of X j prime X i and your K is symmetric. So, whichever way I write it as does not matter. So, this entire summation nothing but alpha of X j i mean f of X j prime. So, I can also write f the inner product between f and g as summation j is equal to 1 to n b prime beta i j f of X j prime. So, it depends on the value of the function f at X j prime where does not does not specifically depend on the specific vectors f i or alpha I, which shows that the inner product does not depend on the specific expansion coefficients and hence it is well defined. So, this f well defined inner product. So, next you have to ask is this indeed an inner product I just said is an inner product, we have to show that this is an inner product.

(Refer Slide Time: 30:19)



So, what is an inner product satisfies? An inner product vector space h takes space of vectors and gives you a real number. So, it is a function h class h 2 l, what does this function satisfy? It has to be symmetric, it has to be bilinear inner product of any, any X any element with itself is always greater than or equal to 0. An inner product is 0 if and only if the element is 0 inner product of some element with itself is 0 if and only if element is 0 these are the properties of inner product.

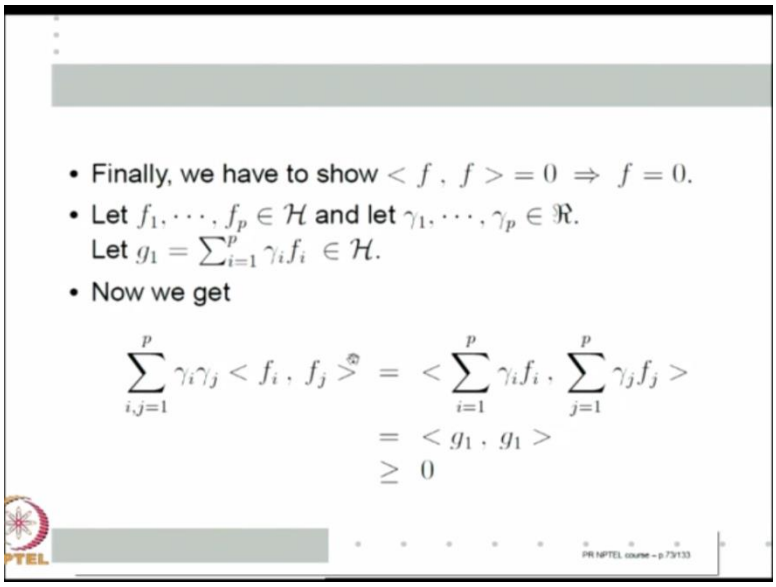
Now, our inner product is symmetric by definition, because this is how we define this. So, you given this and this there is an dependant on you know which is f and which is, because these alpha i beta j K i X i X j prime and K is symmetric. So, in a product of f and g is same as product of g and f i. I would like to remind you once again that will considering field of real numbers, this our vector space is vector space over reals that is why we are only looking at symmetry otherwise we have to worry about complex conjugateness. But we are not allowing complex scalars now for simplicity of exposition I am just taking everything to be real.

Similarly, it has to be bilinear that is inner product of f g 1 plus g 2 should be inner product of f comma g 1 plus f comma g 2. Similarly, the other way which is also very straight forward, because of the definition of the inner product it is very easy to verify that it is bilinear. So, if I put f 1 plus f f 2 here. So, I will get some alpha i K dot X i plus

some gamma gamma j K K dot some X bar j so, accordingly they come in the summation.

So, f 1 plus f 2 comma g will be f 1 comma g plus f 2 comma g and so on. So, by definition, it is symmetric and it is bilinear. And similarly, you can show that if one of the, are given multiply by a constant, the inner product gets multiplied by that constant. Next, what you have to show? You have to show that inner product of f comma f is greater than or equal to 0 while end product of f comma f if f is alpha i K dot X i inner product of f comma f is alpha i alpha j K X i comma X j sum over i i j. So, this is nothing but the quadratic form of the gram matrix and because K is positive definite kernel, this is greater than or equal to 0. So, we also shown that inner product of f comma f is greater than or equal to 0.

(Refer Slide Time: 34:17)



So, the only thing that remains to be shown now is that if the inner product is 0 f is 0. Mind you, if I had assumed, if I defined positive definite kernels as a function K such that the Gram matrix is positive definite rather than positive semi definite then I am done. If I am asking for this to be positive definite then except when R alpha i alpha j is 0 which means function is 0 this quadratic form has to be strictly greater than 0. But if I want positive definite functions only there will be difficult for, for us to get kernels. Because we want kernels should represent similarity in general as it turns out in many

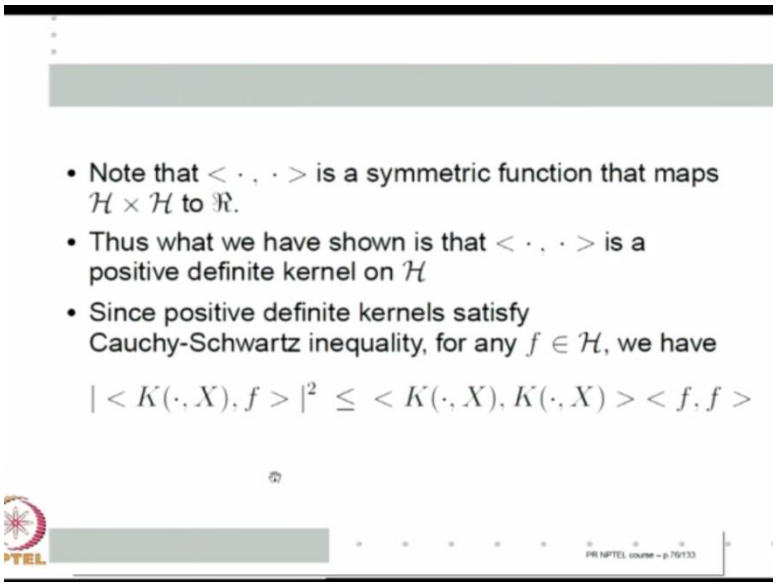
applications. While it is easy enough to show Gram, Gram matrix is positive semi definite is much more difficult to show that Gram matrix is positive definite.

So, we do not want to assume positive definiteness, the kernel, because we do not know need it I, I just want you to understand that because we assume only positive semi definiteness of the Gram matrix I only get f comma f inner product of f and f is greater than equal to 0 and it is not I I still have to separately show that if inner product is 0 f is equal to 0. So, let us go a go ahead and choose this separately. Show this, let us take any, any some P functions from H and some P scalars gamma 1 to gamma P and let g 1 is gamma f i, because H is the vector space g will also be in the in H .

Now, if I look at gamma i gamma j f i f j gamma i gamma j inner product of f i f j summed over i j is equal to 1 to n. By the bilinearity of inner product, it is same as inner product of i is equal to 1 to P gamma f i gamma is j is equal to 1 to P gamma j f j. Because of the bilinearity of the inner product, we have seen that the inner product of f 1 plus f 2 comma g is inner product of f 1 comma g plus f 2 comma g. And similarly, the other way and hence this summation double summation can be written as inner product between these two.

Now, this is nothing but what we call the function g 1. So, this is nothing but the inner product between g 1 and g 1, because these two are the same summation i and g are dummy variables after all. And this we already show n to be greater than equal to 0. So, what it means is given any P functions from H f 1 f 2 f P and any P scalars gamma 1 gamma P gamma i gamma j inner product f i comma f j summed over i j equal to 1 to P is greater than or equal to 0. What is this mean? This is exactly what we wanted from a positive definite kernel, earlier positive definite kernel meant as we seen C i is C is a K X i comma X j is greater than or equal to 0. So, I can think of the inner product itself as kernel on H our original K the the kernel I am considering K is a kernel on script X or feature space, but now it looks like f i f j can.

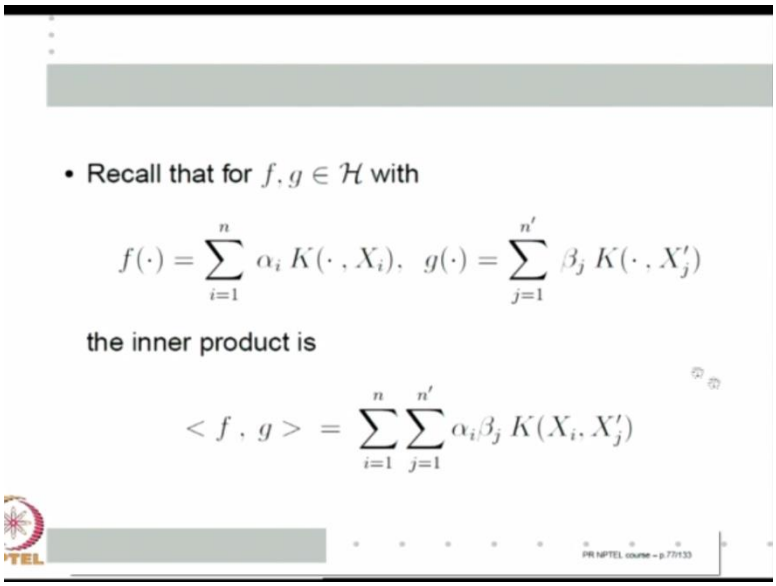
(Refer Slide Time: 37:29)



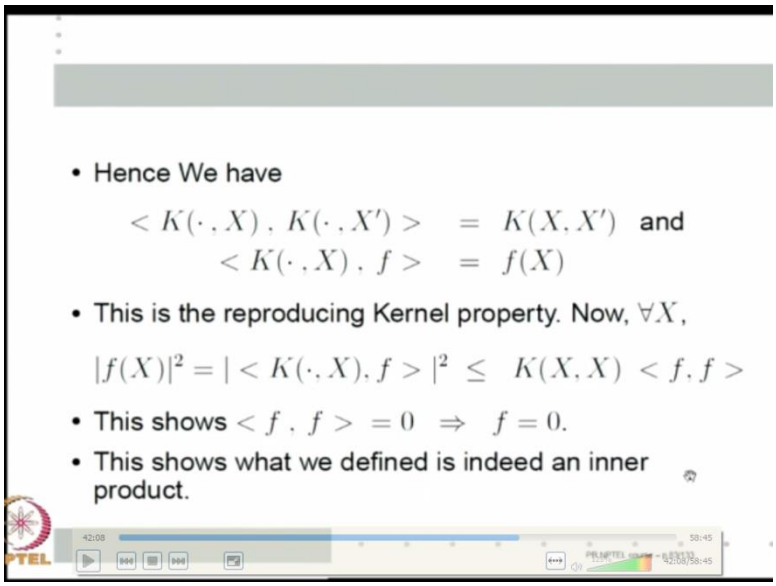
What is f i f j f i f j? This inner product is a symmetric function that maps H cross H, the inner product is given any 2 elements of H, it maps it to real number and the function is symmetric and any such symmetric function by virtue that it satisfies this mean that it is a kernel on H. So, that is what we have shown is that this function; this function linear product function which maps H cross H to R is in fact, a positive definite kernel on H. Because it is a positive definite kernel on H, we know that positive definite kernel satisfy Cauchy Schwartz inequality. What does, what did the Cauchy Schwartz inequality mean? The kernel value at X 1 comma X 2 whole square is less than or equal to kernel valued function X 1 comma X 1 into kernel value at X 2 comma X 2.

So, if I take any 2 element from H their inner product square. So, this is element 1; this is element square is less than or equal to inner product of element 1 comma element 1 into inner product of element 2. So, in particular I know that if I choose f as 1 element and K dot X as another then because of the because this satisfies Cauchy Schwartz inequality. We know have inner product of K dot X comma f whole square is less than or equal to inner product of K dot X comma K dot X and inner product of f comma f [s]. So, let us calculate each of these inner products from our definition of inner product.

(Refer Slide Time: 39:04)



(Refer Slide Time: 39:12)



Recall that f is equal to alpha i K dot X i g is equal to beta j K dot X j then inner product of f comma j is this. So, what it means? For example, if I want K dot X comma K dot X prime then both of these sums are only one element sums both these expenses are one element expansion. So, my f comma g will be nothing but K f X comma X prime so, K dot X comma K dot X prime is K X comma X prime.

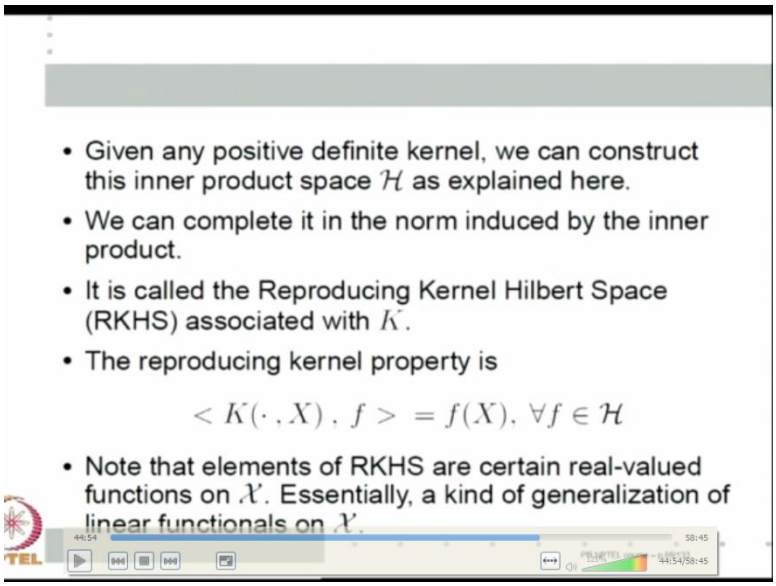
Similarly, if f is this, but g is simply K X prime 1 element then what will be the inner product? Alpha i K X prime comma X i. So, if g is simply K X prime then the inner

product will be summation alpha i K X prime comma X i which is nothing but f of X prime I hope this is clear, K dot X comma f is f of X. So, very, very important property, just for our definition of inner product shows this let us understand this again. Let us say f is this alpha i K dot X i sum over i. And let us say g is simply a single function K dot X then what will be this double summation be? This implies single summation i is equal to 1 to n alpha i K X comma X i.

Now, summation alpha i K X comma X i is nothing but f of X. So, definition of inner product is such that K dot X comma f is f of X. Now this is called the reproducing property of kernel. If I take a kernel centered at X and take its inner product with f what I get is the value of f at X. If I take a kernel centered at X and take it is centered product with respect to f then what I get is the value of f at that X this is called the reproducing kernel property, because the reproducing property what we have is the following.

Now, f X square which is nothing but the inner product of K X comma f whole square by Cauchy Schwartz inequality. This is inner product of K dot X comma K dot X which is nothing but K X X into inner product of f comma f. So, what is that mean? It means if f comma inner product of f comma f is 0 then for every X f X whole square is 0 that means for every X f X is equal to 0 that means f is equal to 0. So, we have shown that our inner product is such that if inner product of f comma f is equal to 0 then f is identically equal to 0 using this so called reproducing kernel property. So, this shows that what we have defined indeed a proper inner product.

(Refer Slide Time: 42:11)



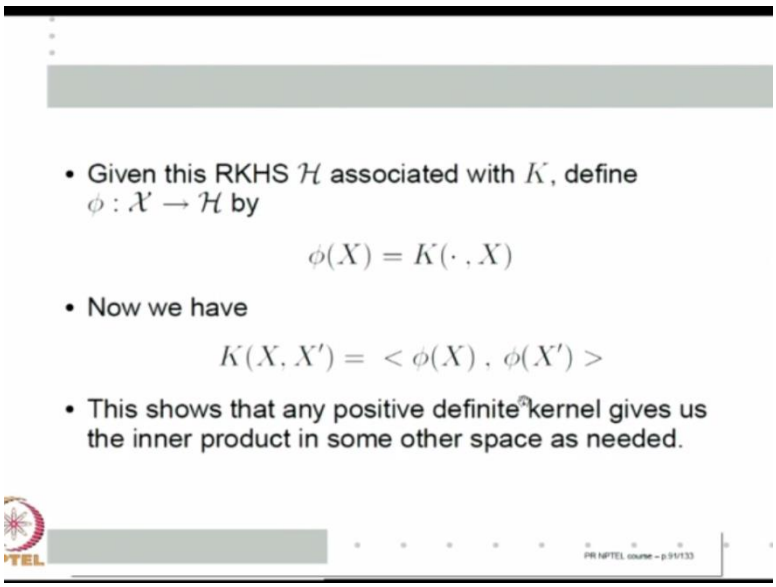
So, given a any positive definite kernel, we can construct the inner product space H as explained give me a f kernel K and have the feature space X. I will take kernel centered at each point on X which we called K dot X, we take the set of K dot X for every i for X in my feature space. Then I make all possible finite linear combinations of these, these kernels alpha i K dot X i summed over i. And I can consider the set of all functions that is my space h, this space happens to be a vector space on which I can put an inner product. And you know once I put this inner product technically I want this space to be complete, what does what do I mean by complete? That a an inner product; obviously, gives me metric.

If I have an inner product between 2 elements f comma g then I can have a norm for this as inner product norm f square is inner product f comma f. And I will define distance between f and g as norm of f minus g. Under such a distance metric if any sequence is Cauchy that is a a sequence of that as n n n m tends to infinity. The nth and mth element of the sequence comes close to each other in the distance metric. Then the sequence should also have a limit that is what is meant by complete. This space is not complete, we can always complete, it this just a technical diognisation if you do not understand it does not matter essentially, H is a vector space is the inner product. And we simply assume that under this inner product all convergent sequences have their limits in the space that is what is called completing the H.

This space is called a rapid using Kernel Hilbert space. Essentially Hilbert space is a vector space on which you are define a inner product and in the metric induced by the inner product this space is complete. So, that why it is called a Hilbert space, so this H, we that, we constructed is called a reproducing kernel Hilbert space and the reproducing kernel properties. So, given a positive definite kernel K we constructed the H and that H there is an inner inner product and the inner product is such that inner product of K dot X comma f will give me the value of that f and that X.

This is called the reproducing property of the kernel, this is true for every f in the space. Essentially the elements of this RKHS are real valued functions on H not all real valued functions of certain real valued functions at f X, which can essentially be written as linear combinations of kernel centered at X. So, this is a kind of generalization of linear functional on X. So, that is another way of looking at RKHS, which have essentially the special reproducing kernel property.

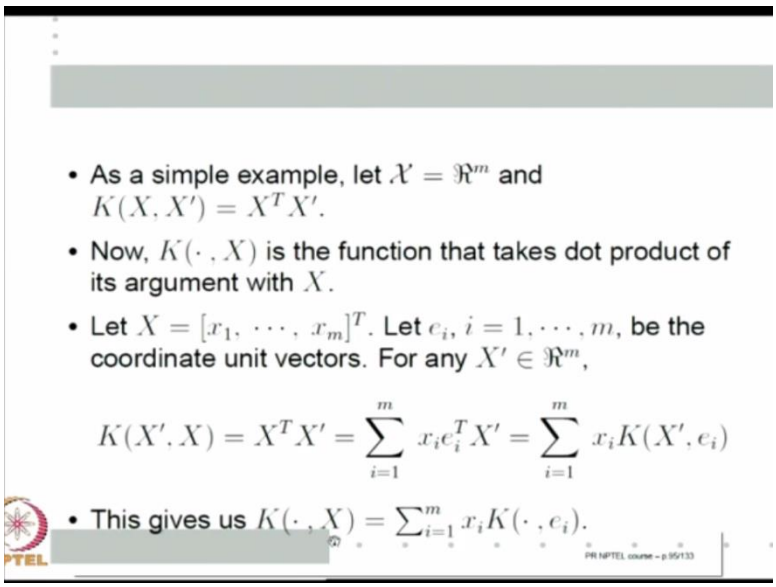
(Refer Slide Time: 45:14)



So, given this RKHS K be associated with K, now we can define f i that maps X to H that is it maps every element of X to every element of H namely i map the element X in my feature space to the element K dot X in h i mapped X to the kernel functions centered at X. So, phi maps X to K dot X, if I take this phi then the inner product of phi X comma phi X prime is nothing but inner product of K dot X comma K dot X prime which is nothing but K X comma X prime.

So, K X comma X prime is nothing but inner product between phi X and phi X prime in H. So, H is the space you are looking for give a given a positive definite kernel K. Now, I have constructed a specific space H and I showed you a function phi that maps X to H such that K X comma X prime is nothing but inner product between phi X and phi X prime. So, this is; this means that any positive definite kernel gives us an inner product in some other space as needed. Now, let us just get a little more idea of what this RKHS is a very familiar setting if you look at it then we make at some more idea about this RKHS is.

(Refer Slide Time: 46:41)



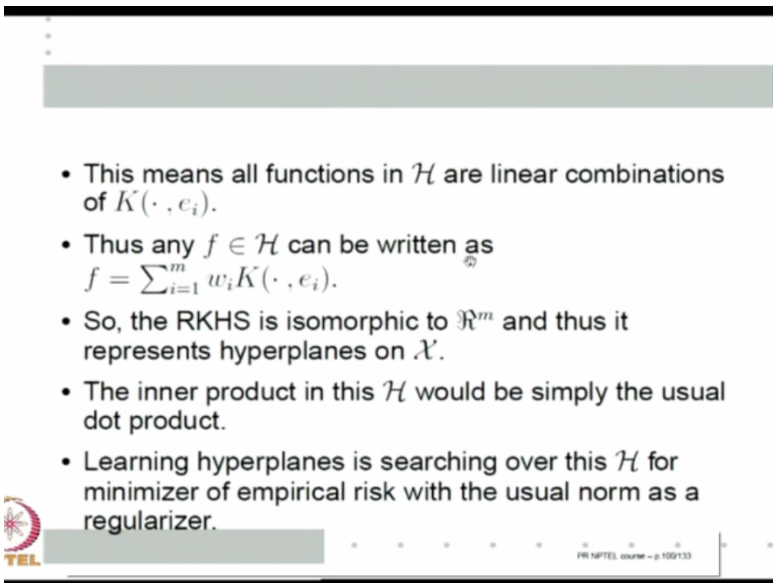
So, we look at a very simple example, let us assume X is R m, and let us take this simplest linear kernel K X comma X prime is transpose of X prime. So, what will be K dot X now K dot X is a function that takes dot product of its argument with X for each X I have function K dot X which is essentially taking inner dot product with X that is the name of the function.

So, you give X prime as the argument of the function outcomes X prime transpose X. So, essentially K dot X is the function that takes dot product of its argument with X. So, let us take any, any vector X in R m with components X 1 to X m. And let us say e i are the coordinate unit vectors that is e 1 is 1 0 0 0, e 2 is 0 1 0 0 1 and so on. If you give me any X prime belonging to R m. Now, K X prime comma X is by definition X transpose X

prime X transpose X prime is nothing but summation of over i of ith element of X and i-the element of X prime.

So, i summation over i I can write X i and I can get ith element of X prime as e i transpose X prime, because e i is the coordinate vector. Now, by definition of kernel e i transpose X prime is nothing but K of X prime comma e i or e i comma X prime does not matter whichever way we write. So, if summation X i K of X x prime transpose g i, so K X prime comma X is nothing but summation i is equal to 1 to m X i K X prime comma e i, what is that mean? K dot comma X is i is equal to 1 to m X i K dot comma e i. So, for any arbitrary X if you want a kernel function centered at X that itself can be written as a linear combination of these m specific kernel functions K dot e i. That means every K dot X can be written as a linear combination of K dot e i which means any linear combinations of K dot X for all X for different X's can once again be rewritten as a linear combination of K dot e I.

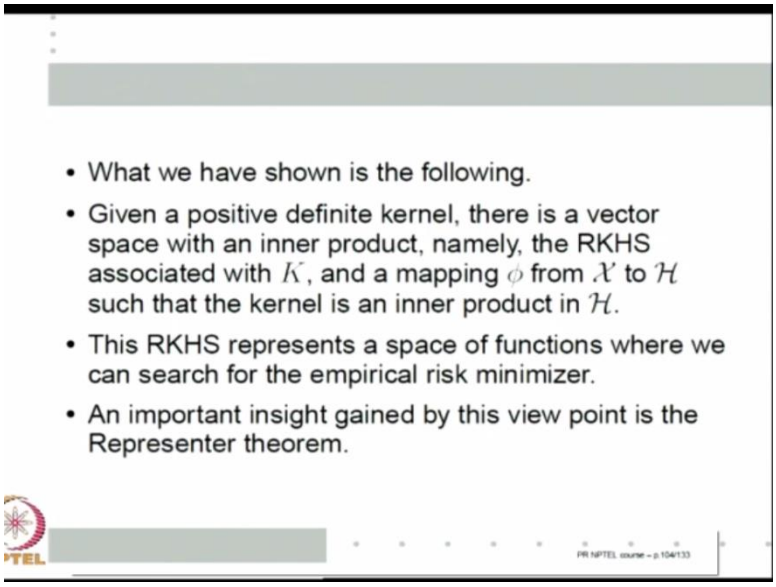
(Refer Slide Time: 48:40)



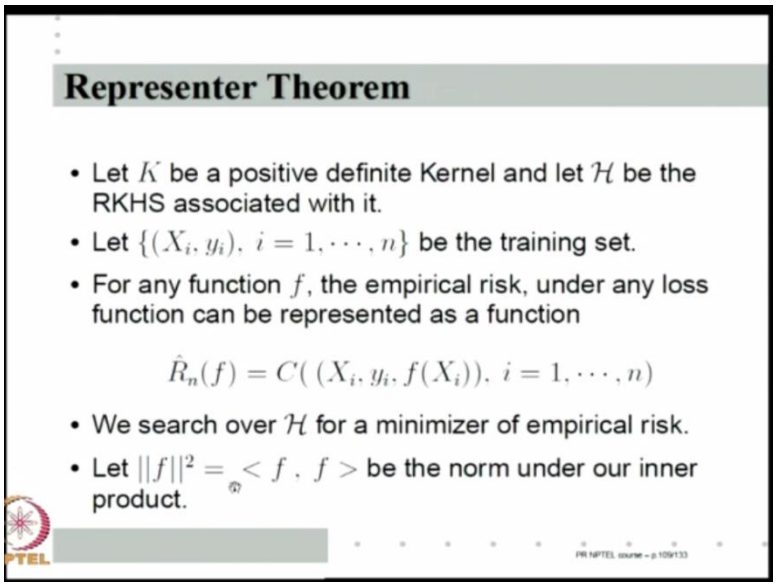
Which means all functions in H are simply linear combinations of K dot e i. And for i is equal to 1 to m there only m such kernel functions. And every other function H can be written as linear combination of these, which means any f can be written as i is equal to 1 to m w i K dot e i. So, any f can be represented now by m w i's which means it can be represented by a vector in R m, which means every f can be uniquely defined by a m component vector which means my RKHS itself is isomorphic to R m. And that means

every element in H can be associated with a hyperplane on H. So, my for my linear kernel the reproducing kernel Hilbert space is nothing but the set of all the hyperplanes on X. That is what a linear classifier gives me essentially if my if I take the linear kernel extra X transpose X then what I am doing is I am searching over hyperplane over x. So, I have actually searching over the RKHS so, the inner product H now is simply the usual dot product in R m and learning hyperplanes is nothing but searching over this H for a minimize of empirical risk.

(Refer Slide Time: 50:02)



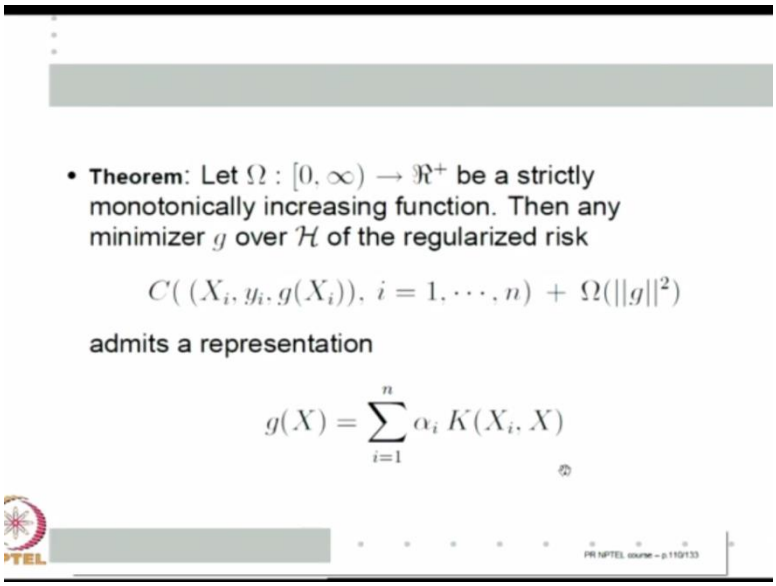
(Refer Slide Time: 50:32)



So, we shown the following given a positive definite kernel there is a vector space within a inner product namely the RKHS associated with K and a mapping phi from X to H. Such that the kernel is an inner product in H the RKHS represents a space of functions where you can search for the empirical risk minimize is a is is is essentially that that kind of functions. Now, given this, we will just do one very important insight which is called the Representer theorem. What is Representer theorem say? Let K be a positive definite kernel and H be the associated RKHS and X i y i be the training data set as earlier. Now, any given any function f suppose, I want empirical risk under this training data set.

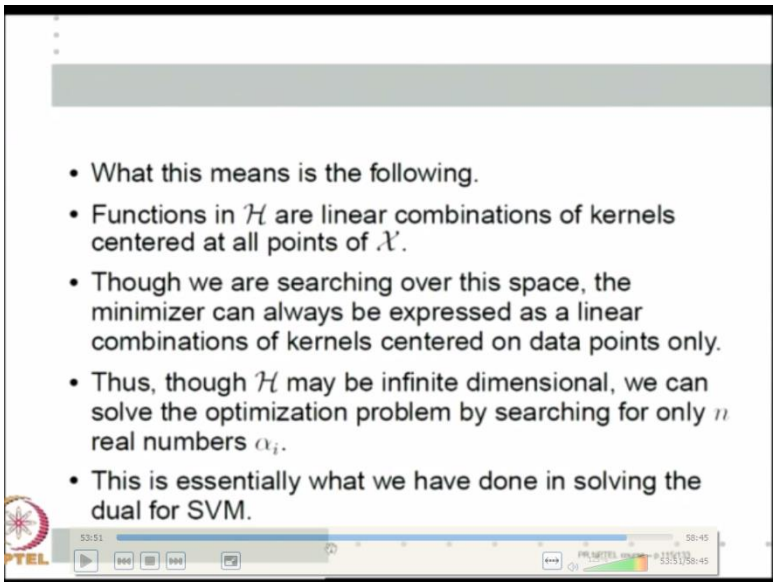
So, empirical risk depends on some loss function l of y i comma f X i whatever so, no matter what is the loss function is, the empirical risk ultimately depends on what X i y i f X i for i is equal to 1 to n. It cannot depend on anything else, given this training examples and a function f. The empirical risk can depend only on X i y i f X i for i is equal to 1 to n. So, let us write empirical risk of any function f as some function C of X i y i f xi i is equal to 1 to n. So, we do not have to worry about what our loss function is, let us say we want to minimize H i empirical risk. But we want to do a regularized empirical risk minimization that means we use a regularization term for which we use the norm in the H space. And let us say the inner product f comma f, we will represent as norm f square like this.

(Refer Slide Time: 51:38)



So, this is the theorem, let omega be any strictly monotonically increasing function then if g is any minimizer minimize is I am searching over the RKHS. And g is the minimizer of the regularized risk there regularized risk is this empirical risk under any arbitrary loss function plus the regularization term omega of which is a function of the norm of g in H. So, essentially the regularization term should be some increasing function of the norm of g. So, if, if I am using norm of g in H are the regularizing under not, not directly norm of g any increasing function of norm of g would do. Then any minimizer can be represented as the linear combination of kernels centered at X m. So, g dot is alpha K X i dot this. So, the the final minimizing function can be simply written as linear combination of kernel functions centered at data points X i.

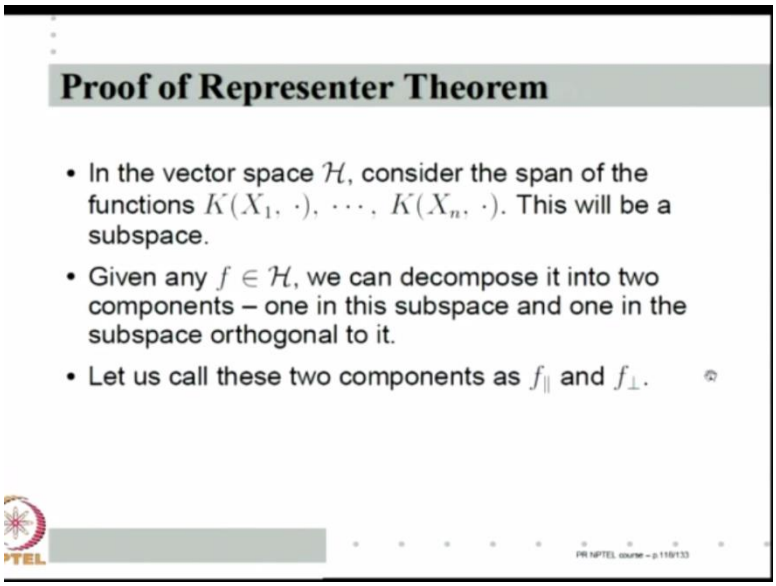
(Refer Slide Time: 52:45)



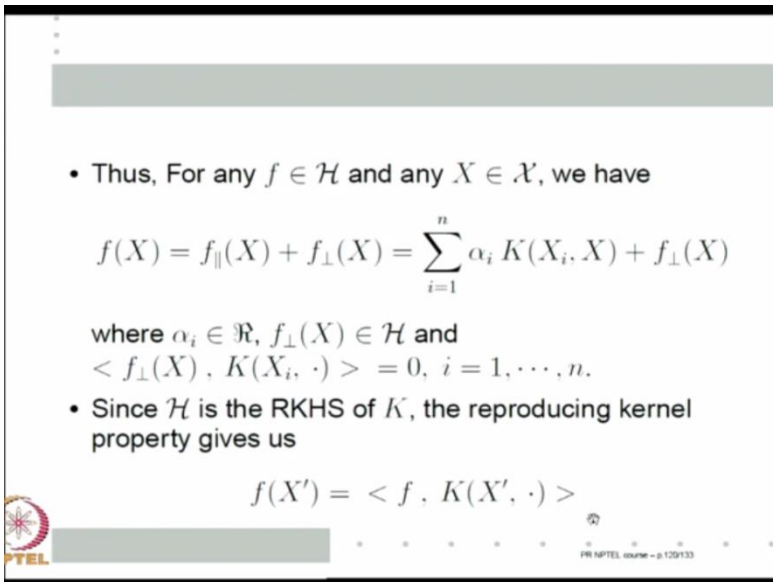
Let us see, what it means, functions in H are linear combination of kernel centered at all points of X there be, there may be uncountably infinitely many points in X. So that many functions are there in H though, we are searching over the space, the minimizer can always be expressed as a linear combination of kernel centered at data points only. And they are only finitely many data points, which means though H may be infinite dimensional, we can solve the optimization problem by searching for only n real numbers alpha i.

I do not have to worry about what the g's representation is my minimzer g always representation like this where X i's are always all given. So, my g dot is nothing but alpha i K X i dot. So, essentially I need only alpha 1 to alpha n to find my g, my minimizer. Even though H, H contains linear combination kernel centered all points of X. And H may be alpha 1 to alpha n 2, find my g my minimize even though H, H contains linear combination kernel centered all points of X. And H may be infinite dimensional I can find the minimizer by searching for only these n numbers alpha i. This is essentially what we did when we solve the dual in SVM irrespective of the, as I said are the dimensions of the transform of the feature vector. The dual is a dimensionality equal to the number of examples, because this is a very generic property when we work with kernel. If you are doing a search over the RKHS for a minimize of the regularized risk under any loss function as long as regularizer uses the norm of the function in H.

(Refer Slide Time: 54:19)



(Refer Slide Time: 54:47)

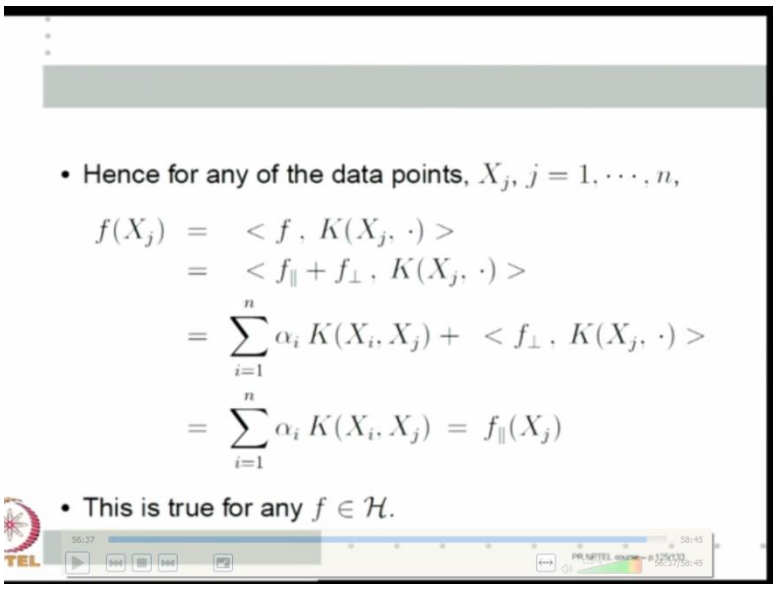


So, let us quickly prove the Representer theorem in the vector space of H, consider this span of the functions K X 1 dot X i. So take the kernel centered at each of the n data points and take their linear span that is a sub space so, because these are subspace given any f in my RKHS that f can be resolved into that component which is in the subspace. And the component that is a orthogonal to it, let us call these 2 components f parallel and f perpendicular which means for any f in H and any H in X. Because f is f can be written as f parallel plus f perpendicular f X is f parallel X plus f perpendicular X f parallel X is

what is that part of f which is in the linear span of X i that means f parallel X can be written as a linear combination of K X i dot.

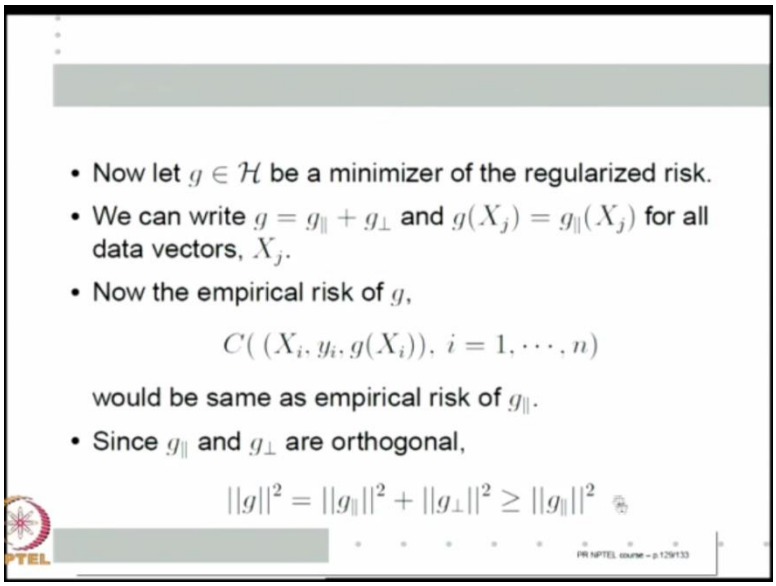
So, f parallel X is nothing but i is equal to 1 to n alpha i K i X for some alpha i's and f perpendicular X what can I say for f perpendicular X. Firstly it is also in H and secondly it is orthogonal to this, that means its inner product with this with f parallel should be 0. And by bilinearity its inner product with each of the K X i dot should be 0 for if X i's are the data points. Then f perpendicular is a function such that f perpendicular comma K X i dot is equal to 0 for phi is equal to 1 to n, this is true for every function. Given my data points X 1 X x n I can resolve every function as the parallel and perpendicular components of this K of the linear span of K X i dot. And hence this kind of decomposition holds. Since H is RKHS reproducing kernel property gives me f X prime is f comma X prime dot, the inner product of f and K X prime dot is is the f X prime.

(Refer Slide Time: 56:02)



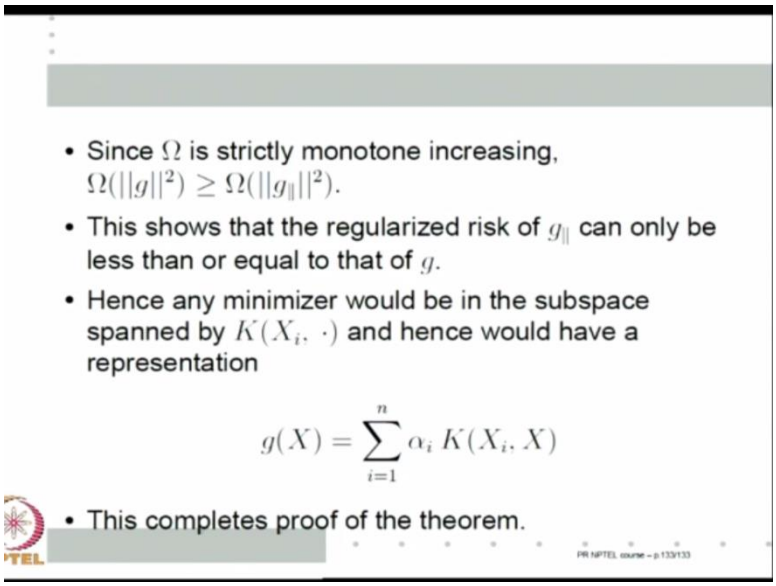
Which means if I take any data points X j f X j is the inner product of f and K X j dot, f is f parallel plus f perpendicular K X j dot. By bilinearity this is f parallel inner product f X j the K X j and f perpendicular K X j f parallel K X j inner product can be written like this f perpendicular K X j is 0, which means for each of the data points X i j is same as X parallel X j this is true for every function f in H.

(Refer Slide Time: 56:39)



Now, let g be any minimizers of the regularized risk we can write g as g parallel plus g perpendicular. And hence we know g X j is g parallel X j for all data vectors X j. Now, the empirical risk only depends on X i y i g X i. So, it depends on values of g's only on the data points on the data points g and g parallel are same. So, the empirical risk of g is same as empirical risk of g parallel. So, we know that empirical risk of g and g parallel are same. Now, let us look at the regularizing term. So, we know norm g square is norm g parallel square plus g perpendicular square these two are orthogonal and so hence is greater than g parallel square.

(Refer Slide Time: 57:22)



So, which means omega, the omega g square g norm square is greater than omega g parallel square, which means g parallel cannot have a regularized risk anymore than that of g that means g parallel is also minimizer of risk. And hence my minimizer always admits this representation. So, apart from the proof this particular theorem is very important, it shows that essentially my minimizer if I am doing empirical risk over empirical, risk minimzier any loss function over the RKHS using the norm of a the function under H as the regularizer. Then it has this nice support vector expansion, this is a very generic property of all kind of kernels as long as kernels are pointed out, that is the reason why one looks at positive definite kernels. So, we will stop about the kernel based methods here. So, next class, we will, we will just round up about what all we have done about learning non linear classifiers. And then move on to a few special topics to wind up the course.

Thank you.