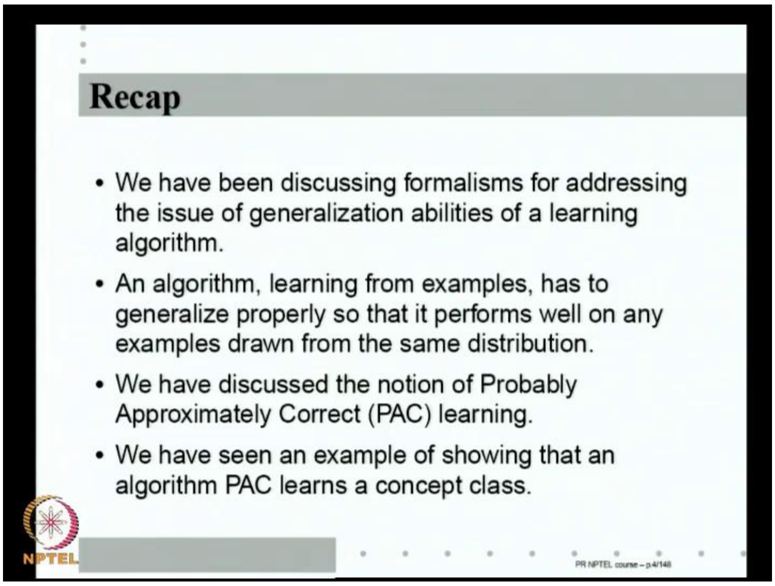
Pattern Recognition
Prof. P. S. Sastry
Department of Electronics and Communication Engineering
Indian Institute of Science, Bangalore

Lecture - 21
Overview of Statistical Learning Theory;
Empirical Risk Minimization

Hello and welcome to this next lecture's in the pattern recognition course. We have been looking at some basics of statistical learning theory. We have looked at some aspects last time. So, as we said for the next couple of lectures, we will be looking at this. So, just briefly recall what we have done last class.

(Refer Slide Time: 00:40)



Recap

- We have been discussing formalisms for addressing the issue of generalization abilities of a learning algorithm.
- An algorithm, learning from examples, has to generalize properly so that it performs well on any examples drawn from the same distribution.
- We have discussed the notion of Probably Approximately Correct (PAC) learning.
- We have seen an example of showing that an algorithm PAC learns a concept class.

PR NPTEL course – p.4/148

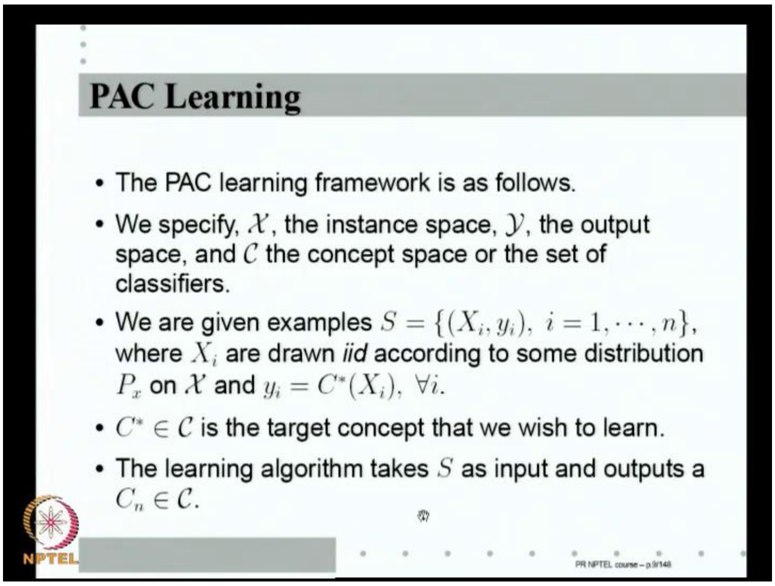
We have, we are actually looking at some formalisms for addressing the issue of generalization. How can you theoretically talk about correctness of the generalizations made by learning algorithm, right? You see in any learning algorithm essentially given some examples, and it learns from the examples. A general rule for classification or regression or a general function of a regression and the generalization issue is that the algorithm should properly generalize. So, that it performs well on new examples which are drawn from the same distribution as the training samples.

So, given some training samples if I am learning a classifier, then, similar examples, new examples given to me I should be able to correctly classify that I the issue of whether I am generalizing properly or not? Last class we looked at the notion of what is called

Probably Approximately Correct learning. It is called PAC learning. And, we have presented the entire formalism. We also looked at an example of how we show that a specific algorithm, PAC learns a concept class. So, we take a very simple example of AR 2 classification problem, what I called learning the concept of medium built persons.

Essentially the target concept is an access parallel rectangle. We have seen example of trying to learn with two bisecting over two different class of classifiers. One is over the set of all possible axis parallel rectangles on R 2 and we showed that this algorithm PAC learn. The algorithm simply looks for any classified in its bag that is consistent with all the examples. If there is more than one you take the smallest. This algorithm, if we are searching over all possible axis parallel rectangles we showed that it PAC learns. Whereas, if we change it to searching over every possible classifier two class classifier over r two then it does not PAC learn, okay?

(Refer Slide Time: 02:42)



PAC Learning

- The PAC learning framework is as follows.
- We specify, $\mathcal{X}$, the instance space, $\mathcal{Y}$, the output space, and $\mathcal{C}$ the concept space or the set of classifiers.
- We are given examples $S = \{(X_i, y_i),\ i = 1, \cdots, n\}$, where X_i are drawn *iid* according to some distribution P_x on $\mathcal{X}$ and $y_i = C^*(X_i),\ \forall i$.
- $C^* \in \mathcal{C}$ is the target concept that we wish to learn.
- The learning algorithm takes S as input and outputs a $C_n \in \mathcal{C}$.

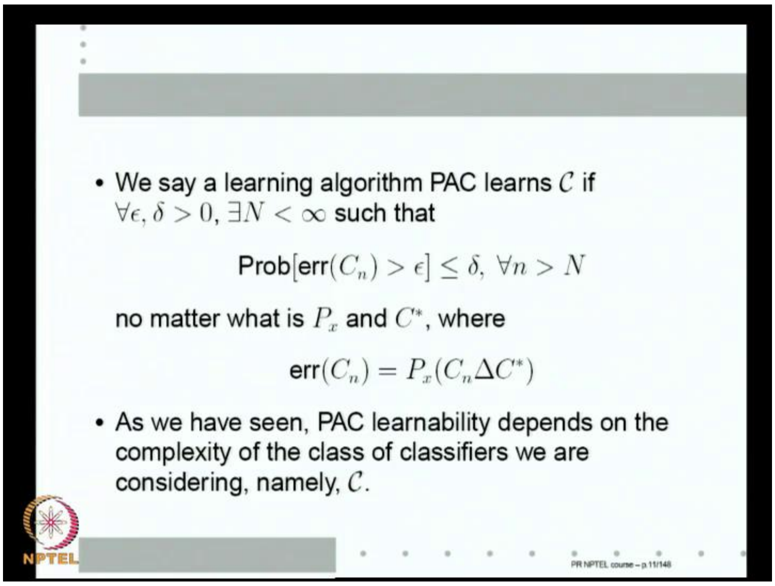
PR NPTEL course – p.9/148

So, let us start by briefly recalling the PAC learning framework, so that we can now extend it. The PAC learning framework is as follows, we are actually given the instance space what, which we denote it as script x. So, all your feature vectors are input to a learning algorithm is an element of the instance space or script x, it is called so called the input space. y is the output space, that is the set of class labels for considering two class classifier. y will be either the set 0 comma 1 or the set z plus 1 gamma minus 1 and so on.

And, a script C is said to be a concept space. It is a set of subsets of x. It is a subset of C itself is a subset of power set of x. So, elements of C are subsets of x. So, we have taken some specific subsets of x, and call it the concept space or the set of classifiers. This determines what class of classifiers we are searching over. And, we had given examples, where X i are drawn i i d according to some unknown distribution P x on X, and y i is obtained by classifying X i using a one particular element from C which is called C star. C star is the target concept. Of course, we do not know C star, we do not know P x.

But we know X, y and C. So, its etching over C to find a good approximation to C star based on the examples X i, y i given to us that is the set of examples S. So, given the set of examples S, we have to set over the concept space C to find some classifier which we hope will be a good approximation to C star if there are enough examples given or notation was that, if I am learning with S. S has exactly n examples. So, after seeing n examples i output some element of C which is denoted by C n.

(Refer Slide Time: 04:36)



- We say a learning algorithm PAC learns $\mathcal{C}$ if $\forall \epsilon, \delta > 0, \exists N < \infty$ such that

$$\text{Prob}[\text{err}(C_n) > \epsilon] \leq \delta, \ \forall n > N$$

no matter what is P_x and C^*, where

$$\text{err}(C_n) = P_x(C_n \Delta C^*)$$

- As we have seen, PAC learnability depends on the complexity of the class of classifiers we are considering, namely, $\mathcal{C}$.

PR NPTEL course – p.11/148

Now, the issue of correctness is addressed as follows. We say the algorithm PAC learns the concept class C. If given any epsilon delta there exists some n, so that probability errors C n greater than epsilon is less than delta for all n greater than n, which essentially means error C n converges to 0 in probability that is, what this means? So, essentially given sufficiently given many examples with a large probability, I make less than epsilon error, error less than epsilon is greater than one minus delta.

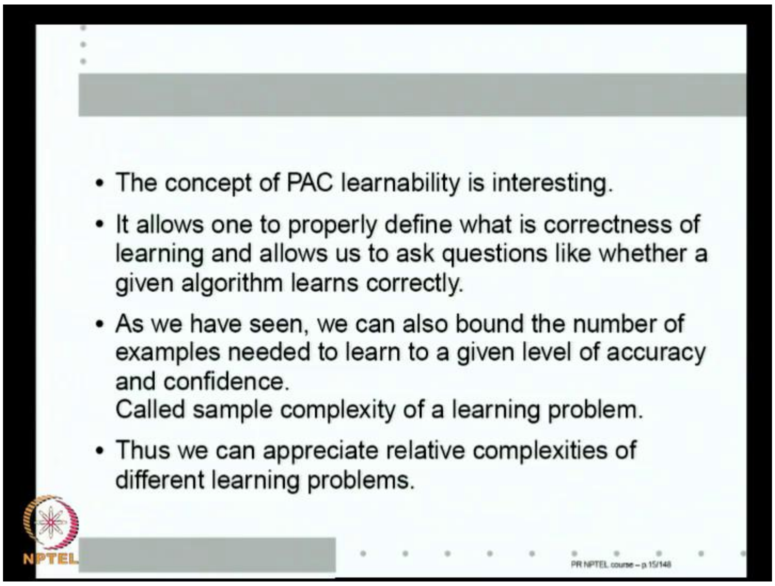
So, with a large probability I can make, I will make small error. This must be 2 for every epsilon delta. You give me the accuracy to which I have to learn that is epsilon and the confidence with which I have to learn delta, then I will give you a set of number of examples I need. If you give me that many examples I will always be able to learn to that level of accuracy and confidence. No matter what the distribution P x is and what the target concept C star is, as long as C star is in C, where error is defined as the P x probability of C n delta C star. What is C n delta C star?

It is the symmetric difference between the set C n and C star, so the set of all points that are in C n but not in C star, in C star, not in C n. So, these are essentially C n delta C star consists of the set of points in x where the classification based C n and C star will differ. So, we are asking what is the probability assigned to that set of points under the distribution P x. That is the probability of C n making an error under randomly drawn sample. When the random sample is drawn according to distribution P x, because our samples are IID with respect to P x, we are also testing our generalization on samples drawn with respect to P x.

That is why, this is the proper definition for error. Of course, as we have seen errors C n itself is a random variable, because it depends on C n which in turn depends on the random n samples that we got in the sample set S. So, a PAC leaning probably approximately correct, is a learn an approximately correct classifier with a loss probability. Of course, I had written it the other way. We learn a approximately incorrect classifier with a small probability.

Now, this with the large probability we learn a approximately correct classifier. As we have seen last class, whether an algorithm can PAC learn or not depends on the complexity of the class of classifiers we are considering. So, we seen a example where everything else is same but, if we search our C 1 then the algorithm PAC learns. If we search over C 2 the algorithm does not PAC learn. So, essentially if the class of classifiers is too complex, then I may not be able to learn anything. That is that is what we gathered from the the illustration illustrative example problem that we considered last class to explain PAC learning.

(Refer Slide Time: 07:33)



- The concept of PAC learnability is interesting.
- It allows one to properly define what is correctness of learning and allows us to ask questions like whether a given algorithm learns correctly.
- As we have seen, we can also bound the number of examples needed to learn to a given level of accuracy and confidence.
  Called sample complexity of a learning problem.
- Thus we can appreciate relative complexities of different learning problems.

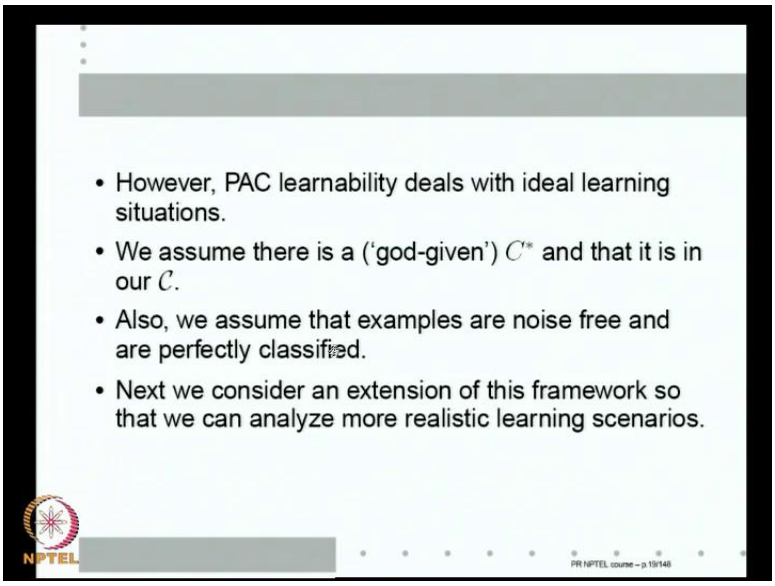
PR NPTEL course – p.15/148

Now, the concept of PAC learning; PAC learnability is a very interesting concept which historically it is at least in computer science circles. This is the first ever time people formally addressed the issue of correctness for learning. It is said statistical circles it was there even in sixties or at least early seventies, similar concepts were there that is what we are going to look at next. PAC learnability the reason why we started PAC learnability is easy to understand. It is very simple framework, it allows one to properly define what is correctness of learning, allows one to ask questions like whether a given algorithm learns correctly?

We have seen an example of showing that an algorithm learns correctly and showing that an algorithm does not learn correctly. We can, in the in the proof that we have given about showing the example learns correctly. We have also seen how given an epsilon and delta we can actually calculate our bound the number of examples that we need to learn to that given level of accuracy in confidence.

So, this number of examples needed as a function of epsilon and delta is often called the sample complexity of the learning algorithm, of the learning problem. So, the PAC learnability also allows us to calculate complexities of learning problems, which in turn means that we can calculate complexity of different learning problems and hence the PAC framework allows us to appreciate which learning problem of given different learning problems which are more complicated that one problem is more complex and the other and so on. So, we can get an appreciation of the relative complexities of different learning problems, right?

(Refer Slide Time: 09:18)

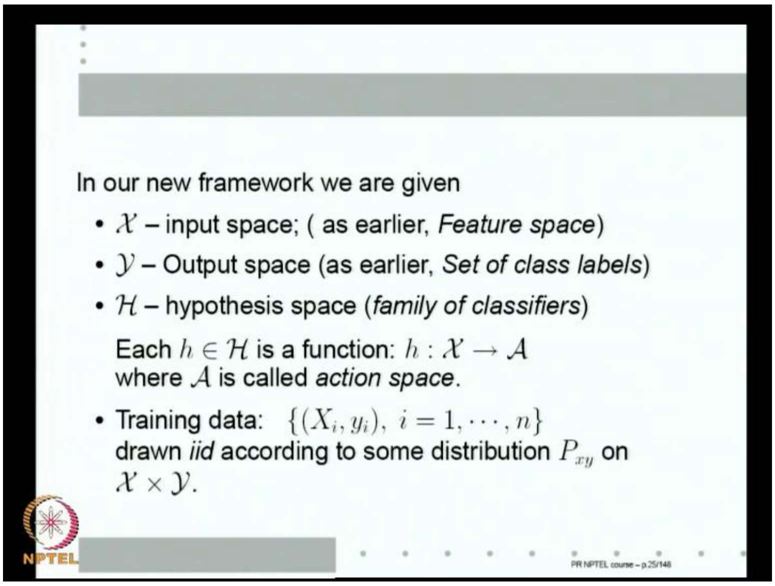


- However, PAC learnability deals with ideal learning situations.
- We assume there is a ('god-given') C^* and that it is in our $\mathcal{C}$.
- Also, we assume that examples are noise free and are perfectly classified.
- Next we consider an extension of this framework so that we can analyze more realistic learning scenarios.

Having said this, the inadequacy of PAC learning is that it deals with what can be called a ideal learning situation. What is ideal about PAC learning? We assume that there's a god given classifiers C star right. And that C star is in our C. In our class of classifier that were searching over right. And the examples or all noise free and perfectly classified, which means at any given time over the class of classifiers. I am searching there is at least one classifier that correctly classifies all examples, right? This this is both these are really difficult assumptions to satisfy, that the God given if even if there is a God given correct classifier that it is there in the bag of classifiers I am searching over. Because before hand, I do not know what classifier structure would perfectly classify perfectly solve this problem and that all examples are perfectly classified is also not a very nice assumption.

So, next we consider an extension of the PAC framework, so that we can analyze more realistic learning scenarios. Where we do not have to assume that there exists a target concept with respect to which all examples are correctly classified. And we can take care of at least the ordinary noise that we expect in any examples for any classification problem.

(Refer Slide Time: 10:46)



In our new framework we are given

- $\mathcal{X}$ – input space; (as earlier, *Feature space*)
- $\mathcal{Y}$ – Output space (as earlier, *Set of class labels*)
- $\mathcal{H}$ – hypothesis space (*family of classifiers*)

  Each $h \in \mathcal{H}$ is a function: $h : \mathcal{X} \rightarrow \mathcal{A}$
  where $\mathcal{A}$ is called *action space*.
- Training data: $\{(X_i, y_i),\ i = 1, \cdots, n\}$
  drawn *iid* according to some distribution P_{xy} on
  $\mathcal{X} \times \mathcal{Y}$.

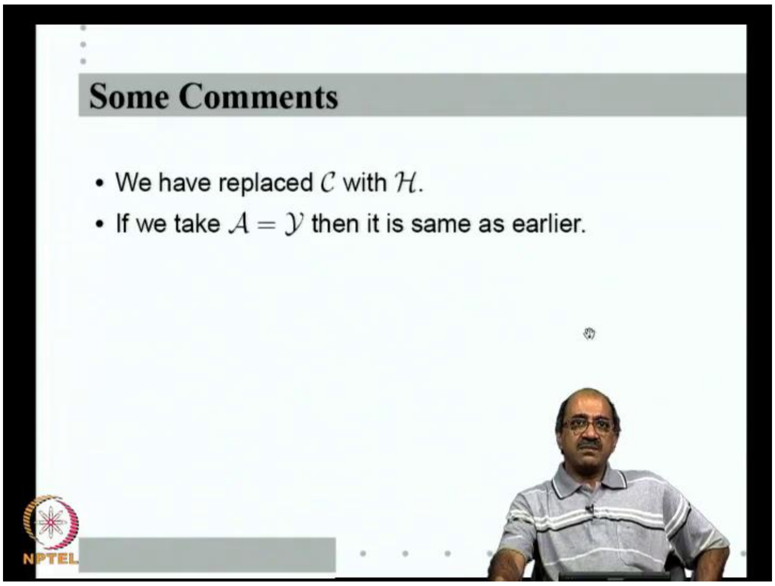
PR NPTEL course – p.25/148

So, let us now describe this new frame work. Essentially, we are going to extend it in such a way that we do not have to worry about C star and we can talk about examples are are not perfectly classifiable. That means, same x may come from different classes with different probabilities. So, the notation for the new frame work is as follows. As a as earlier we have the input space x, which is the feature space. The output space y which is set of class labels or it will be equal to or if I consider regression problem.

Now, we have what is called a hypothesis space. This is going to be our family of classifiers. Of course, it is not just that we changed the name from C to h. We are not assuming that h is a necessarily a subset of 2 power x. Instead what will do is each element of the hypothesis space is some function that maps x to some other set A. If we take y to be 0 1 and every element of h maps x to 0 1, then it is same as h being a subset of 2 power x and is same as concept. But, here we will make h such that each element of h is a function that maps x to some other set. We call that set A which is, which stands for action space. We will currently see why this will allow for many different learning situations to be incorporated.

Another difference is, now the training set x i y i of n examples is drawn iid according to some distribution P x y and x cross y. We are not saying you first draw from x and then classify with a God given classifier C star. Now, training data is obtained by sampling from x cross y using some distribution P x y. Once again like P x in the other case we do not know P x y. But, the training set is drawn according to some distribution x cross y, okay?

(Refer Slide Time: 12:55)

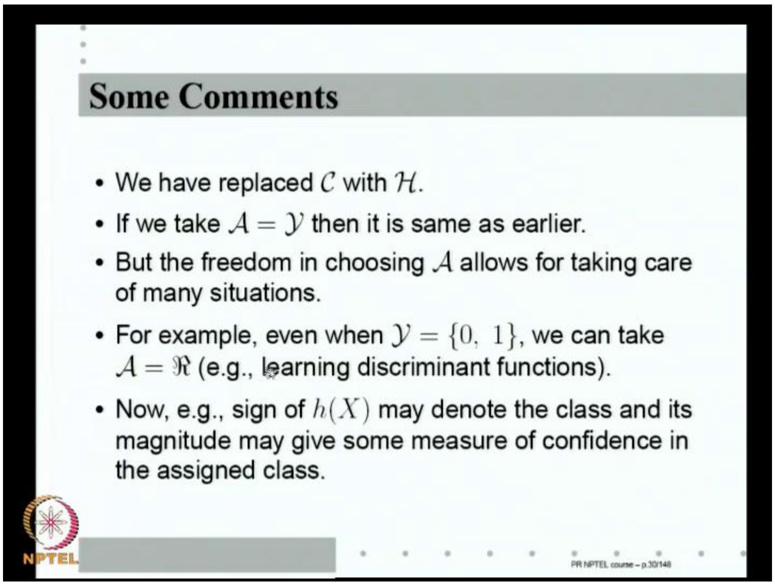


Some Comments

- We have replaced $\mathcal{C}$ with $\mathcal{H}$.
- If we take $\mathcal{A} = \mathcal{Y}$ then it is same as earlier.

So, let us ask what does this framework give us extra. The first thing as I said we have done here is we have replaced what we call concept space C with the hypothesis space H. C was defined to be 2 power x. We are taking y to be 0 1 and every we are searching over 2 class classifiers only. But now, so what it means is in the, in the new setup each element is the hypothesis space H is a function that maps the input space to the action space, right?

(Refer Slide Time: 13:47)



Some Comments

- We have replaced $\mathcal{C}$ with $\mathcal{H}$.
- If we take $\mathcal{A} = \mathcal{Y}$ then it is same as earlier.
- But the freedom in choosing $\mathcal{A}$ allows for taking care of many situations.
- For example, even when $\mathcal{Y} = \{0, \ 1\}$, we can take $\mathcal{A} = \Re$ (e.g., learning discriminant functions).
- Now, e.g., sign of $h(X)$ may denote the class and its magnitude may give some measure of confidence in the assigned class.

PR NPTEL course – p.30/148

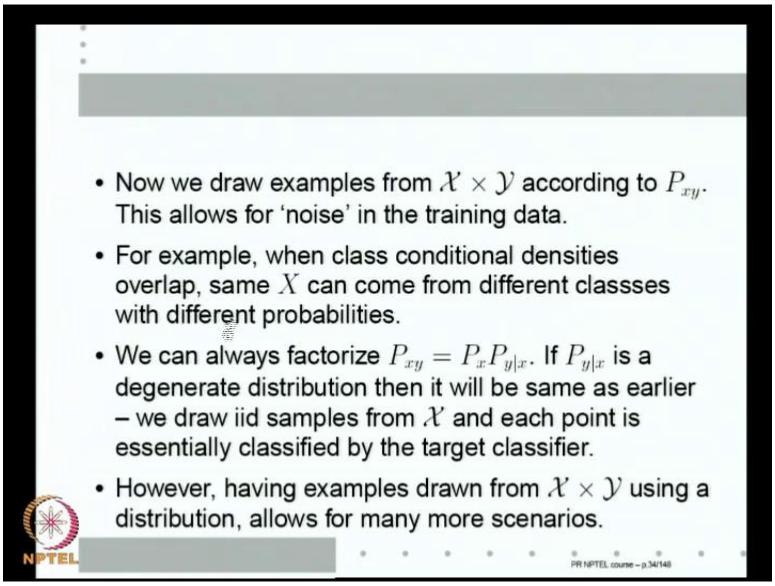
Now, first let us see that if I take A to be equal to y, which of course I can. Then it is same as earlier. Then h becomes same as what we call C earlier. But, because we can choose A, it allows for taking out a many situation. For example, suppose y is still 0 1 a 2 class classifier. But, let us take A to be R, then what I am actually learning are functions on h that take real values. For example, a discriminant function. So, I may be learning a discriminant function, that maps x to real numbers, does not map x to 0 1, right? I can use the discriminant function to classify ultimately. But, I am searching over the class of discriminant functions.

Now, that I can easily incorporate by taking my action space to be R even though y is equal to 0 1 only the examples have to come from x cross y. So, I cannot take y to be R because the examples come from class 0 or class 1 say examples still come from x cross y but, the learning algorithm is searching over functions that map exterior line. So for example, I can easily incorporate learning of discriminant functions here, which I could not have done in the earlier one.

So, in particular once again as an example not that it is always easy to do this as an example you can think that when I take A to be R, right? I am actually learning real valued functions. So, which means I can ultimately think that sign of h X will denote the class label because I am still considering two class problems and the magnitude of h X may give me some measure of confidence in the assigned class. So, suppose X should be in class 1 and h X is positive. If h X is 10, then I would say I have lot more confidence that X is class 1 rather than h X is say 10 to the power minus minus 3.

It is it is just greater than 0. So, in some sense I may be able to use learning problems, which actually deal with such things. We will see later on in this course that we will actually do this. We will use measure of value of h X sometimes as a measure of confidence. So, this is one extra thing that this framework gives us, the flexibility in choosing y. We will see a few more example of choosing A the action space. We will see a few more examples later on.

(Refer Slide Time: 16:07)



- Now we draw examples from $\mathcal{X} \times \mathcal{Y}$ according to P_{xy}. This allows for 'noise' in the training data.
- For example, when class conditional densities overlap, same X can come from different classses with different probabilities.
- We can always factorize $P_{xy} = P_x P_{y|x}$. If $P_{y|x}$ is a degenerate distribution then it will be same as earlier – we draw iid samples from $\mathcal{X}$ and each point is essentially classified by the target classifier.
- However, having examples drawn from $\mathcal{X} \times \mathcal{Y}$ using a distribution, allows for many more scenarios.

PR NPTEL course – p.34/148

The second change in our framework is that the examples are drawn from x cross y, x cross y by x cross y. I mean the Cartesian product of the sets x and y, according to some distribution P x y and x cross y. This allows for much of natural noise. This training see for example, in the in the two class classification problem the two class conditional densities may all. What does that mean? We actually generate the training data which generated by sampling from one class conditional density and the other class conditional density, right? So, which means that the same X can come from different classes of different probabilities.

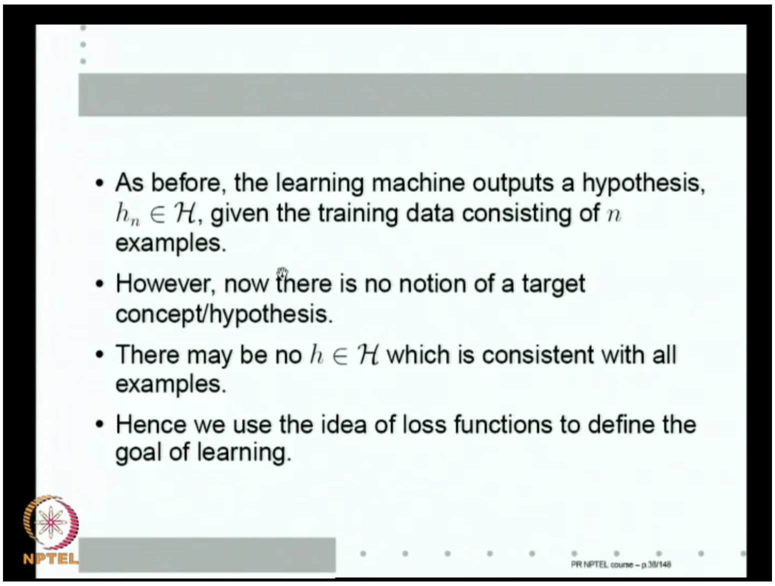
This for example, it is not possible if I have to assume that there is a C star, because C star is a function. Given an X, X as one god given class label. But that is in general not true when class conditional densities overlap my samples never satisfy this condition.

But now this is very nice because my examples are only drawn from x cross y. According to P x y and there is no God given classifier. This natural thing that happens in all classification problems is taken care off. Of course, in general given the joint density of any two random variables x and y, the P x y can always be factorized as to the marginal density of x marginal distribution of x multiplied by the conditional density y given x.

Now, if P y given x say these generate distribution. What is that mean for a given value of x? Because it is a conditional density. Only one value of y has non zero probability and all others have zero probability. That is, when P y given x has degenerated distribution that is same as having a god given classifier. For a given x there is only one possible y. But in general when class conditional densities overlap, the P y given x is a proper density for the same x different y come with different probabilities, right?

So, because we can always factorize like this and we can assume P y given x is a degenerate distribution. The old PAC PAC framework is still part of this new framework right. This new framework is a proper extension of the PAC framework. PAC framework can be still considered a special case of this but we get more flexibility in modelling learning situations. So, having examples done from a class y using the distribution allows for many more scenarios such as overlapping class conditional densities and many more.

(Refer Slide Time: 18:43)



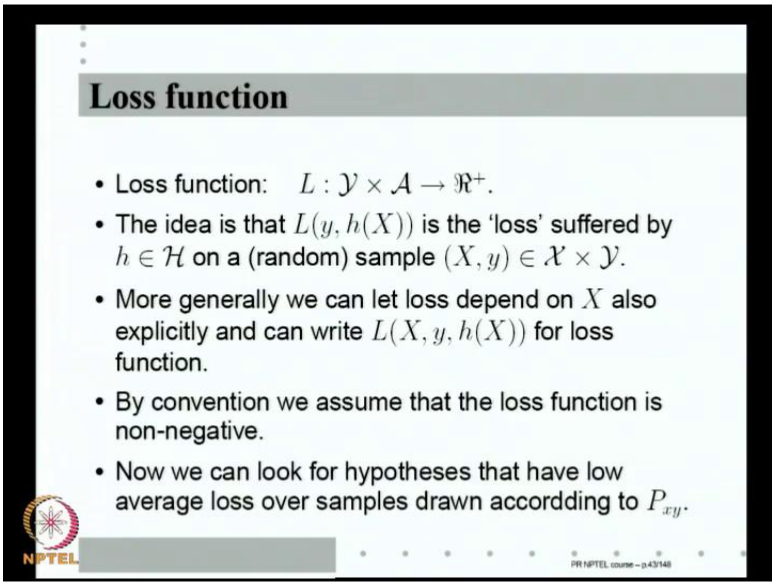
- As before, the learning machine outputs a hypothesis, $h_n \in \mathcal{H}$, given the training data consisting of n examples.
- However, now there is no notion of a target concept/hypothesis.
- There may be no $h \in \mathcal{H}$ which is consistent with all examples.
- Hence we use the idea of loss functions to define the goal of learning.

Now, moving on like earlier, what does a learning machine do? It takes a training data consisting of n examples as input and output some hypothesis from h now. Earlier it is a concept. Now, it is etching over h so it outputs some element of h. Let us call it h of n. So, h subscript n is what the learning algorithm or learning machine will output given training data consisting of n examples. Now, to rate h n how do we rate h n? Earlier there is a God given target concept. So, we can say is h n close to the target concept in terms of classifying new examples.

But, now there is no target concept. So, we have to find some other way of defining by their, the goodness of h n. We have already done it in the beginning of the course. We introduced briefly the idea of loss functions. We used a loss function this 0 1 loss function while developing the bayes classifier. But, that time we have not done this very formally. So, that is what we are going to do now again, but a little more formally now. Since, now to appreciate where you need loos function is not only that there is no notion of a target concept. In general there may be no h that is consistent with all examples.

So, we cannot even say that the the the classification actually is C of h on the samples, is a good indicator of how good it is. Because may be the best possible classification accuracy which does not give you hundred percent accuracy, right? So, there may be no h is consistent with all examples. So, we need a different way to rate different h and for that we use the idea of loss functions to rate different h and hence we define the goal of learning.

(Refer Slide Time: 20:47)



Loss function

- Loss function: $L : \mathcal{Y} \times \mathcal{A} \rightarrow \Re^+$.
- The idea is that $L(y, h(X))$ is the 'loss' suffered by $h \in \mathcal{H}$ on a (random) sample $(X, y) \in \mathcal{X} \times \mathcal{Y}$.
- More generally we can let loss depend on X also explicitly and can write $L(X, y, h(X))$ for loss function.
- By convention we assume that the loss function is non-negative.
- Now we can look for hypotheses that have low average loss over samples drawn accordding to P_{xy}.

PR NPTEL course – p.43/148

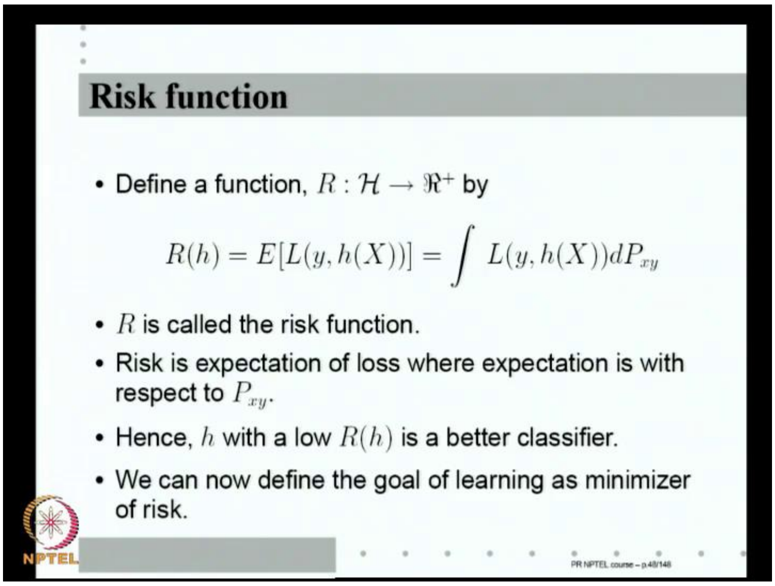
So, what is your loss function? We have already as I said, we already encountered loss functions earlier in the course. A loss function in our new notation now is a function whose domain is y cross A. The Cartesian product of the output space y and the action space A, and maps it to r plus the positive real line. So, the idea is L of y comma h x. L is a function of two variables one is in the space y, the capital script y. That is the for example, the classification problem and classification labels and the other is in the range space of the functions h over, which we are searching.

So, essentially on a random sample X y, a particular hypothesis h it would say h of X. Whereas, it should have said y so L y comma h X is the loss suffered by h on a random sample X y. This is how we earlier also introduced loss functions. In general of course, we can we can make the loss function also depend explicitly on X. So, we can write loss function as L of x comma y comma h X. What it means is; for example, in different regions of the feature space, we may choose to measure loss in using different criteria depending on the application that may be important.

So, we could actually take loss function to have a domain which is x cross y cross A. Most of what we say here in this course will also hold for this mode general loss functions. But to keep notation and other things simple we will only going to look at loss function, a function of only y and h X. Also, by convention we assume that the loss function is non-negative or say the range space of loss function R plus because you are considering it as loss we assume it to be always non negative.

Matter of fact, later on we assume loss function is bounded so that, bounded on one side anywhere it is bounded by zero. We assume it is bounded on the other side also. Given this what is that we want? An random samples X y, A h does h suffers a loss L of y comma h X. Now, I have learnt from samples drawn according to distribution P x y using the same idea of fairness, that we used in the PAC learning. Now, I can say that a h is good if an samples drawn according to P x y, its average loss is small, right? That is, how we are going to define the goal of learning. Essentially on samples drawn according to the same distribution P x y our hypothesis that have low average loss or better.

(Refer Slide Time: 23:33)



Risk function

- Define a function, $R : \mathcal{H} \rightarrow \Re^+$ by

$$R(h) = E[L(y, h(X))] = \int L(y, h(X)) dP_{xy}$$

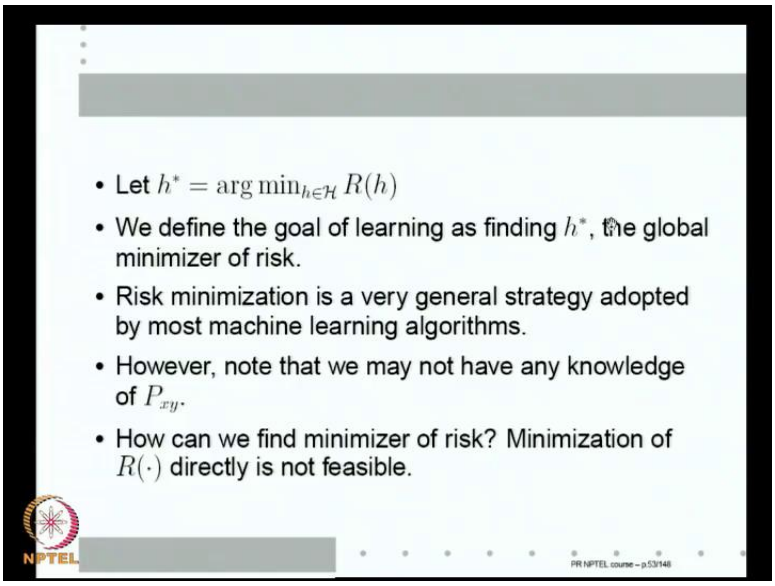
- R is called the risk function.
- Risk is expectation of loss where expectation is with respect to P_{xy}.
- Hence, h with a low $R(h)$ is a better classifier.
- We can now define the goal of learning as minimizer of risk.

PR NPTEL course – p.48/148

So, we define a function R that maps H to R plus. That means it assigns a real number to every hypothesis so it is essentially a functional. By R of h is expected value of L y comma h X, where expectation with respect to P x y. So, that is that is the expectation taken this like a Rieman still this integral. But anyway this is essentially a expectation integral with respect to the distribution P x y. So, R is called the risk function right. So, R of h is called the risk of h and R is called the risk function. What is risk? Risk is expectation of loss, where the expectation is with respect to P x y.

So, the samples are given drawn with respect to P x y and I measure goodness of h, with the average loss h will suffer on samples drawn according to P x y, because risk is expectation of loss, where expectation with respect to the same distribution P x y. So, if we have A h with low R h that is a better classifier, right? Because on the average on similar samples it suffers low loss. So, this is what we are going to use to define the goal of learning. So, the goal of learning is to find A h that minimizes risk. So, the goal is find the minimizer of the risk function, okay?

(Refer Slide Time: 24:56)



- Let $h^* = \arg\min_{h \in \mathcal{H}} R(h)$
- We define the goal of learning as finding h^*, the global minimizer of risk.
- Risk minimization is a very general strategy adopted by most machine learning algorithms.
- However, note that we may not have any knowledge of P_{xy}.
- How can we find minimizer of risk? Minimization of $R(\cdot)$ directly is not feasible.

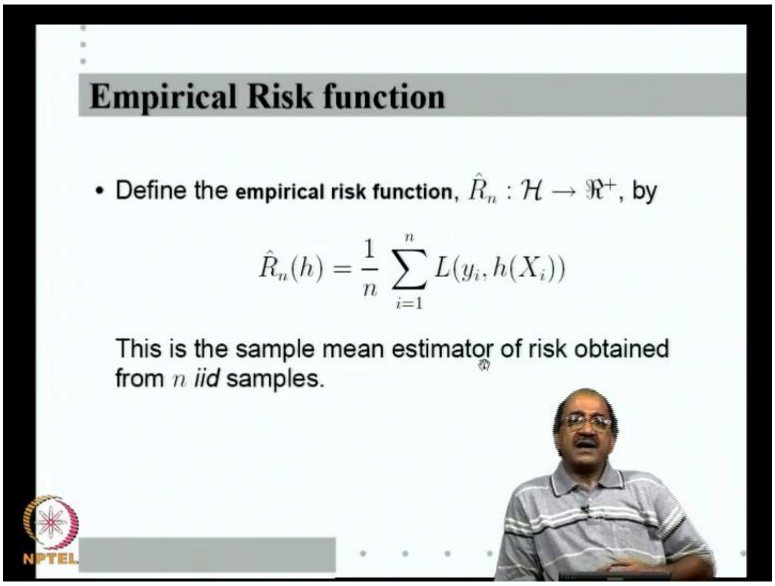
PR NPTEL course – p.53/148

So, let us formulize this. Let us denote by h star. The arg, the overall script h find the minimizing h to R h. So, thus arg mean h belonging to h R h. So, h star is nothing but the global minimizer of risk. R h star is the global minimum of risk. So, h star is the global minimizer of risk and we define the goal of learning as finding h star, the global minimizer of risk. In a given learning problem, where you are given x y and A and the loss function and P x y is arbitrary, h star may not be unique. There may be many different h star all of which achieve the global minimum.

But it really does not matter. For us risk is what we want to minimize. So, any h star that minimize the risk is same and we are going to compare different classifiers only in terms of the risk value that they have and hence, whether or not h star is unique does not concern us, right? h star may be unique may not be unique. But the goal of learning is to find a global minimizer of risk. So, risk minimization is a very, very general strategy adopted by almost all machine learning algorithms. Everything that we considered so far, all our linear classifier learning algorithms are essentially risk minimization algorithms right under a particular loss function.

Many other algorithm that we consider later on in the course can all be viewed in this general framework of risk minimization. But minimizing risk is not a easy problem, because we have no knowledge of P x y and R of h is expectation of L y comma h x with respect to the distribution P x y. And since we do not have P x y, given A h, I cannot calculate R of h. So, how do how can I minimize R? Minimization of R directly is not feasible because I cannot even calculate R h given A h, because R definition of R h involves an expectation integral with respect to P x y and I have no knowledge of P x y.

(Refer Slide Time: 27:10)



Empirical Risk function

- Define the **empirical risk function**, $\hat{R}_n : \mathcal{H} \to \Re^+$, by

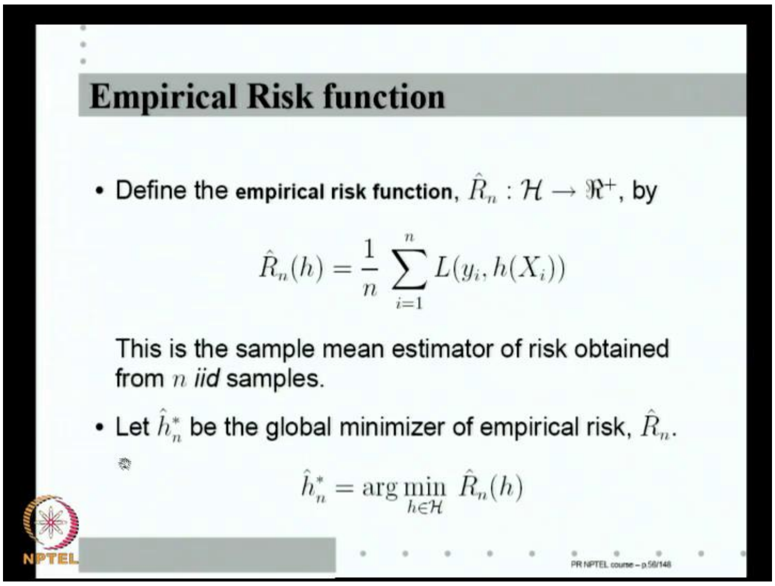
$$\hat{R}_n(h) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, h(X_i))$$

This is the sample mean estimator of risk obtained from n *iid* samples.

So, what do I do? As you have already seen briefly in the beginning of the course, we define another function which I am calling empirical risk function Where we call it R hat of n. If you recall our notation from the estimation part of this course when we consider various as a parameter estimation. A hat denotes an estimate of some quantity. So, I do not know R so I am estimating R. So, this is R hat, and the subscript of an estimator denotes the sample size. Because I am estimating with n examples, n is the subscript that is why we call this function R hat of n, which also assigns a real number to every h and that number is simply 1 by n summation i is equal to 1 to n. L of y i h of X i, where x i y i, i is equal to 1 to n is the sample set, right?

So, R hat n is nothing but the sample mean estimate of the risk. Actual risk is the expectation of L y comma h X. If I cannot calculate this expectation, but I have iid samples y i h x i. Then I can approximate this expectation by its sample mean, right? That is what we are doing. This is the sample mean estimator of risk obtained from an IID samples. That is why is called R hat. Say the idea is that sample mean is a good estimator. So, R hat and H would be close to R h. So, because I cannot minimize R I will minimize R hat n.

(Refer Slide Time: 28:48)



Empirical Risk function

- Define the **empirical risk function**, $\hat{R}_n : \mathcal{H} \rightarrow \Re^+$, by

$$\hat{R}_n(h) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, h(X_i))$$

This is the sample mean estimator of risk obtained from n *iid* samples.

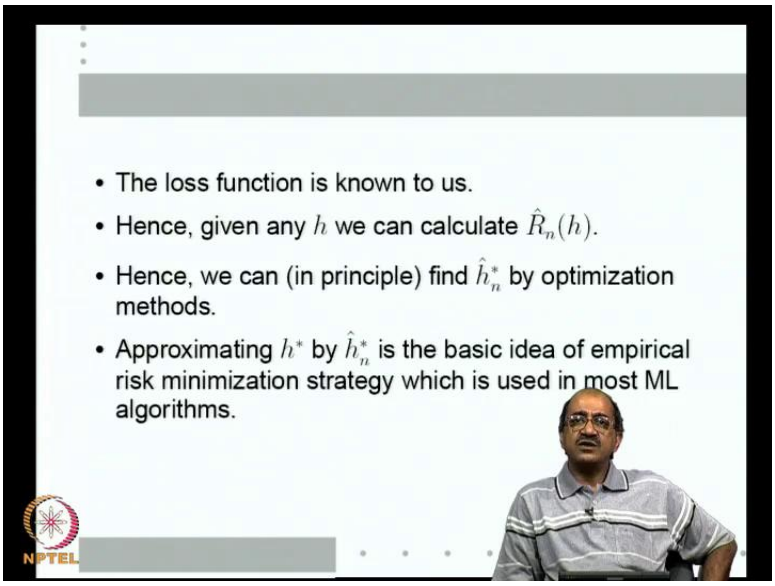
- Let $\hat{h}_n^*$ be the global minimizer of empirical risk, $\hat{R}_n$.

$$\hat{h}_n^* = \arg\min_{h \in \mathcal{H}} \hat{R}_n(h)$$

PR NPTEL course – p.58/148

So, let the global minimizer of R hat n be h hat star n. This is quite a mouthful. The reason why we chose what might look like a strange notation of the following. What is your goal? h star, we do not know h star, we want to estimate h star. So, the estimator for h star will be h hat star, right? A hat always denotes estimate. So, this is estimating h star and n is the sample size. So, h hat star n is the estimator for h star based on n samples and that is nothing but the global minimizer of the empirical risk or hat n. Now, given a particular h because these samples are there and I know the loss function, I can always calculate this, right? Let us let us remember that.

(Refer Slide Time: 29:39)

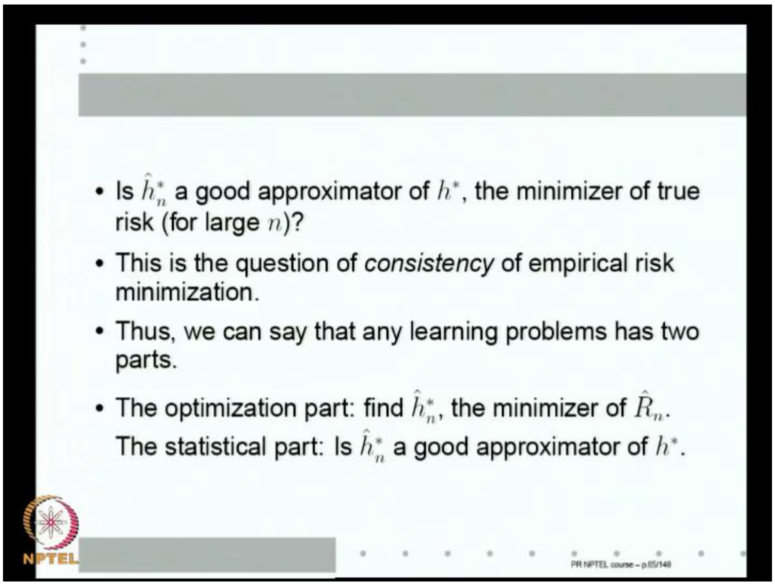


- The loss function is known to us.
- Hence, given any h we can calculate $\hat{R}_n(h)$.
- Hence, we can (in principle) find $\hat{h}_n^*$ by optimization methods.
- Approximating h^* by $\hat{h}_n^*$ is the basic idea of empirical risk minimization strategy which is used in most ML algorithms.

The loss function is known to us. Hence, given A h we can calculate R hat n of h, which means in principle we can always find h hat star n by optimising R hat n. Of course, we the it depends on what the loss function is that is. The summation in the empirical risk minimization may be simple to minimize, may be complicatedly minimize we do not know the parameterization for h. How complicated different h functions are. But all that apart given any h, I can calculate R hat n h. The expression is there.

Hence, in principle we might be able to find h hat star n by optimizing, right? So, this is a very generic strategy followed by all algorithms. We essentially want to do risk minimization. But we cannot do risk minimization. So, we do empirical risk minimization. That is we are approximating h star by h hat star n. This is the strategy used by all machine learning algorithm or at least most of the machine learning algorithms, okay?

(Refer Slide Time: 30:52)



- Is $\hat{h}_n^*$ a good approximator of h^*, the minimizer of true risk (for large n)?
- This is the question of *consistency* of empirical risk minimization.
- Thus, we can say that any learning problems has two parts.
- The optimization part: find $\hat{h}_n^*$, the minimizer of $\hat{R}_n$.
  The statistical part: Is $\hat{h}_n^*$ a good approximator of h^*.

PR NPTEL course – p.85/148

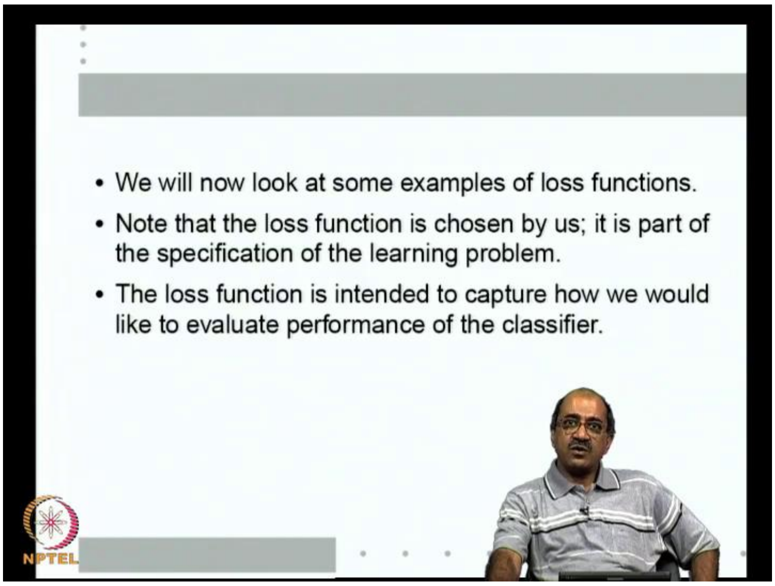
So, the question is if I did this is h hat star n a good approximate of h star? Right, this is called the question of consistency of the empirical risk minimization. I actually want to minimize risk. I cannot so I am minimizing the empirical risk. So, I am asking is this consistent? If I find the minimizer of empirical risk; is that a good approximation to the minimizer of the true risk? Is h hat star n good approximate of h for large n of n, right? This is the question we want to ask. We already know this question is not trivial, even in the simple situation when we have PAC and a target concept.

We have exhibited concept classes where this h hat star n is not a good approximator of h star. So, the issue is we we are going to develop some ways of addressing the issue or at least understanding when h hat star n would be a good approximator of h star. So, before we go there we can now sum up the whole idea of empirical risk minimization as saying that any learning problem has two parts. The first part is an optimization part. That is find h hat star n.

So, once I once i i start with the loss function I choose a loss function I specify a loss function, then I can calculate given the samples I can calculate the empirical risk. Now, that is a optimization problem one needs clever algorithms, the optimization problem might be very complex. We have to find efficient ways of dealing with it. So, that is one half of the learning problem. Find the minimizer of the empirical risk. The second half of the learning problem is the statistical part.

Now, to minimize R hat n, I would have to decide what is the range of h, what is the domain of R hat n? That is, over what class of functions h am I optimizing R hat n. Now, while I can choose it to make the optimization problem simpler, I should also pay attention to so that I chose it, so that it is it becomes a good approximate of h star. That is the statistical part. Right now, we are we are asking the statistical question. How can you say for what kind of h's can you see h R star n will be good approximated to h star?

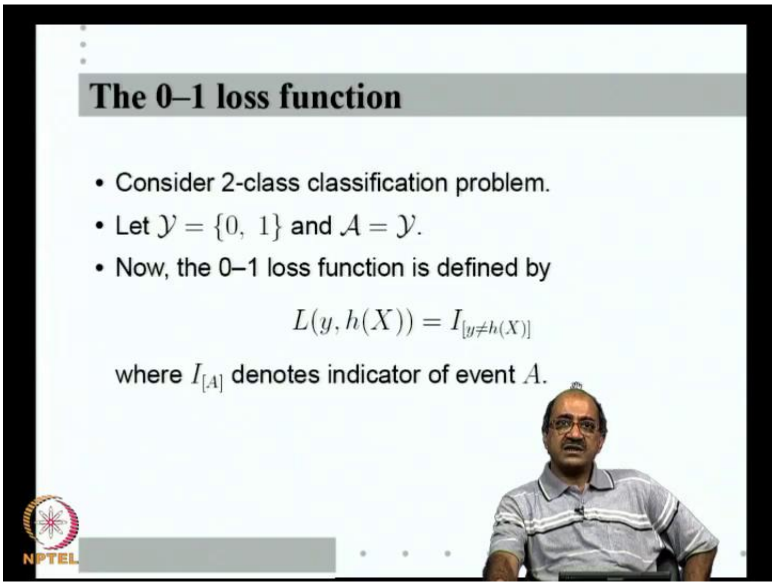
(Refer Slide Time: 33:15)



- We will now look at some examples of loss functions.
- Note that the loss function is chosen by us; it is part of the specification of the learning problem.
- The loss function is intended to capture how we would like to evaluate performance of the classifier.

Before we go there, let us look at a few example loss functions. We only looked at 0 1 loss earlier. We will look at a few more example of loss functions. The loss function is chosen by us, right? It is part of this specification as a learning problem. So, we can choose class functions for whatever reasons. The loss function is essentially intended to capture how we would like to evaluate performance of our classifiers, right? So, a loss function tells you know how we we rate h x versus y. When y is what a what I should have said but I said h x, right? So, it is our specification of how to evaluate performance, okay?

(Refer Slide Time: 33:51)



The 0–1 loss function

- Consider 2-class classification problem.
- Let $\mathcal{Y} = \{0,\ 1\}$ and $\mathcal{A} = \mathcal{Y}$.
- Now, the 0–1 loss function is defined by

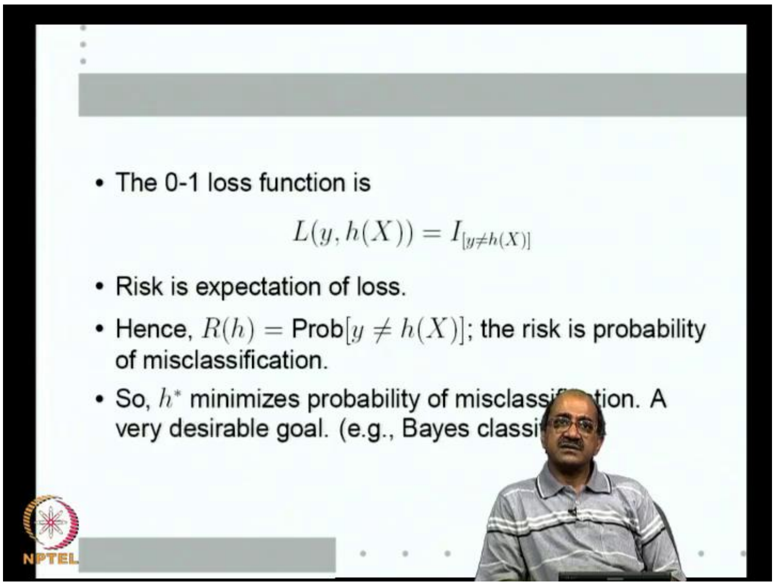
$$L(y, h(X)) = I_{[y \neq h(X)]}$$

where $I_{[A]}$ denotes indicator of event A.

So, we run through a few loss functions lets go back to the zero one loss function. Let me consider only the two class classification problem though it can be done from multiclass also as we know. So, take y is equal to 0 1 and in this case we take A is equal to y. So, the class of classifiers over which you are searching are actually classifiers. They are binary valued functions on x that is what taking A is equal to y means.

Now, the 0 1 loss function is defined by given any y and h x L of y comma h X, is the indicator function of y not equal to h X. That is, if y is not equal to h X that is 1 if y is equal to h x that is equal to 0. That is what I of A is indicator of A means the event occurs then its value is 1, otherwise value is 0. So, L y comma h X is 1 if y is not equal to h X is equal to 0 otherwise, okay?

(Refer Slide Time: 34:42)



- The 0-1 loss function is

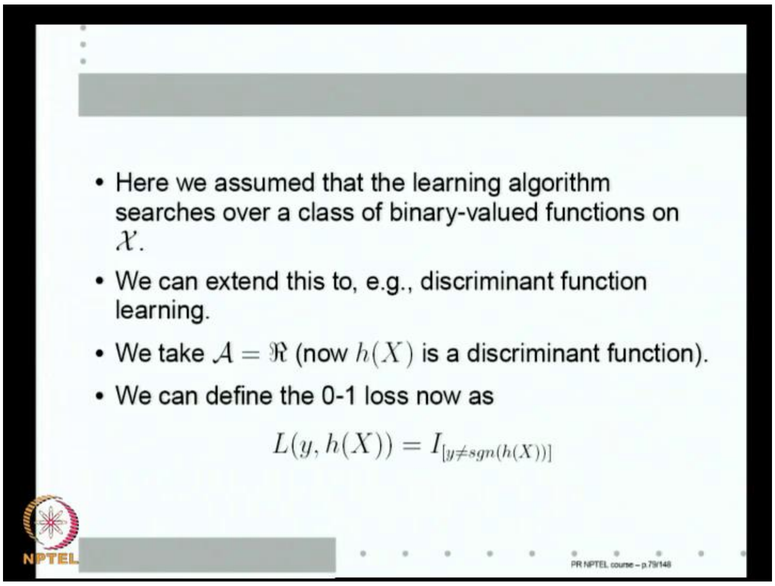
$$L(y, h(X)) = I_{[y \neq h(X)]}$$

- Risk is expectation of loss.
- Hence, $R(h) =$ Prob$[y \neq h(X)]$; the risk is probability of misclassification.
- So, h^* minimizes probability of misclassification. A very desirable goal. (e.g., Bayes classi

So, that is the 0 1 loss function, because risk is expectation of loss. If I want to find risk of h, this is expectation of this. This is a binary random variable, so its expectation is nothing but the probability that takes value 1. So, R of h is probability y, is not equal to h X that is when this random variable the indicator will take 1. So, R h is nothing but probability of y is not equal to h X. So, risk is nothing but the probability of misclassification.

So, h star minimizes probability of misclassification. So, if I choose 0 1 loss function and search over a class of classifiers, that only because we have taken A is equal to y. Our goal of learning is a function h star or a classifier h star that minimizes problem misclassification. So, that is a very nice very desirable goal. For example, that is what we did for a Bayes classifier, right? h star minimizes probability of misclassification. If I choose 0 1 loss function.

(Refer Slide Time: 35:45)



- Here we assumed that the learning algorithm searches over a class of binary-valued functions on $\mathcal{X}$.
- We can extend this to, e.g., discriminant function learning.
- We take $\mathcal{A} = \Re$ (now $h(X)$ is a discriminant function).
- We can define the 0-1 loss now as

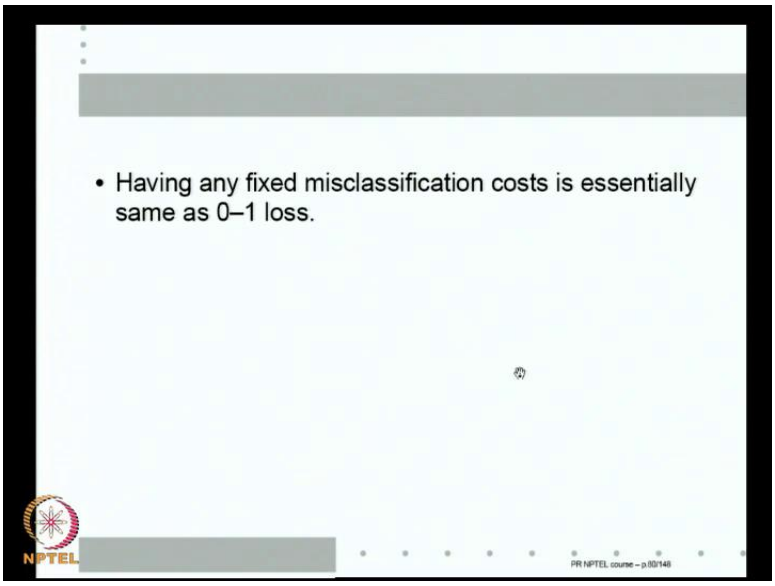
$$L(y, h(X)) = I_{[y \neq sgn(h(X))]}$$

PR NPTEL course – p.79/148

Now, in defining this so far we assumed that we were searching over class of binary valued functions we have taken A to be y, right? We do not have to hat is the beauty of our formalism. We can extend this for example, to take discriminant function learning. What do we do? We take A is equal to R. Instead of taking A is equal to 0 1 we take A is equal to R. Now each h is a function that maps X to real line so is a discriminant function.

So, we are searching over the family of discriminant functions. Now, we can define the 0 1 loss as L of y comma h X is indicator of y not equal to sgn of h X. So, even though we are learning our h X, the loss depends on only on y n sgn of h X. So, we will define L on y cross A that is 0 1 cross real line but given A y and h X, the loss value is 1 if y is not equal to sgn of h X, is 0 otherwise, right? This will allow me to extend the 0 1 loss to loss function even to the case of learning discriminant functions.

(Refer Slide Time: 37:05)

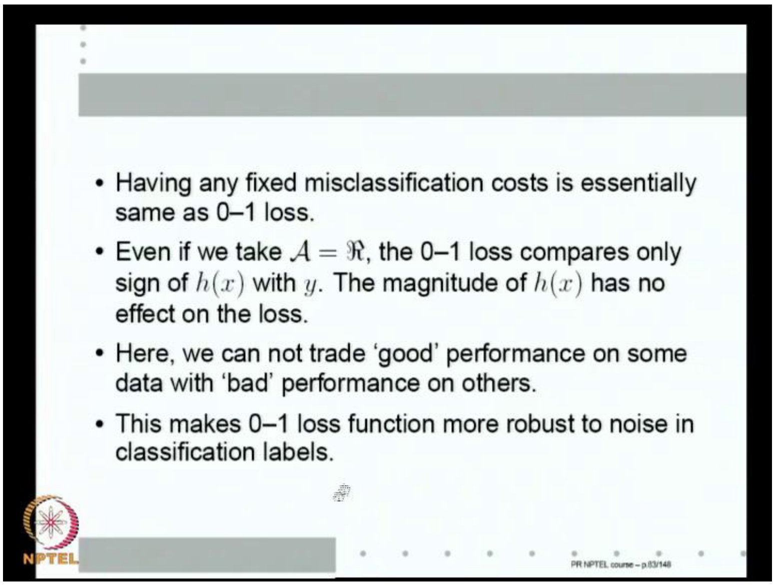


- Having any fixed misclassification costs is essentially same as 0–1 loss.

PR NPTEL course – p.80/148

There are a few other interesting things that we can do here. First either in this or in the earlier case where we assume h X itself is binary, so I can write y is not equal to h X. Essentially if I classify correctly the loss is 0 if I classify incorrectly the loss is 1.

(Refer Slide Time: 37:37)



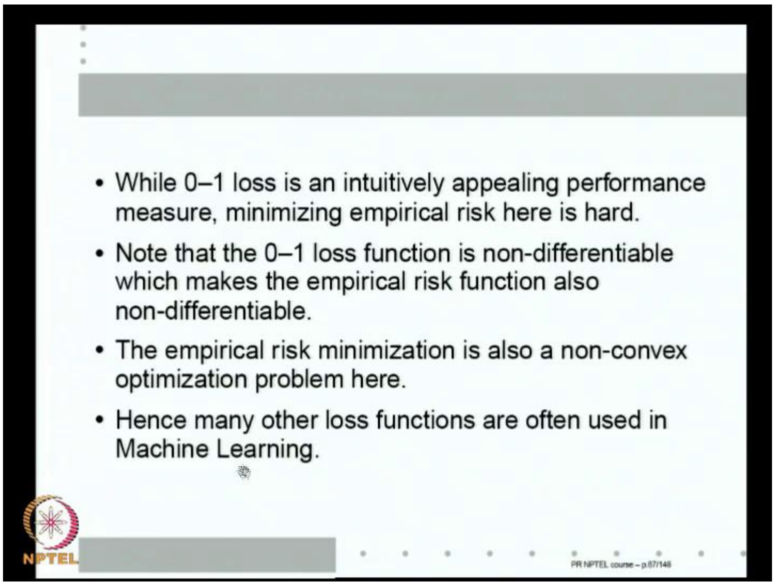
- Having any fixed misclassification costs is essentially same as 0–1 loss.
- Even if we take $\mathcal{A} = \Re$, the 0–1 loss compares only sign of $h(x)$ with y. The magnitude of $h(x)$ has no effect on the loss.
- Here, we can not trade 'good' performance on some data with 'bad' performance on others.
- This makes 0–1 loss function more robust to noise in classification labels.

PR NPTEL course – p.83/148

One is of course arbitrary right i instead of 1 I can take 10, it makes no difference. So, having any fixed misclassification costs is essentially same as 0 1 loss, right? We have already seen that the two is even when we take A is equal to R. Even when we consider discriminant function learning, this 0 1 loss compares only sign of h x with y. The magnitude of h x has no effect on the loss, right? This is very important property of the 0 1 loss function. It only depends on correctness or otherwise of the classification you can do by using the function h, right? Because it only compares sign of h x with y which for example means, sometimes I classify h x correctly with a high confidence, does not give me any extra brownie points, right?

I classify correctly or not is all that 0 1 loss function can. While at this point it may not be clear whether it is good or bad. It turns out that this makes 0 1 loss function more robust to noise in classification labels. So, even if or sometimes some of the labels given are wrong by by mistakenly given it by training set. Minimizing empirical risk are risk under 0 1 loss function gives me some robustness in noise. We currently would not consider it may be later on in the course if if occasion arises I will show you why.

(Refer Slide Time: 38:58)



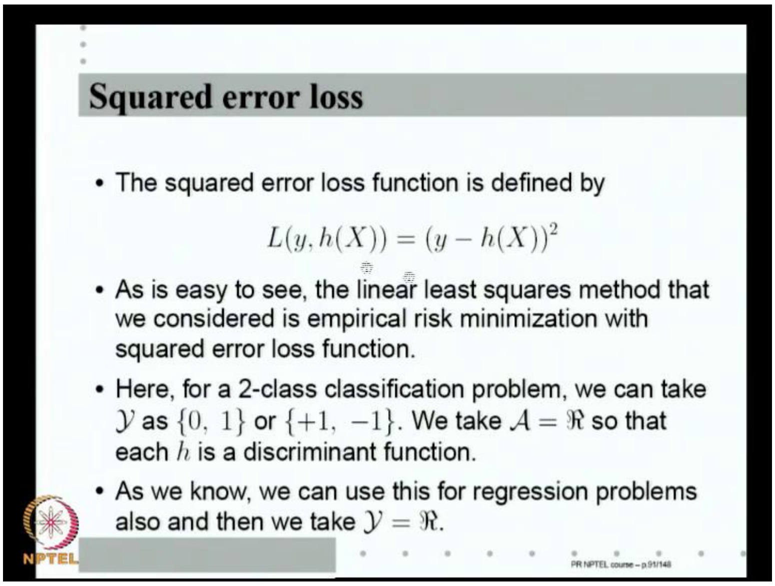
- While 0–1 loss is an intuitively appealing performance measure, minimizing empirical risk here is hard.
- Note that the 0–1 loss function is non-differentiable which makes the empirical risk function also non-differentiable.
- The empirical risk minimization is also a non-convex optimization problem here.
- Hence many other loss functions are often used in Machine Learning.

PR NPTEL course – p.87/148

So, 0 1 loss is intuitively appealing it under 0 1 loss the the best function h star is 1 that minimizes probability of misclassification. So, it is intuitively appealing. But minimizing empirical risk under zero one loss function is a hard problem. Firstly 0 1 loss function is non differentiable and hence empirical risk function is also non differentiable. So, optimizing a non differential function is that much more complicated and the empirical risk minimization the 0 1 loss often turns out to be a non convex problem.

We will see it shortly a little while give you some basic idea of this non convex it is a loss functions later on. So, because a non convex optimization problem, the optimization is always a little more complicated and there are many other loss functions which are used to make the optimization problem easier, okay?

(Refer Slide Time: 39:57)



Squared error loss

- The squared error loss function is defined by

$$L(y, h(X)) = (y - h(X))^2$$

- As is easy to see, the linear least squares method that we considered is empirical risk minimization with squared error loss function.
- Here, for a 2-class classification problem, we can take $\mathcal{Y}$ as $\{0,\ 1\}$ or $\{+1,\ -1\}$. We take $\mathcal{A} = \Re$ so that each h is a discriminant function.
- As we know, we can use this for regression problems also and then we take $\mathcal{Y} = \Re$.

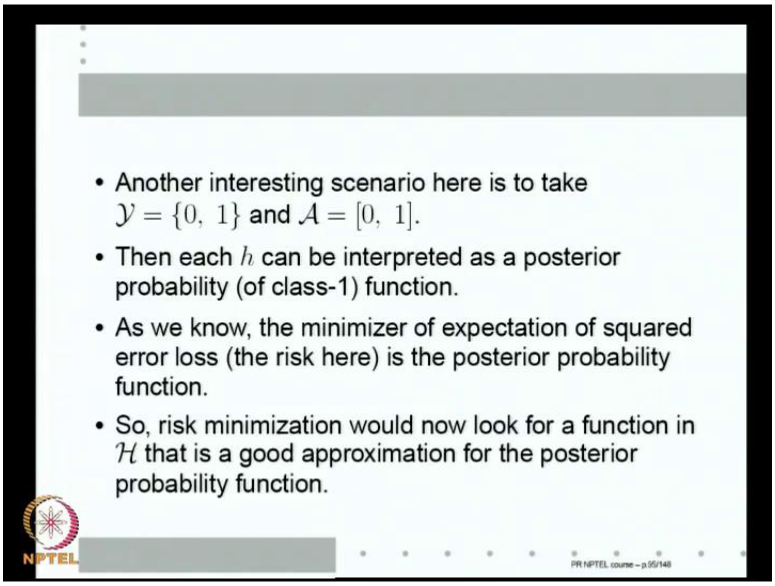
PR NPTEL course – p.91/148

So, look at just a couple of them there are others which we will through the course later on. One is one that you already know though we may not have explicitly emphasized the loss function aspect of it. It is called the squared error loss function defined by L y comma h X is y minus h X whole square, right? As is easy to see the linear least squares method is essentially empirical risk minimization squared error loss function. What are those minimizing i is equal to 1 to n, y i minus h X i whole square.

So, that is nothing but the empirical risk except for the 1 by n factor as we said when we considered linear least squares method that 1 by n factor makes no difference. So, the linear least squares method we considered, linear least squares classification regression all of those are essentially empirical risk minimization with square error loss function. Because for a two class classification problem as we have seen we can take y as 0 1 or plus 1 minus 1.

We take A is equal to R, right, where where we we have considered h of X which is a f n f n function w transpose x plus w naught. So, we have taken A to be R. So, each h is a discriminant function as we have seen this kind of empirical risk minimization is a feasible only under new framework not under the PAC framework, right? Because we take A to be R here or the functions for which we are searching are all real valued functions. Of course, we can also use squared error loos for regression in which case we take y to be real line.

(Refer Slide Time: 41:26)



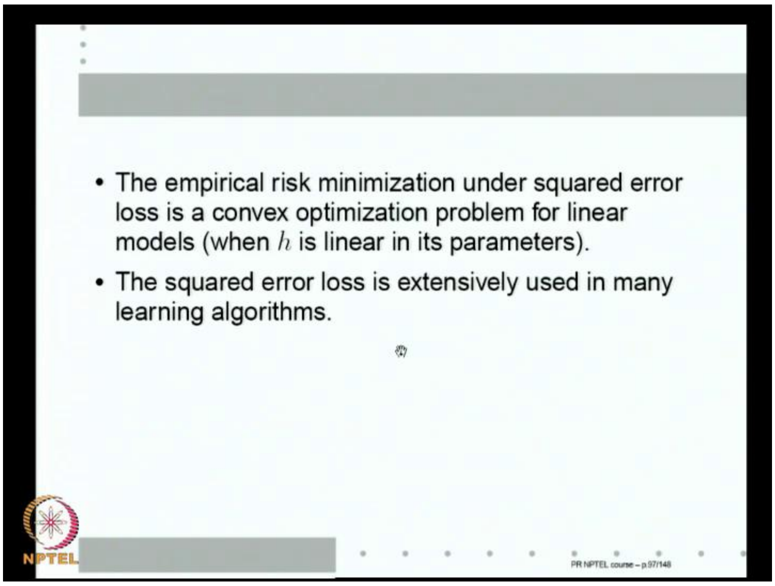
- Another interesting scenario here is to take $\mathcal{Y} = \{0,\ 1\}$ and $\mathcal{A} = [0,\ 1]$.
- Then each h can be interpreted as a posterior probability (of class-1) function.
- As we know, the minimizer of expectation of squared error loss (the risk here) is the posterior probability function.
- So, risk minimization would now look for a function in $\mathcal{H}$ that is a good approximation for the posterior probability function.

PR NPTEL course – p.95/148

Another interesting scenario in under squared error loss is you take y to be 0 1 and A to be the interval 0 comma 1. Then what happens each h is a function taking values in 0 1, so you can interpret it as a posterior probability function, right? Given x why the posterior probability of class one for x. Now, we know that under squared error loss the minimizer or the expectation of this square of error is the, is the conditional expectation and hence is the posterior probability of function.

We have already seen it when we considered the linear least squares as a minimizer of the expectation of this squared error is the posterior probability function, right? Now, we are searching over, if I take A to be 0 1 then we are searching over all possible posterior probability functions. And hence, risk minimization now we will look for a function in H which is a good approximator for the posterior probability function, right? Something like the logistic regression.

(Refer Slide Time: 42:32)

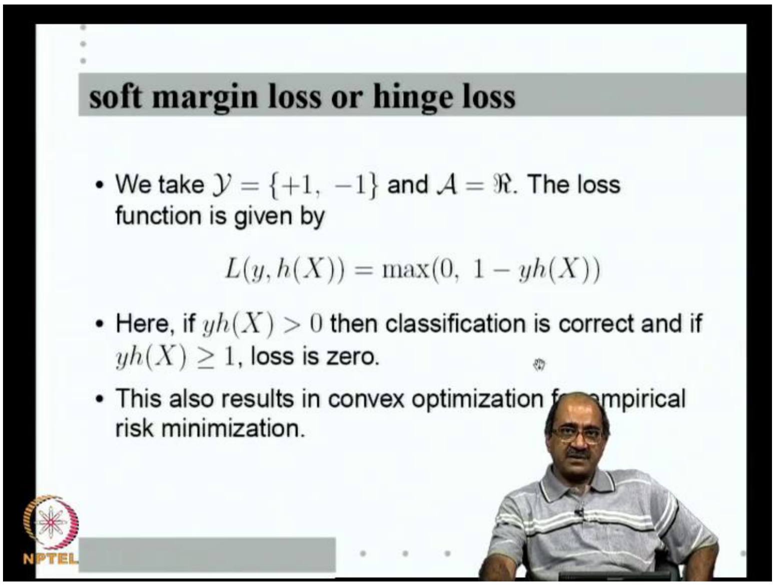


- The empirical risk minimization under squared error loss is a convex optimization problem for linear models (when h is linear in its parameters).
- The squared error loss is extensively used in many learning algorithms.

PR NPTEL course – p.97/148

Empirical risk minimization under squared error is a convex optimization turns out to be convex optimization problem for linear models that we have already seen. The linear least squares a convex optimization problem. This squared error loss is also a very often used in many learning algorithm. For example, much in neural network Bayes classifiers learning is empirical risk minimization under squared error loss.

(Refer Slide Time: 42:57)



soft margin loss or hinge loss

- We take $\mathcal{Y} = \{+1,\ -1\}$ and $\mathcal{A} = \Re$. The loss function is given by

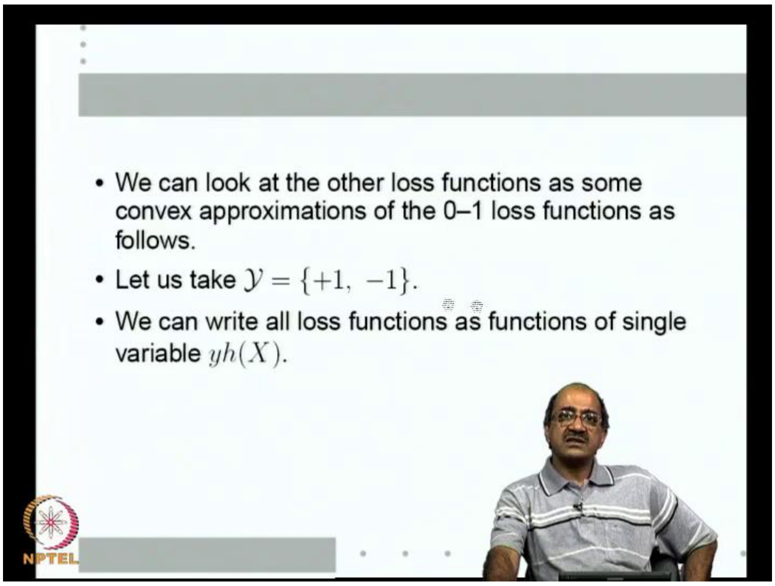
$$L(y, h(X)) = \max(0,\ 1 - yh(X))$$

- Here, if $yh(X) > 0$ then classification is correct and if $yh(X) \geq 1$, loss is zero.
- This also results in convex optimization for empirical risk minimization.

It is another loss function I called it soft margin loss or hinge loss. We will come across this when we do what are known as support vector machines, we will briefly consider that in the entire three classes. So, for here we take y to be plus 1 minus 1 and A as R, so each of my h is a real valued function and loss of y comma h X is given as max of 0 comma 1 minus y times h X. Essentially, because I take sign of h X to be the classification and I am taking y to be plus 1 minus 1. If y into h X is positive right, then my classification is correct and y into h X is negative, my classification is wrong. So, if y into h X is greater than 0 then classification is correct.

But even if y into h X is greater than 0, I may still suffer some loss. Till y into h X reaches 1, the loss is not 0. y h X greater than 1 the loss is 0, y h X less than 1 the loss is not 0. Of course, y h X less than 0 also loss is not 0, but y h X is greater than 0 is this classification is correct. This also is an interesting loss function and results in a convex optimization problem for empirical risk minimization.

(Refer Slide Time: 44:11)

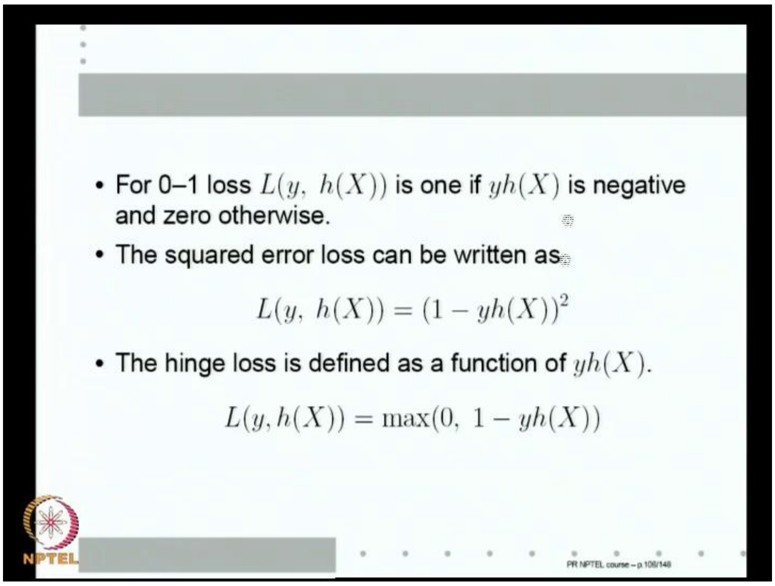


- We can look at the other loss functions as some convex approximations of the 0–1 loss functions as follows.
- Let us take $\mathcal{Y} = \{+1, \ -1\}$.
- We can write all loss functions as functions of single variable $yh(X)$.

We considered three different loss functions, two of them we have already seen earlier and one which we have not seen earlier. And we can look at the other two loss functions like many others that we come across later on. As convex approximations of the 0 1 loss function, right? So, look at this what we what we will do is, we take y to be plus 1 minus 1 for a two class problem and write all loss functions as functions of a single variable which is denoted by y into h X, because even we in general take A is equal to R so h X is are real valued function.

But because you are using sign of h X as a classifier. If y into h X is positive then y into h X are the same sign is negative y into h X have different sign. So, y into h X as a variable denotes both correctness of classification and any magnitude that you’re talking about. So, you can think we can rewrite all loss functions, function of a single variable y into h X.

(Refer Slide Time: 45:05)



- For 0–1 loss $L(y,\ h(X))$ is one if $yh(X)$ is negative and zero otherwise.
- The squared error loss can be written as

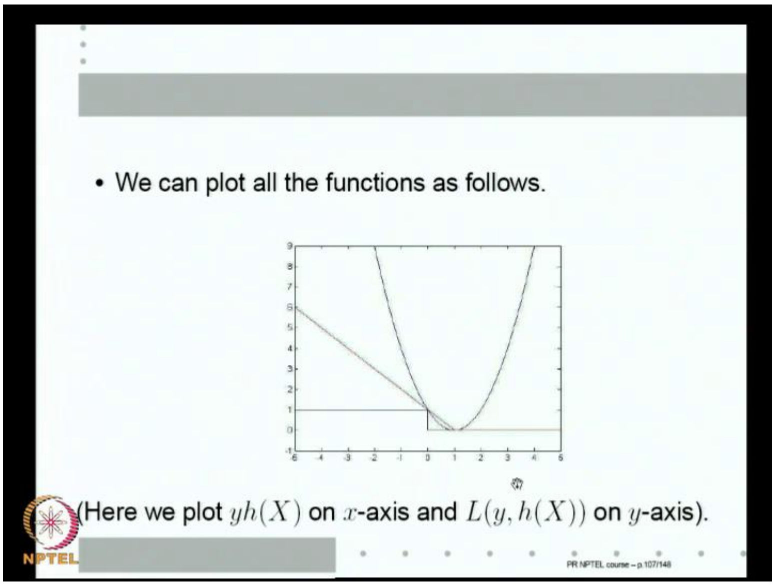
$$L(y,\ h(X)) = (1 - yh(X))^2$$

- The hinge loss is defined as a function of $yh(X)$.

$$L(y, h(X)) = \max(0,\ 1 - yh(X))$$

If I write that for the 0 1 loss L y into h X is 1, if y times h x is negative and 0 otherwise. If it is negative then classification is it correct? So, loss is 1. If it is positive the classification is correct, so loss is 0. This squared error loss can all almost also be written as 1 minus y into h X whole square. If y is plus 1 then this is anyway true. If y is minus 1 then originally y minus h X whole square is minus 1 minus h X whole square, which is same as 1 plus h X whole square which is same as 1 minus y h X whole square. So, square error loss can also be written like this and this soft margin or hinge loss is anyway written as a function of y times h X. Now, because all three three functions are function of one variable we can plot them, right?

(Refer Slide Time: 45:51)



- We can plot all the functions as follows.

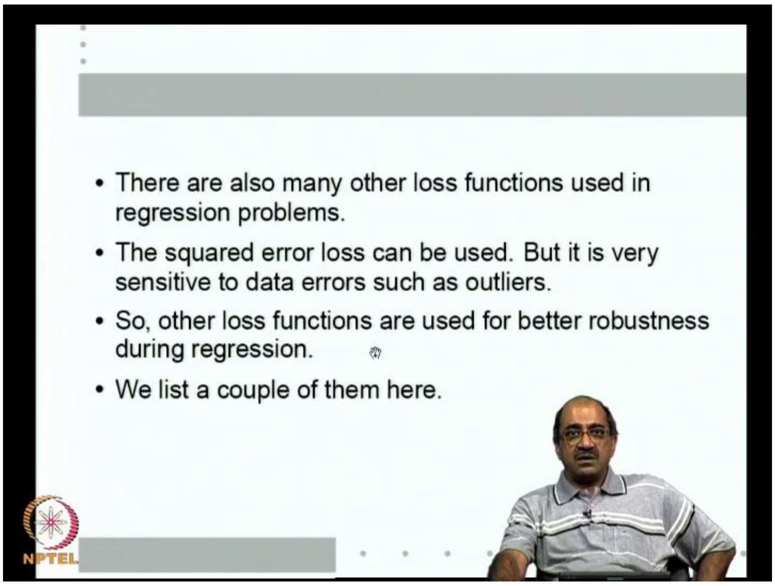
(Here we plot $yh(X)$ on x-axis and $L(y, h(X))$ on y-axis).

PR NPTEL course – p.107/148

So, that is the step function is the 0 1 loss this line right. It see my hinge loss is 1 minus x. So, for x less than one it grows like this. x that y into h X greater than 1, it is 0. So, did not mark the axis here but on the x axis I am plotting y into h X and y axis I am plotting L y comma h X, right? So, that is the hinge loss and this parabola is the familiar squared error loss. And we are looking at, if I look at this 0 1 loss function, this is non convex. Because if I take a point there and a point here and join them. So, for a convex function if I join any two point points on the curve the entire curve between them is below the straight line.

But here if I take a point here and I take a point then join them right, part of the curve is above the line part of the curve is below the line so this 0 1 loss function is non convex. So, you can think of the the hinge loss or the squared error loss as kind of approximating this 0 1 loss function so to say but convexifying it, right? So, for example the hinge loss tries to come up till here and then push it like that so that it becomes convex. So, all the other loss functions all the other two loss functions here can be thought of as trying to find a convex approximation to the 0 1 loss function. And essentially that is why minimizing empirical risk under these loss functions is much easier than under the 0 1 loss function.

(Refer Slide Time: 47:35)

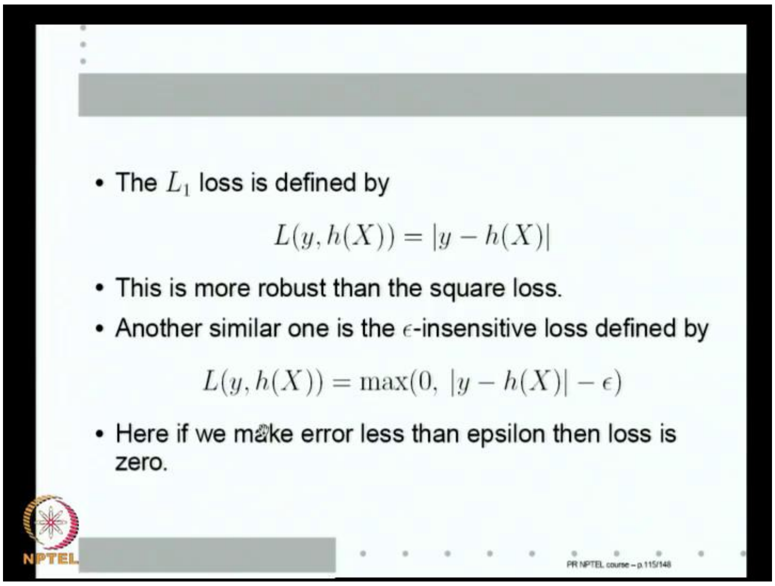


- There are also many other loss functions used in regression problems.
- The squared error loss can be used. But it is very sensitive to data errors such as outliers.
- So, other loss functions are used for better robustness during regression.
- We list a couple of them here.

These are used for classification mostly what we have seen though these squared error loss is also is for regression there are also other loss functions which are used in modern regression. For regression also you can use squared error loss but, it is very sensitive to data errors. In squared error loss, the loss is y minus suppose I am I am fitting a straight line if I have got four points, three of them are closed to a straight line but fourth one is very far away.

Now, if I take the intuitively of a straight line because the fourth one is far away and I take square of the distance. The risk for it which is the sum of all the squared errors will be very large. On the other hand if I move the line towards the out layer then may be all the errors will become intermediate so the square of the errors might be much smaller. So, the squared error loss becomes very sensitive to data errors such as out layers. There are loss functions which are better from the from such robustness point of view.

(Refer Slide Time: 48:38)



- The L_1 loss is defined by

$$L(y, h(X)) = |y - h(X)|$$

- This is more robust than the square loss.
- Another similar one is the ϵ-insensitive loss defined by

$$L(y, h(X)) = \max(0, \ |y - h(X)| - \epsilon)$$

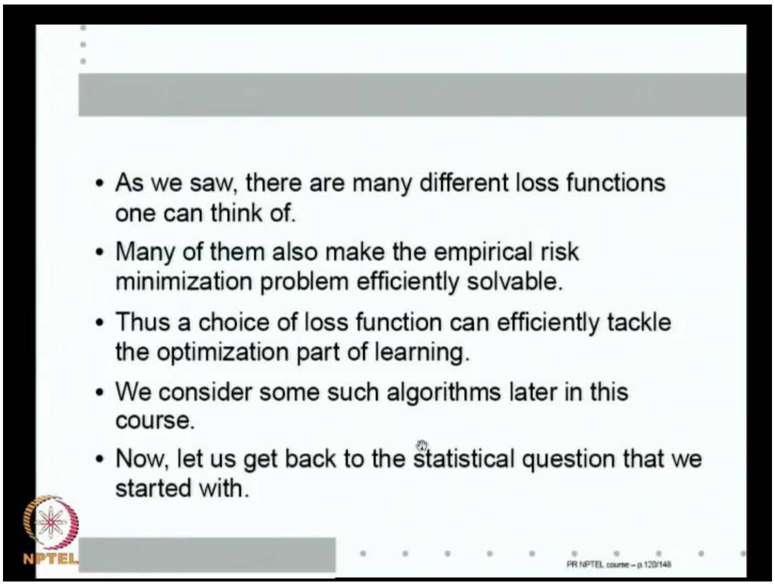
- Here if we make error less than epsilon then loss is zero.

PR NPTEL course – p.115/148

Just for completeness here are two. A simple way to make squared error more robust to out layers is instead of taking square you take absolute value. Absolute value does not grow as fast as square that is the reason why it is a little less sensitive to out layers. But obviously optimizing absolute value is a harder problem than optimizing square, okay? This is non differentiable. There is another loss function that we will once again consider when we look at a same kind of methods for regression. It is called the epsilon insensitive loss.

Essentially is once again like the hinge loss. I predict h X as the target y is the true target y minus h X absolute value is the error. If the error is less than epsilon, then I suffer 0 loss, otherwise I suffer a linearly growing loss, right? So, this is called an epsilon insensitive loss which also has some interesting robustness properties and also makes the optimization problem resulting empirical risk minimization problem a convex optimization problem.

(Refer Slide Time: 49:44)

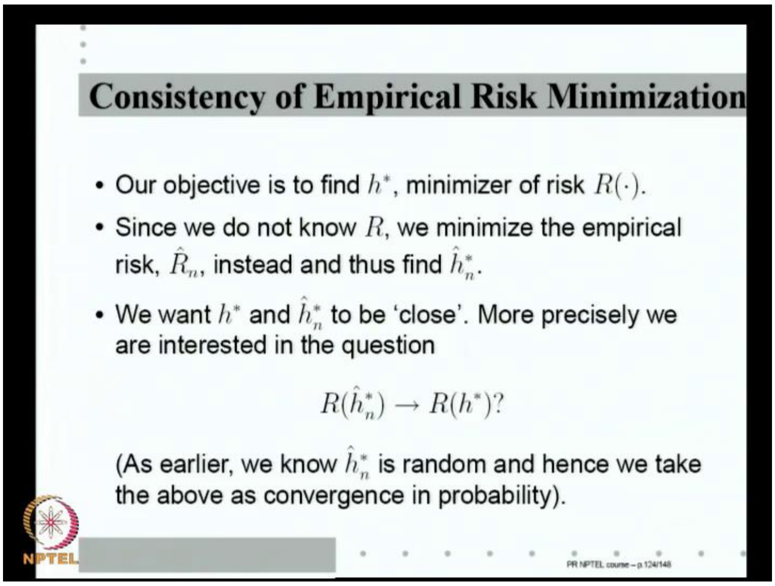


- As we saw, there are many different loss functions one can think of.
- Many of them also make the empirical risk minimization problem efficiently solvable.
- Thus a choice of loss function can efficiently tackle the optimization part of learning.
- We consider some such algorithms later in this course.
- Now, let us get back to the statistical question that we started with.

PR NPTEL course – p.120/148

So, as we said there are many different loss functions one can think of. Many of them also make the empirical risk minimization problem efficiently solvable. That is the reason why we chose them. By the choice of loss function can efficiently tackle the optimization part of learning, right? But let us get back we will see many algorithms for this later on anyway. So, let us get back to what we started with the statistical part of the learning problem, right? The consistency of empirical risk minimization.

(Refer Slide Time: 50:14)



Consistency of Empirical Risk Minimization

- Our objective is to find h^*, minimizer of risk $R(\cdot)$.
- Since we do not know R, we minimize the empirical risk, $\hat{R}_n$, instead and thus find $\hat{h}^*_n$.
- We want h^* and $\hat{h}^*_n$ to be 'close'. More precisely we are interested in the question

$$R(\hat{h}^*_n) \rightarrow R(h^*)?$$

(As earlier, we know $\hat{h}^*_n$ is random and hence we take the above as convergence in probability).

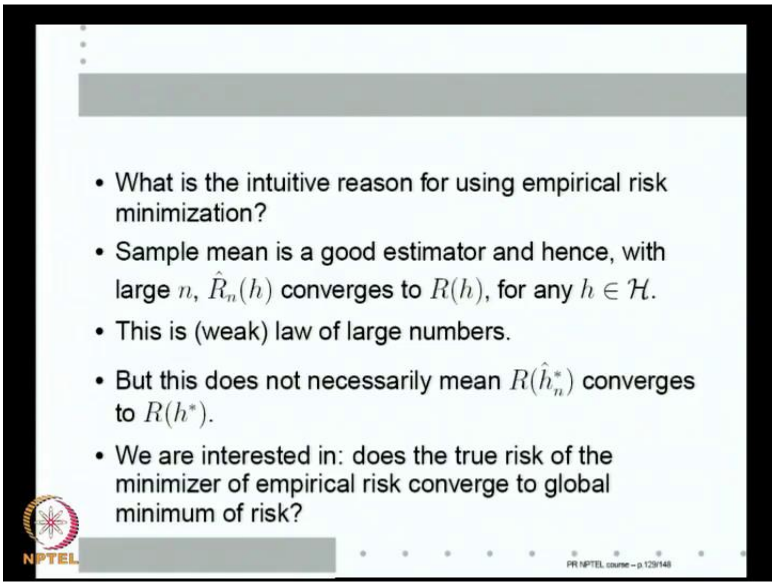
PR NPTEL course – p.124/148

So, our objective is to find h star. Let us let us once again briefly review the issue of consistency. Our objective is to find h star the global minimizer of risk. We do not know how to minimize R, we do not even know how to calculate R h for any given h. So, we mean by the empirical risk R hat n instead of R, right? Because we know how to calculate R hat and we know how to minimize it. So, because I cannot minimize R I will minimize R hat n, and hence I find h hat star n. So, we for our for our approach to work we want h R star n to be close to h star.

And, we already know what close means by our experience PAC learning. Close only means in terms of their classification accuracies on new samples and that is given by a risk. So, what we are interested in is? Is the risk of h hat star n same as risk of h star. Risk of h star of course, h star is a unknown but fixed function. So, R of h star is some constant but what I am learning is h hat star n. So, I am asking is the true risk which I cannot calculate but, is the true risk of h hat star n does it converge to h star? The convergence is as n tends to infinity.

Of course, like in the case of PAC learning we know h hat star n is random, because it depends on the random sample, right? So, because of that R h hat star n is a sequence of random variable. So, we have to define in what sense this convergence is and like in the PAC case we take convergence to be in probability. But essentially the question is does the risk of the true risk of the minimizer of the empirical risk does it converge to the global minimum of true risk

(Refer Slide Time: 52:04)



- What is the intuitive reason for using empirical risk minimization?
- Sample mean is a good estimator and hence, with large n, $\hat{R}_n(h)$ converges to $R(h)$, for any $h \in \mathcal{H}$.
- This is (weak) law of large numbers.
- But this does not necessarily mean $R(\hat{h}_n^*)$ converges to $R(h^*)$.
- We are interested in: does the true risk of the minimizer of empirical risk converge to global minimum of risk?

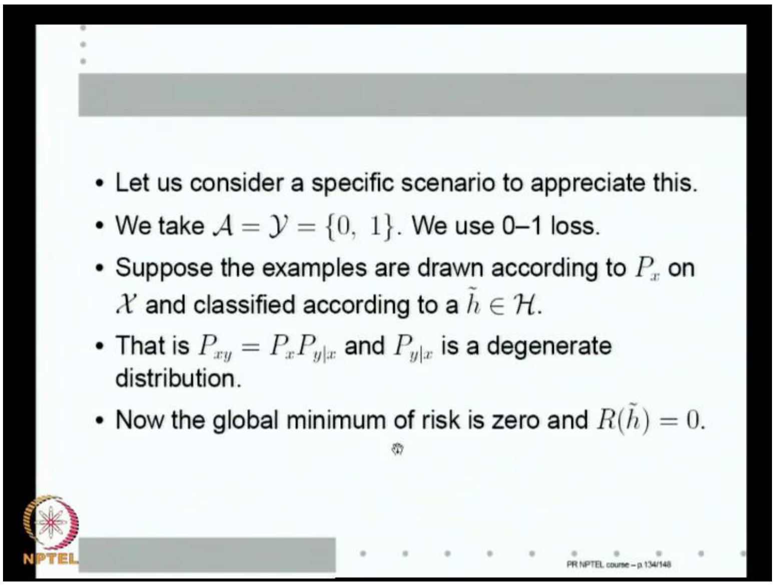
PR NPTEL course – p.129/148

That is what consistence of empirical risk minimization about. What is the reason why we think empirical risk minimization works? Sample mean is a good estimator and empirical risk is simply sample mean estimator of true risk. So, for any given h. R hat n h converges to R h if n is large. So, for every h whether I calculate R hat n h or R h it should not make much of a difference, right?

This is the law of large numbers. Because we only want convergence probability is weak law of large numbers. Weak law of large numbers says that the sample mean converges to the expectation of the random variable. So, for any h, R hat n h as n grows will become very close to R of h. Because the true risk of any hypothesis can be approximated by its empirical risk like this. It stands to reason that, the if I minimize this it should be as minimising this. But, this convergence does not mean that R of h hat star n converges to R h star. It only says R hat n of h converges to R of h that is the same h. Here, I am not worried about R hat n.

But I am asking R of some other sequence of random variable h hat star n. Does that converge to R of h star? So, a priory there is no reason why is the law of large numbers should imply this, right? Our reason intuitive reason for preferring empirical risk minimization is law of large numbers in our mind. We know that for sufficiently large sample R hat n of h will be close to R of h and hence it is all right minimized empirical risk. But we also know that this convergence says nothing about whether R of h hat star n converges to R of h star. So, what we are interested in is that the risk of the minimizer of empirical risk, that the true risk of the minimizer of empirical risk. Does it converges to the global minimum of true risk?

(Refer Slide Time: 54:04)



- Let us consider a specific scenario to appreciate this.
- We take $\mathcal{A} = \mathcal{Y} = \{0, 1\}$. We use 0–1 loss.
- Suppose the examples are drawn according to P_x on $\mathcal{X}$ and classified according to a $\tilde{h} \in \mathcal{H}$.
- That is $P_{xy} = P_x P_{y|x}$ and $P_{y|x}$ is a degenerate distribution.
- Now the global minimum of risk is zero and $R(\tilde{h}) = 0$.

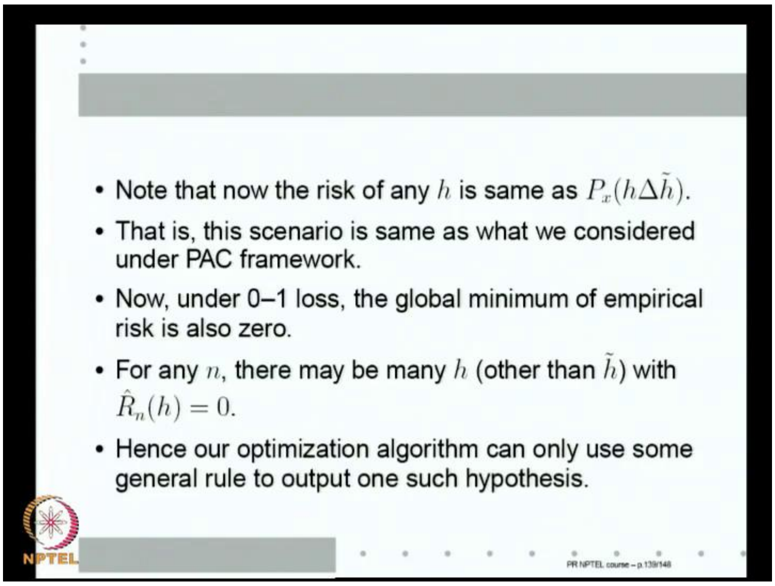
PR NPTEL course – p. 134/148

Let us consider the familiar scenario. The examples we have seen in last class to appreciate this a little more. Let us take A and y to be zero one. Let us suppose we use zero one loss function and example, let us assume that the examples are actually generated by drawing them from x using a P x and then classifying using a particular function, let us call it h delta and H. In general I want P x y but this only means that I am taking P x y to be P x into P y given x and P y given x is degenerated distribution. As we have already seen doing this is within our framework.

So, suppose examples are drawn according to some P x on x and classified using a h delta. Now, I am giving 0 1 loss function, 0 1 loss function’s risk is probability of misclassification and every example is of course classified according to h delta, right? So, the global minimum of true risk is 0 and one of the elements of h is that achieve this global minimize h delta. Of course, there might be others as we said but we settled on R of h delta is 0. So, and global minimum of risk is 0. So, we will be happy learning any h, such that R of h any h hat star n such that R of h hat star n ultimately goes to zero as n tends to infinity. That is what we want to do in this scenario. Let us ask is that feasible?

Because of this case, my risk is the expectation with respect to P x y but P x y is essentially only P x is random.

(Refer Slide Time: 55:40)



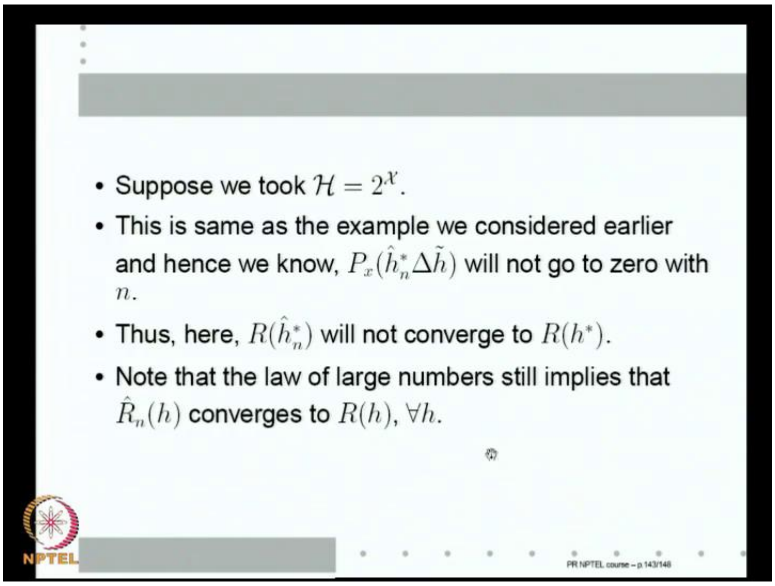
- Note that now the risk of any h is same as $P_x(h \Delta \tilde{h})$.
- That is, this scenario is same as what we considered under PAC framework.
- Now, under 0–1 loss, the global minimum of empirical risk is also zero.
- For any n, there may be many h (other than $\tilde{h}$) with $\hat{R}_n(h) = 0$.
- Hence our optimization algorithm can only use some general rule to output one such hypothesis.

PR NPTEL course – p.139/148

So, my risk for under 0 1 loss function, so it only tell ask me risk of any h is the P x probability of h delta h delta. As we have already seen if we take P x y to be actually P x and P y given as a degenerate then our new framework is same as PAC framework. So, the risk under 0 1 loss function of any h is same as P x of h delta h h delta. That is, this scenario is roughly same as what not roughly actually same as what we considered under the PAC framework.

Now, under 0 1 loss the global minimum of empirical risk is also zero. Because, h delta is in h, I am searching over h there might be others. But, certainly I still that gives me zero value for empirical risk under any samples. So, the global minimum of empirical risk is also 0. Of course, for any given n there may be many h other than h delta with R hat n of h is zero. Now, what is my optimization algorithm to do? There are many, many h’s so I have to choose any one. So, my optimization algorithm chooses some general rule to output such a hypothesis, right? Like for example, the smallest h as we have considered in our example last time.

(Refer Slide Time: 56:56)



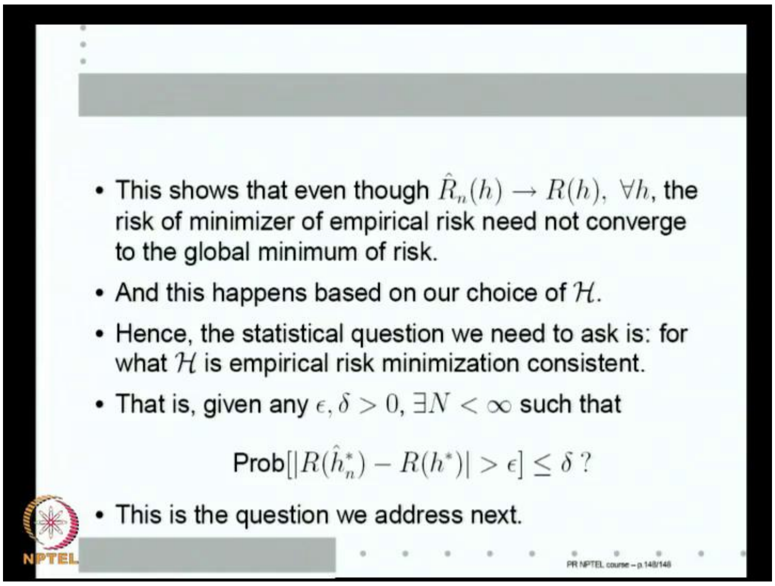
- Suppose we took $\mathcal{H} = 2^{\mathcal{X}}$.
- This is same as the example we considered earlier and hence we know, $P_x(\hat{h}_n^* \Delta \tilde{h})$ will not go to zero with n.
- Thus, here, $R(\hat{h}_n^*)$ will not converge to $R(h^*)$.
- Note that the law of large numbers still implies that $\hat{R}_n(h)$ converges to $R(h), \forall h$.

NPTEL
PR NPTEL course – p.143/148

So now, we are close we are same as our previous example. Say if I take h to be 2 power x, then this is same as the example we have considered earlier. And hence we know that h hat star n delta h delta P x of probability of this will not go to 0 with n, right? That is what we showed last time in the PAC example. Thus in this scenario my minimize of the empirical risk the two risk of the minimize of the empirical risk will not converge to the global minimum of R h, right?

Of course, there is law of large numbers still holds, even in this case even if I take h equal to two power x law of larger numbers holds for any specific h, R hat n is still converges h to R of h. But that is not good enough for this to happen, right? So, we know that simply because law of large numbers guarantees that sample mean converges to population mean.

(Refer Slide Time: 57:55)



- This shows that even though $\hat{R}_n(h) \rightarrow R(h),\ \forall h$, the risk of minimizer of empirical risk need not converge to the global minimum of risk.
- And this happens based on our choice of $\mathcal{H}$.
- Hence, the statistical question we need to ask is: for what $\mathcal{H}$ is empirical risk minimization consistent.
- That is, given any $\epsilon, \delta > 0$, $\exists N < \infty$ such that

$$\text{Prob}[|R(\hat{h}_n^*) - R(h^*)| > \epsilon] \leq \delta\ ?$$

- This is the question we address next.

PR NPTEL course – p.148/148

It does not mean that the risk of the minimize of the empirical risk converges to the global minimum of risk. And as we saw in from our PAC example, it depends on the choice of h. If we choose 1 h, it happens we choose another h it does not happen. Hence, the statistical question is for what h is empirical risk minimization consistent? Consistent means given any epsilon delta greater than 0 exists a n, so that R of h hat star n minus R of h. This difference be greater than epsilon has probability less than delta. So this is the question of empirical consistency or empirical risk minimization. In the next class, we will see, how can we answer this in a sound statistical sense?

Thank you.