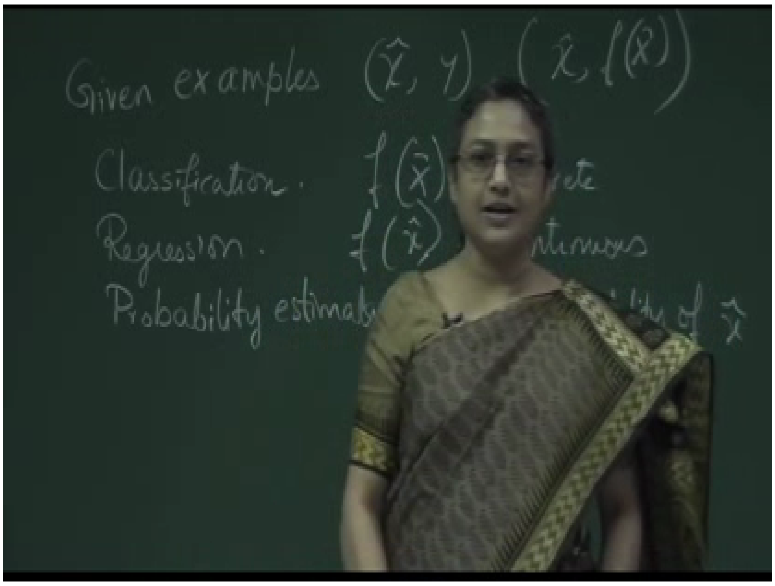
Introduction to Machine Learning

Prof. Sudeshna Sarkar
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Module - 1
Lecture - 03
Hypothesis Space and Inductive Bias

Good morning, today we will have the first module of machine learning part C. I will talk about hypothesis space and inductive bias will give you brief introduction to this, so that when we talk about a different machine learning algorithms, we can refer to this discussion.

(Refer Slide Time: 00:42)


Given examples $(\hat{x}, y)$ $(\hat{x}, f(\hat{x}))$
Classification . $f(\hat{x})$ ...rete
Regression . $f(\hat{x})$ ...tinuous
Probability estima... ...ity of $\hat{x}$

So, as we have seen that in inductive learning or prediction, we have given a examples of data. And the example are of the form as we have seen x, y, where x for a particular instance x comprises of the values of the different features of that instance; and y is the output attribute. And we can also think of that as being given x and f(x).

So, if you assume that the output of an instance is a function of the input vector input feature vector; and this is the function that we are trying to learn, we are given x, f(x) pairs as examples. And we want to learn x. For a classification problem, in the earlier class, we talked about two types of supervised learning problems - classification and regression depending on whether the output attributes type is discrete valued or continuous valued.

In classification problem, this function f(x) is discrete; in regression, the function f (x) is continuous. And we can also apart from classification and regression, in some cases we may want to find out the probability of a particular value of y. So, for those problems, where we look at probability estimation, our f(x) is the probability of x; so this is the type of inductive learning problems that we are looking at.

Why do we call this inductive learning? We are given some data and we are trying to do induction to try to identify a function, which can explain the data. So, induction as oppose to deduction, unless we can see all the instances all the possible data points or we make some restrictive assumption about the language in which the hypothesis is expressed or some bias, this problem is not well defined so that is why it is called an inductive problem.

(Refer Slide Time: 04:20)

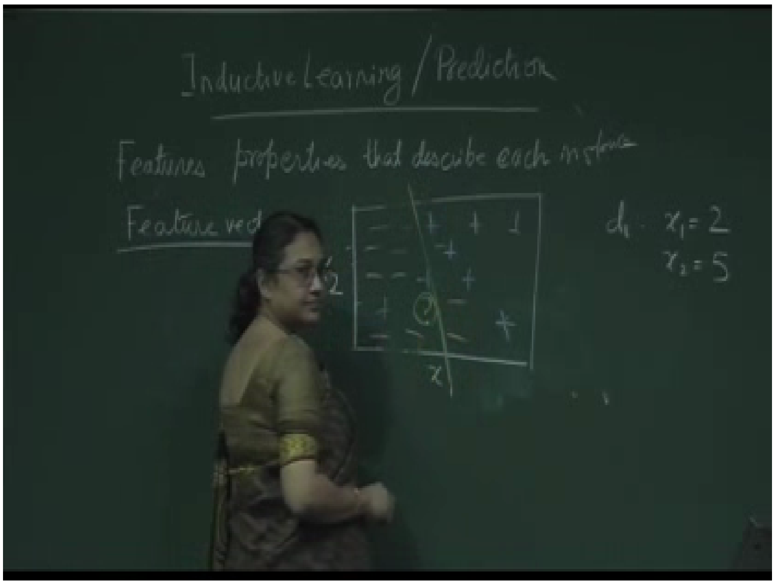
Inductive Learning / Prediction
Features properties that describe each instance
Feature vector
x_2
d_1
$d_1 \cdot x_1 = 2$
$x_2 = 5$

Then in the last class, we talked about features. So when we say we have to learn a function, it is a function of the features; so instances are described in terms of features. So, features are properties that describe each instance; and each instance can be described in a quantitative manner using features. And often we have multiple features so we have what we call a feature vector, for example, for a particular instance we may be or a particular task we may be describing all the instances in terms of ten features, so the feature vector will be a one-dimensional vector of size 10.

Now, based on this we can define a features space. Suppose, for simplicity, let us assume that there are two features; and we can say that the features are x 1 and x 2. In general, we can have n number of features we have two features the features define a two-dimensional space if we have n features the define an n dimensional space if you take a particular instance let us say d 1 is an instance and for d 1 x 1 equal to 2, x 2 equal to 5. So, let us say x 1 is 2 here and x 2 is 5 here. So, this is d 1 so d 1 can be thought of as a point in this feature space point in the two dimensional feature space or you can think of it as a vector in this space so each instance is a point in the feature space.

Now, let us look at a classification problem. And let us say that it is a two class classification problem so we have given a number of instances for examples some of which belong to class 1, the others belong to class 2, so there are two class classification problems. We have two types of instances those belonging to class 1 and those belonging to class 2. You are given a training set which comprises a subset of the instances, some of them are marked class 1, some of them are marked class 2, and we can say that class 1 is positive and class 2 as negative.

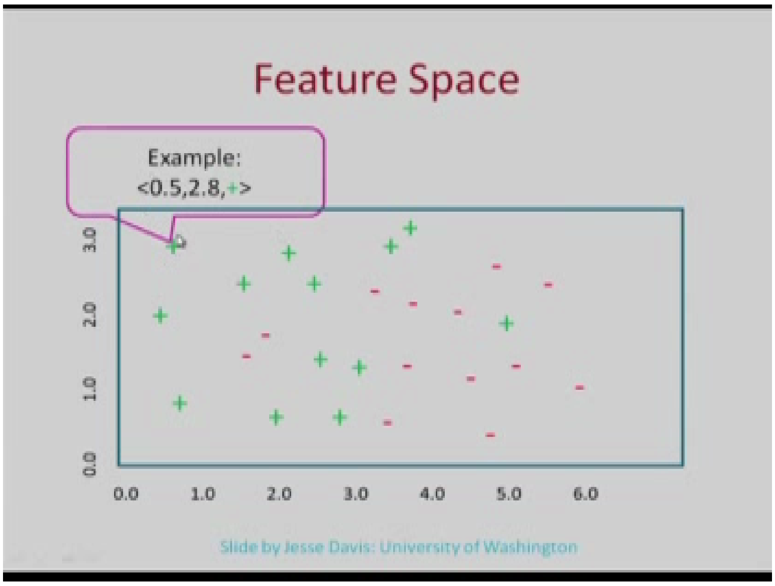
(Refer Slide Time: 07:12)


Inductive Learning / Prediction
Features properties that describe each instance
Feature vect
x_2
x
d_1 · $x_1 = 2$
$x_2 = 5$

So, we find that we can map different points in this feature space, let us say these are the positive points. And we have some other points in this feature space which are negative points. Now, what we want to do is we want to learn a function, so that based on the function, we want the function to predict whether a new instance, which is given to you. Suppose, this is a new instance, which is given to you, we want to know whether this should be positive or negative.

In order to do this, we have to learn a function or the function could be a particular curve or a line, which separates the positive from the negative instances. For example, the function that we learn could be this function. And we can say that any point which lies to this side of the function is positive any point which lies to this side of the function is negative. And since this yellow point lies to the left of the function it is negative, so this is what inductive learning is about.

(Refer Slide Time: 08:25)



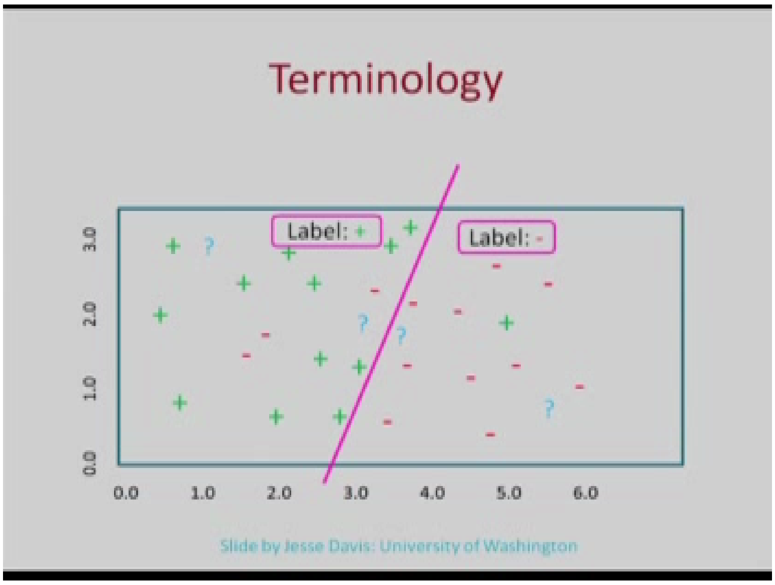
Feature Space

Example:
<0.5,2.8,+>

Slide by Jesse Davis: University of Washington

So, let us look at the slide in this slide, which I have taken from a slide by Jesse Davis of University of Washington. We can see a feature space is described in terms of the positive and negative examples. The green pluses are the positive points; the red minuses are the negative points. Now this is a particular instance, for which x 1 is 0.5, x 2 is 2.8 and the label of this instance is positive.

(Refer Slide Time: 08:55)



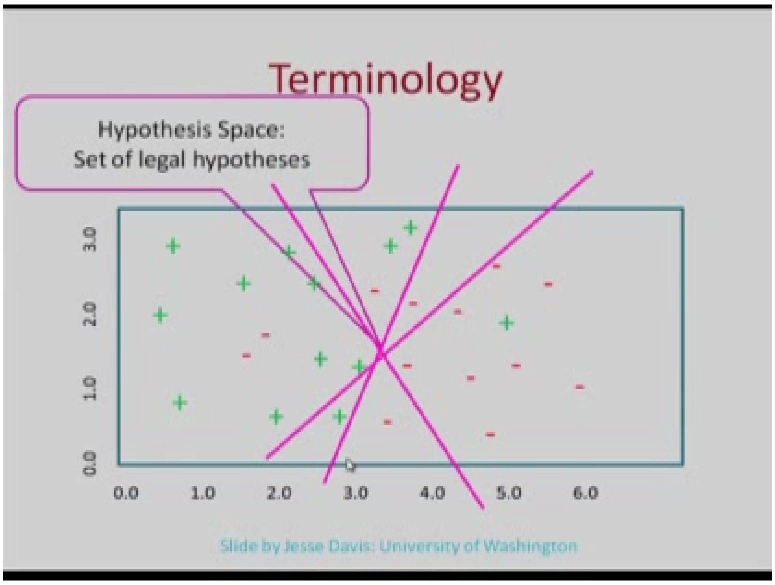
Terminology

Slide by Jesse Davis: University of Washington

Now these question mark points are the test points. And we are asked to find out what should be the class of those points may be positive or negative in the prediction problem. So, in order to answer the prediction problem we have to come up with the function, for example, let us say we come up with this pink function pink line, and we say lines points that lie to the right of the pink line is negative the points which lie to the left of the pink line is positive.

In this case, this point and this point will be marked positive; and this point and this point will be marked negative. So, this pink line is the function that we have come up with and so this is the hypothesis or function that we used to do our prediction.

(Refer Slide Time: 09:53)



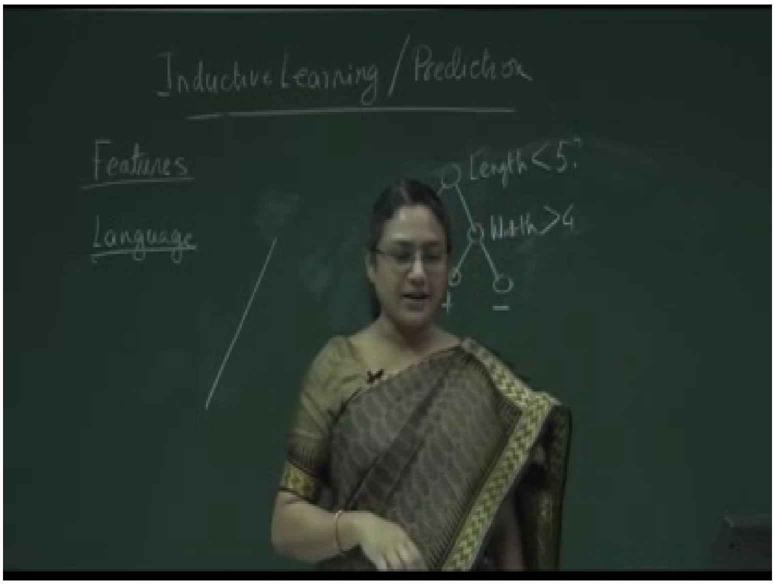
Terminology

Hypothesis Space:
Set of legal hypotheses

Slide by Jesse Davis: University of Washington

Now, we could have instead of this particular line, we could have hypothesized other functions. So, all these are possible functions, which we could have found. And the set of all such legal functions that we could have come up with they define the hypothesis space. In a particular learning problem, you first defined the hypothesis space that is the class of function that you are going to consider then given the data points, you try to come up with the best hypothesis given the data that you have.

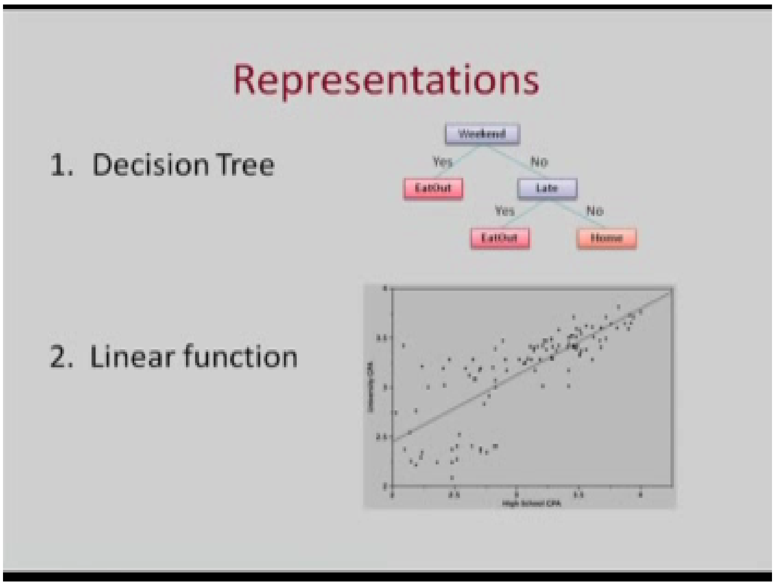
(Refer Slide Time: 11:02)



We have briefly talked about in the last module about how a function is represented so as we have discussed a function is represented in terms of features. There are two things that we need in order to describe a function, we have to decide the features of the vocabulary, and we have to decide the function class or the type of function or the language of the function that we will have to we will be using. So, based on the features and the language, we can define our hypothesis space. Various types of representations have been considered for making predictions.

For example, we just saw that we could have a linear function to act as a discriminator between two classes, we will in a subsequent class, we will look at a representation by using a structure, which we called a decision tree. Where at a decision tree is a tree, where at every node, we take a decision based on the value of an attribute. And based on that, we go to different branches, so at every node, we make a decision based on the value of an attribute and every leaf node is labeled by the value of y.

(Refer Slide Time: 12:40)



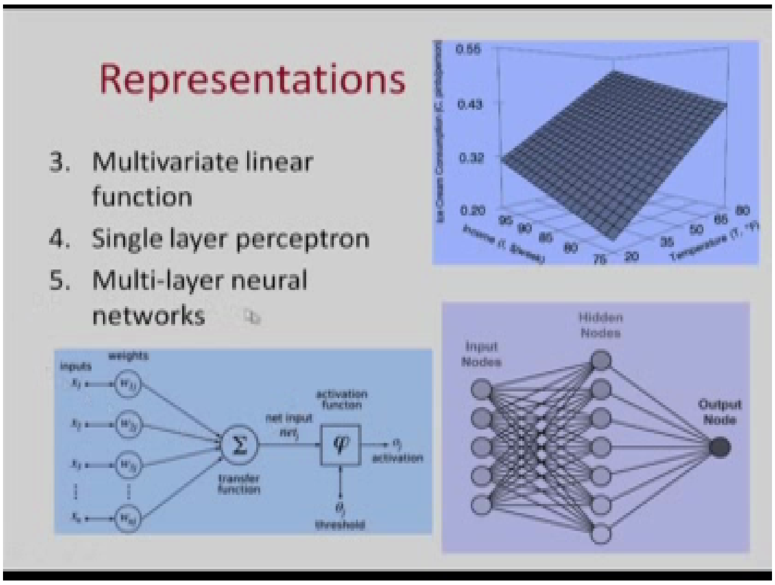
Representations

1. Decision Tree

2. Linear function

So, decision tree is a type of representation; linear function is one type of representation.

(Refer Slide Time: 12:46)

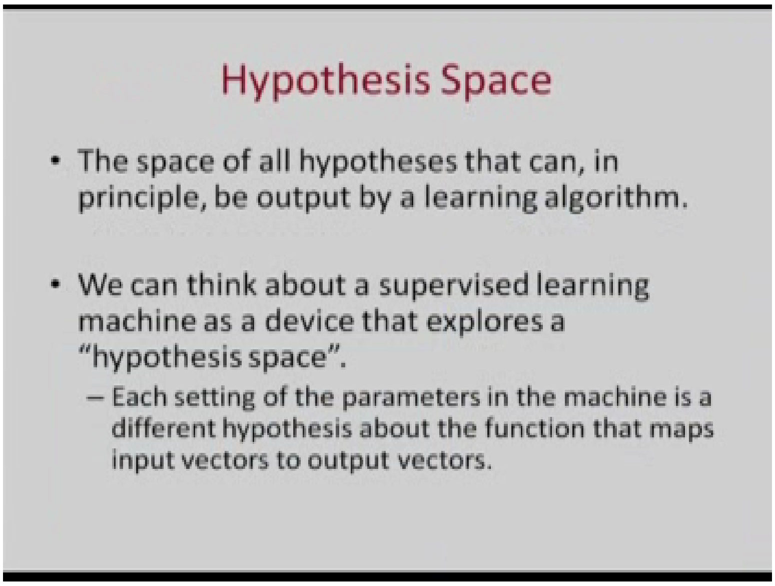


Representations

3. Multivariate linear function
4. Single layer perceptron
5. Multi-layer neural networks

You could also have multivariate linear function linear function you can have neural networks. These are some examples of representation and we have some examples in the slide here. A decision tree, a linear function, a multivariate linear function, a single layer perceptron - the basic unit of a neural network, a multi layer neural network; these are some of the representations that we will talk about later in this class.

(Refer Slide Time: 13:03)

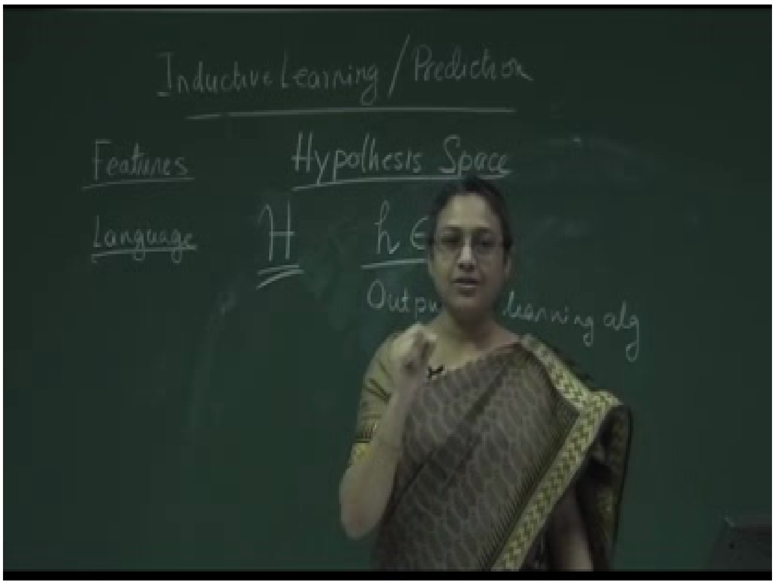


Hypothesis Space

- The space of all hypotheses that can, in principle, be output by a learning algorithm.
- We can think about a supervised learning machine as a device that explores a "hypothesis space".
  - Each setting of the parameters in the machine is a different hypothesis about the function that maps input vectors to output vectors.

So, once you have chosen the features and the language or the class of functions, what you have is a hypothesis space.

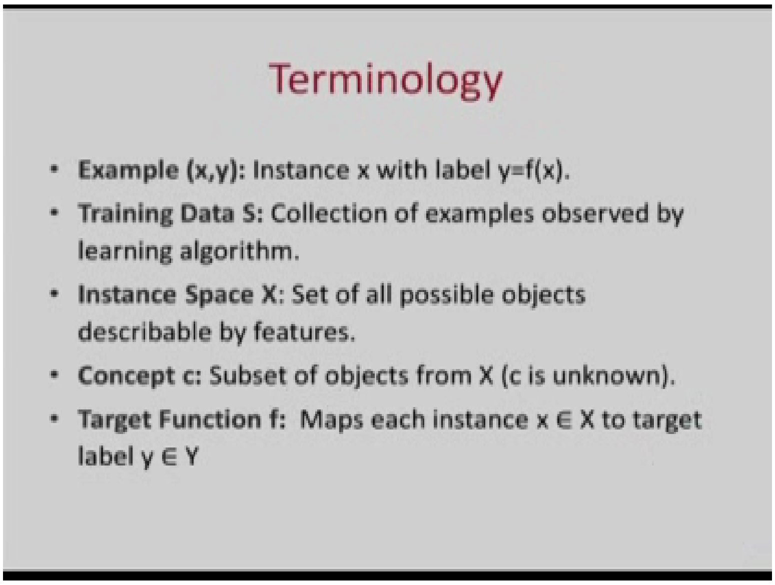
(Refer Slide Time: 13:16)



So, hypothesis space is the space of all legal hypothesis, is a set of all legal hypothesis that you can describe using the features that you have chosen, and the language that you have chosen. And this is the set from which the learning algorithm will pick a hypothesis. So, hypothesis space we may represent a hypothesis space by H and the learning algorithm outputs a hypothesis h belonging to H, this is the output of a learning algorithm. So, capital H denotes all legal hypothesis, all possible outputs by the learning algorithm.

Given the training set given the particular data points, the learning algorithm will come up with one of the hypothesis in the hypothesis space which hypothesis it comes up with will depend on the data, and it also will depend on what type of restrictions or biases that we have imposed, which we will describe later. So, supervised learning, we can think of is a device which explore the hypothesis space or which searches the hypothesis space in order to find out one of the hypothesis which satisfies certain criteria.

(Refer Slide Time: 14:58)

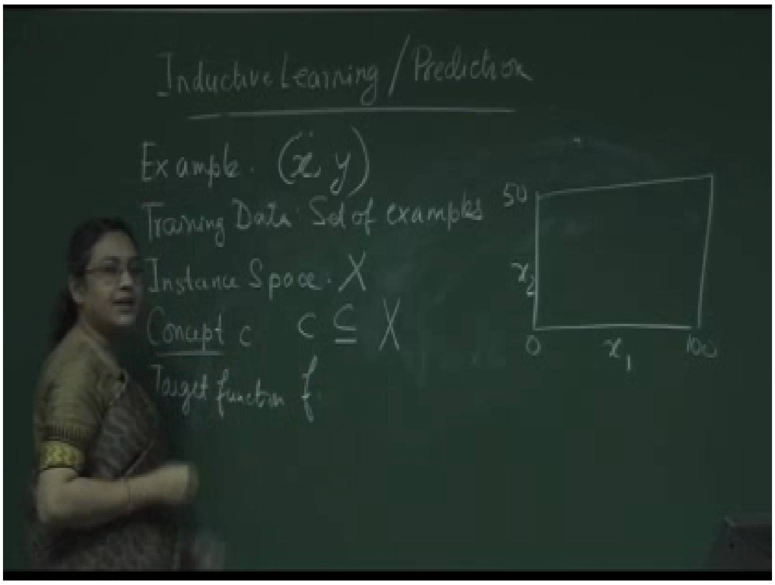


Terminology

- **Example (x,y):** Instance x with label y=f(x).
- **Training Data S:** Collection of examples observed by learning algorithm.
- **Instance Space X:** Set of all possible objects describable by features.
- **Concept c:** Subset of objects from X (c is unknown).
- **Target Function f:** Maps each instance x ∈ X to target label y ∈ Y

Now first some more terminology before we proceed.

(Refer Slide Time: 15:15)

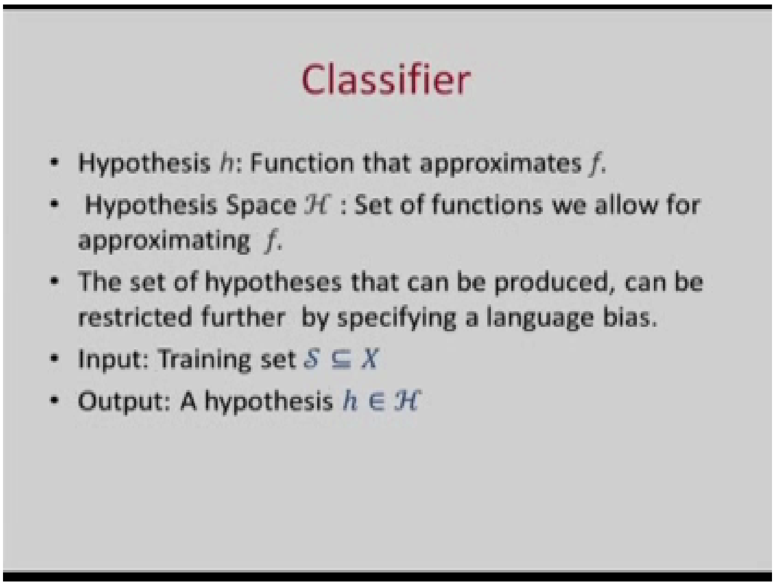


We have already talked about an example as x, y the value of the input and the value of the output x, y pair. Training data is a set of examples is a collection of a examples, which have been observed by the learning algorithm or which is input to the learning algorithm. We have instance space or feature space, which describes all possible instances, so if we have two features x 1 and x 2; let us say x 1 takes value between 0 and 100, x 2 takes value between 0 and 50; and all points in this plane can describe an instance, so this is the instance space. So, instance space is the set of all possible objects that can be described by the features.

And we are trying to learn a concept c. Let us think of a classification problem where we have a particular class that we are trying to learn. So, let us think of a two class classification problem, we can define one of the classes is positive, the other is negative, we can think of the positive examples as the concept which we are trying to learn. So, out of all possible objects that we can describe in the instance space, subsets of those objects are positive that is they belong to the concept.

So, the concept c can be a subset of the instance space X, so which define the positive points. C is unknown to us and this is what we are trying to find out. In order to find out c, we are trying to find a function f, so f is what we are trying to learn. What is f? f is a function which maps every input X to an output Y. Now what is the difference between c and f, f is used to be a function used to describe the concept they may be same, they may be different, because f is defined by the language and the features that you have chosen. So, this is a certain difference between f and c.

(Refer Slide Time: 18:06)

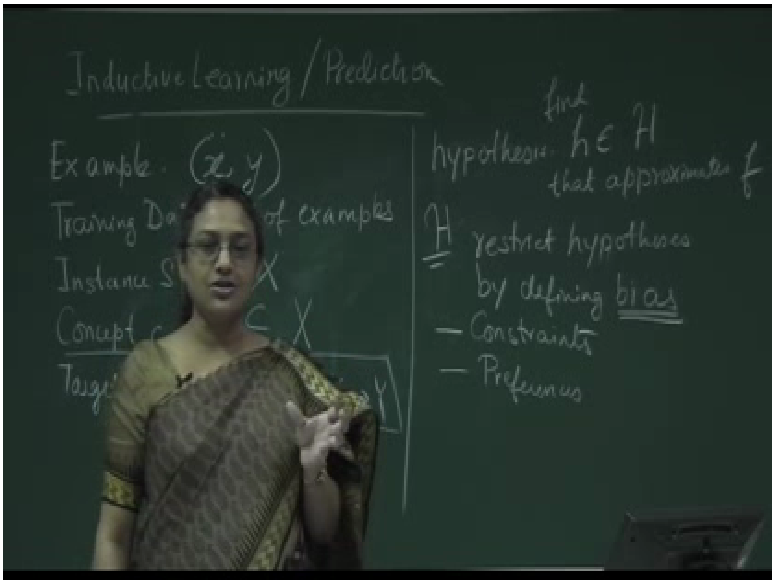


Classifier

- Hypothesis h: Function that approximates f.
- Hypothesis Space $\mathcal{H}$: Set of functions we allow for approximating f.
- The set of hypotheses that can be produced, can be restricted further by specifying a language bias.
- Input: Training set $S \subseteq X$
- Output: A hypothesis $h \in \mathcal{H}$

Now, what you are trying to do in learning is given a hypothesis space h.

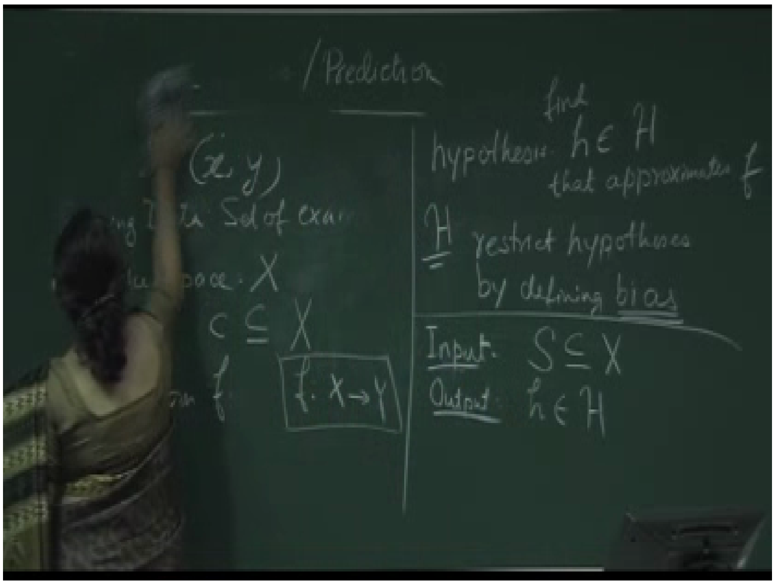
(Refer Slide Time: 18:18)


Inductive Learning / Prediction
Example (x, y)
Training Da... of examples
Instance S... X
Concept c... X
Targ... Y
hypothesis: find h ∈ H that approximates f
H restrict hypotheses by defining bias
— Constraints
— Preferences

You are trying to come up with the hypothesis small h belonging to the hypothesis H that approximates f. You want to find h that approximates f based on the training data that you have been given. Now the set of hypothesis that can be produced can be restricted further by specifying a language bias. So, hypothesis space defines all possible set of hypothesis, you can restrict hypothesis by defining some bias. So, you can specify some constraints on the language or some preferences.

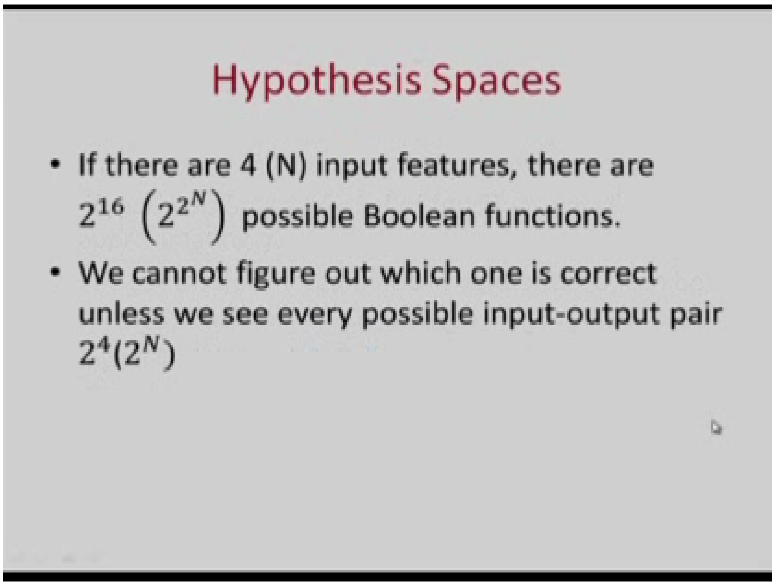
So, bias is of two types, bias can be in terms of constraints or the bias can be in terms of preferences. We will define them more precisely soon, but what we mean by constraints is suppose your features are Boolean variables, now if you say that you want to consider only Boolean functions, which are conjunctions of monomials, so that is providing a bias or the language. If you say that you want a function, which is simpler then you are putting a preference bias. So, we will talk about this later.

(Refer Slide Time: 20:23)


/Prediction
(x, y)
Data: Set of exa
space: X
c ⊆ X
f
f: X→Y
hypothesis: find h ∈ H that approximates f
H restrict hypotheses by defining bias
Input: S ⊆ X
Output: h ∈ H

So, given these definitions in a learning problem, the input is a training set let us say S, S is a subset of the instance space, X is the instance space, which comprises of all possible instances and the training examples that you are given is a subset of this. And output, you are required to output, a hypothesis small h belonging to the hypothesis space capital H, so this is for a classification problem. So, let me rub the board before we proceed.

(Refer Slide Time: 21:27)

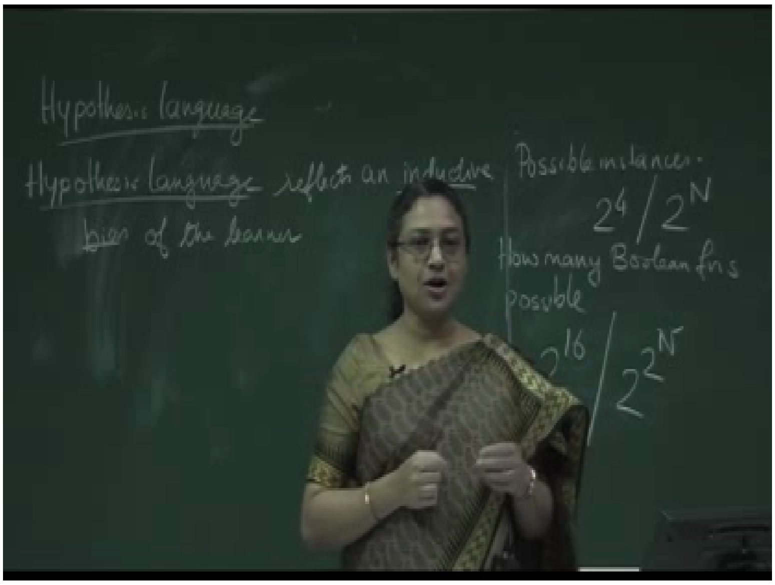


Hypothesis Spaces

- If there are 4 (N) input features, there are 2^{16} $\left(2^{2^N}\right)$ possible Boolean functions.
- We cannot figure out which one is correct unless we see every possible input-output pair $2^4(2^N)$

Now, let us look at hypothesis space. Now suppose we look at functions just to take an example.

(Refer Slide Time: 21:35)


Hypothesis language
Hypothesis language reflects an inductive bias of the learner
Possible instances -
$2^4 / 2^N$
How many Boolean fns possible
$2^{16} / 2^{2^N}$

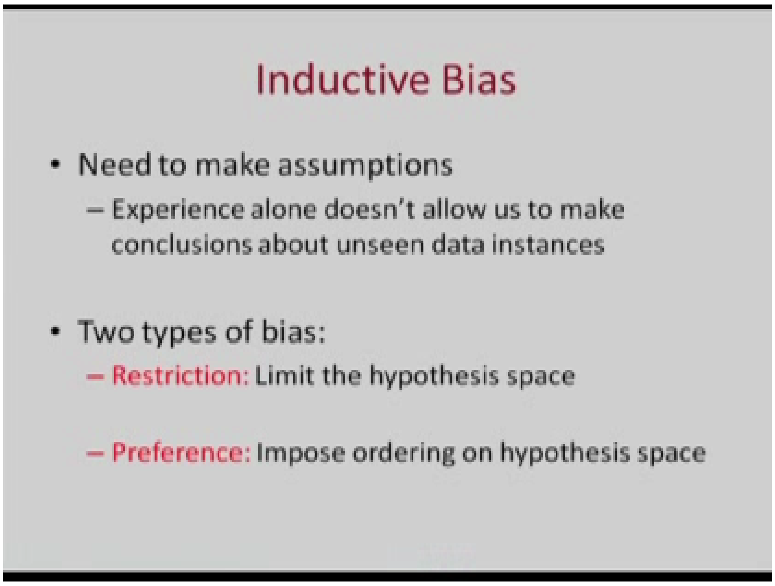
We take features which are Boolean. Suppose, x 1, x 2, x 3, x 4 are four features, and they are Boolean features the value of the features are true or false. Now, if there are four Boolean features, and how many possible instances can you have, in a particular instance x 1 can be true or false 0 or 1, x 2 can be 0 or 1, x 3 can be 0 or 1, x 4 can be 0 or 1. So, there is 1 to the power 4 or 16 possible instances. So, number of possible instances is 2 to the power 4 or 16.

Now how many possible function are there, how many Boolean functions are possible. So, what is a function, a function will classify some of the points as positive others as negative out of the 16 points, so that means the number of functions is the number of possible subsets of this 16 instances. So, how many possible subsets are there, there are 2 to the power 16 subsets or 2 to the power 2 to the power 4 subsets.

Instead of 4 Boolean variables as feature, if you had N Boolean features, then the number of possible instances will be 2 to the power N. And number of possible function will be 2 to the power 2 to the power N. So, this is the size of the hypothesis space. As you can see the hypothesis space is very large, and it is not possible to look at every hypothesis individually in order to select the best hypothesis that you want.

So, what do you do you put some restrictions on the hypothesis space, you can put some restrictions. So, you select a hypothesis language. So, this hypothesis language may be an unrestricted language, for example, all possible Boolean functions or may be a restricted language. We have seen already some examples of hypothesis languages as decision tree, linear functions, neural networks etcetera or there could be polynomial function, linear function, or there could be conjunction Boolean formulas, CNF Boolean formulas, unrestricted Boolean formulas so you choose a hypothesis language. The hypothesis language if you restrict the hypothesis language, the hypothesis language reflects bias, so this reflects a bias or inductive bias of the learner.

(Refer Slide Time: 25:20)



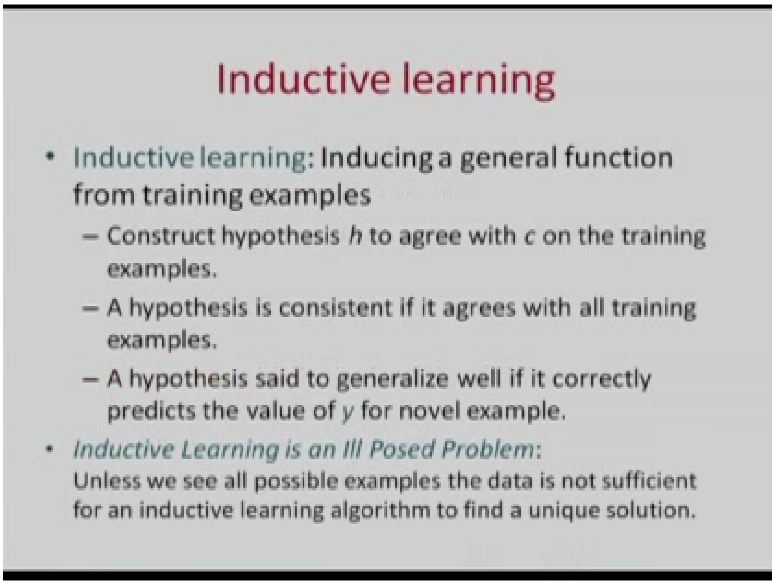
Inductive Bias

- Need to make assumptions
  - Experience alone doesn't allow us to make conclusions about unseen data instances
- Two types of bias:
  - Restriction: Limit the hypothesis space
  - Preference: Impose ordering on hypothesis space

Now, so let us define formally what is inductive bias. So, when we choose a hypothesis space, we need to make some assumptions. And there as I said there are two types of assumptions that you can make. You can put restrictions on the type of functions that is you can say instead of considering all Boolean formulas, we are going to consider only conjunctive Boolean formulas. You can say that for regression problem, you can say that we are looking at linear functions, or you can say that we can look at fourth degree polynomials or nth degree polynomials or we can say we look as any polynomial. So, specifying the form of the function is called restriction bias.

The second type of bias that you can use is preference bias, where given a particular language that you have chosen you say that I am considering all possible polynomials, but I will prefer polynomials of lower degree. So, you can say that I am considering all possible Boolean functions, but I want a Boolean function which can be described in small size. So, you can put different types of bias on your learning algorithm.

(Refer Slide Time: 26:51)



Inductive learning

- Inductive learning: Inducing a general function from training examples
  - Construct hypothesis *h* to agree with *c* on the training examples.
  - A hypothesis is consistent if it agrees with all training examples.
  - A hypothesis said to generalize well if it correctly predicts the value of *y* for novel example.
- *Inductive Learning is an Ill Posed Problem*: Unless we see all possible examples the data is not sufficient for an inductive learning algorithm to find a unique solution.

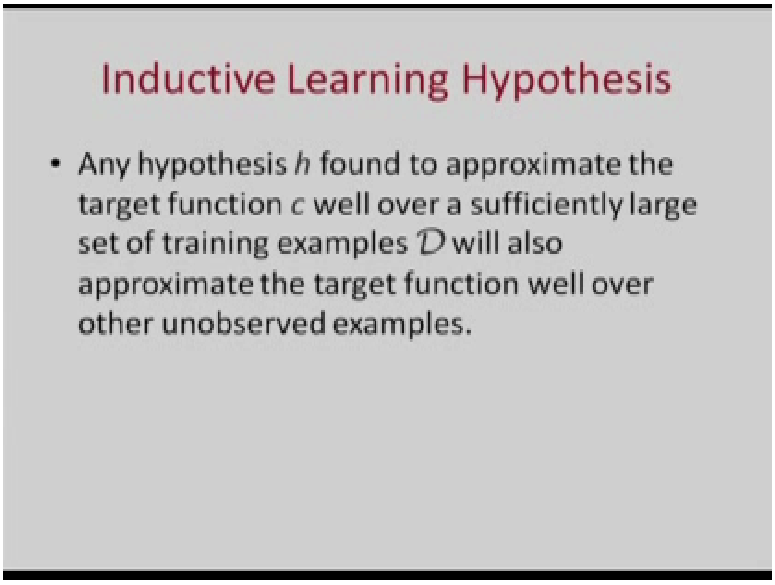
So, inductive learning means to come up with the general function from training examples. Given some training examples, you want to generalize. So, you construct a hypothesis h, you are given some training examples which comes from a concept c, and you want to find out a hypothesis h. You can come up with the hypothesis that is consistence with all the training examples given, then such hypothesis are called consistence hypothesis; it is sometimes not possible to come up with the consistence hypothesis and sometimes we will not come up with the consistence hypothesis.

But even when you are coming up with the consistence hypothesis, given a hypothesis space and given a training data multiple possible consistence hypothesis can be there, and you have to select which one of them you want to output based on your preference bias.

The hypothesis that you want to output is most often, you are guided by you want to come up with the hypothesis that generalizes well over the unseen examples, you form your hypothesis based on the training data. But you want come up with the hypothesis that does not just do well on the training data, but is likely to do well on unseen data.

Now inductive learning is an ill post problem. If you do not look at all, suppose, your hypothesis space is all Boolean formulas, and if you do not look at all the 2 to the power N possible examples if you look at a subset of those examples multiple possible hypothesis is possible, and they have they will behave differently with the rest of the examples. So, you cannot come up with the correct hypothesis by logical being by you know which is which is guaranteed to be true without seeing all the training examples. So, inductive learning is a ill post problem, you are looking for generalization guided by some bias or some criteria.

(Refer Slide Time: 29:15)

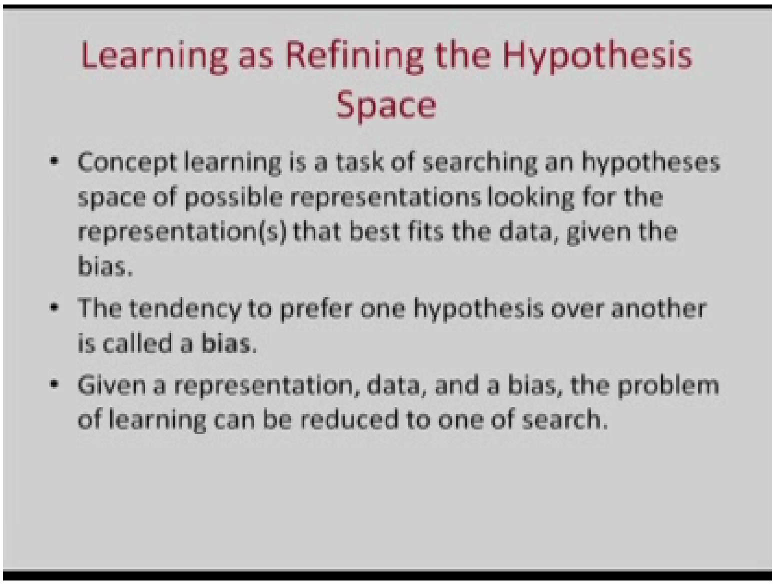


Inductive Learning Hypothesis

- Any hypothesis *h* found to approximate the target function *c* well over a sufficiently large set of training examples $\mathcal{D}$ will also approximate the target function well over other unobserved examples.

So, why you are being able to generalize, it is based on a assumption we call this assumption the inductive learning hypothesis. The hypothesis states that a hypothesis h is found to approximate the target function c well over a sufficiently large set of training examples. So, if you come up with a hypothesis which has a low training error over a sufficiently large training set you expect that hypothesis to do well on unseen examples. So, this is the inductive learning hypothesis.

(Refer Slide Time: 29:59)

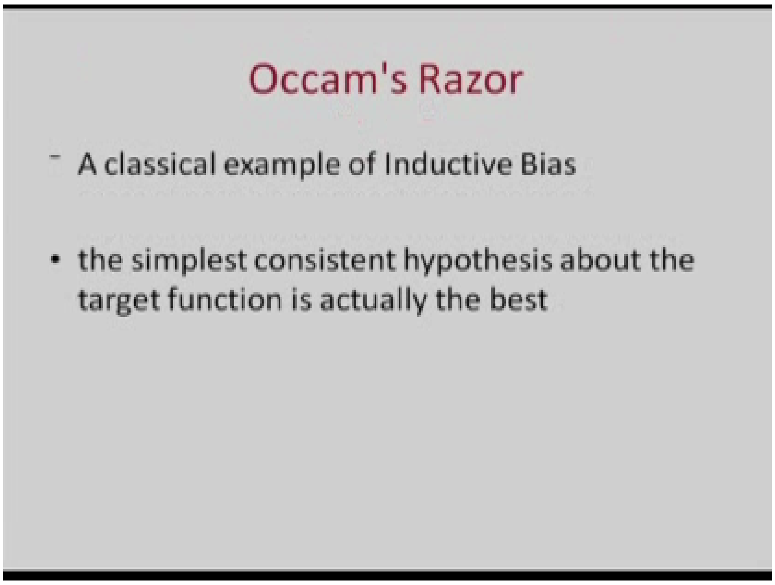


Learning as Refining the Hypothesis Space

- Concept learning is a task of searching an hypotheses space of possible representations looking for the representation(s) that best fits the data, given the bias.
- The tendency to prefer one hypothesis over another is called a bias.
- Given a representation, data, and a bias, the problem of learning can be reduced to one of search.

And learning can be looked upon as searching through the hypothesis space.

(Refer Slide Time: 30:04)

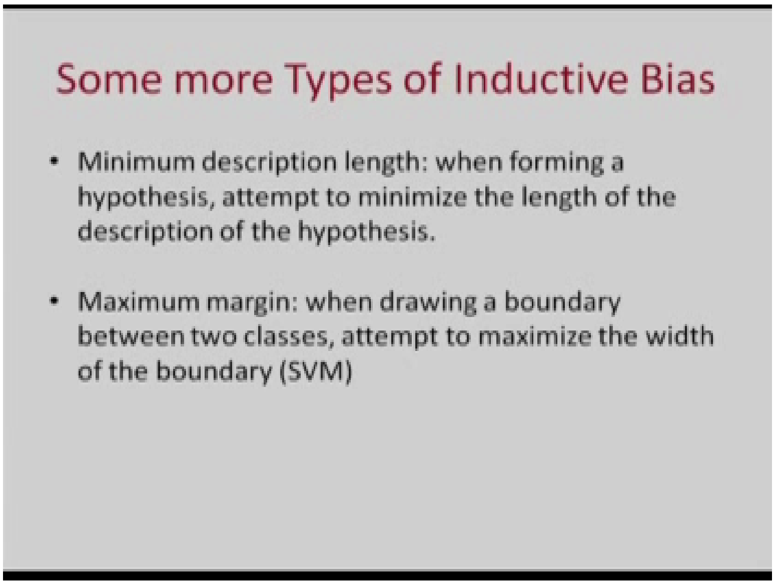


Occam's Razor

- A classical example of Inductive Bias
- the simplest consistent hypothesis about the target function is actually the best

Based on the training examples and the bias that you have imposed, there are different types of bias for example, one classical bias is a bias called Occam's Razor. Occam's razor states that you will prefer the simplest hypothesis. So, this is a principle or this is a philosophical principle that if something can be described in a short language that hypothesis is to be preferred over a more complex hypothesis.

(Refer Slide Time: 30:40)

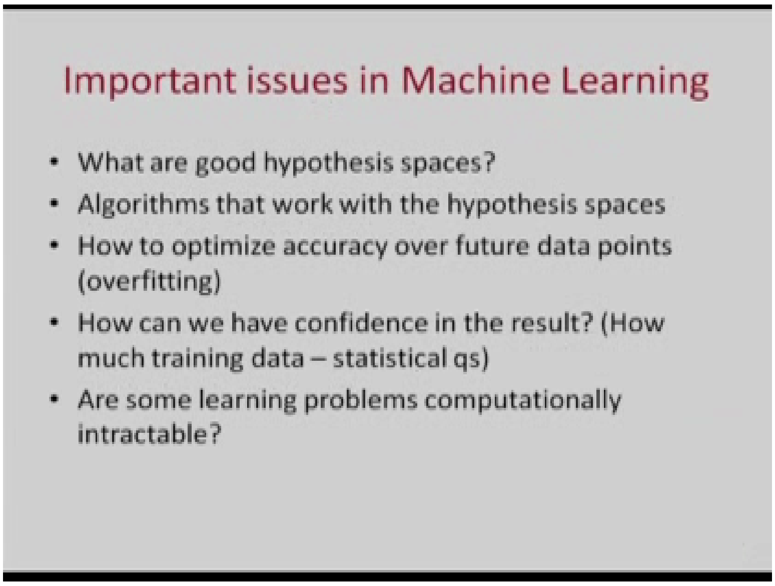


Some more Types of Inductive Bias

- Minimum description length: when forming a hypothesis, attempt to minimize the length of the description of the hypothesis.
- Maximum margin: when drawing a boundary between two classes, attempt to maximize the width of the boundary (SVM)

And there are other types of inductive bias like minimum description length, like maximum margin etcetera which will be only which can be explained, when we talk about the specific algorithms where such biases is used.

(Refer Slide Time: 30:51)

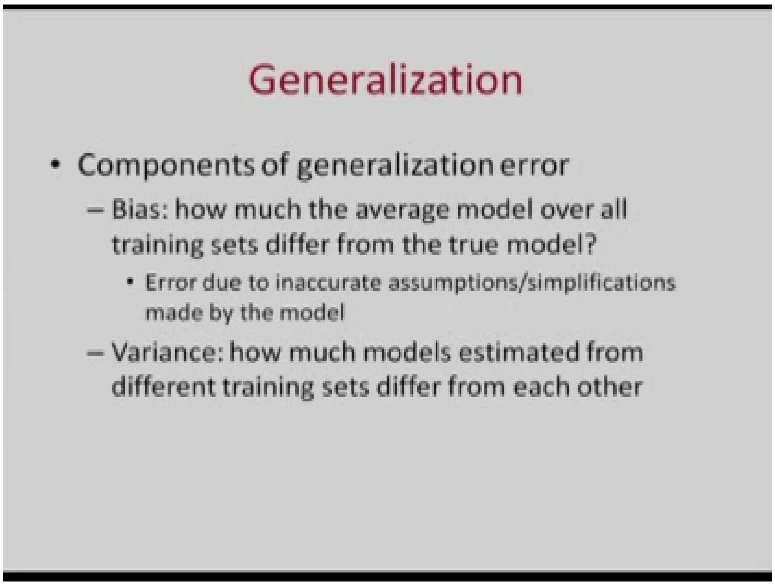


Important issues in Machine Learning

- What are good hypothesis spaces?
- Algorithms that work with the hypothesis spaces
- How to optimize accuracy over future data points (overfitting)
- How can we have confidence in the result? (How much training data – statistical qs)
- Are some learning problems computationally intractable?

So, in machine learning, you have to come up with a good hypothesis space, you have to find an algorithm that works well with the hypothesis space, you have to come up with the hypothesis algorithm that works well with the hypothesis space, and outputs on hypothesis that is expected to do well over future data points. And you have to understand what is the confidence that you have on the hypothesis and these are the things that we will discuss.

(Refer Slide Time: 31:24)



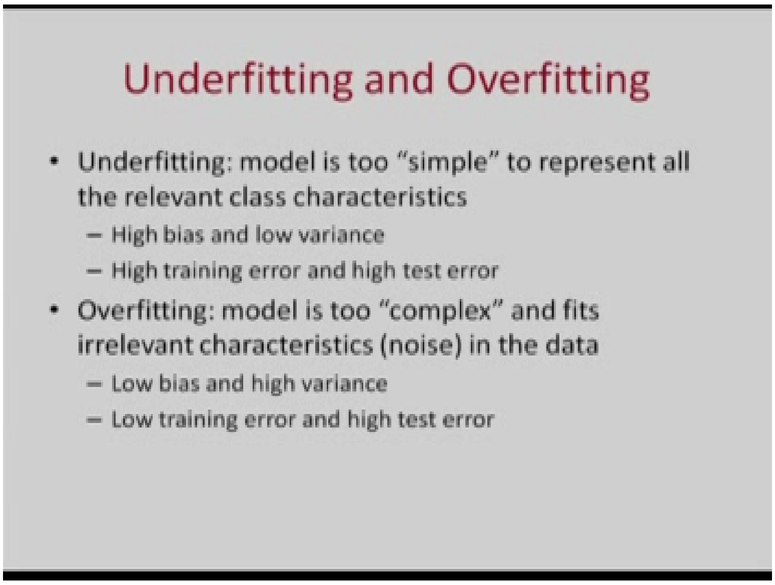
Generalization

- Components of generalization error
  - Bias: how much the average model over all training sets differ from the true model?
    - Error due to inaccurate assumptions/simplifications made by the model
  - Variance: how much models estimated from different training sets differ from each other

So, machine learning coming up with a function is all about doing generalization. And when you are doing generalization, you can make some errors. And the errors are of two types, bias errors and variance errors. So, bias as we saw is a restriction on the hypothesis space or the preference in choosing hypothesis. By deciding a particular hypothesis, you impose a bias. So, this is error due to incorrect assumptions or restrictions on the hypothesis space, the error introduced by that is called bias error.

Variance error is introduced when you have a small test set, so variance error means the model that you estimate from different training sets will differ from each other. If you come up with the model from some 50 training set, 50 data points, and you take another 50 data points on the distribution you can come up with the very different model, then we say that there is a variance among the results.

(Refer Slide Time: 32:30)



Underfitting and Overfitting

- Underfitting: model is too "simple" to represent all the relevant class characteristics
  - High bias and low variance
  - High training error and high test error
- Overfitting: model is too "complex" and fits irrelevant characteristics (noise) in the data
  - Low bias and high variance
  - Low training error and high test error

And this point, we will discuss later when we talk about different learning algorithms. This is a very important concept, but we will talk about it when we talk about the algorithms, this is overfitting and underfitting. You may come up with the hypothesis that does well over the training examples, but does very poorly over the test examples, and then we say overfitting has occurred. Overfitting comes from using very complex functions so you are using too few training data. And the reverse of overfitting is underfitting, if you have a very simple function then it cannot capture all the nuances of the data. So, we will talk about details of overfitting and underfitting when we talk about specific algorithms.

With this, we come to the end of this module.

Thank you.