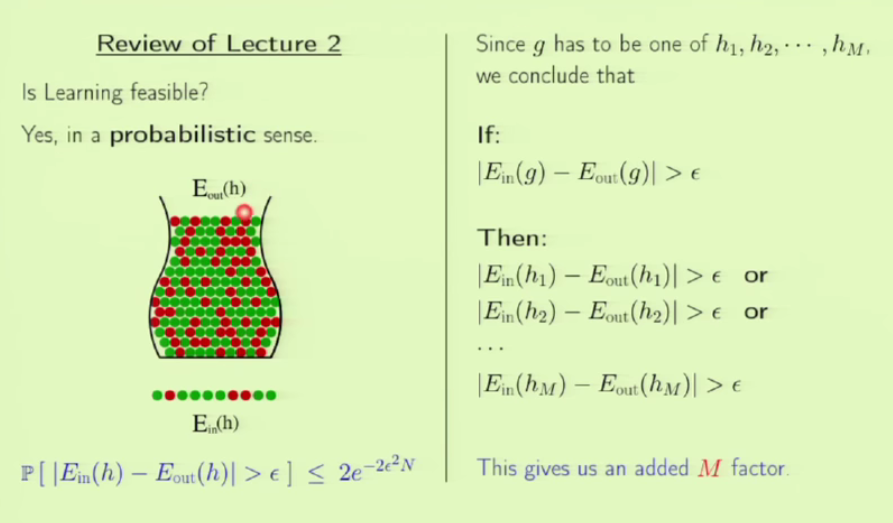
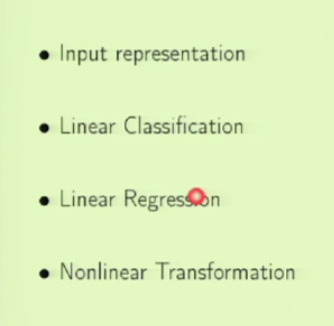
**Lecture 3: Linear model:**

**if we use a complex h set, then we will have high value for m, which will increase the value of RHS. We are not using this as the final value, we are using this for proving that learning is feasible. It has to be refined further.**

This lecture is included to have some practical task.

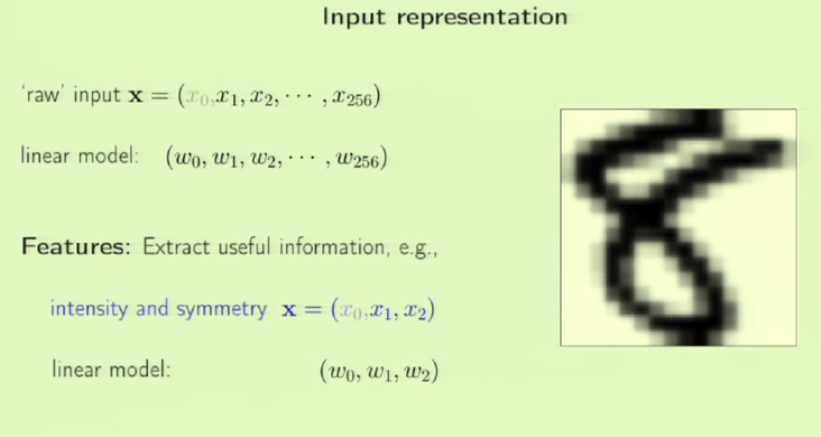
We will use a dataset. We will use perceptron for linear data and also generalize it for not linearly separable data.

We will try regression where output is not a class but some continuous value.

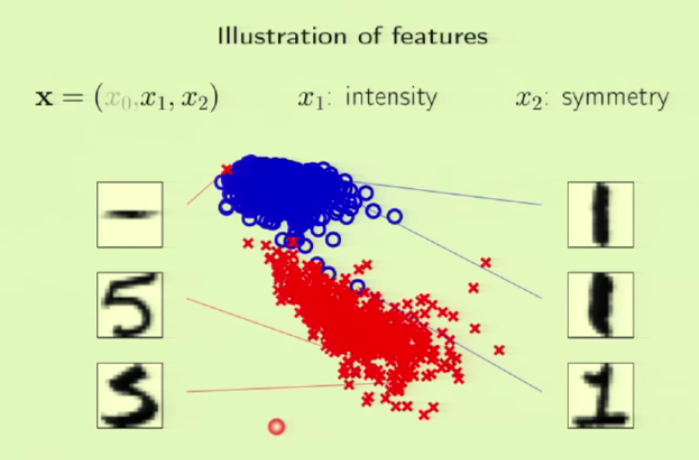


We work on digit data set. These are digits as written on post cards. There is about 2.5% error in correctly recognizing the digits by humans. Have to see how the learning model performs!

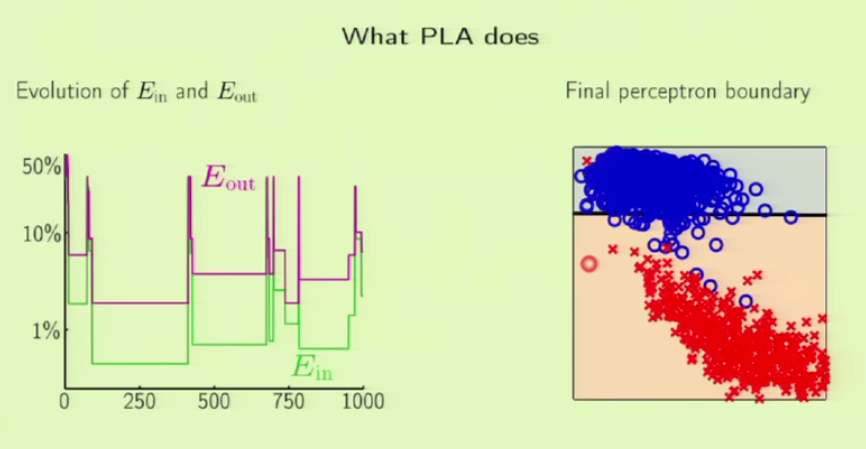
We will try it on Perceptron Algorithm.

Each image is 16X16 pixels. So there will be 256 real numbers in an image. For each cell we can have a feature which results in 256 cells. We are using PA model for this for which a linear model with 256 features is too much(256D). We should simplify it. So instead of taking all cells, we use some other way. We can use intensity and symmetry to narrow down the feature list. Different digits have different symmetry and intensity(like 8 has more intensity or black cells than 1). So now for each digit we have only 2 features – x1 as symmetry and x2 as intensity. In this process we did lose some information, but its ok to lose irrelevant information. We came down to 2 features from 256! From 256D to 2D

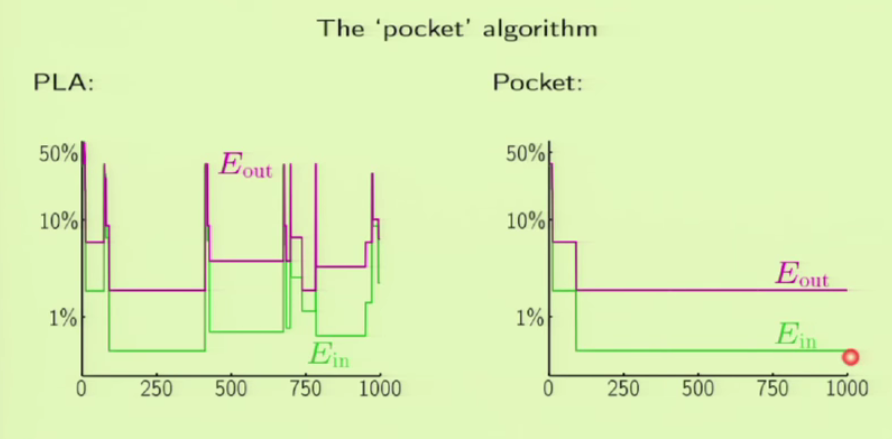
We add x0 for every feature list(which is bias).

Lets compare 1 and 5. 5 has higher intensity and 1 has more symmetry.

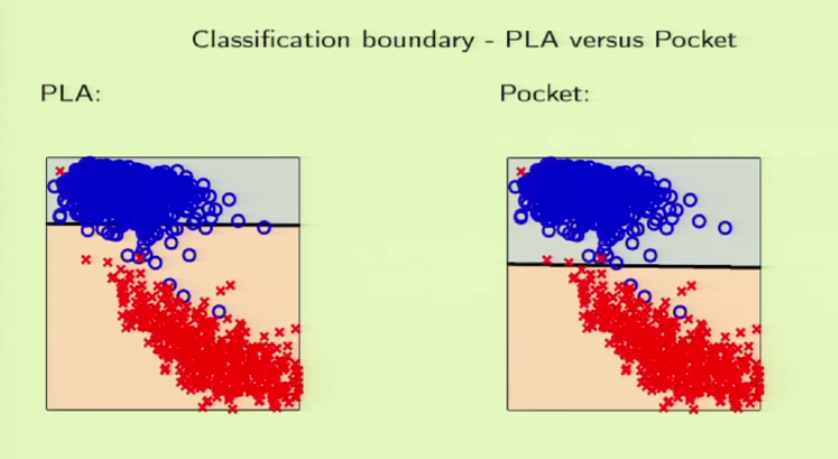
We are using PLA on this.

If the data is not linearly separable, PLA doesn’t work well on it. It keeps going for good to bad. For nonlinear data, PLA never converges, so we program it in such a way that it goes on until 1000 iters and then take whatever value we have. We see in the plot that there is a lot of difference between Ein and Eout. But one positive thing is that, Eout traces Ein pretty well, ie Eout decreases when Ein decreases and increases when Ein increases. What will be the final boundary?

We see that PLA didn’t give an efficient boundary.

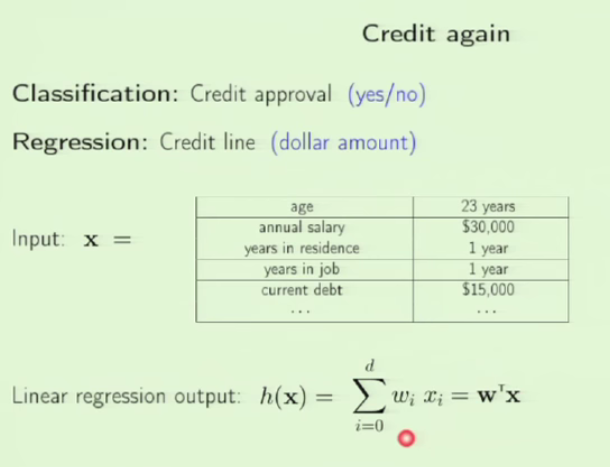
To do better, instead of taking the last H or 1000th H, we keep track of the best H so far and use that as our H.

We call this the Pocket algorithm. We see that nowhere does Ein gets worse or go up in the plot. It will only keep track of the hypothesis that are doing better.

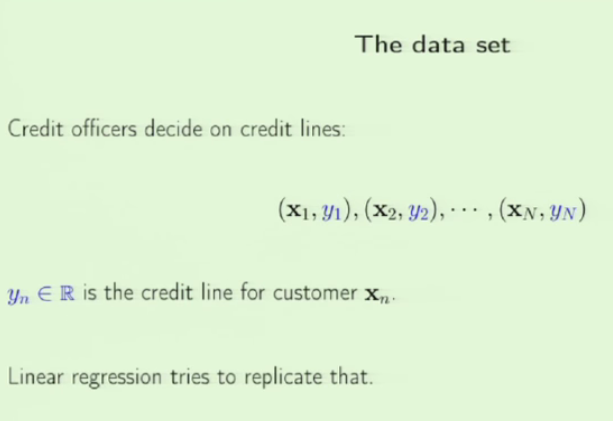


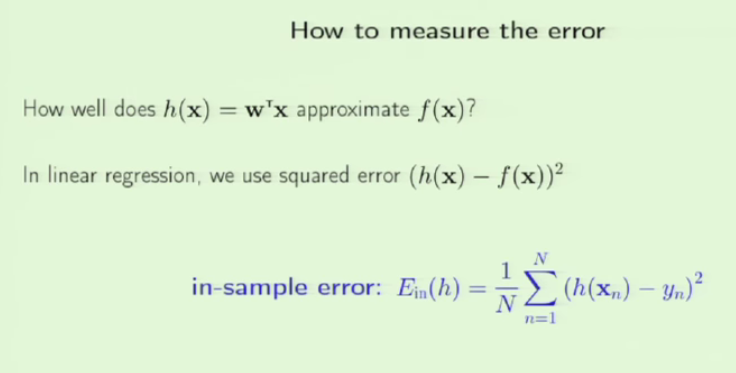
The plot proves how the Pocket algo improves the boundary.

This way we will be able to handle partial inseparable data.

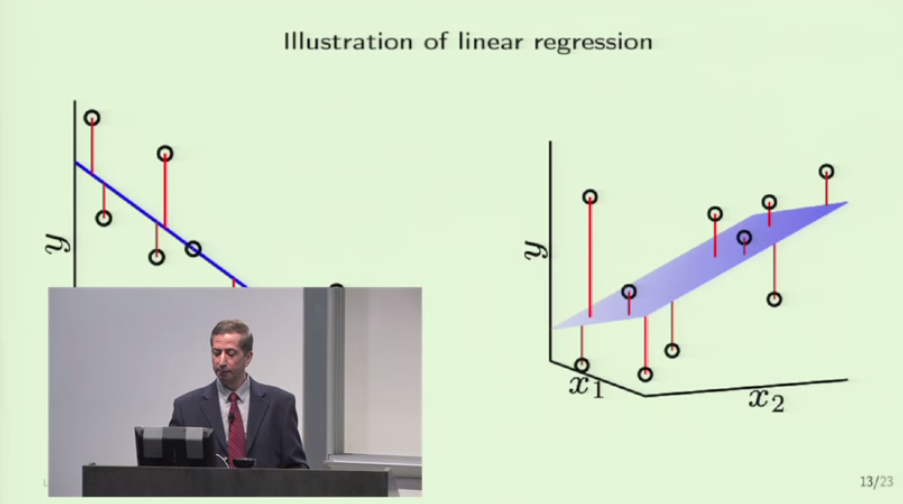
Regression simply means real valued. It is widely used in Statistics. Suppose we want to relate the courses to future earnings. We use GPA of different courses to predict salaries after 10 years. WE can apply regression to make this prediction. We can use the credit problem for regression too. Earlier we used it for classification.

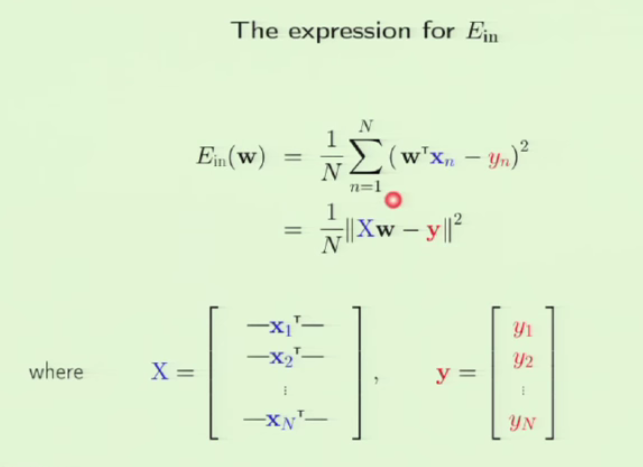
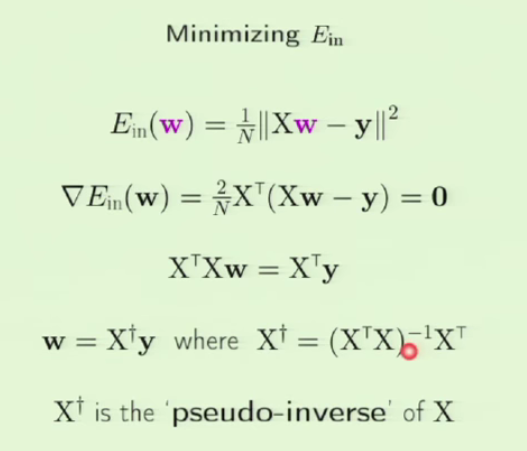
The RHS(Wtx) is just a vector form for simplifying.

We have some historical data from prior customers. We use that to predict credit for new customer.

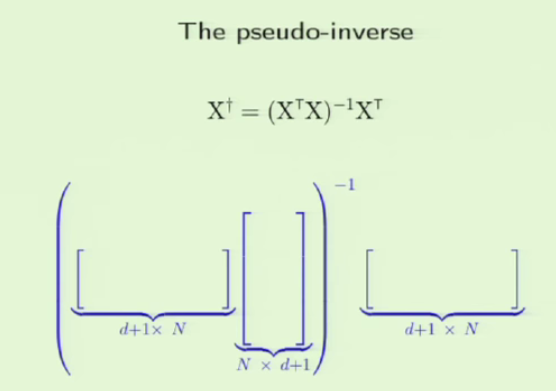
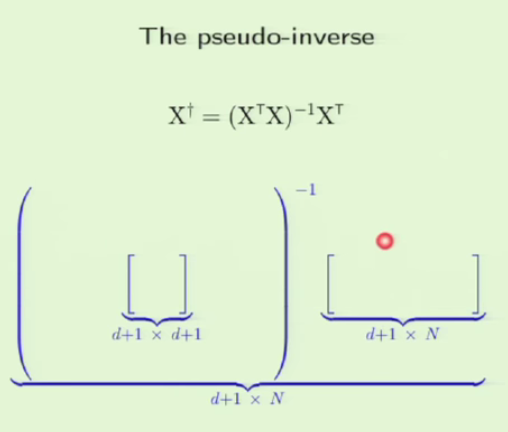
We should first measure the error. We will have an algo that determine the weight vector. These result in different h. In classification, it is simple in terms of output, it is either 1 or 0(something like that). But with regression where you should predict some real value, each prediction will be given some kind of a feedback, like it is really bad, or it is close, good…

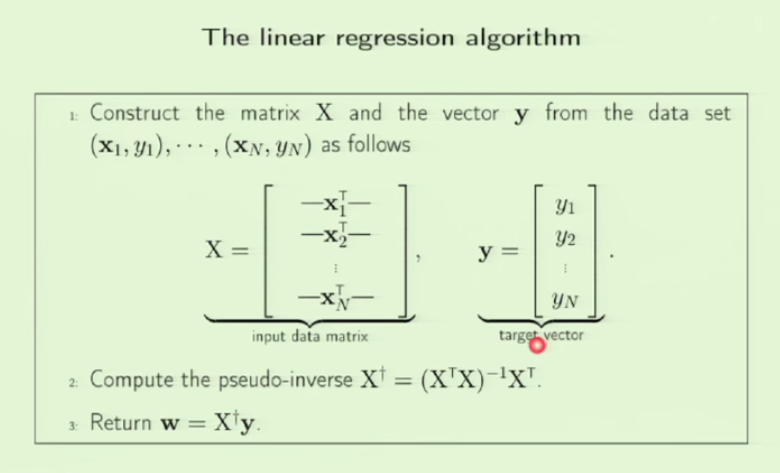
H depends on w. h(xn)-yn)squared gives us the error of each h, We can calculate its average to have average error.

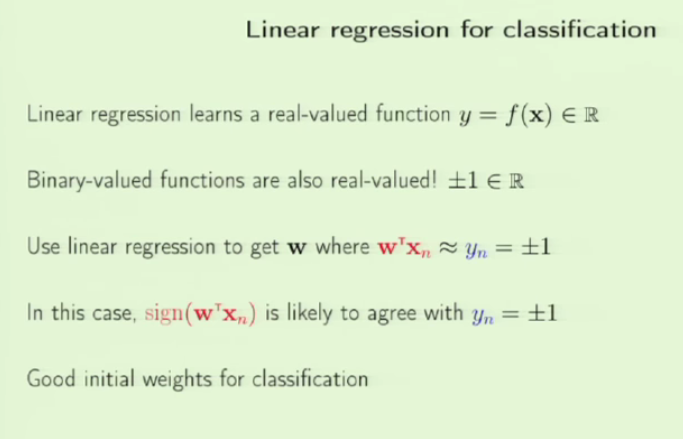


We should minimize that error. Gpa is X and earnings 10 years later is y. For different x we have different y. When we fit a model to the data, we will notice the error in predictions we would get if we use that model. If we use higher dimension, we will have a hyperplane instead of a line.

We see that X and y are constants, whats varying is the w. So we should reduce the error in w or adjust it. So we have Ein as a function of W. XTX gives us a square matrix.



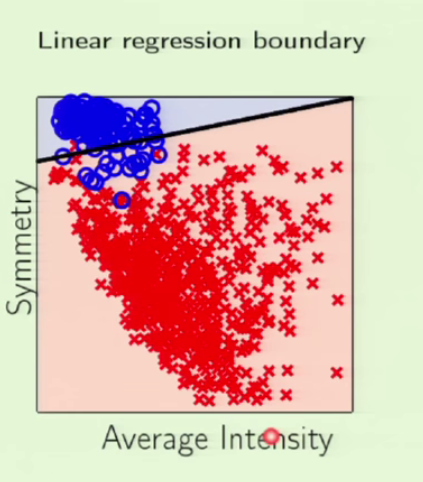
The above fig breaks down the calculation of pseudo inverse. The programming languages has functions to do this for us. This is one step of learning. As long as its correct, it can be used. This is often used as a building block.



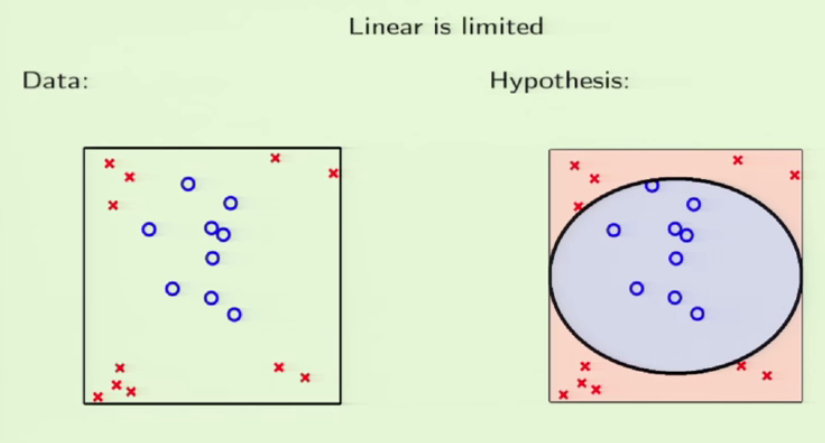
Linear regression can also be used for classification.

But binary valued functions are also real valued, 1 and 0 are also real besides binary.

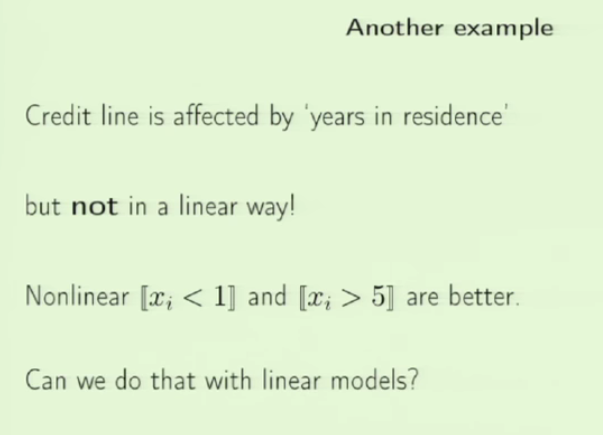
If the value is close to +1 or -1, we can use it as classification.

By choosing good initial weights using LR, we can get better classification using regression than classification models.

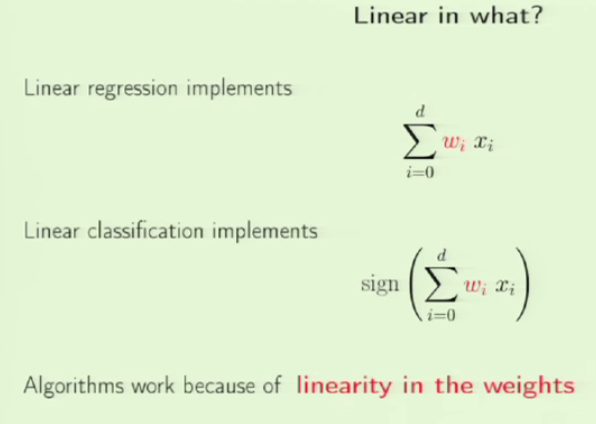
We see that the boundary isn’t good. For values closer to boundary, it assigns +1(blue points above the boundary) and -1(red points below the boundary). But there are red points which are far from the boundary which the model cant approximate to -1.

Nonlinear transformation: The data can be anything. Linear or nonlinear. Some data is not linearly separable. We can use nonlinear separator.

The credit line example

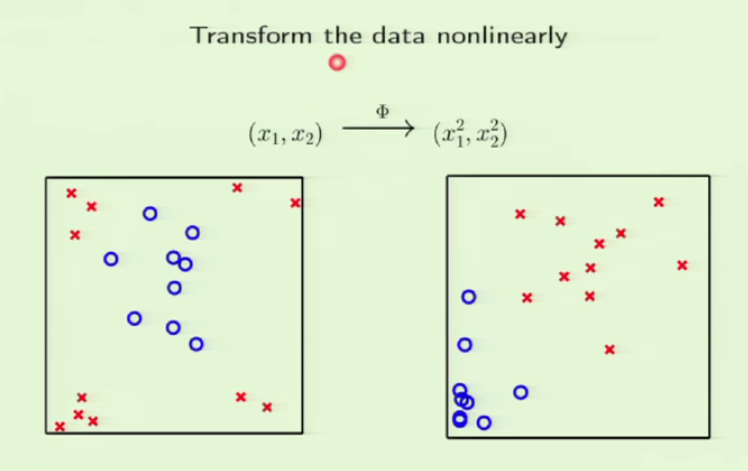
Is the applicant a resident for less than one year? More than 5 years? The double braces mean the condition inside will return 1 if

true, 0 otherwise.



When we are talking about learning, it is dictated by the weights chosen rather than the features.

The data features or input remains as a constant here. It is the weights that keep varying.



We take constants and find their squared value, which again are constants. Using these squared constants, we can linearly separate the nonlinear data. But there is a big catch here.

**Q & A**

How to choose transformation function? There are some guidelines for choosing TF. We can try different functions like x, x2,x3 .. More about this in the next lectures.

If you do well in in sample, you take it that it will do well on out sample. We can observe and handle only the in sample error.

Why is w0 included? The linear regression line doesn’t start from origin. There is some off set which is why we need w0. What will happen in case of binary classification?

How to come up with features? Go through the features, know the domain, and understand the problem. Draw insights, derive the features or input. You should study both data and the problem.

We can have features, feature of features.

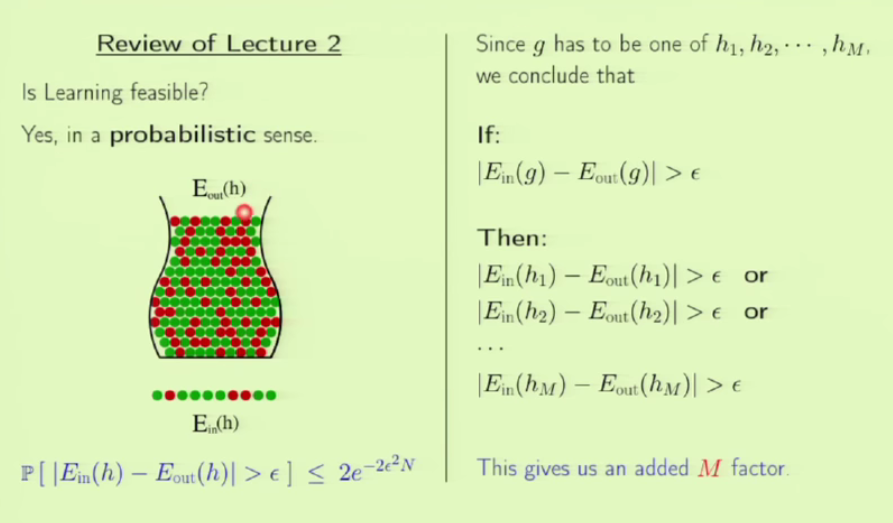
The linear model with soft or hard threshold is the building block of many models like NN, SVM, kernel methods. Linear model is used by many.

How to evaluate Ein and Eout value? We can assess Ein but not Eout. But we try to infer that Eout and Ein track each other.

If we have a linear function f(x) = y, we can transform that into nonlinear by using x2, x3 etc. This in high dimension gives a different model than a line.

For credit approval use classification, for credit line use regression. It depends on target function and requirement.

What makes a nonlinear function good? Later chapter..



g is one of the hypothesis functions h1 or h2 or h3 ..

As shown in the equations, if g does poorly, then that means using that at least one of the hypothesis will cause bad results.

This chapter should actually start with M factor, but to not make it totally theoretic, linear models is chosen

The bin is analogous to real time problem like credit. The red marbles are hypothesis that gave wrong predictions and the green marbles represent hypothesis that gave correct predictions. U is the error value of out sample and v is in sample error.