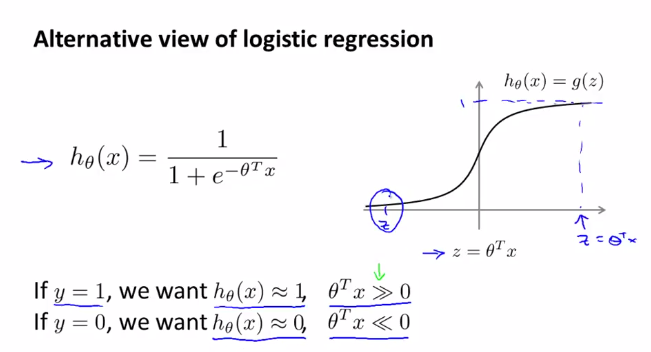
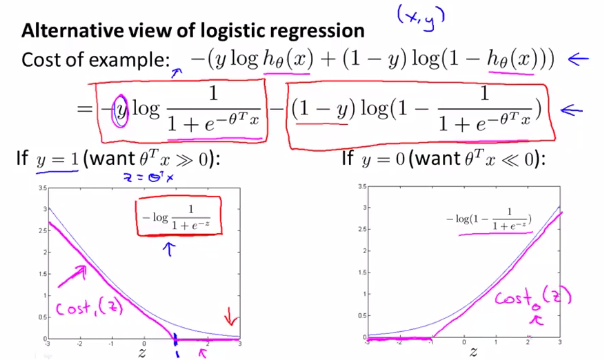
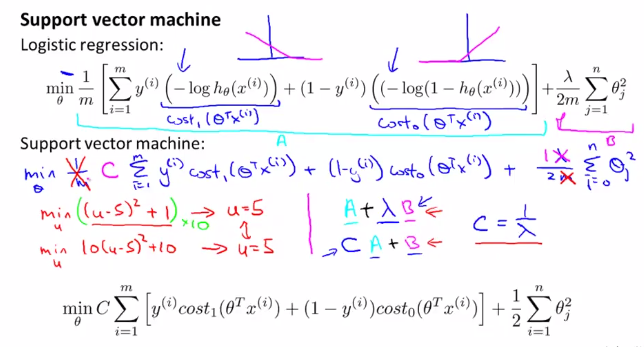
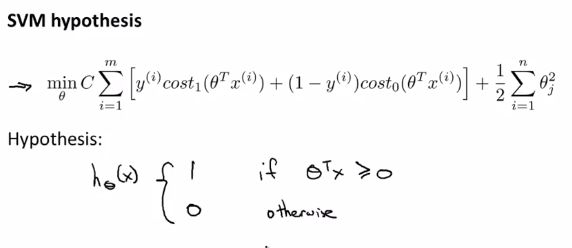
Lesson 1 of 4 – Large Margin Classification

# OPTMIZATION OBJECTIVE

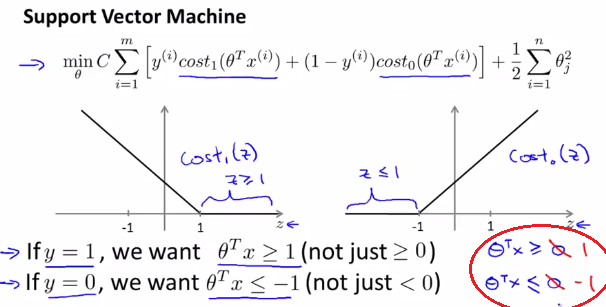




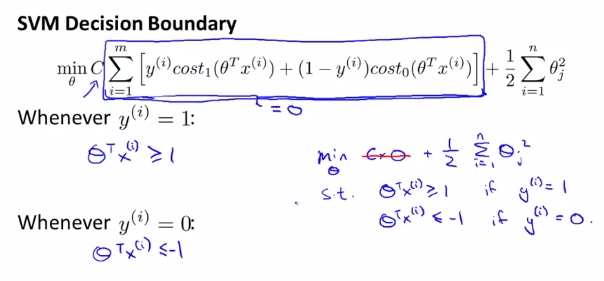




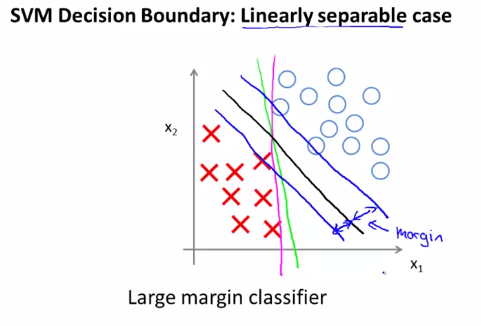
# LARGE MARGIN INTUITION

Sometimes Support Vector Machines are called Large Margin Intuition. The image below shows the safety margin factor.

If C is too large then the optimization algorithm will try to transform de A factor to zero.

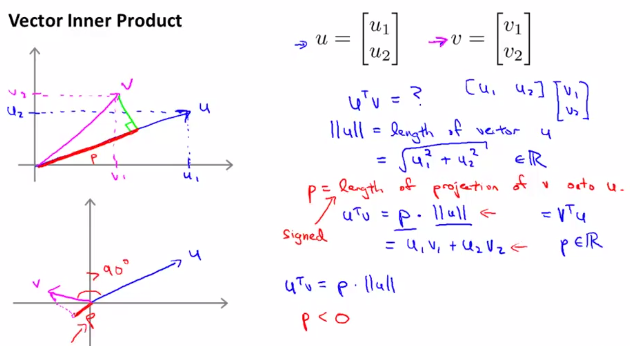


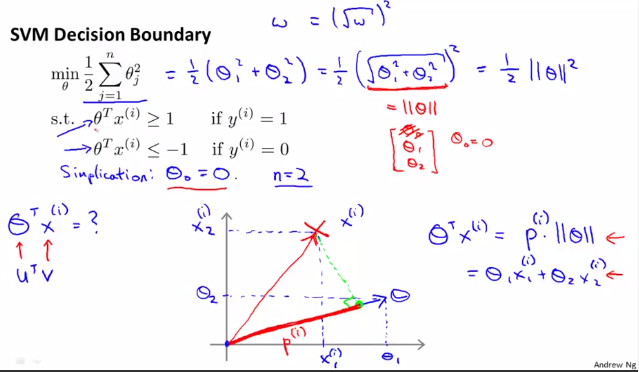
When you minimize this as a function of the parameters theta you get a very interesting decision boundary.



The SVM will choose the black line. The distance is called the margin, when I draw up this two extra blue lines, we see that the black decision boundary has some larger minimum distance from any of my training examples, whereas the magenta and the green lines they come awfully close to the training examples. And so this distance is called the margin of the support vector machine and this gives the SVM a certain robustness, because it tries to separate the data with as a large a margin as possible.

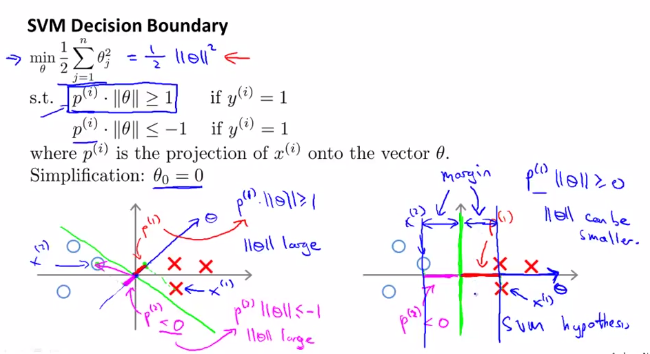
# MATHEMATICS BEHIND LARGE MARGIN CLASSIFICATION





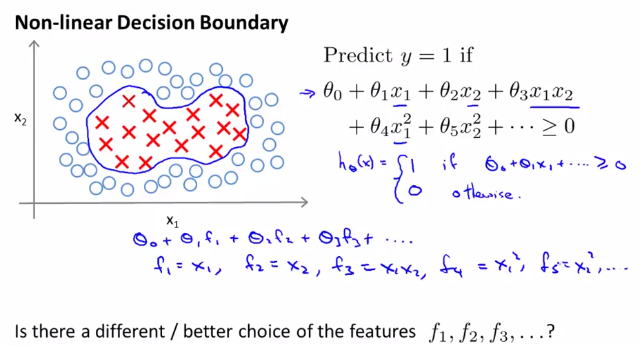
Using the simplification means that the theta vector will pass on the origin. The green solution from the left is not a good one because it has very little margin. The projection of theta vector is 90° from the solution.

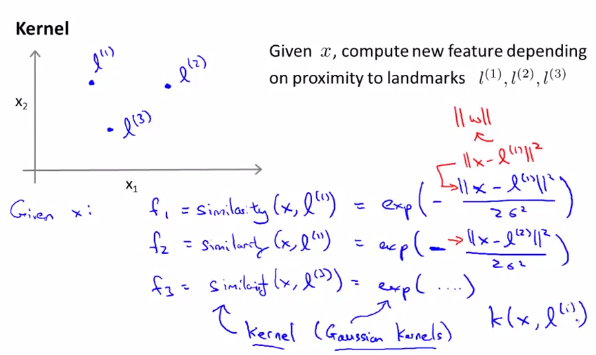
If you look at the green hypothesis from the right, we want the projections of my positive and negative examples onto theta to be large.

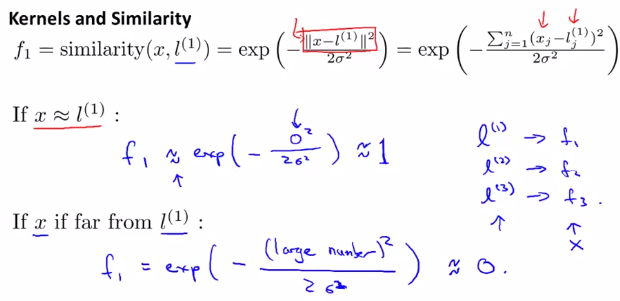


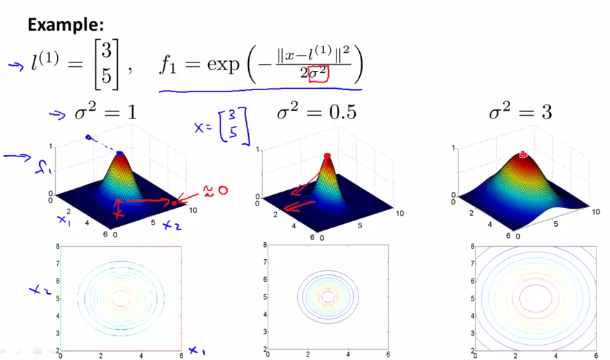
Lesson 2 of 4: Kernels

# KERNELS I

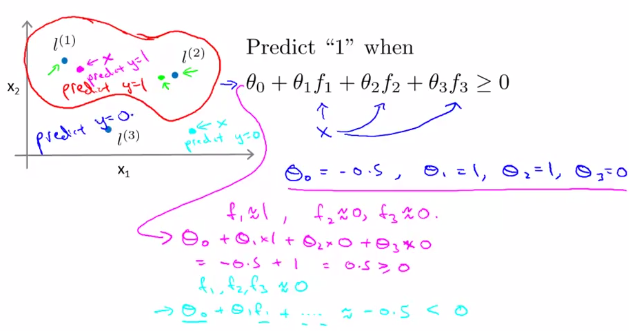






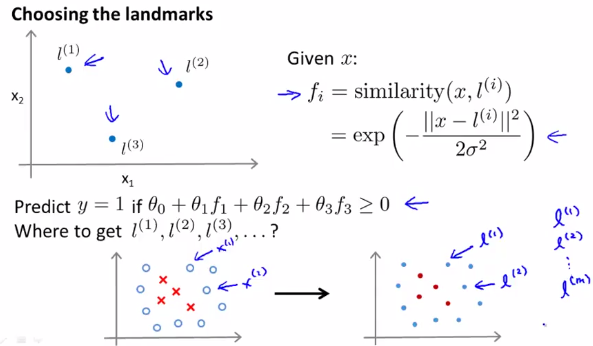


And in fact, what you end up doing is, you know, if you look around this boundary, this space, what we'll find is that for points near l1 and l2 we end up predicting positive. And for points far away from l1 and l2, that's for points far away from these two landmarks, we end up predicting that the class is equal to 0. As so, what we end up doing, is that the decision boundary of this hypothesis would end up looking something like this where inside this red decision boundary would predict Y equals 1 and outside we predict.

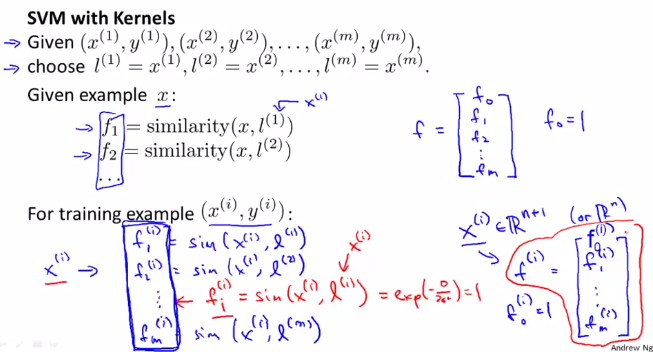


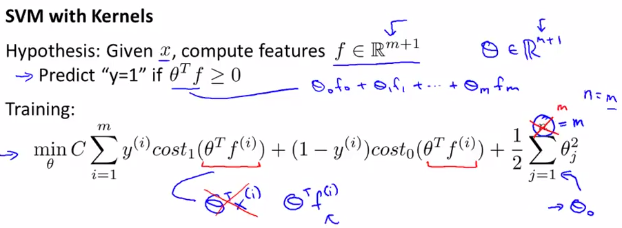
And so this is part of the idea of kernels of and how we use them with the support vector machine, which is that we define these extra features using landmarks and similarity functions to learn more complex nonlinear classifiers.

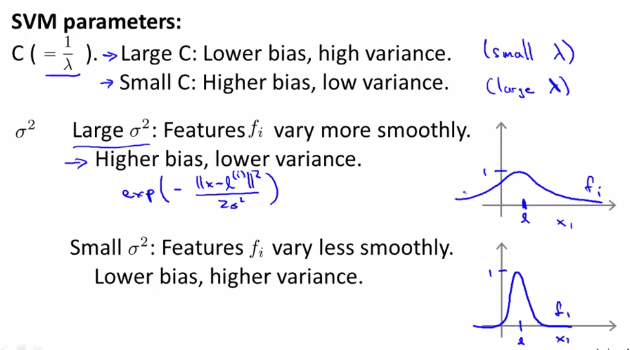
# KERNELS II



Given m training examples, I'm going to choose the location of my landmarks to be exactly near the locations of my m training examples.







Suppose you train an SVM and find it overfits your training data. Which of these would be a reasonable next step? R: Decrease C; Increase σ² (sigma).

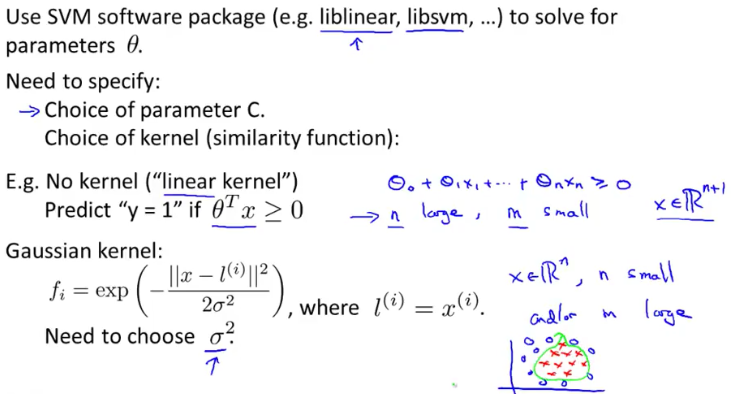
Lesson 3 of 4: SVMs in Practice

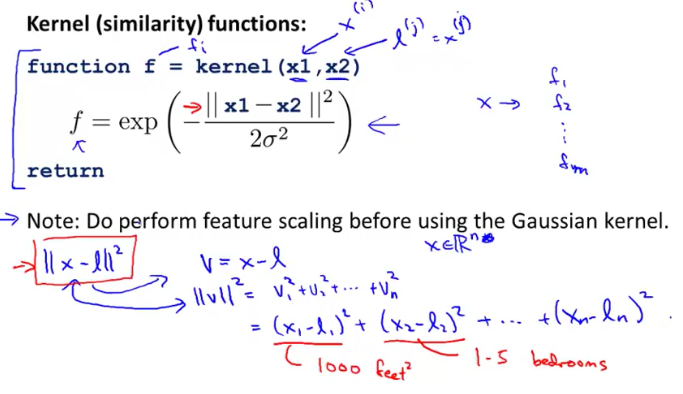
# Using an SVM

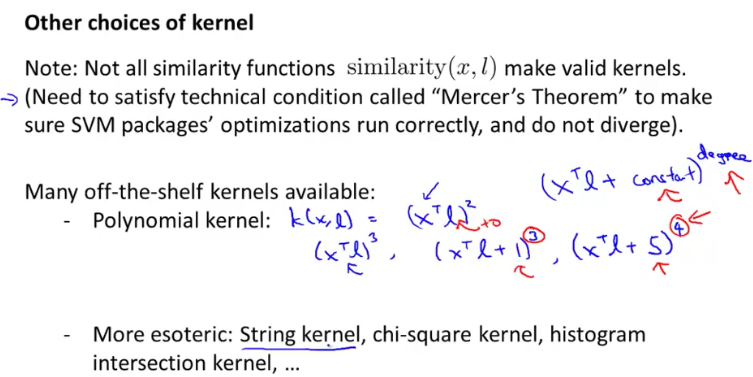
When not to use kernel (linear kernel): if you are trying to fit a very complicated function, in a very high dimension feature space, but your training set sample is small. If you use kernel you can cause overfitting.

If sigma squared is small, then you have a higher variance, lower bias classifier.

When to use Gaussian kernel: when your training set is big and you have a higher number of features.







Suppose you are trying to decide among a few choices of kernel and are choosing parameters such as C, σ², etc. How should you make the choice? R: Choose whatever performs best on the cross-validation data.

