Overfitting and Regularization

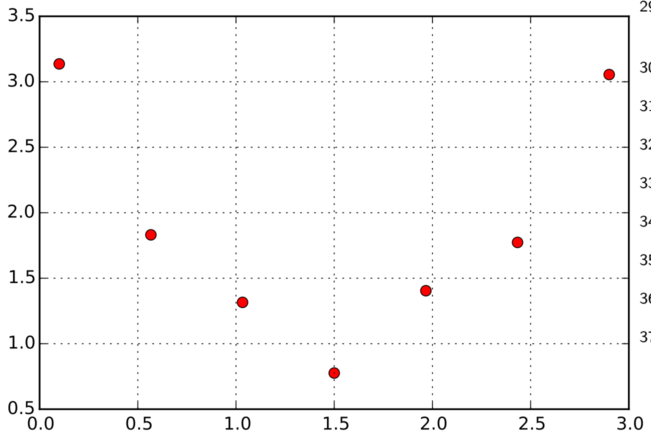
Don't try too hard.

Executive Summary

Don't try to hard to approximate the target function. Trying hard will make E_{in} really low but E_{out} will be terrible.

Overfitting

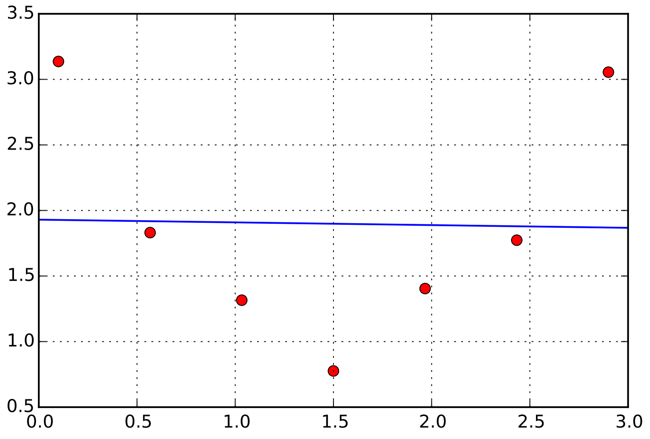
Consider the following problem: we have the following data set which I generated from parabola with some noises.



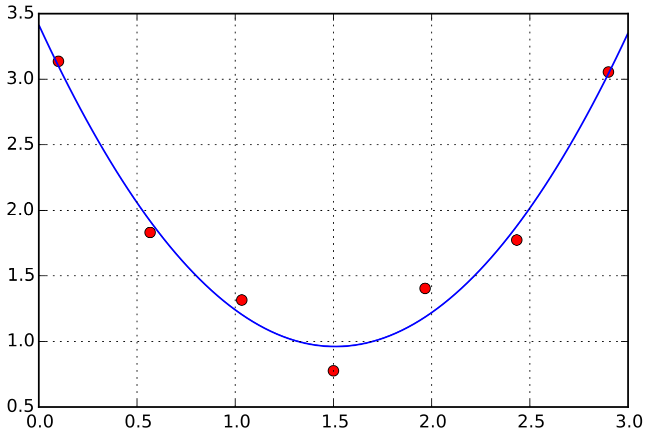
Now, we will try three things

1. Not trying much. First we will try to fit it with a straight line $\vec{w} \cdot [1, x]$
2. Trying just right. We will try to fit it with a parabola $\vec{w} \cdot [1, x, x^2]$
3. Trying too hard. Fit it with a sixth order polynomial. $\vec{w} \cdot [1, x, x^2, x^3, x^4, x^5, x^6]$

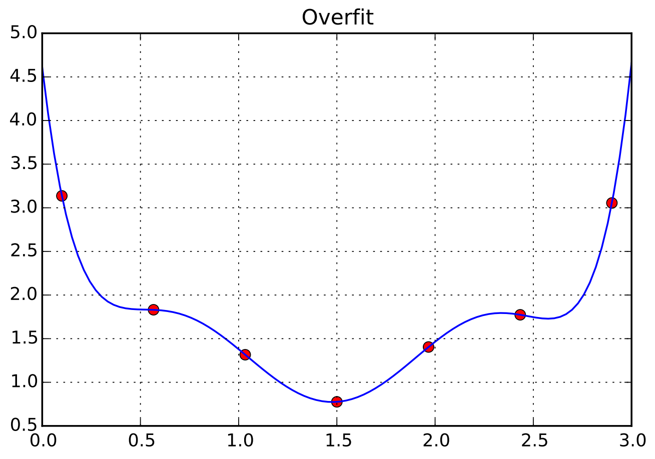
Trying to fit it with a straight line is of course a terrible idea. It gives you a terrible cost and the line doesn't really approximate the training dataset. We call this situation underfitting. An analogy would be that you don't study at all for the exam so you do terrible on practice test and you will most likely do terrible on the real test too.



Now we are trying just about right. The cost function is good eventhough not perfect. The model looks simple enough that it should generalize out of sample. An analogy would be you actually try to do the practice exam to learn the material. You are not trying to get the perfect score on the practice test. Then you will probably do similarly well on the exam.



Using the sixth order polynomial is most likely overkill. You may thing that it couldn't hurt. But it actually does. If you look the the plot, you can see that the line pass through every point making E_{in} very close to zero. This sounds great for us; what can be so wrong about that. The problem comes when you try to use this in the real world which is what really matters. A complicated line like this doesn't have much chance of generalizing. That means even though we do perfectly in the training sample we will not be able to do as well out of sample. This situation is called overfitting. An analogy for this one would be you are trying to memorize all the questions on the practice test but you learn nothing from the practice; you just memorize the answers. You got a perfect score on the practice exam but you probably won't do well on the real exam.



The same situation happens as well when you try to do classification and your model is too complex or you have way too many features than the data points you have. You will find a fake pattern because of having "Too much information". The chance of we being able to generalize is not so good. The most obvious way to solve "Too much information" problem is to get rid of the information. We will learn this later on in the course. We are not going to do that today.

Regularization

The topic for today is called regularization. The basic idea can be described with the following analogy: if you are a studious person and you don't want to try too hard, find a friend who ask you out for Destiny every evening. Then you won't be able to try too hard anymore.

Recall that for linear regression we found $\vec{w}$ by minimizing the following cost function:

$$\text{cost}(\vec{w}) = \frac{1}{2N}\sum_{i=1}^{i=N}\left(\vec{w}\cdot\vec{x}^{(i)} - y^{(i)}\right)^2 \tag{1}$$

This cost function was built on the premise that if the guess/line is close to the training point then it is a better line. Yes, it is a better line but there is a catch. It is a better line "in sample". However, what we really care is the out of sample error not in sample error. Being the best in sample is not really what we want, we want the best weight for out of sample error.

We showed in the previous section that you can cheat this cost function in sample by adding more and more features and have $\vec{w}$ holds more and more parameters until the line passes through every point exactly. It is always possible to have polynomial of degree N passes through N data points. This is called Lagrange interpolation.

So, the goal of regularization is to deter this kind of cheating. As we saw in the previous example. The second order polynomial is probably a better idea than sixth order polynomial. But the second order polynomial is just a sixth order polynomial with $w_3, w_4, w_5, w_6 = 0$. So if we some how penalized the fact that the value of these w_i are too large then this should give us better generalization.

Thus, if we don't want w_3, w_4, w_5, w_6 to be too big, we could modify the cost function in Equation 1 to (for illustration purpose only)

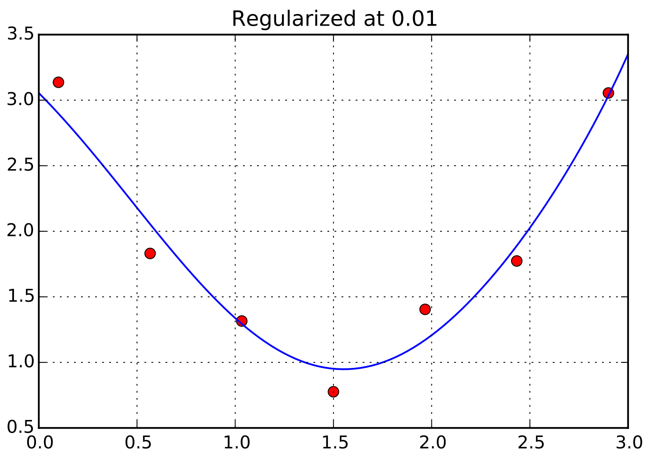
$$\begin{aligned}\text{cost}_1(\vec{w}) =& \frac{1}{2N}\sum_{i=1}^{i=N}\left(\vec{w}\cdot\vec{x}^{(i)} - y^{(i)}\right)^2 \\ &+ 100000w_3^2 + 100000w_4^2 \\ &+ 100000w_5^2 + 100000w_6^2 \end{aligned} \tag{2}$$

The equation above can be understood intuitively as the following: The first term will try make the line match the training points. But if we have to touch w_3 and w_4 we will be heavily penalized by the latter terms for making that usage of w_3 and w_4 not worth matching a few more points. So, the optimizer will choose the weight vector with low w_3 and w_4 but still fit the point quite well instead. All it does is that it helps the minimizer decide the trade off between using those weights and how well it fits the data.

The popular regularizer is similar to the above:

$$\begin{aligned}\text{cost}_{reg}(\vec{w}) =& \frac{1}{2N}\sum_{i=1}^{i=N}\left(\vec{w}\cdot\vec{x}^{(i)} - y^{(i)}\right)^2 \\ &+ \lambda \sum_{j=1}^{j=d} w_j^2. \end{aligned} \tag{3}$$

All it does is that it penalized by the the sum of square of the weight from w_1 and so on. The λ is called the regularization parameter it really doesn't have to be large to get a good result. The λ is telling you how much you should trade off between fitting all the data by using complicated model and fitting most of the data but use simpler model. The figure below shows the sixth order polynomial fit when I regularized it with $\lambda = 0.01$. If you really think about it, we don't really need large λ. All we want the regularization term to do is that when we have a bunch of similarly good lines please pick the less complicated line.



It is important to note that the regularization term in Equation 3 does not include w_0 which is the intercept. This is because there is no reason to penalize the intercept; it just shift the graph up and down and doesn't add any complexity to the model. Another way to view it is that it is the factor that is not multiplied by the real feature when you try calculate you hypothesis. It gets multipled by x_0 which we made up to be 1.

You may also ask how do I know which λ to pick. The answer is if you can draw them, draw them. But, if you can't draw, we can do something with it too using Cross Validation. We will cover this later in the course.