CSI 5138 [Intro:DL/RL](https://uottawa.brightspace.com/d2l/home/119135)

Homework Exercise I

Student(s): Wei Li

1. Problem Description

The assignment requires to learn polynomial regression models using data sets generated from a given structure:

where X is taken from a uniform distribution U(0,1) and Z is taken from a Gaussian distribution N(0, ).

The polynomial model takes the following form:

is the degree of the polynomial model, the higher value takes, the higher capacity the polynomial model has (which means a bigger hypothesis space). In our experiment, is a parameter to be examined, representing the model’s complexity.

A size of N data set is generated to serve as training set, and being fed into the model to learn the parameters . The training set size N is another parameter of interested, it is the amount of data we can get in a real learning case.

In the learning process, square loss is used as the loss function, which take the form:

is the essentially a matrix built from stacking up vectors , ,

A mini-batch SGD technique is used to optimize the parameter . The initial is generated from normal distribution, then 20 instances are randomly taken (batch size equals 20) to fed into the model to calculate the gradient, which is then used to calculate the new . Note that when N<20, mini-batching will be disabled.

The learning rate is set to 0.25 based on some quick tests, and a max iteration of 10000 is set as the termination condition. The reason to choose max iteration instead of a small loss or update value as termination condition is: 1) Given the limitation of calculation power, a max iteration is easier to estimate the running time during experiment. 2) The objective is to evaluate the performance of the model given different , and , if we train models that take different , and combination to the same training error (i.e. ), then there is no point to do comparison. It is more reasonable to give them the same training time, which is large enough to make sure that each model’s loss become stable.

After the model is built, its training MSE (the mean square loss over all examples) is calculated as . A test data set of size 1000 is generated from the same distribution as the training set to test the model’s generalization ability. The test set’s MSE is calculated as .

The experiment is divided into two part. In the first part, we take parameter combination from the set (their Cartesian product), where ∈ {2, 5, 10, 20, 50, 100, 200}, ∈ {0, 1, 2, . . . , 20}, ∈ {0.01, 0.1, 1}. For each parameter combination, we generate data set, train model and calculate itsand for 30 times. The average and is then calculated as and . The average model’s generalization ability is evaluated as .

In the second part, we add a L2 regularization term to the loss function, that is:

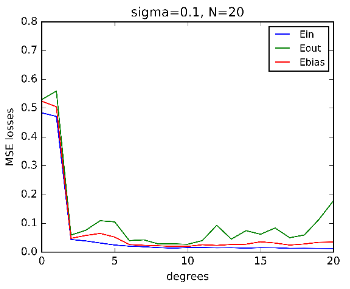
,

Where Then we will do the same thing as the previous part.

Lastly, the resulting , and is plotted and analyzed for each combination of , and whether or not we use regularization.

1. Outcome and analysis
   1. **An example of outcome analysis**

Our analysis and plots will take the following form. The parameter of interest will be plotted in the x-axis (in this case, the models’ degree), while models’  ***,*** and is plotted as three different lines in the y-axis -- the MSE axis. Here is the example:



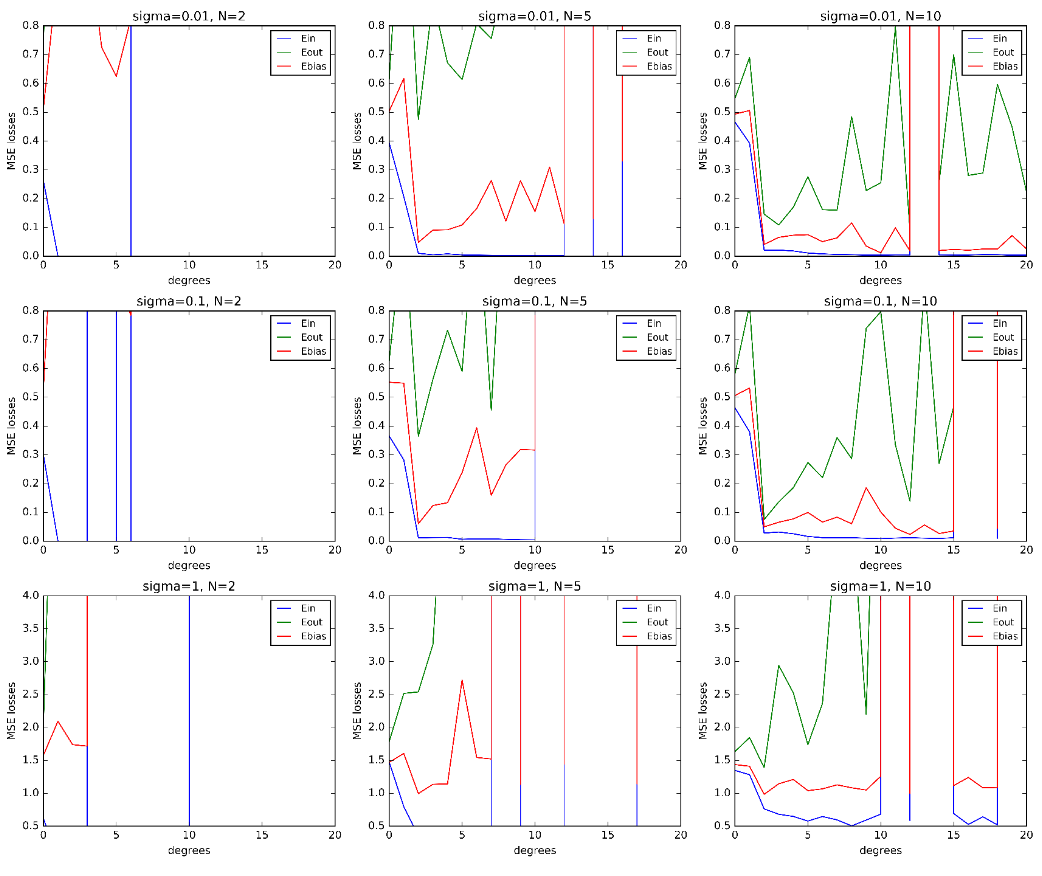
**Figure 1. An example**

Given different ,combination and whether or not we apply regularization, the models’ MSE and the shape of the plot will be compared and discussed.

* 1. **Effect of model complexity**

Let us firstly look at how the change of model capacity can affect the models’  **,**  and . The parameter is an indicator of model complexity, as the higher is, the more possible models we have, the more parameters there are, and the higher the chance the model overfit.

The first pattern that is discovered is that when the number of training data is severely insufficient, the performance of the models go overfitting quickly. The fewer data we have and the higher noise there is (the lower-left side of figure 2), the severer and quicker (means overfitting occurs when degree is still not very big) the overfit is.



**Figure 2. Pattern of change in low data size setting**

As shown in the figure 2, while the upper-right figure still display**s** a clear pattern of **,** the overfitting in the lower-left side is so severe that it is hardly recognizable in this MSE range.

However, when sufficient data is fed into the model to some extent, the models’ complexity stop to cause overfitting (at least in this degree range), and high complexity actually leads to better result, either or **.**

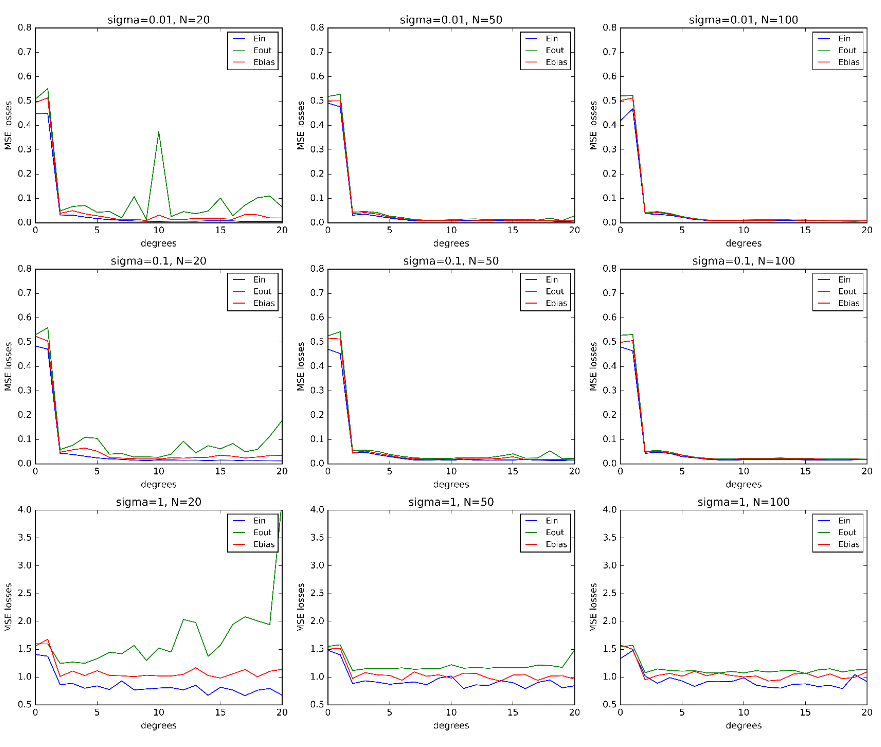
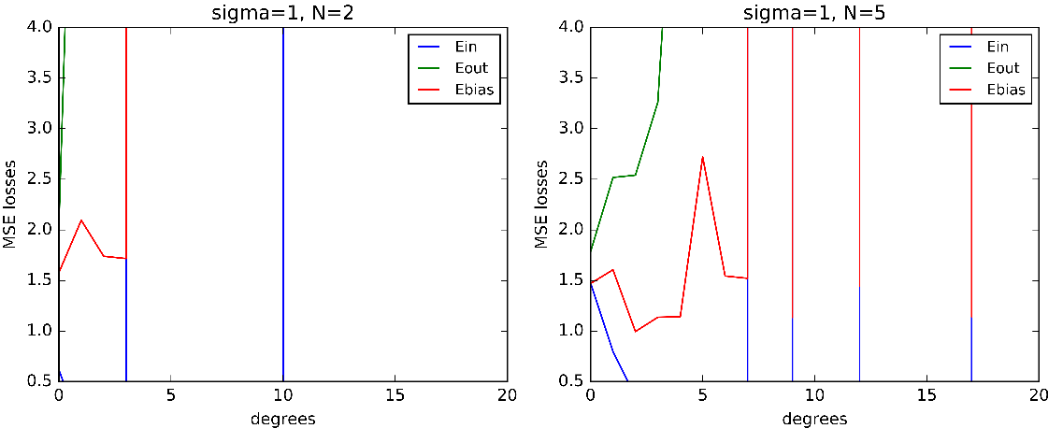


Figure 3. Pattern of change in high data size setting

The figure shows how adding parameters stop to be a problem as we add more training data. The first column act likes an equilibrium where you can still see a trend that goes up wildly if we keep on adding parameters.

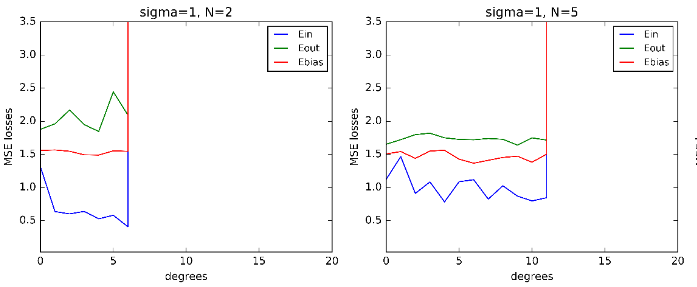
Adding regularization term can help to tame the overfitting given insufficient data and high model complexity. The outcome shows that given small data set and fix noise, the number of parameters it needs to go overfitting increase as we add the regularization term into model. Moreover, the Generalization Gap also goes down significantly. Here is an example of the effect of regularization in small and noisy data set:

Before applying regularization:



**Figure 4-1. Pattern of change before regularization**

After applying regularization:

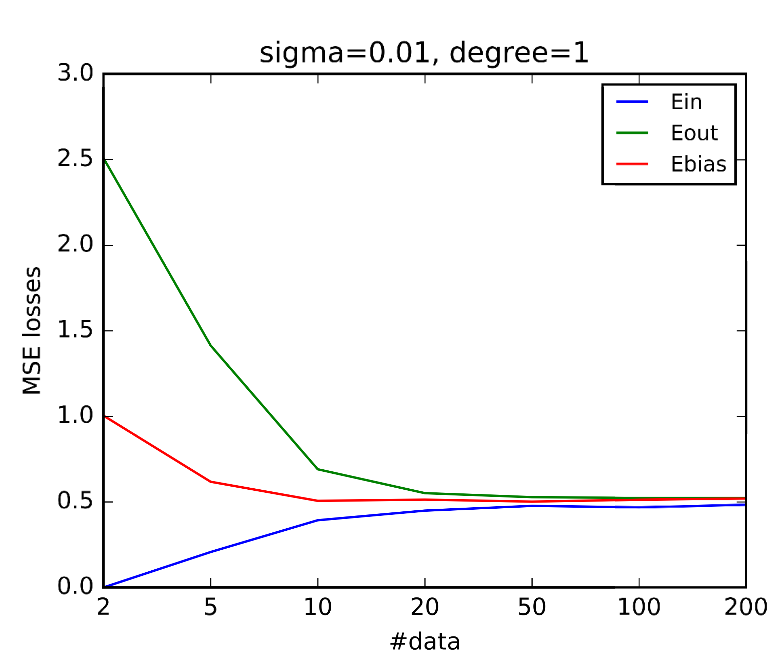


**Figure 4-2. Pattern of change after regularization**

* 1. **Effect of Number of data**

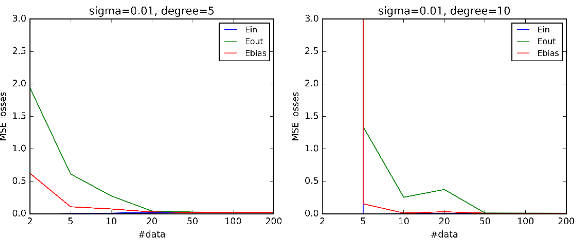
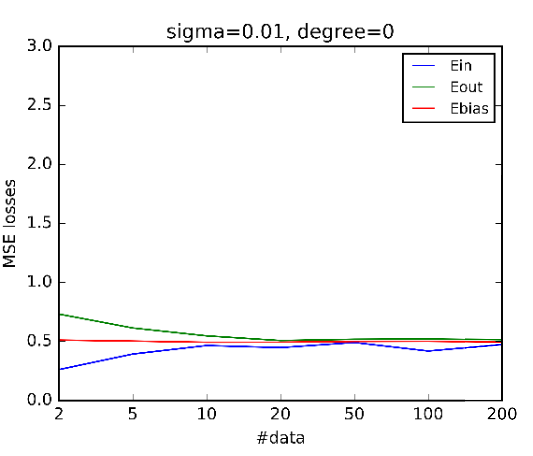
The analysis above has already shown how the number of data can affect models when models’ complexity change, the analysis below gives us a different perspective.

General speaking, the plot over change under different circumstance follows the same pattern: the goes up as increase, while goes in the opposite direction. and have a tendency to converge to the same level, while always in between the two. Here is an example:



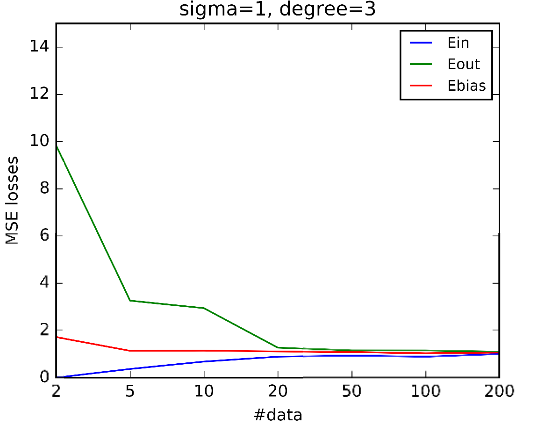
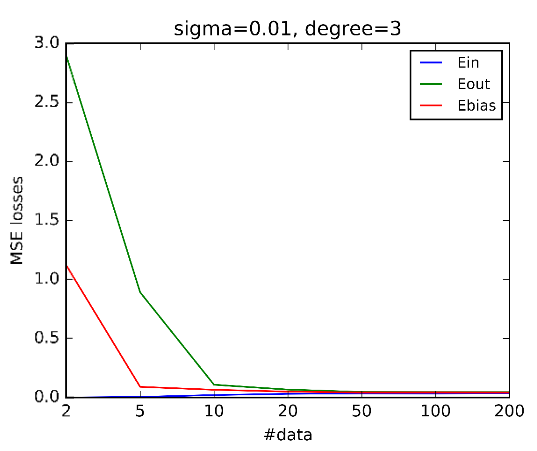
**Figure 5. Pattern of change**

However, this pattern takes different shape in different - setting. Firstly, we observed that the higher the degree (model complexity), the quicker the drop to converge with , here is a comparison:



**Figure 6. Pattern of change in different setting**

The vertical line means that the MSE drop so fast that it is almost vertical in this MSE range.

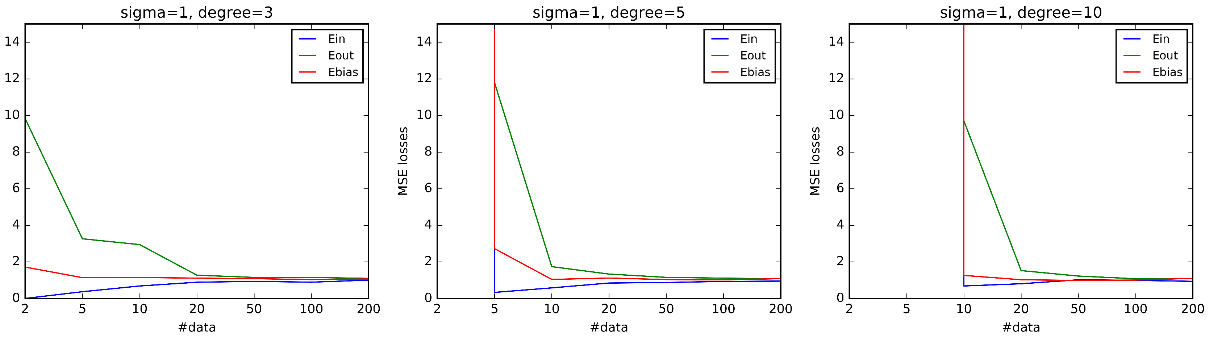
Another observation that the higher the noise is, the higher the converging MSE level will be: 

**Figure 7. Pattern of change in different setting**

Although MSEs are plotted in different range, we can see that the converge level when = 0.01 is smaller than that when = 1.

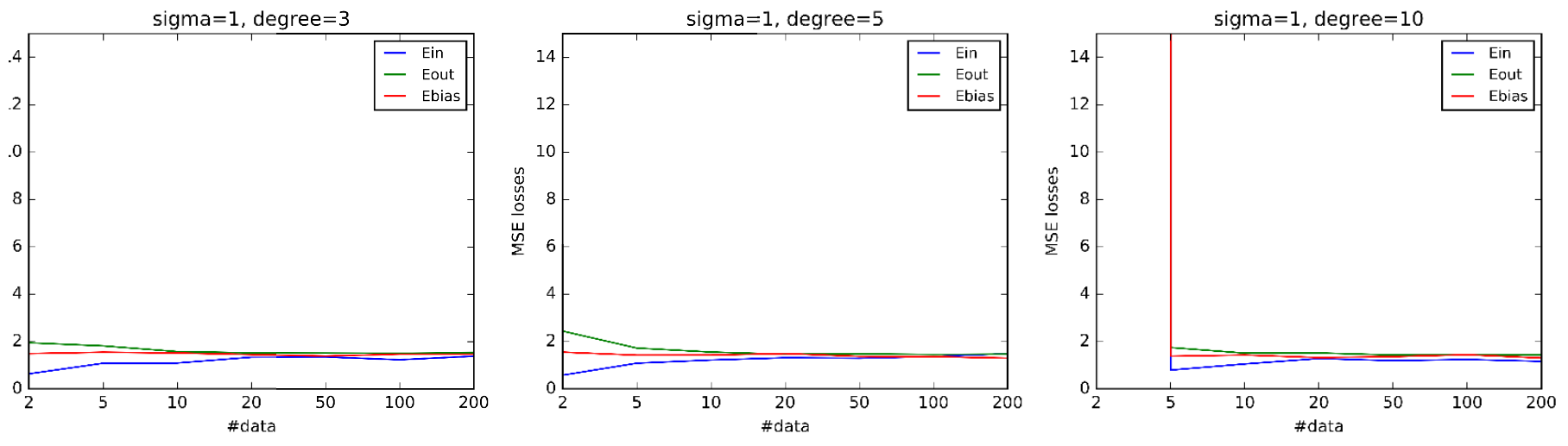
Adding regularization term also makes different to these plots – makes the decrease of and much less steep, as it controls that generalization gap when data size N is small:

Before applying regularization:



**Figure 8-1. Pattern of change before regularization**

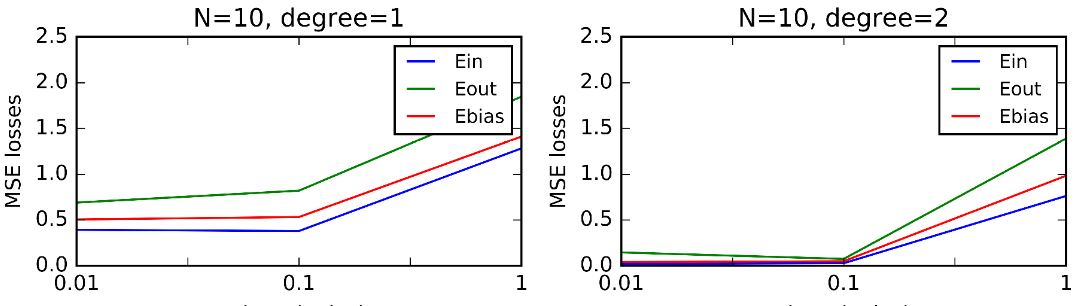
After applying regularization:

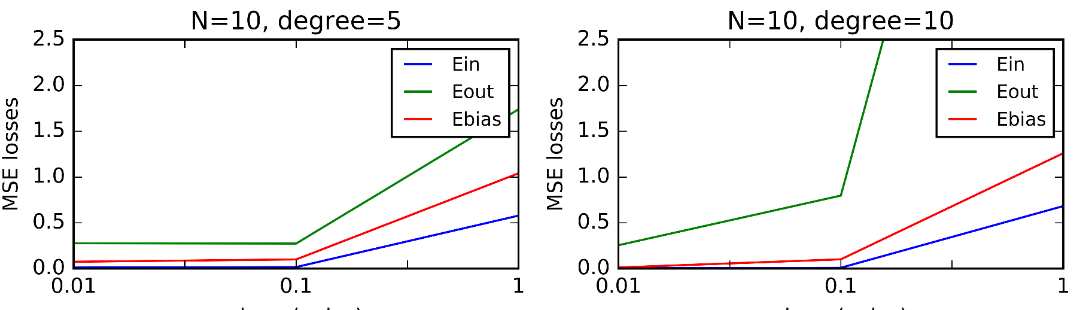


**Figure 8-2. Pattern of change before regularization**

* 1. **Effect of (data noise)**

The first pattern that can be found regarding change is that adding noise to data set (both training set and test set) will increase , and , which is obvious. The second pattern is that noise can also push up generalization gap as it increases test error () with a faster rate than training error (). In situation where overfitting is likely to occur (high degree and low training size), the second pattern is much more obvious. On the other hand, when the models are underfitting, there will not be an increase in generation gap as we increase noise. Here is the example:

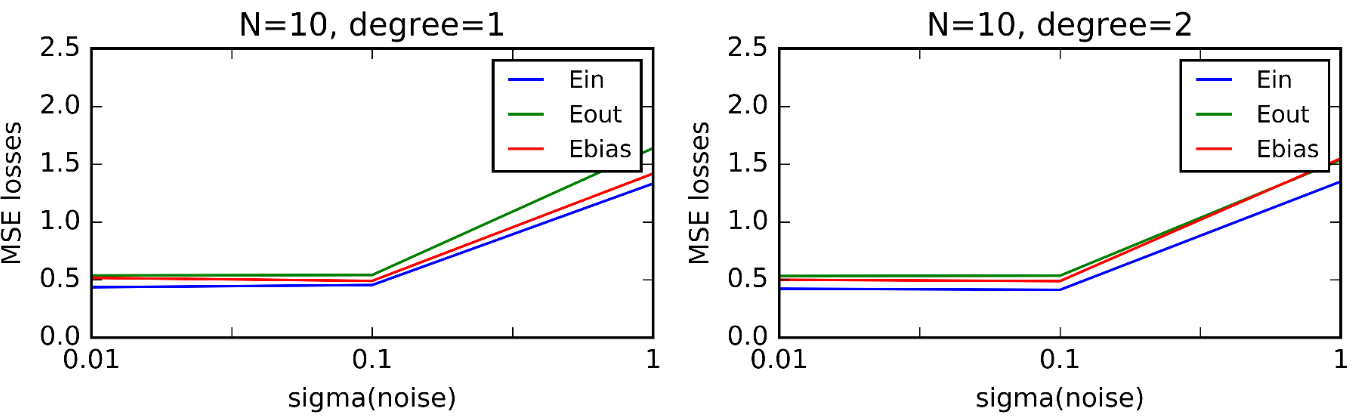




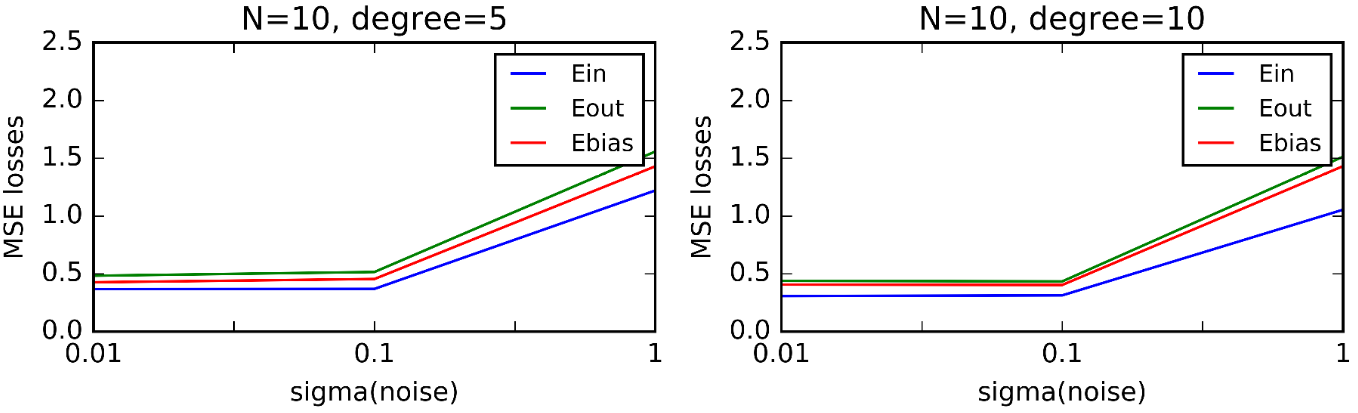
**Figure 9. Pattern of change before and after overfitting**

We can see in the figure that when the model is underfitting, , stay in relative the same distance (figure1) when increases from 0.01 to 1. When the model is slightly overfitting, there is still no apparent difference between and , while the generalization gap starts to increase as we go from to . When overfitting is severe enough, even adding a little noise can drive up generalization gap significantly (the lower-right one).

After applying regularization to the models, all the difference between and become much smaller, as shown in the figure:



**Figure 10-1. Pattern of change before regularization**



**Figure 10-2. Pattern of change after regularization**

The three line **,**  and become closer after the regularization, which is a sign of less model variance. However, we can also see that the overall MSE -- the models’ bias goes up, a bias-variance trade off.

# 