P556 Homework 3

Fall 2020

Due on Oct 28th, 11:59pm

Question 1: Bias Variance Decomposition (40 points)

Recall that the squared error can be decomposed into bias, variance and noise:

$$\underbrace{\mathbb{E}[(h_D(x)-y)^2]}_{\text{Error}} = \underbrace{\mathbb{E}[(h_D(x)-\bar{h}(x))^2]}_{\text{Variance}} + \underbrace{\mathbb{E}[(\bar{h}(x)-\bar{y}(x))^2]}_{\text{Bias}} + \underbrace{\mathbb{E}[(\bar{y}(x)-y(x))^2]}_{\text{Noise}}$$

We will now create a data set for which we can approximately compute this decomposition. The function toydata.py generates a binary data set with class 1 and 2. Both are sampled from Gaussian distributions:

$$p(\vec{x}|y=1) \sim \mathcal{N}(0, I) \text{ and } p(\vec{x}|y=2) \sim \mathcal{N}(\mu_2, I), \tag{1}$$

where $\mu_2 = [2; 2]^\top$ (the global variable OFFSET = 2 regulates these values: $\mu_2 = [\text{OFFSET} ; \text{OFFSET}]^\top$).

You will need to implement four functions: computeybar.py, computehbar.py, computevariance.py and biasvariancedemo.py.

(a) Noise (computeybar.py): First we focus on the noise. For this, you need to compute $\bar{y}(\vec{x})$ in computeybar.py. With the equations, $p(\vec{x}|y=1) \sim \mathcal{N}(0, I)$ and $p(\vec{x}|y=2) \sim \mathcal{N}(\mu_2, I)$, you can compute the probability $p(\vec{x}|y)$. Then use Bayes rule to compute $p(y|\vec{x})$.

Note: You may want to use the norm probability density function, which you can define by yourself or you can directly use some package to call this function if you find some. With the help of computeybar.py you can now compute the "noise" variable within biasvariancedemo.py. Here is a plot that what your data is supposed to be like in Figure 1:

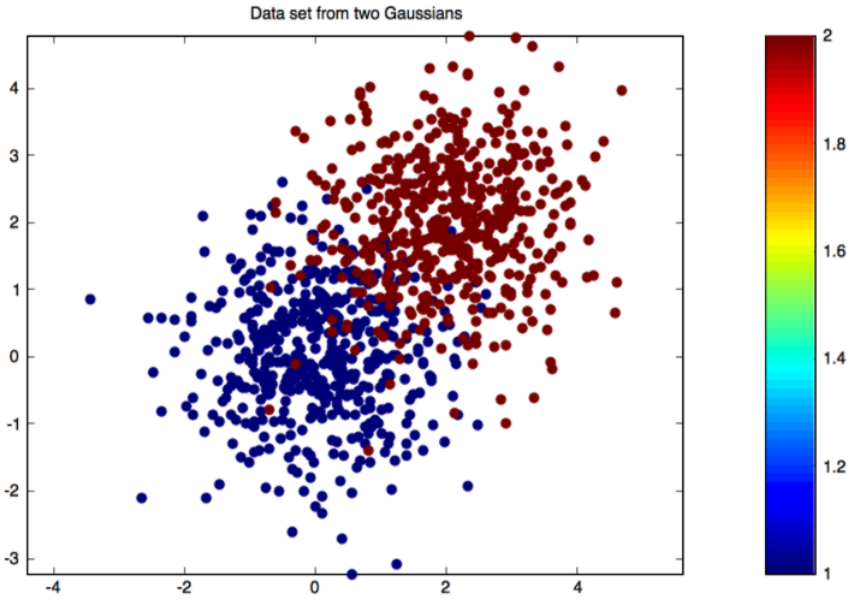


Figure 1: Question 1

(b) Bias (computehbar.py): For the bias, you will need $\bar{h}$. Although we cannot compute the expected value $\bar{h} = \mathbb{E}[h]$, we can approximate it by training many h_D and averaging their predictions. Edit the file computehbar.py: average over n_models different h_D, each trained on a different data set of n inputs drawn from the same distribution. Feel free to call toydata.py to obtain more data sets.

Note: You can use ridge regression for h_D. With the help of computehbar.py you can now compute the "bias" variable within biasvariancedemo.py.

(c) Variance (computevariance.py): Finally, to compute the variance, we need to compute the term $\mathbb{E}[(h_D - \bar{h})^2]$. Once again, we can approximate this term by averaging over n_models models. Edit the file computevariance.py.

With the help of computevariance.py you can now compute the "variance" variable within biasvariancedemo.py.

(d) Demo (biasvariancedemo.py): In this function, you need to implement a plotting function that how the error decomposes (roughly) into bias, variance and noise when regularization constant λ increases. You can see the trend if you did everything correctly.

Note: You can set a training dataset with a size of 500, for a really big dataset you can set the size as 100000. You can try average over 25 models. But all these parameters you can change freely.

The bigger number for the number of models and/or the training dataset, the better your approximation will be for $\mathbb{E}[h]$ and $\mathbb{E}[(h_D - \bar{h})^2]$).

If you get everything correct, you should get some plot like this:

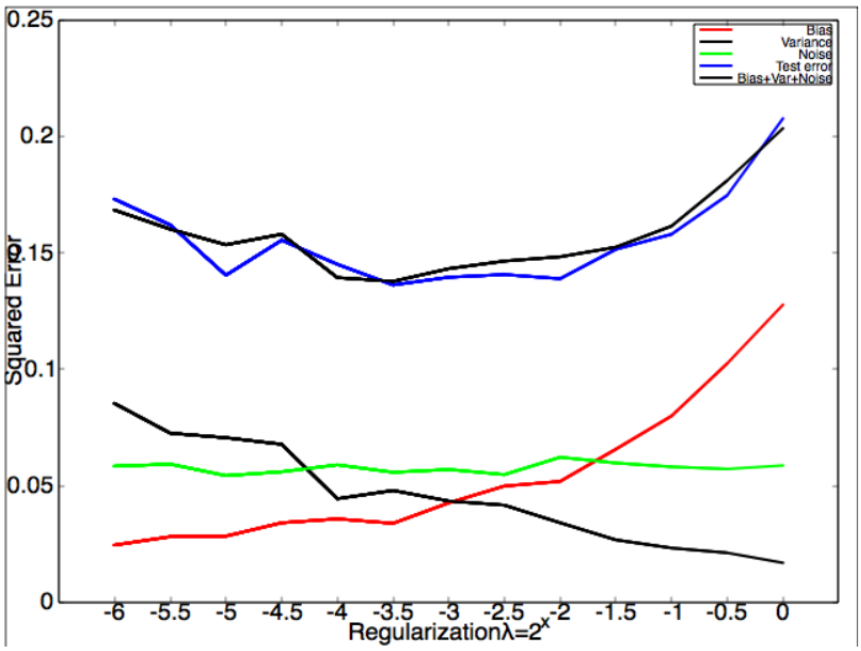


Figure 2: Question 1

Question 2: SVM (30 points)
Here is an visualization of a set of 3600 points in 2D space ($hw3\_data2.txt$).

- Which classifier would be able to achieve better performance on this distribution? Justify your choice.
- Implement your chosen classifier and report your accuracy.

- Produce a plot that shows your final classifier as a dotted line, along with the original data points. You can make the plot either in the original space or feature space.

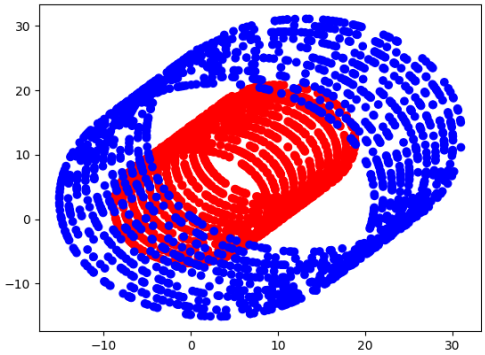
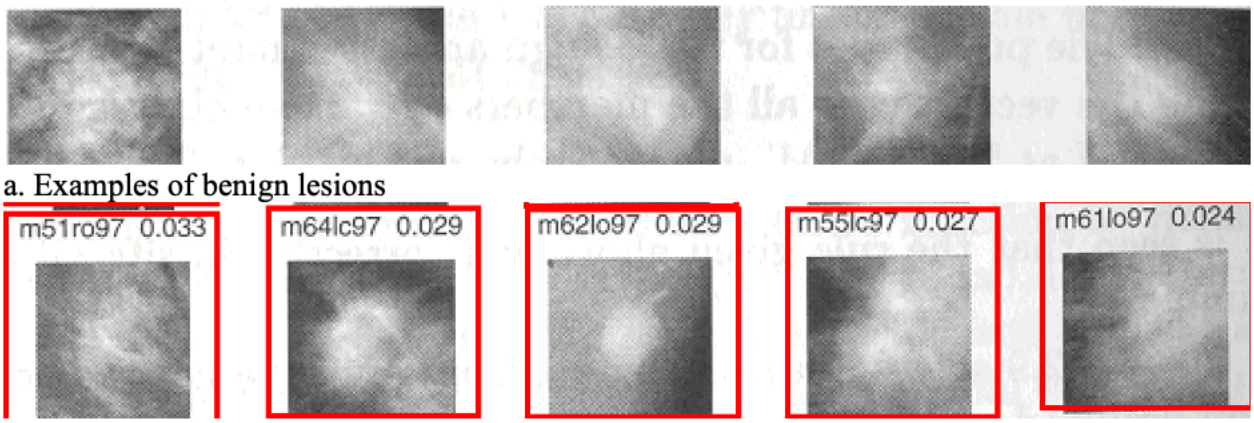


Figure 3: Question 2

Question 3: ROC Curves (30 points)

We want to generate a classifier that is able to distinguish between malign and benign tumors in mammography. Assume that malign as well as benign tumors appear equally often. Tumor images can be found in figure 1a and 1b. Notice that the original images of the lesions have been normalized to a standard size. Assume you already have a method which segments the tumor. The physicians give us a table (see figure 1c) with the properties that they use to distinguish benign from malign tumors. Answer the following questions:

- Give three features that would allow you to distinguish between malign and benign tumors. Describe shortly how to calculate them given the images in figure 1a and 1b, or segmented images. Describe shortly why you chose these features.
- Given these features design a classifier. Assume that the cost of misclassifying a malign tumor is 2 times the cost to misclassify a benign. Explain what other assumptions you are making (if any).
- We want to compare our computer aided diagnoses system with the performance achieved nowadays in clinical practice by radiologists and a CAD system that is sold and available at the moment in the clinic. The system at the clinic gives us as a result a probability that a tumor is malign. The comparison is done using ROC curves. Three ROC curves which are shown in figure 2 are generated. ROC(1) corresponds to the radiologist performance, ROC(2) corresponds to the already existing system and ROC(3) corresponds to the system we just design.
  - How could we generate ROC(1)? Describe the steps and the information that you need to gather to obtain ROC(1).
  - How could we generate ROC(2) and ROC(3)? Describe the steps and the information that you needed to gather to obtain ROC(2) and ROC(3).
  - Given the ROC curves how can you compare the performance of the different systems?
  - What can you say of the performance of the radiologists in comparison to the CAD systems?
  - What can you say about the performance of the CAD systems in relation to each other?
- Imagine that you find out looking at the feature space that each class seems to cluster in more than one location in the feature space. Would that have influence in your choice of the classifier? Why? If you would choose another classifier which one? Which assumptions are you making? Describe how you would implement it? What are the disadvantages of this method compared to the one you already had?



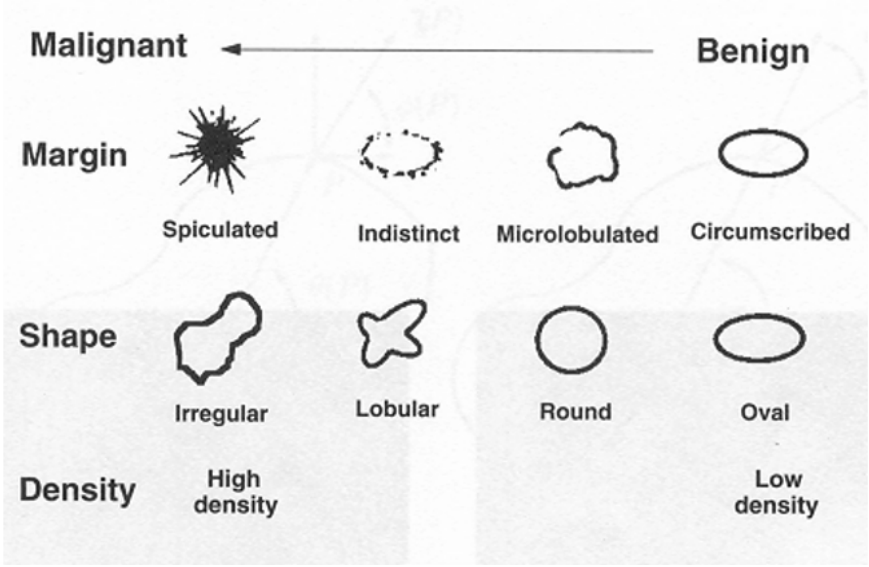


Figure 4: Question 3

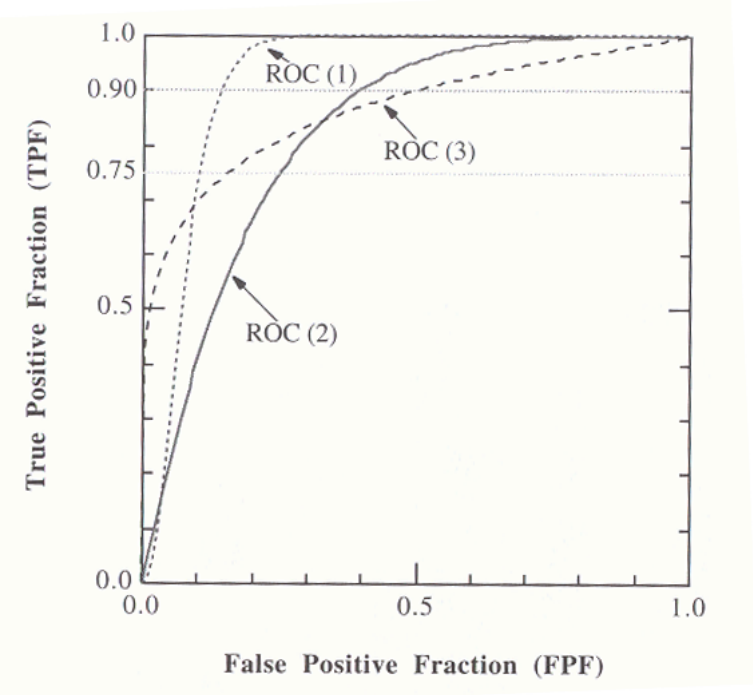


Figure 5: Question 3