Artificial Intelligence
Assignment 3
Machine Learning

Yingtao Xu
yx2318

June 26th, 2016

Problem 1: Linear Regression

In this problem, linear regression with multiple features will be practiced by using gradient descent. The data used is from CDC growthchart with girls' age, weight and height.

Data Preparation and Normalization

a). Firstly, data will be preprocessed for linear regression. It will be loaded and then added the intercept column.Then, each feature will be scaled by its standard deviations and its mean will be set to zero as the formula give. After that, the new data can be stored by features and labels separately for later use. The means and standard deviations can be seen in Figure 1 below:

```
Data: Mean and Standard deviation for each feature:
          Mean            Stdev
age    5.21050632911   1.89948154068
weight  18.3066043038   6.11745394303
```

Figure 1: Mean and Standard Deviation for each feature

b). Also, part of the scaled data can been in Figure 2:

	intercept	scaledAges	scaledWeights
0	1	-1.690201	-1.323481
1	1	-1.669143	-0.855008
2	1	-1.621762	-1.121322
3	1	-1.579645	-0.632087
4	1	-1.537528	-0.933458
5	1	-1.490147	-1.272867
6	1	-1.448030	-0.884589
7	1	-1.405913	-1.011807
8	1	-1.358532	-0.252679
9	1	-1.316415	-1.160166
10	1	-1.274298	-0.787497
11	1	-1.226917	-1.054501
12	1	-1.184800	-1.105572
13	1	-1.142684	-0.883437
14	1	-1.095302	-0.861663
15	1	-1.053185	-1.050206
16	1	-1.011069	-1.031435
17	1	-0.963687	-1.058740
18	1	-0.921571	-0.916235
19	1	-0.879454	0.209381
20	1	-0.832072	-0.301706
21	1	-0.789956	-0.506648

Figure 2: Scaled data with intercept (partially)

Gradient Descent

a).

After the preparation of the data, gradient descent algorithm has been designed for linear regression for the data. The detailed implementation of the algorithm please refer to the code.

Figure 3 and Figure 4 give the plots for the risk function with respect to different learning rates.

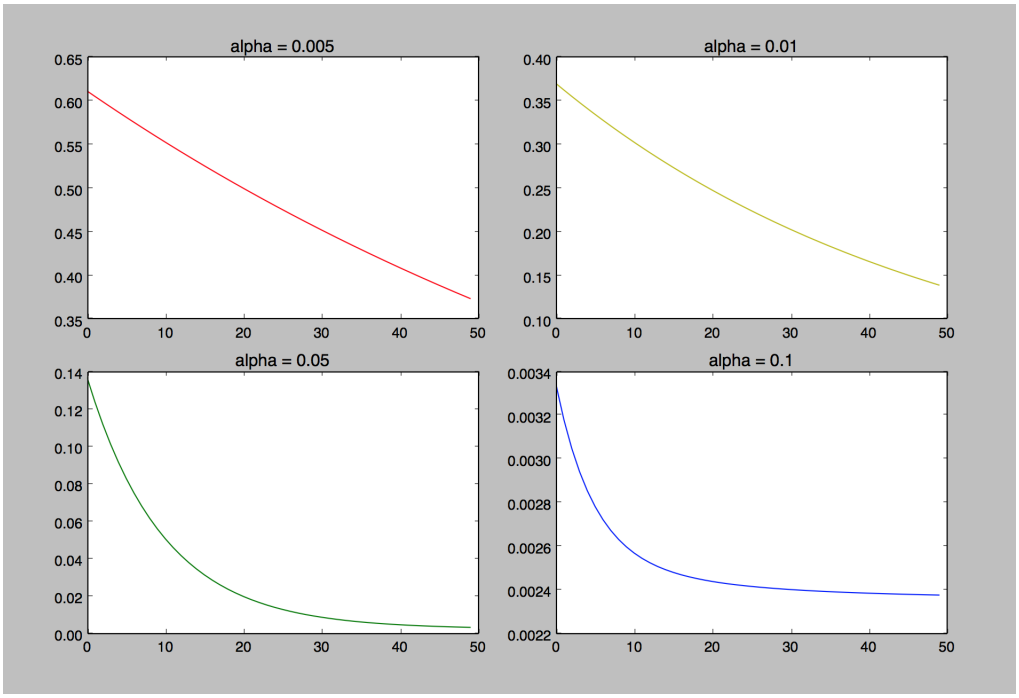

Figure 3: Risk function with respect to different learning rates

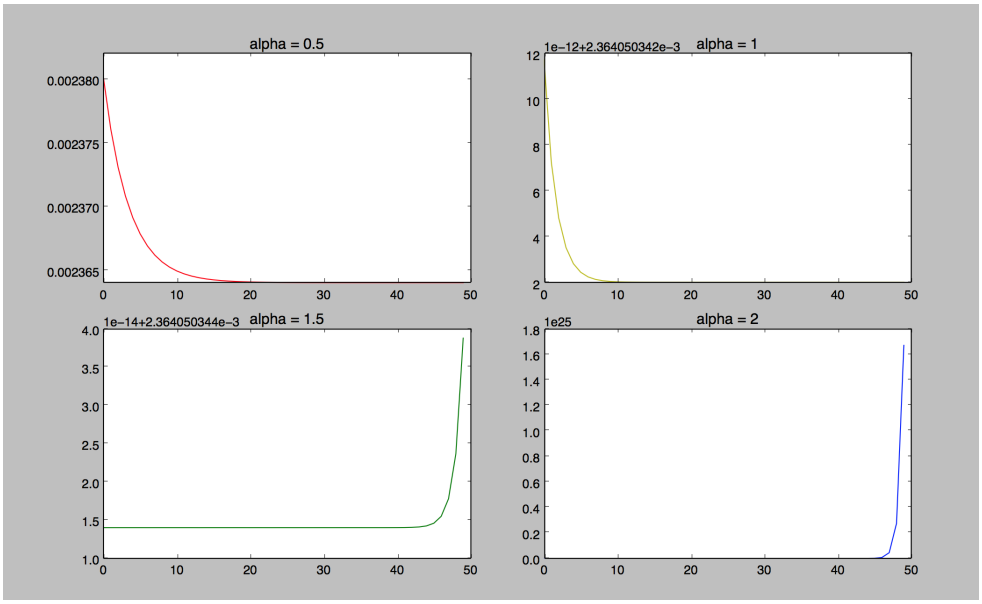

Figure 4: Risk function with respect to different learning rates cont's

b).
From the plots above, it can be seen that with very small α, the risk function cannot converge. When α becomes slightly larger, risk function can converge but very slowly. For example, as α is 0.1, after 50 iteration it seems to converge finally and when α is 0.5, it converges after 20 iterations. As α keeps increasing, the risk function converge faster. However, when α is too large the risk function starts bounce up.

c) and d).
It can be seen that when α is 1, the converge speed is very fast using only several iteration and there is no bounce up of the risk function.
Thus, with the consideration of converge speed and avoidance of the risk bounce back, the best α chosen for iteration 50 is 1. With the α chosen in part c), the β and the height prediction for the girl with 5-year age and 20-kg weight can be seen in Figure 5.

```
************************************************************
c):
beta  0
[ 1.09646081]
beta  1
[ 0.12861721]
beta  2
[ 0.00140248]
************************************************************
d):
************************************************************
height prediction:
[ 1.08259529]
************************************************************
```

Figure 5: β and the height prediction with $\alpha = 1$

Problem 2: Classification

In this section, sklearn will be applied as the main tool to test one of the most popular machine learning problems: classification. The classifiers tested in this section includes SVM with different kernels, logistic regression, and decision trees.

The data used in this section is the chessboard.csv data

The scatter plot of the data can be seen in Figure 6:

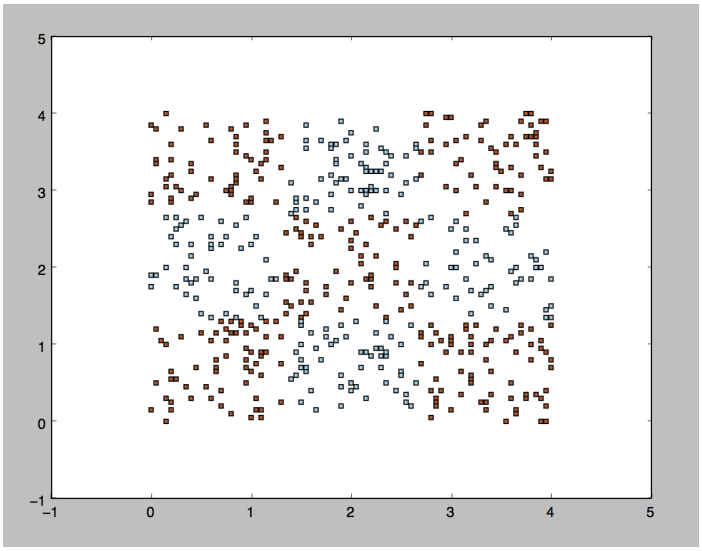

Figure 6: Scatter plot of the chessboard data

The main steps for the classification test shown as the followings:

- Step 1: Split the dataset into training data (60% of the raw data) and test data (40% of the raw data), which can be achieved by either manually or by the sklearn tool for cross validation: cross_validation.train_test_split.

- Step 2: Use the kfold cross validation on the training data (60% ones) in sklearn to evaluate the parameters with different combination for each classifier.

- Step 3: Choose the best combination of the parameters and apply them onto both training data the test data and plot the decision boundaries for each classifier.

SVM

Linear Kernel

The parameter changed for linear kernel is C, the penalty for the error term in SVM. The range tested for C is 1 to 20. Since the chessboard data is impossible to find any line to separate, no matter how to change the C, the accuracy is the same and always be very low. The sample results can be seen in Figure 7 and Figure 8 which show the C changing from 1 to 5 and from 16 to 20. Thus, the parameter chosen for linear kernel SVM is just 1.

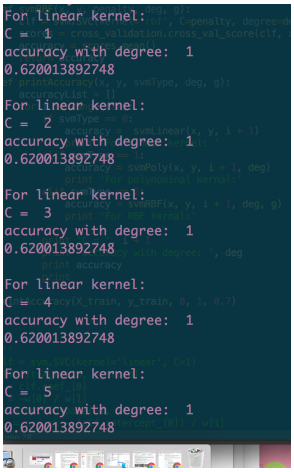

Figure 7: Accuracy for Linear kernel SVM (C = 1 to 5)

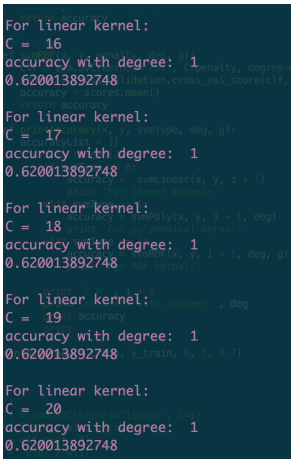

Figure 8: Accuracy for Linear kernel SVM (C = 16 to 20)

Polynomial Kernel

The parameters changed for polynomial kernel are C and degree.

The range test for C is still from 1 to 20 and the range for degree is 1 to 5. Figure 9 and 10 show the accuracy with degree 1.

Figure 11 shows the accuracy with degree 3 and Figure 12 to Figure 15 show results with degree 4.

Figure 16 and Figure 17 are the results for degree 5.

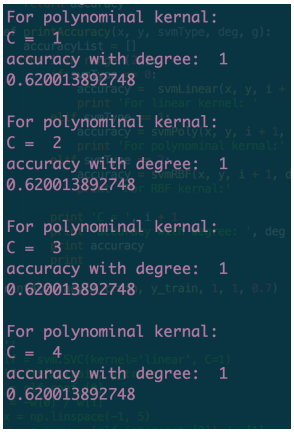

Figure 9: Accuracy for Polynomial kernel SVM (degree = 1)

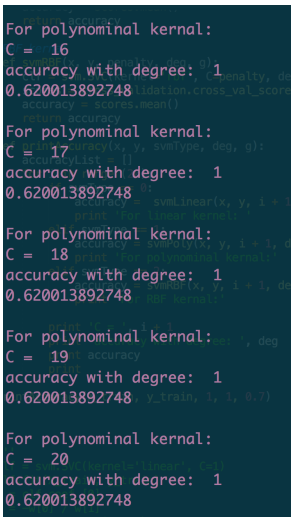

Figure 10: Accuracy for Polynomial kernel SVM (degree = 1)

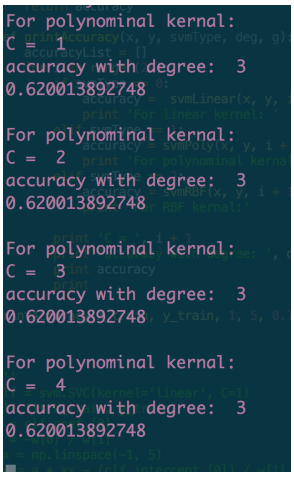

Figure 11: Accuracy for Polynomial kernel SVM (degree = 3)

From the results, it can be seen that the change of C affects little for polynomial case and the degree is the main to affect the accuracy is the degree, as the degree increases, the accuracy improves.

Since when degree is too large the running time is too long, the maximal degree is chosen for this test is just 5 so the accuracy is not very high but we still can expect that the higher degree may give better results. It is reasonable because the chessboard data is very complex to be classified by low degree polynomial line.

Also, there is another finding is that when the accuracy with a fixed degree reaches a certain value, as the increase of penalty, the accuracy de-

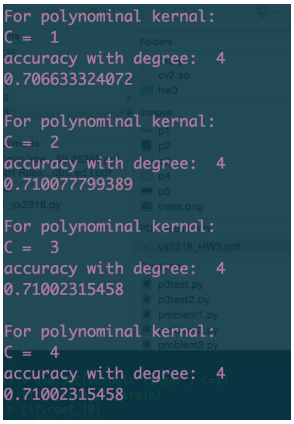

Figure 12: Accuracy for Polynomial kernel SVM (degree = 4)

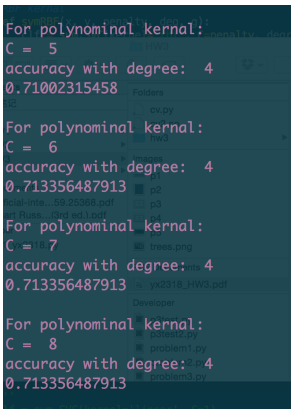

Figure 13: Accuracy for Polynomial kernel SVM (degree = 4)

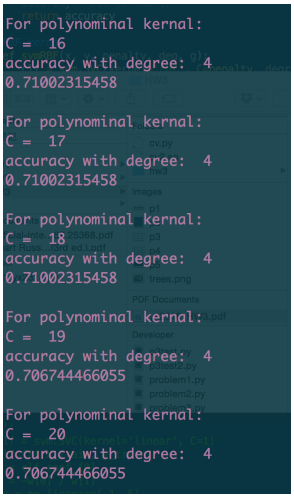

Figure 14: Accuracy for Polynomial kernel SVM (degree = 4)

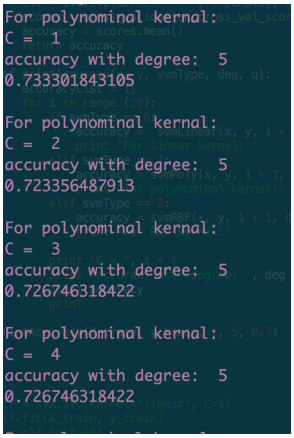

Figure 15: Accuracy for Polynomial kernel SVM (degree = 5)

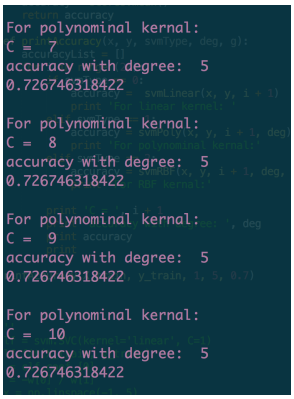

Figure 16: Accuracy for Polynomial kernel SVM (degree = 5)

creases. This is interesting because it reflects that as the degree increased to a certain level and fit the data the increase of penalty may give negative side effect for the fitting, maybe overfitting, which should be avoided.

Thus, based on the results, the parameter combination for polynomial case is $C = 1$ and $degree = 5$.

RBF Kernel

For RBF kernel parameter selection, the parameters includes not only the penalty and degree but also gamma. The range for is also 1 to 20 and degree is also 1 to 5. The range for gamma is 0.5 to 0.9. According to the tests, the most significant parameter for RNF kernel is gamma. C also has effects on accuracy but degree seems have little effect.

The selected results in the following figures can reflect this finding.

Firstly, Figure 17 to Figure 19 are the results for $degree = 1$ and $gamma = 0.5$

Then, Figure 20 and Figure 21 show results for degree changed to 3 and 5 with the same gamma.

Next, Figure 22 and Figure 25 show the results for gamma is 0.7 and degree changed from 1 to 5.

Finally, Figure 26 shows the results for gamma is 0.9.

From the results, some significant findings can be concluded for RBF kernel as below:

- Gamma is the most significant factor for the accuracy. As the value of gamma increases, the accuracy increases.

- Degree is the least significant factor for accuracy. Actually, for the fixed gamma, the change of degree seems have no effect for the accuracy.

- C can also affect the accuracy. With fixed gamma and degree, as the value of C increases, the accuracy increases also but like before, when C's value reaches a certain value the increase of C will cause decrease of the accuracy.

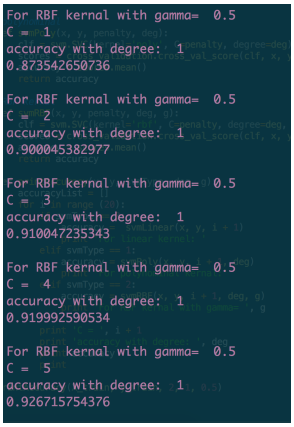

Figure 17: Accuracy for RBF kernel SVM (degree = 1 and gamma = 0.5)

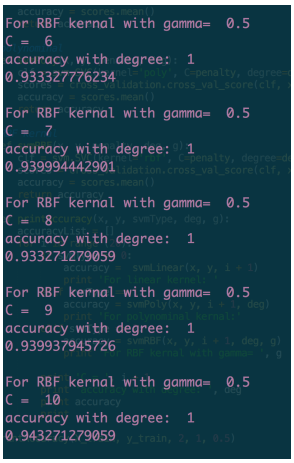

Figure 18: Accuracy for RBF kernel SVM (degree = 1 and gamma = 0.5)

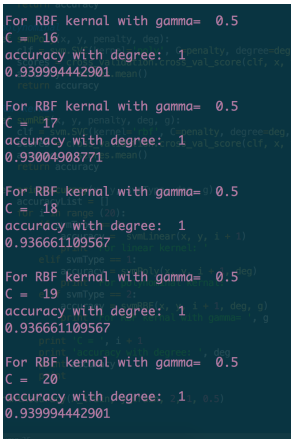

Figure 19: Accuracy for RBF kernel SVM (degree = 1 and gamma = 0.5)

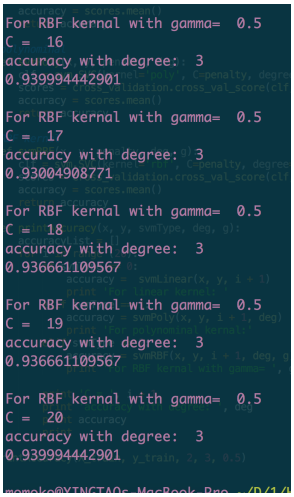

Figure 20: Accuracy for RBF kernel SVM (degree = 3 and gamma = 0.5)

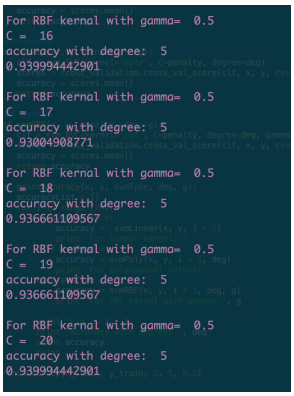

Figure 21: Accuracy for RBF kernel SVM (degree = 5 and gamma = 0.5)

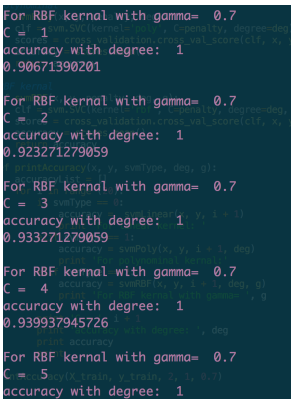

Figure 22: Accuracy for RBF kernel SVM (degree = 1 and gamma = 0.7)

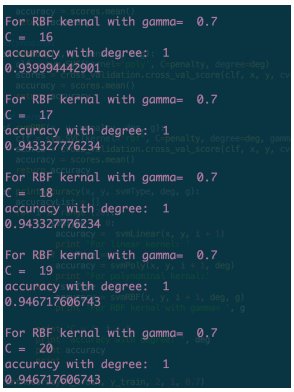

Figure 23: Accuracy for RBF kernel SVM (degree = 1 and gamma = 0.7)

Figure 24: Accuracy for RBF kernel SVM (degree = 3 and gamma = 0.7)

Figure 25: Accuracy for RBF kernel SVM (degree = 5 and gamma = 0.7)

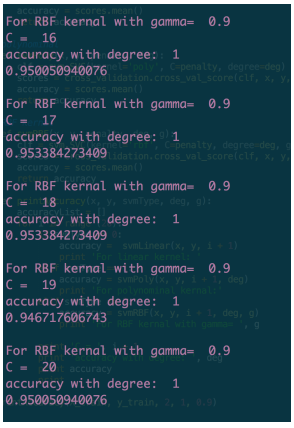

Figure 26: Accuracy for RBD kernel SVM (degree = 1 and gamma = 0.9)

The results are reasonable. In the first place, as the chessboard data are very complex and two classes cross each others, with larger deviation, that is larger gamma, can give a better fitting of the data. The effect of the penalty is the same idea as discussed before: it may not be good to have a C as large as possible because of overfitting. Thus, based on the results, the parameters chosen for RBF kernel SVM is $C = 17$, $degree = 1$, and $gamma = 0.9$.

Logistic Regression

After the test for SVM with different kernel, another classifier, logistic regression, will be tested in this section. Figure 27 and Figure 28 show the results as the parameter C changing from 1 to 20 and Figure 29 give an extra test for C is 500.

Actually, the range for C has been tested from 0.5 to 500 and the accuracy is always the same. This can be explained because logistic regression is still a linear model so it cannot fit very well with the chessboard data which are nonlinear at all.

Thus, the parameter chosen for logistic regression is just C which is 1.

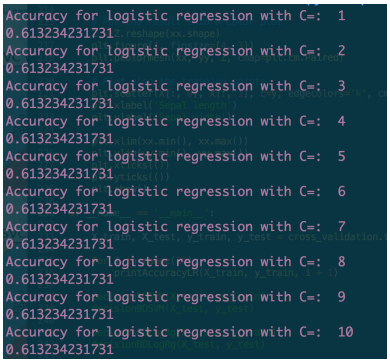

Figure 27: Accuracy for Logistic Regression (C from 1 to 10)

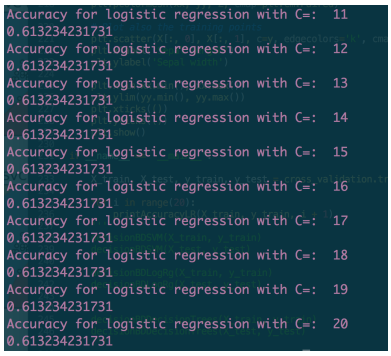

Figure 28: Accuracy for Logistic Regression (C from 10 to 20)

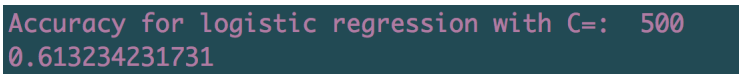

Figure 29: Accuracy for Logistic Regression (C = 500)

Decision Trees

The parameters for the final classifier, decision trees, will be tested in this section. Since there is so many parameters can be changed and tested for decision trees. In this section only max_depth and min_sample_split have been shown for results. Figure 30 to Figure 31 are the selected results.

Judging from the results of the test, it can be seen that as the max depth for the decision trees increases, the accuracy increases but the change of the number for splitting at each level seems have no effect on the results. Actually, using the default value for max depth, it will expand until all leaves are pure which will give the best results. Similarly, if default values are used for other parameters, the decision trees will make the best effort to do the classification.

Thus, all parameters used are the default ones for decision trees to get the best results.

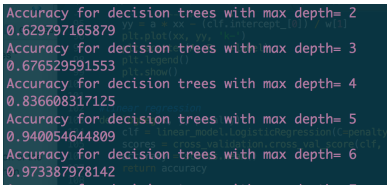

Figure 30: Accuracy for Decision Trees with different max depth

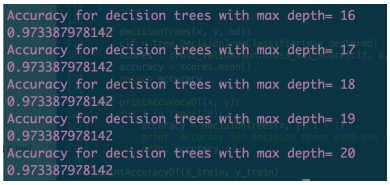

Figure 31: Accuracy for Decision Trees with different max depth

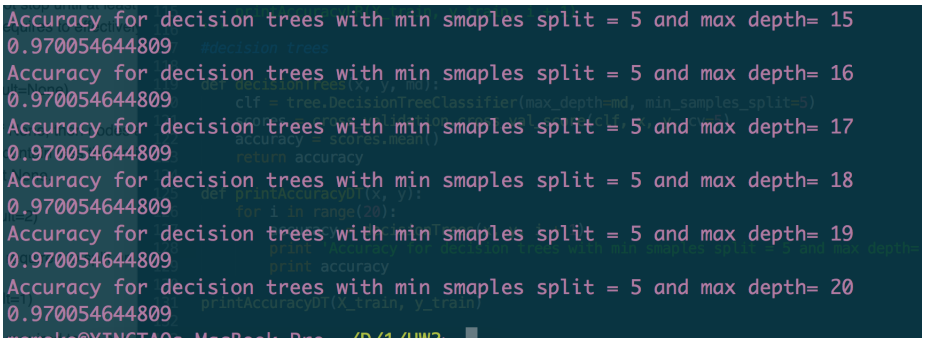

Figure 32: Accuracy for Decision Trees with different max depth (split number is 5)

Decision Boundaries For Diffrent Classifiers

Now, with the evaluated results for different parameters of each classifier, decision boundaries can be drawn.
Each classifier will be shown with the decision boundaries for both training data and test data and all classifiers use the same set of training data and test data for comparison.
Figure 33 and Figure 34 will show the decision boundaries SVM with different kernels. Figure 33 is the one using training data while Figure 34 is the one using test data.

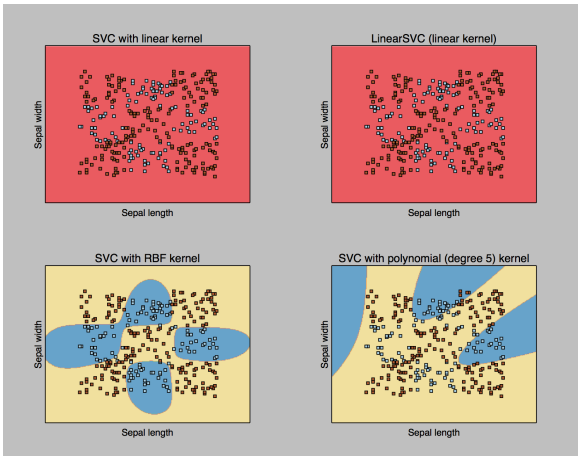

Figure 33: Decision boundaries for SVM with different kernels (Training data)

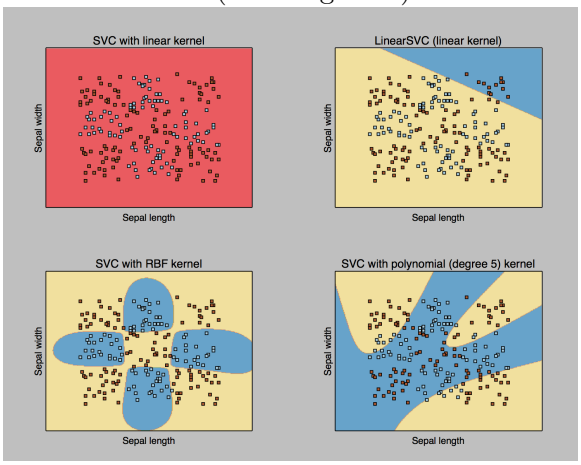

Figure 34: Decision boundaries for SVM with different kernels (Test data)

Figure 35 and Figure 36 are the decision boundaries for logistic regression. Similarly, Figure 35 is for training data and 36 is for test data.

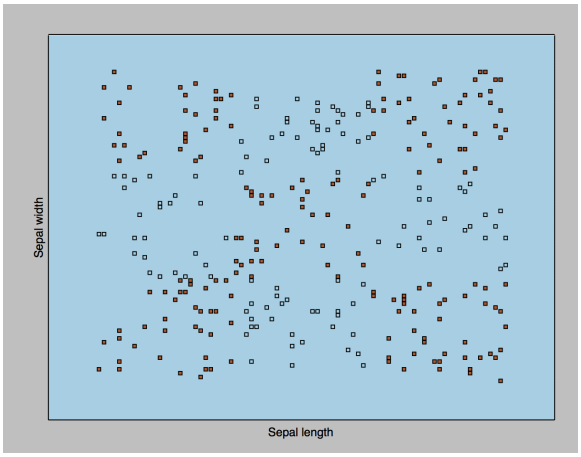

Figure 35: Decision boundaries for Logistic regression (Training data)

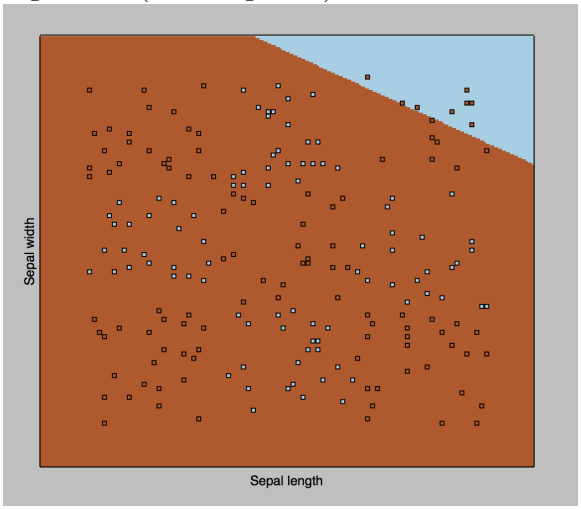

Figure 36: Decision boundaries for Logistic regression(Test data)

Figure 37 and Figure 38 are the decision boundaries for decision trees. Also, Figure 37 is for training data and 38 is for test data.

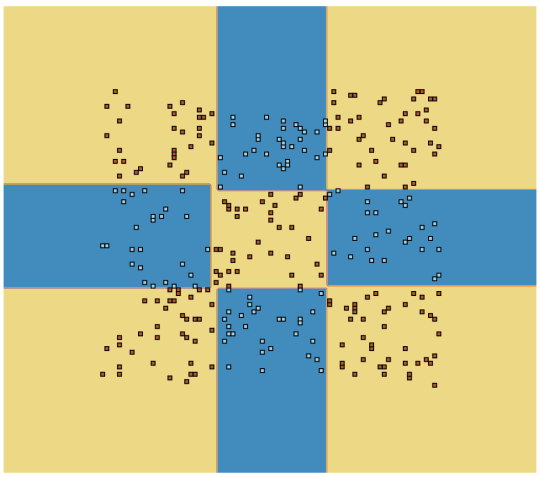

Figure 37: Decision boundaries for Decision trees (Training data)

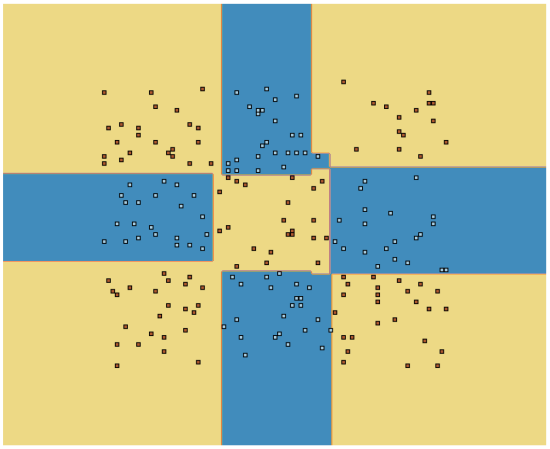

Figure 38: Decision boundaries for Decision trees(Test data)

Based on the decision boundaries drawn with different classifiers, it is clear that the linear model is not suitable for the chessboard data. No matter using the SVM with linear kernel, SVM with degree 1 polynomial model or the logistic regression model, all of them cannot give any classification at all because all of them are linear classifiers.

Polynomial SVM with higher degree gives a sightly better result but the classification is far away from an acceptable level as a good classifier for the data.

RBF SVM actually gives a much better classification for the chessboard data but there are still some misclassified data.

The best one for the chessboard data is the decision trees because as the results show, all data are classified correctly.

Therefore, the best model for chessboard is decision trees. The main reason is that decision trees do not need any linear assumptions in the data. Thus, it performs well for the nonlinear data like chessboard data in this homework.

Actually, classes are not always separable by a hyperplane like the data used in this homework. Linear model functions badly for this case. Non linear decision function usually works well for this case like using polynomial model or exponential model. But the best is to choose the one does not depend on the linearity at all, like decision trees.

Problem 3: Clustering for Image Segmentation

In this problem, unsupervised classification will be tested. The main algorithms are KMeans clustering and spectral clustering.

Kmeans

In the first section, KMeans clustering will be used to do color clustering so that the most significant colors in an image can be found.The image used in this section is trees.png.
The main steps can be seen as below:

- Load the image and store the image in to a numpy array
- Reshape the image array for clustering
- Do Kmeans on the reshaped data and find the centroids (colors).
- Assign each pixel on the original image data to its nearest centroid (color)
- Plot the new image

Three representative k values chosen are 3, 5, and 10. Figure 39 is the original image. Figure 40, 41 and 42 are the images after applying KMeans clustering with $k = 3$, 5, and 10 respectively.

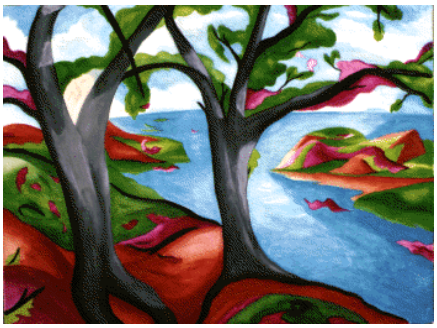

Figure 39: Original trees image

Figure 40: Image after KMeans (k=3)

Figure 41: Image after KMeans (k=5)

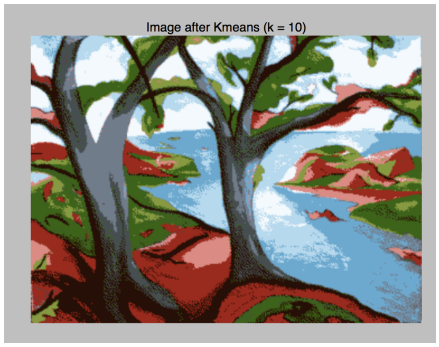

Figure 42: Image after KMeans (k=10)

From the results, it can be seen that when k is very small, like k is 3, it can capture the most significant colors and give the some outline of the items in the picture . When k is larger, more features of the image can be shown. When is large enough, it will be very close to the original image.

The results imply that KMeans clustering can be used as a tool for image segmentation because it can help to capture the most significant features in a picture. Thus it is reasonable why it can be used to compress images or used for preprocessing of images for further image processing.

Spectral Clustering

In this section, the main idea is to compare KMeans and spectral clustering. The datasets used is generated data by sklearn([3]). The results are shown in Figure 43 and Figure 44.

From the results, it can be seen that spectral clustering do a better classification for the case where data is connected. The main reason why spectral clustering can do better is because it projects the data into a lower-dimensional space so that they are easy to separate ([4]).

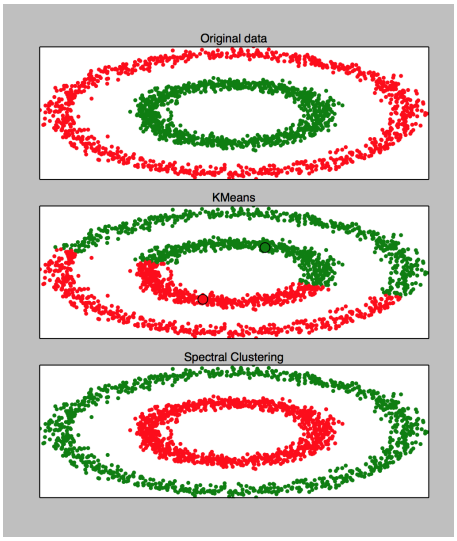

Figure 43: KMeans VS Spectral Clustering

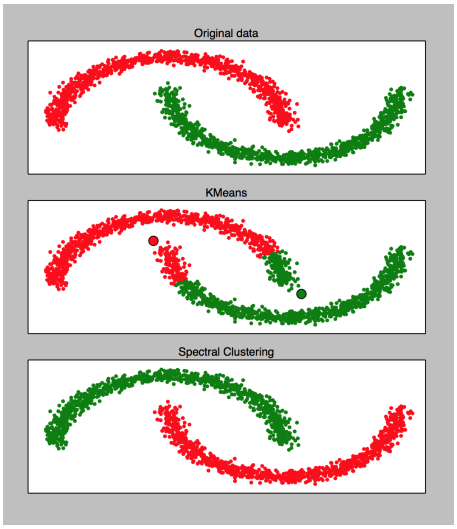

Figure 44: KMeans VS Spectral Clustering

Bibliography

[1] Scikit-Learn: http://scikit-learn.org/stable/

[2] KMeans color clustering: http://scikit-learn.org/stable/auto_examples/cluster/plot_color_quantization.html

[3] Spectral Clustering VS Kmeans plots: http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

[4] Spectral Clustering: https://www.cs.cmu.edu/~aarti/Class/10701/slides/Lecture21_2.pdf