Homework 3 for Machine Learning. Fall 2015

Chengjun Yuan cy3yb@virginia.edu

1 naïve Bayes Classifier for Text-base Movie Review Classification.

Q5.

```
thetaPos = [  9.91497078e-01   1.05483349e-03   5.62220897e-04   1.04412452e-03
  1.03341555e-03   9.28110688e-05   7.15716127e-04   4.31928436e-04
   4.94397424e-04   6.06841604e-05   6.24689886e-05   5.71145039e-05
   1.26722805e-04   1.97045038e-03   7.96033398e-04]
thetaNeg = [  9.89128337e-01   8.09951861e-04   3.70149990e-04   7.02488960e-04
   5.45274717e-04   2.58706983e-05   6.62687886e-04   1.99005371e-04
   1.48059996e-03   3.58209668e-04   3.52239507e-04   3.44279292e-04
   4.79602944e-04   3.03682196e-03   1.50448061e-03]
```

Q6.

MNBC classification accuracy = 0.695

Q7.

Sklearn MultinomialNB accuracy = 0.695

Q9.

Directly MNBC tesing accuracy = 0.695

Q11.

```
thetaPosTrue = [ 0.9985755   0.43162393  0.2977208   0.49287749  0.43447293  0.06552707
  0.37179487  0.21225071  0.27207977  0.04558405  0.03988604  0.04273504  0.08547009
0.54273504  0.27777778]
thetaNegTrue = [ 0.9985755   0.35185185  0.22222222  0.35470085  0.27492877  0.01851852
  0.32621083  0.11538462  0.51424501  0.19230769  0.18660969  0.21082621  0.25213675
0.68803419  0.3988604 ]
```

Q12.

BNBC classification accuracy = 0.67

2. Content-based image classification through Supervised Classifier in Python

2.1. Decision Tree

Here, three parameters are chosen. They are “criterion”, “splitter”, and “max_features”. Their values and function are listed below:

Parameters	Values	Function
criterion	“entropy”	The function to measure the quality of a split.
splitter	“best”	The strategy used to choose the split at each node.
max_features	“sqrt”	The number of features to consider when looking for the best split.

The test error is **0.194818**.

2.2. K nearest Neighbors

Here, five values of K for the K nearest neighbors (weights = ‘distance’) are tried. Their test error and training error are listed below:

K	Test error	Training error
2	0.05630	0.0
4	**0.05282**	**0.0**
6	0.05481	0.0
8	0.05531	0.0
9	0.06079	0.0

K = 4 is chosen because it brings about the smallest test error.

KNN is a very simple and suitable method to classify this data set because it provides low test error rate. As we know, image points of different handwritten digits have different characteristics in distance between strokes. The “distance” weighting assigns weights proportional to the inverse of the distance from the query point. The point with long distance from the query point will get small weights, so that the nearer neighbors contribute more to the average than the more distant ones. This makes the classification better.

2.3. Neural Net – Perceptron

Test error is **0.1036**. Since perception is a type of linear classifier, which has limited accuracy when the problem is nonlinear and its class boundaries cannot be approximated well with linear hyperplanes. Apparently, the nonlinear classifier is more suitable to the problem in this homework.

2.4. Support Vector Machine

Kernel	C	Test Error
linear	1	0.0737
rbf	1	0.0578
poly	1	0.0533

In this problem, SVM with poly kernel is a suitable method to classify the data. Because SVM is very effective in high dimensional space and this data has 256 features. In addition, the number of training samples (=7291) are much larger than the number of features (=256), which makes SVM be effective.

2.5. PCA

PCA with KNN: k for KNN is set as 4. Three different values for PCA are chosen. Their test error and accumulative explained_variance_ratio are listed below:

n_components	Test Error	explained_variance_ratio
32	0.04883	0.8205
64	0.05032	0.9190
128	0.05132	0.9762

KNN relies on an arbitrary distance metric to compute the nearest neighbors for classification. All components are weighted "equal" in distance calculation. This is non-ideal because the initial components are far more "discriminative" and different "important". So more and more components are involved in KNN, they become less and less informative and more and more noisy. Therefore, PCA could improve the performance of KNN by reducing the dimensions of features while keeping most of the information, which means decreasing noise in KNN.

PCA with poly SVM:

n_components	Test Error	explained_variance_ratio
32	0.04883	0.8202
64	0.04385	0.9189
128	0.04983	0.9761

In this case, combining PCA with SVM can improve the SVM performance. Since PCA is kind of feature selection, which gets rid of irrelevant or redundant features and reduces the number of dimensions.