CS 6375
Prof. Nick Ruozzi
Problem Set 4
Student: Tri M. Cao
Date: November 12, 2017

Problem 1a:

- Top six eigenvalues of the data covariance matrix are: 6.676, 3.260, 2.251, 1.847, 1.640, 1.593.

- Report the error of the learned classifier on the validation set for each k and c pair:

K	C	Valid set error
1	1	0.32125
1	10	0.31875
1	100	0.32125
1	1000	0.0225
2	1	0.16875
2	10	0.16875
2	100	0.17875
2	1000	0.13
3	1	0.17875
3	10	0.17875
3	100	0.1425
3	1000	0.36375
4	1	0.17875
4	10	0.17625
4	100	0.2275
4	1000	0.46375
5	1	0.18125
5	10	0.18125
5	100	0.1675
5	1000	0.15875

- Best k/c is (1, 1000) with the validation set error of 0.0225.
  Error on test set is 0.005.
  The best SVM classifier without PCA uses c = 1000 and has the test set error of 0.0387. Clearly PCA improves the accuracy of the classifier in this data set.

- If I have to pick a value of k, I will probably run a clustering algorithm and see roughly how many clusters the training data have. Then I will choose k to be nearly equal to the number of clusters. Of course, it’s not a bad idea to test different values of k.

Problem 1b: PCA for Feature Selection

- Naïve Bayes classifier’s accuracy on test set is 0.5955.

- We know the eigenvectors are all positive (due to the fact that the covariance matrix is positive-definite), and the eigenvectors are orthonormal, i.e. the sum square of each eigenvector = 1. Therefore, π has each entry >= 0 and all entries summing to 1, and it means π is a probability distribution.

- Report the average test error over 100 iterations of Naïve Bayes:
  Because there are too many combinations of (k, s), the numbers are provided in the text file: *pca_feature_selection_test_error.txt*.

- I don’t think this provides a reasonable alternative to naïve Bayes without feature selection on this dataset considering the accuracy of Naïve Bayes model without feature selection is 0.5955, and the best accuracy with feature selection is 0.47.

- Pros:
  - The model may become more interpretable after we determine which set of features best explain the data.
  - For some dataset, using less features will lead to faster model training.
  - If we get lucky, the accuracy may even improve with less features.

- Cons:
  - Hard to choose k and s beforehand, and it takes a lot of time to do grid search for the best (k, s) pair.
  - Performance may decrease with less features.

Problem 2:

1. The optimization problem solved by EM is to maximize log probability of the training data based on the Mixture of Naïve Bayes models.

$$argmax_{\lambda_z,\theta_z}\, l(z) = \sum_{i=1}^{N}\sum_{z=1}^{k} \lambda_z . p(x^{(i)}, y^{(i)} | \theta_{z^{(i)}})$$

   where N is the number of samples in the training set, k is the number of NB models in the mixture, and z denotes a NB model in the mixture.

2. Admittedly I cannot derive the closed form equations for the E and M step updates of the EM algorithm (because I'm still learning the math to do it). But I know intuitively what the closed form equations for E and M steps should be by referring the equations of Multinomial Naïve Bayes model and the Mixture model in general.

   E-step: in each iteration, we update the distribution $q_i(z^{(i)})$ where $z^{(i)}$ is the NB (Naïve Bayes) model that generated the sample *i*th.

$$q_i(z^{(i)}) = p(z^{(i)}|x^{(i)}, y^{(i)})$$

$$q_i(z^{(i)}) = \frac{p(x^{(i)}, y^{(i)}, z^{(i)})}{\sum_{z^{(i)}}^{k} p(x^{(i)}, y^{(i)}, z^{(i)})}$$

$$q_i(z^{(i)}) = \frac{\lambda_{z^{(i)}}.p(x^{(i)}, y^{(i)}|\theta_{z^{(i)}})}{\sum_{z^{(i)}} \lambda_{z^{(i)}}.p(x^{(i)}, y^{(i)}|\theta_{z^{(i)}})}$$

   Clearly, in the E-step we can compute $q_i(z^{(i)})$ given $\lambda_{z^{(i)}}$ and the parameters of each NB model.

   M-step: Basically, in the M-step we will update each NB model's parameters. The parameters in each NB model include $p(y|z)$ and $p(x|y,z)$, where y is the label and x is the feature vector.

   For each model z, the update equation for λ_z is:

$$\lambda_z = \frac{1}{N}\sum_{i}^{N} q_i(z^{(i)})$$

   For each label $y = l$, the update equation for $p(y = l|z)$ is as follows:

$$p(y = l|z) = \frac{\sum_i^N q_i(z^{(i)}).1\{y^{(i)} = l\}}{\sum_i^N q_i(z^{(i)})}$$

   where $1\{y^{(i)} = l\}$ is an indicator function.

   For each feature $x = j$ (this means the word *j* in the dictionary) and label $y = l$, the update equation for $p(x = d \mid y = l, z)$ is:

$$p(x = j \mid y = l, z) = \frac{\sum_i^N q_i(z^{(i)}).1\{y^{(i)} = l\}.count\{x^{(i)} = j\}}{\sum_i^N q_i(z^{(i)}).1\{y^{(i)} = l\}.\sum_d^D count\{x^{(i)} = d\}}$$

where $count\{x^{(i)} = j\}$ is equal to the number of word *j* in the sample $x^{(i)}$ (in other words, number of times the word *j* appears), and *D* is the number of different words in the dictionary.
In our problem, we have 3 labels (y) and 4 characters (x). That means we have 12 parameters $p(x|y,z)$ and 3 parameters $p(y|z)$ for each NB model.

3. Experiment: Running EM algorithm for 10 random initializations.

Model	Training set accuracy	Test set accuracy
1	0.5381	0.5226
2	0.5414	0.5273
3	0.5546	0.5489
4	0.5099	0.5367
5	0.5602	0.5367
6	0.5550	0.5526
7	0.5428	0.5367
8	0.5405	0.5235
9	0.5475	0.5479
10	0.5541	0.5461
Average	0.5444	0.5379

4. Experiment: Running EM algorithm using Dirichlet distribution

Model	Training set accuracy	Test set accuracy
Average	0.5024	0.5195

I am not sure whether using Dirichlet distribution helps or not. But my result shows using Dirichlet distribution actually decreases the accuracy of the mixture model, at least for this dataset.