Homework 3

CS5785 Modern Analytics

Harrison Gregg
Yihan Chen

November 11, 2015

Programming Exercises

1. Sentiment analysis of online reviews.

   (a) The labels are ballanced, with 500 sentences labeled 0 and 500 sentences labeled 1. There are also a number of unlabeled sentences which we will be ignoring. I process the files by iterating through the lines, splitting them at the tab character, and checking if the line included a label. If it does, I preprocess it as follows and add it to a list.

   (b)
   - We lower case all the sentences, as capitalization shouldn't decide the overall sentiment of the sentence, though it might affect the degree of the sentiment.
   - We lemmatize words, as grammatical variations on the same word shouldn't affect the sentiment of a sentence.
   - We strip all punctuation from sentences. Periods, commas, and most other punctuation should be completely irrelevant, and exclamation and question marks will probably be used in sentences with both labels.
   - We also strip stop words, as they should not be relevant to the sentiment.

   (c) Done.

   (d) Testing data should be kept completely seperate on principle, but additionally, we shouldn't use testing set when building feature vectors because there are probably many words in the testing set that don't occur in training set. These would be absolutely meaningless for any classifiers, as there would never be a number in that position during training.

   (e) The matrix is sparse which means most of the elements in feature vectors will be 0. So if there are some high frequency words occur, that would result in huge variance and strong bias in clustering. We apply l2 norm to reduce the influence of high frequency words and reduce the variance of matrix.

   (f) The means of k-means are reported in . The accuracy of the clustering is rather poor, at around 55This might be due to the extreme sparsity of the feature vectors. All of the vectors have relatively low distance, and the relevant data is among large amounts of irrelevant data.

   (g) The accuracy of logistic regression was 78.9%. By inspecting the coefficient, we found the words made biggest contribution are "cool", "liked", "interesting", "comfortable", "definitely", "recommend", "delicious", "fine", "fantastic", "beautiful", "amazing", "loved", "wonderful", "well", "good", "nice", "best", "excellent", "love", and "great".

   (h) 1h) The kmeans accuracy of n-grams model is around 57%. Logistic regression accuracy is around 73%. With this data, using 2-grams as opposed to single words did not have a statistically significant effect. k-means clustering performed slightly better, and logistic regression performed slightly worse. N-grams may perform better than single words in some scenarios, but the problem in this case may be that the original reviews are rather short, and

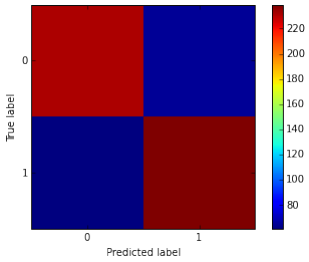


Figure 1: Representation of the covariance matrix of the logistic regression model on individual words.

we don't have very many samples. Therefore, the vast majority of 2-grams would not appear more than once, so the resulting feature vectors would be even more extremely sparse. The key n-grams detected by logistic regression are "well made", "great film", "film great", "definitely worth", "highly recommended", "love phone", "great price", "pretty good", "great deal", "good product", "excellent product", "good price", "good quality", "easy use", "great product", "work fine", "highly recommend", "one best", "great phone", and "work great".

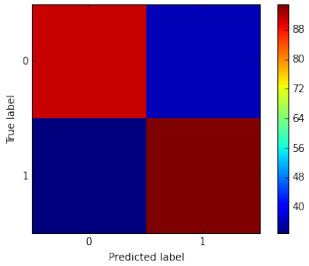


Figure 2: Representation of the covariance matrix of the logistic regression model on n-grams.

(i) The accuracy of k-means on reduced feature vectors was around 60%. The accuracy of logistic regression using PCA data was around 50%.

(j) Using 2-grams and PCA for bag of words allowed the clustering to perform slightly better in this case, but it was still only marginally better than the base rate. We believe this is because n-gram feature vectors and PCA reduced feature vectors are more distinct than bag of words. For logistic regression, both 2-gram and PCA for bag of words performed worse than standard bag of words. We believe this is because logistic regression is already good at picking out the feature which are important and only focusing on them. PCA for bag of words just got in the way of this. Inspecting the results, we learned that people use strong sentimental words like "excellent" and "good" to express there sentimental propensity.

2. EM algorithm and implementation

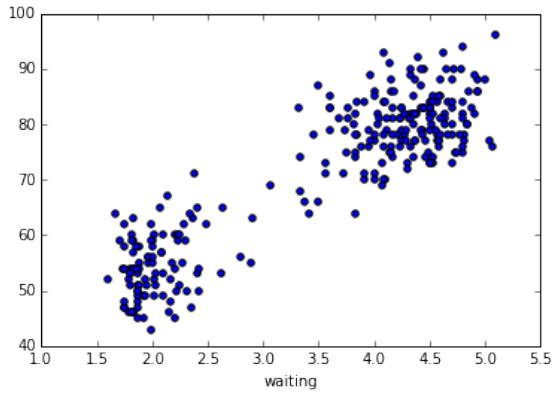
(a) See 14.2.2. $\hat{\gamma}_{ik} = 0$ unless

$$k = \arg\max_k g_k(x),$$

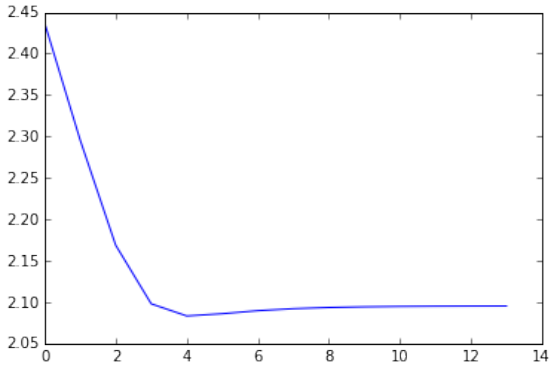
in which case, $\hat{\gamma}_{ik} = 1$. The M-step does not then have to be changed, as it just depends on the values from the E-Step.

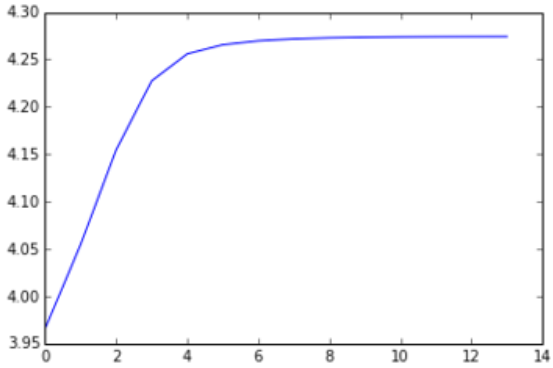
$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} x_i}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}$$

(b) Old Faithful Dataset:



(c) Trajectories of mean vectors:

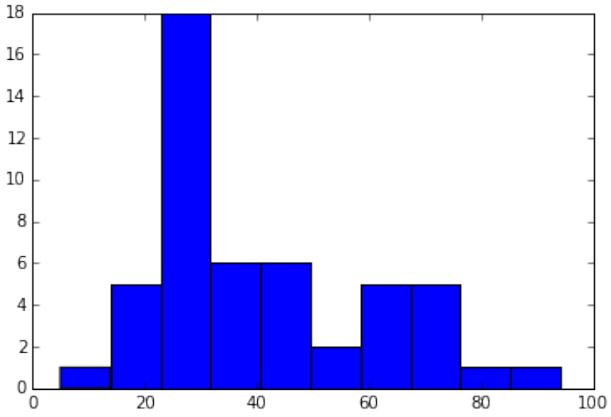


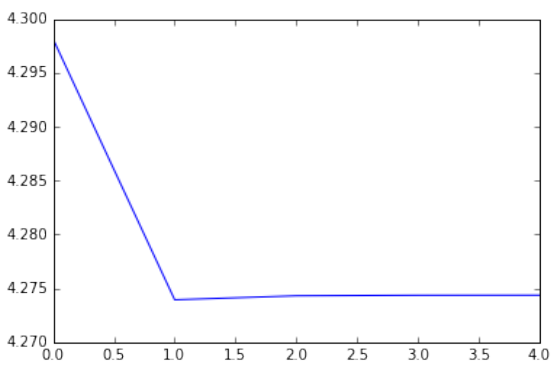


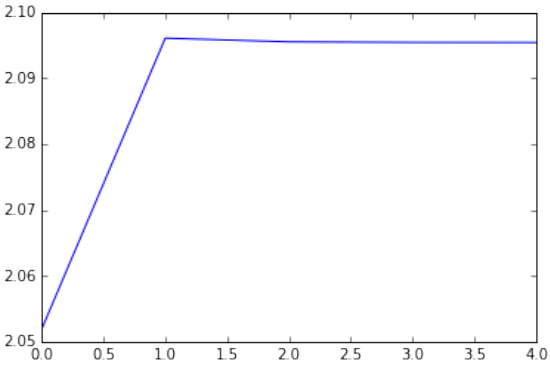
Iterations until convergence:

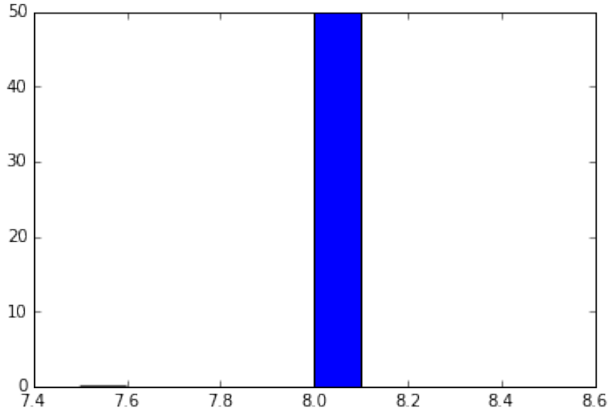
(d) Trajectories of mean vectors:

Iterations until convergence: When using k-means to initialize the EM algorithm, the algorithm converged faster than in (c). This is because the initial means and covariance generated by k-means is a reasonable depiction of the original data, rather than random parameter set in the last question.









WRITTEN QUESTIONS

14.2 1. The log-likelihood of the data is:

$$\log \prod_{i=1}^{N} \left(\sum_{k=1}^{k} \pi_k g_k(x_i) \right) = \sum_{i=1}^{N} \log \sum_{k=1}^{k} \pi_k g_k(x_i)$$

2. i. Take initial guesses for $\hat{\mu}_i, \hat{\sigma}_i^2$, and $\hat{\pi}_i$ for $i = 1 \ldots k$.

ii. E-step: compute responsibilities

$$\hat{\gamma}_{ik} = \frac{\hat{\pi}_k g_k(x_i)}{\sum_{j=1}^{K} \hat{\pi}_j g_j(x_i)}$$

iii. M-step: compute weighted means and variances

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik} x_i}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}$$

$$\hat{\sigma}_k^2 = \frac{\sum_{i=1}^{N} \hat{\gamma}_{ik}(x_i - \hat{\mu}_k)^2}{\sum_{i=1}^{N} \hat{\gamma}_{ik}}$$

iv. Iterate steps ii and iii until convergence.

3. Let all σ_k have fixed value instead of being updated in the algorithm. As the value of σ_k approaches 0, each $g_k(x)$ becomes less of a density functional until it becomes entirely discrete, with $g_k(x)$ equal to either 0 or a fixed constant value. When this happens, in each iteration of the algorithm, each data point belongs to only component, and the algorithm becomes effectively k-means.

14.8 To determine solutions for Procrustes problem,

$$\min_{\mu,R} \left\| X_2 - (X_1 R + 1\mu^T) \right\|_F,$$

we can equivalently solve:

$$\min_{\mu,R} \left\| X_2 - (X_1 R + 1\mu^T) \right\|_F^2 = \min_{\mu,R} tr\left((X_2 - (X_1 R + 1\mu^T))^T (X_2 - (X_1 R + 1\mu^T)) \right).$$

Now, let's expand the inside of the formula:

$$\begin{aligned}
&(X_2 - X_1 R - 1\mu^T)^T (X_2 - (X_1 R + 1\mu^T)) \\
=&(X_2 - X_1 R - 1\mu^T)^T (X_2 - (X_1 R + 1\mu^T)) \\
=&((X_2 - X_1 R)^T - \mu 1^T)((X_2 - X_1 R) - 1\mu^T)) \\
=&(X_2 - X_1 R)^T (X_2 - X_1 R) - (X_2 - X_1 R)^T (1\mu^T) - (\mu 1^T)(X_2 - X_1 R) + (\mu 1^T 1\mu^T) \\
=&A^T A - A^T (1\mu^T) - (\mu 1^T) A + (\mu 1^T 1\mu^T),
\end{aligned}$$

finally substituting $A = (X_2 - X_1 R)^T$ for convenience. We can now plug this back into the formula for the Frobenius norm, and take the derivative with respect

to μ.

$$\begin{aligned}
&\frac{\partial}{\partial\mu}tr(A^TA - A^T(1\mu^T) - (\mu 1^T)A + (\mu 1^T 1\mu^T)) \\
=&\frac{\partial}{\partial\mu}tr(A^TA) - \frac{\partial}{\partial\mu}tr(A^T(1\mu^T)) - \frac{\partial}{\partial\mu}tr((\mu 1^T)A) + \frac{\partial}{\partial\mu}tr((\mu 1^T 1\mu^T)) \\
=&0 - A\frac{\partial}{\partial\mu}(1\mu^T) - A\frac{\partial}{\partial\mu}(1\mu^T) + (1^T 1 + 11^T)\mu \\
=&0 - A1 - A1 + 2N\mu \\
=&2(-A1 + N\mu) \\
=&2(-(X_2 - X_1R)^T 1 + N\mu)
\end{aligned}$$

We can then set this derivative to 0 to find the minimum, and solve for μ.

$$\begin{aligned}
0 &= -(X_2 - X_1R)^T 1 + N\mu \\
N\mu &= (X_2 - X_1R)^T 1 \\
\mu &= \frac{1}{N}(X_2 - X_1R)^T 1 \\
&= \overline{x}_2 - R^T\overline{x}_1
\end{aligned}$$

Then, to solve for R, we plug μ back into the original equations, and substitue with our definitions for $\tilde{X}_1$ and $\tilde{X}_2$.

$$\begin{aligned}
X_2 - X_1R - 1\mu^T &= X_2 - X_1R - 1(\overline{x}_2 - R^T\overline{x}_1)^T \\
&= (X_2 - 1\overline{x}_2^T) - (X_1 - 1\overline{x}_1^T)R \\
&= \tilde{X}_2 - \tilde{X}_1R
\end{aligned}$$

Now, we need to determine R. There exists a matrix, let's call it Q, such that $\tilde{X}_2 = \tilde{X}_1Q$. However, Q will probably not be orthogonal. So, we want to find the best orthogonal approximation, R, of Q:

$$\begin{aligned}
&\arg\min_R \|R - Q\|_F^2 \text{ subject to } R^TR = I \\
=&\arg\min_R tr((R - Q)^T(R - Q)) \\
=&\arg\min_R (tr(I) - 2tr(R^TQ) + tr(Q^TQ)) \\
=&\arg\min_R -tr(R^TQ) \\
=&\arg\max_R tr(R^TQ)
\end{aligned}$$

Now, let $Q = UDV^T$ be the SVD of Q. Additionally, define an orthogonal matrix

$Z = V^T R^T U$. We can then write,

$$\begin{aligned} tr(R^T Q) =& tr(R^T U D V^T) \\ =& tr(V^T R^T U D) \\ =& tr(ZD) \\ =& \sum_{i=1}^{N} z_{ii} D_i \\ \leq& \sum_{i=1}^{N} D_i. \end{aligned}$$

This inequality is possible because Q is a diagonal matrix, so it's trace is simply the sum of it's elements. Therefore, multiplication by R, a matrix with determinant 1, cannot increase the trace. Therefore, to maximize $tr(R^T Q)$, it is sufficient to simply set $R = UV^T$ and thus $Z = I$.

14.11 Let D_k be a diagonal matrix of the square roots of the first k eigenvalues of S and E_k be a matrix of the corresponding eigenvectors. By the properties of eigendecomposition, $E_k D_k$ is the best rank k approximation of X, by the Frobenius norm of the covariance matrices, where $S = X^T X$ is the covariance matrix of X. Therefore,

$$\left\| S - (E_k D_k)^T (E_k D_k) \right\|_F^2 = \min_Z \left\| S - Z^T Z \right\|_F^2 = \min_Z \sum_{i,i'} (s_{ii'} - \langle z_i, z_{i'} \rangle)^2.$$

Because the mean of each eigenvector is 0, the mean of all rows of $E_k D_k$ is a zero vector, so $\overline{z}$ is a k-dimensional 0 vector. Therefore, we can write the above equation as,

$$\min_Z \sum_{i,i'} (s_{ii'} - \langle z_i - \overline{z}, z_{i'} - \overline{z} \rangle)^2 = \min_Z S_C(z_1, z_2, \ldots, z_N).$$

That is to say, the solutions z_i to $\min S_C(z_1, z_2, \ldots, z_N)$ are the rows of $E_k D_k$.

References

https://en.wikipedia.org/wiki/Orthogonal_Procrustes_problem
https://en.wikipedia.org/wiki/Matrix_norm#Frobenius_norm
http://research.microsoft.com/en-us/um/people/zhang/Papers/TR98-71.pdf
http://www.stat.nthu.edu.tw/~swcheng/Teaching/stat5191/lecture/06_MDS.pdf
https://en.wikipedia.org/wiki/Eigendecomposition_of_a_matrix
http://www.visiondummy.com/2014/04/geometric-interpretation-covariance-matrix/
#Eigendecomposition_of_a_covariance_matrix