CS 5785: APPLIED MACHINE LEARNING

HOMEWORK 3

November 11, 2017

Sarah Le Cam - sdl83

Yunie Mao - ym224

Cornell Tech

Contents

SENTIMENT ANALYSIS FOR ONLINE REVIEWS

Question 1 (a)

We imported Numpy to parse each file and generated the training data and labels. The labels are balanced across the three files. In total, there are 1500 negative and 1500 positive labels.

Question 1 (b)

Since the dataset consist of online reviews that may contain noises and garbage, we performed the following transformations on the original dataset. Transformed all words into lowercase This makes it easier to find the set of matching words in dataset Stripped all punctuations Punctuations are noises that disrupts the word matching and should be removed from the dataset Stripped all stopwords such as ?and?, ?or? and ?the? Stopwords add minimal value and disrupt the word matching Lemmatized all words to convert plural nouns to singular, past tenses to present and comparative and superlative tenses to positive. Imported WordNetLemmatizer from the nltk.stem module to lemmatize words in each sentence in the data This helps to generalize the words which improves the matching accuracy

Question 1 (c)

For each file, we split the data into training and testing set by taking the first 400 positive and negative samples as the training data and the remainder as the testing data. In total, our training set contained 1200 positive samples and 1200 negative samples. Our testing set contained 300 positive samples and 300 negative samples.

Question 1 (d)

To extract features, we used the training set to built a dictionary of all unique words in the reviews. We did not use the testing set because using it to select features could lead to overfitting of data. Using our dictionary, we built a feature vector for each review in the training data, with its ith index as the occurrences of the ith dictionary word in the review. We then added each feature vector to a training feature matrix. We also created a testing feature matrix using the same method performed for the training feature. In total, there are 4356 features represented by the feature vector. We randomly chose two reviews from the training set and reported their feature vectors. ?wow love place? ? [1,1,1,0,0,...] ?crust not good? ? [0,0,0,1,1,1,0,...]

Question 1 (e)

Because majority of the English words do not appear in the reviews, most of the feature vector elements will be 0. In addition, the sentences are not of the same length, long sentences will yield higher frequencies of certain words. In order to handle the huge variance and to reduce the influence of the high frequency words in the feature vector, we applied l2 normalization as the postprocessing strategy. Since l2 minimizes sum of the square of the differences between the target value and the estimate values, applying l2 adjusts our model to handle sparsity of our feature vectors and provides us a much more stable solution.

Question 1 (f)

To perform sentiment predictions, we trained a logistic regression model on the training data and fitted it on the testing data using Scikit-learn?s Logistic Regression package. We reported the classification accuracy score and plotted the confusion matrix. Based on the weight vector of our model, we displayed the top 10 words that play the most important roles in deciding the sentiment of the reviews. In addition, we trained a Multinomial Naive Bayes classifier using Scikit-learn?s built-in package and performed a similar analysis. Here, we compare the performance of the Logistic Regression to that of the Multinomial Naive Bayes in predicting sentiments.

Question 1 (g)

Similar to the bag of words model, we constructed a dictionary of n-grams, contiguous sequences of words with n=2. We did not perform any postprocessing to normalize the training set as we had done with the bag of words. In this case, since we are looking at sequences of 2 words, any frequent words would not necessarily be frequent sequences, so there is not a strong need to reduce the effect of these frequent words. We then performed the training and testing with Logistic Regression and the Multinomial Naive Bayes models and reported the accuracy, the confusion matrix, and the top 10 most frequent sequences of 2-grams.

Question 1 (h)

Since the features in the bag of words model contain large redundancy, we implemented PCA to reduce the dimensions of the features to 10, 50 and 100 respectively. To perform PCA, we implemented the following: Computed the means of the feature data using np.mean Adjusted the feature data by subtracting its means Using Numpy?s linalg svd function, computed the unitary matrix V Computed the dot product of the feature data and the nth rank of the the conjugate transpose of V Using our PCA, we repeated the sentiment predictions using bag of words and n-grams to construct dictionaries and trained logistic regression and naive bayes models to fit our test data.

Question 1 (i)

We compare the performances of bag of words, 2-gram, and PCA for bag of words by examining the classification results, accuracy scores, and the weights learned from logistic regression and naive bayes training.
In general, people tend to use same set of words when providing reviews online, regardless of the services or product provided. By examining the set of most popular words used in user reviews, we can determine whether a given review expresses positive or negative sentiment.

CLUSTERING FOR TEXT ANALYSIS

Question 2 (a)

Question 2 (b)

EM Algorithm and Implementation

Question 3 (a)

Question 3 (b)

Question 3 (c)

Question 3 (d)

Written Exercises

Question 1

Ex. 14.2 Consider a mixture model density in p-dimensional feature space,

$$g(x) = \sum_{k=1}^{K} \pi_k g_k(x), \tag{14.114}$$

where $g_k = N(\mu_k, \mathbf{L}\cdot\sigma^2)$ and $\pi_k \geq 0 \; \forall k$ with $\sum_k \pi_k = 1$. Here $\{\mu_k, \pi_k\}, k = 1, \dots, K$ and σ^2 are unknown parameters.

Suppose we have data $x_1, x_2, \dots, x_N \sim g(x)$ and we wish to fit the mixture model.

1. Write down the log-likelihood of the data
2. Derive an EM algorithm for computing the maximum likelihood estimates (see Section 8.1).
3. Show that if σ has a known value in the mixture model and we take $\sigma \to 0$, then in a sense this EM algorithm coincides with K-means clustering.

Question 1 (a)

We are given:

$$g(x) = \sum_{k=1}^{K} \pi_k \cdot g_k(x)$$

Therefore, for each data point x_i:

$$g(x_i) = \sum_{k=1}^{K} \pi_k \cdot g_k(x_i)$$

Then, the likelihood of the whole dataset is:

$$\text{likelihood} = \prod_{i=1}^{N} g(x_i)$$

$$\begin{aligned}\text{Log-likelihood} &= log(\prod_{n=1}^{N} g(x_i)) \\ &= \sum_{i=1}^{N} log(g(x_i)) \\ &= \sum_{i=1}^{N} log(\sum_{k=1}^{K} \pi_k \cdot g_k(x_i))\end{aligned} \tag{1}$$

Question 1 (b)

[INITIALISATION] First, for each cluster, take initial guesses for the values of the mean $\hat{\mu}_k$, the variance $\hat{\sigma}_k^2$, and the prior $\hat{\pi}_k$, $k \in \{1,2,...,K\}$.

[E-STEP] Compute the responsibility of the model, $\hat{\gamma}_i$, for each datapoint, x_i, $c \in \{1,2,...,K\}$:

$$\gamma_i(c) = \frac{\pi_c \cdot g_c(x_i)}{\sum_{k=1}^{K} \pi_k \cdot g_k(x_i)}$$

[M-STEP] Compute the new weighted means and variances:

$$\mu_c = \frac{\sum_{i=1}^{N} \gamma_i(c) \cdot x_i}{\sum_{i=1}^{N} \gamma_i(c)}$$

$$\sigma_c^2 = \frac{\sum_{i=1}^{N} \gamma_i(c) \cdot (x_i - \mu_c)^2}{\sum_{i=1}^{N} \gamma_i(c)}$$

$$\pi_c = \frac{\sum_{i=1}^{N} \gamma_i(c)}{N}$$

[REPEAT] Repeat the E-Step and M-Step until convergence.

Question 1 (c)

We would like to show that if $\sigma \to 0$, this EM algorithm coincides with K-Means clustering. If $\sigma \to 0$, then, in the E-Step, the responsibility of the model will be:

$$\gamma_i(c) = \begin{cases} 1, & \text{if } x_i \text{ is classified as } c \\ 0, & \text{otherwise} \end{cases}$$

Then, in the M-Step, the means will only be updated for the data points in a given class:

$$\mu_c = \frac{\sum_{i=1}^{N} \gamma_i(c) \cdot x_i}{\sum_{i=1}^{N} \gamma_i(c)} \text{ if } x_i \in \text{ c}$$

This is equivalent to the K-Means algorithm.

Question 2

Ex. 14.11 *Classical multidimensional scaling.* Let $\mathbf{S}$ be the centered inner product matrix with elements $\langle x_i - \bar{x}, x_j - \bar{x} \rangle$. Let $\lambda_1 > \lambda_2 > \cdots > \lambda_k$ be the k largest eigenvalues of $\mathbf{S}$, with associated eigenvectors $\mathbf{E}_k = (\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k)$. Let $\mathbf{D}_k$ be a diagonal matrix with diagonal entries $\sqrt{\lambda_1}$, $\sqrt{\lambda_2}$, ..., $\sqrt{\lambda_k}$. Show that the solutions z_i to the classical scaling problem (14.100) are the *rows* of $\mathbf{E}_k\mathbf{D}_k$.

$$S_C(z_1, z_2, \ldots, z_N) = \sum_{i,i'} (s_{ii'} - \langle z_i - \bar{z}, z_{i'} - \bar{z} \rangle)^2 \qquad (14.100)$$

Question 3

Question 3 (a)

Question 3 (b)

Question 3 (c)

Question 3 (d)

Sources & External libraries

Stéfan van der Walt, S. Chris Colbert and Gaël Varoquaux. *The NumPy Array: A Structure for Efficient Numerical Computation*, Computing in Science & Engineering, 13, 22-30 (2011), DOI:10.1109/MCSE.2011.37

John D. Hunter. *Matplotlib: A 2D Graphics Environment*, Computing in Science & Engineering, 9, 90-95 (2007), DOI:10.1109/MCSE.2007.55

Jones E, Oliphant E, Peterson P, et al. *SciPy: Open Source Scientific Tools for Python*, 2001-, `http://www.scipy.org/`

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay. *Scikit-learn: Machine Learning in Python*, Journal of Machine Learning Research, 12, 2825-2830 (2011)

Wes McKinney. *Data Structures for Statistical Computing in Python*, Proceedings of the 9th Python in Science Conference, 51-56 (2010)