HOMEWORK 3

Class: CS 5785

Authors: Anna Guidi, Samantha Yip

November 10th, 2017

A. Purpose of the Study

Problem 1 – Sentiment Prediction: Understand the best way to format, clean and structure data. Learn how to pick the best Post-processing strategy. Decide how to best construct feature vectors for certain models as well as which models are the most opportune given the data.

Problem 2 – Clustering for text analysis: Learn how to analyze articles and cluster titles and vocabularies using k-means clustering. Determine the optimal k when doing k-means clustering.

Problem 3 – EM algorithm and implementation: Demonstrate that the alternating algorithm for k-means is a special case of the EM algorithm.

B. Procedure

Problem 1 – Sentiment Prediction

We import three different text files containing 1000 reviews each for three different companies (yelp, amazon, IMDB), select 800 rows from each file for training and the remaining 200 for testing, and decide on how to clean and format data (see results section on more details).

We then decide on how to best normalize the data (again, see results section), and then try out several different models (Logistic Regression and Bernoulli Gaussian), both with the original feature vectors for the training and testing set, but also with a 2-gram model. Lastly, we repeat all of the aforementioned steps, but with PCA, testing out all models with 10, 50 and 100 dimensions respectively.

Problem 2: Clustering for text analysis

In this problem, we clustered articles in two methods: 1) based on titles, which is considered as document-wise features 2) based on vocabularies, which is considered as term-wise features.

First, we decide what optimal k to use for each method based on the elbow method. We plot a line graph with number of k as the x-axis and distortion as the y-axis, and decide on the k right when the distortion has a large drop. To calculate distortion, we obtain, for each point, its euclidean distance from every cluster mean. We take the minimum distance of each data point and sum all of them up.

For the first part, we open the “science2k-titles.txt” and “science2k-doc-word.npy” files, and append the titles and document in separate lists, *titles* and *document*. The *document* list contains the normalized per-document smoothed word frequencies. We then fit a KMeans model with number of clusters = 5 onto the *document* list, and store the labels (0-4) and cluster centers for each document in the *labels* and *centers* list respectively. We create a nested dictionary where the key is the label and the list associated is another dictionary. Within the

nested dictionary, the key is the index of each document and the list associated is the euclidean of each document from its cluster center. The format of the nested dictionary is as follows:

{'Cluster 0': {25: [208.9887]}}

Then, within each cluster, we sort the dictionary based on ascending order of the euclidean distance of each document from the cluster center. In each cluster, we take the index of the first 10 articles from the dictionary, grab the titles of the articles from the *titles* list and append the titles into the *top_10_titles list*. In the end, we print these titles out to analyze the different clusters.

For the second part, we did similar steps for clustering articles based on vocabularies. The main differences are that we fit the Kmeans model onto the "science2k-word-doc.npy" instead and store the vocabularies in a list. Also, the number of clusters we use is 6.

Problem 3: EM algorithm and implementation

We download the Old Faithful Geyser Dataset, transform each entry into a 2 dimensional feature vector and plot all data points on 2-D plane. We then Implement a bimodal GMM model to fit all data points using the EM algorithm, for which we randomly initialize Gaussian parameters and run the program 50 times (each time with different initial parameter guesses).

Finally, we repeat the same procedure but the initial guesses for the parameters generated by running a k-means algorithm over all the data points with K = 2. We then estimate the first guess of the mean and covariance matrices using maximum likelihood over the labeled data points.

C. Results

Problem 1: Sentiment Prediction

Parse each file. Are the labels balanced? If not, what's the ratio between the two labels?
Yes, the labels are balanced for each .txt file (500 positive reviews and 500 negative reviews each).

Pick your preprocessing strategy. Explain the reasons for each of your decision (why or why not)

- Lowercase all of the words
    - **Yes**. We believe that whether or not a word is lower case or upper case does not contribute to the classification of the review (whether it is a positive or a negative sentiment). The author of the review likely picked the case of the word arbitrarily, so it does not make sense to add many more feature vectors based solely on lower case vs. upper case, making the operation more expensive and

potentially inefficient. By making all words lowercase, we get rid of this redundancy.

- Lemmatization of all the words
    - **No**. While Lemmatization is useful for some words (whether they are singular or plural), we were hesitant on homogenizing some words, for example turning better into best might be misleading because "best" almost always is indicative of a positive sentiment, but better could be used in a negative or positive sentiment ("even my dog's food tastes better than what they served"). We did a lot of research on libraries, but did not find functions that we wanted.
- Strip punctuation.
    - **Yes**. It could be argued that *certain* punctuation ("!" for example) tends stronger towards a positive review or a negative review, but not being experts or having sufficient evidence, we decided to strip all punctuation out since most of it is not useful (like periods "." and commas "," for example which are necessary in a sentence regardless of intent)
- Strip the stop words, e.g., "the", "and", "or"
    - **Yes**. Stop words are quintessential to form any type of sentence, whether positive or negative, so we removed them.
- Something else? Tell us about it.
    - **Yes.** We appended numbers 0 through 9 to the stop words array, because we decided that numbers are a stronger indicator of the *type* of product rather than the sentiment of the review.

Bag of Words model. Extract features and then represent each review using bag of words model, i.e., every word in the review becomes its own element in a feature vector. In order to do this, first, make one pass through all the reviews in the training set ***(Explain why we can't use testing set at this point)***

We cannot use the testing set at this point because this will bias our model. When the time comes to test our testing set, it will not be a fair test, and the result of such a mistake would be overfitting.

Report feature vectors of any two reviews in the training set

```
({'get': 2, 'jiggle': 1, 'plug': 1, 'line': 1, 'right': 1, 'decent': 1,
'volume': 1})
np.sum(reviews_training_feature_vectors[2])
Out[772]: 8.0


({'needless': 1, 'say': 1, 'wasted': 1, 'money': 1})
np.sum(reviews_training_feature_vectors[4])
4.0
```

(**Admittedly these are not feature vectors, but this might be more useful than what the question asked for since you would see mainly zeros and 8 & 4 1s respectively**).

Pick your postprocessing strategy, explain why

There are several strategies. Log normalization (log (x +1)) and L1 normalization are good with sparse data, however a lot of literature on the internet suggests that despite this, L2 normalization is more useful in practice, which was the case with our data.

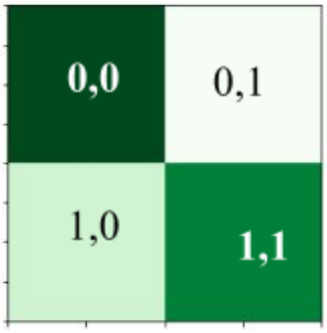
Another option that was not mentioned in the assignment was tf-idf (term frequency–inverse document frequency). This is a popular postprocessing strategy for bag-of-words models specifically, because term frequencies are not necessarily the best representation for the text, even when we do strip out the most common (or "stop") words.

Tf-idf addresses this problem, by weighing a term by the inverse of document frequency instead.

However, we found that tf-id resulted in around the same percentages as L2 normalization, so we decided not to change our strategy.

Sentiment prediction. Train a logistic regression model Report the classification accuracy and confusion matrix.

Accuracy: 79.67%




```
[[258  42]
 [ 80 220]]
```

Inspecting the weight vector of the logistic regression, what are the words that play the most important roles in deciding the sentiment of the reviews?

These are the top eight words regarding importance for classification:

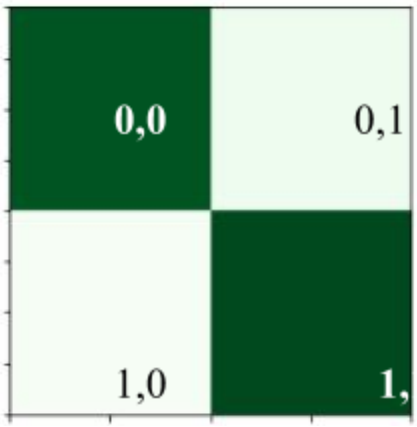
```
[(4.3409840737164398, 'great'),
 (3.3070532970949853, 'bad'),
 (2.8585988747199269, 'love'),
 (2.6791828038834198, 'excellent'),
 (2.5909793395046421, 'nice'),
 (2.3545127615314891, 'delicious'),
 (2.3526616141619727, 'poor'),
 (2.2438013364418032, 'good')]
```

These words are strong adjectives and it makes sense why they would be the most important indicators of a sentiment.

Repeat this with a Naive Bayes classifier and compare performance.

We picked Bernoulli because it seemed like an opportune model given the semi-binary nature of the feature vectors.

```
Accuracy: 77.33%
```



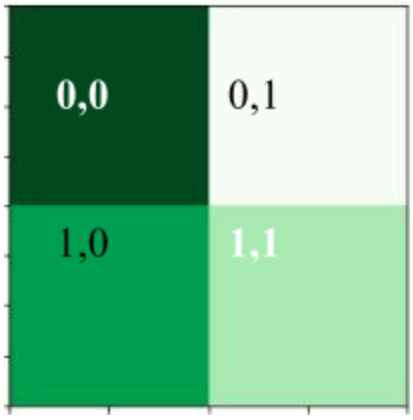

```
[[229  71]
 [ 65 235]]
```

The Bernoulli classifier is more balanced but slightly less accurate.

N-gram model.

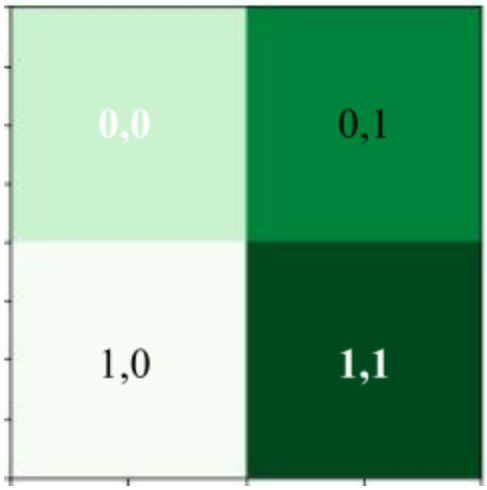
This model fares significantly worse with Logistic Regression having an accuracy around **62.167%,** and Bernoulli w/ an accuracy around **58.67%**. While the advantages of word associations are obvious, the reason for this decline in accuracy might be due to the increased sparsity of the data.

Logistic:



```
[[274  26]
 [201  99]]
```

Bernoulli:




```
[[ 67 233]
 [ 15 285]]
```

PCA. Report corresponding clustering and classification results

PCA – Logistic Regression

10	55%
50	52%
100	54%

PCA – Bernoulli

10	55.67%
50	54.83%
100	55.17%

PCA – n-Gram Logistic Regression

10	51%
50	52%
100	53.3%

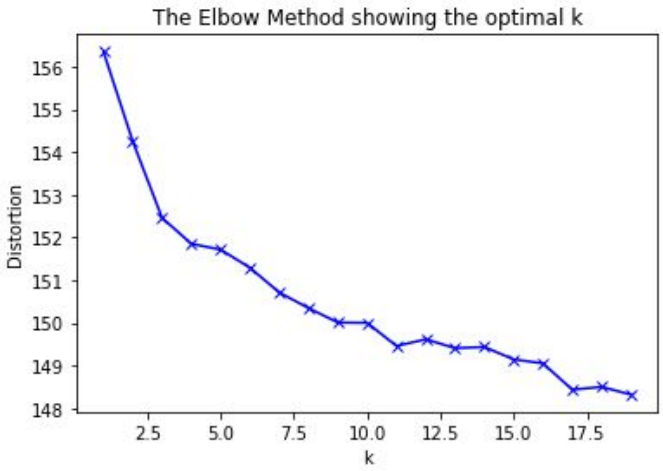
PCA – n-gram Bernoulli

10	52.8%
50	51.33%
100	50%

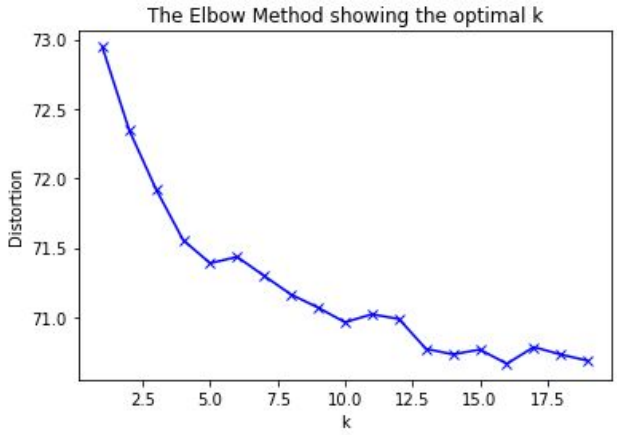
Problem 2: Clustering for text analysis

Select a value of k.

Clustering articles based on titles:
Below are line graphs that show the relationship between value of k and distortion. From Graph 1, we can see that k = 5 is the most optimal, because that is right after there is a large drop in distortion. After k = 5, the distortion decreases more gradually. From Graph 2, we can see that k = 6 is the most optimal, because that is right after there is a large drop in distortion. Therefore, based on elbow method, we decide to choose k = 5 for our k-means clustering on titles and k=6 for our k-means clustering on vocabularies.



Graph 1: k vs Distortion when clustering based on titles



Graph 2: k vs distortion when clustering based on vocabularies

Comment on these results. What has the algorithm captured? What is different about clustering terms from clustering documents?

Clustering based on titles:

Below are the top 10 titles within each cluster:

```
Cluster 0
[['"Population Dynamical Consequences of Climate Change for a Small Temperate Songbird"\n', '"Reconstruc
tion of the Amazon Basin Effective Moisture Availability over the past 14,000 Years"\n', '"Greenland Ice
Sheet: High-Elevation Balance and Peripheral Thinning"\n', '"Isotopic Evidence for Variations in the Mar
ine Calcium Cycle over the Cenozoic"\n', '"Mass Balance of the Greenland Ice Sheet at High Elevations"\n
', '"Rapid Kimberlite Ascent and the Significance of Ar-Ar Ages in Xenolith Phlogopites"\n', '"Glacial C
limate Instability"\n', '"Variable Carbon Sinks"\n', '"The Role of the Southern Ocean in Uptake and Stor
age of Anthropogenic Carbon Dioxide"\n', '"Remobilization in the Cratonic Lithosphere Recorded in Polycr
ystalline Diamond"\n']]
Cluster 1
[['"Requirement of NAD and SIR2 for Life-Span Extension by Calorie Restriction in Saccharomyces Cerevisi
ae"\n', '"Suppression of Mutations in Mitochondrial DNA by tRNAs Imported from the Cytoplasm"\n', '"Dist
inct Classes of Yeast Promoters Revealed by Differential TAF Recruitment"\n', '"Efficient Initiation of
HCV RNA Replication in Cell Culture"\n', '"Ubiquitination: More Than Two to Tango"\n', '"Negative Regula
tion of the SHATTERPROOF Genes by FRUITFULL during Arabidopsis Fruit Development"\n', '"T Cell-Independe
nt Rescue of B Lymphocytes from Peripheral Immune Tolerance"\n', '"Reduced Food Intake and Body Weight i
n Mice Treated with Fatty Acid Synthase Inhibitors"\n', '"Patterning of the Zebrafish Retina by a Wave o
f Sonic Hedgehog Activity"\n', '"Coupling of Stress in the ER to Activation of JNK Protein Kinases by Tr
ansmembrane Protein Kinase IRE1"\n']]
Cluster 2
[['"Structure of Yeast Poly(A) Polymerase Alone and in Complex with 3\'-dATP"\n', '"Structure of Murine
CTLA-4 and Its Role in Modulating T Cell Responsiveness"\n', '"Structure of the S15,S6,S18-rRNA Complex:
Assembly of the 30S Ribosome Central Domain"\n', '"Atomic Structure of PDE4: Insights into Phosphodieste
rase Mechanism and Specificity"\n', '"Twists in Catalysis: Alternating Conformations of Escherichia coli
Thioredoxin Reductase"\n', '"The Productive Conformation of Arachidonic Acid Bound to Prostaglandin Synt
hase"\n', '"Redox Signaling in Chloroplasts: Cleavage of Disulfides by an Iron-Sulfur Cluster"\n', '"Con
vergent Solutions to Binding at a Protein-Protein Interface"\n', '"Structural Basis of Smad2 Recognition
by the Smad Anchor for Receptor Activation"\n', '"Structure of the Protease Domain of Memapsin 2 (b-Secr
etase) Complexed with Inhibitor"\n']]
Cluster 3
[['"Algorithmic Gladiators Vie for Digital Glory"\n', '"Reopening the Darkest Chapter in German Science"
\n', '"Information Technology Takes a Different Tack"\n', '"National Academy of Sciences Elects New Memb
ers"\n', '"Archaeology in the Holy Land"\n', '"Heretical Idea Faces Its Sternest Test"\n', '"Corrections
and Clarifications: A Short Fe-Fe Distance in Peroxodiferric Ferritin: Control of Fe Substrate versus Co
factor Decay?"\n', '"Corrections and Clarifications: Charon\'s First Detailed Spectra Hold Many Surprise
s"\n', '"Corrections and Clarifications: Unearthing Monuments of the Yarmukians"\n', '"Divining Diet and
Disease from DNA"\n']]
Cluster 4
[['"The Formation of Chondrules at High Gas Pressures in the Solar Nebula"\n', '"Information Storage and
Retrieval through Quantum Phase"\n', '"Synthesis and Characterization of Helical Multi-Shell Gold Nanowi
res"\n', '"A Monoclinic Post-Stishovite Polymorph of Silica in the Shergotty Meteorite"\n', '"Nitric Aci
d Trihydrate (NAT) in Polar Stratospheric Clouds"\n', '"Ambipolar Pentacene Field-Effect Transistors and
Inverters"\n', '"Crossing the Hopf Bifurcation in a Live Predator-Prey System"\n', '"A Stable Bicyclic C
ompound with Two Si=Si Double Bonds"\n', '"How to Power a Nanomotor"\n', '"Xenon as a Complex Ligand: Th
e Tetra Xenono Gold(II) Cation in <latex>$AuXe_4^{2+}(Sb_2F_{11}^-)_2$</latex>"\n']]
```

The algorithm worked very well, because, based on observation, each cluster has a topic associated:

- Cluster 0: Climate and environmental changes
- Cluster 1: Genomes and human DNA
- Cluster 2: Bio-Chemistry
- Cluster 3: Examining past achievements and discoveries in various academic fields
- Cluster 4: Chemical Compounds/and the energy that they exert

The algorithm captured the fact that the titles can be categorized into 5 different genres.

Clustering based on vocabularies

Below are the top 10 vocabularies within each cluster:

```
Cluster 0
[['recalls\n', 'clinton\n', 'geneticist\n', 'security\n', 'fight\n', 'prize\
n', 'spending\n', 'campaign\n', 'hes\n', 'rights\n']]
Cluster 1
[['dispersion\n', 'photon\n', 'finite\n', 'angles\n', 'excited\n', 'nonlinea
r\n', 'diffraction\n', 'energies\n', 'regime\n', 'lattice\n']]
Cluster 2
[['start\n', 'rev\n', 'res\n', 'comparison\n', 'correspondence\n', 'signific
antly\n', 'concentration\n', 'two\n', 'sci\n', 'calculated\n']]
Cluster 3
[['aptamers\n', 'rory\n', 'dnag\n', 'doxy\n', 'nompc\n', 'trxr\n', 'ag7\n',
'lg268\n', 'proteorhodopsin\n', 'autophagy\n']]
Cluster 4
[['sciencemag\n', 'science\n', 'terra\n', 'ipcc\n', 'forcings\n', 'millennia
l\n', 'eastward\n', 'vol\n', 'troposphere\n', 'interglacial\n']]
Cluster 5
[['immunoprecipitated\n', 'polyacrylamide\n', 'lysates\n', 'immunoglobulin\n
', 'wildtype\n', 'homology\n', 'phosphorylated\n', 'kinases\n', 'saline\n',
'monoclonal\n']]
```

The algorithm worked very well, because, based on observation, each cluster has a topic associated:

- Cluster 0: Politics
- Cluster 1: Physics/Light
- Cluster 2: Measurements
- Cluster 3: Typographical error
- Cluster 4: Science/Direction
- Cluster 5: Biology/Chemistry

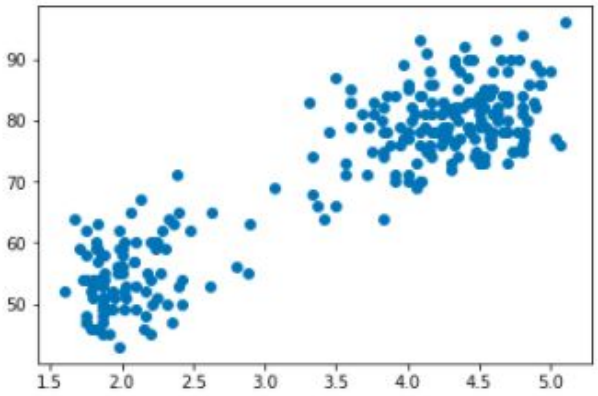
The algorithm has captured the fact that the vocabularies can be categorized into 6 different genres.

There is a difference between clustering terms and clustering documents, because the genres generated can be very different. For example, typographical error or physics/light would not be

genres within the titles. Therefore, depending on how you cluster the features, the genres you obtain can be very different.

Problem 3: EM algorithm and implementation

Plot all data points on 2-D plane

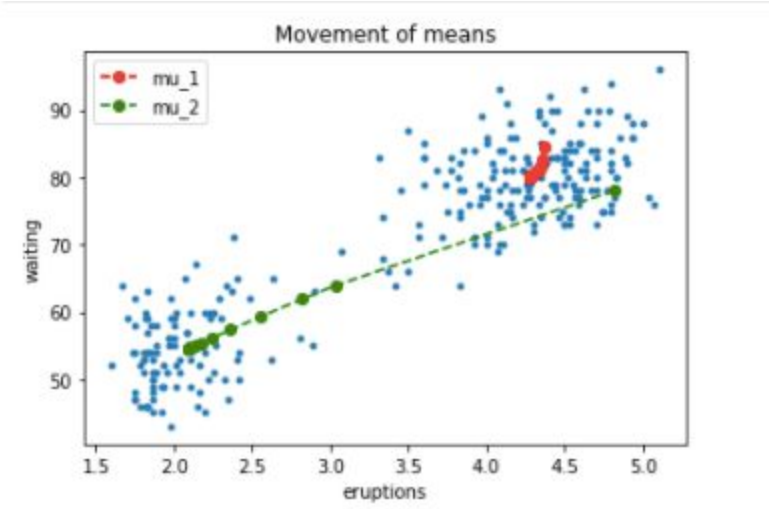


Explain the reasoning behind your termination criteria

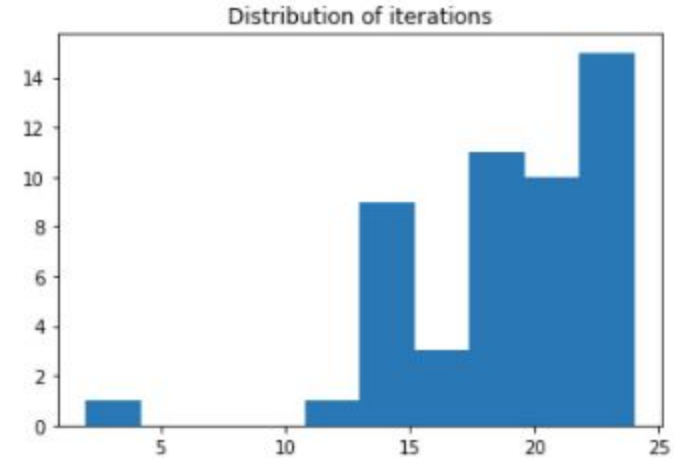
We have to termination criteria:

1. when the difference between the mus is less than .00001, then we terminate the process (sufficiently close to the mean)
2. we decide to terminate after 100 iterations regardless of the distances between the *mus* because we assume something went wrong and we want to move on to the next trial

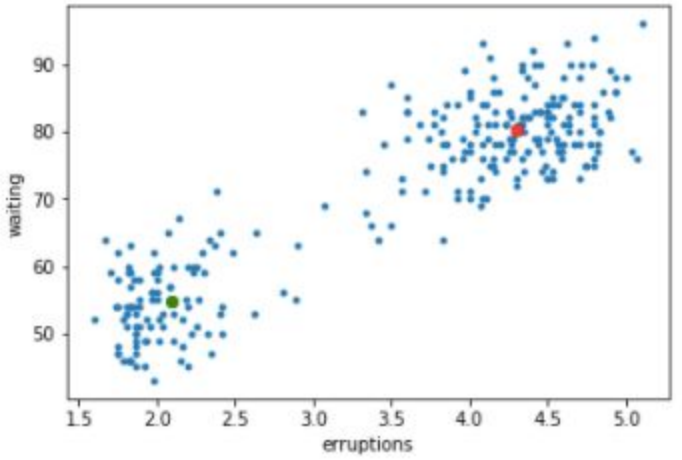
Plot the trajectories of two mean vectors in 2 dimensions (i.e., coordinates vs. iteration).

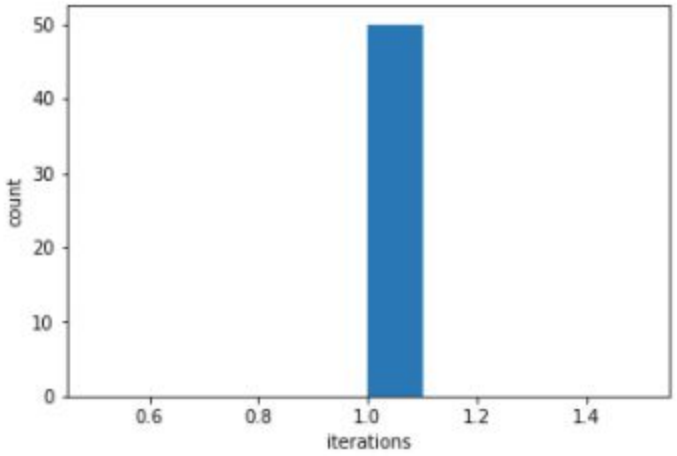


Show the distribution of the total number of iterations needed for algorithm to converge



now with k-means:





D. Conclusion

How well your solution works and any insights you found

Problem 1:

According to the above results, compare the performances of bag of words, 2-gram and PCA for bag of words. Which method performs best in the prediction task and why?

Logistic Regression with regular bag of words was the most suitable of all of the classifiers, because it can fit to the feature vectors better and in a more flexible way than the other models. Logistic Regression is good with sparse vectors because it detects even small differences between the vectors.

The Bernoulli NB model with bag of words was not as accurate as logistic regression, but comparable. It follows that bag of words is a good model for reviews and when working with text in general.

2-gram performed poorly with both Logistic Regression and especially with Bernoulli. 2-gram creates a very high number of features that do not contribute meaningful information. Additionally, the vectors become dramatically sparser than they already were, and it becomes difficult to detect any meaningful difference between the vectors.

PCA performed very poorly across the board. This makes sense because it is very possible that the components of PCA have a high variance, both in general and within reviews of the same class (with respect to inter-class variance). As a result, there might not be a good way for PCA to discard certain features, or it may have picked a bad axis and gotten rid of the features that were responsible for separating the two types of reviews.

What do you learn about the language that people use in online reviews (e.g., expressions that will make the posts positive/negative)?

As far as language that people use in online reviews, unsurprisingly some of the most heavily-used words tend to be generic but strong **adjectives,** such as great, poor, bad, excellent, delicious, good and nice (not in order of weight, for that see Results section above). This is very intuitive and most words are also applicable across industries (with the exception of "delicious"). What is exciting is that we could look for these words in reviews and if it is present, probably make a reasonable call as to whether that review is positive or not.

Problem 2: Clustering for text analysis

How might such an algorithm be useful?

The k-means clustering algorithm is very useful, because we are able to categorize titles and vocabularies, based on analysis on articles, efficiently without manual observation. For example, using k-means clustering, Wall Street Journal can parse the titles or words within different articles, and cluster them into various categories, such as Politics, Health, Technology etc. Another example would be Buzzfeed. They use k-means clustering to categorize quizzes based on tag words, which, in this case, would be similar to what we did when we clustered vocabularies. When there are millions of articles, it is impossible for humans to categorize them manually. Therefore, k-means clustering would be a powerful tool to use.

Problem 3: EM algorithm and implementation

Show that the alternating algorithm for k-means is a special case of the EM algorithm and show the corresponding objective functions for E-step and M-step.

Please see written question 1.

Compare the algorithm performances of (c) and (d).

K-means requires fewer iterations on average to converge compared with random initial parameters. This is because when we use K-means to initialize, the mu_1 and mu_2 are within the data points, whereas random initial parameters might have mu_1 and mu_2 far from the data points.

Through this problem, we understand that k-means is a special case of EM algorithm, since both of them find the cluster mean through iterations and stop when the cluster mean stabilizes.

E. <u>Questions</u> (Excuse the blurriness, we scanned the papers with the questions, the pdf of the)

Written Questions

1a) likelihood

$$\ell = \log\left(\prod_{i=1}^{N} g(x_i)\right)$$

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

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

1b) a) Take initial guesses for μ_k, π_k, σ_k^2, $k = 1, \ldots K$

b) E-step: compute responsibilities

$\hat{\gamma}_{jk}$ is the probability that jth observation is generated by variable k

$$\hat{\gamma}_{jk} = P(x = k \mid y_j, \theta)$$

$$= \frac{P(x = k, y_j \mid \theta)}{\sum_{k=1}^{K} P(x = k, y_j \mid \theta)}$$

$$= \frac{P(y_j \mid x = k, \theta) P(x = k \mid \theta)}{\sum_{k=1}^{K} P(y_j \mid x = k, \theta) P(x = k \mid \theta)}$$

$$= \frac{\pi_k \phi(y_j \mid \theta_k)}{\sum_{k=1}^{K} \pi_k \phi(y_j \mid \theta_k)}$$

c) M-step: compute weighted mean and variance

Log likelihood: $\ell = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k g_k(x_i) \right)$

Set $\frac{d\ell}{d\mu_k}$ to $0 \Rightarrow \mu_k = \frac{\sum_{j=1}^{N} \gamma_{jk} y_j}{\sum_{j=1}^{N} \gamma_{jk}}$

Set $\frac{d\ell}{d\sigma_k^2}$ to $0 \Rightarrow \sigma_k^2 = \frac{\sum_{j=1}^{N} \gamma_{jk} (y_j - \mu_j)^2}{\sum_{j=1}^{N} \gamma_{jk}}$

Set $\frac{d\ell}{d\pi_k}$ to $0 \Rightarrow \pi_k = \frac{\sum_{j=1}^{N} \gamma_{jk}}{N}$

16) (cont'd) Let initial guess be μ_k°, σ°, π_k°

$$\gamma_{jk}^\circ = \frac{\pi_k \phi(y_j | \theta_k)}{\sum_{k=1}^{K} \pi_k \phi(y_j | \theta_k)}$$

Then update μ_k, σ_k^2 and π_k

$$\mu_k' = \frac{\sum_{j=1}^{K} \gamma_{jk}^\circ y_j}{\sum_{j=1}^{N} \gamma_{jk}^\circ}$$

$$\sigma_k^{2\prime} = \frac{\sum_{j=1}^{N} \gamma_{jk}^\circ (y_j - \mu_j)^2}{\sum_{j=1}^{N} \gamma_{jk}^\circ}$$

$$\pi_k' = \frac{\sum_{j=1}^{N} \gamma_{jk}^\circ}{N}$$

Repeat the updating procedure until μ_k', $\sigma_k^{2\prime}$, π_k' converges

1c) If $\sigma \to 0$,

$$\sigma_k^2 = \frac{\sum_{j=1}^{N} \gamma_{jk} (y_j - \mu_j)^2}{\sum_{j=1}^{N} \gamma_{jk}} = 0$$

$$\sum_{j=1}^{N} \gamma_{jk} (y_j - \mu_j)^2 = 0$$

$$\sum_{j=1}^{N} \pi_k g_k(x_j) (y_j - \mu_j)^2 = 0$$

$$\pi_k \xrightarrow{\text{Converges to}} \begin{cases} 1, & \text{for most probable class} \\ 0, & \text{for the rest} \end{cases}$$

In this case, if $\sigma \to 0$, then in E-step, $\gamma_{jk} \to 1$

In M-step, μ_k is the nearest center

$$X_n \to 0$$

$$\therefore \gamma_{jk} = \begin{cases} 1, & \text{if } k = \underset{k}{\operatorname{argmin}} (\|x_j - x_k\|^2) \text{ for } j = 1, \ldots, N \\ 0, & \text{o.w.} \end{cases}$$

1c) (cont'd) Therefore, we are assigning each point to a cluster that has the mean closest to it, which coincides with k-means clustering.

2) $E_k D_k = (e_1, e_2, \ldots, e_k) \begin{pmatrix} \sqrt{\lambda_1} & \cdots & 0 \\ 0 & \sqrt{\lambda_2} & \cdots \\ \vdots & \vdots & \ddots \\ 0 & 0 & \sqrt{\lambda_k} \end{pmatrix}$

The inner product matrix for $S = XX^T$

and the elements of S, $S_{ij} = (x_i - \bar{x})^T (x_j - \bar{x})$

$\text{Rank}(S) = \text{Rank}(XX^T) = \text{Rank}(X) = k$

S is symmetric and positive with rank k, it has p non-negative eigenvalues and $n-p$ zero eigenvalues

S can be written as $V \Delta V^T$ where $\Delta = D^2$

$S = XX^T$

$V \Delta V^T = XX^T$

$V D^2 V^T = XX^T$

$X = \sqrt{(D^2)^{1/2}}$

$X = VD$

$X = ED$

Let $\lambda_1 > \lambda_2 > \ldots > \lambda_k$ be the k largest eigenvalues of S,

X_i would be the rows of $E_k D_k$

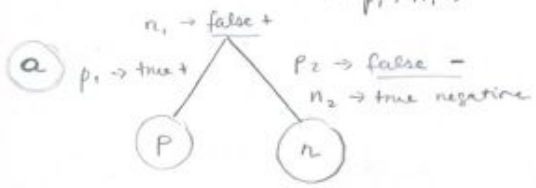
3

$I(r) = \min\{r, 1-r\}$ — where there are the min ratios

$p_i \rightarrow$ positive ; $n_i \rightarrow$ negative examples

$$(p_1 + n_1) \cdot I\left(\frac{p_1}{p_1+n_1}\right) + (p_2 + n_2) \cdot I\left(\frac{p_2}{p_2+n_2}\right)$$

(a)



thus, total # of mistakes

$= \boxed{n_1 + p_2}$

Using min-error impurity formula above:

$$(p_1 + n_1)\min\left\{\frac{p_1}{p_1+n_1}, 1 - \frac{p_1}{p_1+n_1}\right\} + (p_2 + n_2)\min\left\{\frac{p_2}{n_2+p_2}, \frac{1-p_2}{n_2+p_2}\right.$$

$$(p_1+n_1)\left(1 - \frac{p_1}{p_1+n_1}\right) + (p_2+n_2)\left(\frac{p_2}{p_2+n_2}\right)$$

$$(p_1+n_1) - p_1 + p_2 = \boxed{n_1 + p_2} =$$

(b) Using Gini Index $\rightarrow 1 - \sum_{i=1}^{k} p_i^2$

Step 0 : $I = 1 - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2 = .49 \Rightarrow$ overall

4 zeros, 6 ones — 7 zeros, 3 ones

	a_1	a_2	a_3	y
1	0	0	0	+
2	1	1	0	+
3	0	1	0	+
4	1	0	1	−
5	0	0	1	−
6	0	1	0	−
7	1	1	0	−
8	1	1	1	−
9	1	0	0	−
10				

a_1 : $a_1 = 0 \rightarrow 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2 = .5$

$a_1 = 1 \rightarrow 1 - \left(\frac{1}{6}\right)^2 - \left(\frac{5}{6}\right)^2 = .28$

$\Rightarrow .49 - (.5)\left(\frac{4}{10}\right) - (.28)\left(\frac{6}{10}\right) = .122$

a_2 : $a_2 = 1 \rightarrow 1 - \left(\frac{4}{6}\right)^2 - \left(\frac{2}{6}\right)^2 = .44$

$a_2 = 0 \rightarrow 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2 = .375$

$\Rightarrow .49 - (.375)\left(\frac{4}{10}\right) - (.44)\left(\frac{6}{10}\right)$

$= .076$

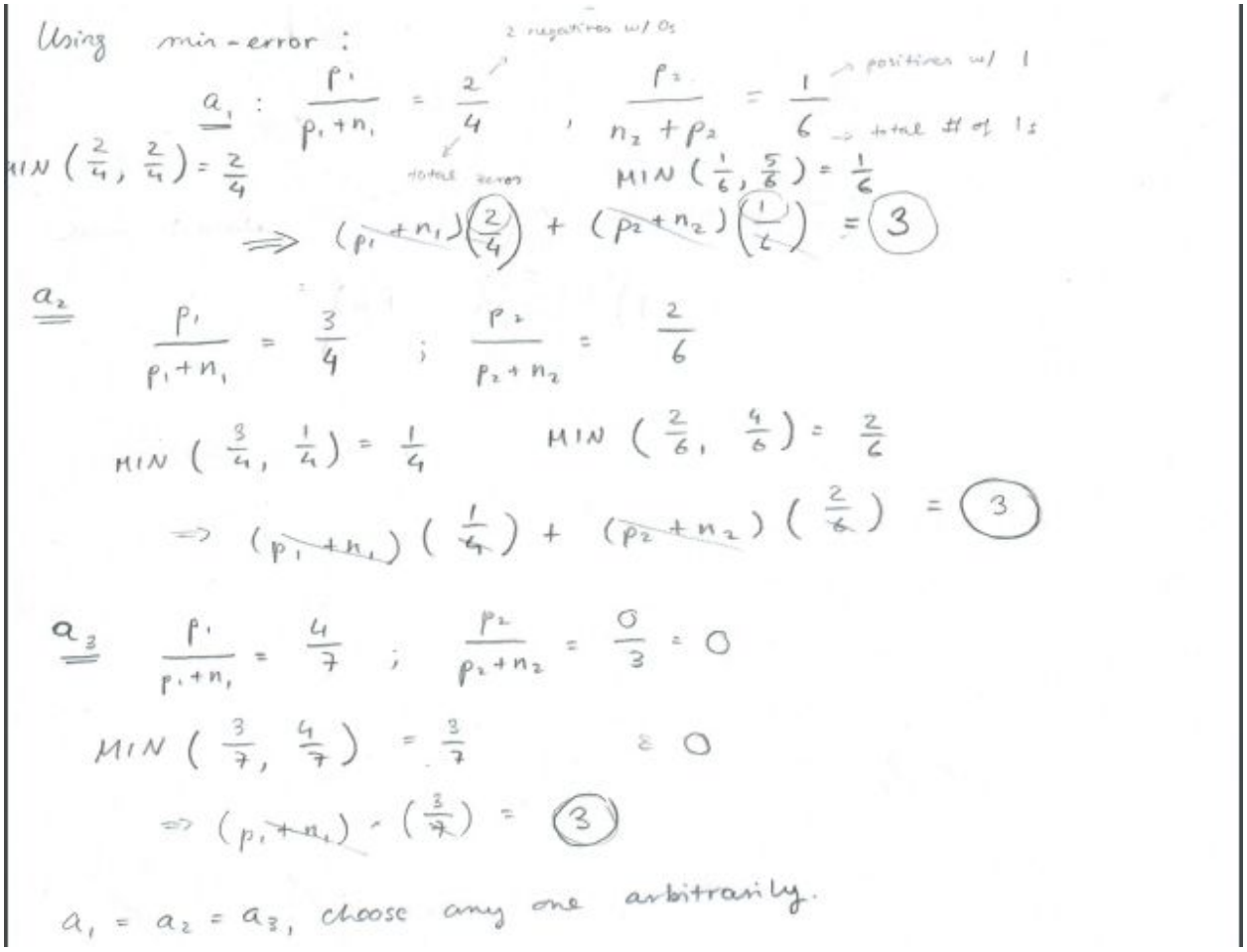
a_3 : $a_3 = 0 \rightarrow 1 - \left(\frac{4}{7}\right)^2 - \left(\frac{3}{7}\right)^2 = .49$

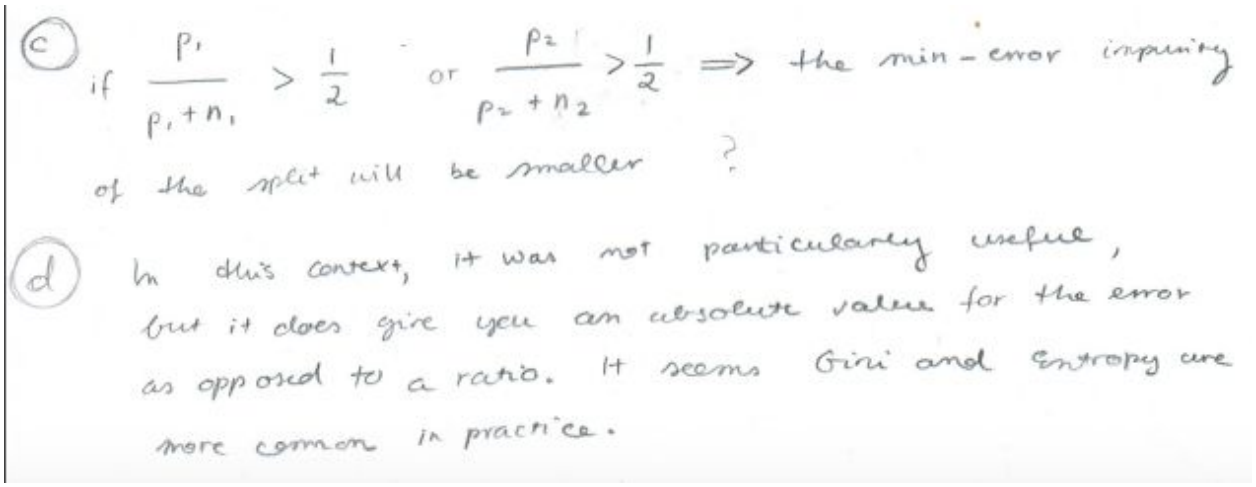
$a_3 = 1 \rightarrow 1 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 = 0$

(b continued)

$\Rightarrow .49 - \left(\frac{7}{10}\right)(.49) - \left(\frac{3}{10}\right)(0) = .147$

.147 is the biggest value, a_3 will be chosen w/ Gini





F. Citations

https://en.wikipedia.org/wiki/Tf%E2%80%93idf

https://en.wikipedia.org/wiki/Bag-of-words_model

http://locallyoptimal.com/blog/2013/01/20/elegant-n-gram-generation-in-python/

https://books.google.com/books?id=2S64-ZZ1fREC&pg=PA80&lpg=PA80&dq=postprocessing+strategy+machine+learning&source=bl&ots=BBVW2Bdli4&sig=9WUyh0O8xy0jIp3MI9YMK_gon5c&hl=en&sa=X&ved=0ahUKEwjMi4S6lajXAhUC4yYKHWfXCa0Q6AEIQTAD#v=onepage&q=postprocessing%20strategy%20machine%20learning&f=false