**Homework 4: To be or not to be…the author – Report**

Bin Gao, Bin Yan

# Summary

Something like this: We split the complete works of William Shakespeare into different acts. 13 features are extracted from all these acts, and normalized for further clustering. We have applied which, which, which… clustering methods. We concluded which method is better in terms of (accuracy, performance). Based on the result, we deduced that which/those acts are most likely not written by William Shakespeare because of (some short reasons).

# Part I Data Wrangling

To collect the necessary data, we have different options initially. The complete works of William Shakespeare is in HTML format in <http://shakespeare.mit.edu/>. To avoid the problem of downloading data from different links or the time required doing it by hand, we located the complete works of William Shakespeare from Project Gutenberg at <http://www.gutenberg.org/cache/epub/100/pg100.txt>. Locally it’s stored as “pg100.txt”.

In DataWrangling.ipynb, we use each play’s year as the separator to divide the complete txt file into individual txt file. So in each play’s txt file, the first line is the year the play was written, and everything after and including string “THE END” is discarded. We also removed all blank single lines. After obtaining each play, we further divide each play into different acts and store them into a dictionary key’ed by act titles. For each act, it will have tokenized words, non-stemmed words (which are also cleaned by removing stop words and punctuations), sentences, full text and filename being stored. The dictionary is stored in the works.json file. Once we have all the basic building blocks of the text. The next step is to extract text-mining features.

# Part II Feature Selection

Some people in machine learning society believe selecting the right features is more important than improving algorithms. As a first step, we have performed some experiments to determine which features to use. We use Affinity Propagation to test different features. The preference is set to the median of the input similarities. The advantage of using Affinity Propagation to optimize the features is that we do not need to optimize the number of clusters at the same time.

**TFIDF**

We started with TFIDF (the first 20 components of SVD) only. As shown in Figure 4, acts are grouped mainly by topics. Acts of the same play go to the same cluster. This single feature cannot reflect the writing style.

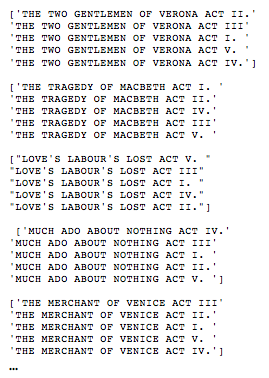
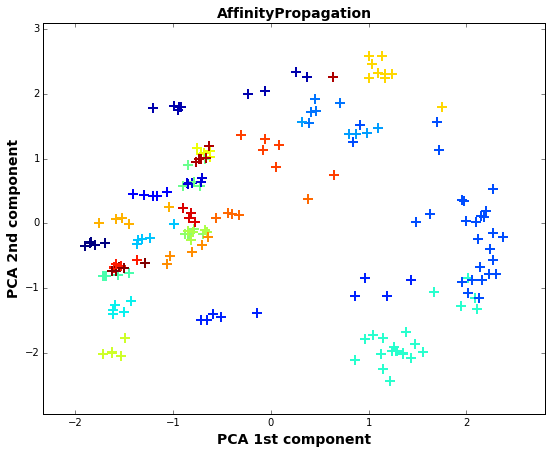


Figure Clustering results using TFIDF + Affinity Propagation (a) PCA visualization (b) Clip of clustering results

**N-gram**

Experts experience shows that n-gram might be more suitable to identify authorship. Then we have tried 2~4-gram and 3~5-gram. We find that 2~4-gram creates very strange outliers. Therefore, we choose 3~5-gram. However, by intuition, we still need more features to represent the wring style.

**Add frequency of special words, punctuation and sentence length**

Based on the papers of Stanko et al. (2013) [[1]](#footnote-1) and Selman & Husagic-Selman (2011)[[2]](#footnote-2), we use both sklearn and NTLK to construct frequency of special words (a, an, the, in, on, to, of, pronouns, conjunctions and ect.), frequency of unique words and sentence length. Table 1 lists all the features we use for this study. Features 3~12 are ***normalized by the total number of words*** in an act. Therefore, more precisely, they are ***frequencies***.

Table List of features

|  |  |
| --- | --- |
| 1. 3~5-gram normalized by TFIDF 2. Average length of sentences in the text 3. Hapax Legomena  (number of words that occur exactly once) 4. Dis Legomena  (Number of words that occur exactly twice) 5. Number of unique words | 1. Number of nominative pronouns 2. Number of conjunctions 3. Number of commas 4. Number of periods 5. Number of “a”, “an”, “the” 6. Number of “in”, “on”, “to”, “of” 7. Number of “is”, “are”, “was”, “were” |

Figure 2 shows the scatter matrix plot of all features excluding TFIDF and n-grams. According to Figure 2, Hapax Legomena, Dis Legomena and number of unique words are strongly correlated. The average length of sentence is correlated with number of dots. Others features are relatively independent.

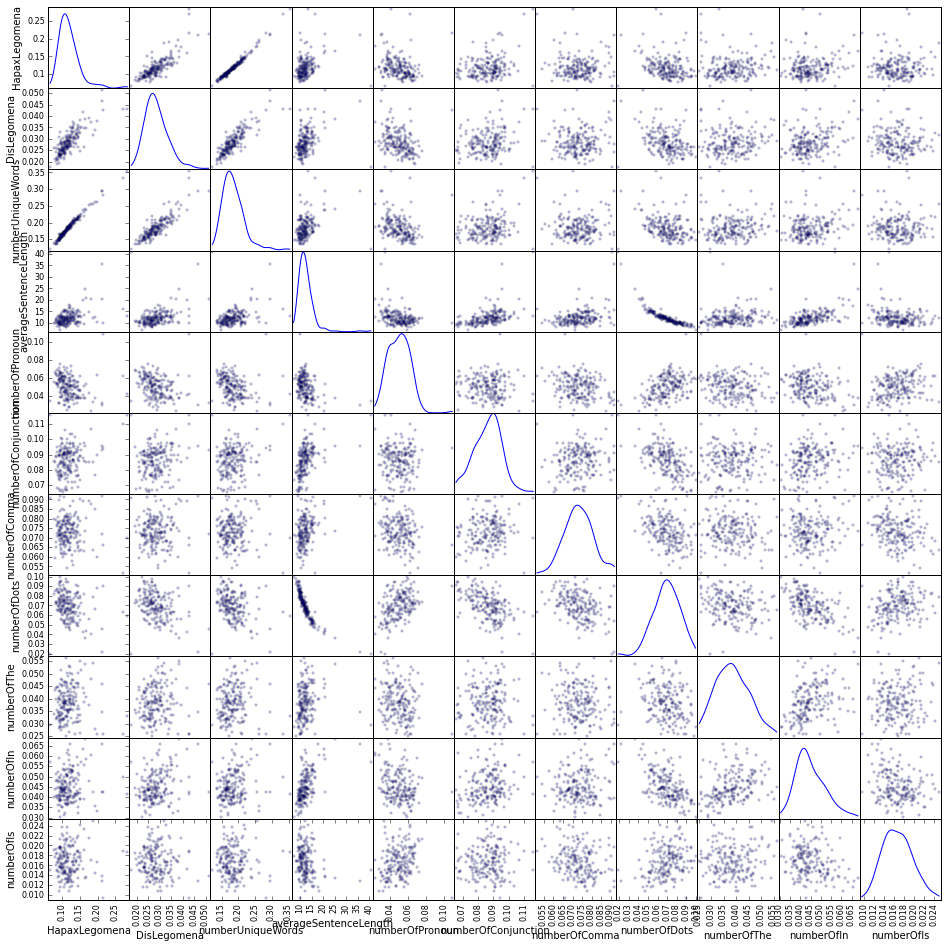


Figure Scatter matrix plot of 11 features

**Normalization/Standardization**

Do we need to normalize/standardize the feature matrix for clustering and PCA? After some research, the answer is yes. In this study, the average sentence length ranges from 10 to 30, while other features normalized by TFIDF or by total number of words are way below 0.1. As a result, if we run clustering based on Euclidean distance, the feature of sentence length dominates. Acts with long sentences become outliers and other features hardly play any role. Figure 2 shows covariance matrix of all features, non-standardized versus standardized. If the features are not standardized, the variance of average sentence length is on a different scale compared with other features. After standardization, the covariance matrix makes much more sense, as shown in Figure 2(b).

We also need to standardize the features when using PCA. Figure 3 shows the explained variance ratio of PCA components, non-standardized versus standardized. If the features are not standardized, the first PCA component, which is dominated by average sentence length, accounts for over 95% variance. By contrast, PCA after feature standardization makes more sense. In this study, the features are standardized to , a zero mean and unit standard deviation.

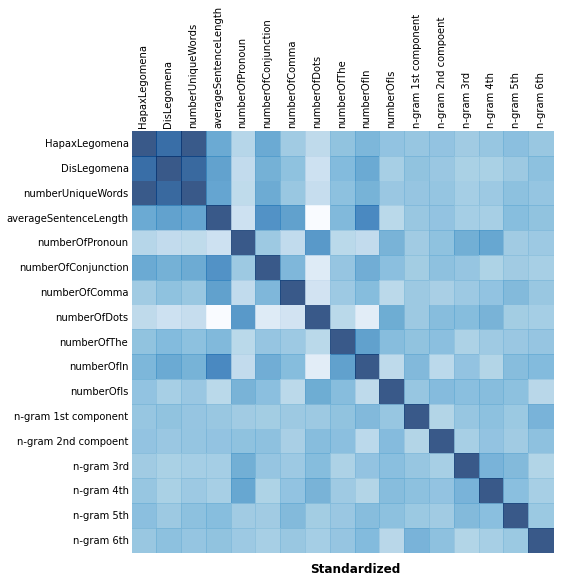
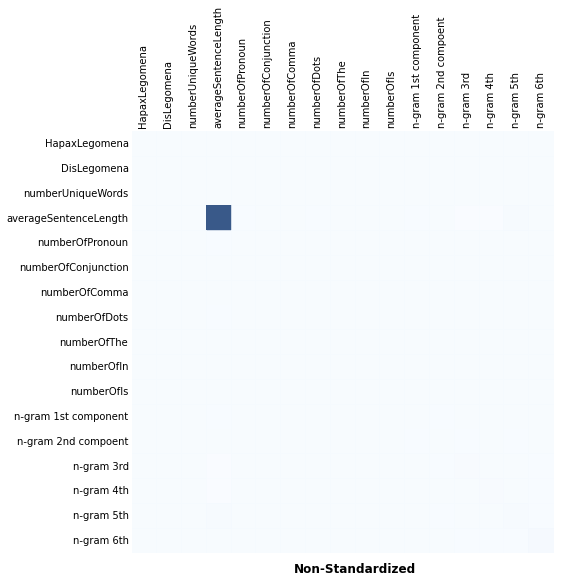


Figure Covariance matrix of all features (a) Non-Standardized features (b) Standardized features

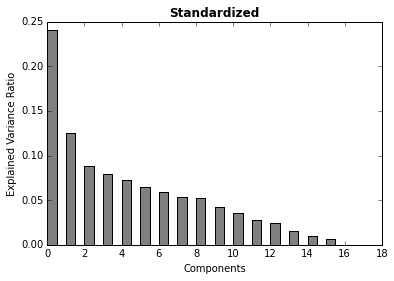
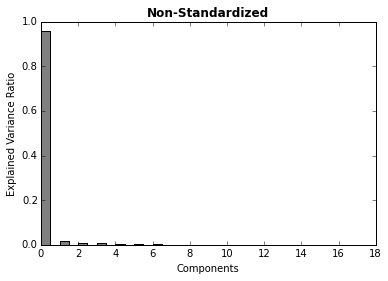


Figure Explained variance ratio of PCA components

Combined with n-gram features, we get the following results. Two poetries (the Sonnets, a Lover’s Complaint) are classified as outliers, as their styles are completely different from other plays. Some other acts are left out as well. We conclude the feature selection and proceed to different clustering methods.

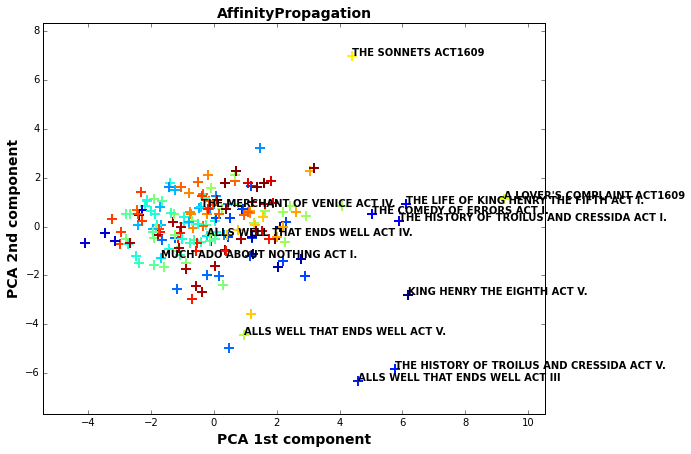


Figure Clustering results using the selected 12 features

# Part III Clustering Experiments

3.1 Clustering Methods

Six clustering methods are tested.

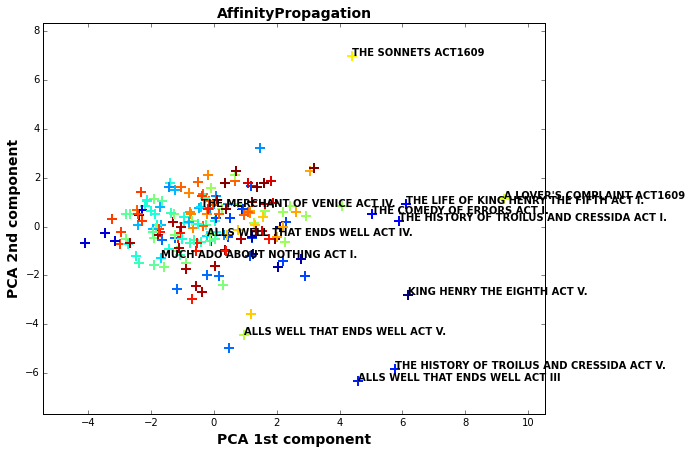
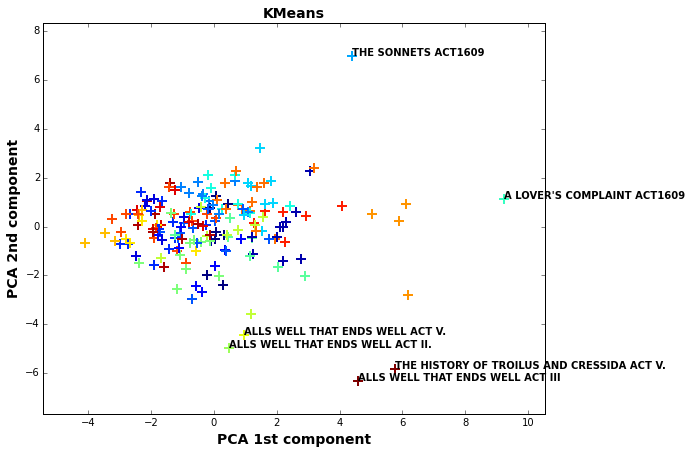
* KMeans: we use k-means++ to initialize the clusters. We use silhouette coefficient to optimize the number of clusters, as shown in Figure 6.

Figure Optimize the number of clusters

* Affinity Propagation: preference is set to the median of the similaries.
* Spectral Clustering
* Agglomerative Clustering: use silhouette coefficient to optimize the number of clusters.
* Birch
* Meanshift

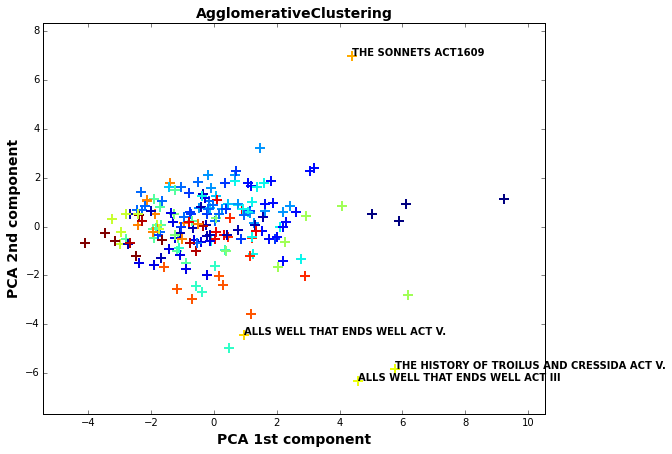
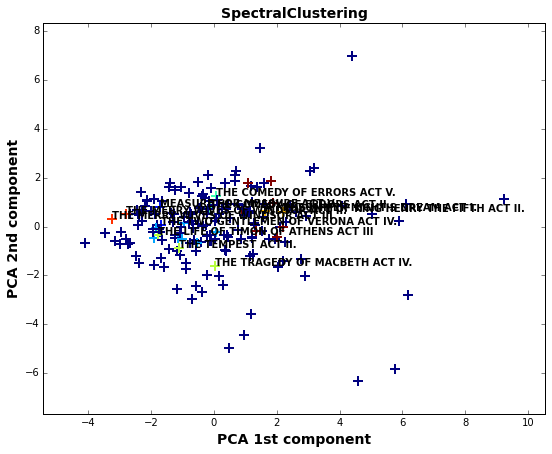
3.2 Comparison of different methods

Figure 7 shows the clustering results of six methods visualized in 2D PCA. Silhouette coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The Silhouette Coefficient for a sample is (b - a) / max(a, b) [[3]](#footnote-3). The best value is 1 and the worst value is -1. Values near 0 indicate overlapping clusters. Negative values generally indicate that a sample has been assigned to the wrong cluster, as a different cluster is more similar. The Silhouette coefficient shown in Figure 7 is the average of all samples. Then clusters where there are fewer than four members are labeled with the title of the act, except birch clustering where the smallest cluster is labeled.



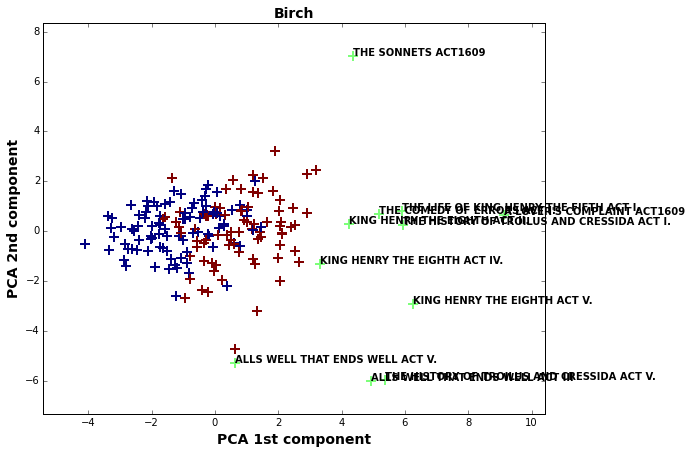
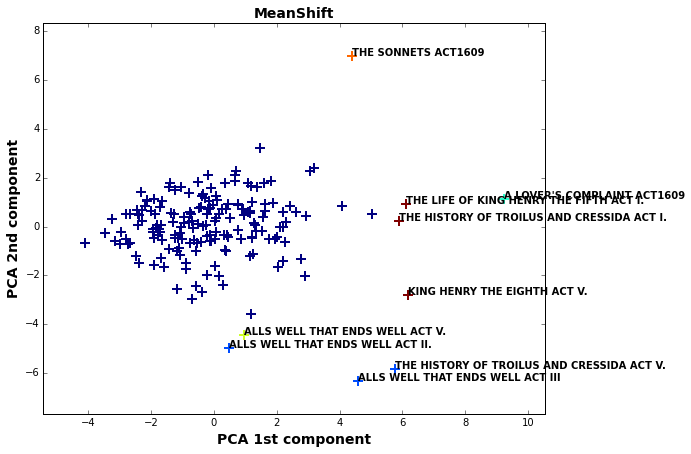
**SC = 0.070**

**SC = 0.081**



**SC = -0.191**

**SC = 0.081**

**SC = 0.315**

**SC = 0.085**

Figure Clustering results of different methods (SC: Silhouette coefficient)

According to the Silhouette coefficient, the spectral clustering is ruled out, as the average Silhouette coefficient is really low.

# Part IV Discussion and Conclusion

Visualization: PCA vs. MDS

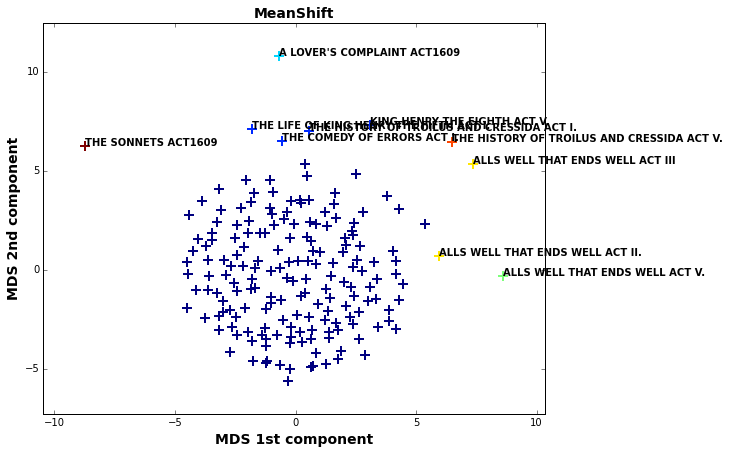
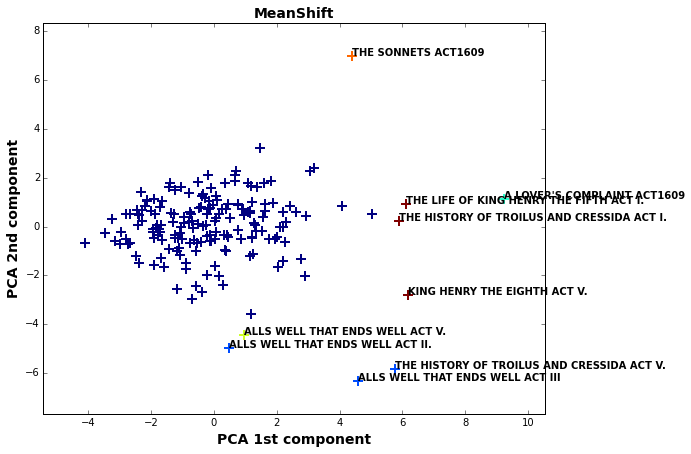
We use PCA reduction and 2D plot to draw the distribution and labeling of the clusters.

Features:

Clustering Methods:

4.2 Do any of the techniques give consistent results?

4.3 What did you conclude about the authorship?



1. Stanko, S., Lu, D., & Hsu, I. (2013). Whose Book is it Anyway? Using Machine Learning to Identify the Author of Unknown Texts. Machine Learning Final Projects. [↑](#footnote-ref-1)
2. Selman, S., & Husagic-Selman, A. (2011). Multilayered feedforward neural networks as a tool for distinction of the authors of texts. In Information, Communication and Automation Technologies (ICAT), 2011 XXIII International Symposium on (pp. 1-6). IEEE. [↑](#footnote-ref-2)
3. http://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette\_score.html [↑](#footnote-ref-3)