# Multiclass Classification of Tweets and Twitter Users Based on Kindness Analysis

Wanzi Zhou wanziz@stanford.edu
Chaosheng Han hcs@stanford.edu
Xinyuan Huang xhuang93@stanford.edu

## I. Introduction

Nowadays social networks such as Twitter and Facebook are most indispensable in people's daily lives, and thus it is important to keep the social community healthy. Establishing a kindness assessment mechanism is very helpful for maintaining a healthy environment, which could be used for applications like a rewarding system or parent control modes for children using social network.

Our goal is to set up a kindness rating system for tweets/ Twitter users. To accomplish this, we decompose the task into two stages: firstly, for stream data of tweets, we run unsupervised learning algorithms to classify them into three clusters: positive, negative and neutral. Secondly, we choose a group of Twitter users and apply our trained model to assess their kindness.

## II. Related work

In 2015, Cheng *et al* [1] from Stanford and Cornell Universities have developed a logistic regression model, using labeled posts to predict antisocial behavior in online discussion communities. Their study focuses on spotting out whether a user is a troll or not, which is a binary classification problem. Earlier in 2011, Sood *et al* [2] from Pomona College and Yahoo Company developed a model for automatic identification of personal insults on social news sites, which is also a supervised learning work and belongs to binary classification problem. They got their data labeled via Amazon Mechanical Turk. Meanwhile, sentiment analysis using Twitter data has been a popular topic in machine learning. Bifet and Frank [3] conducted a supervised learning with multinomial naive Bayes classifier to predict the sentiment and opinion of tweets. Pak and Paroubek [4] improved this model by better cleaning the input data. Agarwal *et al* [5] from Columbia University further explored tweets with a 3-way classification, namely positive, negative and neutral. All the mentioned research studies are supervised learning, however, it is infeasible to label enough training data in short time. Thus, different from former work, we propose to give each tweet/Twitter user a kindness rating, leading to an unsupervised multinomial classification or regression.

## III. Dataset and Features

Twitter has always been a great resource for Natural Language Processing researchers. It has sufficiently large size of data, along with outstanding qualities - it comprises of real-life conversations, uniform length (140 characters), rich variety, and real-time data stream. With Twitter API, we captured a random sample of tweets in continuous 24 hours in a regular day and picked out all the English tweets. After the above procedures, we obtained 58292 tweets as our dataset for this project.

We first used the lexicon features. We collected two lexicons of positive words [6] such as "amazing" and negative words [7] such as "bastard", which has 723 and 236 words respectively. We clean the data by transforming all the letters into lowercases and neglecting the punctuations. For every tweet we obtained from the dataset, we compare them to the words in the dictionary of both positive words and negative words, and obtain a $959 \times 1$ feature vector, where each value in the vector represents the number of times the word appears in a certain tweet. We then use the features to implement the learning part.

For a second try, since our data does not have labels, we want the features to be more reason-

able and objective so that the later unsupervised learning can lead to a better result, so we also tried considering the semantics and relations between words to assign a different weight to words in the dictionary. To achieve this, we used a word2vec method using an online dataset from the Data Compression Programs by Matt Mahoney[8]. Using the package in the program and fitting it into our model, we pre-processed the data file to obtain 17005207 words of all kinds(including positive words such as "optimistic" and negative words such as "bastard" in our positive and negative dictionaries). Among these there are 50000 unique words in all. Then we built a skip gram model and trained the model with SGD optimization for 40000 steps to obtain a $50000\times$ 128 word embedding matrix, where each row is the word embedding vector of each of the 50000 words. We extracted vectors for words in our positive and negative dictionaries from the matrix. Then, for each word, we compute its cosine similarity with every other words and take the average similarities of positive/negative words as a measurement towards "negative"/"positive". Based on above we assign different weights to build the $959 \times 1$ feature vector of every tweet for the learning part.

## IV. Unsupervised Learning Model

For the project we are using three methods to implement the unsupervised clustering: K-means, principle component analysis (PCA) incorporated with K-means and Gaussian Mixture Model with EM algorithm. We then compare the results between these methods.

### i. K-means

Since the data we have obtained are unlabeled data, we do unsupervised learning by classifying the tweets into three clusters: positive, negative and neutral. We first try very straightforward method of K-means clustering.

The input vector we have obtained through feature extraction is the feature vector containing information of use of positive words and negative words. We run K-means for all the 58295 tweets data.

We initial the cluster centroids based on the prior knowledge that the K cluster centroids should be well separated from each. We also add a random process in generating the centroids to avoid local minimum. Then we repeat the following K-means algorithms until convergence:
For every $i$, $i = 1, \ldots, 58295$, set

$$c^{(i)} = \arg\min_j ||x^{(i)} - \mu_j||^2$$

For every $j$, $j = 1, 2, ..., K$, set

$$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^i = j\}}$$

To choose the optimal cluster number $K$, we visualize the clustering results in a two dimension space, where the two dimensions represent the normalized sum of positive and negative feature number counts, respectively. We use this 2D projection result as a criterion to determine the optimal $K$ based on the fact that if the samples are well clustered in a low dimensional space, they must be better if not equally clustered in a higher dimensional space.

According to our clustering results with $K = 2, 3, 4, 5$, as shown in section V, we find the optimal cluster number $K = 3$. We then look into the values of the three cluster centroids. One of them is extremely close to a zero vector while the other two's positive and negative components are distinctly recognized, which shows that the three clusters correspond to the three categories: positive, negative and neutral as we discussed in the previous section.

### ii. PCA

After trying out straight forward K-means, we think it might be helpful to reduce the computation time by applying PCA (principal component analysis) before the K-means algorithms.

We first shrink the $959 \times 1$ (723 negative + 236 positive) feature vector to $605 \times 1$ by eliminating the word that never showed up in the dataset.

Then we normalize the feature data to zero mean and unit-variance for each component.

Afterwards, we calculate the the empirical covariance matrix $\Sigma$ of the feature data. Then we project our data into a $k$-dimensional subspace ($k < m$). Here we choose $k = 100$. Specifically we choose $u_1, \ldots, u_k$ to be the top $k$ eigenvectors of $\Sigma$. Then we present the feature vector on the basis of $u_i$'s.

## iii. Gaussian Mixture Model

To reflect the correlation between the individual components in the feature vector, we also use Expectation-Maximization (EM) algorithm to learn a Gaussian mixture model.

Since we have already demonstrated $K = 3$ is the optimal cluster number in the previous discussion, for Gaussian mixture model we use three Gaussians representing cluster 1 for neutral, cluster 2 for positive and cluster 3 for negative words. Our goal is to maximize the log likelihood

$$\ell(\phi, \mu, \Sigma) = \sum_{i=1}^{m} \log \sum_{z^{(i)}=1}^{k} p(x^{(i)}|z^{(i)}; \mu, \Sigma) p(z^{(i)}; \phi)$$

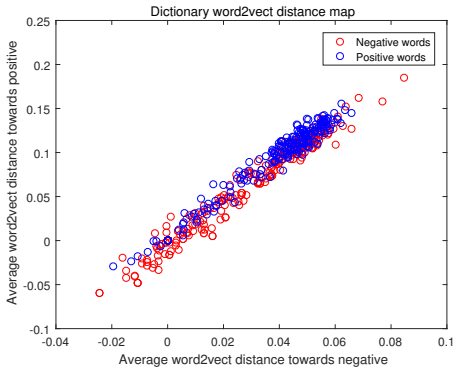
where $x^{(i)}$ is data of every tweet $i$, and $z^{(i)}$ is its corresponding latent variable in GMM. Here $k = 1, 2, 3$. The parameters $\phi, \mu, \Sigma$ for our GMM model is maximized by the EM algorithm. Then repeat the following EM algorithm until convergence:
E-step: we "guess" the value of $z^{(i)}$'s. Set

$$w_j^{(i)} = p(z^{(i)} = j|x^{(i)}; \phi, \mu, \Sigma)$$

M-step: update our parameters $\phi_j, \mu_j, \Sigma_j$ for every $j$.

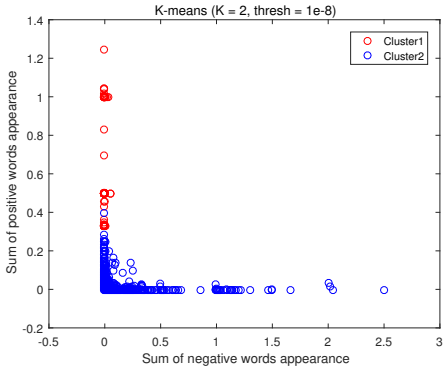
# V. Experimental Results & Discussion

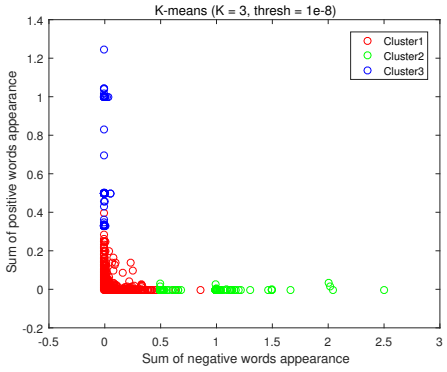
**Word2vec** The following figures show the 2D distance map of all the words in the word2vec model.

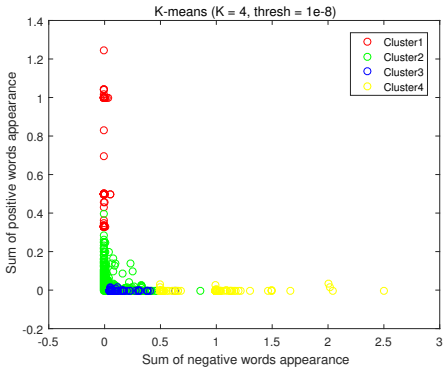


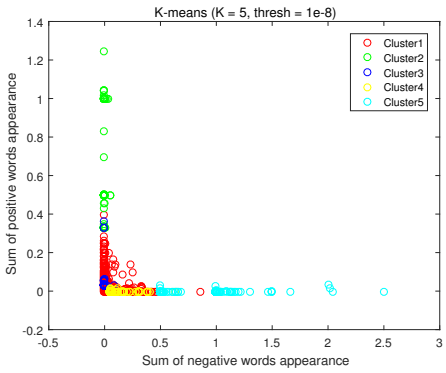

Comparing positive/negative words distribution in this 2D distance map, positive words tends to appear more on the upper-left of the map than negative words, which gives us a quantitative description of how "positive" or "negative" a word can be.

**K-Means** The following figures show the 2D projection results of applying K-means clustering with different cluster number $K = 2, 3, 4, 5$.





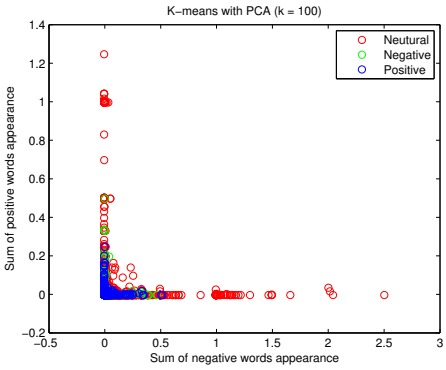




As mentioned in section IV, we use the 2D visualized result to determine the effectiveness of clustering with different cluster number $K$.

- For $K = 2$, the clustering is biased in either negative or positive direction which apparently is not a good result.
- For $K = 3$, the three clusters are symmetrically well separated from each other.
- For $K = 4, 5...$, we begin to see some finer structures inside of the clusters, while the clustering on the far end of the two direction follows the same pattern as $K = 3$.

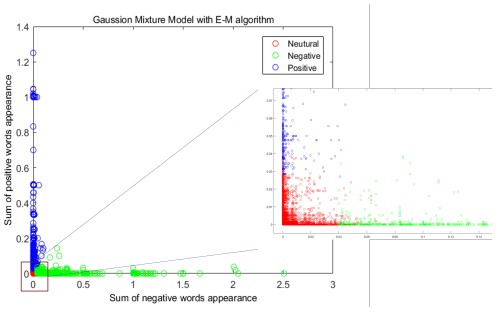
Therefore, we consider $K = 3$ as our optimal cluster number. Taking a deeper look into the three centroids values, we find that the green cluster represents the tweets containing more positive words and the blue cluster represents the tweets containing more negative words. The red cluster contains tweets that are mostly neutral, i.e, not containing many positive words or negative words. The result shows using K-means with $K = 3$ does a pretty good job in discerning negative, neutral and positive tweets.

With cluster number $K = 3$ and shrink dimension $k = 100$, below shows the result of applying K-means with PCA.

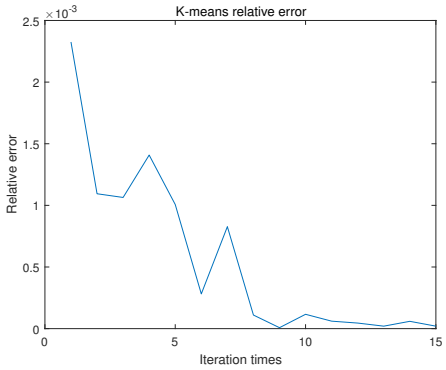


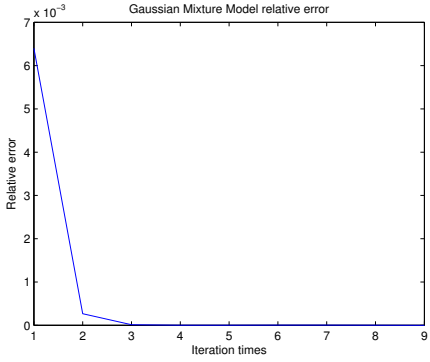

Surprisingly, K-means with PCA does not give us satisfactory result as it fails to distinguish among negative, neutral and positive tweets. We think it is because as the dimension of the feature vector shrinks, we lose some of the non-trivial information of the original tweets. Although some words in the dictionary could be strongly correlated such as "cock", "c-o-c-k" and "cocks", the amount of redundancy could only allow us to shrink the dimension to 1/2 or even less, since most of the words are unique to others.

**Gaussian Mixture Model** The following figure shows the 3-way classification with Gaussian mixture model. The plot of classification is similar to that of K-means with $K = 3$, but the neutral tweets area is smaller. Meanwhile, the area of positive and negative tweets expand a lot.




**Comparison** The following plots are the learning curves of K-means and GMM.




Both of the two algorithms converge quickly, and GMM converges even quicker than K-means. We then list the number of tweets in each category with K-means and GMM.

**Table 1:** *Multi-class Misclassification on 58295 Tweets*

| | K-Means | GMM |
|---|---|---|
| Positive | 110 | 1461 |
| Negative | 137 | 1129 |
| Neutral | 58048 | 55705 |

Comparing the results from K-means and Gaussian mixture model, we find that most of the tweets online are neutral. With K-means and Gaussian mixture model a proportion of 0.4% and 9% tweets are classified either positive or negative respectively, which shows that Gaussian mixture model can better recognize positive or negative tweets. This result is because with hard assignment, K-means only realizes spherical clusterings, while GMM considers probability and incorporates the covariance structure of data and adjusts itself to elliptic clusters.

**Application** We apply our trained GMM model to test on the recent 200 tweets of three US politicians Barack Obama, Donald Trump and Hillary Clinton. The result shows that they all follow the same pattern: while most of their tweets are neutral, their proportion of positive tweets are significantly higher than general public. This should be an expected result because intuitively politicians tend to convey more positive ideas and information to the public.

**Table 2:** *Model Test on Three US Politicians*

| | Barack Obama | Donald Trump | Hillary Clinton |
|---|---|---|---|
| Positive | 35 | 50 | 47 |
| Negative | 10 | 10 | 8 |
| Neutral | 155 | 140 | 145 |

## VI. Conclusion & Future Work

So far, the basic structure of the model is understood, and we have implemented the classification of positive, neutral and negative tweets and comparison between different unsupervised learning methods. We found that classifying tweets into three clusters(positive, neutral and negative) is currently most reasonable. Most tweets are neutral and a small portion of tweets are either positive or negative. K-means with PCA is not doing as good as K-means alone, we think it is because PCA actually removes non-trivial information in the feature vectors. Compared to K-means, Gaussian mixture model is performing better at classifying tweets into the three clusters because it considers the correlation between different components of feature. We tested our model on three US politicians and the result aligns with intuition.

Our next step is to use data of a set of individual tweeter users, based on their tweets history and build a model to give them a kindness score, thus establishing the kindness assessment system. Also we want to know deeper in the logic gap between positive and negative words on a psychological level.

## References

[1] Cheng, Justin, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec. "Antisocial behavior in online discussion communities." arXiv preprint arXiv:1504.00680 (2015).

[2] Sood, Sara Owsley, Elizabeth F. Churchill, and Judd Antin. "Automatic identification of personal insults on social news sites." Journal of the American Society for Information Science and Technology 63.2 (2012): 270-285.

[3] Bifet, Albert, and Eibe Frank. "Sentiment knowledge discovery in twitter streaming data." International Conference on Discovery Science. Springer Berlin Heidelberg, 2010.

[4] Pak, Alexander, and Patrick Paroubek. "Twitter as a Corpus for Sentiment Analysis and Opinion Mining." LREc. Vol. 10. 2010.

[5] Agarwal, Apoorv, et al. "Sentiment analysis of twitter data." Proceedings of the workshop on languages in social media. Association for Computational Linguistics, 2011.

[6] http://www.frontgatemedia.com/a-list-of-723-bad-words-to-blacklist-and-how-to-use-facebooks-moderation-tool/

[7] http://www.the-benefits-of-positive-thinking.com/list-of-positive-words.html

[8] http://www.mattmahoney.net/dc/