Sentiment Analysis of Tweets by Gaussian Mixture Models

Svetlana Sodol
Department of Computer Science
University of British Columbia
37381143

MohammadHossein Olyaiy
Department of Electrical and Computer Engineering
University of British Columbia
48502141

Abstract

Sentiment analysis in text is an important problem in Natural Language Processing, especially in high-volume contexts like Twitter. We investigate the use of Gaussian mixture models trained by the expectation maximization algorithm (EM) for the task of classifying positive or negative emotional colouring of tweets. This approach is compared to clustering by k-means algorithm with two approaches to feature generation: bag of words and 3-grams. We also compare samples generated by the resulting EM models to samples generated by kernel density estimation (KDE) to gain insight into what is the underlying model the EM has learned. Our results indicate that the EM models outperform the k-means models when using bag of words, and using 3-grams is more accurate than bag of words. When using n-grams, k-means and EM have similar performance.

1 Introduction

Social media, for example Twitter, have become vast collections of society's comments and opinions. In 2014, Twitter was accumulating 500 million tweets per day. This becomes problematic when we want to run analysis on the data, for example gathering opinions of groups of users, tagging fake news sharing, or blocking bullying and racism on the platform. With this amount of incoming data, this is impossible to do manually and thus natural language processing (NLP) techniques with usage of machine learning (ML) have been in development for these types of problems. However, when the task comes to working with the implicit content of the tweet text, for example the analysis of sentiment - the emotional colouring, many applications still are limited in what they can do, as they require at least some form of manual labelling or complex feature engineering. In this project we investigate Gaussian mixture models (GMM) in usage for tweet classification by sentiment. The GMM is an unsupervised clustering approach with a generative model. The clustering performance is compared to k-means and the generative performance of text samples is compared to kernel density estimation (KDE). Both clustering and generative results are compared across different dataset sizes and 2 simple feature generation strategies - bag of words (BOW) and 3-gram. We go on to describe some related works in Section 2, followed by details on our methods and experiments in Sections 3 and 4. We conclude with presentation of results in Section 5 and a discussion in Section 6.

2 Related Work

Processing of Twitter data does not need to be complicated. S. Gandharv [2017] describes a pipeline for cleaning and analysing tweets that is fast and readily available as packages in R. This, however, comes at the cost of flexibility of what kind of analysis can be performed and the paper focuses on a more simple task than unsupervised clustering on sentiment. The results of the pipeline presented are a simple visual representation of most commonly used words in the chosen dataset. Although this is

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

a useful area of insight, this is too simplistic compared to our investigation. The inferences drawn from this analysis are readily available in the data waiting to be extracted and it has been shown that this is solvable easily by computational approaches. Showcasing the big picture across the dataset, results such as these are not able to say anything about a specific tweet.

The work by Stefanov [2019] looks deeper into interpretation of the data available from Twitter. The work attempts to cluster Twitter users on 8 different topics based on similarity of their tweets. This can then be used as the labels for a supervised step where label-propagation is used to label less outspoken users. The strength of combining unsupervised learning with supervised is showcased as a very powerful tool in this domain, as it takes out the need of manual labelling. Labelling of users based on opinion sharing is useful for sociology and political research; however, for identification of negative sentiments or any implicit meanings of specific tweets - not that much, as these kind of tweets might be well out of character for a specific user.

Ledeneva [2011] showcasea an approach of using GMM for text summarization. The authors use the expectation maximimization (EM) algorithm to fit the GMM on the sentences of a text document. Each cluster of sentences then forms a sentence in the document summary. Summarization is more closely related to sentiment analysis, however, in usage with tweets we want to cluster whole tweets - whole "documents" - and not parts of it. Moreover, the type of language used in the news articles Ledeneva have used, differ greatly from the language used in tweets, especially in the presence of the character limit. The results of Ledeneva do indicate that EM is able to outperform k-means. In their comparison between feature models, the n-gram features also outperform the bag of words vectorization. This allows us to form some expectations that the EM performance with n-gram dictionary will be the best with all dataset sizes.

3 Methods

3.1 Dataset

The dataset we used is from a Kaggle competition "Twitter sentiment analysis". The training dataset presented has the texts of thousands of tweets with the sentiment tag. The sentiment tag is either of a positive emotional colouring tagged as 1, or the negative emotional colouring, tagged as 0. We use a subset of the training data so that we can measure the classification performance that we base on the fitted clusterings. Based on previous work, ML approaches, specifically k-means and EM, are able to find structures in the textual data and thus this dataset could be analyzed by them [Ledeneva, 2011].

We do very simplistic processing of the tweet data, commonly used in NLP: bag of words and n-grams. In the bag of words vectorization, each tweet becomes an array of 0's and 1's. The 1's correspond to an occurrence of the specific word in the tweet and the 0's to its absence, by index. An indexed dictionary is also created to keep track of which index is which English word. In the n-gram features, each dictionary index instead represents a 3-word phrase that has to occur together in the same order in the tweet in order for that tweet to have a 1 in the corresponding index of the final vectorization.

3.2 Algorithms

We use three main algorithms in our investigation. The EM algorithm is our main point of interest and has been shown to have better performance on clustering text data then the most common clustering algorithm of k-means [Ledeneva, 2011]. The other algorithm we compare the EM to is the KDE, which is also a generative model. KDE has been shown to also be appropriate for working with textual data in form of tweets by Ozdikis [2019] but in this context we use it as an alternative generative model for sample generation.

4 Experiment

Our contribution is based on 3 investigations, run on the same EM-generated model for each data size and feature type combination. The data size varies from 100 to 1000 tweets in increments of 100.

4.1 Sentiment Clustering

For this part of the investigation, two models are trained on the data - k-means and GMM by EM. The number of clusters is 2, as this is the number of sentiment labels available on which we will compare the classification. The classification is based on the predicted label cluster for each tweet. For each data size we include a 100 more tweets to use as a validation set to check for label permutation, and then we report the count of correctly classified tweets over the total number. These numbers show us if EM is able to outperform the k-means model on the clustering by sentiment.

As a measure of interest we also record the top 5 words of each cluster center description for the two models. This can indicate what kind of words the two models found as the most relevant for each cluster and if the words found as such are even relevant for sentiment at all.

4.2 Feature Generation

For each data size the previous investigation is repeated on different types of features of the input data - BOW and 3-grams. We can use the reported numbers to judge performance of BOW features as compared to 3-grams, as well as the combinations of algorithm and feature types. The 3-grams feature models might perform better as they account for more complex structures in a tweet, as the dictionaries are formed form 3-word sequences.

4.3 Sample Generation

For the last part of the investigation we use the same EM models as trained in the previous steps and compare its generated samples to samples formed from the same data using the same dictionary and feature representation by KDE. Furthermore, we used Principal Component Analysis (PCA) to project features into a lower dimensional space. Various number of PCAs were used in KDE to investigate what is the best trade off between feature dimensions and compute resources. KDE can tell us what kind of structures the models are able to pick up, if these are relevant for sentiment of the text at all, and if the text generated differs in how similar it is to human-generated text. All of the previous considerations of comparisons between BOW and 3-grams are also relevant; it might be the case that one of the feature representation type results in more human-like samples.

5 Results

5.1 Sentiment Clustering

As indicated in our results presented in Figures 1 (for BOW) and in Figure 2 (for 3-grams), EM consistently outperforms k-means at most data sizes. For the largest data size of 1000 tweets, the k-means cluster centers had these words as the most prominent: cluster 1 - "to the my it and" and for cluster 2 - "the my is http you" . The EM model had these words for the same dataset: cluster 1 - "to the my is and" and cluster 2 - "lot šà ff fettucini fetal". It is interesting to note that one of the clusters in both Em and k-means models share almost all the same words, which are all very common in English and do not carry much meaning in the sentence.

5.2 Feature Generation

Our results presented in Figure 2 (for 3-grams) indicate that the n-gram feature generation provides a small advantage in clustering used for sentiment classification for both k-means and EM. The reported metric of ratio of correctly classified tweets is larger (maximum being 0.593 for 900 tweets) and has a more rapidly growing slope over the increase in data size for the first few datasets, followed by a slow down, as compared to the results of Figure 1 with BOW features. In the 3-gram feature model, the differences in performance of EM over k-means are negligible.

For the words generated from the cluster descriptions in the 3-gram model, the k-means had these words: cluster 1 - "http tr im http bit ly http tumblr com http tinyurl com http blip fm " and cluster 2 - " lot lot lot šà à hitech guys too think had fun with had fun looking", with the EM having these words: cluster 1 - "http tr im http bit ly http tumblr com http tinyurl com http blip fm" and cluster 2 - "http bit ly there many gr8 dreamingspires there many 18sjxi http bit bit ly 6ew4q" . The 3-gram words

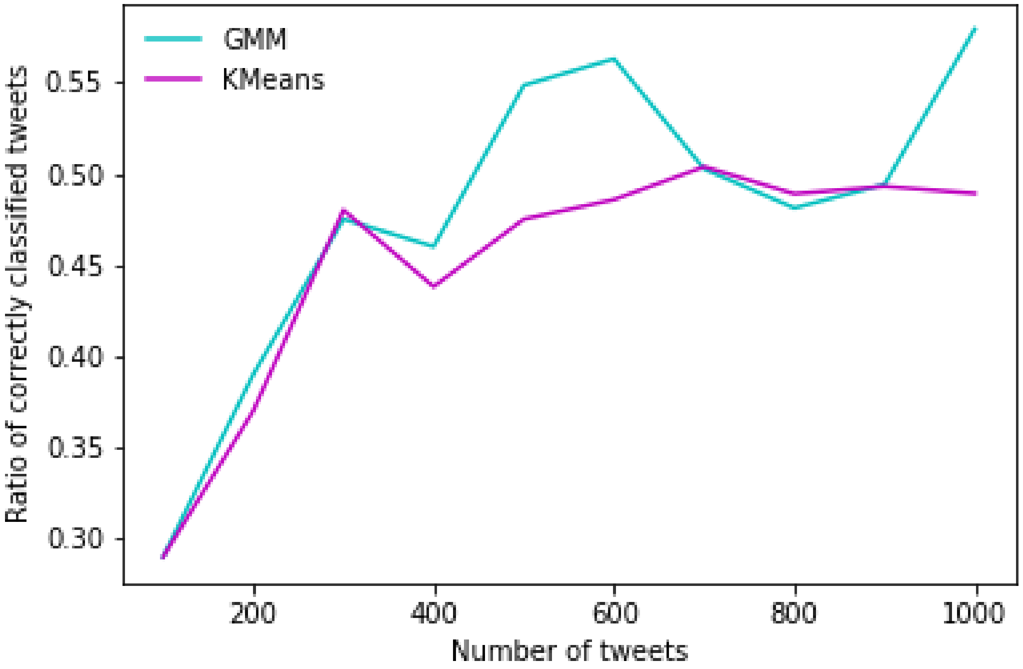


Figure 1: *This figure shows results of BOW features used for classification of sentiment based on the clusters produced by GMM and k-means.*

generated are not much more sensible. We do however see some actual "words" in 3-word phrases that occur together in the available text.

5.3 Sample Generation

Here, we present some samples generated from the fitted EM and KDE models. The KDE samples are much more readable and human-like, with the EM samples being much more nonsensical. The order of the words in the samples are alphabetical as they are directly transformed from the dictionary, but the KDE samples could be used to get some meaningful sentences.

5.4 GMM with EM samples

and for have in is it just my
about but do im my out quot that the to up was work
can http is my now quot the
imisscath just lot
lot you
and me
mcflyforgermany me no of the
http my quot
and is lot my that the this with

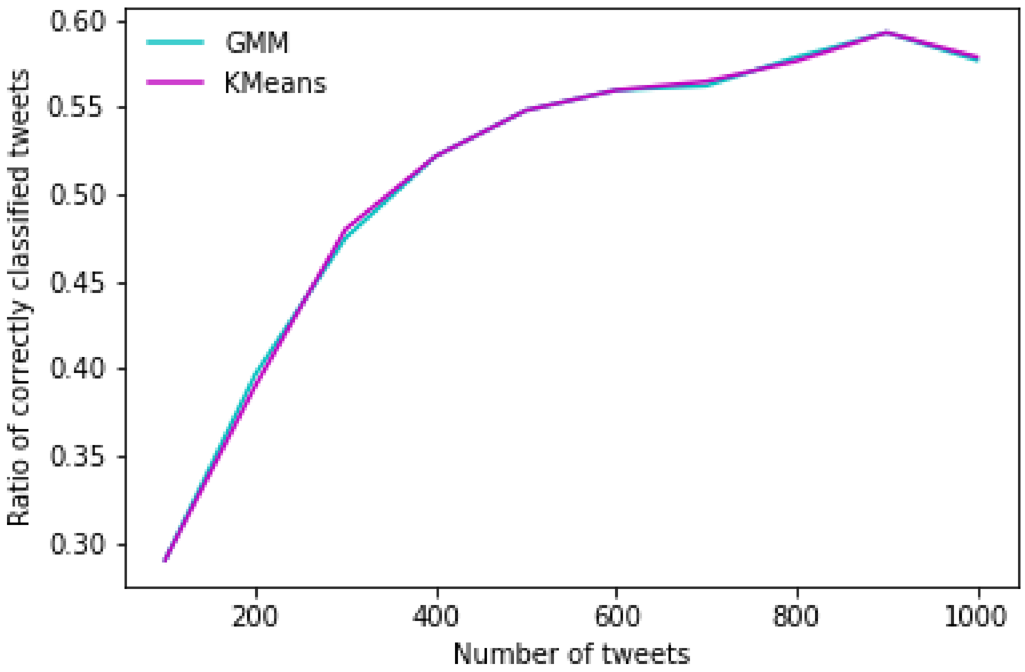
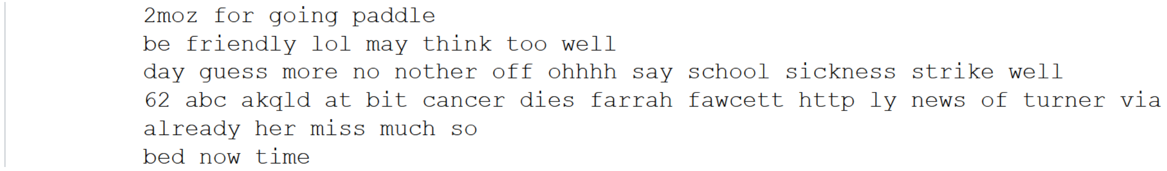


Figure 2: *This figure shows results of 3-gram features used for classification of sentiment based on the clusters produced by GMM and k-means.*

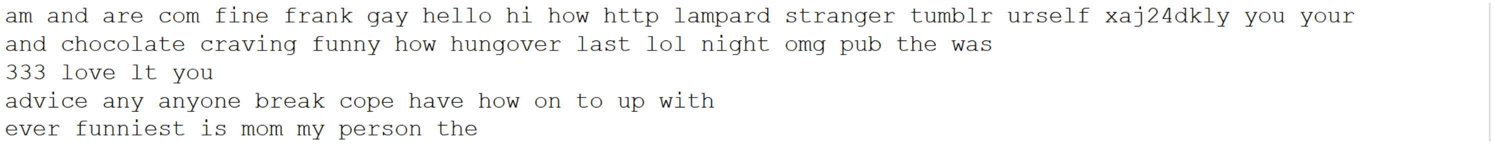


```
2moz for going paddle
be friendly lol may think too well
day guess more no nother off ohhhh say school sickness strike well
62 abc akqld at bit cancer dies farrah fawcett http ly news of turner via
already her miss much so
bed now time
```

Figure 3: *This figure shows KDE-generated samples from 500 tweets and 1 of original features as the number of PCA components.*

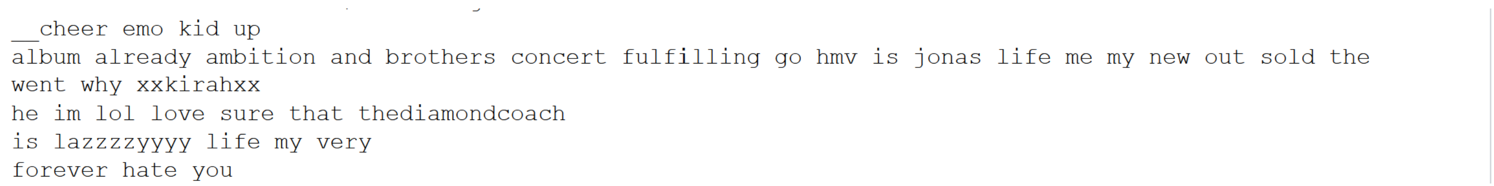
5.5 KDE

We tried KDE using different number of principal components and data points. Below are examples of data generated with the first quarter and first half of the principal components. Also, we have included the results for KDE on both 500 and 1000 data samples.



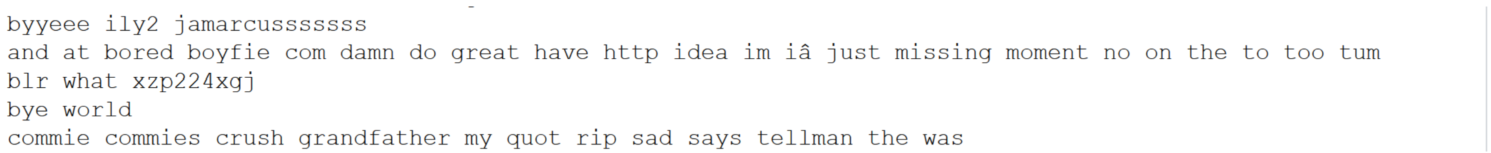
```
am and are com fine frank gay hello hi how http lampard stranger tumblr urself xaj24dkly you your
and chocolate craving funny how hungover last lol night omg pub the was
333 love lt you
advice any anyone break cope have how on to up with
ever funniest is mom my person the
```

Figure 4: *This figure shows KDE-generated samples from 500 tweets and 1 of original features as the number of PCA components.*



```
__cheer emo kid up
album already ambition and brothers concert fulfilling go hmv is jonas life me my new out sold the
went why xxkirahxx
he im lol love sure that thediamondcoach
is lazzzzyyyy life my very
forever hate you
```

Figure 5: *This figure shows KDE-generated samples from 900 tweets and 1 of original features as the number of PCA components.*



```
byyeee ily2 jamarcussssssss
and at bored boyfie com damn do great have http idea im iâ just missing moment no on the to too tum
blr what xzp224xgj
bye world
commie commies crush grandfather my quot rip sad says tellman the was
```

Figure 6: *This figure shows KDE-generated samples from 900 tweets and 1 of original features as the number of PCA components.*

6 Discussion and future work

Our investigation indicates that the models trained on 3-gram features outperform BOW models. EM was able to achieve better performance in sentiment classification over the models trained by k-means only using the BOW features. The EM and k-means with 3-grams were able to attain 0.58 accuracy on the largest data size we tested. This performance was still achieved in light of poor quality of generated samples and descriptive words for clusters, as compared to KDE.

The strength of our contribution is the number of comparisons we show results for in the usage of EM in this context. We not only compare its classification on sentiment ability to k-means, but we also report generated samples from these models, words found to be most descriptive of the found clusters and report on what feature representation is beneficial to the usage of EM. This is also almost a replication of previous results comparing EM with k-means, and BOW and n-grams, the only difference is that EM did not outperform k-means in the 3-gram cases [Ledeneva, 2011].

One of the greatest weaknesses of our contribution is the size for the datasets that we were able to use due to limitations of the computational resources available to us. Another weak point comes from the lack of pre-processing of the tweet data. Part of the low correctness of classification can be attributed to this, as the models would find clusters of tweets that are, for example, similar as they share bit.ly links or any https links in general. This is an interesting insight, but is irrelevant for sentiment analysis. However, not all of the misclassification can be accounted by this. Even though we attempt to classify only a binary sentiment, the actual tweet data does not necessarily come from a simple binary model. Thus, any model is likely to fit the existing similarities in the text and then only approximate the sentiment distinction. This is of course only in light of the simple features that we have used like BOW and 3-grams. If the features also contained the sentiment data of the actual words that could be extended form something like a word2vec representation, then it is possible that better results might be attained. Although, our results were surprisingly good given these limitations and the small data sizes.

For future work we would consider increasing our data and feature representations. As well, the KDE samples generated look promising and as a future direction more algorithms like this could be compared to performance of EM. It would be interesting to also focus more fine-grainely on just one aspect of using EM to gain more detailed results. Our attempt to look at the resulting cluster descriptions from the EM and k-means models lead to questionable results and would need more investigation to draw any concrete conclusions. It was an interesting finding in the cluster words and samples to see that the models picked up a lot of link sharing from the available tweets. Investigating Twitter as a platform to share links to other platforms would be another interesting direction.

Acknowledgements

The work presented in this paper was done using the Python scikit-learn algorithm implementations. Specifically, for the KDE and PCA we have consulted an example from documentation available at https://scikit-learn.org/stable/auto_examples/neighbors/plot_digits_kde_sampling.htmlsphx-glr-auto-examples-neighbors-plot-digits-kde-sampling-py .
The dataset was taken from the Kaggle competition "Twitter sentiment analysis" available at https://www.kaggle.com/c/twitter-sentiment-analysis2/data.

References

P. Stefanov, K. Darwish, and P. Nakov, “Predicting the Topical Stance of Media and Popular Twitter Users,” arXiv:1907.01260 [cs], Jul. 2019, Accessed: Mar. 31, 2020. [Online]. Available: http://arxiv.org/abs/1907.01260.

S. Gandharv, V. Richhariya, and V. Richhariya, “Real Time Text Mining on Twitter Data,” IJCA, vol. 178, no. 3, pp. 24–28, Nov. 2017, doi: 10.5120/ijca2017915779.

Y. Ledeneva, R. G. Hernández, R. M. Soto, R. C. Reyes, and A. Gelbukh, “EM Clustering Algorithm for Automatic Text Summarization,” in Advances in Artificial Intelligence, Berlin, Heidelberg, 2011, pp. 305–315, doi: 10.1007/978-3-642-25324-9_26

O. Ozdikis, H. Ramampiaro, and K. Nørvåg, "Locality-adapted kernel densities of term co-occurrences for location prediction of tweets", Information Processing Management, vol. 56-4, Jul. 2019, pp. 1280-1299. https://doi.org/10.1016/j.ipm.2019.02.013

P. Stefanov, K. Darwish, and P. Nakov, “Predicting the Topical Stance of Media and Popular Twitter Users,” arXiv:1907.01260 [cs], Jul. 2019, Accessed: Mar. 31, 2020. [Online]. Available: http://arxiv.org/abs/1907.01260.