Emojis and Weather

Darryl Hannan
Villanova University

Emma Chin
Villanova University

Matt Mador
Villanova University

December 11, 2017

Abstract

A significant amount of research has been done analyzing the impact weather has on emotional states. Most of these studies are survey based, and therefore have a small sample size. Here, we seek to overcome this problem through the use of social media. We analyze tweets from various geographic locations, extracting emojis. We use the distribution of emojis to represent the emotional state of a population. We use these representations to train a machine learning classifier that predicts the weather. Our model is approximately 88% accurate, providing support for the link between weather and mood.

1 Introduction

The impact that weather has on mood is well established. Spasova (2011) found that abrupt transitions to unfavorable weather conditions has a negative effect on emotion.[7] Keller et al. (2005) found that time spent in pleasant weather results in better mood and improved memory.[2] Some effects are more serious; seasonal effective disorder occurs occurs during winter months, when the days are shortest, and can result in serious cases depression.[4]

The effect that weather has on emotions varies significantly among individuals.[7] Therefore, large sample sizes are needed to discover correlations between specific weather patterns and emotional states. Traditionally, the only way to study this set of phenomena was through survey based methods, resulting in small sample sizes. We attempt to overcome this problem through the use of Twitter. By analyzing the emotions expressed in tweets for a given geographic location, we obtain an approximation of the collective emotional state of the population. This method also makes it significantly less expensive to collect data. By running the collection program on a server, we can continuously collect data indefinitely.

2 Related Work

Park et al. (2013) also utilize Twitter to gather emotional data.[6] They use a linear regression model to predict various variables such as temperature, humidity, and precipitation. Our approach differs in that Park et al. perform sentiment analysis on the tweets, whereas we use emojis to represent sentiment. By using emojis, we require significantly less preprocessing and impose less structure on the emotional representation that the model ultimately learns.

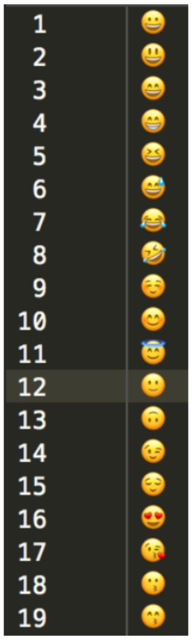
Baylis et al. (2017) use both Twitter and Facebook to collect emotional data. Using more than 3.5 billion social media posts in their model.[1] Their weather variable, referred to as cloud cover, is similar to the variable that we predict in our model. The primary difference in our model again lies in the use of emojis.

3 Data

3.1 Emotion Data

We collect tweets from 21 cities using Twitter's Streaming API via Tweepy[1], targeting locations with high populations in order to maximize the number of data points. We preselected a list of 75 emojis that commonly represent various emotional states (Figure 1). As each tweet is received, we count the occurrences of each emoji in the tweet. We collect data in 2 hour intervals. The distribution of emojis for each interval is represented in a vector format, where each dimension of the vector contains the count of a given emoji. Data was collected over a period of approximately 2 weeks.

[1] http://www.tweepy.org/



```
1  😀
2  😃
3  😄
4  😁
5  😆
6  😅
7  😂
8  🤣
9  ☺️
10 😊
11 😇
12 🙂
13 🙃
14 😉
15 😌
16 😍
17 😘
18 😗
19 😙
```

Figure 1: Example of emojis that we extract from tweets

3.2 Weather Data

We access weather data for each location through the OpenWeatherMap API. We use PyOWM[2], an open-source python wrapper, to simplify API calls. Weather data is collected concurrently with Twitter data. At the end of each 2 hour interval, we collect the current atmospheric condition in each city. Examples of atmospheric conditions include rain, snow, clear, etc. In the data we collected, there are 8 classes of conditions: clouds, clear, rain, drizzle, mist, fog, snow, haze, and dust.

3.3 Preprocessing

One issue that we encountered was that some of the emoji vectors were extremely sparse. In some cities, such as Nashville, almost every vector only had a few emojis. Therefore, we only include vectors in which the sum of its elements is greater than 40. For classification, each atmospheric condition is represented using a number from the range 0-7. In the case of

[2] https://github.com/csparpa/pyowm

binary classification, we consider clear to be a positive weather condition and represent it with a 0. We consider all other conditions to be negative and represent these using a 1. Our final data set consisted of 977 data pairs.

4 Methods

We compare 5 learning algorithms on our classification tasks: ZeroR, a Gaussian Mixture Model (GMM), an Artificial Neural Network (ANN), a Decision Tree, and Naive Bayes. We use ZeroR to establish a baseline, and we use a GMM to evaluate how our data clusters. We implement all models, with the exception of the ANN, using Scikit-Learn[3]. The models do not benefit significantly from modifying parameters. Results can be achieved by leaving them to their default values.

Our ANN is implemented using Keras[4]. The model architecture can be seen in Figure 2. We use a ReLU activation function, binary crossentropy to evaluate our loss, and adam as our optimizer[5][3]. The model is trained for 20 epochs.

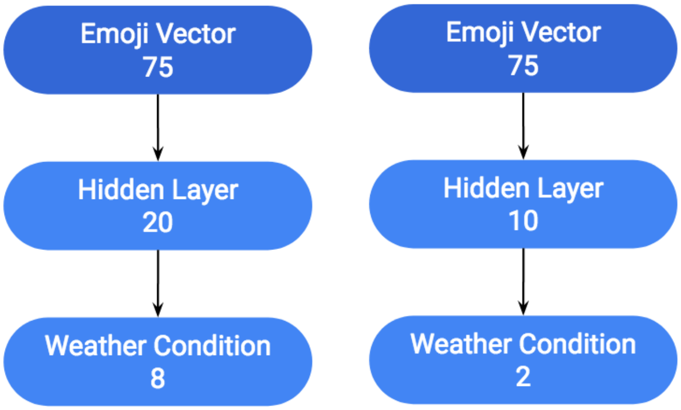


Figure 2: Structure of our artificial neural networks. **Left:** ANN for standard classification. **Right:** ANN for binary classification. The numbers below the name of each layer represents the number of nodes in the layer.

[3] http://scikit-learn.org/stable/

[4] https://keras.io/

5 Results

5.1 Binary Classification

We first frame our problem as binary classification. We thought that a simpler form of classification may yield better results. A description of the 2 classes can be seen in section 3.3. The Gaussian Mixture Model performs the worst, at 51%, indicating that the data does not cluster well. On average, ZeroR outperforms all of our other models, at 59%. This indicates that the models are unable to learn a relationship between the emoji vectors and our 2 classes. Most likely, the limited success is due to the nature of the classes. Some atmospheric conditions that we label as negative might not belong in that class. For instance, snow may actually be a positive condition. See Figure 3 for a detailed breakdown of the accuracies associated with each model.

5.2 Standard Classification

We next framed our problem as a classification task with 8 classes. A description of the classes can be seen in section 3.3. Here, Naive Bayes performs the worse, at 19% accuracy. This is most likely due to the lack of statistical independence among emojis. Most notably, multiple emojis can come from the same tweet; such emojis are certainly not independent. The ANN significantly outperforms the other models, averaging 88% accuracy. These results indicate that there does exist a relationship between emoji vectors and their atmospheric class. However, the relationship is too complex for models other than the ANN. Figure 4 provides a detailed breakdown of the accuracy of each model. Figure 5 provides the accuracy of the ANN on each of our 8 classes. The model does not just learn a few classes well, but achieves reasonable accuracy on all common classes.

6 Conclusions

Our results support the relationship between weather and emotion. Based upon the distribution of emojis used in a given city, our ANN was able to predict the atmospheric condition of that city with 88% accuracy. Although our hypothesis was supported, we cannot draw specific conclusions regarding the relationship. To do this, further analysis is needed.

Most of the challenges that we encountered were related to the Twitter API. Initially, we believed that we could use prior tweets to generate our data. However, we learned that the Twitter API only allows you to collect tweets from the previous week. Furthermore, there is a cap on the number of tweets you can collect of 1500. We resolved this problem by collecting tweets in real time using the Twitter Streaming API. The consequences of collecting data this way was that we had to wait a couple weeks before we had any data to work with. Given the restricted time frame for the project, this limited the time that we could spend developing and evaluating our models.

Future work would involve determining which emojis have the strongest correlation with each weather condition. From this information, we could directly infer which emotions are elicited from each weather condition. We would also benefit from having a larger dataset. We could expand our scripts to gather tweets from all around the world with relative ease. This would also open up the possibility of analyzing the effect across cultures.

References

[1] Patrick Baylis, Nick Obradovich, Yury Kryvasheyeu, Haohui Chen, Lorenzo Coviello, Esteban Moro, Manuel Cebrian, and James H. Fowler. Weather impacts expressed sentiment. aug 2017.

[2] M. C. Keller, B. L. Fredrickson, O. Ybarra, S. Cote, K. Johnson, J. Mikels, A. Conway, and T. Wager. A Warm Heart and a Clear Head: The Contingent Effects of Weather on Mood and Cognition. *Psychological Science*, 16(9):724–731, sep 2005.

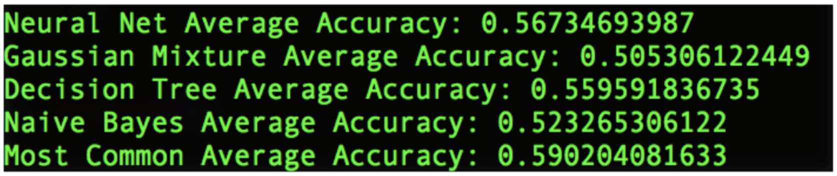
[3] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *ICLR*, dec 2015.

[4] Sherri Melrose. Seasonal Affective Disorder: An Overview of Assessment and Treatment Approaches. *Depression Research and Treatment*, 2015:1–6, nov 2015.

[5] Vinod Nair and Geoffrey E Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. *Proc. 27th International Conference on Machine Learning*, 2010.

[6] K Park, S Lee, E Kim, M Park, J Park, and M Cha. Mood and weather: Feeling the heat? *7th International AAAI Conference on Weblogs and Social Media, ICWSM 2013*, pages 709–712, 2013.

[7] Z Spasova. The effect of weather and its changes on emotional state individual characteristics that make us vulnerable. *Adv. Sci. Res*, 6:281–290, 2011.



```
Neural Net Average Accuracy: 0.56734693987
Gaussian Mixture Average Accuracy: 0.505306122449
Decision Tree Average Accuracy: 0.559591836735
Naive Bayes Average Accuracy: 0.523265306122
Most Common Average Accuracy: 0.590204081633
```

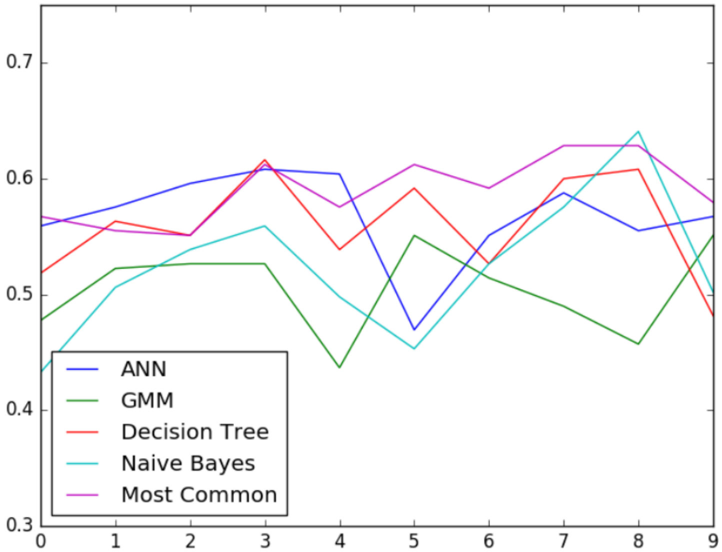
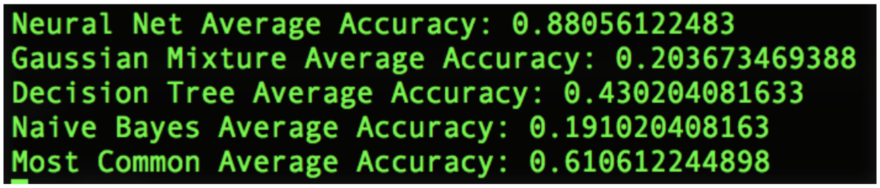


Figure 3: Binary classification results. **Top:** Average Accuracy for each model **Bottom:** Accuracy for each model over 10 trials.



```
Neural Net Average Accuracy: 0.88056122483
Gaussian Mixture Average Accuracy: 0.203673469388
Decision Tree Average Accuracy: 0.430204081633
Naive Bayes Average Accuracy: 0.191020408163
Most Common Average Accuracy: 0.610612244898
```

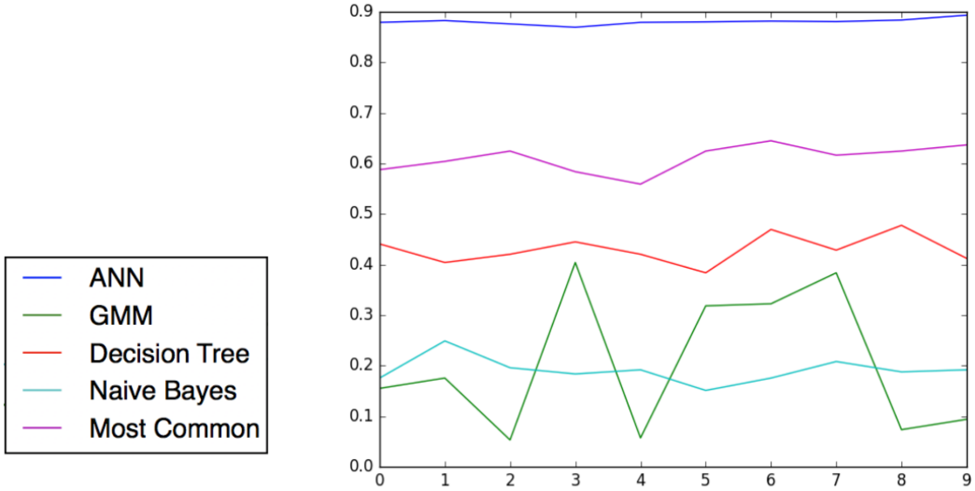
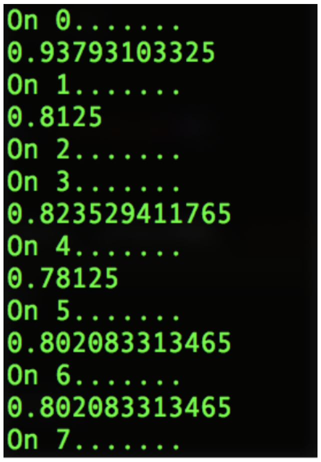


Figure 4: Standard classification results. **Top:** Average Accuracy for each model **Bottom:** Accuracy for each model over 10 trials.



```
On 0.......
0.93793103325
On 1.......
0.8125
On 2.......
On 3.......
0.823529411765
On 4.......
0.78125
On 5.......
0.802083313465
On 6.......
0.802083313465
On 7.......
```

Figure 5: Accuracy of the artificial neural network for each of our 8 classes. Each classes integer value is printed, along with the percentage immediately below it.