Deep learning for bipolar disorder: anticipating manic and depressive episodes

Shreya Shankar
Department of Computer Science
Stanford University
shreya@cs.stanford.edu

Abstract

We explore deep learning for mental health, particularly how to anticipate manic and depressive episodes for bipolar disorder. Text messages have been shown to indicate moods in bipolar people, and we use unsupervised learning methods to identify anomalies in text message data and correlate such anomalies to episodes. A LSTM autoencoder achieves around 57% test precision and 43% test recall. We hope these results motivate future study of how machine learning methods can be used to help people with mental illnesses.

1 Introduction

Bipolar disorder, a mental illness, is defined as a brain disorder that causes unusual shifts in mood, energy shifts, and activity levels (NIH). It deeply affects a person's ability to carry out everyday tasks, and more than 2.8% of US adults have this disorder. A person with bipolar disorder shifts between manic (high) and depressive (low) episodes. During manic episodes, the following symptoms may be exhibited: abnormal upbeat manner, increased activity or energy, increased ego, decreased sleep, engagement in "risky" behaviors, and more. During depressive episodes, the following symptoms may be exhibited: low mood, irritability, hopelessness, fatigue, indecisiveness, and more.

I have been diagnosed with bipolar disorder, and I find it hard to cope with the symptoms and rapid cycling. I feel as though my episodes come out of nowhere, and I struggle to make decisions that are best for me in the long-run. A large reason for why I am unable to do so is because I am unable to recognize that I am in a manic or depressive state, which are different from my original state.

Although manic and depressive episodes vary in terms of symptoms exhibited, one common theme between the two types of episodes is shared: both types of episodes are "anomalies" in patients' behaviors. Over the past decade, technology has evolved such that most people frequently engage in text messaging as one of the more dominant forms of communication in their lives. Lots of data is produced by a single person's text messaging patterns, which makes text message data interesting to study to analyze patients' change in behaviors through manic and depressive episodes.

The question we ask is: is it possible to predict in advance when I will have manic and depressive episodes based on my text messages? Major challenges for this project include dataset cleaning and curation, defining the right features, constructing a model to learn from my data, and the fact that this is an unsolved problem. Additionally, most people don't have labeled manic or depressive episodes until after the episodes, so a large part of this work is an unsupervised learning problem of trying to identify manic or depressive episodes.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

2 Related work

Machine learning has been applied to several areas in mental health with varying degrees of success. In a survey paper, Shatte et al. find that the most common applications of machine learning techniques include depression, schizophrenia, and Alzheimer's [7]. Most methods include traditional statistical techniques such as SVMs and decision trees, but some deep learning work has been done. Specifically, they found that neural networks have been applied to clinical anxiety, autism, dementia, depression, OCD, and schizophrenia. Durstewitz et al. state that a large amount of the work in deep learning applied to mental health relates to diagnosing mental health problems [2].

In this paper, we are interested in exploring the intersection of deep learning and biploar disorder. Some work in behavioral science has indicated that text messaging patterns exhibited by bipolar patients are different when the patients are experiencing hypomanic or depressive episodes. Emeagwali et al. show that young adults with bipolar disorder dramatically increase the quantity of texts sent when experiencing a hypomanic episode and call a need to consider forms of communication such a texting when evaluating patients' hypomanic and depressive patterns [3]. Additionally, some work has been done in analyzing mobile keyboard activity of bipolar patients. Zulueta et al. analyze mobile keystroke speed and patterns in bipolar patients' use of text messaging and social media to find that hypomanic episodes come with fast typing speed and large quantities of messages, while depressive episodes come with large quantities of text in a single message and slower typing speeds [8].

Since it is clear that text message data might have some signal in identifying episodes in bipolar patients, we can formulate this problem as an anomaly detection problem. Chalapathy et al. perform a survey on deep learning techniques in anomaly detection and find that autoencoders comprise a lot of models that solve unsupervised anomaly detection [1]. Maurya et al. state that supervised anomaly detection can be difficult when there is a large class imbalance [5], which in our case holds true because there aren't too many episodes compared to "normal" days.

Based on prior and related work, we choose to treat the problem as an unsupervised anomaly detection problem to identify hypomanic or depressive episodes based on text message data. In my case, we have labels from my psychiatrist, which we will compare to after training the unsupervised model.

3 Dataset and Features

I collected a dataset of my text messages from March 8, 2019 until October 15, 2019. My primary messaging platforms are:

- iMessage
- Messenger
- Signal

Collecting the messages sent and received from each of the platforms proved to be a challenging task. Fortunately, iMessages are stored in a `sqlite3` database in the filesystem. I wrote a script to read the messages from the database. The earliest message date was in March 2019, and the latest message date was in October 2019. I wrote out all of the messages I sent, along with the timestamp, to a `csv` file.

To get my Facebook Messenger messages, I downloaded a JSON of my personal data from Facebook. This JSON includes messages, so I filtered them by the following criteria to mimic the data from iMessage:

- Timestamp greater than mid-March 2019 (to match that of iMessage data)
- Sender is me

I even included messages I sent to group chats from both iMessage and Messenger. Unfortunately, I was unable to scrape my Signal messages, since those are stored in an encrypted format on my computer, and having the plaintext is necessary to perform any analysis.

In total, the number of messages collected was 60,960. I joined dataframes of my scraped iMessages and Messenger messages into another dataframe, and saved that dataframe into a `csv` to be fed into the model. The columns of the `csv` were *message content* and *timestamp*.

Running `df.head()` (printing out the first 5 messages) gives us:

```
tbh i’ll walk & call, 2019-09-07 20:02:41.427
oh i love backyard brew okay i’ll come right after, 2019-09-07 20:02:30.248
also there is some event at the googz at 4pm, 2019-09-07 19:58:52.246
just have phone call 1pm-1:30pm, 2019-09-07 19:58:22.875
or can stay here idc, 2019-09-07 19:58:18.482
```

I treated each message as an individual data point, making the number of data points in the dataset approximately 60,000. To preprocess the data before feeding into the model, I used 100-dimensional GloVe embeddings trained on text data from Twitter [6]. I also used the Python NLTK library to remove stop words from the text message data [4]. Furthermore, I stripped punctuation and lowercased all the words.

The dataset is randomly split into 60% train, 20% dev, and 20% test sets, but stratified by week – meaning, for every week, 60% of days are in the train split, 20% of days are in the dev set, and 20% of days are in the test set. The features fed into the model are the GloVe vectors representing words in the text message data. More information on the model is in the methods section.

4 Methods

Now that we have the text message data, the goal is to identify anomalies in the data in an unsupervised fashion. At a high level, I train an autoencoder to reconstruct each individual text message. We use autoencoders because they reconstruct the original sample to the best of their ability, and we can use the reconstruction error as a measure of how similar data points are to each other. Samples that have high reconstruction error are classified as anomalies. For each day, I add up my reconstruction error to get a “daily deviation value.” I choose a threshold for the daily deviation value based on my training results.

The autoencoder is built using Keras’ Sequential Model API. The encoder consists of two `LSTM` layers with hidden size 100 each and activations of sigmoid. A `RepeatVector` layer exists between the encoder and decoder. Then, the decoder consists of `LSTM` layers with hidden size 100 each and activations of sigmoid. The final layer is a dense layer of size 1 wrapped in a `TimeDistributed` layer. We use `LSTM` layers because they take advantage of the time-dependent nature of texts, since each word in a text has a dependence on previous words in the texts.

The model is compiled with mean squared error loss, Adam optimizer, input max sequence length of 50 (50 words maximum per text message), and batch size of 128. The model is run for 300 epochs and takes about 12 hours to complete training.

Now, each message has a matrix associated with it. As mentioned previously, the input maximum sequence length is 50. Messages shorter than 50 words are zero-padded, while messages greater than 50 words are truncated. 50 is chosen as a hyperparameter because most messages are less than 50 words.

5 Experiments/Results/Discussion

For the analysis, it is helpful to know when I had hypomanic and depressive episodes. Labeled data from my psychiatrist appointments includes the following:

- Week of 4/1/2019
- Week of 4/22/2019
- Week of 6/10/2019
- Week of 7/1/2019
- Week of 8/5/2019
- Week of 8/12/2019
- Week of 8/26/2019

Data split	Average MSE	Number of days identified	Precision	Recall
Train	325340.48	11	64%	57%
Dev	373148.62	26	56%	43%
Test	384620.92	28	57%	43%

Figure 1: Results for data splits

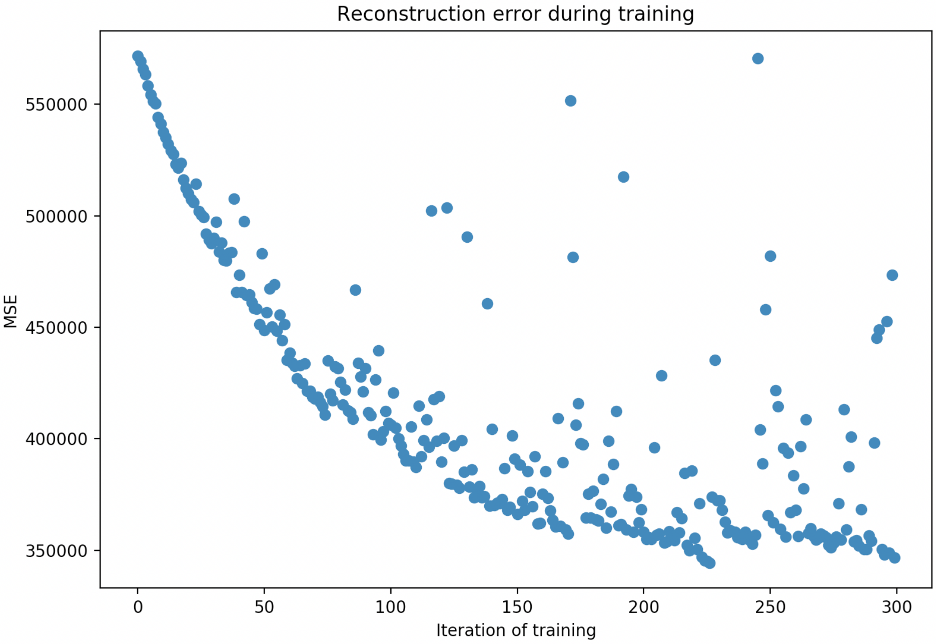

Figure 2: Reconstruction error over training

Unfortunately, we only have labeled data on a week-by-week basis, not individual dates in the year, because I only visit the psychiatric hospital every week. During training, our goal is to minimize MSE in the reconstruction of individual text messages, and then choose a cutoff for the MSE to identify which dates should be classified as episodes. We visualize the reconstruction error over time as training in figure 2 to make sure the training looks reasonable. Since the overall error is decreasing over time, we move to deciding the MSE cutoff. When we set the threshold for MSE to be around 500,000, we get the precision and recall in train, dev, and test splits as shown in figure 1. This figure also shows the MSE for each data split. Here, a data point counts towards precision if it lies within one of the weeks labeled by the psychiatrist data. A week in the psychiatrist data is "recalled" if at least one of the model's predicted dates lies in that week.

We can see that the model overfits somewhat to the training set, as the results for the training set are better than the results on the dev and test sets. Unfortunately, I did not have time to explore techniques to minimize overfitting, but given more time, I would add dropout to the network. The hyperparameters used for the network are as follows: 4 LSTM layers, hidden size of 100, activations of sigmoid, max sequence length of 50, 300 epochs, learning rate of 0.001, and batch size of 128. I chose these hypermarameters by doing hyperparameter testing on the dev set. I did not change the hyperparameters of the Keras default Adam optimizer, so that could be some future work.

After verifying that training was working as intended and choosing the right hyperparameters, I visualized the reconstruction errors for every day represented in the dataset, as seen in 3, and picked some examples of texts where the reconstruction errors were high, and examples where the reconstruction errors were low, as seen in 4. From these results, it was evident that the model did well in reconstructing short texts, and did poorly in reconstructing long texts that seemed to be copy-pasted from other sources (i.e. medicine description, error message found while debugging, etc). All in all

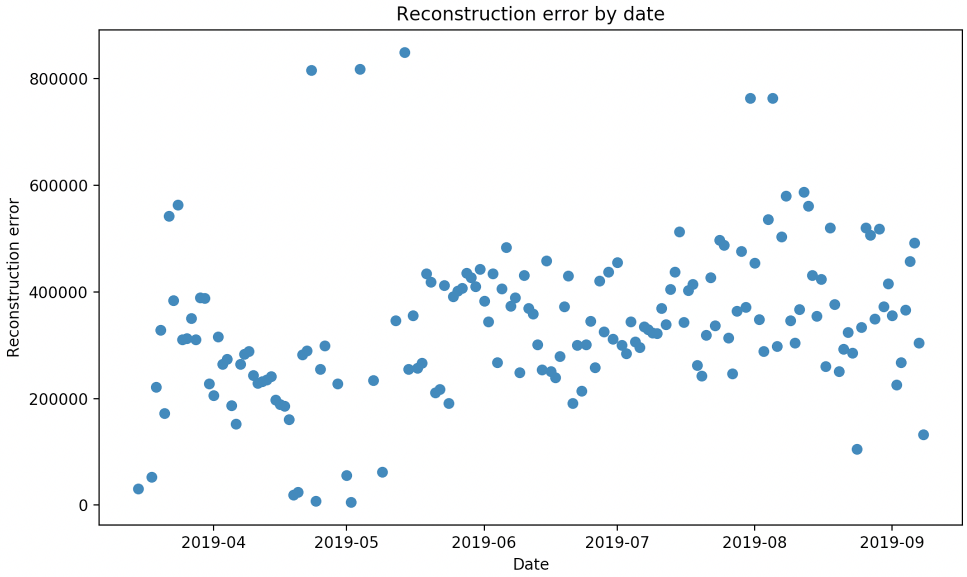

Figure 3: All reconstruction errors

Text	High or low	Reconstruction error
"When avoid drinking alcohol completely Completely avoid drinking alcohol taking metronidazole antibiotic sometimes used clear infections clear infected leg ulcers pressure sorestinidazole antibiotic sometimes used treat many infections metronidazole well help clear bacteria called Helicobacter pylori H pylori gu"	High	20366316.3
"The server could connect client verify domain Fetching Timeout connect server may slow overloaded http01 urnietfparamsacmeerrorconnection The server could connect client verify domain Fetching Timeout connect server may slow overloaded SkippingAll renewal attempts failed The following certs could renewed etcletsencrypt failure"	High	28464768.63
"Yeah thats okay"	Low	130.13
"Just today though"	Low	135.21

Figure 4: Examples of messages with low and high reconstruction errors

though, the results seem to have results significantly better than guessing, and although the model doesn't recall all hypomanic / depressive episodes, the precision is better than average.

6 Conclusion/Future Work

In summary, my model successfully identified several days in which I had hypomanic or depressive episodes. The LSTM autoencoder for anomaly detection seems to work, with a certain set of hyperparameters as outlined in the previous sections. I did not have time to write any other algorithms, but if time permitted, I would have constructed a k-Means clustering baseline on the frequencies of text messages as an unsupervised learning algorithm baseline. I would have also tried to include frequencies as text messages as a feature in the deep learning model if I had more time.

Another key finding I took away from this project was that hypomanic and depressive episode text messages are very different from each other, although both are anomalies in the whole dataset. It might be worthwhile training separate algorithms for identifying both hypomanic and depressive episodes.

All in all, I learned a lot from the project, and thank you to the CS230 staff for guidance.

Code: https://github.com/shreyashankar/cs230-final.

7 Contributions

Shreya Shankar (I) did the entire project. Acknowledgements to project TA Conner Smith for feedback throughout the project.

References

[1] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey, 01 2019.

[2] Daniel Durstewitz, Georgia Koppe, and Andreas Meyer-Lindenberg. Deep neural networks in psychiatry. *Molecular Psychiatry*, 24:1, 02 2019.

[3] Nkiruka Emeagwali, Rahn Bailey, and Fatima Azim. Textmania: Text messaging during the manic phase of bipolar i disorder. *Journal of health care for the poor and underserved*, 23:519–22, 05 2012.

[4] Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP ’02, pages 63–70, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[5] Chandresh Maurya, Durga Toshniwal, and Gopalan Venkoparao. Online anomaly detection via class-imbalance learning. pages 30–35, 08 2015.

[6] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[7] Adrian Shatte, Delyse Hutchinson, and Samantha Teague. Machine learning in mental health: A scoping review of methods and applications. *Psychological Medicine*, 49:1–23, 02 2019.

[8] John Zulueta, Andrea Piscitello, Mladen Rasic, Rebecca Easter, Pallavi Babu, Scott Langenecker, Melvin McInnis, Olusola Ajilore, Pete Nelson, Kelly Ryan, and Alex Leow. Predicting mood disturbance severity with mobile phone keystroke metadata: The biaffect digital phenotyping study (preprint). *Journal of Medical Internet Research*, 20, 01 2018.