Journal of Data Analysis and Information Processing, 2014, 2, 60-66
Published Online May 2014 in SciRes. http://www.scirp.org/journal/jdaip
http://dx.doi.org/10.4236/jdaip.2014.22008

Sentiment Analysis on the Social Networks Using Stream Algorithms

Nathan Aston, Timothy Munson, Jacob Liddle, Garrett Hartshaw, Dane Livingston, Wei Hu[*]

Department of Computer Science, Houghton College, Houghton, USA
Email: [*]wei.hu@houghton.edu

Received 22 March 2014; revised 28 April 2014; accepted 16 May 2014

Abstract

The rising popularity of online social networks (OSNs), such as Twitter, Facebook, MySpace, and LinkedIn, in recent years has sparked great interest in sentiment analysis on their data. While many methods exist for identifying sentiment in OSNs such as communication pattern mining and classification based on emoticon and parts of speech, the majority of them utilize a suboptimal batch mode learning approach when analyzing a large amount of real time data. As an alternative we present a stream algorithm using Modified Balanced Winnow for sentiment analysis on OSNs. Tested on three real-world network datasets, the performance of our sentiment predictions is close to that of batch learning with the ability to detect important features dynamically for sentiment analysis in data streams. These top features reveal key words important to the analysis of sentiment.

Keywords

Modified Balanced Winnow, Sentiment Analysis, Twitter, Online Social Networks, Feature Selection, Data Streams

1. Introduction

Since the early 1990s the Internet has exploded as a means by which people communicate information in various forms. Between 1990 and 2000, it was estimated that Internet traffic was doubling every three to four months [1]. Among the most rapidly growing means of Internet communication in recent times has been the developing world of online social networks (OSNs) such as Twitter, LinkedIn, Facebook, as well as a wide array of web logging (blogging) services [2]. The astounding amount of data flowing through social networks has made min-

[*]Corresponding author.

How to cite this paper: Aston, N., Munson, T., Liddle, J., Hartshaw, G., Livingston, D. and Hu, W. (2014) Sentiment Analysis on the Social Networks Using Stream Algorithms. *Journal of Data Analysis and Information Processing*, **2**, 60-66. http://dx.doi.org/10.4236/jdaip.2014.22008

ing for useful pieces of knowledge within social networks a field of significant interest in recent times.

Due to its large volume of data flow, data mining in social networks has become a popular research field, with sentiment analysis being an area of particular interest. The users of a social network can frequently be split into distinct groups based on common interests. By distinguishing between these groups it is possible to model their overall sentiment as a representative of a larger population, using the sub-denomination in a particular OSN as a sample. Sentiment analysis inspects data presented by individuals within the larger groups and, given a sample, allows for the determination of the overall attitude or opinion of that group towards certain topics.

Related Works

Pfitzer, Garas, and Shweitzer [3] classified twitter posts as serving distinctly one of two functions. A tweet is either information creation or the pure distribution of information where a user reposts another's original idea or thought (retweet). It was found that emotional divergence has an impact on the probability of a piece of information being retweeted; tweets with higher emotional divergence have a higher probability of being retweeted. Previous work in sentiment analysis on Twitter has revealed that there is a distinct correlation of the swing of the collective mood due to the ability to retweet another user's post.

In a study by Garas, Garcia, Skowron, and Shweitzer [4] the communication patterns used across online chatrooms were analyzed for patterns of both real time information exchange and overall emotion. Models of online interaction were used to determine and analyze emotional triggers in conversation, as well as a comparison of behavior both online and in the real world. It was discovered that people did not change their pattern of expression significantly between previously studied forms of communication. Additionally, instant chatrooms do not show the same heated emotional patterns that are often found in alternate forms of online communication; instantaneous reply chats are more balanced in emotion about various topics.

The language of tweets is unique due to the 140-character limit imposed upon individual posts, causing users to often utilize shorthand notation as well as emoticons in sentiment expression [5]. Pak and Paroubek [6] have done groundbreaking work in the field of short text analysis while classifying tweets based on the correlation between emoticons and parts of speech. The emoticons were used to determine the overall sentiment of the tweet [7], as the character limit makes it more likely for there to be only a single sentiment. The remainder of the tweet was split into distinct parts of speech using the Tree Tagger algorithm, showing which part of speech has the greatest impact on a post's overall sentiment [6] [8].

Most closely related to the research done in this paper is the work of Aston, Liddle, and Hu [9] on Twitter sentiment classification in data stream. The Perceptron algorithm and its voted version with static feature selection were used to predict sentiment in a data stream environment for large amounts of Twitter data. Their streaming prediction achieved similar accuracy to that of the batch prediction of [2]. The words in the top features were uncovered.

Similarly we aim to apply the Modified Balanced Winnow algorithm to social network sentiment classification in data stream and to implement online feature selection in conjunction with this in order to account for changing data over time.

2. Materials and Methods

Three datasets were used for sentiment analysis on data streams. We also incorporated online feature selection to improve runtime and reveal the change in feature importance over time.

2.1. Data Stream

Traditional studies in sentiment analysis have primarily used batch mode learning to repeatedly traverse relatively small datasets. However, in most real world applications of algorithms designed to analyze sentiment from OSNs, the datasets are much larger and constantly changing due to the nature of online social networking. The combination of the sheer size and dynamic nature of OSNs makes batch learning an impractical solution [10].

In the case of OSN sentiment classification using a streaming algorithm is a more efficient solution. Analyzing data in a stream allows posts to be processed in real time as they appear on social networks. Running algorithms in a data stream environment means that it is only viable to perform a single pass over the data. This method results in decreased accuracy, but means that we gain a considerable improvement in runtime. For these reasons as well as the ability to handle small changes in data streams over time [10], we chose stream algorithms to predict sentiment, using Massive Online Analysis (MOA) and Weka frameworks [11].

2.2. Datasets

For the purpose of this research we utilized three public datasets: Sanders Corpus [12], STS-Gold [13], and SentiStrength [5].

2.3. Subsets of Sanders Corpus

Sanders Corpus consists of 5513 manually classified tweets. As a large number of these have become unavailable since the creation of the dataset and because certain tweets were irrelevant to our study of sentiment analysis the total number used was reduced to 3320. We were interested in two subsets of the reduced Sanders Corpus, specifically analyzing tweets for positive or negative sentiment. In the positive/negative subset all tweets with a neutral label were removed from the dataset, then the remaining tweets are analyzed for positive or negative sentiment overall.

2.4. STS_Gold

This dataset contains 2034 tweets, which are hand labeled according to their sentiment, either positive or negative. They are assigned sentiment values of either 0 or 4 based on how negative (0) or positive (4) the tweet is. This dataset annotates the tweets and entities (target subject) separately, allowing for finer sentiment of tweets containing annotated entities.

2.5. SentiStrength

This dataset is composed of 4242 posts from 5 different social media sites (MySpace, YouTube, Digg, BBC, Runners World) which have been manually labeled according to their sentiment, with a score being given for both positive and negative sentiment appearing in the post. Each post is assigned two scores, one from −1 to −5 representing how negative the post is with −5 being the most negative, and another from 1 to 5 representing how positive the post is with 5 being the most positive.

2.6. Online Feature Selection

Because we are representing the posts as n-grams, each one has 95^n features. As a result of the large number of features for each post, classification takes a great deal of time, which is problematic in a data stream environment. For this reason it is advantageous to reduce the number of features that must be considered when performing classification. In addition to time consumption, many features of each post do not play a significant role in the classification of the sentiment. Thus we performed online feature selection to discover the top features of each gram representation [14]. Unlike static feature selection methods, which find the top features once in a batch learning environment [9], we performed feature selection dynamically, with a newly calculated set of top features at each time interval of 100 sequential posts. At first we removed all but the top features when performing classification but found that this resulted in a considerable reduction in accuracy. We discovered that in order to maintain a high level of accuracy while reducing runtime it was necessary to keep a percentage of the less significant features in the classification of posts.

3. Modified Balanced Winnow

3.1. Winnow

The Winnow algorithm is a mistake driven machine learning technique that uses a promotion parameter α in order to train on pre-labeled instances [14]. The instances are labeled with boolean values and classified as either 0 or 1. Each instance is input to the learner, classified, and finally compared to the actual class label. The process of the Winnow algorithm is outlined briefly in **Winnow Algorithm** below.

Winnow Algorithm

```
For each instance $i$:
1. If prediction($i$) = true class
   a. Continue
2. If prediction($i$) = 0 and true class = 1
   a. Multiply weight matrix entries by $\alpha$
3. Else if prediction = 1 and true class = 0
   a. Multiply weight matrix entries by 0
```

3.2. Balanced Winnow

Balanced Winnow extends upon the Winnow algorithm by adding a demotion parameter β. It utilizes a promotion parameter $\alpha > 1$, a demotion parameter $0 < \beta < 1$, and a threshold θ. For each instance trained on, if the instance is predicted correctly then the training step is skipped, if it is predicted as 1 with an actual class 0 the weight matrix is multiplied by the demotion parameter, and if it is predicted as 0 with actual class 1 the weight matrix is multiplied by the promotion parameter.

3.3. Modified Balanced Winnow (MBW)

We used MBW to classify posts. Like Balanced Winnow, MBW requires a promotion parameter α and demotion parameter β. It is separated from Balanced Winnow by utilizing larger margins and a minor change in the update rules for the weight matrices [14]. MBW is outlined in detail in the figure **Modified Balanced Winnow**. We initially trained the MBW classifier over the first 100 instances. For each following instance we first classified it using MBW; if the classification was correct we continued to the next instance and updated the correct count. If it was incorrect we updated the incorrect count and updated the weight matrix.

Modified Balanced Winnow

```
1. Initialize $i = 0$, counter $c = 0$, and models $u_0$ and $v_0$
2. For $t = 1, 2, \ldots, T$:
   a. Receive new example $x_t$ and add bias.
   b. Normalize $x_t$ to 1.
   c. Calculate score = $(x_t,u_i) - (x_t,v_i) - \theta_{th}$
   d. Receive true class $y_t$
   e. If prediction was wrong:
      i. Update Models. For each feature $j$ where $x_t > 0$:

         $u_{i+1}^{j} = \begin{cases} u_i^j * \alpha * \left(1 + x_t^j\right), if\ y_t > 0 \\ u_i^j * \beta * \left(1 - x_t^j\right), if\ y_t < 0 \end{cases}$

         $v_{i+1}^{j} = \begin{cases} v_i^j * \beta * \left(1 - x_t^j\right), if\ y_t > 0 \\ v_i^j * \alpha * \left(1 + x_t^j\right), if\ y_t < 0 \end{cases}$

      ii. $i$++
   f. Else $c_i$++
```

4. Results and Analysis

Once data had been collected and a data stream simulation was set up we used the MBW algorithm in conjunction with massive online analysis (MOA) in order to train and run a classifier. We present two sets of results, one on sentiment analysis and the other on feature selection.

The Sanders corpus, SentiStrength, and STS_Gold datasets were all analyzed for sentiment. We achieved the highest accuracy on STS_Gold, with the lowest accuracy on the Sanders corpus.

We performed dynamic online feature selection on each of the utilized datasets. For each feature we calculated the importance score as $I_t = |u_t - v_t|$ for a time stamp t, as outlined by Carvalho *et al.* [14]. After computing the score of each feature we selected the top 20 features to use in classification, thus increasing the efficiency of the algorithm.

4.1. Sentiment Based on MBW

MBW requires four different user defined parameters (promotion, demotion, feature selection, good feature selection), which may lead to a decrease in accuracy when chosen randomly. Therefore we performed extensive checks over these parameters to reveal possible good ranges for each. It was determined that the larger the gram size used, the more accurate our prediction became, since larger gram sizes reveal more of a word than smaller gram sizes. Also, when using feature selection and good feature selection, we noticed that accuracy was greatest with the more features and good features used, as shown in **Table 1**. We divided our datasets into sequential segments of size 100 to reveal the importance of features over time. We then ran MBW on each segment of 100 as a stream.

4.2. Sanders Corpus

Our division created 22 segments (or timestamps) for Sanders. **Figure 1** illustrates the feature selection of the top 20 features over the 22 timestamps of Sanders 5 gram representation. As the figure illustrates, the top features oscillate in importance in the beginning, but as the data is continually fed in, the few top features begin to stabilize and retain their higher position. Features that are subsets of the words "Google", "Apple", and "Android" are some of these features that retain a higher importance over time. For all gram sizes of Sanders, high accuracy predictions fell within the range of promotion and demotions values of 0.9 and 1.1.

A high accuracy is seen when α and β values increase together from around {1, 1} onward. By incorporating dynamic feature selection in our MBW we achieved an accuracy of 73.3% while [9] achieved an accuracy of 77% using 5 grams with manual feature selection. In terms of a data stream, it is important to perform dynamic feature selection due to the changing importance of features with new incoming data.

The work of [9] performed feature selection on a static batch and then performed sentiment analysis on this reduced feature set. This approach has downfalls as unknown data coming in over time will change the importance of features.

Table 1. Sentiment prediction accuracy of Sanders 5 grams check over percent of features to use (rows) to percent of good features to use (columns). Color ranges with accuracy low to high (red to green).

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.1	58.3	61.6	64.2	65.4	66.6	67.4	66.9	68.8	69.8	71.8
0.2	58.9	61.7	64.4	65.4	67.1	66.7	68.4	68.6	69.8	72.6
0.3	60.2	62.9	64.4	66.5	67.5	67.7	68.9	68.4	69.4	71.9
0.4	59.6	62.7	64	66.1	66.8	67.6	68.8	69.8	70.3	73.3
0.5	60.9	62.5	65.1	66.1	66.9	67.9	68.4	70.7	69.3	71.7
0.6	61.2	63.6	65	64.5	67	68.1	69.9	68	68.8	72.6
0.7	61.4	63.2	64.3	65.3	68	69.1	67.3	69	69.9	72.2
0.8	63.4	65.2	64.5	65.5	67.7	66.3	68.4	69.2	69.3	72.6
0.9	64.8	66.5	66.7	65.4	68.2	65.6	67.8	68.3	69.5	72
1	67.4	67.4	67.4	67.4	67.4	67.4	67.4	67.4	67.4	67.4

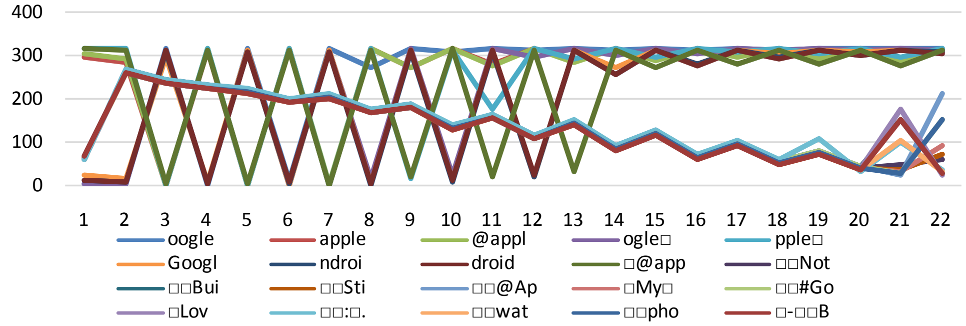

Figure 1. Sanders 5 grams top 20 features as they change over time. X axis denotes the timestamps and the Y axis denotes the importance of each feature. Note: box is a space character.

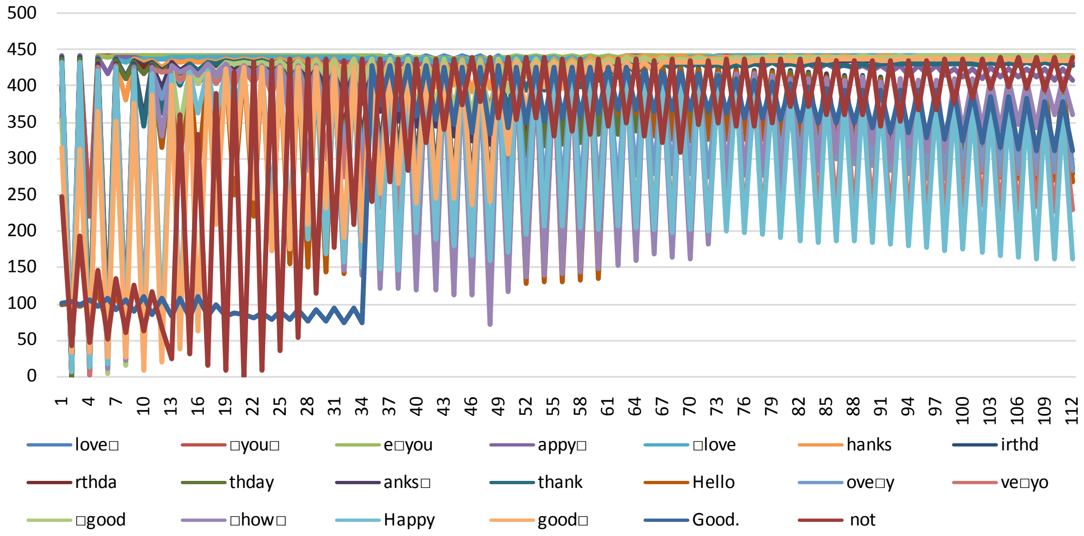

Figure 2. Sentiment Strength 5 grams top 20 features as they change over time. X axis denotes the timestamps and the Y axis denotes the importance of each feature. Note: box is a space character.

4.3. SentiStrength and STS-Gold

The SentiStrength dataset was divided into 112 sets of size 100 each. We achieved an accuracy of 73.6% using promotion and demotion values of 0.7 and 1.4 respectively. Features such as "love", "you", "happy", "thank", and "good" contain the highest importance through this dataset, shown in **Figure 2** of 5 grams representation. The STS_Gold dataset was divided into 42 sets, with the highest accuracy of 87.5% for promotion and demotion values of 2 and 1 respectively.

The promotion (α) and demotion (β) table of SentiStrength 3 grams representation revealed that values of α and β which optimize the accuracy of prediction seem to follow a logarithmic pattern from α in the range 1 - 3 and β in the range 0.2 - 0.9. In the STS_Gold dataset high accuracy is seen when α is greater than 1 but less than 3 and β is less than α.

In terms of accuracy with feature selection, it is best to use all features. In Sanders and STS_Gold, the difference in accuracy from using all features to using 10% of the features is around 10%, while with SentiStrength, the difference is only around 2% to 3%.

5. Conclusion

We discovered that the values of promotion and demotion and the feature selection contribute independently to the sentiment prediction accuracy. The varying range of accuracy resulting from differing percentages of selected features makes it difficult to determine what a good feature selection percentage should be without a significant decrease in accuracy. Our MBW achieved the highest accuracy of 87.5% with STS_Gold on 5 grams representation and only 73.3% on 3 grams and 73.6% on 5 grams for Sanders and Sentiment Strength respectively. Our results for Sanders are close to the accuracy of [9] for their 5 grams on Sanders. MBW is capable of performing feature selection dynamically on a date stream, instead of performing a batch feature selection beforehand, such as [9] does. The top features that we found for each dataset revealed the nature of these datasets. For example, Sanders, which was a specified dataset for technology organization sentiment, contains features of Apple, Google, and Android. Sentiment Strength and STS_Gold are datasets of social media posts from various sites, and the features we observed from these datasets involve words such as "love", "happy", "thank", and "good", which are sentimental words that would influence a post's sentiment. Our streaming algorithm MBW yields a minor decrease in accuracy compared to its batch counterpart [2], but in a real world scenario, it is more realistic to analyze data that is continually coming from new social media posts.

Acknowledgements

We thank Houghton College for its financial support.

References

[1] Coffman, K.G. and Odlyzko, A.M. (2002) Growth of the Internet. In: Kaminow, I.P. and Li, T., Eds., *Optical Fiber Telecommunications IV B*: *Systems and Impairments*, Academic Press, San Diego, 17-56. http://dx.doi.org/10.1016/B978-012395173-1/50002-5

[2] Deitrick, W. and Hu, W. (2013) Mutually Enhancing Community Detection and Sentiment Analysis on Twitter Networks. *Journal of Data Analysis and Information Processing*, **1**, 19-29. http://dx.doi.org/10.4236/jdaip.2013.13004

[3] Pfitzner, R., Garas, A. and Schweitzer, F. (2012) Emotional Divergence Influences Information Spreading in Twitter. ICWSM-12.

[4] Garas, A., Garcia, D., Skowron, M. and Schweitzer, F. (2012) Emotional Persistence in Online Chatting Communities. *Scientific Reports*, **2**, Article Number: 402. http://dx.doi.org/10.1038/srep00402

[5] Thelwall, M., *et al*. (2010) Sentiment Strength Detection in Short Informal Text. *Journal of the American Society for Information Science and Technology*, **61**, 2544-2558. http://dx.doi.org/10.1002/asi.21416

[6] Pak, Alexander, and Patrick Paroubek (2010) Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *LREC* 2010.

[7] Go, A., Bhayani, R. and Huang, L. (2009) Twitter Sentiment Classification Using Distant Supervision. *CS*224*N Project Report*, 1-12.

[8] Schmid, H. (1994) Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of International Conference on New Methods in Language Processing*, Vol. 12.

[9] Aston, N., Liddle, J. and Hu, W. (2014) Twitter Sentiment in Date Streams with Perceptron. *Journal of Computer and Communications*, **2**, 11-16. http://dx.doi.org/10.4236/jcc.2014.23002

[10] João, G. (2010) Knowledge Discovery from Date Streams. Taylor & Francis, Boca Raton.

[11] Bifet, Albert, and Eibe Frank. (2010) Sentiment Knowledge Discovery in Twitter Streaming Data. *Proceedings of* 13*th International Conference on Discovery Science*, Canberra, 6-8 October 2010, 1-15.

[12] Sanders, Niek. Twitter Sentiment Corpus. Sanders Analytics. http://www.sananalytics.com/lab/twitter-sentiment/

[13] Saif, Hassan. Tweenator. http://www.tweenator.com/index.php?page_id=13

[14] Carvalho, V.R. and William, W.C. (2006) Single-Pass Online Learning: Performance, Voting Schemes and Online Feature Selection. *Proceedings of the* 12*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Philadelphia, 20-23 August 2006, 548-553.