Brian Mooney & Tri Nguyen

Professor Liu Xiaobai

CS 596 Machine Learning

17 December 2015

Sentiment Analysis of Live Tweets using Support Vector Machines

Abstract:

The purpose of this project was to determine the positive or negative sentiment attached to each twitter tweet. By training a Linear SVM model to categorize sentences based on sentiment we have developed a method to categorize incoming live twitter feeds that are of interest. The API searches for incoming tweets that include a user selected topic or keyword using the hashtag system to run relevant tweets through the trained Linear SVM. The SVM then determines the sentiment attached to the tweet by labeling each tweet with a binary 1 (positive) or 0 (negative) label.

Introduction:

Having the ability to autonomously determine the sentimental value attached to tweets can be a valuable tool with a broad spectrum of uses. The API allows the user to select topics of their choosing, allowing only tweets relevant to the current topic to be judged for it’s sentiment. The project thus enables us to determine the general public’s sentiment on specific topics relevant to today’s society. We can use the Linear SVM to determine the sentiment behind public statements made by a community of roughly 1 Billion twitter users worldwide. Having the ability to determine the public’s sentiment on a specific topic can be an invaluable tool for marketing or other sociologically related research.

Task Description:

The process of determining the general public’s sentiment on a specific topic using tweets begins with running each sample (tweet) through a classification technique to determine it’s individual sentiment. Training the classifier to accurately determine the sentiment of individual samples becomes the main task in determining public sentiment. After a classifier has been adequately trained a method must be developed to pull live tweets based on a search keyword as testing samples. These testing samples must also be parsed to fit the dimension of the training samples. Once the testing samples are parsed they may be classified by the SVM in order to determine the sentiment behind the individual sample.

The work was split in the following way:

Brian:

SVM implementation coding

Documentation and Powerpoint

Tri:

NN implementation and data parsing implementation coding

Live Twitter Demonstration

Major Challenges and solutions:

Developing the correct method to parse the training data becomes paramount in developing an accurate classifier. Once this method is determined a similar method must be used on testing samples to match the dimensionality of the training set. In this project the Bag of Words method was used to create a matrix of the training set of size W x S where W is the total number of unique words in the training set and S is the number of samples (sentences) in the training set. This created a training matrix where the dimensionality of the data was equal to the number of unique words the SVM was designed to learn. Each sample had a corresponding label provided by the training set.

Many factors led to an incredibly long training time of the SVM. The large dimensionality of both training and testing samples resulted in an incredibly long training time of the SVM, and as a solution we performed PCA on both the training and testing samples. The use of javascript as an implementing language also lengthened the computational time so steps were taken to shorten the training set at the cost of testing accuracy.

javascript is also an asynchronous language so implementing this project in javascript proved very challenging. The choice to go with javascript was made due to the ease of implementing the Twitter API in the language. The coding challenges were met, however, at the cost of computational capability. Although the training set had to be reduced in size, relevant results were still achieved through the Linear SVM model.

Experiments:

1. Dataset description:

The training set used for training our SVM was the labeled IMDB sentence sentiment data set. The data set was parsed using the Bag of Words method into a matrix and each training sample had a corresponding label in the label array.

Training Matrix: W x S

Label Array: 1 x S

W = Total # of Unique words identified in the Training data set (Dimensionality of the SVM)

S = # of Samples used to train the SVM

The training set was first parsed to drop words and punctuation that were not considered to be identifiers, such as articles preceding nouns. Once each sample was parsed into words deemed identifiers each sample was converted into the Matrix by counting the number of times an identifier occurred in the sentence. A value N in the matrix meant a word represented by that row of the matrix was used N times in the specific sample sentence in the training set determined by the column of the matrix.

Given 2 sentences:

Sample A: “I love cheese” Label: 1

Sample B: “I hate class” Label: 0

The corresponding matrix would be:

|  |  |  |
| --- | --- | --- |
|  | A | B |
| I | 1 | 1 |
| Love | 1 | 0 |
| Cheese | 1 | 0 |
| Hate | 0 | 1 |
| Class | 0 | 1 |

|  |  |  |
| --- | --- | --- |
| Label Vector: | 1 | 0 |

1. Evaluation Metrics:

Using the Matrix of the Training Set described above a Linear SVM was trained after PCA was used to reduce the dimensionality of the data. The parameters chosen for the SVM were determined by experimentation after determining parameters that would lead to the most accurate results.

kernelType: 'LINEAR'

kFold: 2,

normalize: true,

reduce: false,

retainedVariance: 0.99,

eps: 1e-3,

cacheSize: 2000,

shrinking : true,

probability : false

c: [3],

normalize: true,

reduce: false,

retainedVariance: 0.99,

eps: 1e-3,

cacheSize: 2000,

shrinking : true,

probability : false

c: [3],

normalize: true,

reduce: false,

retainedVariance: 0.99,

cacheSize: 2000,

shrinking : true,

probability : false

After determining the SVM that would train the most accurately given our training data set we can begin to run testing samples through our trained model.

1. Major results

Incoming Live tweets are parsed to match the format of the training set used. The tweets are parsed similar to the training data in which punctuation and non-identifier words are dropped. A W – dimension vector is 0 initialized and each word in the tweet is counted to create the testing vector. For every word encountered in the tweet the value in the corresponding row for that word is incremented. Because of time constraints if a word in a tweet is not one of the unique words encountered and learned from the training set it is simply ignored. This means that tweets are analyzed only for words recognized as words that have been learned from the training set; a better training set containing a higher variety of words would result in better accuracy, but perhaps lose it’s robustness as overfitting may become an issue.

Here are some results from the Live Twitter sentiment analysis:

Key word to find sentiment:

obama

===========================================

\*\*\*\*\*\*\*\*\*\*\*\*\*\*SVM Data Report\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Positives: 111

Negatives: 389

{ accuracy: 0.6247464503042597,

  fscore: 0.5144356955380578,

  recall: 0.4279475982532751,

  precision: 0.6158357771260997,

  class:

   { '0':

      { precision: 0.6158357771260997,

        recall: 0.7954545454545454,

        fscore: 0.6942148760330579,

        size: 264 },

     '1':

      { precision: 0.6447368421052632,

        recall: 0.4279475982532751,

        fscore: 0.5144356955380578,

        size: 229 } },

  size: 493,

  reduce: false,

  retainedVariance: 1,

  retainedDimension: 2338,

  initialDimension: 2338 }

===========================================

Negative sentiment.

1. Analysis

Due to time constraints and the interpreted and asynchronous nature of javascript the training sample we used was only a portion of the labeled set provided by imdb. In the case that a larger training set was used prediction accuracy would probably be higher as long as time was not a constraint. javascript was used due to the convenience of it’s twitter API, which was the focus of our research as our goal for this project was to demonstrate Machine Learning techniques on Live Twitter feeds.

The results achieved from analyzing the sentiment in these tweets coupled with our implementation of the API which allows users to seek the sentiment revolving only around specific topics is very useful. The tool can be used to judge the general consensus twitter users have on any topic from “Love” to “Star Wars” to “ISIS”.

Conclusion and Future Works:

Due to time constraints we could not finish our implementation of the RBF Kernel, Logistic Regression, and FF-BP Neural Network classifiers that we were hoping to also implement. All of these classifiers are almost implemented in the current state of the program. There is a strong chance better results can be achieved by implementing these different classifiers and bagging the results in a weighted manner.