Extracting emotion from twitter

John S. Lewin **Alexis Pribula**

INTRODUCTION

Problem Being Investigate

Our projects consider *tweets* or status updates of www.twitter.com users. Based on the information contained in a tweet, we want our system to be able to answer the following question: does the author of the tweet feel happy, sad or neither about a particular word of her tweet. For example, if the tweet is: *my boyfriend is amazing*, and the word of interest is *boyfriend*, the goal of the system is try to tell us that the person who wrote the tweet feels positive about her boyfriend. Moreover, in cases where the emotion of the tweet is different from the emotion related to the word of interest, we want to try to predict the emotion of the word of interest. For example, for the tweet: *w00t!!!!! Last day of school!!!!!! 9th grade over and 10th grade comeing*, where the word of interest is *school*, we want the system to tell us that the user feels unhappy about school.

Data

No database describing positive, negative and neutral feelings relating to specific words in a tweet currently exists. We have hence created a script which downloads the result of twitter keyword searches (c.f. search.twitter.com). We ran the script a few times and saved the search results in an SQLite database. After that we scored each tweet individually by hand. We have managed to accumulated 540 scored tweets.

Implementation

We have implementations in Python (where the twitter parser lives), Java, and analysis in Matlab. For the java implementation, we decided to use the Maximum Entropy classifier code from our NLP homework 3 and some code from the Stanford NLP Parser. In a nutshell, we tried to first increase the precision of prediction as much as possible using MaxEnt alone and then we the parser to increase further.

CLASSIFIERS

Maximum Entropy I

We first created a ScoredTweet class which, among other things, has an ArrayList containing each word of a tweet and a score equal to $+1$ for positive feeling, -1 for negative feeling, 0 for neutral feeling. Then, we created a transformTweetData method, which converts this to a BasicLabeledDatum¡String,Double¿ datastructure, which is compatible with the course provided MaxEnt algorithm.

The most significant part of the work consisted in finding relevant features. We measured importance of tweets both looking at the prediction accuracy and also by looking at the sum of the absolute values of MaxEnt weights for each label. Here are some of the ones which were the most useful to us (a part from of course the feature marking the search word itself):

- One of the following smileys is anywhere in the tweet: `:)` `:-)` `:~)` `;)` `;-)` `:D` `:-D` `:>` `:->` `:P` `XD` `^5` `<3` `lol` `lmao`. The most relevant ones seemed to be `lol` and `<3` which decreased error by about 1% each when added to this feature. Sum of absolute MaxEnt weights: ~7.77
- Previous word ends with *ing*. Sum of absolute MaxEnt weights: ~6.63
- Next
- An inverse meaning feature which is activated if the word just before the word of interest is one of the following: *anti*, *no*, *sans* or *except*. Sum of absolute MaxEnt weights: ~5.13
- Either the word just before the word of interest or the word before that is one of the follwoing: *into*, *love*, *loving*, *miss*, *missing*, *long*, *longing*, *beautiful*, *great*, *fantastic*, *awesome*, *beautiful*, *cool*, *nice*. Sum of absolute MaxEnt weights: ~3.13
- Taking the word 2 words before the word of interest and marking it as such. Taking the the words after the word of interest and marking as such. Even though it is hard to tell how much these features

contribute exactly since every one of them weighs little, these features increased accuracy by ˜3%.

Using only MaxEnt, we reached a classification rate of 47.3%.

MaxEnt with Stanford Parser

We used the Stanford Parser in order to get grammatical relationships on the word of interest in the tweet. This data helped a lot because it is general and relevant to classification. The grammatical relationships which included the word of interest and which were the most relevant were:

- MODIFIER with a sum of absolute MaxEnt weights of ˜119.33
- NN with with a sum of absolute MaxEnt weights of ˜19.48
- AMOD_DEP with a sum of absolute MaxEnt weights of ˜12.18
- POSS_DEP with a sum of absolute MaxEnt weights of ˜8.14
- CONJ_DEP with a sum of absolute MaxEnt weights of ˜7.31

Further we decided that since MODIFIER accounted for so much of the weights, we should associate the word in the grammatical relationship with the word of interest. This increased precision by about 4%.

At the end, using a combination of both MaxEnt and the Stanford Parser, we reached an accuracy of 47.6% on crossvalidated data.

Maximum Entropy II

Let + denote a positive emotional response, − a negative emotional response, and 0 a neutral resonse. For a given set of n features $f_1, \cdots, f_n$, a central component to the classifier is calculating $P(\cdot, f_i = 1)$ and $P(\cdot, f_i = 0)$, where $\cdot \in \{+, -, 0\}$. We will first consider the case where we have extracted neutral responses. Define A to be a matrix such that

$$A_{ij} = \begin{cases} 1 & \text{if tweet } i \text{ contains features } j \\ 0 & \text{otherwise} \end{cases}$$

Let b_i be a vector of labeled classifications. Furthermore, let $b^{(+)}$ be the components of b that are labeled as +, $b^{(-)}$ the components of b labeled as −, and $A^{(+)}$ and $A^{(-)}$ the coresponding rows of A.

Finally, define

$$x^{(+)} = \begin{bmatrix} P(+, f_1) \\ P(+, f_2) \\ \vdots \\ P(+, f_n) \end{bmatrix} \quad x^{(-)} = \begin{bmatrix} P(-, f_1) \\ P(+, f_2) \\ \vdots \\ P(-, f_n) \end{bmatrix}$$

With $\mathbf{x} = [x^{(+)}\ x^{(-)}]$.

For each $b_j = +$, we require $\mathbf{x}$ such that

$$\sum_i \mathbf{1}\{\text{feature } i \text{ is present}\} P(+, f_i)$$

$$> \sum_i \mathbf{1}\{\text{feature } i \text{ is present}\} P(-, f_i)$$

Where we are using the notation $\mathbf{1}\{\cdot\}$ to be 1 if $\cdot$ is true and 0 otherwise. This is equivalent to

$$A^{(+)} x^{(+)} > A^{(+)} x^{(-)}$$

Similarly, we require $\mathbf{x}$ such that

$$A^{(-)} x^{(-)} > A^{(-)} x^{(+)}$$

Subject to the constraints

$$\mathbf{x}_i \geq 0 \quad , \quad x_i^{(+)} + x_i^{(-)} = p^f \tag{1}$$

Where p_i^f is the probablity of feature i. The final equality in 1 follows from the fact that

$$P(+, f_j = 1) + P(-, f_j = 1) + P(+, f_j = 0) + P(-, f_j = 0)$$

$$= P(f_j = 0) + P(f_j = 1) = 1$$

We know $P(f_j = 1)$ from the data, and we form

$$P(+, f_j = 1) + P(-, f_j = 1) = P(f_j = 1)$$

Which gives us (1). We apply the principle that the best fit to the data is the solution that maximizes the entropy between the variables. Putting this all together

we arrive at the following optimization problem:

$$\begin{aligned} \text{maximize} \quad & -\textstyle\sum_i \mathbf{x}_i \log(\mathbf{x}_i) \\ \text{subject to} \quad & \begin{bmatrix} A^{(+)} & -A^{(+)} \end{bmatrix} \begin{bmatrix} x^{(+)} \\ x^{(-)} \end{bmatrix} > 0 \\ & \begin{bmatrix} A^{(-)} & -A^{(-)} \end{bmatrix} \begin{bmatrix} x^{(+)} \\ x^{(-)} \end{bmatrix} < 0 \\ & \begin{bmatrix} I & I \end{bmatrix} \begin{bmatrix} x^{(+)} \\ x^{(-)} \end{bmatrix} = 1 \end{aligned} \tag{2}$$

The objective in (2) is convex and the constraints are all affine, so we can find a unique solution (if one exists). The formulation in (2) is readily extended to other classes. For example, allowing neutral labels we have a three way maximum entropy classifier given as

$$\begin{aligned} \text{maximize} \quad & -\textstyle\sum_i \mathbf{x}_i \log(\mathbf{x}_i) \\ \text{subject to} \quad & A^{(+)}x^{(+)} > A^{(+)}x^{(\#)} \\ & A^{(+)}x^{(+)} > A^{(+)}x^{(-)} \\ & A^{(-)}x^{(-)} > A^{(-)}x^{(+)} \\ & A^{(-)}x^{(-)} > A^{(-)}x^{(\#)} \\ & A^{(\#)}x^{(\#)} > A^{(\#)}x^{(+)} \\ & A^{(\#)}x^{(\#)} > A^{(\#)}x^{(-)} \\ & \begin{bmatrix} I & I & I \end{bmatrix} \begin{bmatrix} x^{(+)} \\ x^{(-)} \\ x^{(\#)} \end{bmatrix} = p^f \end{aligned} \tag{3}$$

Where we use the # symbol to denote neutral and define the probabilities $x^{(\#)}$ and matrix $A^{(\#)}$ analagously to the events + and −. In some cases equation (3) was not feasible. To get around this we relax the poblem to

$$\begin{aligned} \text{maximize} \quad & -\textstyle\sum_i \mathbf{x}_i \log(\mathbf{x}_i) \\ \text{subject to} \quad & A^{(+)}x^{(+)} > A^{(+)}x^{(\#)} \\ & A^{(+)}x^{(+)} > A^{(+)}x^{(-)} \\ & A^{(-)}x^{(-)} > A^{(-)}x^{(+)} \\ & A^{(-)}x^{(-)} > A^{(-)}x^{(\#)} \\ & A^{(\#)}x^{(\#)} > A^{(\#)}x^{(+)} \\ & A^{(\#)}x^{(\#)} > A^{(\#)}x^{(-)} \\ & \left| \begin{bmatrix} I & I & I \end{bmatrix} \begin{bmatrix} x^{(+)} \\ x^{(-)} \\ x^{(\#)} \end{bmatrix} - p^f \right| \leq \epsilon \end{aligned} \tag{4}$$

And solve (4) for the smallest ϵ possible (I used $\epsilon = .01$).

Maximum Liklihood over a Guassian Distribution

We consider a straightforward implementation of Maximum Liklihood with the assumption that the distribution of words is gaussian,and solve

$$\text{maximize} \sum_{i=1}^{m} \log p_x(y_i - a_i^T x) \tag{5}$$

Where $p_x(z)$ is the normal distribution.

Data	Description
3-Way	Compares neutral, positive, and negative emotions in a three way classifier
$\pm \cdot \#$	Compares emotion present versus no emotion (2-way)
$+ \cdot -$	Compares positive emotion to negative emotion (2-way)

Table 1: Data Sets used in Classification

Support Vector Machine

We implement a SVM solution on the problem. In particular, we find the minimum margin seperating two data sets by solving

$$\begin{aligned} &\text{minimize } ||\alpha||_2 \\ &\text{subject to } \alpha^T a_i - b \geq 1 \quad i = 1, \cdots, n \\ &\qquad\qquad \alpha^T a_i - b \leq -1, \quad i = 1, \cdots, n \end{aligned} \tag{6}$$

Assuming the data is seperable. In our case the data is *not* seperable, so we solve the associated problem

$$\begin{aligned} &\text{minimize } ||\alpha||_2 + k \\ &\text{subject to } \alpha^T a_i - b \geq 1 \quad i = (1-k), \cdots, n \\ &\qquad\qquad \alpha^T a_i - b \leq -(1-k), \quad i = 1, \cdots, n \end{aligned} \tag{7}$$

DATA

In the following sections we will reference several data sets using the notation in table 1. Our data is selected from the Twitter API and labeled as having one of the emotions denoted as $\{+, -, \#\}$ relative to the search terms. We were able to hand-label about 1000 tweets from a data set of over 10,000.

FEATURE SELECTION

Our goal feature selection is to reduce the total required computation by limiting the feature set to only those features that have an impact on decisions, and to allow us to find combinations of useful features (otherwise too many combinations are possible). We limit feature selection to positive versus negative features and exclude neutral features.

Our approach maximizes the slab in $\mathbb{R}^n$ seperating the two data sets[1]. We assume that our data is strictly seperable (otherwise we would choose to keep the features that prevent it from being so and remove them from consideration in feature seperation). Therefore there is a hyperplane such such that for some α and β,

$$\alpha^T x^{(i)} - \beta \geq 1, \quad \alpha^T y^{(i)} - \beta \leq -1$$

I.E., there is a slab $S = \{z || \alpha^T z - \beta| \leq 1\}$ with thickness $2/||\alpha||_2$. The maximum slab can be found by

$$\begin{aligned} &\text{minimize } ||\alpha||_2 \\ &\text{subject to } \alpha^T x^{(i)} - \beta \geq 1 \\ &\qquad\qquad \alpha^T y^{(i)} - \beta \leq -1 \end{aligned} \tag{8}$$

We want to find α and β that find the maximum slab while also being very sparse in the features. Here, removing the effects of the feature will have litle effect on the final judgment. However, maximizing (8) over α and β is hard. Rather, we use the relaxation

$$\begin{aligned} &\text{minimize} && ||\alpha||_2 + \lambda||\alpha||_1 \\ &\text{subject to} && \alpha^T x^{(i)} - b \geq 1 \\ &&& \alpha^T y^{(i)} - b \leq= -1 \end{aligned} \tag{9}$$

The $\lambda||\alpha||_1$ has the effect of penalizing cardinality and pushing the minimization toward sparse vectors. We solve (9) for many values of λ and choose the maximum slab with an acceptable number of features.

Results of Feature Selection

Our training data was too small to (around 1000 tweets) get much density in bigram and trigram counts, so our features were selected from the words in the tweet corpus and some custom tweets (see appendix one). Against the training data, see figure 1. For each point, if $\bar{f}_i^c$ is the frequency that feature i occurs in class $c \in \{+, -, \#\}$, then

$$\text{red}_i = \frac{f_i^+}{f_i^-}, \text{blue}_i = \frac{f_i^-}{f_i^\#}, \text{green}_i = \frac{f_i^+}{f_i^\#}$$

Figure 1 is instructive. While there is some variation in differentiating the $+$ and $-$ classes from the $\#$ class (green and blue points, respectively), the relative freqeuency of the $+$ to $-$ features are very close (the red line marked with $+$'s). We expect that it will be much easier to differentitate emotional versus non-emotional than to differentiate Finding the optimal features very much depened on the data set. Figure 1 suggests it will be much easier to find the valuable features in comparing $\pm \cdot \#$ than it will in comparing the 3-way data or $+ \cdot -$, which turns out to be true. See figure

[1]Based on an algorithm by Steven Boyd [1]

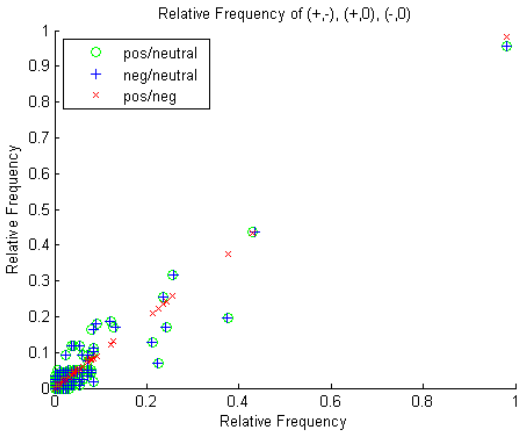

Figure 1: Relative Frequencies

2. The uppermost graph compares the margin (in the SVM sense) to the number of used features in the $\pm \cdot +$ data set. There is a sharp increase in the margin using around 50 features. From this we are able to identify the top 50 features and classify on using these. This is in stark contrast to the second graph, where the margin increases considerably with each removed feature. Against the $+ \cdot -$ data we need to use all the features.

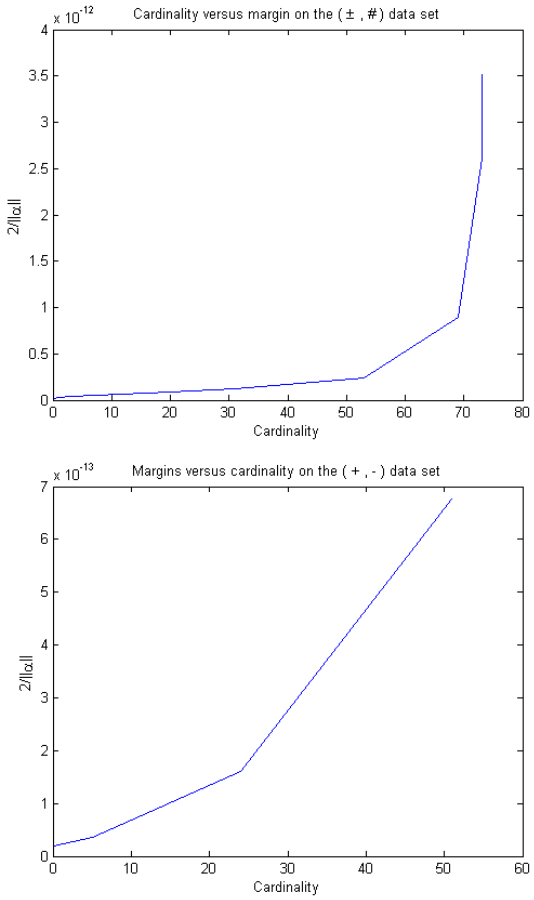

Figure 2: Features selection. lower $2/||\alpha||_2$ values represent a better classifier

RESULTS

Maximum Entropy Classifiers and Maximum Liklihood

See table **??** for classifier results against a number of algorithms.

We find our classifiers capable of identifiying emotion versus neutral text in twitter over 70% of the time, and selecting between positive, negative and neutral emotions approximately 50% of the time. There is considerable room for improvement. The section on feature ientification cooroberates what the data in table **??** suggests – we do not yet have enough data or the right features to differentiate between positive and negative emotions. This is not surprising – with only 500-1000 labeled tweets, there is not much for our classifiers to choose from.

That said, 50% on 500 labeled tweets is room for optimism that on a larger corpus we will have excellent performance. The next step is two fold: to generate a

Algorithm	Data Set	Accuracy	σ_c	% above guessing
MaxEnt I	3-way	47.60%	–	14%
MaxEnt II	3-way	41.09%	5.9%	7.76%
MaxEnt II	$\pm \cdot \#$	72.79%	5.1%	22.79%
MaxEnt II	$+ \cdot -$	51.08%	4.5%	1.08%
MaxL N	$+ \cdot -$	55.91%	7.2%	4.91%
SVM*	$\pm \cdot \#$	70.01%	6.34%	20.6%

Table 2: Results of various algorithms relative to baseline

larger corpus, and imporve our