STAT 215B Final Project, Spring 2012

Christine Kuang, Siqi Wu, and Angie Zhu

May 6, 2012

1 Introduction

Due to the restriction on overseas websites such as Facebook or Twitter, domestic substitutes have become the major platforms for the internet citizens to express their opinions towards various social or political issues. Studying posts on those website thus provides interesting insights into the public opinion. For example, some events in the recent years, such as the tragic Wenzhou train collision on July 23 2011 and the more recent Wang Lijun incident, have split the public into two major groups among which one considers the government's way of dealing with those incidences is good and one does not. In principle, we can draw text data from those websites identifying whether a particular post is related to the event of interest, and which group the post should be classified into.

Sina Weibo 新浪微博 is the largest microblogging website and one of the most popular social network website in China. It had more than 300 million registered users as of February 2012 ([**?**]) and accounted for 65% of China's microblog market by pageviews as of December 2011([**?**]). In this project, we develop a framework for for studying public opinions using Sina Weibo as a corpus for a given topic. Posts are sampled from Weibo and then processed taking the characteristics of both Chinese language and Weibo posts into consideration.

related works here

2 Methods

2.1 Data Collection

2.2 Tagging

In all, we tagged a total of 4000 Weibo posts. We made a search for the topic we were looking into and did this over time to get a total of 10,000 Weibo posts, from which we picked out 3000 posts as our training set. We each tagged 1000 posts. The final 1000 posts were for our test set and we tagged these together. We had a total of four different categories that we tagged the posts as - neutral, positive, negative, and irrelevant (spam).

As we each tagged the posts, we encountered and realized a few of the limitations involved. One limitation was in the fact that tagging these posts produced subjective responses. What one of us read as a negative response to the topic we chose, another may have read as a positive response. For example, the English phrase 'that wasn't too bad' could be taken as positive or negative. Positive - the experience was better than expected; negative - the experience wasn't great. Hence, it is difficult sometimes to truly know whether the original author of the post had in mind a negative or positive reaction to the topic he was posting about.

Another limitation was that some posts we were not sure how to tag. Many posts consisted of just a quote by the author we were looking into. These could have been seen as positive posts since the writer may have liked the quote and so posted it. But at the same time, the author could have been neutral and was just merely using the quote to apply to a specific circumstance in his life at the time. We were not entirely sure of how to tag each of the posts that fell into this category, and so again, the subjectiveness of tagging the posts by hand comes into play as a limitation in the forming of our model. Another uncertainty that occurred in trying to manually tag the posts was that some of the posts didn't talk about our chosen topic specifically, but a related topic. With our chosen topic, people who had negative responses towards 韩寒 (Hanhan), were usually on the side of the opposing author who was discrediting him. Some of the posts did not directly mention 韩寒, but would instead show support for the opposing side. With these posts, we typically labeled as a negative response. However, an argument could be made for just throwing out those posts since they do not directly say anything about 韩寒, and they could possibly just be supporting the opposing author in his own literary works and not necessarily in his stance against 韩寒. Another uncertainty in tagging is what to tag posts that have no subject. There were a few posts that had nothing to do with the chosen topic at all, but there were also posts that had no subject but contained phrases such as "Keep it up!", "always a supporter!", etc. that could very well be taken as positive posts since our search is for our specific topic. But because there is no subject in the posts, this cannot be taken with 100% certainty. In these instances, where there is no subject in the posts, we marked them as irrelevant to be on the conservative side in our predictions.

These limitations in tagging will affect our model's accuracy in predicting whether a post contains a positive or negative response to the topic at hand. These limitations also indicate to us that in general, models for predicting whether or not a post is negative or positive towards a chosen topic is limited greatly by the subjectiveness of the sentence's meaning/interpretation which affects the training set used to form the model. This limitation could present a potential future question to look into on how we can improve tagging - whether we should just remove posts that have too subjective of an interpretation to tag or if by studying these types of ambiguous posts, we can create certain priors to help in the tagging process.

2.3 Processing

UTF-8 encoding, GBK, Unicode, Big5

2.3.1 Characteristics of Chinese Language

There has been quite a bit of research done into natural language processing in the English language, but not much on the Chinese language. This is due to the fact that the Chinese language contains unique characteristics that makes it difficult to do natural language processing well.

First, the Chinese language, unlike the English language, has no explicit delimiter between words. In the English language, spaces separate words and so to segment a sentence into its appropriate components is not too difficult since one could just separate based on spaces. The Chinese language however has no such delimiter between words.

An additional difficulty in segmenting a sentence into its appropriate components and words is that the Chinese languages made of separate characters where each character has a meaning of its own, but if you combine two or more characters together into a phrase, the meaning can be completely different. These types of ambiguities are typically easily resolved by humans reading the sentence and realizing what would make most sense in terms of the context of the sentence, but it is not so easy for a computer to preform those types of automatic word/phrase segmentations.

For example, 他好吃 - this sentence could be separated in two ways. The first character means 'he'. The other two words can be taken as two different phrases: 'tastes delicious' or 'loves to eat'. It is obvious that the phrase should be taken as "he loves to eat" because "he tastes delicious" does not make sense but it is difficult to have a computer automatically recognize which sentence makes more sense. Some words also have more than one meaning. For example, 打 can be used in different ways with different meanings. It can be used in the contexts of playing a sport, hitting a person or object, or playing a game.

Second, the Chinese language has many OOV words (out of vocabulary). These are new phrases that are not in dictionaries. These phrases typically come out of cultural references, current hot topics, acronyms, abbreviations, names/nicknames, or just plain slang words. Specifically to the posts that we looked into, out of vocabulary words typically occurred in the case of cultural references, where there are nicknames created for some hot topic issue/person/event of the week just popular slang or informal words and phrases used to convey an emotion. Especially since social media posts are more popular with the young adult generation, many of the words used in the posts are popular informal phrases that would not normally be found in dictionaries. Knowledge of these types of OOV words comes out of knowing the current trends in Asian countries and being up to date with the typical language used by the younger generation.

Third, the Chinese language has two forms - traditional and simplified and it is not a 1-1 correspondence between the two languages which makes it difficult to convert between the two languages for translation or natural language processing.

2.3.2 Characteristics of Sina Weibo Posts

Our analysis takes not only the characteristics of Chinese language, but also the characteristics of Sina Weibo posts into consideration.

The writing of Weibo posts are generally informal. The users may not use standard punctuation marks for separation of sentences and parts of sentences. The most important features are described as follows:

Reposting A user may repost other post. Reposting does not automatically imply agreement or liking. This type of post usually consists of two parts: the reposting user's comment and the post being reposted. The user's comment may be empty or set as default text "Repost" or "转发微博" ("Repost Weibo"). The reposted post itself may include multiple reposting. The topic of keeping track of reposting and identifying agreement or disagreement can be a project itself. In our analysis, only the reposting user's comment is kept.

Spams There are a fair amount of spams on Weibo. Some spam posts are identical except for the URL. Hence, URLs are removed and then we check for duplication in the pre-tagging processing step.

Mentioning A user may mention other users whose usernames are preceded by the **@** symbol. The mentioned usernames may be an integrated part of the post. We define a set of topic-related usernames and substitute the mentioning of these usernames by the corresponding proper nouns. The other mentioned usernames are removed.

Emotion Symbols and Internet Slangs Sina Weibo provides the users a set of emotion symbols, which are corresponding words surrounded by square brackets in text. The users may use other emotion symbols, such as `:)` for smile and `T_T` for crying. Internet slangs are a large part of Out-of-Vocabulary (OOV). Some substitute the characters in a word with the

characters which have similar pronunciation, such as "蜀黍" (Shu3 Shu2) for "叔叔" (Shu1 Shu1, means "uncle"). Some are Internet popular interjections, such as "喵了个咪" (喵: "meow," 了: past tense marker, 个: universal measure word, 咪: "mew") which means "dog my cats."

Topic Topic words are surrounded by the pound signs `#` since Chinese language has no explicit delimiter between words. The topic word can be an integrated part of the post.

2.3.3 Pre-tagging Processing

The data set `Han.txt` contains 22,398 posts.

In order to obtain labeled messages for training and testing purpose, the authors manually provided sentiment tags to a data set containing 3000 messages. The three types of sentiment tags used here are positive, negative, and noninformative.

The data need to be cleaned before manual labeling. A typical post looks like:

`1165303315 2012-04-16 09:55:40 《韩寒收到网友死亡威胁》(来自 @新浪娱乐) http://t.cn/zOprKap`

1. The user identification number and time stamp are removed.
2. Only the reposting user's comment is kept. The reposted part is removed from further analysis. If the resulting string is empty, it will be eliminated as well.
3. URLs are removed.
4. Duplicates are removed.

The output file `hanhanweibo.txt` consists of 13,070 unique posts. 3000 posts are chosen from this data set and will be manually tagged.

2.3.4 Pre-segmentation Processing

As discussed in Section **??**, sentences in Chinese are normally strings of Chinese characters without spaces between words. Hence, word segmentation is crucial for our word-based analysis. According to the characteristics of Weibo posts described in Setion **??**, the following processing is preformed:

1. A set of topic-related usernames are defined. Then the mentioning of these usernames are substituted by the corresponding proper nouns. The other mentioned usernames are removed.
2. A set of emotional symbols and Internet slangs are defined. Then they are substituted by the corresponding word surrounded by square brackets.

2.3.5 Segmentation

汉语词法分析系统 ICTCLAS (Institute of Computing Technology, Chinese Lexical Analysis System) is a well known Chinese word segmentation system developed by Institute of Computing Technology, Chinese Academy of Sciences [**?**]. It offers the functionality of Chinese word segmentation, lexical tagging, named entity recognition, unknown words detection, and the user-defined dictionary. The current version is ICTCLAS 2011, which supports GB2312, GBK, UTF8 and several encodings and has precision rate of 98.45%.

The Java version of ICTCLAS 2011 preforms word segmentation and lexical tagging on a Linux 32-bit machine. A user-defined dictionary is provided. The entries in this dictionary contains proper nouns and some common Internet slangs. For instance, 微博 (Weibo, wei1 bo2) can be written as 围脖 (wei2 bo2, means "scarf"). Some users refer Han Han as 韩少 (韩: Han Han's surname, 少: abbreviation of 少爷, which means "young master of the house").

Even with the user-defined dictionary, some appearance of Han Han's name 韩寒 can not be segmented and tagged correctly. This is corrected directly using regular expression.

2.3.6 Conjunction Rules

Lee and Renganathan [**?**] presented that special consideration should be given to the sentences whose parts are linked by contrasting transitional expressions. In particular, if a sentence contains conjunctions such as "although" and "but," only the part being emphasized will be kept and used to infer the sentiment polarity of this sentence. There are three cases:

1. Although (part A), (part B).
2. (Part A), but (part B).
3. Although (part A), but (part B).

For each case, only part B will be kept.

The four words for "although" are 虽然, 虽说, 虽, and 尽管. The words for "but" are 但, 但是, 不过, 可是, 然而, 只是, 可, 只, 然, and 却.

2.3.7 Stop Words and Punctuation Elimination

Moreover, stop words, non-text strings, and punctuation marks are eliminated. The detailed process is as follows:

1. Remove prepositions, punctuation marks, English character strings, interjections, modal particles, onomatopoeia, and auxiliary words.
2. Remove pre-defined stop words and number strings.

Note that the pre-defined stop words do not contain the following six negation words: 不, 不是, 没有, 没, 无, and 别. These negation words will be used in sentiment score assignment.

2.3.8 Sentiment Score Assignment

Dictionary-based sentiment score can provide us some intuitive understanding of the sentiment polarity of the posts. The dictionaries are obtained from HowNet [**?**]. HowNet is an online extralinguistic common-sense knowledge system for the computation of meaning in human language technology.

Each post is examined and the numbers of positive and negative words are recorded. Positive word contributes $+1$ to the sentiment score, whereas negative word contributes -1. If there is a negation words among the three words before the positive/negative word, their combination will be treated as an entity and their updated contribution is -1 times the original contribution. The six negation words used are 不, 不是, 没有, 没, 无, and 别. The sentiment score of the post is the sum of all the contributions.

Also topic-related positive/negative words are added to the dictionaries. For instance, the users who refer Han Han as 韩少 (韩: Han Han's surname, 少: abbreviation of 少爷, which means "young master of the house") clearly have positive feelings about him.

Another interesting quantities are the numbers of positive and negative words in a neighborhood of a particular person. The neighborhood used in our analysis is three words before and after the person's name.

2.4 Feature analysis

It is of interest to study the relation between words or phrases in the posts. In what follows, we will base our analysis on the frequency matrix $X \in R^{n \times p}$, where the entry x_{ij} stores the frequency of occurencies of the j-th word(or phrase) in the i-th post. We include only words or phrases that have total occurence of at least 10 times over all 3000 posts. Analyzing feature relation can help us better understand word usage and identify possible cluster in the feature space.

One way to study the word usage relation is to look at the cooccurence between any two pair of words. Denote by C the cooccurence matrix, where the entry c_{ij} records the number of cooccurences of the i-th and the j-th word in the same post. Figure XX and XX give the matrix plot and the network plot of the top 50 most frequent phrases.

2.4.1 Sparse principal component analysis (SPCA)

Principal component analysis (PCA) is a standard approach for feature extraction and dimension reduction. In high dimensional setting, Zou et al (2006) [**?**] suggest the following variant of the traditional PCA:

$$(A, B) = \arg\min_{A,B} \left\{ \sum_{i=1}^{n} ||x_i - AB^T x_i||_2^2 + \lambda \sum_{j=1}^{k} ||\beta_j||^2 + \sum_{j=1}^{k} \lambda_{1,j} ||\beta_j||_1 \right\} \quad (1)$$

subject to $A^T A = I_k$.

In the above formulation, only the first k leading principal components are kept, with the corresponding loading matirx $B = (\beta_1, ..., \beta_k) \in R^{p \times k}$. The feature vector x_i's are required to be demeaned. The same λ is used to penalized the square l_2 norm of the all the loading coefficients whereas different regularization parameters $\lambda_{1,j}$ are used for the l_1 norm of the loadings. To solve the SPCA problem, we use the efficient algorithm proposed by Zou et al (2006).

By doing a sparse PCA, we hope to find possible principal components that can summarized the 3000 posts we collected. For simplicity, we set the number of principal component $k = 3$ and tune for the regularization parameters. The results are summarized in Table XX.

Table here. simply include the words; no need the coefficients

Comments on the result here: it is easy to see that the first principal seems to be about the writing of Han Han; the second column is related to the fact that the fraud-crusader Fang Zhouzi suspects the authorship of HanHan's works; the third is about the death threat to Han Han. These three components correspond to three major aspects of Han Han that are known to us.

2.4.2 Sparse graphical models

Graphical models are commonly used in machine learning to study the relation between random variables (See, for example, [**?**]). Here we consider undirected graphical representation of random variables. Each node of the graph represents a random variable. An edge connecting two nodes

represents the conditional dependency between the two random variables given all other random variables (The missing of an edge indicates conditional independency). If, in addition, the joint distribution of the random variables is a multivarite normal with mean μ and covariance matrix Σ, then the i-th node and j-th node are conditionally independent (or, equivalently, missing an edge) if and only if $(\Sigma^{-1})_{ij} = 0$. Therefore, given data $x_1, x_2, ..., x_n \in R^p$, to explore the relation between features, we can estimate the inverse covariance matrix by computing the MLE:

$$\max_S \left\{ \log\det(S) - \mathbf{Tr}(\hat{\Sigma}S) \right\} \tag{2}$$

If the number of features p is large, certain regularization is needed to control the number of edges. Banerjee et al. [**?**] propose the following optimization problem to recover the sparse structure in a gaussian graphical model

$$\max_S \left\{ \log\det S - \mathbf{Tr}\left(\hat{\Sigma}S\right) - \lambda||S||_1 \right\} \tag{3}$$

where Σ is the covariance matrix of the data/design matrix X and $||S||_1 = \sum_{i=1}\sum_{j=1}|s_{ij}|$. To solve for S, Banerjee et al (2007) propose a block coordinate ascent method (COVSEL) (updating one row and one column of S at one time). Their approach is exact but is time consuming. Meinshausen and Buhlmann (2006) uses an approximation approach that is substantially faster. In our study, we adopt the fast and accurate graphical LASSO procedure (R package glasso) by Friedman et al (2007). Figure XX shows the word usage network by fitting a sparse graphical model. Due to space limit, only the top 50 most frequent words are shown.

The network plot of the estimated inverse covariance matrix S. As discussed, an edge (i.e. $S_{ij} \neq 0$) represents conditional dependency between two nodes given all the other nodes. Red edges indicate positive correlated occurence $S_{ij} < 0$ (Given all other words, word i is more likely to occur if word j is observed) and black edges indicate negative correlated occurence $S_{ij} > 0$. Edge width is proportional to $|S_{ij}|$, representing the strength of the tie.

A comparison of the cooccurence network and sparse graphical model: the cooccurence network is heavily influenced by usage frequency of words. For example, "bu"(the negation word) and "Han han"(Han Han) are strongly connected in the cooccurence network, but this might not imply that there is an nontrivial relation between these two words. Sparse graphical models, on the other hand, give more interpretable results. For example, "bu" and "han han" are not longer heavily connected; in addition, some interesting word combinations are revealed: the word "GuangMing" and "LeiLuo" are words that constitute the name of the Han Han's new book Light and Upright; and the clique form by "FangZhouZi","suspect" and "HanHan" seems to reveal the fact that Fang suspects Hanhan has a ghost writer. Other obvious relation revealed include "death" and "threat"; "buy" and "book"; "write" and "article" etc.

2.5 Classification

We are also interested in classifying the posts into different categories. Let $x_i \in R^p$ be the i-th row of the frequency matrix X and y_i the corresponding category. For simplicity, let us assume that $y_i \in \{-1, +1\}$ is binary, where the "+1" can be used to label the following four categories: 1) positive opinion towards Han Han; 2) negative opinion towards Han Han; 3) netural or unidentifiable opinion; 4) spam and we use "-1" to label the complement of the inividual category (e.g. if "+1" means positive, "-1" would mean anything but positive. Note that the complement of positive is not negative, but rather, the union of negative, neutral and spam). For each of the above four cases, we apply LASSO and l_1-norm support vector machine to classify the data points into "-1" and "+1".

2.5.1 Sparse regression with the LASSO

The LASSO (Tibshirani, 1996 [**?**]) is a sparse regression method which adds a l_1-norm penalty to the linear least squares objetive to promote sparsity in the regression coefficients:

$$\hat{\beta}(\lambda) = \arg\min_{\beta} \frac{1}{2}||y - (\beta_0 + X\beta)||_2^2 + \lambda||\beta||_1 \tag{4}$$

In our study, we regress the class label vector y onto the word frequency matrix X, yielding the intercept $\hat{\beta}_0$ and the sparse regression vector $\hat{\beta}(\lambda)$. The classifier can be determined to be $f(x) = \mathbf{sign}(\hat{\beta}_0 + \hat{\beta}^T x) \in \{-1, +1\}$, where the resulting coefficient regression coefficient $\hat{\beta}$ has the following explanation: for each feature/word j, given all other feature/word variable fixed, the increase of the j-th word frequency by one lead to increase in regression function $\beta_0 + \beta^T x$ by an amount of β_i (if β_i turns out to be positive, this means the chance of classifying the data point into the $+1$ category is increased).

To look at which words were most relevant to each category we modeled (positive, negative, neutral and spam), we looked at three sets of 20 words based on, respectively, the highest absolute beta values, most positive coefficient values and most negative coefficient values. The top 30 words based on the top 50 highest absolute beta values tells us in general which words were most relevant to predicting that particular category. The top 50 words based on the top 50 highest positive beta values tells us which words typically were common in posts that fell into the category observed while the top 50 words based on the top 50 highest negative beta values tells us which words typically were not in posts that fell into the category observed but were rather more common in the categories outside of the one we were modeling.

Since the estimated β depend on the regularization parameter, we are left with the issue of choose the "best" λ. A commonly used approach is to do a grid search for λ: for each value of λ, do a 10-fold cross validation; then choose the λ that yields the smallest cross validation testint sample error. For this purpose, we used the approach least-angle regression (LARS) by Efron et al (2007) to do the model selection. Their R package lars efficiently fits an entire lasso sequence with the least squares loss function. The results are summarized in Table XX-XX.

For the positive responses, the top 20 most relevant words (based on taking the absolute values of the betas) returned by our model are shown in the table. As can be seen, the top 20 relevant words have an assortment of fairly neutral or positive words. The top 20 most relevant words for just the positive betas resulted in words that were very positive in nature. For example, words such as 'mature', 'support', and 'keep going' are very positive in nature and would certainly indicate a positive reaction to 韩寒. The top 20 most relevant words for just the negative betas resulted in words that showed a great dislike for 韩寒. Words such as 'liar' indicate a negative response to the author.

For the negative responses, the top 20 most relevant words (based on taking the absolute values of the betas) returned by our model are shown in the table. As can be seen, the words are typically negative or neutral. The top 20 most relevant words for just the positive betas are very telling in the posts sentiment towards 韩寒. Words such as 'disgusting, 'hate', 'liar' and 'annoying' demonstrates easily that the post has a negative sentiment towards the author. The top 20 most relevant words for just the negative betas are hence typically more positive towards the author. With words such as 'support', 'good', and 'like', the posts would not have a negative sentiment. What is interesting to note with these betas is the fact that these betas are all close to zero except the highest phrase.

For the neutral responses, the top 20 most relevant words (based on taking the absolute values of the betas) as well as the top 20 most relevant words based on the positive betas are all neutral words. The top 20 most relevant words based on the negative betas consist mainly of phrases that

Table 1: Positive category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
加油 (keep going)	0.820	加油 (keep going)	0.820	样子 (manner)	-0.396
韩少 (Master Han)	0.644	韩少 (Master Han)	0.644	恋 (love)	-0.344
成熟 (mature)	0.546	成熟 (mature)	0.546	发表 (announce)	-0.336
顶 (support)	0.533	顶 (support)	0.533	道理 (rational)	-0.336
宽容 (tolerant)	0.518	宽容 (tolerant)	0.518	利益 (benefit)	-0.335

Table 2: Negative category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
讨厌 (hate)	0.481	讨厌 (hate)	0.481	支持 (support)	-0.008
无耻 (shameless)	0.412	无耻 (shameless)	0.412	不 (no)	0.000
恶心 (disgusting)	0.395	恶心 (disgusting)	0.395	人 (people/person)	0.000
骗子 (liar)	0.380	骗子 (liar)	0.380	说 (say)	0.000
扁 (beat up)	0.353	扁 (beat up)	0.353	方舟子 (FangZhouZi)	0.000

have some clear emotion attached to them, such as ”support”, ”hate”, ”agree”, and ”ghostwriter”. For the spam posts, the top 20 most relevant words (based on taking the absolute values of the betas) as well as the top most relevant words based on the positive betas are words/phrases that have no relation whatsoever with the author we picked to look into. As expected, the top 20 most relevant words based on the negative betas are words/phrases that do have to do with the topic, such as the author's name.

for the spam responses...

2.5.2 l_1-norm support vector machine

The support vector machine (SVM) is another commonly used machine learning method to classify data points into two categories. Consider again the linear decision function $f(x) = \beta_0 + \beta^T x$ and the sign classifier $\text{class}(x) = \textbf{sign}(f(x))$. The SVM minimizes the training misclassification

Table 3: Neutral category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
上调 (increase)	0.586	上调 (increase)	0.586	加油 (keep going)	-0.491
道理 (rational)	0.566	道理 (rational)	0.566	韩少 (Master Han)	-0.358
账号 (account)	0.534	账号 (account)	0.534	苦肉计 (the ruse of self-injury to win somebody's confidence)	-0.327
加油 (keep going)	0.491	铁证 (clear evidence)	0.459	支持 (support)	-0.290
铁证 (clear evidence)	0.459	称 (refer)	0.453	善良 (kind)	-0.268

Table 4: Spam category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
查看 (examine)	3.777	查看 (examine)	3.777	韩少 (Master Hanhan)	-1.033
抽 (win)	1.998	抽 (win)	1.998	韩寒 (Master Hanhan)	-0.716
每天 (everyday)	1.251	每天 (everyday)	1.251	别 (don't)	-0.232
往往 (often)	1.208	往往 (often)	1.208	支持 (support)	-0.217
外 (outside)	1.043	外 (outside)	1.043	这种 (this kind)	-0.202

rate and the margin of the decision boundary. Following the notations in [**?**]:

$$\min_{\beta_0,\beta} \sum_{i=1}^{n} (1 - y_i(\beta_0 + \beta^T x_i))_+ + \frac{\lambda}{2}||\beta||_2, \tag{5}$$

where $z_+ = \max(0, z)$ (the function $h(z) = (1 - z)_+$ is also known as the hinge loss function). Similarly, the sparse version of SVM simply replaces the l_2-norm by l_1-norm (see, e.g. [**?**]):

$$\min_{\beta_0,\beta} \left\{ \sum_{i=1}^{n} (1 - y_i(\beta_0 + \beta^T x_i))_+ + \lambda||\beta||_1 \right\}.$$

We repeat the same data analysis for the text data using l_1-norm SVM as we did for the LASSO classification. The results are summarized in Table XX. To fit the sparse SVM, we use the matlab package `lpsvm` by Fung and Mangasarian (2004) [**?**]. Again, 10-fold cross validations are performed in order to select the regularization parameter λ. Again the model seems to produce plausible results. Also, the top coefficients obtained by l_1-norm SVM seems to be consistent with those obtained by fitting a LASSO.

Table 5: l_1-norm support vector machine results: positive category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
加油 (keep going)	2.340	加油 (keep going)	2.340	铁证 (clear evidence)	2.305
铁证 (clear evidence)	2.305	家人 (family)	2.269	接受 (accept)	2.061
家人 (family)	2.269	韩少 (Master Han)	1.969	媒体 (media)	1.907
接受 (accept)	2.061	成熟 (mature)	1.806	默默 (quietly)	1.883
韩少 (Master Han)	1.969	顶 (support)	1.803	四娘 (GUO Jingming)	1.762

3 Discussion

ROC curve precision and recall curve

3.1 Limitations

sampling [**?**] [**?**] [**?**]
reposting
other language, such as English
simplified Chinese and traditional Chinese: no simple one-to-one correspondence; word segmentation and then substitute words

4 Conclusion

Table 6: l_1-norm support vector machine results: negative category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
扁 (beat up)	1.777	扁 (beat up)	1.777	脑子 (mind)	1.447
苦肉计 (the ruse of self-injury to win somebody's confidence)	1.708	苦肉计 (the ruse of self-injury to win somebody's confidence)	1.708	彻底 (completely)	1.290
恶心 (disgusting)	1.527	恶心 (disgusting)	1.527	送给 (give)	1.221
脑子 (asdf)	1.447	骗子 (liar)	1.301	感觉 (feel)	1.109
骗子 (liar)	1.301	公开 (open)	1.220	热点 (hot interest)	1.101

Table 7: l_1-norm support vector machine results: neutral category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
加油 (keep going)	2.233	铁证 (clear evidence)	2.054	加油 (keep going)	2.233
铁证 (clear evidence)	2.054	片 (piece)	1.896	成熟 (mature)	1.756
片 (piece)	1.896	至今 (so far)	1.884	同意 (agree)	1.725
至今 (so far)	1.884	战 (fight)	1.845	水 (water)	1.661
战 (fight)	1.845	意思 (meaning)	1.824	昨天 (yesterday)	1.611

Table 8: l_1-norm support vector machine results: spam category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
票子 (ticket)	1.000	票子 (ticket)	1.000	书 (book)	0.000
书 (book)	0.000	每天 (everyday)	0.000	围观 (surround to watch)	0.000
围观 (surround to watch)	0.000	抽 (win)	0.000	写 (write)	0.000
写 (write)	0.000	性 (sex)	0.000	骂 (curse)	0.000
每天 (everyday)	0.000	网 (Internet)	0.000	粉丝 (fans)	0.000

References

[1] China shuts down microblog accounts amid rumors of coup attemps. http://www.washingtonpost.com/business/china-shuts-down-microblog-accounts-amid-rumors-of-coup-attemps/2012/04/28/gIQAxYTJoT_story.html.

[2] Sina's Weibo outlook buoys Internet stock gains: China overnight. http://www.bloomberg.com/news/2012-02-28/sina-s-weibo-outlook-buoys-internet-stock-gains-in-n-y-china-overnight.html.

[3] Boyd, S., Diaconis, P., and Xiao, L. Fastest mixing markov chain on a graph. *SIAM review* (2004), 667–689.

[4] Dong, Z., and Dong, Q. 知网 HowNet. http://www.keenage.com/.

[5] Institute of Computing Technology, Chinese Academy of Sciences. 汉语词法分析系统 ICTCLAS (Institute of Computing Technology, Chinese Lexical Analysis System). http://ictclas.org/, 2011.

[6] Lee, H., and Renganathan, H. Chinese sentiment analysis using maximum entropy. *Sentiment Analysis where AI meets Psychology (SAAIP)* (2011), 89.

[7] Leskovec, J., and Faloutsos, C. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (2006), ACM, pp. 631–636.

[8] Wang, T., Chen, Y., Zhang, Z., Xu, T., Jin, L., Hui, P., Deng, B., and Li, X. Understanding graph sampling algorithms for social network analysis. In *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on* (2011), IEEE, pp. 123–128.

[9] Wong, K., Li, W., Xu, R., and Zhang, Z. Introduction to chinese natural language processing. *Synthesis Lectures on Human Language Technologies 2*, 1 (2009), 1–148.

[10] LAURENT EL GHAOUI, GUAN-CHENG LI, VIET-AN DUONG, VU PHAM, ASHOK SRIVASTAVA, AND KANISHKA BHADURI. Sparse machine learning methods for understanding large text corpora *Proc. Conference on Intelligent Data Understanding*, 1 (2011).

[11] HUI ZOU, TREVOR HASTIE, ROBERT TIBSHIRANI. Sparse Principal Component Analysis *Journal of computational statistics and graphics*, (2006).

[12] WAINRIGHT M AND JORDAN M. Graphical Models, Variational Methods and Message-Passing. *Foundations and Trends in Machine Learning*, (2003).

[13] O.BANERJEE AND L. EL GHAOUI AND A. D'ASPREMONT Model Selection Through Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data. *Journal of Machine Learning Research*, (2008).

[14] MEINSHAUSEN N. AND BÜHLMANN P. High-dimensional graphs and variable selection with the Lasso *Annals of Statistics*. (2006).

[15] TREVOR HASTIE AND ROBERT TIBSHIRANI AND SAHARON ROSSET AND JI ZHU. The Entire Regularization Path for the Support Vector Machine. *The Journal of Machine Learning Research*. (2004).

[16] JI ZHU, SAHARON ROSSET, TREVOR HASTIE, ROB TIBSHIRANI. 1-norm Support Vector Machines. *NIPS*. (2003).

[17] G. FUNG AND O. L. MANGASARIAN. A Feature Selection Newton Method for Support Vector Machine Classification *Data Mining Institute, Computer Sciences Department, University of Wisconsin*. (2002)

[18] TIBSHIRANI R. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.* (1996)

A LASSO Results

Table 9: LASSO results: positive category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
加油 (keep going)	0.820	加油 (keep going)	0.820	样子 (manner)	-0.396
韩少 (Master Han)	0.644	韩少 (Master Han)	0.644	恋 (love)	-0.344
成熟 (mature)	0.546	成熟 (mature)	0.546	发表 (announce)	-0.336
顶 (support)	0.533	顶 (support)	0.533	道理 (rational)	-0.336
宽容 (tolerant)	0.518	宽容 (tolerant)	0.518	利益 (benefit)	-0.335
支持	0.477	支持	0.477	称	-0.323
家人	0.467	家人	0.467	遭受	-0.323
样子	0.396	尤其	0.395	媒体	-0.319
尤其	0.395	欣赏	0.383	翻	-0.314
欣赏	0.383	感动	0.381	铁证	-0.289
感动	0.381	影响力	0.370	骗子	-0.248
影响力	0.370	新书	0.327	上调	-0.248
恋	0.344	铁	0.316	投票	-0.234
发表	0.336	不错	0.309	女	-0.230
道理	0.336	终于	0.274	四娘	-0.226
利益	0.335	每个	0.274	关系	-0.215
新书	0.327	咬	0.261	广告	-0.210
称	0.323	文字	0.260	接受	-0.208
遭受	0.323	蛋	0.244	网	-0.204
媒体	0.319	纠缠	0.244	底	-0.196

Table 10: LASSO results: negative category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
讨厌 (hate)	0.481	讨厌 (hate)	0.481	支持 (support)	-0.008
无耻 (shameless)	0.412	无耻 (shameless)	0.412	不 (no)	0.000
恶心 (disgusting)	0.395	恶心 (disgusting)	0.395	人 (people/person)	0.000
骗子 (liar)	0.380	骗子 (liar)	0.380	说 (say)	0.000
扁 (beat up)	0.353	扁 (beat up)	0.353	方舟子 (FangZhouZi)	0.000
装	0.321	装	0.321	韩少	0.000
选项	0.292	选项	0.292	真	0.000
苦肉计	0.290	苦肉计	0.290	好	0.000
利益	0.283	利益	0.283	没	0.000
全	0.261	全	0.261	一个	0.000
国家	0.247	国家	0.247	微博	0.000
智商	0.216	智商	0.216	写	0.000
告	0.198	告	0.198	喜欢	0.000
虚伪	0.192	虚伪	0.192	想	0.000
演	0.191	演	0.191	威胁	0.000
语	0.186	语	0.186	只	0.000
烦	0.178	烦	0.178	太	0.000
掉	0.142	掉	0.142	事	0.000
下去	0.141	下去	0.141	没有	0.000
公开	0.141	公开	0.141	看到	0.000

Table 11: LASSO results: neutral category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
上调 (increase)	0.586	上调 (increase)	0.586	加油 (keep going)	-0.491
道理 (rational)	0.566	道理 (rational)	0.566	韩少 (Master Han)	-0.358
账号 (account)	0.534	账号 (account)	0.534	苦肉计 (the ruse of self-injury to win somebody's confidence)	-0.327
加油 (keep going)	0.491	铁证 (clear evidence)	0.459	支持 (support)	-0.290
铁证 (clear evidence)	0.459	称 (refer)	0.453	善良 (kind)	-0.268
称	0.453	想起	0.353	成熟	-0.263
韩少	0.358	杀	0.331	终于	-0.239
想起	0.353	最终	0.329	家人	-0.233
杀	0.331	意思	0.323	同意	-0.228
最终	0.329	遭遇	0.319	越来越	-0.220
苦肉计	0.327	金	0.308	欢乐	-0.217
意思	0.323	片	0.287	崇拜	-0.213
遭遇	0.319	应	0.278	讨厌	-0.202
金	0.308	变成	0.265	顶	-0.197
支持	0.290	有点	0.262	代笔	-0.194
片	0.287	之间	0.254	跳	-0.191
应	0.278	右边	0.250	真善美	-0.186
善良	0.268	民主	0.238	真正	-0.185
变成	0.265	郭敬明	0.237	欣赏	-0.181
成熟	0.263	久	0.226	无耻	-0.180

Table 12: LASSO results: spam category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
查看 (examine)	3.777	查看 (examine)	3.777	韩少 (Master Hanhan)	-1.033
抽 (win)	1.998	抽 (win)	1.998	韩寒 (Master Hanhan)	-0.716
每天 (everyday)	1.251	每天 (everyday)	1.251	别 (don't)	-0.232
往往 (often)	1.208	往往 (often)	1.208	支持 (support)	-0.217
外 (outside)	1.043	外 (outside)	1.043	这种 (this kind)	-0.202
韩少	1.033	征集	0.948	感	-0.196
征集	0.948	容	0.849	韓	-0.191
容	0.849	风	0.649	没有	-0.179
韩寒	0.716	票子	0.570	上调	-0.174
风	0.649	考	0.540	方舟子	-0.160
票子	0.570	主	0.438	光明	-0.141
考	0.540	性	0.430	一定	-0.137
主	0.438	总是	0.416	照妖镜	-0.132
性	0.430	儿	0.416	写	-0.132
总是	0.416	结论	0.405	觉得	-0.119
儿	0.416	后面	0.397	甚	-0.110
结论	0.405	法律	0.388	韩	-0.092
后面	0.397	机会	0.376	真相	-0.084
法律	0.388	公知	0.357	挺	-0.072
机会	0.376	中	0.357	不	-0.068

B l_1-Norm Support Vector Machine Results

Table 13: l_1-norm support vector machine results: positive category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
加油 (keep going)	2.340	加油 (keep going)	2.340	铁证 (clear evidence)	2.305
铁证 (clear evidence)	2.305	家人 (family)	2.269	接受 (accept)	2.061
家人 (family)	2.269	韩少 (Master Han)	1.969	媒体 (media)	1.907
接受 (accept)	2.061	成熟 (mature)	1.806	默默 (quietly)	1.883
韩少 (Master Han)	1.969	顶 (support)	1.803	四娘 (GUO Jingming)	1.762
媒体	1.907	宽容	1.764	骗子	1.524
默默	1.883	人士	1.758	想起	1.519
成熟	1.806	支持	1.593	恋	1.491
顶	1.803	欢乐	1.315	韩寒和	1.476
宽容	1.764	影响力	1.237	关	1.415
四娘	1.762	诛	1.175	变成	1.411
人士	1.758	欣赏	1.138	圈	1.243
支持	1.593	幸福	1.126	曾经	1.238
骗子	1.524	纠缠	1.030	战	1.225
想起	1.519	代笔	1.025	网	1.219
恋	1.491	喜欢	0.943	发表	1.190
韩寒和	1.476	蛋	0.923	闲	1.190
关	1.415	明	0.919	小四	1.125
变成	1.411	终于	0.914	底	1.050
欢乐	1.315	咬	0.884	套	1.044

Table 14: l_1-norm support vector machine results: negative category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
扁 (beat up)	1.777	扁 (beat up)	1.777	脑子 (mind)	1.447
苦肉计 (the ruse of self-injury to win somebody's confidence)	1.708	苦肉计 (the ruse of self-injury to win somebody's confidence)	1.708	彻底 (completely)	1.290
恶心 (disgusting)	1.527	恶心 (disgusting)	1.527	送给 (give)	1.221
脑子 (asdf)	1.447	骗子 (liar)	1.301	感觉 (feel)	1.109
骗子 (liar)	1.301	公开 (open)	1.220	热点 (hot interest)	1.101
彻底	1.290	全	1.154	恶	1.077
送给	1.221	国家	1.149	青春	1.013
公开	1.220	讨厌	1.053	少	0.949
全	1.154	网	1.034	算	0.940
国家	1.149	烦	0.889	清楚	0.921
感觉	1.109	虚伪	0.843	愿意	0.920
热点	1.101	装	0.812	新书	0.868
恶	1.077	告	0.712	写作	0.840
讨厌	1.053	讨论	0.674	争论	0.827
网	1.034	难	0.636	地方	0.796
青春	1.013	时代	0.633	铁	0.789
少	0.949	天才	0.629	言	0.737
算	0.940	样子	0.626	子	0.727
清楚	0.921	选项	0.596	来自	0.705
愿意	0.920	喝	0.559	精彩	0.685

Table 15: l_1-norm support vector machine results: neutral category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
加油 (keep going)	2.233	铁证 (clear evidence)	2.054	加油 (keep going)	2.233
铁证 (clear evidence)	2.054	片 (piece)	1.896	成熟 (mature)	1.756
片 (piece)	1.896	至今 (so far)	1.884	同意 (agree)	1.725
至今 (so far)	1.884	战 (fight)	1.845	水 (water)	1.661
战 (fight)	1.845	意思 (meaning)	1.824	昨天 (yesterday)	1.611
意思	1.824	遭遇	1.729	韩少	1.604
成熟	1.756	儿子	1.701	路	1.568
遭遇	1.729	关系	1.648	国家	1.553
同意	1.725	生日	1.646	讨厌	1.533
儿子	1.701	道德	1.626	咬	1.485
水	1.661	称	1.610	人士	1.438
关系	1.648	杀	1.578	掉	1.322
生日	1.646	接受	1.567	小丑	1.311
道德	1.626	狂	1.508	偏执	1.282
昨天	1.611	账号	1.485	抽	1.257
称	1.610	理解	1.444	语	1.254
韩少	1.604	道理	1.341	声	1.252
杀	1.578	不幸	1.332	家人	1.235
路	1.568	加	1.293	一直	1.213
接受	1.567	上调	1.267	自由	1.198

Table 16: l_1-norm support vector machine results: spam category

Word	Absolute Coef.	Word	Positive Coef.	Word	Negative Coef.
票子 (ticket)	1.000	票子 (ticket)	1.000	书 (book)	0.000
书 (book)	0.000	每天 (everyday)	0.000	围观 (surround to watch)	0.000
围观 (surround to watch)	0.000	抽 (win)	0.000	写 (write)	0.000
写 (write)	0.000	性 (sex)	0.000	骂 (curse)	0.000
每天 (everyday)	0.000	网 (Internet)	0.000	粉丝 (fans)	0.000
抽	0.000	分享	0.000	寫	0.000
骂	0.000	容	0.000	犯	0.000
性	0.000	公知	0.000	不错	0.000
粉丝	0.000	总	0.000	转	0.000
网	0.000	小	0.000	愤	0.000
分享	0.000	图	0.000	韩寒	0.000
寫	0.000	说	0.000	磊落	0.000
容	0.000	真	0.000	错	0.000
犯	0.000	支持	0.000	韩寒和	0.000
不错	0.000	韩	0.000	韩少	0.000
转	0.000	喜欢	0.000	方舟子	0.000
公知	0.000	想	0.000	觉得	0.000
愤	0.000	威胁	0.000	已经	0.000
韩寒	0.000	事	0.000	娱乐	0.000
总	0.000	看到	0.000	新	0.000