Emotion Estimation of Comments on Web News by SVM and Naive Bayes Based Classifiers

Yasuhiro Tajima and Genichiro Kikui
Department of Systems Engineering, Okayama Prefectural University,
111, Kuboki, Soja, Okayama 719-1197, Japan

Abstract—*Social communication tools such as Twitter or Facebook spread the web service ability. Using their APIs, we can gather many users' comments easily. Such comments are usually short sentences but they also have many emotional comments. In this paper, we propose emotion estimation methods for multilabeled short comments of web news. Our methods can be applied to sentiment analysis and opinion mining. At first, we show the performance evaluation of a naive Bayes classifier and an SVM classifier. Then, we propose two improved methods. The first is an improved naive Bayes method which classifies each emotion label into two opposite emotions and uses their weights. We call this the weighting method. The second method consists of two stages of classifiers. The first stage distinguishes these oppositely classes, and the second stage selects one emotion from the opposite emotions. From our evaluation, we conclude that the weighting method is better among the naive Bayes classifiers and its performance is as good as SVM's.*

Keywords: emotion estimation, Twitter, naive Bayes, SVM

1. Introduction

In recent years, social networking tools have very important role on human communication such as Twitter or Facebook and so on. They usually have APIs for mash up with other web services. Especially, many web news sites use this function for gathering users' comments. Some TV programs also use these tools to make a bidirectional communication. These comments are useful for both of article writers and readers, but oftenly there is no retrieval system. Even though, there will be text base retrieval such as search engines and marker based systems such as "hash tag", but there is no system which responses to the request as "search funny comments."

In this paper, we propose an emotion labeling method to such comments. The emotions is comment writer's emotion. For example, if there is a news article about some crime and a comments such that "It will happen near my town.", then the comment writer may feel "fear" and "anticipation." Comments of web news articles have the following properties.

- They will be more emotional than other tweets. The comments to the news article are usually impressive.
- They will be short sentences. Twitter restricts the length of comments up to 140 characters, and other social tools have the same restriction.
- There will be no discussion. Some board systems have the comment tree making function, but many systems do not have.

Automatically emotion estimation of tweets is useful from these reasons.

In this paper, we propose two naive Bayes based classifier and performance evaluation with SVM and the simple naive Bayes classifier. Our new method uses the class of emotions which consists of two opposite emotions such as "joy" and "sadness". This is because, we use Plutchik's wheel of emotions[5], and there are eight emotions which can be classified into four classes. For the evaluation, we made experiments by Japanese news articles and their about 2000 tweets. The SVM and our proposed new method marked high performances comparing to the simple naive Bayes classifier.

There are some related studies about emotion estimation. In [1], a Japanese valency pattern dictionary for emotions has been made and emotion estimation for a sentence has been tried. An emotion corpus has also been made in [2]. Machine learning approach to emotion estimation has been tried in [3] and [4].

2. Vector models for emotion estimation

We denote a sentence of a tweet t which consists of n words by $t = w^{(1)}w^{(2)}\cdots w^{(n)}$. W_l and W_t denote vocabularies which appear in learning data and evaluation data, respectively. Let $W = W_l \cup W_t$, then $|W|$ denotes the size of the vocabulary of all data. Without loss of generality, assume an order on $W = \{w_1, w_2, \cdots, w_m\}$ and another order on $W_l = \{w_1, w_2, \cdots, w_l\}$. Now, $m = |W|$ and $l = |W_l|$ hold. On the naive vector modeling, we assume a map from a tweet t to m-dimensional vector $(u_1, u_2, \cdots, u_m)$. In this paper, u_i is the number of w_i which appears in t, i.e. $u_i = |\{j | w_i = w^{(j)}\}|$ where $t = w^{(1)}w^{(2)}\cdots w^{(n)}$. We denote the appearance of w_i by δ_i such that

$$\delta_i = \begin{cases} 1 & \exists j, w_i = w^{(j)} \\ 0 & otherwise \end{cases}$$

for $i = 1, 2, \cdots, m$. To avoid the zero frequency problem, we use additive smoothing for naive Bayes based methods.

We do not care the words which only exists in evaluation data such that $w \in (W_t - W_l)$ for SVM classification.

Target emotions are the following eight emotions.

- joy
- trust
- fear
- surprise
- sadness
- disgust
- anger
- anticipation

These are components of Plutchik's wheel of emotions[5]. In our setting, every tweet can have multilabels of emotions. A tweet t can be labeled by both of "joy" and "surprise" for example. Every tweet must have at least one label of the above emotions.

These eight emotions can be classified into four classes such that

- joy $\Longleftrightarrow$ sadness
- trust $\Longleftrightarrow$ disgust
- fear $\Longleftrightarrow$ anger
- surprise $\Longleftrightarrow$ anticipation

because of the pair of opposite emotions.

3. Estimation method

3.1 Simple naive Bayes

For probabilistic variables X, Y, it hols that

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

and this is called Bayes' theorem. Y denotes the target event. In our method, Y can take an event from {joy, trust, fear, surprise, sadness, disgust, anger, anticipation}. X denotes the vector which corresponds to a tweet t, i.e. X is an m-dimensional vector $(u_1, u_2, \cdots, u_m)$. If it holds that $P(Y|(u_1, u_2, \cdots, u_m)) \geq Th$ then the tweet t is labeled by the emotion Y. Here, Th is the threshold value and we define it $\frac{1}{8} = 0.125$ because there are eight emotions. If there exist more than two emotions, for example $P(Y = \text{“}joy''|(u_1, u_2, \cdots, u_m)) > Th$ and $P(Y = \text{“}trust''|(u_1, u_2, \cdots, u_m)) > Th$ holds, then t is a multilabeled tweet by "joy" and "trust". It holds that

$$P(Y|(u_1, u_2, \cdots, u_m)) = \frac{P((u_1, u_2, \cdots, u_m)|Y)P(Y)}{P((u_1, u_2, \cdots, u_m))}$$

from Bayes' theorem. In addition, $P((u_1, u_2, \cdots, u_m)|Y)$ and $P((u_1, u_2, \cdots, u_m))$ can be approximated by the followings.

$$P((u_1, u_2, \cdots, u_m)|Y) = \prod_{i=1,\cdots,m} P(w_i|Y)^{u_i}$$

$$P((u_1, u_2, \cdots, u_m)) = \prod_{i=1,\cdots,m} P(w_i)^{u_i}$$

Thus, $P(w|Y)$ and $P(w)$ for all $w \in W$ are needed to decide the labels of t. These values are estimated from learning data. $P(w|Y)$ is the probability that w appearance in all tweets with the emotion label of Y. $P(w)$ is the probability that w appearance in learning data.

3.2 Weighted naive Bayes

We can classify the set of emotions introduced by Plutchik[5]. That is four classes and each of them consists of opposite emotions: joy and sadness, trust and disgust, fear and anger, surprise and anticipation. Now, we assume that only one emotion on the each pair tends to be labeled. Let

$$\begin{array}{ll} y_1 = \text{“joy”}, & n_1 = \text{“sadness”}, \\ y_2 = \text{“trust”}, & n_2 = \text{“disgust”}, \\ y_3 = \text{“fear”}, & n_3 = \text{“anger”}, \\ y_4 = \text{“surprise”}, & n_4 = \text{“anticipation”} \end{array}$$

and $C_i = \{y_i, n_i\}$ for $i = 1, 2, 3, 4$. One emotion can be written by (C_i, m_i) where $m_i \in C_i$ for $i = 1, 2, 3, 4$. Let C and M are probabilistic variables of C_i and m_i, respectively. Then, $P(Y|w)$ can be written by the following for a word $w \in W$.

$$\begin{aligned} P(Y|w) &= P(C, M|w) \\ &= P(M|w, C)P(C|w) \\ &= \frac{P(w|C, M)P(M|C)}{P(w|C)} P(C|w) \end{aligned}$$

here, $P(M|w, C)$ means the emotion distribution when w and C are given. We approximate $P(C|w)$ by the probability that the emotion C is labeled to the tweet t which has w. For example, assume that there are x tweets in which w appears, and y tweets are labeled by C_i among these x tweets. Then, $P(C = C_i|w) = \frac{x}{y}$. $P(M|C)$ is also calculated from number of tweets. For example, if there are z tweets labeled by C_1 and x tweets labeled by "joy", then $P(M = \text{“}joy''|C_1) = \frac{x}{z}$.

For a tweet t which corresponds to $(u_1, u_2, \cdots, u_m)$, we approximates $p(Y|t)$ as follows.

$$\begin{aligned} P(Y|t) &= P(Y|(u_1, u_2, \cdots, u_m)) \\ &= \prod_{i=1,2,\cdots,m} P(Y|w_i)^{u_i} \end{aligned}$$

If $P(Y|t) > Th$ then t has the emotion label of Y.

We call this method "weighted naive Bayes" because $P(C|w)$ looks like a weight for $P(M|w, C)$.

3.3 Two stages naive Bayes

We use four classes of emotions C_i for $i = 1, 2, 3, 4$ which are defined in the previous section. In this method, two threshold value Th and Tc is used. At the first stage, $P(C_i|w)$ for every $i = 1, 2, 3, 4$ is calculated and check them whether $P(C_i|w) > Tc$ or not. If $P(C_i|w) \leq Tc$ then no label $m_j \in C_i$ is labeled to the target tweet. If $P(C_i|w) > Tc$ then select emotion m_i from C_i according to whether $P(m_i|C_i, t) > Th$ or not. The target tweet takes the label m_i When $P(m_i|C_i, t) > Th$. We call this step the second stage. Fig. 1 shows the flow of this method. In this paper, we use $Th = Tc = 0.1$ from some preliminary experiment.

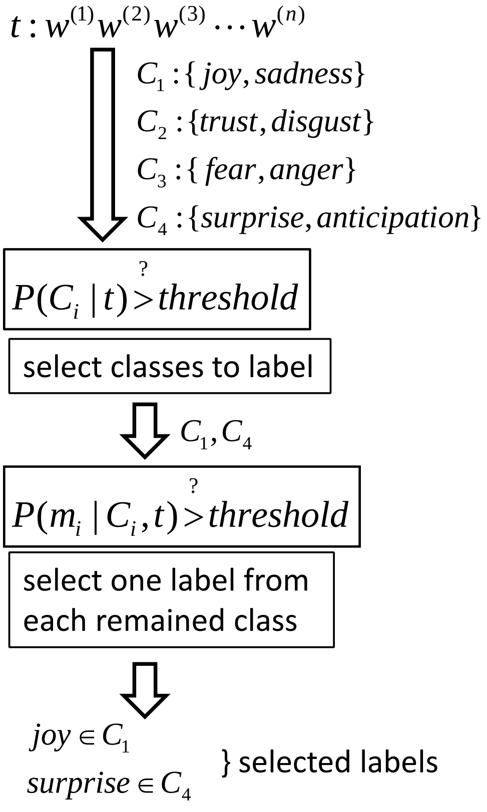
$t : w^{(1)} w^{(2)} w^{(3)} \cdots w^{(n)}$
$C_1 : \{joy, sadness\}$
$C_2 : \{trust, disgust\}$
$C_3 : \{fear, anger\}$
$C_4 : \{surprise, anticipation\}$
$P(C_i \mid t) \overset{?}{>} threshold$
select classes to label
C_1, C_4
$P(m_i \mid C_i, t) \overset{?}{>} threshold$
select one label from each remained class
$joy \in C_1$
$surprise \in C_4$
} selected labels

Fig. 1: The flow of two stage method

When either Tc or Th is too low, the target tweet may have many labels and it increases the recall but decreases the precision.

3.4 SVM

SVM is a discriminative classifier which is based on margin maximization. In this paper, emotion labeling via SVM is processed as follows.

- Train an SVM for every emotion label which discriminates one emotion from the others. Then, there are eight SVMs and such SVMs are denoted by S_i for $i = 1, 2, \cdots, 8$. The input of each SVM S_i is a vector $(\delta_1, \delta_2, \cdots, \delta_l)$ of a tweet t which expresses the word appearance in t of the vocabulary of learning data. The output of S_i is whether the input tweet has the i-th emotion label or not.
- To predict that a tweet t has i-th emotion or not, make the input vector of t for S_i such that $(\delta_1, \delta_2, \cdots, \delta_l)$. Then, predict the emotion label according to the output of S_i for $i = 1, 2, \cdots, 8$.

SVM has a parameter C which is the weight of slack variables. We examine some values of C and their performances. In this paper, we use linear classifier with slack variables and L_2 norm. "liblinear" is one of the most effective implementation of linear SVM and we use this software for our experiments.

4. Evaluation

4.1 Data description

For experiment, web news and their comments are gathered. comments are tweets which attached to the news article. The news site is news.nicovideo.jp and tweets are processed as follows.

- Re-tweets are all deleted.
- All meaningless spaces and tabs are deleted.
- Comments for other tweets (re-tweets with original comment) are remained.

Then, our data descriptions are as follows.

- Total number of news article : 28
- Total number of tweets : 2075
- The average number of tweets per article : 78.04
- The maximum number of tweets per article : 100
- The minimum number of tweets per article : 12
- The average number of words per tweet : 19.34
- The maximum number of words per tweet : 65
- The minimum number of words per tweet : 1
- The size of vocabulary : 4987

We do not use news article body for learning and evaluation. Learning data only consist of tweets. All these news articles and tweets are in Japanese. Thus, we must do morphological analyze to all tweets. The morphological analyzer by which all tweets are processed is "mecab." Every word consists of the pair of morpheme and whose tag.

Correct labels of emotions are made by hand. There are twelve persons to make the correct labels. One person can label one emotion per tweet. We call such a label "point." Every tweet must be labeled by at least two persons to avoid bias. Thus, every tweet has at least two points. Learning data is a pair of a tweet and an emotion vector such that

$$(z_1, z_2, \cdots, z_8)$$

here,

$$z_i = \begin{cases} 0 & t's\ i-th\ emotion\ is\ 0\ point \\ 1 & otherwise \end{cases}$$

i.e. if tweet t is labeled on some emotion, the emotion has more than or equal to 1 point.

The average of points per tweet is 2.52. The maximum point is 9 and the minimum point is 2. The followings are point distribution of correct data.

- joy : 516
- sadness : 756
- trust : 147
- disgust : 1347
- fear : 291
- anger : 936
- surprise : 580
- anticipation : 656

The followings are the number of tweets whose emotion vector has more than 1 point.

- joy : 326 tweets
- sadness : 583 tweets
- trust : 130 tweets
- disgust : 954 tweets
- fear : 217 tweets
- anger : 621 tweets
- surprise : 405 tweets
- anticipation : 474 tweets

The average number of emotions whose point is more than or equals to 1 per tweet is 1.79. The maximum is 5 emotions and the minimum is 1 emotion.

It is expected that there are many points on "disgust" because no one has responsibility to comments of web news. Indeed, "disgust" is the most labeled emotion. We choice the base line that emotion vector is only labeled by "disgust." Then, the performance of the base line is as follows.

- precision : 0.46
- recall : 0.25
- F value : 0.32

4.2 Experiment and results

For evaluation, experiments for each method with our learning data are executed and the performances are measured. In our all experiments, the cost C of slack variable on SVM is set to $C = 1.0$.

4.2.1 Simple cross validation

Table 1: Simple 5-fold cross validation

	simple	weighted	2stage	SVM
precision	0.4590	0.4718	0.4632	0.5610
recall	0.5970	0.6056	0.5818	0.5159
F value	0.5190	0.5304	0.5158	0.5375

Fig. 2 and Table 1 show the results of our methods using a 5-fold cross validation. The 5-fold is made by the followings.

1) For all tweets of one article are divided into 5 parts.
2) The evaluation data is the set of every one part of tweets from all articles. Thus, there are $\frac{1}{5}$ of all tweets.

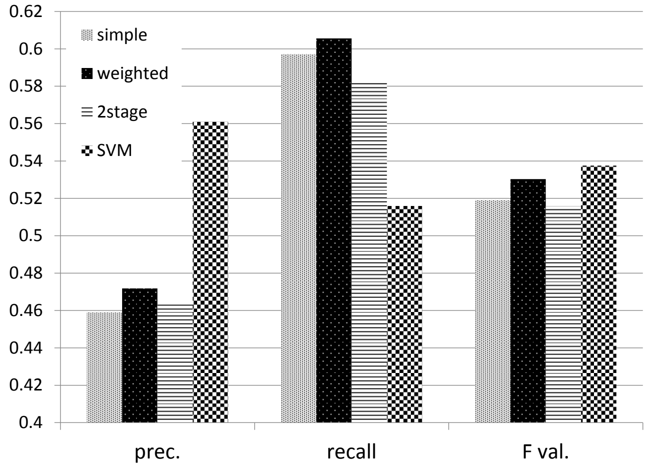

Fig. 2: Simple 5-fold cross validation (graph)

3) The learning data is the rest of them. Thus, there are $\frac{4}{5}$ of all tweets.

By this validation, there are at least $\frac{1}{5}$ tweets of one article in the learning data. Thus, any classifiers can obtain trends of every article. The recall is higher than the precision by the SVM, on the other hand, the precision is higher than the recall by naive Bayes based methods. The F value is almost 0.51 to 0.53 but the SVM has the highest performance and weighted naive Bayes has the second performance.

4.2.2 Cross validation among news articles

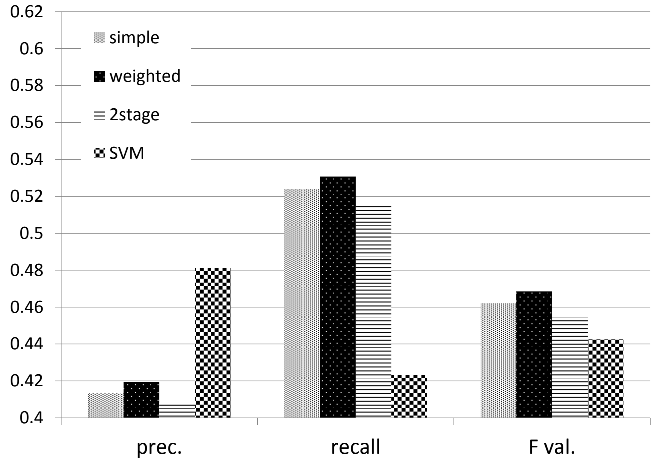

Fig. 3: Leave one file out (graph)

Fig. 3 and Table 2 show the results by leave one file out cross validation. The learning data and evaluation data are made by as follows.

Table 2: Leave one file out

	simple	weighted	2stage	SVM
precision	0.4133	0.4193	0.4079	0.4810
recall	0.5238	0.5307	0.5147	0.4231
F value	0.4620	0.4685	0.4551	0.4425

1) The evaluation data is all tweets of one article.
2) The learning data is all tweets of the all rest articles.

By the SVM, the precision is higher than its recall in this validation. The recall is higher than the precision by naive Bayes based methods. From these facts, the SVM tends to label less than naive Bayes methods. The F value of all naive Bayes based methods are higher than that of the SVM. We think this is caused that the SVM labels few emotions then one miss label decreases the F value comparing to naive Bayes based methods.

4.2.3 Closed data test

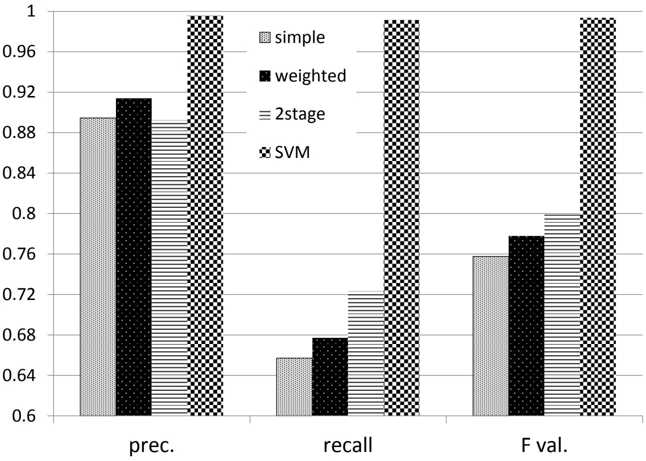

Fig. 4: Closed test (graph)

Table 3: Closed test

	simple	weighted	2stage	SVM
precision	0.8944	0.9139	0.8924	0.9957
recall	0.6569	0.6771	0.7231	0.9914
F value	0.7575	0.7779	0.7989	0.9936

Fig. 4 and Table 3 show the result of a closed test. In this test, all data are used for both learning and evaluation. The SVM scores almost 1.0 for the precision, the recall and the F value. From the previous two open data experiments, the F value of the SVM is higher than that of naive Bayes based method if learning data contain the trends of evaluation data. This trend is clear in the closed test.

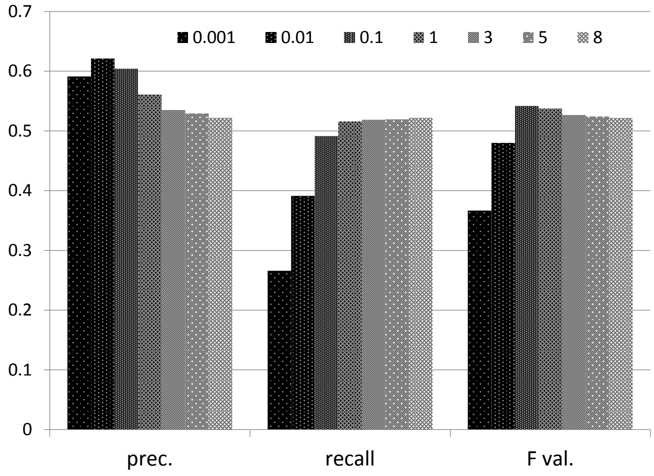

Fig. 5: Slack variable cost and performance (5-fold)

Naive Bayes based methods has different behavior against the previous two experiments. The precision is higher than the recall.

4.3 Soft margin cost on SVM

We show the difference between C values. SVM finds the hyperplane which has the maximum margin. This task is the optimization problem to maximize the function $L(w)$ where w is the normal vector of the hyperplane. Now, the total of slack variables is denoted by S. Including slack variables, the optimization function is $L(w) + C \cdot S$ where C is the cost of soft margins. We investigate the performance when C value is changed.

Table 4: Slack variable cost and performance (5-fold)

cost	0.001	0.01	0.1	
precision	0.5911	0.6212	0.6041	
recall	0.2659	0.3913	0.4914	
F value	0.3666	0.4800	0.5419	
cost	1.0	3.0	5.0	8.0
precision	0.5610	0.5350	0.5292	0.5218
recall	0.5159	0.5186	0.5194	0.5218
F value	0.5375	0.5266	0.5242	0.5217

Table 5: Slack variable cost and performance (leave one file out)

cost	0.001	0.01	0.1	
precision	0.5453	0.5457	0.5195	
recall	0.2189	0.2998	0.3905	
F value	0.3071	0.3767	0.4370	
cost	1.0	3.0	5.0	8.0
precision	0.4810	0.4620	0.4616	0.4462
recall	0.4231	0.4288	0.4354	0.4314
F value	0.4425	0.4367	0.4394	0.4300

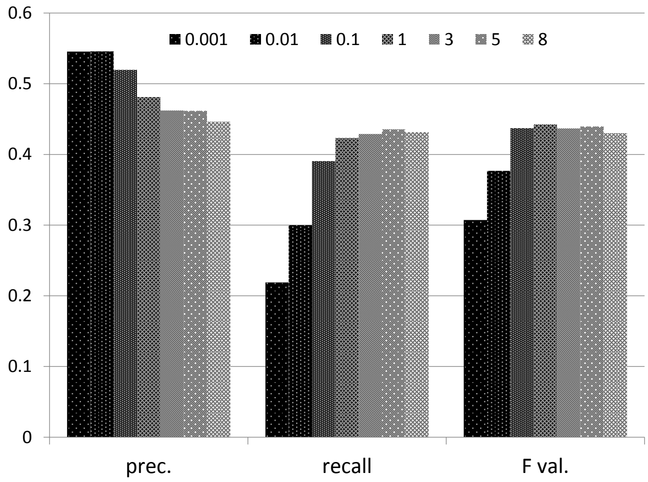

Fig. 6: Slack variable cost and performance (leave one file out)

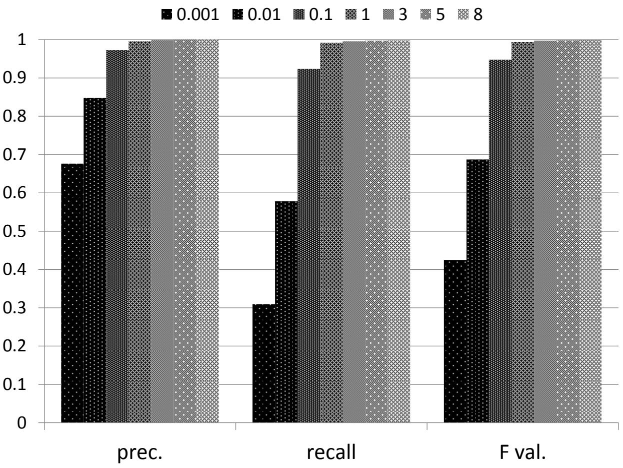

Fig. 7: Slack variable cost and performance (closed)

Fig. 5, 6 and 7 are the performances of the SVM by the 5-fold test, leave one file out and closed test, respectively. Table 4, 5 and 6 shows the values of the performance.

On the open data test (5-fold and leave one file out), the F value is the highest at $C = 1.0$. When the soft margin cost C is decreased, i.e. soft margins can be useable lightly, the recall value is rapidly decreased. On the other hand, the precision is not decreased so fast. This means that the hyperplane will be placed far from the center of learning vectors when the large soft margins are allowed. Then, the classifier tends to labels many emotions. From these figures and tables, $C = 1.0$ leads the best performance for our experiment.

Table 6: Slack variable cost and performance (closed)

cost	0.001	0.01	0.1	
precision	0.6763	0.8476	0.9726	
recall	0.3091	0.5779	0.9234	
F value	0.4242	0.6872	0.9474	
cost	1.0	3.0	5.0	8.0
precision	0.9957	0.9986	0.9996	0.9997
recall	0.9914	0.9955	0.9969	0.9979
F value	0.9936	0.9970	0.9983	0.9988

5. Conclusions

We introduced two naive Bayes based method for emotion estimation of tweets which are appended as comments to news articles. The new method uses the fact that the emotions can be classified into four classes and each of them consists of the two opposite emotions. The new methods are called the “weighted naive Bayes” and the “two stage naive Bayes”.

Then, we compared their performances with the simple naive Bayes method and the SVM by the evaluation experiments. From these results, the SVM marks high performance when the learning data contains the trends of the evaluation data. Naive Bayes based methods have robustness to learning settings. The weighted naive Bayes is the best performance among naive Bayes based methods. The performance of this method marked about 5.5% more than that of the SVM in leave one file out test, but 1.3% less in the simple cross validation.

For the future study, decision of Th and Tc are important problem to use our methods. Since two stage naive Bayes uses both of Th and Tc, effective threshold decision method is more important for this classifier. Any other classifier can be applied at every stage of the two stage naive Bayes method. This problem is also remained for the future study.

In this paper, news article body has not been used. If we can make some bias of emotion distribution, it will contribute to the performance of our methods.

Acknowledgment

This study was partially supported by JSPS KAKENHI Grant Number 24500296.

References

[1] A. Kurozumi, Y. Murakami, M. Tokuhisa, J. Murakami, S. Ikebara, “Semantic analysis of emotional verbs in valency pattern dictionary”, Proc. of the 2006 IEICE Society Conference, A-13-1, 2006.

[2] R. Tokuhisa, K. Inui, Y. Matsumoto, “Emotion classification using massive examples extracted from the Web”, IPSJ Journal, vol.50, no.4, pp.1365–1374, 2009.

[3] T. Ogawa, K. Matsumoto, F. Ren, “About emotion estimation of “Emonyu” short sentence”, IPSJ SIG Technical Report, vol.2010-NL0195, no.2, pp.1–6, 2010.

[4] Y. Tajima, “Emotion Estimation of multilabeled comments on web news sites”, IPSJ SIG Technical Report, vol.2013-MPS-95, no.18, pp.1–6, 2013.

[5] R. Plutchik, “The emotions”, University Press of America, Lanham MD, 1962.