レビューに対する評価指標の自動付与

岡野原大輔[†]・辻井　潤一[†,††,†††]

本論文では，ある対象を評価している文章（レビュー）が与えられた時，対象物に対する評価が「良い」か「悪い」かでレビューを二値分類するのではなく，どの程度「良い」か「悪い」かの指標（sentiment polarity score （SP score））をレビューに与える新しいタスクを提案する．SP score はレビューの簡潔な要約であり，単純な「良い」か「悪い」かの二値分類より詳細な情報を与える．このタスクの難しさは連続した量である SP score をどのようにしてレビューから得られるかにある．本稿では support vector regression を用いて SP score を求める方法を提案する．5 段階評価がついた本に対するレビューを用いた実験で，我々の手法が support vector machines を用いた多値分類より高い精度であり，人による指標の予測結果に近いことを示す．また，Naive Bayes Classifier を用いた文単位での主観性分析を用いることにより我々の手法の頑健性が増すことを示す．

キーワード：評判分析，文章分類，機械学習

Assigning Polarity Scores to Reviews Using Machine Learning Techniques

Daisuke Okanohara[†] and Jun'ichi Tsujii[†,††,†††]

We propose a novel type of document classification task that quantifies how much a given document (review) appreciates the target object by using a continuous measure called *sentiment polarity score* (SP score) rather than binary polarity (*good* or *bad*). An SP score gives a concise summary of a review, and provides more information than binary classification. The difficulty of this task lies in the quantification of polarity. In this paper we use support vector regression (SVR) to tackle this problem. Experiments on book reviews using five-point scales show that SVR outperforms a multi-class classification method using support vector machines, and the results are close to human performance. We also examine the effect of sentence subjectivity detection using a Naive Bayes classifier, and show that this improves the robustness of the classifier.

Key Words: *Sentiment Analysis, Document Classification, Machine Learning*

[†] 東京大学情報理工学系研究科コンピュータ科学専攻, Department of Computer Science, University of Tokyo
[††] School of Computer Science, University of Manchester
[†††] NaCTeM (National Center for Text Mining), University of Manchester

1 Introduction

In recent years, discussion groups, online shops, and blog systems on the Internet have gained popularity, and the number of documents, such as reviews, is growing dramatically. *Sentiment classification* refers to classifying reviews not by their topics but by the polarity of their sentiment (e.g, positive or negative). It is useful for recommendation systems, fine-grained information retrieval systems, and business applications that collect opinions about a commercial product.

Recently, sentiment classification has been actively studied and experimental results have shown that machine learning approaches perform well (Pang, Lee, and Vaithyanathan 2002; Pang and Lee 2004; Mullen and Collier 2004; Turney 2002). The present study asserts, however, that the polarity of reviews can be estimated more precisely than is possible with existing classification techniques. For example, both reviews A and B in Table 1 would be classified simply as *positive* in binary classification, but clearly this classification loses information concerning the difference in the degree of polarity that is apparent in the review texts.

We propose a novel type of document classification task where we evaluate reviews with scores, such as selecting from a scale of one to five stars for example. We call this score the *sentiment polarity score* (SP score). If, for example, the range of the score is from one to five, we could give five to review A and four to review B. This task, namely, ordered multi-class classification, is considered as an extension of binary sentiment classification[1]. In ordered multi-class classification, the classes are not independent, but are ordered. While it is possible to treat this problem as a multi-class classification task ignoring the order information, the performance of the classifier can be improved by incorporating this information into the classifier.

In this paper, we describe a machine learning method for this task. Our system uses support

Table 1 Examples of book reviews.

	Example of Review	binary	SP score (1,...,5)
Review A	I believe this is very good and a "must read" I can't wait to read the next book in the series.	plus	5
Review B	This book is not so bad. You may find some interesting points in the book.	plus	4

[1] Ordinal regression (R. Herbrich 1999; Herbrich, Graepel, and Obermayer 2000; Wei Chu 2005) is another formulation of similar task which solves the problem of predicting variables of ordinal scale. We discuss the difference of these problems in Section 2.

vector regression (SVR) (Vapnik 1995) to determine the SP scores of reviews. This method enables us to annotate SP scores to arbitrary reviews, such as comments in bulletin board systems or blog systems. We explore several types of features beyond a bag-of-words to capture key phrases to determine SP scores: n-grams and references (the words around the reviewed object).

In addition, our system determines the subjectivity of each sentence using a Naive Bayes classifier, since a review includes many irrelevant sentences. This approach performs well when training data includes many irrelevant sentences, but may lead to a reduction in the classifier's accuracy. This is because Naive Bayes classification cannot correctly classify subjective sentences completely, while objective sentences can contain information that is useful in determining SP scores. We show that this problem can be overcome, however, simply by adding a constant factor to the Naive Bayes estimation.

We conducted experiments with book reviews from *amazon.com*, each of which had a five-point scale rating along with text. We compared pairwise support vector machines (pSVMs) and SVR and found that SVR outperformed better than pSVMs by about 30% in terms of the squared error, which is close to human performance. We also demonstrated that the detection of sentence subjectivity by using a Naive Bayes classifier improved the robustness of the classifier.

2 Related Work

Recent studies on sentiment classification focused on machine learning approaches. Pang (Pang et al. 2002) represents a review as a feature vector and estimates polarity using SVM, which is almost the same method as those used for topic classification (Joachims 2002). This paper essentially follows this work, but we extend this task to an ordered multi-class classification task.

There have been many attempts to analyze reviews to a deeper level in order to improve accuracy. Mullen (Mullen and Collier 2004) used features from various information sources such as references to the "work" or "artist", which were annotated by hand, and showed that these features have the potential to improve accuracy. We use reference features given by the words around the fixed review target word ("book").

Turney (Turney 2002) used *semantic orientation*, which measures the distance from phrases to "excellent" or "poor" by using search engine results and gives the word polarity. Kudo (Kudo and Matsumoto 2004) used decision stumps to capture substructures embedded in text (such as word-based dependency), and suggested that subtree features are important for opinion/modality classification.

Independently of and in parallel with our work, two other papers consider the degree of polarity

for the purposes of sentiment classification. Koppel (Koppel and Schler 2006) exploited a neutral class and applied a regression method similar to that of the present study. Pang (Pang and Lee 2005) applied a metric labeling method for the task in which similar reviews tend to have same polarities. Our work differs from these two studies in several respects. In the present study evaluation was carried out by exploiting square errors rather than precision errors, with a five-point scoring scale used in the experiments, in contrast to Koppel (Koppel and Schler 2006), who used three ("good", "bad", "neutral"), and Pang (Pang and Lee 2005), who used three/four point scores. Therefore we use regression which minimize not a precision error but a square error. We argue that the precision errors are not enough to capture the task. Because if we use the precision errors mistakes of assigning 5 SP score to a review whose correct SP score is 1 can occur many times, which becomes unacceptable problem in real applications. We also examine various features to capture the characteristics of reviews, which are found to be effective in experiences.

3 Analyzing Reviews with Polarity Scores

In this section we present a novel task setting where we predict the degree of sentiment polarity of a review. We first define SP scores and the task of assigning them to review documents. We then describe the present evaluation data set. Using this data set, we examined the performance of human classifiers on this task, to clarify the difficulty of quantifying polarity.

3.1 Sentiment Polarity Scores

We extend the sentiment classification task to the more challenging task of assigning rating scores to reviews. We call this score the SP score. Examples of SP scores include *five-star* scales and *scores out of 100.* Let SP scores take discrete values[2] in a closed interval $[min...max]$. The task is to assign correct SP scores to unseen reviews as accurately as possible. Let $\hat{y}$ be the predicted SP score and y be the SP score assigned by the reviewer. We measure the performance of an estimator with the mean square error,

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2, \tag{1}$$

where $(x_1, y_1), ..., (x_n, y_n)$ is the test set of reviews. In contrast to conventional multi-class classification, which gives equal penalties to all mistakes, penalties for the present estimator are larger

[2] We could allow SP scores to have continuous values. However, in this paper we assume SP scores take only discrete values since the evaluation data set was annotated only with discrete values.

when the mistake in predicted SP score is large.

Ordinal regression (R. Herbrich 1999; Herbrich et al. 2000; Wei Chu 2005) is another framework to predict variables of ordinal scale. Since our task setting gives a large penalty for large mistake in SP score, the regression approach is more suitable for the task than ordinal regression which considers only mistakes for order of SP score (R. Herbrich 1999; Herbrich et al. 2000). We also note the efficiency of training. In several methods of ordinal regression (Herbrich et al. 2000) the problem size is a quadratic function of the training data size which is not accptable for trining large data. Our method uses well-studied formulations (SVMs, SVR) and can employ efficient algorithms and softwares.

3.2 Evaluation Data

We used book reviews on *amazon.com* for evaluation data[3,4]. Each review has stars assigned by the reviewer, with the number of stars ranging from one to five, where one is the worst score, while five is the best. We converted the number of stars into SP scores $\{1, 2, 3, 4, 5\}$[5]. Although each review may include several paragraphs, we did not exploit paragraph information.

From these data, we made two data sets. The first was a set of reviews for books in the *Harry Potter* series (Corpus A). The second was a set of reviews for books of arbitrary kinds (Corpus B). It was easier to predict SP scores for Corpus A than Corpus B because Corpus A books have a smaller vocabulary and each review was about twice as large. To create a data set with a uniform score distribution (the effect of skewed class distributions is out of the scope of this paper), we selected 330 reviews per SP score for Corpus A and 280 reviews per SP score for Corpus B[6]. Table 2 shows the number of words and sentences in the corpora. There is no significant difference in the average number of words/sentences among different SP scores.

3.3 Preliminary Experiments: Human Performance for Assigning SP scores

We treat the SP scores assigned by the reviewers as correct answers. However, the content of a review and its SP score may not be related. Moreover, SP scores may vary depending on the reviewers. Accordingly, we examined the universality of the SP score.

[3] *http://www.amazon.com*

[4] These data were gathered from google cache using google API.

[5] One must be aware that different scales may reflect the different reactions than just scales as Keller indicated (Sorace and Keller 2005).

[6] We actually corrected 25000 reviews. However, we used only 2900 reviews since the number of reviews with 1 star is very small. We examined the effect of the number of training data, as is discussed in Section 5.4.

Table 2 Corpus A: reviews for a Harry Potter series book. Corpus B: reviews for all kinds of books. The *words* column shows the average number of words in a review, while the *sentences* column shows the average number of sentences in a review.

SP score	Corpus A			Corpus B		
	review	words	sentences	review	words	sentences
1	330	160.0	9.1	250	91.9	5.1
2	330	196.0	11.0	250	105.2	5.2
3	330	169.1	9.2	250	118.6	6.0
4	330	150.2	8.6	250	123.2	6.1
5	330	153.8	8.9	250	124.8	6.1

Table 3 Human performance of SP score estimation. Test data: 100 reviews of Corpus A with 1,2,3,4,5 SP scores.

		Square error
	Human A	0.77
	Human B	0.79
	Human average	0.78
cf.	Random	3.20
	All3	2.00

We asked two computational linguists to independently assign an SP score to each review from Corpus A. These two linguists first learned the relationship between reviews and SP scores using 20 reviews, and were then given 100 reviews with a uniform SP score distribution as test data. Table 3 shows the results given in terms of the mean square error. The *Random* row shows the performance achieved by random assignment, and the *All3* row shows the performance achieved by assigning 3 to all the reviews. These results suggest that SP scores would be estimated solely from the contents of reviews with a square error of 0.78.

Table 4 shows the distribution of the estimated SP scores and correct SP scores. In the table we can observe the difficulty of this task; the precise quantification of SP scores. For example, it can be seen from the table that human B tended to overestimate SP scores for reviews whose correct scores were in the range between 2 and 4, assigning a 1 or 5. We should note that if we consider this task as binary classification by treating the reviews whose SP scores are 4 and 5 as positive examples and those with 1 and 2 as negative examples (ignoring the reviews whose SP scores are 3), the classification precisions by humans A and B are 95% and 96% respectively.

Table 4 Results of SP score estimation: Human A (left) and Human B (right).

	Assigned					
	1	2	3	4	5	Total
Correct						
1	12	7	0	1	0	20
2	7	8	4	1	0	20
3	1	1	13	5	0	20
4	0	0	4	10	6	20
5	0	1	2	7	10	20
Total	20	17	23	24	16	100

	Assigned					
	1	2	3	4	5	total
Correct						
1	16	3	0	1	0	20
2	11	5	3	1	0	20
3	2	5	7	4	2	20
4	0	1	2	1	16	20
5	0	0	0	2	18	20
Total	29	14	12	9	36	100

4 Assigning SP scores to Reviews

This section describes a machine learning approach to predict the SP scores of review documents. Our method consists of the following two steps: extraction of feature vectors from reviews, and estimation of SP scores from these feature vectors. The first step basically uses existing techniques for document classification. In contrast, the prediction of SP scores is different from previous studies because we consider ordered multi-class classification, that is, each SP score has its own class and the classes are ordered. Unlike usual multi-class classification, large mistakes in terms of the order should have large penalties. In this paper, we discuss two methods of estimating SP scores: pSVMs and SVR.

4.1 Review Representation

We represent a review as a feature vector. Although this representation ignores the syntactic structure, word positions, and the order of words, it is known to work reasonably well for many tasks such as information retrieval and document classification. We use *binary*, *tf*, and *tf-idf* as feature weighting methods (Sebastiani 2002). The feature vectors are normalized to have L^2 norm 1.

4.2 Support Vector Regression

Support vector regression (SVR) is a method of regression that follows a similar underlying idea to that of SVM (Cristianini and Taylor 2000; Smola and Sch 1998). SVR predicts the SP score of a review by the following regression:

$$g : R^n \mapsto R, y = g(x) = \langle w \cdot x \rangle + b. \tag{2}$$

where y is the predicted SP score, x is the feature vector of a review, w and b are parameters of SVR. SVR uses an *ϵ-insensitive loss function.* This loss function means that all errors inside an ϵ cube are ignored. This allows SVR to require only a few support vectors, and gives a generalization ability. Given a training set, $(x_1, y_1), ..., (x_n, y_n)$, parameters w and b are determined by solving the following problem,

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\langle w \cdot w \rangle + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ \text{subject to} \quad & (\langle w \cdot x_i \rangle + b) - y_i \leq \epsilon + \xi_i \\ & y_i - (\langle w \cdot x_i \rangle + b) \leq \epsilon + \xi_i^* \\ & \xi_i^{(*)} \geq 0 \qquad i = 1, ..., n. \end{aligned} \tag{3}$$

The factor $C > 0$ is a parameter that controls the trade-off between training error minimization and margin maximization. The loss in training data increases as C becomes smaller, while generalization is lost as C becomes larger. Moreover, we can apply a kernel-trick to SVR, as in the case for SVMs, by using a kernel function.

This approach captures the order of classes and does not suffer from data sparseness. While we could use conventional linear regression instead of SVR (Hastie, Tibshirani, and Friedman 2001), in the present study we use SVR because it can exploit the kernel-trick and avoid over-training. Another good characteristic of SVR is that we can identify the features contributing to determining the SP scores by examining the coefficients (w in (2)), while pSVMs do not give such information, because multiple classifiers are involved in determining final results. A difficulty associated with the present approach, however, is that it is difficult to learn non-linear regression by SVR. For example, when given training data is $(x = 1, y = 1), (x = 2, y = 2), (x = 3, y = 8)$, SVR cannot perform regression correctly without adjusting the input space (feature values) so that the output plane becomes linear-one. Note that this problem does not occur in classification problems, but in regression problems. We can solve this problem by choosing an appropriate kernel for the task, but this selection is not straightforward.

4.3 Pairwise Support Vector Machines

We apply a multi-class classification approach to estimating SP scores. pSVMs (Kresel 1999) consider each SP score as a unique class, ignoring the order among the classes. Given reviews with SP scores $\{1, 2, .., m\}$, we construct $m \cdot (m-1)/2$ SVM classifiers for all the pairs of possible values of SP scores. The classifier for an SP score pair (a vs b) assigns the SP score to a review with a or b. The class label of a document is determined by majority voting of the classifiers.

Any ties in the voting are resolved by choosing the class that is closest to the neutral SP score (i.e, $(1+m)/2$).

This approach ignores the fact that SP scores are ordered, which causes the following two problems: First, it allows large mistakes. Second, when the number of possible values of the SP score is large (e.g, $n > 100$), this approach suffers from a data sparseness problem. This is because pSVMs cannot employ examples that have close SP scores (e.g, SP score = 50) for the classification of other SP scores (e.g, the classifier for a SP score pair (51 *vs* 100)).

4.4 Features beyond Bag-of-Words

Previous studies (Lewis 1992; Apte, Damerau, and Weiss 1994) suggested that complex features do not work as expected because data becomes sparse when such features are used, and a bag-of-words approach is sufficient to capture the information in most reviews. Nevertheless, we observed that reviews include many chunks of words such as "very good" or "must buy" that are useful for estimating the degree of polarity. We confirmed this observation by using n-grams.

Since the words around the review target might be expected to influence the overall SP score more than other words, we use these words as features. We call these features *reference*. We assume the review target is only the word "book", and we use "inbook" and "aroundbook" features. The "inbook" feature are the words appearing in the sentence which includes the word "book". The "around book" feature is given by the words lying within two places either side of the word "book". Table 5 summarizes the list of features for the experiments.

4.5 Identification of Subjectivity Sentences

A review document includes many sentences that are irrelevant to sentiment polarity of the document, such as explanation of a reviewed object or objective sentences. There exist some methods for detecting subjective sentences by a knowledge-based approach or machine learning (Hong and Hatzivassiloglou 2003; Pang and Lee 2004).

Table 5 List of features for the experiments.

Features	Description	Example in Fig.1 review 1
unigram	single word	(I) (believe) .. (series)
bigram	pair of two adjacent words	(I believe) ... (the series)
trigram	adjacent three words	(I believe this) ... (in the series)
inbook	words in a sentence including "book"	(I) (can't) ... (series)
aroundbook	words near "book" within two words.	(the) (next) (in) (the)

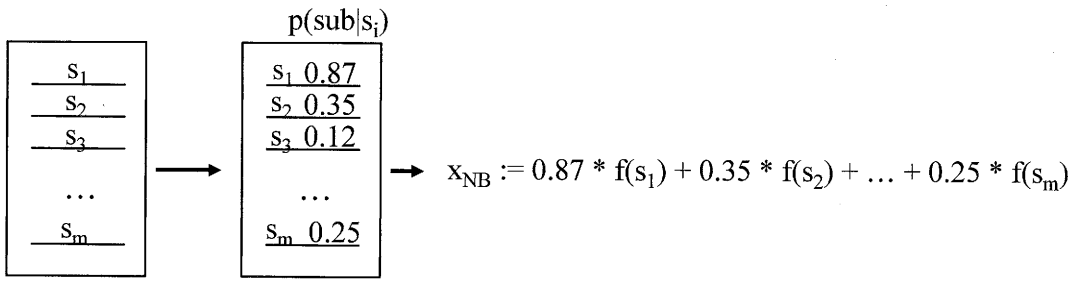

Fig. 1 An example of a feature vector for a review using subjectivity information. Every sentence is assigned a subjectivity probability: $p(sub|s_i)$ using a Naive Bayes classifier. The feature vector for a review : x_{NB} is obtained by summing up the feature vector for each sentence ($f(s_i)$) weighted by a subjectivity probability ($p(sub|s_i)$). The formal definition is (14).

Here, we propose a method for estimating the probability that a given sentence is subjective using Naive Bayes classifiers (further details and other Naive Bayes models can be found in (Nigam, McCallum, Thrun, and Mitchell 2000; McCallum and Nigam 1998)). Figure 1 shows an overview of our approach, that is, we assign the probability that a given sentence in review is subjective, and then weight each feature using this probability.

Although it is hard to obtain a corpus in which individual sentences are annotated with whether the sentence is objective or subjective, we can obtain documents consisting of subjective or objective sentences only (Pang and Lee 2004). Using these documents, we construct a Naive Bayes classifier to estimate the probability of each sentence's subjectivity.

We assign a class which may be either *subjective* (sub) or *objective* (obj) to each sentence (s_i). We use a multinomial Naive Bayes Classier to estimate the probability sentence subjectivity $p(sub|s_i)$,

$$p(sub|s_i) = \frac{p(sub)p(s_i|sub)}{p(s_i)} = \frac{p(sub)p(s_i|sub)}{p(sub)p(s_i|sub) + p(obj)p(s_i|obj)}. \quad (4)$$

We decompose $p(s_i|sub)$ and $p(s_i|obj)$ into the probability of $p(w_t|sub)$ and $p(w_t|obj), w_t \in s_i$,

$$p(s_i|sub) = C \prod_{t=1}^{|s_i|} p(w_t|sub), \quad (5)$$

$$p(s_i|obj) = C \prod_{t=1}^{|s_i|} p(w_t|obj), \quad (6)$$

$$C = P(|s_i|)|s_i|! \prod_{t=1}^{|V|} \frac{1}{|\{w_j | w_j = w_t, w_j \in s_i\}|!}. \tag{7}$$

We substitute expressions (5) and (6) into equation (4), and obtain

$$p(sub|s_i) = \frac{p(sub) \prod_{t=1}^{|s_i|} p(w_t|sub)}{p(sub) \prod_{t=1}^{|s_i|} p(w_t|sub) + p(obj) \prod_{t=1}^{|s_i|} p(w_t|obj)}. \tag{8}$$

We estimate $p(w_t|sub)$, $p(w_t|obj)$, $p(sub)$, $p(obj)$ from training data. The training data consists of subjective sentences $\{s_1, s_2, .., s_m\}$ and objective sentences $\{\hat{s_1}, \hat{s_2}, ..., \hat{s_m}\}$. Let $c_{sub}(w_t)$ be the frequency of the word w_t in the subjective corpus, and $c_{obj}(w_t)$ be the frequency of the word w_t in the objective corpus, then the above parameters are given by

$$p(w_t|sub) = \frac{p(w_t)p(sub|w_t)}{p(sub)} \tag{9}$$

$$= \frac{1 + c_{sub}(w_t)}{|V| + \sum_{s=1}^{V} (c_{sub}(w_s))}, \tag{10}$$

$$p(w_t|obj) = \frac{1 + c_{obj}(w_t)}{|V| + \sum_{s=1}^{V} (c_{obj}(w_s))}, \tag{11}$$

$$p(sub) = p(obj) = \frac{1}{2}. \tag{12}$$

We use Laplacian smoothing for estimating $p(w_t|sub), p(w_t|obj)$. Using $p(sub|s_i)$, we weight each sentence.

Using the probability $p(sub|s_i)$, we recalculate the value of each feature. The disadvantage of this approach is that the information of the training data becomes small in comparison with the original data, because the estimation of Naive Bayes classification tends to overestimate the probability. For example, even if the true probability is close to 0.5, the result of NB would be 0 or 1. Futhermore, the objective sentences may have information that could be useful in determining the SP scores. We therefore introduce a smoothing factor α, which means that we assignes probability from α to 1 (by ignoring the normalization factor). The feature vectors for a review are calculated as follows: the first (x) is the original feature vector, the second x_{NB} is a feature vector weighted by the probability that the particular sentence is subjective, while the

third (x_{NBS}) is a feature vector which is smoothed by using the pre-defined smoothing factor (α),

$$x := \sum_{i=1}^{m}(f(s_i)), \tag{13}$$

$$x_{NB} := \sum_{i=1}^{m}(p(sub|s_i)f(s_i)), \tag{14}$$

$$x_{NBS} := \sum_{i=1}^{m}((p(sub|s_i) + \alpha)f(s_i)), \tag{15}$$

where $f(s_i)$ is the feature vector for the sentence s_i.

The feature values in x_{NB} and x_{NBS} can be considered to be the expected feature values in a subjective sentence. We could alternatively adopt an approach (Pang and Lee 2004) whereby first the objective sentences are eliminated, and then we solve the problem as before. This approach would be faster than our approach since the training data is smaller than the original training data set. However, the results of this approach would be less accurate than our approach using the feature vectors x and x_{NB}, since these vectors can be seen as an approximation of feature values. However, clearly we can tradeoff the speed and accuracy of the classifier by the choice of approach, and we plan to investigate this further as part of future work.

5 Experiments

We performed two series of experiments. First, we compared pSVMs and SVR and examined the performance of various features and weighting methods. Second, we compared the method using sentence subjectivity detection with the method which does not.

The corpora A and B introduced in Section 3.2 were used as the experimental data. We first removed all HTML tags and punctuation marks, and then applied the Porter stemming method (Porter 1980) to the reviews.

We divided the data into ten disjoint subsets, maintaining the uniform class distribution. All the results reported below are the averages of ten-fold cross-validation. In SVMs and SVR, we used *SVMlight*[7] with the quadratic polynomial kernel $K(x, z) = (\langle x \cdot z \rangle + 1)^2$ and set the control parameter C to 100 in all the experiments.

For sentence subjectivity detection, we used Pang's sentence corpus version 1.0[8] (Pang and

[7] http://svmlight.joachims.org/
[8] http://www.cs.cornel.edu/People/pabo/movie-review-data/

Lee 2004).

5.1 Comparison of pSVMs and SVR

We compared pSVMs and SVR to see differences in the properties of the regression approach compared with those of the classification approach. Both pSVMs and SVR used unigram/tf-idf to represent reviews. Table 6 shows the square error results for SVM, SVR and a simple regression (least square error) method for Corpus A/B. These results indicate that SVR outperformed SVM in terms of the square error and suggests that regression methods avoid large mistakes by taking account of the fact that SP scores are ordered, while pSVMs does not. We also note that the result of a simple regression method is close to the result of SVR with a linear kernel.

Figure 2 shows the distribution of estimation results for humans (top left: human A, top right: human B), pSVMs (below left), and SVR (below right). In all the plots the horizontal axes show the estimated SP scores, the vertical axes show the correct SP scores, while shading indicates the number of reviews. These figures suggest that pSVMs and SVR were able to capture the gradualness of SP scores better than the human classifiers. They also show that pSVMs cannot predict neutral SP scores well, whereas SVR accurately predicts these scores.

5.2 Comparison of Different Features

We compared the different features presented in Section 4.4 and feature weighting methods. First we compared different weighting methods, using only unigram features for this comparison. We then compared different features, using only tf-idf weighting methods for this comparison.

Table 7 summarizes the comparison results of different feature weighting methods. The results show that *tf-idf* performed well on both test corpora. We should note that simple representation methods, such as *binary* or *tf*, give comparable results to tf-idf, which indicates that we can add more complex features without considering the scale of feature values. For example, when we add word-based dependency features, we have some difficulty in adjusting these feature values to

Table 6 Comparison of multi-class SVM and SVR. Both use unigram/tf-idf.

	Mean Square error	
Method	Corpus A	Corpus B
pSVMs	1.32	2.13
simple regression	1.05	1.49
SVR (linear kernel)	1.01	1.46
SVR (polynomial kernel $(\langle x \cdot z \rangle + 1)^2$)	**0.94**	**1.38**

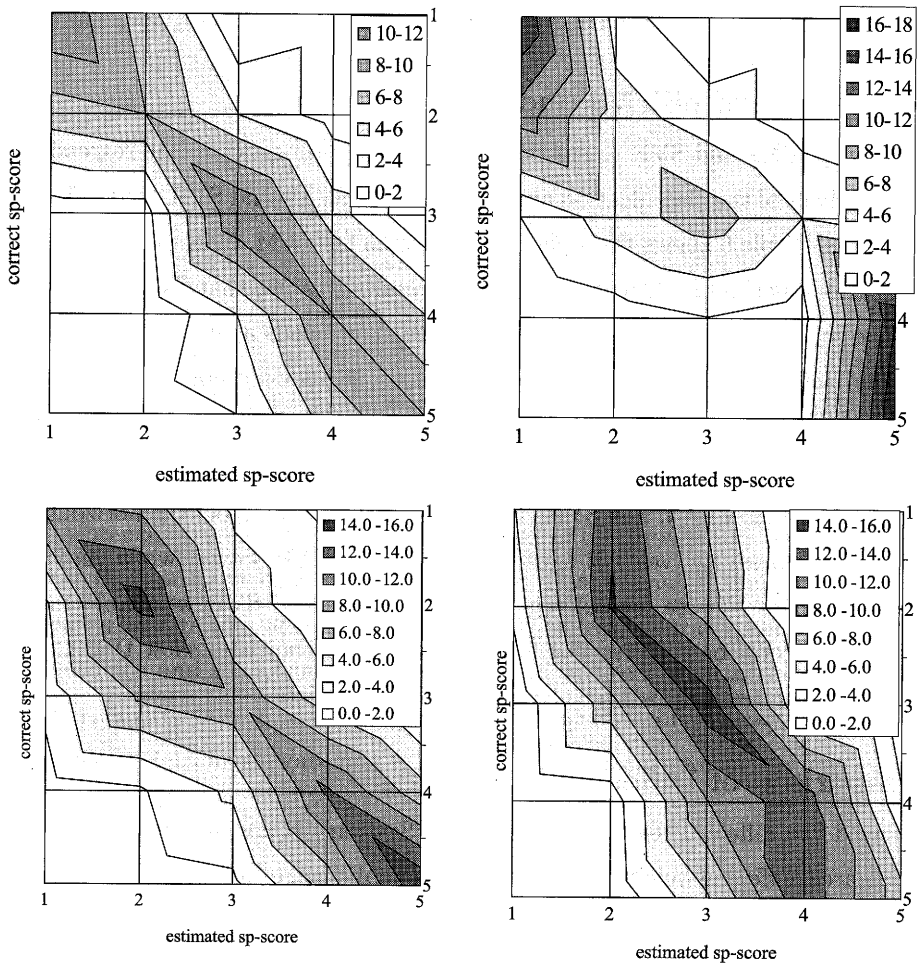

Fig. 2 Distribution of estimation results. Shading indicates the number of reviews. Top left: Human A, top right: Human B, below left: pSVMs, below right: SVR

those of unigrams. However, we could use these features together in binary weighting methods.

Table 8 summarizes the comparison results for different features. For Corpus A, *unigram + bigram* and *unigram + trigram* achieved high performance. The performance of *unigram + inbook* does not achieve as good a performance as expected, contrary to our intuitive belief that the words around the target object are more important than others. However, for Corpus B, the results are less accurate, that is, n-gram features were less able to accurately predict the SP scores. This is because the variety of words/phrases was much larger than in Corpus A, and n-gram features may have suffered from a data sparseness problem. We should note that these feature settings are too simple, and we cannot accept the result of reference or target object

Table 7 Results comparing different feature weighting methods. We used unigrams as the features of the reviews.

	Square error	
Weighting methods (unigram)	Corpus A	Corpus B
tf	1.03	1.49
tf-idf	**0.94**	**1.38**
binary	1.04	1.47

Table 8 Results comparing different features. For the comparison of different features we use tf-idf as the weighting method.

	Square error	
Feature (tf-idf)	Corpus A	Corpus B
unigram (baseline)	0.94	1.38
unigram + bigram	**0.89**	1.41
unigram + trigram	0.90	1.42
unigram + inbook	0.97	**1.36**
unigram + aroundbook	0.93	1.37

(*inbook*/*aroundbook*) directly.

Note that the data used in the preliminary experiments described in Section 3.3 are a part of Corpus A, and so we can compare the results obtained from the human classifiers with those for Corpus A in this experiment. The best result by the machine learning approach (0.89) was close to the human results (0.78).

To analyze the influence of n-gram features, we used the linear kernel $k(x, z) := \langle x \cdot z \rangle$ in SVR training. We used tf-idf as feature weighting, and examined each coefficient of regression. Since we used the linear kernel, the coefficient value of SVR showed the polarity of a single feature, that is, this value expressed how much the occurrence of a particular feature affected the SP score. Tables 9, 10, 11, 12, 13 and 14 show the coefficients resulting from the training of SVR. These results show that phrases such as "all ag (age)", "can't wait" "on (one) star" and "not interest" have strong polarity even if the word which constitutes these phrases does not have strong polarity.

5.3 Using Naive Bayes Classifier to Subjectivity Detection

We examined the effectiveness of subjectivity detection using the Naive Bayes classifier (NB) proposed in Section 4.5. First, we examined the performance of NB itself by using Pang's sentence

Table 9 List of unigram features that have the ten have best polarity values estimated by SVR in corpus A/B. The *count* column expresses the frequency of a feature in reviews. The *value* column expresses the estimated SP score of a feature, i.e., only this feature is fired in a feature vector.

Corpus A			Corpus B		
value	count	unigram	value	count	unigram
2.97	2467	read	2.68	826	read
2.49	377	love	2.60	129	best
2.35	25	philosoph's	2.48	45	knowledge
2.27	265	best	2.28	50	excel
2.17	400	know	2.25	66	highli (highly)
2.16	81	amaz (amaze)	2.24	92	differ
2.12	444	great	2.19	24	romanc (romance)
2.11	36	listen	2.18	28	common
2.01	280	friend	2.12	24	javascript
2.07	42	hook	2.12	196	great

Table 10 List of unigram features that have the ten worst polarity values estimated by SVR in corpus A/B. The *count* column expresses the frequency of a feature in reviews. The *value* column expresses the estimated SP score of a feature, i.e., only this feature is fired in a feature vector.

Corpus A			Corpus B		
value	count	unigram	value	count	unigram
-2.81	68	wast (waste)	-2.85	90	disappoint
-2.44	442	no	-2.75	69	noth (nothing)
-2.40	1537	not	-2.69	40	wast (waste)
-2.40	10	didnt	-2.45	33	worst
-2.24	437	do	-2.21	125	feel
-2.12	28	scar (scare)	-2.20	276	would
-2.08	16	unrealist	-2.11	50	bought
-2.07	238	bad	-2.06	26	terribl (terrible)
-2.05	229	someth (something)	-2.05	16	soon
-1.98	61	instal	-2.00	176	look

corpus version 1.0. The result of ten-fold cross-validation was 90.5% accuracy. A review, however, includes both subjective and objective sentences. Moreover, we have to examine whether the information of subjectivity contributes to the polarity detection. We asked two computer linguists to select the sentence which is most influential on the SP score in each review. The test data is the same as the test data used in Section 3.3. We then assigned subjectivity for each sentence using the NB. Table 15 shows the average subjectivity of all sentences and also of the most influential

Table 11 List of bigram features that have the ten best polarity values estimated by SVR in corpus A/B. The *count* column expresses the frequency of a feature in reviews. The *value* column expresses the estimated SP score of a feature, i.e., only this feature is fired in a feature vector.

Corpus A			Corpus B		
value	count	bigram	value	count	bigram
1.73	61	best book	1.64	89	the best
1.69	638	is a	1.60	84	read it
1.49	301	read it	1.37	92	a great
1.44	65	all ag (age)	1.34	119	on (one) of
1.30	32	can't wait	1.31	11	fast food
1.20	568	it is	1.22	45	harri (harry) potter
1.14	207	the sorcer's	1.19	36	highli recommend (recommend)
1.14	10	great !	1.14	34	an excel
1.13	193	sorcer's stone	1.12	151	to read
1.11	65	come out	1.01	508	in the

Table 12 List of bigram features that have the ten worst polarity values estimated by SVR in corpus A/B. The *count* column expresses the frequency of a feature in reviews. The *value* column expresses the estimated SP score of a feature, i.e., only this feature is fired in a feature vector.

Corpus A			Corpus B		
value	count	bigram	value	count	bigram
-1.61	79	at all	-1.19	17	veri (very) disappoint
-1.50	27	wast (waste) of	-1.13	15	wast (waste) of
-1.38	270	potter book	-0.98	22	the worst
-1.36	134	out of	-0.97	41	is veri (very)
-1.28	11	not interest	-0.96	36	! !
-1.18	15	on (one) star	-0.85	127	i am
-1.14	53	the worst	-0.81	40	the exampl (example)
-1.13	51	first four	-0.79	15	bui (buy) it
-1.11	22	a wast (waste)	-0.76	16	veri (very) littl (little)
-1.08	33	no on (one)	-0.74	23	onli (only) to

sentences as selected by the human classifiers. It is almost certain that subjectivity is correlated with the sentence that is most influential on the SP score.

Figure 3 shows examples of the results of sentence subjectivity detection by the NB[9]. The results

[9] These examples were not extracted from original corpus but were made by the linguist. We do not show any of the original sentences in this paper, in order to observe the copyright of the original corpus.

Table 13 List of trigram features that have the ten best polarity values estimated by SVR in corpus A/B. The *count* column expresses the frequency of a feature in reviews. The *value* column expresses the estimated SP score of a feature, i.e., only this feature is fired in a feature vector.

Corpus A			Corpus B		
value	count	trigram	value	count	trigram
1.42	51	the best book	1.43	72	on (one) of the
1.37	111	thi (this) is a	0.99	18	the best book
1.34	182	the sorcer's stone	0.91	27	read the book
1.14	38	of all ag (age)	0.89	20	is a great
1.13	51	put it down	0.83	82	thi (this) is a
1.04	307	harri (harry) potter and	0.77	18	a great book
1.02	267	thi (this) book is	0.76	43	thi (this) book to
0.97	19	i can't wait	0.74	16	of the best
0.97	59	he is a	0.70	28	is on (one) of
0.93	263	potter and the	0.70	13	thi (this) is on (one)

Table 14 List of trigram features that have the ten worst polarity values estimated by SVR in corpus A/B. The *count* column expresses the frequency of a feature in reviews. The *value* column expresses the estimated SP score of a feature, i.e., only this feature is fired in a feature vector.

Corpus A			Corpus B		
value	count	trigram	value	count	trigram
-1.22	48	the first four	-0.88	10	book is veri (very)
-1.21	245	harri (harry) potter book	-0.75	10	a wast (waste) of
-1.15	20	a wast (waste) of	-0.71	14	for thi (this) book
-0.91	74	order of the	-0.67	10	for a book
-0.87	13	the worst book	-0.60	20	a good book
-0.82	13	did not like	-0.59	45	if you want
-0.81	53	of the phoenix	-0.58	16	from thi book
-0.80	28	first four book	-0.57	88	in thi (this) book
-0.72	117	in thi (this) book	-0.55	21	the rest of
-0.71	14	wast (waste) of time	-0.52	17	you ar (are) look

suggest that the NB analyzes the subjectivity of sentences well. For instance, explanations of the plot of the novel (lines 2 in review 1) are assigned 0.00 subjectivity.

Second, we evaluated the performance of classification using NB sentence subjectivity detection. Table 16 shows the results by *baseline*(SVR + tfidf + unigram), *NB*(baseline + NB sentence subjectivity detection (Eq. 14)) and *NB with C* (baseline + NB sentence subjectivity detection with added constant $\alpha = 0.5$ (Eq. 15)). The results indicate that *NB* is better than *baseline* when

Table 15 Average subjectivity assigned by a Naive Bayes classifier. The *all* column shows the average subjectivity of all reviews. The *human*1/2 columns show the average subjectivity of sentences selected as the most influential by human classifiers A/B

	all	human1	human2
average of subjectivity	0.58	0.76	0.71

```
0.39 It is a fantasy fairytale, sometimes likened to Cinderella, about
     a young orphaned boy transported into a world of magic and
     sorcery.
0.00 Harry Potter finds himself at a school for wizards, where his
     reputation precedes him, and soon becomes embroiled in a classic
     battle of good versus evil.
0.39 From start to finish, I was hooked by this book, and couldn't put
     it down.
1.00 The pages shimmer with creativity, and although an easy read for
     adults, I would recommend it heartily to anyone that enjoys
     escaping the real world for an hour or three.
```

Fig. 3 Examples of the results of sentence subjectivity detection by a Naive Bayes Classifier. The numbers to the left of sentences show the estimated probabilities of sentence subjectivity $\in [0, 1]$.

Table 16 The results of classification with/without Naive Bayes subjectivity detection presented in terms of average square errors. The notation $\alpha \rightarrow \beta$ means training data is α and test data is β. The Square error column lists the average square errors.

	Square error			
Methods	Corpus A	Corpus B	Corpus A $\rightarrow$ B	Corpus B $\rightarrow$ A
baseline	0.94	1.38	1.83	1.93
NB	1.04	1.46	1.81	1.72
NB with C	0.95	1.39	1.80	1.72

the training data and test data are different, especially when the training data is corpus B and test data is corpus A. We suspect that when we use reviews taken from various themes as training data, some proper nouns have polarity and these words cause the classifier to be misled to the wrong polarity. In contrast, when we use reviews on a specific theme as training data, proper nouns tend to occur uniformly through all SP scores, and the effects of proper nouns on polarity scores are not overestimated. The decline of accuracy by NB in corpora A and B is probably caused by the inadequate performance of Naive Bayes classifiers or loss of useful information in

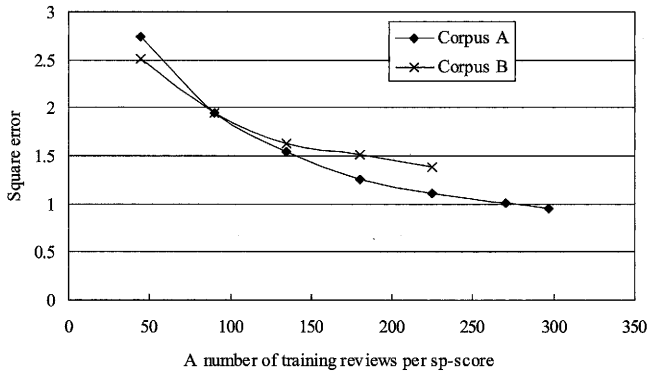

Fig. 4 Learning curve for our task setting for Corpus A and Corpus B. We used SVR as the classifier and unigram/*tf-idf* to represent of reviews.

objective sentences. *NB with C* performs well in each case, suggesting that *NB with C* has the advantages of objective sentence elimination without suffering any significant decline due to the loss of information in objective sentences.

5.4 Learning Curve

We generated learning curves to examine the effect of the size of training data on performance. Figure 4 shows the results of a classification task using unigram/*tf-idf* to represent reviews. The results suggest that performance can be improved further by increasing the training data.

6 Conclusion

In this paper, we described a novel task setting in which we predicted SP scores—degree of polarity—of reviews. We proposed a machine learning method using SVR to predict SP scores.

We compared two methods for estimating SP scores: pSVMs and SVR. Experimental results for book reviews showed that SVR performed better in terms of the square error than pSVMs by about 30%. This result agrees with our intuition that pSVMs do not consider the order of SP scores, while SVR captures the order of SP scores and avoids high penalty mistakes. With SVR, SP scores can be estimated with a square error of 0.89, which is very close to the square error achieved by human classifiers (0.78).

We examined the effectiveness of features beyond a bag-of-words and reference features (the words around the reviewed objects.) The results suggest that n-gram features and reference features contribute to improve accuracy.

The experimental results for sentence subjectivity detection using Naive Bayes classifiers showed that this approach can improve the robustness of a classifier, which may be improved further by adding a constant to the result of Naive Bayes classifiers. This is because the noise from objective sentences is eliminated.

As the next step in our research, we plan to exploit parsing results such as predicate argument structures for detecting precise reference information. As well as attitude, we will also capture other types of polarity, such as modality and writing position (Kudo and Matsumoto 2004), and we will consider the estimation of these types of polarity.

We plan to develop a classifier specialized for ordered multi-class classification using recent studies on machine learning for structured output spaces (Tsochantaridis, Hofmann, Joachims, and Altun 2004; Taskar 2004) or ordinal regression (Herbrich et al. 2000; Wei Chu 2005), since our experiments suggest that pSVMs and SVR have both advantages and disadvantages. We will develop a more efficient classifier that outperforms pSVMs and SVR by combining these ideas. We also examine whether or not our task setting is appropriate to summarize the review.

Acknowledgment

The authors would like to acknowledge the large contributions of our laboratory members to give us valuable comments and annotate reviews.

Reference

Apte, C., Damerau, F., and Weiss, S. (1994). "Automated Learning of Decision Rules for Text Categorization." *Information Systems*, **12** (3), pp. 233–251.

Cristianini, N. and Taylor, J. S. (2000). *An Introduction to Support Vector Machines and other Kernel-based Learning Methods.* Cambridge University Press.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning.* Springer.

Herbrich, R., Graepel, T., and Obermayer, K. (2000). "Large Margin Rank Boundaries for Ordinal Regression." In *Advances in Large Margin Classifiers*, pp. 115–132. MIT press.

Hong, Y. and Hatzivassiloglou, V. (2003). "Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opnion sentences." In *Proceedings of the Conference on Empirical Methods in Natunal Language Processing*, pp. 129–136.

Joachims, T. (2002). *Learning to Classify Text Using Support Vector Machines.* Kluwer.

Koppel, M. and Schler, J. (2006). "The Importance of Neutral Examples in Learning Sentiment." *Computational Intelligence*, **22** (2), pp. 100–109.

Kresel, U. (1999). *Pairwise Classification and Support Vector Machines Methods.* MIT Press.

Kudo, T. and Matsumoto, Y. (2004). "A boosting algorithm for classification of semi-structured text." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 301–308.

Lewis, D. (1992). "An evaluation of Phrasal and Clustered Representations on A Text Categorization Task." In *Proceedings of SIGIR, 15th ACM International Conference on Research and Development in Information Retrieval*, pp. 37–50.

McCallum, A. and Nigam, K. (1998). "A comparison of event models for naive bayes text classification." In *AAAI Workshop on Learning for Text Categorization*, pp. 41–48.

Mullen, A. and Collier, N. (2004). "Sentiment Analysis using Support Vector Machines with Diverse Information Sources." In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL)*, pp. 21–26.

Nigam, K., McCallum, A., Thrun, S., and Mitchell, T. (2000). "Text classification from labeled and unlabeled documents using EM." *Machine Learning*, **39** (2), pp. 103–134.

Pang, B. and Lee, L. (2004). "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts." In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL)*, pp. 271–278.

Pang, B. and Lee, L. (2005). "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales." In *Proceedings of the 43nd Meeting of the Association for Computational Linguistics (ACL)*, pp. 115–124.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). "Thumbs up? Sentiment Classification using Machine Learning Techniques." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 79–86.

Porter, M. (1980). "An algorithm for suffix stripping, Program." *Program*, **14** (3), pp. 130–137.

R. Herbrich, T. Graepel, K. O. (1999). "Regression Models for Ordinal Data: A Machine Learning Approach." Tech. rep., Report TR 99-3, Dept. of Computer Science, Technical University of Berlin.

Sebastiani, F. (2002). "Machine Learning in Automated Text Categorization." *ACM Computing Surveys*, **34** (1), pp. 1–47.

Smola, A. and Sch, B. (1998). "A tutorial on Support Vector Regression." Tech. rep., NeuroCOLT2 Technical Report NC2-TR-1998-030.

Sorace, A. and Keller, F. (2005). "Gradience in Linguistic Data." *Lingua*, **115** (11),

pp. 1497–1524.

Taskar, B. (2004). *Learning Structured Prediction Models: A Large Margin Approach.* Ph.D. thesis, Stanford University.

Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. (2004). "Support vector machine learning for interdependent and structured output spaces." In *Machine Learning, Proceedings of the Twenty-first International Conference (ICML)*, pp. 823–830.

Turney, P. D. (2002). "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews." In *Proceedings of the 40th Meeting of the Association for Computational Linguistics (ACL)*, pp. 417–424.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory.* Springer.

Wei Chu, S. S. K. (2005). "New Approaches to Support Vector Ordinal Regression." In *Proceedings of 22nd International Conference on Machine Learning (ICML)*, pp. 145–152.

略歴

岡野原大輔：1982 年生．2005 年東京大学理学部情報科学科卒，同年東京大学情報理工学系研究科修士課程に進学．専門は統計的自然言語処理．機械学習とアルゴリズムに興味を持つ．

辻井　潤一（正会員）：1949 年生．1971 年京都大学工学部電子工学科卒業，73 年同大学院修士課程終了．京都大学助手，同大学助教授，英国 UMIST 教授（1988 年）を経て，1995 年に東京大学大学院教授，現在に至る．2004 年より，マンチェスター大学教授，英国国立テキストマイニング・センター所長を兼任．ACL 会長（2006 年）．情報処理学会，人工知能学会，ACL 会員．工学博士．

（2006 年 4 月 20 日　受付）
（2006 年 7 月 17 日　再受付）
（2006 年 7 月 22 日　採録）