Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization

Andrew B. Goldberg
Computer Sciences Department
University of Wisconsin-Madison
Madison, W.I. 53706
goldberg@cs.wisc.edu

Xiaojin Zhu
Computer Sciences Department
University of Wisconsin-Madison
Madison, W.I. 53706
jerryzhu@cs.wisc.edu

Abstract

We present a graph-based semi-supervised learning algorithm to address the sentiment analysis task of rating inference. Given a set of documents (e.g., movie reviews) and accompanying ratings (e.g., "4 stars"), the task calls for inferring numerical ratings for unlabeled documents based on the perceived sentiment expressed by their text. In particular, we are interested in the situation where labeled data is scarce. We place this task in the semi-supervised setting and demonstrate that considering unlabeled reviews in the learning process can improve rating-inference performance. We do so by creating a graph on both labeled and unlabeled data to encode certain assumptions for this task. We then solve an optimization problem to obtain a smooth rating function over the whole graph. When only limited labeled data is available, this method achieves significantly better predictive accuracy over other methods that ignore the unlabeled examples during training.

1 Introduction

Sentiment analysis of text documents has received considerable attention recently (Shanahan et al., 2005; Turney, 2002; Dave et al., 2003; Hu and Liu, 2004; Chaovalit and Zhou, 2005). Unlike traditional text categorization based on topics, sentiment analysis attempts to identify the subjective sentiment expressed (or implied) in documents, such as consumer product or movie reviews. In particular Pang and Lee proposed the rating-inference problem (2005). Rating inference is harder than binary positive / negative opinion classification. The goal is to infer a numerical rating from reviews, for example the number of "stars" that a critic gave to a movie. Pang and Lee showed that supervised machine learning techniques (classification and regression) work well for rating inference with large amounts of training data.

However, review documents often do not come with numerical ratings. We call such documents *unlabeled data*. Standard supervised machine learning algorithms cannot learn from unlabeled data. Assigning labels can be a slow and expensive process because manual inspection and domain expertise are needed. Often only a small portion of the documents can be labeled within resource constraints, so most documents remain unlabeled. Supervised learning algorithms trained on small labeled sets suffer in performance. Can one use the unlabeled reviews to improve rating-inference? Pang and Lee (2005) suggested that doing so should be useful.

We demonstrate that the answer is 'Yes.' Our approach is graph-based semi-supervised learning. Semi-supervised learning is an active research area in machine learning. It builds better classifiers or regressors using both labeled and unlabeled data, under appropriate assumptions (Zhu, 2005; Seeger, 2001). This paper contains three contributions:

- We present a novel adaptation of graph-based semi-supervised learning (Zhu et al., 2003)

to the sentiment analysis domain, extending past supervised learning work by Pang and Lee (2005);

- We design a special graph which encodes our assumptions for rating-inference problems (section 2), and present the associated optimization problem in section 3;

- We show the benefit of semi-supervised learning for rating inference with extensive experimental results in section 4.

2 A Graph for Sentiment Categorization

The semi-supervised rating-inference problem is formalized as follows. There are n review documents $x_1 \dots x_n$, each represented by some standard feature representation (e.g., word-presence vectors). Without loss of generality, let the first $l \leq n$ documents be labeled with ratings $y_1 \dots y_l \in C$. The remaining documents are unlabeled. In our experiments, the unlabeled documents are also the test documents, a setting known as transduction. The set of numerical ratings are $C = \{c_1, \dots, c_C\}$, with $c_1 < \dots < c_C \in \mathbb{R}$. For example, a one-star to four-star movie rating system has $C = \{0, 1, 2, 3\}$. We seek a function $f : x \mapsto \mathbb{R}$ that gives a continuous rating $f(x)$ to a document x. Classification is done by mapping $f(x)$ to the nearest discrete rating in C. Note this is ordinal classification, which differs from standard multi-class classification in that C is endowed with an order. In the following we use 'review' and 'document,' 'rating' and 'label' interchangeably.

We make two assumptions:

1. We are given a *similarity measure* $w_{ij} \geq 0$ between documents x_i and x_j. w_{ij} should be computable from features, so that we can measure similarities between any documents, including unlabeled ones. A large w_{ij} implies that the two documents tend to express the same sentiment (i.e., rating). We experiment with *positive-sentence percentage* (PSP) based similarity which is proposed in (Pang and Lee, 2005), and mutual-information modulated word-vector cosine similarity. Details can be found in section 4.

2. Optionally, we are given numerical rating predictions $\hat{y}_{l+1}, \dots, \hat{y}_n$ on the unlabeled documents from a separate learner, for instance ϵ-insensitive support vector regression (Joachims, 1999; Smola and Schölkopf, 2004) used by (Pang and Lee, 2005). This acts as an extra knowledge source for our semi-supervised learning framework to improve upon. We note our framework is general and works without the separate learner, too. (For this to work in practice, a reliable similarity measure is required.)

We now describe our graph for the semi-supervised rating-inference problem. We do this piece by piece with reference to Figure 1. Our undirected graph $G = (V, E)$ has $2n$ nodes V, and weighted edges E among some of the nodes.

- Each document is a node in the graph (open circles, e.g., x_i and x_j). The true ratings of these nodes $f(x)$ are unobserved. This is true even for the labeled documents because we allow for noisy labels. Our goal is to infer $f(x)$ for the unlabeled documents.

- Each labeled document (e.g., x_j) is connected to an observed node (dark circle) whose value is the given rating y_j. The observed node is a 'dongle' (Zhu et al., 2003) since it connects only to x_j. As we point out later, this serves to pull $f(x_j)$ towards y_j. The edge weight between a labeled document and its dongle is a large number M. M represents the influence of y_j: if $M \to \infty$ then $f(x_j) = y_j$ becomes a hard constraint.

- Similarly each unlabeled document (e.g., x_i) is also connected to an observed dongle node $\hat{y}_i$, whose value is the prediction of the separate learner. Therefore we also require that $f(x_i)$ is close to $\hat{y}_i$. This is a way to incorporate multiple learners in general. We set the weight between an unlabeled node and its dongle arbitrarily to 1 (the weights are scale-invariant otherwise). As noted earlier, the separate learner is optional: we can remove it and still carry out graph-based semi-supervised learning.

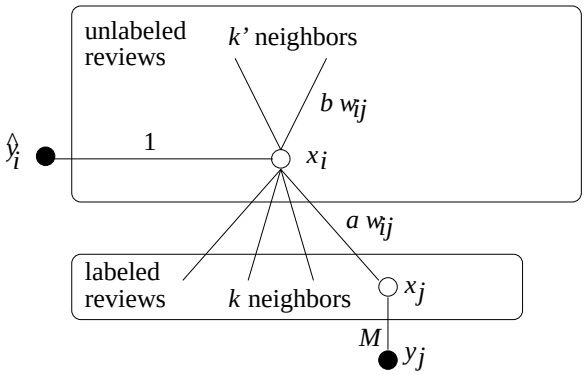

Figure 1: The graph for semi-supervised rating inference.

- Each unlabeled document x_i is connected to $kNN_L(i)$, its k nearest *labeled documents*. Distance is measured by the given similarity measure w. We want $f(x_i)$ to be consistent with its similar labeled documents. The weight between x_i and $x_j \in kNN_L(i)$ is $a \cdot w_{ij}$.
- Each unlabeled document is also connected to $k'NN_U(i)$, its k' nearest *unlabeled documents* (excluding itself). The weight between x_i and $x_j \in k'NN_U(i)$ is $b \cdot w_{ij}$. We also want $f(x_i)$ to be consistent with its similar unlabeled neighbors. We allow potentially different numbers of neighbors (k and k'), and different weight coefficients (a and b). These parameters are set by cross validation in experiments.

The last two kinds of edges are the key to semi-supervised learning: They connect unobserved nodes and force ratings to be smooth throughout the graph, as we discuss in the next section.

3 Graph-Based Semi-Supervised Learning

With the graph defined, there are several algorithms one can use to carry out semi-supervised learning (Zhu et al., 2003; Delalleau et al., 2005; Joachims, 2003; Blum and Chawla, 2001; Belkin et al., 2005). The basic idea is the same and is what we use in this paper. That is, our rating function $f(x)$ should be *smooth* with respect to the graph. $f(x)$ is not smooth if there is an edge with large weight w between nodes x_i and x_j, *and* the difference between $f(x_i)$ and $f(x_j)$ is large. The (un)smoothness over the particular edge can be defined as $w\big(f(x_i) - f(x_j)\big)^2$. Summing over all edges in the graph, we obtain the (un)smoothness $\mathcal{L}(f)$ over the whole graph. We call $\mathcal{L}(f)$ the *energy* or *loss*, which should be minimized. Let $L = 1 \ldots l$ and $U = l+1 \ldots n$ be labeled and unlabeled review indices, respectively. With the graph in Figure 1, the loss $\mathcal{L}(f)$ can be written as

$$\sum_{i\in L} M(f(x_i) - y_i)^2 + \sum_{i\in U} (f(x_i) - \hat{y}_i)^2 + \sum_{i\in U} \sum_{j\in kNN_L(i)} a w_{ij} (f(x_i) - f(x_j))^2 + \sum_{i\in U} \sum_{j\in k'NN_U(i)} b w_{ij} (f(x_i) - f(x_j))^2. \quad (1)$$

A small loss implies that the rating of an unlabeled review is close to its labeled peers as well as its unlabeled peers. This is how unlabeled data can participate in learning. The optimization problem is $\min_f \mathcal{L}(f)$. To understand the role of the parameters, we define $\alpha = ak + bk'$ and $\beta = \frac{b}{a}$, so that $\mathcal{L}(f)$ can be written as

$$\sum_{i\in L} M(f(x_i) - y_i)^2 + \sum_{i\in U} \Big[(f(x_i) - \hat{y}_i)^2 + \frac{\alpha}{k+\beta k'} \Big(\sum_{j\in kNN_L(i)} w_{ij} (f(x_i) - f(x_j))^2 + \sum_{j\in k'NN_U(i)} \beta w_{ij} (f(x_i) - f(x_j))^2 \Big) \Big]. \quad (2)$$

Thus β controls the relative weight between labeled neighbors and unlabeled neighbors; α is roughly the relative weight given to semi-supervised (non-dongle) edges.

We can find the closed-form solution to the optimization problem. Defining an $n \times n$ matrix $\bar{W}$,

$$\bar{W}_{ij} = \begin{cases} 0, & i \in L \\ w_{ij}, & j \in kNN_L(i) \\ \beta w_{ij}, & j \in k'NN_U(i). \end{cases} \quad (3)$$

Let $W = \max(\bar{W}, \bar{W}^\top)$ be a symmetrized version of this matrix. Let D be a diagonal *degree* matrix with

$$D_{ii} = \sum_{j=1}^{n} W_{ij}. \quad (4)$$

Note that we define a node's degree to be the sum of its edge weights. Let $\Delta = D - W$ be the combinatorial *Laplacian* matrix. Let C be a diagonal dongle

weight matrix with

$$C_{ii} = \begin{cases} M, & i \in L \\ 1, & i \in U \end{cases} . \quad (5)$$

Let $\mathbf{f} = (f(x_1), \ldots, f(x_n))^\top$ and $\mathbf{y} = (y_1, \ldots, y_l, \hat{y}_{l+1}, \ldots, \hat{y}_n)^\top$. We can rewrite $\mathcal{L}(f)$ as

$$(\mathbf{f} - \mathbf{y})^\top C(\mathbf{f} - \mathbf{y}) + \frac{\alpha}{k + \beta k'} \mathbf{f}^\top \Delta \mathbf{f}. \quad (6)$$

This is a quadratic function in $\mathbf{f}$. Setting the gradient to zero, $\partial \mathcal{L}(f)/\partial \mathbf{f} = 0$, we find the minimum loss function

$$\mathbf{f} = \left(C + \frac{\alpha}{k + \beta k'} \Delta \right)^{-1} C\mathbf{y}. \quad (7)$$

Because C has strictly positive eigenvalues, the inverse is well defined. All our semi-supervised learning experiments use (7) in what follows.

Before moving on to experiments, we note an interesting connection to the supervised learning method in (Pang and Lee, 2005), which formulates rating inference as a *metric labeling* problem (Kleinberg and Tardos, 2002). Consider a special case of our loss function (1) when $b = 0$ and $M \to \infty$. It is easy to show for labeled nodes $j \in L$, the optimal value is the given label: $f(x_j) = y_j$. Then the optimization problem decouples into a set of one-dimensional problems, one for each unlabeled node $i \in U$: $\mathcal{L}_{b=0, M \to \infty}(f(x_i)) =$

$$(f(x_i) - \hat{y}_i)^2 + \sum_{j \in kNN_L(i)} a w_{ij} (f(x_i) - y_j)^2. \quad (8)$$

The above problem is easy to solve. It corresponds exactly to the supervised, non-transductive version of metric labeling, except we use squared difference while (Pang and Lee, 2005) used absolute difference. Indeed in experiments comparing the two (not reported here), their differences are not statistically significant. From this perspective, our semi-supervised learning method is an extension with interacting terms among unlabeled data.

4 Experiments

We performed experiments using the movie review documents and accompanying 4-class ($C = \{0, 1, 2, 3\}$) labels found in the "scale dataset v1.0" available at http://www.cs.cornell.edu/people/pabo/movie-review-data/ and first used in (Pang and Lee, 2005). We chose 4-class instead of 3-class labeling because it is harder. The dataset is divided into four author-specific corpora, containing 1770, 902, 1307, and 1027 documents. We ran experiments individually for each author. Each document is represented as a $\{0, 1\}$ word-presence vector, normalized to sum to 1.

We systematically vary labeled set size $|L| \in \{0.9n, 800, 400, 200, 100, 50, 25, 12, 6\}$ to observe the effect of semi-supervised learning. $|L| = 0.9n$ is included to match 10-fold cross validation used by (Pang and Lee, 2005). For each $|L|$ we run 20 trials where we randomly split the corpus into labeled and test (unlabeled) sets. We ensure that all four classes are represented in each labeled set. The same random splits are used for all methods, allowing paired t-tests for statistical significance. All reported results are average test set accuracy.

We compare our graph-based semi-supervised method with two previously studied methods: regression and metric labeling as in (Pang and Lee, 2005).

4.1 Regression

We ran linear ϵ-insensitive support vector regression using Joachims' SVM^{light} package (1999) with all default parameters. The continuous prediction on a test document is discretized for classification. Regression results are reported under the heading 'reg.' Note this method does not use unlabeled data for training.

4.2 Metric labeling

We ran Pang and Lee's method based on metric labeling, using SVM regression as the initial label preference function. The method requires an item-similarity function, which is equivalent to our similarity measure w_{ij}. Among others, we experimented with PSP-based similarity. For consistency with (Pang and Lee, 2005), supervised metric labeling results with this measure are reported under 'reg+PSP.' Note this method does not use unlabeled data for training either.

PSP$_i$ is defined in (Pang and Lee, 2005) as the percentage of positive sentences in review x_i. The similarity between reviews x_i, x_j is the cosine angle

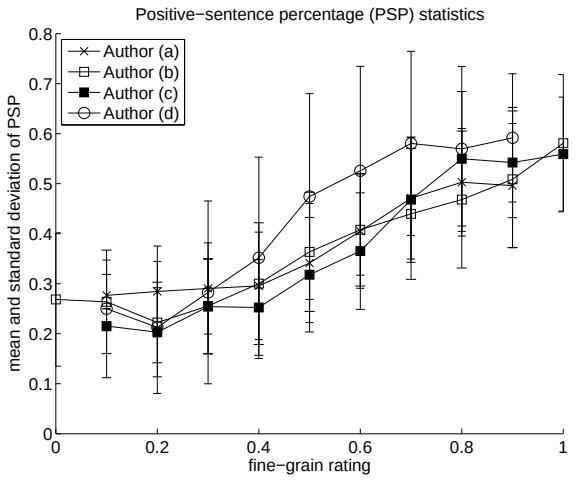

Figure 2: PSP for reviews expressing each fine-grain rating. We identified positive sentences using SVM instead of Naïve Bayes, but the trend is qualitatively the same as in (Pang and Lee, 2005).

between the vectors $(\text{PSP}_i, 1-\text{PSP}_i)$ and $(\text{PSP}_j, 1-\text{PSP}_j)$. Positive sentences are identified using a binary classifier trained on a separate "snippet data set" located at the same URL as above. The snippet data set contains 10662 short quotations taken from movie reviews appearing on the rottentomatoes.com Web site. Each snippet is labeled positive or negative based on the rating of the originating review. Pang and Lee (2005) trained a Naïve Bayes classifier. They showed that PSP is a (noisy) measure for comparing reviews—reviews with low ratings tend to receive low PSP scores, and those with higher ratings tend to get high PSP scores. Thus, two reviews with a high PSP-based similarity are expected to have similar ratings. For our experiments we derived PSP measurements in a similar manner, but using a linear SVM classifier. We observed the same relationship between PSP and ratings (Figure 2).

The metric labeling method has parameters (the equivalent of k, α in our model). Pang and Lee tuned them on a per-author basis using cross validation but did not report the optimal parameters. We were interested in learning a single set of parameters for use with all authors. In addition, since we varied labeled set size, it is convenient to tune $c = k/|L|$, the fraction of labeled reviews used as neighbors, instead of k. We then used the same c, α for all authors at all labeled set sizes in experiments involving PSP. Because c is fixed, k varies directly with $|L|$ (i.e., when less labeled data is available, our algorithm considers fewer nearby labeled examples). In an attempt to reproduce the findings in (Pang and Lee, 2005), we tuned c, α with cross validation. Tuning ranges are $c \in \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3\}$ and $\alpha \in \{0.01, 0.1, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 5.0\}$. The optimal parameters we found are $c = 0.2$ and $\alpha = 1.5$. (In section 4.4, we discuss an alternative similarity measure, for which we re-tuned these parameters.)

Note that we learned a single set of shared parameters for all authors, whereas (Pang and Lee, 2005) tuned k and α on a per-author basis. To demonstrate that our implementation of metric labeling produces comparable results, we also determined the optimal author-specific parameters. Table 1 shows the accuracy obtained over 20 trials with $|L| = 0.9n$ for each author, using SVM regression, reg+PSP using shared c, α parameters, and reg+PSP using author-specific c, α parameters (listed in parentheses). The best result in each row of the table is highlighted in bold. We also show in bold any results that cannot be distinguished from the best result using a paired t-test at the 0.05 level.

(Pang and Lee, 2005) found that their metric labeling method, when applied to the 4-class data we are using, was not statistically better than regression, though they observed some improvement for authors (c) and (d). Using author-specific parameters, we obtained the same qualitative result, but the improvement for (c) and (d) appears even less significant in our results. Possible explanations for this difference are the fact that we derived our PSP measurements using an SVM classifier instead of an NB classifier, and that we did not use the same range of parameters for tuning. The optimal shared parameters produced almost the same results as the optimal author-specific parameters, and were used in subsequent experiments.

4.3 Semi-Supervised Learning

We used the same PSP-based similarity measure and the same shared parameters $c = 0.2, \alpha = 1.5$ from our metric labeling experiments to perform graph-based semi-supervised learning. The results are reported as 'SSL+PSP.' SSL has three

Author	reg	reg+PSP (shared)	reg+PSP (specific)
(a)	**0.592**	**0.592**	**0.592** (0.05, 0.01)
(b)	**0.501**	**0.498**	**0.496** (0.05, 3.50)
(c)	**0.592**	0.589	**0.593** (0.15, 1.50)
(d)	**0.496**	**0.498**	**0.500** (0.05, 3.00)

Table 1: Accuracy using shared ($c = 0.2$, $\alpha = 1.5$) vs. author-specific parameters, with $|L| = 0.9n$.

additional parameters k', β, and M. Again we tuned k', β with cross validation. Tuning ranges are $k' \in \{2, 3, 5, 10, 20\}$ and $\beta \in \{0.001, 0.01, 0.1, 1.0, 10.0\}$. The optimal parameters are $k' = 5$ and $\beta = 1.0$. These were used for all authors and for all labeled set sizes. Note that unlike $k = c|L|$, which decreases as the labeled set size decreases, we let k' remain fixed for all $|L|$. We set M arbitrarily to a large number 10^8 to ensure that the ratings of labeled reviews are respected.

4.4 Alternate Similarity Measures

In addition to using PSP as a similarity measure between reviews, we investigated several alternative similarity measures based on the cosine of word vectors. Among these options were the cosine between the word vectors used to train the SVM regressor, and the cosine between word vectors containing only words with high (top 1000 or top 5000) mutual information values. The mutual information is computed with respect to the positive and negative classes in the 10662-document "snippet data set." Finally, we experimented with using as a similarity measure the cosine between word vectors containing all words, each weighted by its mutual information. We found this measure to be the best among the options tested in pilot trial runs using the metric labeling algorithm. Specifically, we scaled the mutual information values such that the maximum value was one. Then, we used these values as weights for the corresponding words in the word vectors. For words in the movie review data set that did not appear in the snippet data set, we used a default weight of zero (i.e., we excluded them. We experimented with setting the default weight to one, but found this led to inferior performance.)

We repeated the experiments described in sections 4.2 and 4.3 with the only difference being that we used the mutual-information weighted word vector similarity instead of PSP whenever a similarity measure was required. We repeated the tuning procedures described in the previous sections. Using this new similarity measure led to the optimal parameters $c = 0.1$, $\alpha = 1.5$, $k' = 5$, and $\beta = 10.0$. The results are reported under 'reg+WV' and 'SSL+WV,' respectively.

4.5 Results

We tested the five algorithms for all four authors using each of the nine labeled set sizes. The results are presented in table 2. Each entry in the table represents the average accuracy across 20 trials for an author, a labeled set size, and an algorithm. The best result in each row is highlighted in bold. Any results on the same row that cannot be distinguished from the best result using a paired t-test at the 0.05 level are also bold.

The results indicate that the graph-based semi-supervised learning algorithm based on PSP similarity (SSL+PSP) achieved better performance than all other methods in all four author corpora when only 200, 100, 50, 25, or 12 labeled documents were available. In 19 out of these 20 learning scenarios, the unlabeled set accuracy by the SSL+PSP algorithm was significantly higher than all other methods. While accuracy generally degraded as we trained on less labeled data, the decrease for the SSL approach was less severe through the mid-range labeled set sizes. SSL+PSP remains among the best methods with only 6 labeled examples.

Note that the SSL algorithm appears to be quite sensitive to the similarity measure used to form the graph on which it is based. In the experiments where we used mutual-information weighted word vector similarity (reg+WV and SSL+WV), we notice that reg+WV remained on par with reg+PSP at high labeled set sizes, whereas SSL+WV appears significantly worse in most of these cases. It is clear that PSP is the more reliable similarity measure. SSL uses the similarity measure in more ways than the metric labeling approaches (i.e., SSL's graph is denser), so it is not surprising that SSL's accuracy would suffer more with an inferior similarity measure.

Unfortunately, our SSL approach did not do as well with large labeled set sizes. We believe this

			PSP		word vector	
	$\|L\|$	regression	reg+PSP	SSL+PSP	reg+WV	SSL+WV
Author (a)	1593	**0.592**	**0.592**	0.546	**0.592**	0.544
	800	**0.553**	**0.554**	0.534	**0.553**	0.517
	400	**0.522**	**0.525**	**0.526**	**0.522**	0.497
	200	0.494	0.498	**0.521**	0.494	0.472
	100	0.463	0.477	**0.511**	0.462	0.450
	50	0.439	0.458	**0.499**	0.438	0.429
	25	0.408	0.421	**0.465**	0.400	0.404
	12	0.401	0.378	**0.451**	0.335	0.398
	6	0.390	0.359	**0.422**	0.314	0.389
Author (b)	811	0.501	0.498	0.481	**0.503**	0.473
	800	0.501	0.497	0.478	**0.503**	0.474
	400	**0.471**	**0.471**	**0.465**	**0.471**	0.450
	200	**0.447**	**0.449**	**0.452**	**0.447**	0.429
	100	0.415	0.423	**0.443**	0.415	0.397
	50	0.388	0.396	**0.434**	0.387	0.376
	25	0.373	0.380	**0.418**	0.364	0.367
	12	0.354	0.360	**0.399**	0.313	0.353
	6	0.348	**0.352**	**0.380**	0.302	0.347
Author (c)	1176	**0.592**	**0.589**	0.566	**0.594**	0.514
	800	0.579	**0.585**	0.559	0.579	0.509
	400	0.550	**0.556**	0.544	0.551	0.491
	200	0.513	0.519	**0.532**	0.513	0.479
	100	0.484	0.495	**0.521**	0.484	0.466
	50	0.462	0.476	**0.504**	0.461	0.456
	25	0.459	0.472	**0.484**	0.439	0.454
	12	0.420	0.405	**0.477**	0.356	0.414
	6	0.320	**0.382**	**0.366**	0.334	**0.322**
Author (d)	924	**0.496**	**0.498**	**0.495**	**0.499**	**0.490**
	800	0.500	**0.501**	**0.495**	**0.504**	0.483
	400	0.474	0.478	**0.486**	0.477	0.463
	200	0.459	0.459	**0.468**	0.459	0.445
	100	0.444	0.445	**0.460**	0.444	0.437
	50	0.429	0.431	**0.445**	0.429	0.428
	25	0.411	0.411	**0.425**	0.400	0.409
	12	0.393	0.362	**0.405**	0.335	0.391
	6	**0.393**	0.357	**0.403**	0.312	**0.393**

Table 2: 20-trial average unlabeled set accuracy for each author across different labeled set sizes and methods. In each row, we list in bold the best result and any results that cannot be distinguished from it with a paired t-test at the 0.05 level.

is due to two factors: a) the baseline SVM regressor trained on a large labeled set can achieve fairly high accuracy for this difficult task without considering pairwise relationships between examples; b) PSP similarity is not accurate enough. Gain in variance reduction achieved by the SSL graph is offset by its bias when labeled data is abundant.

5 Discussion

We have demonstrated the benefit of using unlabeled data for rating inference. There are several directions to improve the work: 1. We will investigate better document representations and similarity measures based on parsing and other linguistic knowledge, as well as reviews' sentiment patterns. For example, several positive sentences followed by a few concluding negative sentences could indicate an overall negative review, as observed in prior work (Pang and Lee, 2005). 2. Our method is transductive: new reviews must be added to the graph before they can be classified. We will extend it to the inductive learning setting based on (Sindhwani et al., 2005). 3. We plan to experiment with cross-reviewer and cross-domain analysis, such as using a model learned on movie reviews to help classify product reviews.

Acknowledgment

We thank Bo Pang, Lillian Lee and anonymous reviewers for helpful comments.

References

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. 2005. On manifold regularization. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT 2005)*.

A. Blum and S. Chawla. 2001. Learning from labeled and unlabeled data using graph mincuts. In *Proc. 18th International Conf. on Machine Learning*.

Pimwadee Chaovalit and Lina Zhou. 2005. Movie review mining: a comparison between supervised and unsupervised classification approaches. In *HICSS*. IEEE Computer Society.

Kushal Dave, Steve Lawrence, and David M. Pennock. 2003. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 519–528.

Olivier Delalleau, Yoshua Bengio, and Nicolas Le Roux. 2005. Efficient non-parametric function induction in semi-supervised learning. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT 2005)*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of KDD '04, the ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM Press.

T. Joachims. 1999. Making large-scale svm learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

T. Joachims. 2003. Transductive learning via spectral graph partitioning. In *Proceedings of ICML-03, 20th International Conference on Machine Learning*.

Jon M. Kleinberg and Éva Tardos. 2002. Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. *J. ACM*, 49(5):616–639.

Bo Pang and Lillian Lee. 2005. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*.

Matthias Seeger. 2001. Learning with labeled and unlabeled data. Technical report, University of Edinburgh.

James Shanahan, Yan Qu, and Janyce Wiebe, editors. 2005. *Computing attitude and affect in text*. Springer, Dordrecht, The Netherlands.

Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. 2005. Beyond the point cloud: from transductive to semi-supervised learning. In *ICML05, 22nd International Conference on Machine Learning*, Bonn, Germany.

A. J. Smola and B. Schölkopf. 2004. A tutorial on support vector regression. *Statistics and Computing*, 14:199–222.

Peter Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL-02, 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. 2003. Semi-supervised learning using Gaussian fields and harmonic functions. In *ICML-03, 20th International Conference on Machine Learning*.

Xiaojin Zhu. 2005. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison. http://www.cs.wisc.edu/∼jerryzhu/pub/ssl_survey.pdf.