Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales

Bo Pang[1,3] and **Lillian Lee**[1,2,3]
(1) Department of Computer Science, Cornell University
(2) Language Technologies Institute, Carnegie Mellon University
(3) Computer Science Department, Carnegie Mellon University

Abstract

We address the *rating-inference problem*, wherein rather than simply decide whether a review is "thumbs up" or "thumbs down", as in previous sentiment analysis work, one must determine an author's evaluation with respect to a multi-point scale (e.g., one to five "stars"). This task represents an interesting twist on standard multi-class text categorization because there are several different degrees of similarity between class labels; for example, "three stars" is intuitively closer to "four stars" than to "one star".

We first evaluate human performance at the task. Then, we apply a meta-algorithm, based on a *metric labeling* formulation of the problem, that alters a given n-ary classifier's output in an explicit attempt to ensure that similar items receive similar labels. We show that the meta-algorithm can provide significant improvements over both multi-class and regression versions of SVMs when we employ a novel similarity measure appropriate to the problem.

Publication info: *Proceedings of the ACL, 2005.*

1 Introduction

There has recently been a dramatic surge of interest in *sentiment analysis*, as more and more people become aware of the scientific challenges posed and the scope of new applications enabled by the processing of subjective language. (The papers collected by Qu, Shanahan, and Wiebe (2004) form a representative sample of research in the area.) Most prior work on the specific problem of categorizing expressly opinionated text has focused on the binary distinction of positive vs. negative (Turney, 2002; Pang, Lee, and Vaithyanathan, 2002; Dave, Lawrence, and Pennock, 2003; Yu and Hatzivassiloglou, 2003). But it is often helpful to have more information than this binary distinction provides, especially if one is ranking items by recommendation or comparing several reviewers' opinions: example applications include collaborative filtering and deciding which conference submissions to accept.

Therefore, in this paper we consider generalizing to finer-grained *scales*: rather than just determine whether a review is "thumbs up" or not, we attempt to infer the author's implied numerical rating, such as "three stars" or "four stars". Note that this differs from identifying opinion *strength* (Wilson, Wiebe, and Hwa, 2004): rants and raves have the same strength but represent opposite evaluations, and referee forms often allow one to indicate that one is very confident (high strength) that a conference submission is mediocre (middling rating). Also, our task differs from *ranking* not only because one can be given a single item to classify (as opposed to a set of items to be ordered relative to one another), but because there are settings in which classification is harder than ranking, and vice versa.

One can apply standard n-ary classifiers or regression to this *rating-inference problem*; independent work by Koppel and Schler (2005) considers such methods. But an alternative approach that explicitly incorporates information about item similarities together with label similarity information (for instance, "one star" is closer to "two stars" than to "four stars") is to think of the task as one of *metric labeling* (Kleinberg and Tardos, 2002), where label relations are encoded via a distance metric. This observation yields a meta-algorithm, applicable to both semi-supervised (via graph-theoretic techniques) and supervised settings, that alters a given n-ary classifier's output so that similar items tend to be assigned similar labels.

In what follows, we first demonstrate that humans can discern relatively small differences in (hidden) evaluation scores, indicating that rating inference is indeed a meaningful task. We then present three types of algorithms — one-vs-all, regression, and metric labeling — that can be distinguished by how explicitly they attempt to leverage similarity between items and between labels. Next, we consider what item similarity measure to apply, proposing one based on the *positive-sentence percentage*. Incorporating this new measure within the metric-labeling framework is shown to often provide significant improvements over the other algorithms.

We hope that some of the insights derived here might apply to other scales for text classifcation that have been considered, such as clause-level opinion strength (Wilson, Wiebe, and Hwa, 2004); affect types like disgust (Subasic and Huettner, 2001; Liu, Lieberman, and Selker, 2003); reading level (Collins-Thompson and Callan, 2004); and urgency or criticality (Horvitz, Jacobs, and Hovel, 1999).

2 Problem validation and formulation

We first ran a small pilot study on human subjects in order to establish a rough idea of what a reasonable classification granularity is: if even people cannot accurately infer labels with respect to a five-star scheme with half stars, say, then we cannot expect a learning algorithm to do so. Indeed, some potential obstacles to accurate rating inference include lack of calibration (e.g., what an understated author intends as high praise may seem lukewarm), author inconsistency at assigning fine-grained ratings, and

Rating diff.	Pooled	Subject 1	Subject 2
3 or more	100%	100% (35)	100% (15)
2 (e.g., 1 star)	83%	77% (30)	100% (11)
1 (e.g., $\frac{1}{2}$ star)	69%	65% (57)	90% (10)
0	55%	47% (15)	80% (5)

Table 1: Human accuracy at determining relative positivity. Rating differences are given in "notches". Parentheses enclose the number of pairs attempted.

ratings not entirely supported by the text[1].

For data, we first collected Internet movie reviews in English from four authors, removing explicit rating indicators from each document's text automatically. Now, while the obvious experiment would be to ask subjects to guess the rating that a review represents, doing so would force us to specify a fixed rating-scale granularity in advance. Instead, we examined people's ability to discern *relative differences*, because by varying the rating differences represented by the test instances, we can evaluate multiple granularities in a single experiment. Specifically, at intervals over a number of weeks, we authors (a non-native and a native speaker of English) examined pairs of reviews, attemping to determine whether the first review in each pair was (1) more positive than, (2) less positive than, or (3) as positive as the second. The texts in any particular review pair were taken from the same author to factor out the effects of cross-author divergence.

As Table 1 shows, both subjects performed perfectly when the rating separation was at least 3 "notches" in the original scale (we define a notch as a half star in a four- or five-star scheme and 10 points in a 100-point scheme). Interestingly, although human performance drops as rating difference decreases, even at a one-notch separation, both subjects handily outperformed the random-choice baseline of 33%. However, there was large variation in accuracy between subjects.[2]

[1]For example, the critic Dennis Schwartz writes that "sometimes the review itself [indicates] the letter grade should have been higher or lower, as the review might fail to take into consideration my overall impression of the film — which I hope to capture in the grade" (http://www.sover.net/˜ozus/cinema.htm).

[2]One contributing factor may be that the subjects viewed disjoint document sets, since we wanted to maximize experimental coverage of the types of document pairs within each difference class. We thus cannot report inter-annotator agreement,

Because of this variation, we defined two different classification regimes. From the evidence above, a **three-class** task (categories 0, 1, and 2 — essentially "negative", "middling", and "positive", respectively) seems like one that most people would do quite well at (but we should not assume 100% human accuracy: according to our one-notch results, people may misclassify borderline cases like 2.5 stars). Our study also suggests that people could do at least fairly well at distinguishing full stars in a zero- to four-star scheme. However, when we began to construct five-category datasets for each of our four authors (see below), we found that in each case, either the most negative or the most positive class (but not both) contained only about 5% of the documents. To make the classes more balanced, we folded these minority classes into the adjacent class, thus arriving at a **four-class** problem (categories 0-3, increasing in positivity). Note that the four-class problem seems to offer more possibilities for leveraging class relationship information than the three-class setting, since it involves more class pairs. Also, even the two-category version of the rating-inference problem for movie reviews has proven quite challenging for many automated classification techniques (Pang, Lee, and Vaithyanathan, 2002; Turney, 2002).

We applied the above two labeling schemes to a **scale dataset**[3] containing four corpora of movie reviews. All reviews were automatically preprocessed to remove both explicit rating indicators and objective sentences; the motivation for the latter step is that it has previously aided positive vs. negative classification (Pang and Lee, 2004). All of the 1770, 902, 1307, or 1027 documents in a given corpus were written by the same author. This decision facilitates interpretation of the results, since it factors out the effects of different choices of methods for calibrating authors' scales.[4] We point out that it is possible to gather author-specific information in some practical applications: for instance, systems that use selected authors (e.g., the Rotten Tomatoes movie-review website — where, we note, not all authors provide explicit ratings) could require that someone submit rating-labeled samples of newly-admitted authors' work. Moreover, our results at least partially generalize to mixed-author situations (see Section 5.2).

3 Algorithms

Recall that the problem we are considering is multi-category classification in which the labels can be naturally mapped to a metric space (e.g., points on a line); for simplicity, we assume the distance metric $d(\ell, \ell') = |\ell - \ell'|$ throughout. In this section, we present three approaches to this problem in order of increasingly explicit use of pairwise similarity information between items and between labels. In order to make comparisons between these methods meaningful, we base all three of them on Support Vector Machines (SVMs) as implemented in Joachims' (1999) SVM^{light} package.

3.1 One-vs-all

The standard SVM formulation applies only to binary classification. *One-vs-all* (OVA) (Rifkin and Klautau, 2004) is a common extension to the n-ary case. Training consists of building, for each label ℓ, an SVM binary classifier distinguishing label ℓ from "not-ℓ". We consider the final output to be a label preference function $\pi^{\mathrm{ova}}(x, \ell)$, defined as the signed distance of (test) item x to the ℓ side of the ℓ vs. not-ℓ decision plane.

Clearly, OVA makes no explicit use of pairwise label or item relationships. However, it can perform well if each class exhibits sufficiently distinct language; see Section 4 for more discussion.

3.2 Regression

Alternatively, we can take a *regression* perspective by assuming that the labels come from a discretization of a continuous function g mapping from the

but since our goal is to recover a reviewer's "true" recommendation, reader-author agreement is more relevant.

While another factor might be degree of English fluency, in an informal experiment (six subjects viewing the same three pairs), native English speakers made the only two errors.

[3] Available at http://www.cs.cornell.edu/People/pabo/movie-review-data as scale dataset v1.0.

[4] From the Rotten Tomatoes website's FAQ: "star systems are not consistent between critics. For critics like Roger Ebert and James Berardinelli, 2.5 stars or lower out of 4 stars is always negative. For other critics, 2.5 stars can either be positive or negative. Even though Eric Lurio uses a 5 star system, his grading is very relaxed. So, 2 stars can be positive." Thus, calibration may sometimes require strong familiarity with the authors involved, as anyone who has ever needed to reconcile conflicting referee reports probably knows.

feature space to a metric space.[5] If we choose g from a family of sufficiently "gradual" functions, then similar items necessarily receive similar labels. In particular, we consider *linear, ε-insensitive* SVM regression (Vapnik, 1995; Smola and Schölkopf, 1998); the idea is to find the hyperplane that best fits the training data, but where training points whose labels are within distance ε of the hyperplane incur no loss. Then, for (test) instance x, the label preference function $\pi^{\text{reg}}(x, \ell)$ is the negative of the distance between ℓ and the value predicted for x by the fitted hyperplane function.

Wilson, Wiebe, and Hwa (2004) used SVM regression to classify clause-level strength of opinion, reporting that it provided lower accuracy than other methods. However, independently of our work, Koppel and Schler (2005) found that applying linear regression to classify documents (in a different corpus than ours) with respect to a three-point rating scale provided greater accuracy than OVA SVMs and other algorithms.

3.3 Metric labeling

Regression *implicitly* encodes the "similar items, similar labels" heuristic, in that one can restrict consideration to "gradual" functions. But we can also think of our task as a *metric labeling* problem (Kleinberg and Tardos, 2002), a special case of the maximum *a posteriori* estimation problem for Markov random fields, to *explicitly* encode our desideratum. Suppose we have an initial label preference function $\pi(x, \ell)$, perhaps computed via one of the two methods described above. Also, let d be a distance metric on labels, and let $nn_k(x)$ denote the k nearest neighbors of item x according to some item-similarity function sim. Then, it is quite natural to pose our problem as finding a mapping of instances x to labels ℓ_x (respecting the original labels of the training instances) that minimizes

$$\sum_{x \in \text{test}} \left[-\pi(x, \ell_x) + \alpha \sum_{y \in nn_k(x)} f(d(\ell_x, \ell_y)) sim(x, y) \right],$$

where f is monotonically increasing (we chose $f(d) = d$ unless otherwise specified) and α is a trade-off and/or scaling parameter. (The inner summation is familiar from work in *locally-weighted learning*[6] (Atkeson, Moore, and Schaal, 1997).) In a sense, we are using explicit item and label similarity information to increasingly penalize the initial classifier as it assigns more divergent labels to similar items.

In this paper, we only report supervised-learning experiments in which the nearest neighbors for any given test item were drawn from the training set alone. In such a setting, the labeling decisions for different test items are independent, so that solving the requisite optimization problem is simple.

Aside: transduction The above formulation also allows for *transductive* semi-supervised learning as well, in that we could allow nearest neighbors to come from both the training and test sets. We intend to address this case in future work, since there are important settings in which one has a small number of labeled reviews and a large number of unlabeled reviews, in which case considering similarities between unlabeled texts could prove quite helpful. In full generality, the corresponding multi-label optimization problem is intractable, but for many families of f functions (e.g., convex) there exist practical exact or approximation algorithms based on techniques for finding *minimum s-t cuts* in graphs (Ishikawa and Geiger, 1998; Boykov, Veksler, and Zabih, 1999; Ishikawa, 2003). Interestingly, previous sentiment analysis research found that a minimum-cut formulation for the binary subjective/objective distinction yielded good results (Pang and Lee, 2004). Of course, there are many other related semi-supervised learning algorithms that we would like to try as well; see Zhu (2005) for a survey.

4 Class struggle: finding a label-correlated item-similarity function

We need to specify an item similarity function sim to use the metric-labeling formulation described in Section 3.3. We could, as is commonly done, employ a term-overlap-based measure such as the cosine between term-frequency-based document vectors (henceforth "TO(cos)"). However, Table 2

[5] We discuss the *ordinal* regression variant in Section 6.

[6] If we ignore the $\pi(x, \ell)$ term, different choices of f correspond to different versions of nearest-neighbor learning, e.g., majority-vote, weighted average of labels, or weighted median of labels.

	Label difference: 1	2	3
Three-class data	37%	33%	—
Four-class data	34%	31%	30%

Table 2: Average over authors and class pairs of between-class vocabulary overlap as the class labels of the pair grow farther apart.

shows that in aggregate, the vocabularies of distant classes overlap to a degree surprisingly similar to that of the vocabularies of nearby classes. Thus, item similarity as measured by TO(cos) may not correlate well with similarity of the item's true labels.

We can potentially develop a more useful similarity metric by asking ourselves what, intuitively, accounts for the label relationships that we seek to exploit. A simple hypothesis is that ratings can be determined by the *positive-sentence percentage (PSP)* of a text, i.e., the number of positive sentences divided by the number of subjective sentences. (Term-based versions of this premise have motivated much sentiment-analysis work for over a decade (Das and Chen, 2001; Tong, 2001; Turney, 2002).) But counterexamples are easy to construct: reviews can contain off-topic opinions, or recount many positive aspects before describing a fatal flaw.

We therefore tested the hypothesis as follows. To avoid the need to hand-label sentences as positive or negative, we first created a *sentence polarity dataset*[7] consisting of 10,662 movie-review "snippets" (a striking extract usually one sentence long) downloaded from www.rottentomatoes.com; each snippet was labeled with its source review's label (positive or negative) as provided by Rotten Tomatoes. Then, we trained a Naive Bayes classifier on this data set and applied it to our scale dataset to identify the positive sentences (recall that objective sentences were already removed).

Figure 1 shows that all four authors tend to exhibit a higher PSP when they write a more positive review, and we expect that most typical reviewers would follow suit. Hence, PSP appears to be a promising basis for computing document similarity for our rating-inference task. In particular, we defined $\overrightarrow{\mathrm{PSP}(x)}$ to be the two-dimensional vector $(\mathrm{PSP}(x), 1 - \mathrm{PSP}(x))$, and then set the item-similarity function required by the metric-labeling optimization function (Section 3.3) to $sim(x, y) = \cos\left(\overrightarrow{\mathrm{PSP}(x)}, \overrightarrow{\mathrm{PSP}(y)}\right)$.[8]

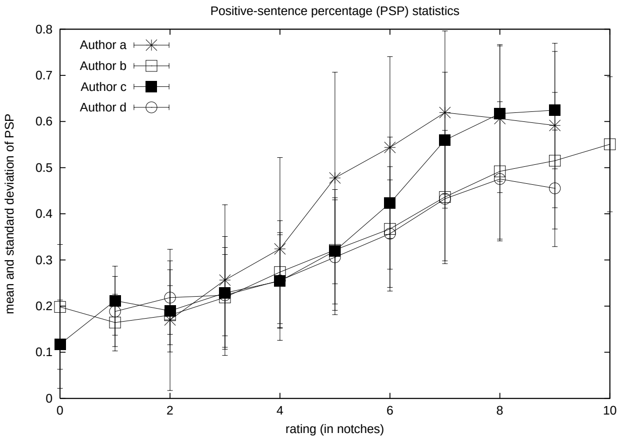


Figure 1: Average and standard deviation of PSP for reviews expressing different ratings.

But before proceeding, we note that it is possible that similarity information might yield no extra benefit at all. For instance, we don't need it if we can reliably identify each class just from some set of distinguishing terms. If we define such terms as frequent ones ($n \geq 20$) that appear in a single class 50% or more of the time, then we do find many instances; some examples for one author are: "meaningless", "disgusting" (class 0); "pleasant", "uneven" (class 1); and "oscar", "gem" (class 2) for the three-class case, and, in the four-class case, "flat", "tedious" (class 1) versus "straightforward", "likeable" (class 2). Some unexpected distinguishing terms for this author are "lion" for class 2 (three-class case), and for class 2 in the four-class case, "jennifer", for a wide variety of Jennifers.

5 Evaluation

This section compares the accuracies of the approaches outlined in Section 3 on the four corpora comprising our scale dataset. (Results using L_1 error were qualitatively similar.) Throughout, when

[7] Available at http://www.cs.cornell.edu/People/pabo/movie-review-data as sentence polarity dataset v1.0.

[8] While admittedly we initially chose this function because it was convenient to work with cosines, *post hoc* analysis revealed that the corresponding metric space "stretched" certain distances in a useful way.

we refer to something as "significant", we mean statistically so with respect to the paired t-test, $p < .05$.

The results that follow are based on SVM^{light}'s default parameter settings for SVM regression and OVA. Preliminary analysis of the effect of varying the regression parameter ε in the four-class case revealed that the default value was often optimal.

The notation "A+B" denotes metric labeling where method A provides the initial label preference function π and B serves as similarity measure. To train, we first select the meta-parameters k and α by running 9-fold cross-validation within the training set. Fixing k and α to those values yielding the best performance, we then re-train A (but with SVM parameters fixed, as described above) on the whole training set. At test time, the nearest neighbors of each item are also taken from the full training set.

5.1 Main comparison

Figure 2 summarizes our average 10-fold cross-validation accuracy results. We first observe from the plots that all the algorithms described in Section 3 always definitively outperform the simple baseline of predicting the majority class, although the improvements are smaller in the four-class case. Incidentally, the data was distributed in such a way that the absolute performance of the baseline itself does not change much between the three- and four-class case (which implies that the three-class datasets were relatively more balanced); and Author c's datasets seem noticeably easier than the others.

We now examine the effect of implicitly using label and item similarity. In the four-class case, regression performed better than OVA (significantly so for two authors, as shown in the righthand table); but for the three-category task, OVA significantly outperforms regression for all four authors. One might initially interprete this "flip" as showing that in the four-class scenario, item and label similarities provide a richer source of information relative to class-specific characteristics, especially since for the non-majority classes there is less data available; whereas in the three-class setting the categories are better modeled as quite distinct entities.

However, the three-class results for metric labeling on top of OVA and regression (shown in Figure 2 by black versions of the corresponding icons) show that employing explicit similarities always improves results, often to a significant degree, and yields the best overall accuracies. Thus, we *can* in fact effectively exploit similarities in the three-class case. Additionally, in both the three- and four- class scenarios, metric labeling often brings the performance of the weaker base method up to that of the stronger one (as indicated by the "disappearance" of upward triangles in corresponding table rows), and never hurts performance significantly.

In the four-class case, metric labeling and regression seem roughly equivalent. One possible interpretation is that the relevant structure of the problem is already captured by linear regression (and perhaps a different kernel for regression would have improved its three-class performance). However, according to additional experiments we ran in the four-class situation, the test-set-optimal parameter settings for metric labeling would have produced significant improvements, indicating there may be greater potential for our framework. At any rate, we view the fact that metric labeling performed quite well for both rating scales as a definitely positive result.

5.2 Further discussion

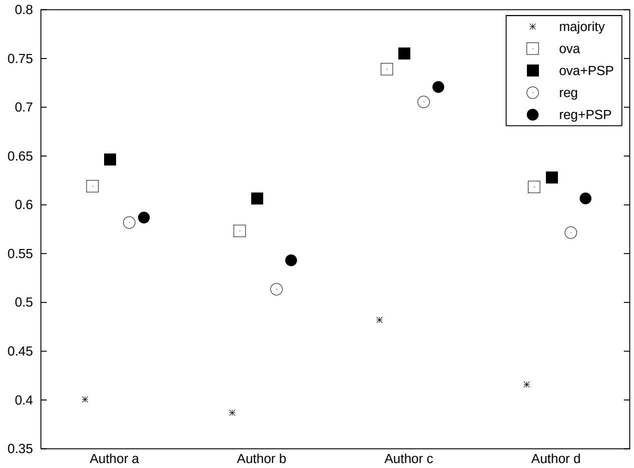
Q: Metric labeling looks like it's just combining SVMs with nearest neighbors, and classifier combination often improves performance. Couldn't we get the same kind of results by combining SVMs with any other reasonable method?

A: No. For example, if we take the strongest base SVM method for initial label preferences, but replace PSP with the term-overlap-based cosine (TO(cos)), performance often drops significantly. This result, which is in accordance with Section 4's data, suggests that choosing an item similarity function that correlates well with label similarity is important. (ova+PSP ◁◁◁◁ ova+TO(cos) [3c]; reg+PSP ◁ reg+TO(cos) [4c])

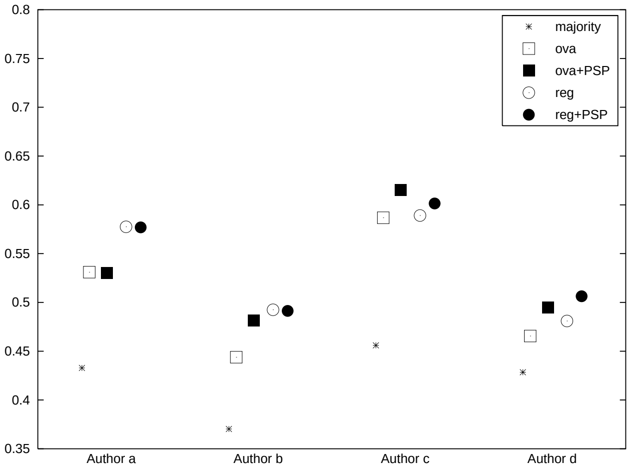
Q: Could you explain that notation, please?

A: Triangles point toward the significantly better algorithm for some dataset. For instance, "M ◁◁▷ N [3c]" means, "In the 3-class task, method M is significantly better than N for two author datasets and significantly worse for one dataset (so the algorithms were statistically indistinguishable on the remaining dataset)". When the algorithms being compared are statistically indistinguishable on

Average accuracies, three-class data



Average accuracies, four-class data



Average ten-fold cross-validation accuracies. Open icons: SVMs in either one-versus-all (square) or regression (circle) mode; dark versions: metric labeling using the corresponding SVM together with the positive-sentence percentage (PSP). The y-axes of the two plots are aligned.

Significant differences, three-class data

	ova	ova+PSP	reg	reg+PSP
	a b c d	a b c d	a b c d	a b c d
ova		△△△ ·	◁◁◁◁	· ◁ · ·
ova+PSP	◀◀◀ ·		◁◁◁◁	◁◁◁ ·
reg	△△△△	△△△△		· △ · △
reg+PSP	· △ · ·	△△△ ·	· ◀ · ◀	

Significant differences, four-class data

	ova	ova+PSP	reg	reg+PSP
	a b c d	a b c d	a b c d	a b c d
ova		· △△△	△△ · ·	△ · · △
ova+PSP	· ◀◀◀		△ · · ·	△ · · ·
reg	◁◁ · ·	◁ · · ·		· · · ·
reg+PSP	◁ · · ◁	◁ · · ·	• • • •	

Triangles point towards significantly better algorithms for the results plotted above. Specifically, if the difference between a row and a column algorithm for a given author dataset (a, b, c, or d) is significant, a triangle points to the better one; otherwise, a dot (.) is shown. Dark icons highlight the effect of adding PSP information via metric labeling.

Figure 2: Results for main experimental comparisons.

all four datasets (the "no triangles" case), we indicate this with an equals sign ("=").

Q: Thanks. Doesn't Figure 1 show that the positive-sentence percentage would be a good classifier even in isolation, so metric labeling isn't necessary?
A: No. Predicting class labels directly from the PSP value via trained thresholds isn't as effective (ova+PSP ◁◁◁◁ threshold PSP [3c]; reg+PSP ◁◁ threshold PSP [4c]).

Alternatively, we could use only the PSP component of metric labeling by setting the label preference function to the constant function 0, but even with *test-set-optimal* parameter settings, doing so underperforms the *trained* metric labeling algorithm with access to an initial SVM classifier (ova+PSP ◁◁◁◁ 0+PSP* [3c]; reg+PSP ◁◁ 0+PSP* [4c]).

Q: What about using PSP as one of the features for input to a standard classifier?
A: Our focus is on investigating the utility of similarity information. In our particular rating-inference setting, it so happens that the basis for our pairwise similarity measure can be incorporated as an

item-specific feature, but we view this as a tangential issue. That being said, preliminary experiments show that metric labeling can be better, barely (for test-set-optimal parameter settings for both algorithms: significantly better results for one author, four-class case; statistically indistinguishable otherwise), although one needs to determine an appropriate weight for the PSP feature to get good performance.

Q: You defined the "metric transformation" function f as the identity function $f(d) = d$, imposing greater loss as the distance between labels assigned to two similar items increases. Can you do just as well if you penalize all non-equal label assignments by the same amount, or does the distance between labels really matter?
A: You're asking for a comparison to the *Potts model*, which sets f to the function $\hat{f}(d) = 1$ if $d > 0$, 0 otherwise. In the one setting in which there is a significant difference between the two, the Potts model does worse (ova+PSP ◁ ova$\hat{+}$PSP [3c]). Also, employing the Potts model generally leads to fewer significant improvements over a chosen base method (compare Figure 2's tables with: reg$\hat{+}$PSP ◁ reg [3c]; ova$\hat{+}$PSP ◁◁ ova [3c]; ova$\hat{+}$PSP = ova [4c]; but note that reg$\hat{+}$PSP ◁ reg [4c]). We note that optimizing the Potts model in the multi-label case is NP-hard, whereas the optimal metric labeling with the identity metric-transformation function can be efficiently obtained (see Section 3.3).

Q: Your datasets had many labeled reviews and only one author each. Is your work relevant to settings with many authors but very little data for each?
A: As discussed in Section 2, it can be quite difficult to properly calibrate different authors' scales, since the same number of "stars" even within what is ostensibly the same rating system can mean different things for different authors. But since you ask: we temporarily turned a blind eye to this serious issue, creating a collection of 5394 reviews by 496 authors with at most 80 reviews per author, where we pretended that our rating conversions mapped correctly into a universal rating scheme. Preliminary results on this dataset were actually comparable to the results reported above, although since we are not confident in the class labels themselves, more work is needed to derive a clear analysis of this setting. (Abusing notation, since we're already playing fast and loose: [3c]: baseline 52.4%, reg 61.4%, reg+PSP 61.5%, ova (65.4%) ▷ ova+PSP (66.3%); [4c]: baseline 38.8%, reg (51.9%) ▷ reg+PSP (52.7%), ova (53.8%) ▷ ova+PSP (54.6%))

In future work, it would be interesting to determine author-independent characteristics that can be used on (or suitably adapted to) data for specific authors.

Q: How about trying —
A: —Yes, there are many alternatives. A few that we tested are described in the Appendix, and we propose some others in the next section. We should mention that we have not yet experimented with *all-vs.-all* (AVA), another standard binary-to-multi-category classifier conversion method, because we wished to focus on the effect of omitting pairwise information. In independent work on 3-category rating inference for a different corpus, Koppel and Schler (2005) found that regression outperformed AVA, and Rifkin and Klautau (2004) argue that in principle OVA should do just as well as AVA. But we plan to try it out.

6 Related work and future directions

In this paper, we addressed the rating-inference problem, showing the utility of employing label similarity and (appropriate choice of) item similarity — either implicitly, through regression, or explicitly and often more effectively, through metric labeling.

In the future, we would like to apply our methods to other scale-based classification problems, and explore alternative methods. Clearly, varying the kernel in SVM regression might yield better results. Another choice is *ordinal regression* (McCullagh, 1980; Herbrich, Graepel, and Obermayer, 2000), which only considers the ordering on labels, rather than any explicit distances between them; this approach could work well if a good metric on labels is lacking. Also, one could use mixture models (e.g., combine "positive" and "negative" language models) to capture class relationships (McCallum, 1999; Schapire and Singer, 2000; Takamura, Matsumoto, and Yamada, 2004).

We are also interested in framing multi-class but *non*-scale-based categorization problems as metric labeling tasks. For example, positive vs. negative vs. neutral sentiment distinctions are sometimes considered in which neutral means either objective (Engström, 2004) or a conflation of objective with a rating of mediocre (Das and Chen, 2001). (Koppel and Schler (2005) in independent work also discuss various types of neutrality.) In either case, we could apply a metric in which positive and negative are closer to objective (or objective+mediocre) than to each other. As another example, hierarchical label relationships can be easily encoded in a label metric.

Finally, as mentioned in Section 3.3, we would like to address the transductive setting, in which one has a small amount of labeled data and uses relationships between unlabeled items, since it is particularly well-suited to the metric-labeling approach and may be quite important in practice.

Acknowledgments We thank Paul Bennett, Dave Blei, Claire Cardie, Shimon Edelman, Thorsten Joachims, Jon Kleinberg, Oren Kurland, John Lafferty, Guy Lebanon, Pradeep Ravikumar, Jerry Zhu, and the anonymous reviewers for many very useful comments and discussion. We learned of Moshe Koppel and Jonathan Schler's work while preparing the camera-ready version of this paper; we thank them for so quickly answering our request for a pre-print. Our descriptions of their work are based on that pre-print; we apologize in advance for any inaccuracies in our descriptions that result from changes between their pre-print and their final version. We also thank CMU for its hospitality during the year. This paper is based upon work supported in part by the National Science Foundation (NSF) under grant no. IIS-0329064 and CCR-0122581; SRI International under subcontract no. 03-000211 on their project funded by the Department of the Interior's National Business Center; and by an Alfred P. Sloan Research Fellowship. Any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views or official policies, either expressed or implied, of any sponsoring institutions, the U.S. government, or any other entity.

References

Atkeson, Christopher G., Andrew W. Moore, and Stefan Schaal. 1997. Locally weighted learning. *Artificial Intelligence Review*, 11(1):11–73.

Boykov, Yuri, Olga Veksler, and Ramin Zabih. 1999. Fast approximate energy minimization via graph cuts. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 377–384. Journal version in *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 23(11):1222–1239, 2001.

Collins-Thompson, Kevyn and Jamie Callan. 2004. A language modeling approach to predicting reading difficulty. In *HLT-NAACL: Proceedings of the Main Conference*, pages 193–200.

Das, Sanjiv and Mike Chen. 2001. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific Finance Association Annual Conference (APFA)*.

Dave, Kushal, Steve Lawrence, and David M. Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of WWW*, pages 519–528.

Engström, Charlotta. 2004. Topic dependence in sentiment classification. Master's thesis, University of Cambridge.

Herbrich, Ralf, Thore Graepel, and Klaus Obermayer. 2000. Large margin rank boundaries for ordinal regression. In Alexander J. Smola, Peter L. Bartlett, Bernhard Schölkopf, and Dale Schuurmans, editors, *Advances in Large Margin Classifiers*, Neural Information Processing Systems. MIT Press, pages 115–132.

Horvitz, Eric, Andy Jacobs, and David Hovel. 1999. Attention-sensitive alerting. In *Proceedings of the Conference on Uncertainty and Artificial Intelligence*, pages 305–313.

Ishikawa, Hiroshi. 2003. Exact optimization for Markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10).

Ishikawa, Hiroshi and Davi Geiger. 1998. Occlusions, discontinuities, and epipolar lines in stereo. In *Proceedings of the 5th European Conference on Computer Vision (ECCV)*, volume I, pages 232–248, London, UK. Springer-Verlag.

Joachims, Thorsten. 1999. Making large-scale SVM learning practical. In Bernhard Schölkopf and Alexander Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, pages 44–56.

Kleinberg, Jon and Éva Tardos. 2002. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. *Journal of the ACM*, 49(5):616–639.

Koppel, Moshe and Jonathan Schler. 2005. The importance of neutral examples for learning sentiment. In *Workshop on the Analysis of Informal and Formal Information Exchange during Negotiations (FINEXIN)*.

Liu, Hugo, Henry Lieberman, and Ted Selker. 2003. A model of textual affect sensing using real-world knowledge. In *Proceedings of Intelligent User Interfaces (IUI)*, pages 125–132.

McCallum, Andrew. 1999. Multi-label text classification with a mixture model trained by EM. In *AAAI Workshop on Text Learning*.

McCullagh, Peter. 1980. Regression models for ordinal data. *Journal of the Royal Statistical Society*, 42(2):109–42.

Pang, Bo and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, pages 271–278.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86.

Qu, Yan, James Shanahan, and Janyce Wiebe, editors. 2004. *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*. AAAI Press. AAAI technical report SS-04-07.

Rifkin, Ryan M. and Aldebaro Klautau. 2004. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141.

Schapire, Robert E. and Yoram Singer. 2000. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

Smola, Alex J. and Bernhard Schölkopf. 1998. A tutorial on support vector regression. Technical Report NeuroCOLT NC-TR-98-030, Royal Holloway College, University of London.

Subasic, Pero and Alison Huettner. 2001. Affect analysis of text using fuzzy semantic typing. *IEEE Transactions on Fuzzy Systems*, 9(4):483–496.

Takamura, Hiroya, Yuji Matsumoto, and Hiroyasu Yamada. 2004. Modeling category structures with a kernel function. In *Proceedings of CoNLL*, pages 57–64.

Tong, Richard M. 2001. An operational system for detecting and tracking opinions in on-line discussion. SIGIR Workshop on Operational Text Classification.

Turney, Peter. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the ACL*, pages 417–424.

Vapnik, Vladimir. 1995. *The Nature of Statistical Learning Theory*. Springer.

Wilson, Theresa, Janyce Wiebe, and Rebecca Hwa. 2004. Just how mad are you? Finding strong and weak opinion clauses. In *Proceedings of AAAI*, pages 761–769.

Yu, Hong and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of EMNLP*.

Zhu, Xiaojin (Jerry). 2005. *Semi-Supervised Learning with Graphs*. Ph.D. thesis, Carnegie Mellon University.

A Appendix: other variations attempted

A.1 Discretizing binary classification

In our setting, we can also incorporate class relations by directly altering the output of a binary classifier, as follows. We first train a standard SVM, treating ratings greater than 0.5 as positive labels and others as negative labels. If we then consider the resulting classifier to output a *positivity-preference function* $\pi_+(x)$, we can then learn a series of thresholds to convert this value into the desired label set, under the assumption that the bigger $\pi_+(x)$ is, the more positive the review.[9] This algorithm always outperforms the majority-class baseline, but not to the degree that the best of SVM OVA and SVM regression does. Koppel and Schler (2005) independently found in a three-class study that thresholding a positive/negative classifier trained only on clearly positive or clearly negative examples did not yield large improvements.

A.2 Discretizing regression

In our experiments with SVM regression, we discretized regression output via a set of fixed decision thresholds $\{0.5, 1.5, 2.5, ...\}$ to map it into our set of class labels. Alternatively, we can learn the thresholds instead. Neither option clearly outperforms the other in the four-class case. In the three-class setting, the learned version provides noticeably better performance in two of the four datasets. But these results taken together still mean that in many cases, the difference is negligible, and if we had started down this path, we would have needed to consider similar tweaks for one-vs-all SVM as well. We therefore stuck with the simpler version in order to maintain focus on the central issues at hand.

[9] This is not necessarily true: if the classifier's goal is to optimize binary classification error, its major concern is to increase confidence in the positive/negative distinction, which may not correspond to higher confidence in separating "five stars" from "four stars".