Document-Word Co-Regularization for Semi-supervised Sentiment Analysis

Vikas Sindhwani and Prem Melville
IBM T. J. Watson Research Center, Yorktown Heights, NY 10598
{vsindhw,pmelvil}@us.ibm.com

Abstract

The goal of sentiment prediction is to automatically identify whether a given piece of text expresses positive or negative opinion towards a topic of interest. One can pose sentiment prediction as a standard text categorization problem, but gathering labeled data turns out to be a bottleneck. Fortunately, background knowledge is often available in the form of prior information about the sentiment polarity of words in a lexicon. Moreover, in many applications abundant unlabeled data is also available. In this paper, we propose a novel semi-supervised sentiment prediction algorithm that utilizes lexical prior knowledge in conjunction with unlabeled examples. Our method is based on joint sentiment analysis of documents and words based on a bipartite graph representation of the data. We present an empirical study on a diverse collection of sentiment prediction problems which confirms that our semi-supervised lexical models significantly outperform purely supervised and competing semi-supervised techniques.

1 Introduction

In recent years there has been an explosion of user-generated content on the Internet in the form of weblogs (blogs), discussion forums and online review sites. This phenomenon presents many new opportunities and challenges to both producers and consumers alike. For producer, this user-generated content provides a rich source of implicit consumer feedback. Tracking the pulse of this ever-expanding blogosphere, enables companies to discern what consumers are saying about their products, which provides useful insight on how to improve or market products better. For consumers, the plethora of information and opinions from diverse sources helps them tap into the wisdom of crowds, to aid in making more informed decisions. These decisions could range from which new digital camera to buy, which movie to watch, or even who to vote for in upcoming elections.

Though there is a vast quantity of information available, the consequent challenge is to be able to analyze the millions of blogs available, and to glean meaningful insights therein. One key component of this process is to be able to gauge the sentiment expressed in blogs around selected topics of interest. The emerging area of Sentiment Analysis (see e.g., [7, 4]) focuses on this task of automatically identifying whether a piece of text expresses a positive or negative opinion towards the subject matter. Detecting the sentiment expressed in documents turns out be an extremely difficult task, and the performance of sentiment classifiers can vary a great deal depending on the domain. As such, one of the grand challenges of sentiment analysis is to build a robust system that provides insights across a growing list of different products and topics of interest. Such a system needs to be able to rapidly adapt to new domains with minimal supervision.

Most prior work in sentiment analysis use knowledge-based approaches (see e.g., [4, 6]), that classify the sentiment of texts based on lexicons defining the sentiment-polarity of words, and simple linguistic patterns. There have been some recent studies that take a machine learning approach (e.g., [7]) and build text classifiers trained on documents that have been human-labeled as *positive* or *negative*. The knowledge-based approaches tend to be non-adaptive, while the learning approaches do not effectively exploit prior knowledge and require much effort through human annotation of documents. In this paper, we present a new machine learning approach that overcomes both drawbacks of previous learning approaches. Firstly, we incorporate prior knowledge of sentiment-laden terms directly into our model. Secondly, in order to adapt to new domains with minimal supervision, we also exploit the large amount of unlabeled data readily available. We present a unified framework in which lexical background information, unlabeled data and labeled training examples can be effectively combined. We demonstrate the generality of our approach, by presenting results on three, very different, domains — blogs discussing enterprise-software products, political blogs discussing US Presidential candidates, and online movie reviews.

2 Linear Sentiment Classification Models

In text classification, a document is typically represented as a bag-of-words feature vector $\boldsymbol{x} \in \mathcal{R}^D$. The entries in this vector usually specify frequencies, or weighted frequencies, for D words in a pre-specified vocabulary $\mathcal{V}$. Given such a representation, a linear classification model is specified by a weight vector $\mathbf{w} \in \mathcal{R}^D$ which defines the binary classification function $h(\boldsymbol{x}) = sign(\mathbf{w}^\top \boldsymbol{x})$[1]. We next discuss ways to set $\mathbf{w}$ given different kinds of information.

2.1 Unsupervised Lexical Classification

In the absence of any labeled data in a domain, one can build sentiment-classification models that rely solely on background knowledge, such as a lexicon defining the polarity of words. Suppose we are given a manually constructed lexicon of positive and negative terms which we denote by $\mathcal{V}_+ \subset \mathcal{V}$ and $\mathcal{V}_- \subset \mathcal{V}$ respectively. One straightforward approach to using this information is to measure the relative frequency of occurrence of positive (e.g., great and negative (e.g, terrible) terms in a document. The classification rule is then be given by $h(\boldsymbol{x}) = sign(\sum_{i \in \mathcal{V}_+} x_i - \sum_{i \in \mathcal{V}_-} x_i)$, which corresponds to the choice $w_i = +1$ for all $i \in \mathcal{V}_+$, $w_i = -1$ for all $i \in \mathcal{V}_-$, and $w_i = 0$ for all other terms.

For this study, we used a lexicon generated by the IBM India Research Labs that was developed for other text mining applications [8]. It contains 2,968 words that have been human-labeled as expressing positive or negative sentiment. In total, there are 1,267 positive and 1,701 negative unique terms after stemming. We eliminated terms that were ambiguous and dependent on context, such as dear and fine. It should be noted, that this list was constructed without a specific domain in mind; which is further motivation for using training examples and unlabeled data to learn domain-specific connotations.

2.2 Supervised Regularization Models

In a setting where l labeled documents, $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^l$, are available with $y_i \in \{+1, -1\}$, one may attempt to learn $\mathbf{w}$ by solving an optimization problem of the form, $\mathbf{w}^\star = \text{argmin}_{\mathbf{w}} \frac{1}{l} \sum_{i=1}^l V(\mathbf{w}^\top \boldsymbol{x}_i, y_i) + \frac{\gamma}{2}\|\mathbf{w}\|_2^2$ where $V(\cdot,\cdot)$ is a loss function and γ is a real-valued regularization parameter. For various choices for the loss function V, this optimization problem spans a large family of learning algorithms. Popular choices include the hinge loss: $V(\mathbf{w}^\top \boldsymbol{x}, y) = \max\left[0, 1 - y\mathbf{w}^\top \boldsymbol{x}\right]$, logistic loss: $V(\mathbf{w}^\top \boldsymbol{x}, y) = \log\left[1 + \exp(-y\mathbf{w}^\top \boldsymbol{x})\right]$ and the squared loss: $V(\mathbf{w}^\top \boldsymbol{x}, y) = \frac{1}{2}(\mathbf{w}^\top \boldsymbol{x} - y)^2$, which respectively lead to the Support Vector Machine (SVM), Logistic Regression and the classical Regularized Least Squares (RLS) algorithms

Our methods build on RLS due to its simplicity and excellent performance on classification tasks. In this case, the solution is given by the $D \times D$ linear system, $\left[\frac{1}{l}\mathbf{X}^\top\mathbf{X} + \gamma\mathbf{I}\right]\mathbf{w} = \frac{1}{l}\mathbf{X}^\top\mathbf{y}$, where $\mathbf{X}$ is the $l \times D$ data matrix whose rows are document vectors, and $\mathbf{y}$ is the vector of labels. Since documents almost always only contain a very small fraction of words in the vocabulary, the data matrix $\mathbf{X}$ is highly sparse. Due to this fact, the above linear system can be very efficiently solved for large-scale problems (where both l and D are large) using sparse iterative techniques such as Conjugate Gradient. Related techniques have also been used for large-scale implementations of linear logistic regression[2] and SVM models[3].

3 Semi-supervised Lexical Classification

In this section, we begin by first incorporating lexical knowledge in supervised learners. In Section 3.2 we extend this approach to also include unlabeled data.

3.1 Incorporating Lexical Knowledge

It is well-known that RLS may be interpreted as maximum a posteriori (MAP) estimation under a Gaussian likelihood model for errors $\left(y_i - \mathbf{w}^\top \boldsymbol{x}_i\right)$, and a zero-mean Gaussian prior for the weight parameters $\mathbf{w}$. A natural way to incorporate lexical prior knowledge is to assume a Gaussian prior for $\mathbf{w}$ with non-zero mean propotional to the lexical weight vector $\mathbf{w}_{lex}$. This immediately implies the following modified MAP estimation problem, $\text{argmin}_{\mathbf{w}} \frac{1}{l} \sum_{i=1}^l V(\mathbf{w}^\top \boldsymbol{x}_i, y_i) + \frac{\gamma}{2}\|\mathbf{w} - \nu\mathbf{w}_{\text{lex}}\|_2^2$, where ν is a parameter. It can be easily seen that the solution is given by the following modified linear system, $\left[\frac{1}{l}\mathbf{X}^\top\mathbf{X} + \gamma\mathbf{I}\right]\mathbf{w} = \frac{1}{l}\mathbf{X}^\top\mathbf{y} + \gamma\nu\mathbf{w}_{\text{lex}}$. We call this approach Lexical-RLS (LEX+RLS). The only difference above with respect to RLS is the second term of the right-hand-side which incorporates the lexical weights. Note that Lexical RLS reduces to RLS when ν is set to 0, and defines the the unsupervised lexical classifier (Section 2.1) when there are no labeled examples, or when $\gamma \to \infty$.

3.2 Incorporating Unlabeled Data

Suppose now that in addition to a lexicon labeled with sentiment polarity, we also have access to a large collection of unlabeled documents. Most semi-supervised classification algorithms implement the classical cluster assumption

[1] We assume $\mathbf{w}$ and $\boldsymbol{x}$ include a standard "bias" component.

[2] http://www.csie.ntu.edu.tw/∼cjlin/liblinear/

[3] See http://vikas-sindhwani.org/svmlin.html

which states the following: *if two documents are in the same cluster, they are likely to be of the same class*. Low-density techniques implement this assumption by attempting to find separators that do not cut through unlabeled data clusters. Similarly, Graph-based techniques use unlabeled examples to find classifiers that give smooth predictions on data clusters. See [1] and references therein for cluster assumption and overview of semi-supervised learning techniques.

In the presence of lexical knowledge we may further qualify the cluster assumption as follows: *if two documents are in the same cluster dominantly supported on positive (negative) sentiment words, they are likely to be positive (negative) sentiment documents*. In other words, the sentiment lexicon may be viewed as prior knowledge on the structure of the data clusters over which the cluster assumption ought to be enforced. Moreover, note that there is a clear duality between documents and words. The sentiment polarity of documents determines the polarity of words, while the polarity of words determines the polarity of documents. We now present a novel semi-supervised learning algorithm that simultaneously implements cluster assumptions for both documents and words while incorporating partial supervision along both dimensions.

We begin by introducing a bipartite graph representation of the data, previously utilized in the context of co-clustering [3]. We then formulate joint sentiment classification of documents and words in terms of transductive prediction on this graph whose nodes are viewed to be partially labeled. However, since this approach is strictly transductive and does not allow prediction on new completely unseen test documents, we formulate a new objective function that simultaneously projects the transductive solution to a linear model. We now outline these steps leading to the proposed algorithm.

Document-Word Bipartite Graph: In the semi-supervised setting, let $\mathbf{X}$ denote the $n \times D$ data matrix whose rows are the set of l labeled and $(n-l)$ unlabeled document vectors. Consider a bipartite graph, denoted by $\mathcal{G}$, with two sets of vertices: one corresponding to the n documents, and another corresponding to the D words in the vocabulary. Thus, $\mathcal{G}$ has $n+D$ vertices. An undirected edge (i,j) exists if the i^{th} document contains the j^{th} word. Since $\mathcal{G}$ is bipartite, there are no edges between words or between documents (though our formulation can be easily extended to incorporate intra-document and/or intra-word linkages). An edge signifies an association between a document and a word. By putting positive weights on the edges, we can capture the strength of this association. We use $\mathbf{X}_{ij}$ as the edge weight which corresponds to frequency (or idf-weighted frequency) of term j in document i. Then, the $(n+D) \times (n+D)$ adjacency matrix of $\mathcal{G}$ can be easily seen to be given by $\mathbf{A} = \begin{pmatrix} 0 & \mathbf{X} \\ \mathbf{X}^\top & 0 \end{pmatrix}$ where the vertices of the graph are ordered by taking the n documents (same order as rows of $\mathbf{X}$) followed by the D words (same order as columns of $\mathbf{X}$).

Transductive Sentiment Prediction: Next, we view $\mathcal{G}$ as a partially labeled graph. Given sentiment labels for a few document and word vertices, consider the problem of completing the labeling of the rest of the vertices of the graph. Such prediction problems on graphs have been well-studied in the graph-based semi-supervised learning literature (see [1] and references therein), but to the best of our knowledge they have never been applied to solve joint prediction problems on document and words. Our goal is to learn a real-valued sentiment-polarity score vector, $\boldsymbol{f}^d$, over document vertices and $\boldsymbol{f}^w$ over word vertices with the following properties: (a) If the i^{th} document is labeled, f_i^d should be close to the ± 1-valued label, (b) If the j^{th} word is labeled, f_j^w should be close to the ± 1-valued label and (c) If the association between the i^{th} document and the j^{th} word is strong, then f_i^d and f_j^w should be similar. It is important to note that the third property can be enforced also over unlabeled documents and unlabeled words. It turns out that the third property has close connections to the classical SVD applied to document-term matrices (see [3] for more details). These three properties can be enforced through the terms of the objective function in the following minimization problem,

$$\underset{\boldsymbol{f}^d, \boldsymbol{f}^w}{\operatorname{argmin}} \frac{1}{l_d} \sum_{i=1}^{l_d} V(f_i^d, y_i^d) + \frac{1}{l_w} \sum_{i=1}^{l_w} V(f_i^w, y_i^w) + \mu \sum_{i=1}^{n} \sum_{j=1}^{D} \mathbf{X}_{ij} \left(f_i^d - f_j^w\right)^2 \quad (1)$$

where V as before is a loss function, l_d is the number of labeled documents, l_w is the number of labeled words, μ is a real-valued parameter. We also assume that the first l_d documents in $\mathbf{X}$ of the n total are the ones that are labeled. The third term can be shown to be a quadratic form involving the graph Laplacian matrix $\mathbf{L}$ of $\mathcal{G}$, i.e., $\sum_{j=1}^{D} \mathbf{X}_{ij} \left(f_i^d - f_j^w\right)^2 = \left(\boldsymbol{f}^{d\top} \boldsymbol{f}^{w\top}\right) \mathbf{L} \begin{pmatrix} \boldsymbol{f}^d \\ \boldsymbol{f}^w \end{pmatrix}$. Here, $\mathbf{L} = \mathbf{D} - \mathbf{A}$ where $\mathbf{D}$ is the diagonal degree matrix associated with $\mathbf{A}$, i.e., $D_{rr} = \sum_s A_{rs}$. In particular, we use the associated *normalized* Graph Laplacian [2] in our formulations below, defined as $\tilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$. The solution to Eqn. 1 can be obtained by solving a sparse linear system (see [1] for details on Graph transduction in general).

Intuitive Interpretations: The transductive sentiment scores obtained by solving Eqn. 1 may be interpreted in different ways (see [1] for more discussion). The Random walk interpretation is as follows. Imagine starting from an *unlabeled* document i and walking to a word j in it with probability $\frac{\mathbf{X}_{ij}}{\sum_j X_{ij}}$. Then, from j we walk to an-

other document k with probability $\frac{\mathbf{X}_{kj}}{\sum_k X_{kj}}$. Continuing in this way bouncing between documents and words until a labeled node (document or word) is found, one can ask for the probability p of terminating the random walk at a positive sentiment document or word. The score given to the unlabeled document i is then $2p-1$. Similarly, the random walk may be started from an unlabeled word to obtain a sentiment polarity score for that word. Another interpretation is the following: Consider $\mathcal{G}$ to be an electric network. Imagine connecting positive sentiment documents and words to a positive voltage source (+1V) and negative sentiment documents and words to a negative voltage source (-1V). Let X_{ij} be the conductance (inverse of resistance) between a document i and a word j. Then the sentiment score given to an *unlabeled* document or a word is the resulting voltage at that node in this electric network. Strictly speaking, these interpretations hold when the scores for labeled nodes are clamped at the labels while the third term in Eqn. 1 is minimized (this corresponds to the limiting solution of Eqn. 1 when $\mu \rightarrow 0$).

Other Smoothness Operators: The Laplacian matrix $\tilde{L}$ defines a large family of smoothness operators on functions defined over the vertices of the corresponding graph. We point the reader to [11] for typical choices of graph-regularizers. In subsequent discussion, we use $\mathbf{M}$ to denote a generic graph regularizer derived from the Laplacian, typically in the form of a power series in $\tilde{L}$. We use iterated Laplacians of the form $\mathbf{M} = \tilde{L}^p$ where p is an integer parameter.

Out-of-Sample Prediction: Note that while $\boldsymbol{f}^d, \boldsymbol{f}^w$ provide sentiment polarity predictions for unlabeled documents and words, they do not provide a model that can be applied to unseen test data. To obtain a linear model, we propose a novel formulation that comprises of solving the following minimization problem,

$$\begin{aligned}\underset{\boldsymbol{f}^d, \boldsymbol{f}^w, \mathbf{w}}{\operatorname{argmin}} \; & \frac{\mu}{2(n+D)} \begin{pmatrix} \boldsymbol{f}^{d\top} \boldsymbol{f}^{w\top} \end{pmatrix} \mathbf{M} \begin{pmatrix} \boldsymbol{f}^d \\ \boldsymbol{f}^w \end{pmatrix} \\ & + \frac{1}{l_d} \sum_{i=1}^{l} V(f_i^d, y_i^d) + \frac{1}{l_w} \sum_{i=1}^{l} V(f_i^w, y_i^w) \\ & + \frac{1}{2n} \sum_{i=1}^{n} \left(\mathbf{w}^\top \boldsymbol{x}_i - f_i^d \right)^2 + \frac{\gamma}{2} \| \mathbf{w} - \nu \boldsymbol{f}^w \|_2^2 \end{aligned} \tag{2}$$

The first four terms are inspired by the transductive formulation in Eqn. 1. The last two terms couple transductive learning with a linear model. In particular, through these terms we enforce the following: (a) the outputs produced by the linear model on documents, $\mathbf{w}^\top \boldsymbol{x}_i$, should be close to the transductive solution on document vertices, f_i^d, and (b) the sentiment polarity of words as expressed through $\boldsymbol{f}^w$ should now effectively act as the non-zero prior for the weights of the linear model.

Proposed Algorithm: Let $\boldsymbol{y}^d$ denote the $n \times 1$ label vector for documents with entry 0 for unlabeled documents. Similarly, let $\boldsymbol{y}^w$ denote the $D \times 1$ label vector for words with entry 0 for unlabeled words (words not in the sentiment lexicon). Choosing V to be the squared loss, Eqn. 2 poses the problem of minimizing an unconstrained quadratic. This reduces to solving the following linear system of size $(n+2D) \times (n+2D)$:

$$\mathbf{Q} \begin{pmatrix} \boldsymbol{f}^d \\ \boldsymbol{f}^w \\ \mathbf{w} \end{pmatrix} = \begin{pmatrix} \frac{1}{l_d} \boldsymbol{y}^d \\ \frac{1}{l_w} \boldsymbol{y}^w \\ 0 \end{pmatrix} \tag{3}$$

where $\mathbf{Q} = \frac{\mu}{n+D} \begin{pmatrix} \mathbf{M} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} \mathbf{I} & 0 & -\mathbf{X} \\ 0 & 0 & 0 \\ -\mathbf{X}^\top & 0 & \mathbf{X}^\top \mathbf{X} \end{pmatrix} + \gamma \begin{pmatrix} 0 & 0 & 0 \\ 0 & \nu^2 \mathbf{I} & -\nu \mathbf{I} \\ 0 & -\nu \mathbf{I} & \mathbf{I} \end{pmatrix} + diag(\frac{1}{l_d}[\boldsymbol{y}^d \neq 0], \frac{1}{l_w}[\boldsymbol{y}^w \neq 0], 0)$

where the elements of $[\boldsymbol{y}^d \neq 0]$ equal 1 for indices corresponding to labeled documents and 0 otherwise. Above, we use $\mathbf{I}$ and 0 to denote identity and zero matrices of appropriate size. We solve the linear system in Eqn. 3 using the Conjugate Gradient (CG) method with a tolerance of $\epsilon = 0.0001$. Note that neither $\mathbf{Q}$ nor $\mathbf{M}$ need to be explicitly computed and stored. Rather, since CG only accesses $\mathbf{Q}$ through matrix-vector multiplication of the form $\boldsymbol{v} = \mathbf{Q}\boldsymbol{u}$, we compute this product efficiently on the fly using just the data matrix and the document-word label vectors. To obtain the exact solution, theoretically $n + D$ CG iterations are needed. However, very high quality approximate solutions are obtained extremely quickly (convergence depends on the practical rank of Q) in practice. We call our approach Semi-supervised Lexical Regularized Least Squares (SS+LEX+RLS) classification.

Advantages of the Proposed Algorithm: Unlike Transductive SVMs [5] our algorithm is based on convex optimization and therefore does not suffer from local minima issues. Unlike, typical graph-based methods which require an expensive construction of a nearest neighbor graph, our algorithm uses regularization operators defined on the bipartite document-word graph. Thus, there is no expensive graph construction step. To the best of our knowledge, our algorithm is the first semi-supervised method that attempts to simultaneously implement cluster assumptions along both dimensions of the data matrix and incorporates both labeled examples as well as labeled features. Joint document-word analysis has previously been explored in the context of co-clustering in [3]. Our algorithm may be seen as providing two additional capabilities on top of the bipartite co-clustering approach: (a) semi-supervision for both document and words, and (b) out-of-sample predictions through a linear model.

4 Empirical Study

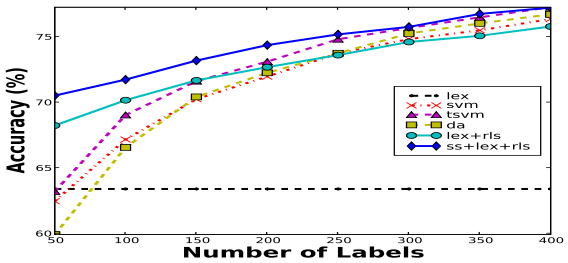
In order to test the generality of our approach we experimented on three qualitatively different domains. We used the MOVIES dataset provided by Pang et al. [7], which consists of 1000 positive and 1000 negative review. We also constructed two blog datasets as described below.

Lotus blogs: We created a data set targeted at detecting sentiment around enterprise software, specifically pertaining to the IBM Lotus brand. The LOTUS data set consists of posts from 14 individual blogs, 4 of which are actively posting negative content on the brand, with the rest tending to write more positive or neutral posts. The data was collected by downloading the latest posts from each blogger's RSS feeds, or accessing the blog's archives. A labeled set of 34 positive and 111 negative examples was manually obtained. In addition, we also generated an unlabeled set by randomly sampling 2000 posts from a universe of 14,258 blogs that discuss issues relevant to Lotus software.

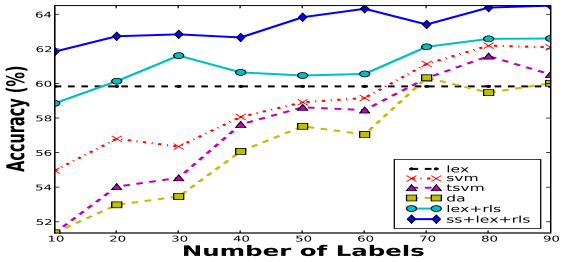
Political candidate blogs: For our second blog domain, we used data gathered from 16,742 political blogs, which contain over 500,000 posts. A post was labeled as having positive or negative sentiment about a specific candidate (Barack Obama or Hillary Clinton) if it explicitly mentioned the candidate in positive or negative terms. The manually labeled POLITICAL data set consisted of 49 positive and 58 negative posts. We created an additional set of 2000 unlabeled examples, that were sampled from all available posts from our political blogs. This unlabeled set contains 1000 posts containing the term "clinton" and 1000 containing "obama" in their URLs.

Methodology: We compare the approaches proposed in this paper: the lexical RLS (LEX+RLS) and the semi-supervised lexical RLS (SS+LEX+RLS), to the following: (a) unsupervised lexical classification (LEX) which gives a baseline, (b) Linear SVMs which are considered state-of-the-art for text classification, and (c) two implementations of the Transductive SVM [5, 10], one based on label switching (TSVM) and another based on deterministic annealing (DA) [10]. We carefully tune the regularization parameter for linear SVMs (in the range $\gamma = \frac{c}{l}$ where $c = \{0.001, 0.01, 0.1, 1, 10, 100\}$ and l is the number of labeled examples) to optimize test performance. Therefore, their performance reported here is meant to represent the best possible results one can hope to obtain with a state-of-the-art purely supervised learner. We report the best performance of TSVM and DA over the parameter settings used in [10]. Furthermore, TSVM and DA require an accurate estimate of positive class fraction. In practical semi-supervised settings, a noisy estimate of this fraction is obtained from the labeled data. In our experimental setting,

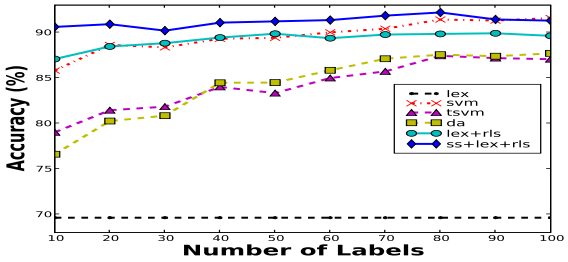
Figure 1. Learning Curves



(a) Performance on MOVIES



(b) Performance on POLITICAL



(c) Performance on LOTUS

we confer an advantage to TSVM and DA by setting the positive class fraction to the true value. For SS+LEX+RLS, we need to set the following parameters: γ, μ, ν and p, the degree of the iterated graph Laplacian $\mathbf{M} = \tilde{\mathbf{L}}^p$. For all datasets, we used $p = 10$. We used $\gamma = 0.0001, \nu = 1.0, \mu = 10$ for MOVIES and $\gamma = 0.001, \nu = 0.1, \mu = 1$ for both POLITICAL and LOTUS. A careful optimization of these parameters may further improve the results presented here. We generated learning curves averaged over 10 runs of 10-fold cross-validation. In the semi-supervised setting this experimental protocol needs more explanation. Let U be the set of truly unlabeled examples in the dataset, as we have in POLITICAL and LOTUS . Let L denote the labeled set. In each of the 10 training-test splits in one run of 10-

fold cross-validation, we partition L into L_{train} and L_{test} in the ratio 9:1. Next we take only a subset L_{lab} of L_{train} as labeled data and study the effect of gradually increasing the size of L_{lab}. Semi-supervised algorithms are provided $(L_{train} - L_{lab}) \cup U$ as the unlabeled set. Supervised algorithms only use L_{lab}. The linear models given by various algorithms are then evaluated on L_{test}. The resulting learning curves are shown in Figure 1.

Results: It is clear from Figure 1 that by utilizing both lexical prior knowledge as well as unlabeled data SS+LEX+RLS significantly outperforms all competing alternatives on all datasets. As expected the smaller the labeled set, the larger the performance boost. On MOVIES, we see that with 50 labeled examples the SVM, TSVM and DA perform no better than the unsupervised lexical classifier. On the other hand, by simply combining these few labeled examples with lexical information, LEX+RLS already gives better performance. Finally, by further including unlabeled data SS+LEX+RLS gives by-far the best performance. Similar observations hold on LOTUS and POLITICAL. Surprisingly, on those datasets TSVM and DA turn out to perform worse than an SVM. Even if suboptimal local minima issues for TSVM and DA are kept aside, when labeled examples are extremely scarce the low-density separators found by these algorithms may not be sufficiently constrained. We conjecture that the blogosphere consists of clusters of bloggers focusing on similar sub-topics while the range of topics is very diverse (e.g., "iraq war" versus "health care"). This implies that without additional labeled data or prior knowledge such as what the lexicon provides, one may find good quality low-density decision boundaries that end up better separating topical sub-clusters as opposed to sentiment classes.

The unsupervised lexical classifier does not perform well, particularly on MOVIES and LOTUS. The underlying assumption of the Lexical Classifier is that a document is positive if there are more positive lexicon terms than negative terms in a document. Apart from the fact that the lexicon does not cover all terms that may appear in our vocabulary, it also does not capture domain-specific connotations of a term. We claim that semi-supervised learning can radically update our knowledge about the sentiment polarity of terms, beyond what can be captured by a limited labeled set. We can support this claim by examining the elements of our lexical background knowledge that have been altered by our semi-supervised model. Such insight can be easily gathered by comparing the sentiment polarity score $\boldsymbol{f}^w$ with the lexical labels $\mathbf{w}_{lex}$. Below are the top 20 stemmed lexicon terms for MOVIES sorted by $-w_{lex_i} f_i^w$ for a model trained with 400 labels and 1400 unlabeled examples; this set constitutes the terms that have changed most dramatically in sentiment: lone, origin, basic, show, revolut, pretti, know, reason, hatr, doubt, captur, complet, complex, talent, upset, secur, call, debat, critic, plain. This analysis gives us some insight into the domain-specificity of the sentimentality of certain terms, which is not possible to encode into a single general-purpose lexicon. For example, words such as revolution, capture and complex can be associated with positive experiences in descriptions of movies, though they may be generally considered negative in other contexts. The down-weighting of positive lexicon terms, such as talent for MOVIES is also consistent with the "thwarted expectation" narratives that Pang et al. [7] observed in this data.

5 Conclusion

We have proposed a general framework for incorporating lexical information as well as unlabeled data within standard regularized least squares for sentiment prediction tasks. Our methods and applications can be immediately extended to the following: (1) a variety of classification problems where partial supervision may be available in the form of a few labeled examples *and features*, (2) a large choice of loss functions such as those defining SVMs and logistic regression, (3) additional graph structures on documents (e.g., in the form of web hyperlinks) and words (e.g, with edges connecting synonyms) and (4) non-linear kernel-based generalizations (see [9]). These are topics of future research. A longer version of this paper is available at http://vikas-sindhwani.org/icdm08-sentiment.pdf.

References

[1] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, 2006.

[2] F. Chung. *Spectral Graph Theory*. AMS, 1997.

[3] I. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD*, 2001.

[4] M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD*, pages 168–177, 2004.

[5] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, 1999.

[6] S.-M. Kim and E. Hovy. Determining the sentiment of opinions. In *Proceedings of International Conference on Computational Linguistics*, 2004.

[7] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *EMNLP*, 2002.

[8] G. Ramakrishnan, A. Jadhav, A. Joshi, S. Chakrabarti, and P. Bhattacharyya. Question answering via bayesian inference on lexical relations. In *ACL*, pages 1–10, 2003.

[9] V. Sindhwani, J. Hu, and A. Mojsilovic. Regularized co-clustering with dual supervision. In *NIPS*, volume 21, 2008.

[10] V. Sindhwani and S. Keerthi. Large scale semi-supervised linear SVMs. In *SIGIR*, 2006.

[11] A. Smola and I. Kondor. Kernels and regularization on graphs. In *COLT*, 2003.