Soft-Supervised Learning for Text Classification

Amarnag Subramanya & Jeff Bilmes
Dept. of Electrical Engineering,
University of Washington, Seattle, WA 98195, USA.
{asubram,bilmes}@ee.washington.edu

Abstract

We propose a new graph-based semi-supervised learning (SSL) algorithm and demonstrate its application to document categorization. Each document is represented by a vertex within a weighted undirected graph and our proposed framework minimizes the weighted Kullback-Leibler divergence between distributions that encode the class membership probabilities of each vertex. The proposed objective is convex with guaranteed convergence using an alternating minimization procedure. Further, it generalizes in a straightforward manner to multi-class problems. We present results on two standard tasks, namely Reuters-21578 and WebKB, showing that the proposed algorithm significantly outperforms the state-of-the-art.

1 Introduction

Semi-supervised learning (SSL) employs small amounts of labeled data with relatively large amounts of unlabeled data to train classifiers. In many problems, such as speech recognition, document classification, and sentiment recognition, annotating training data is both time-consuming and tedious, while unlabeled data are easily obtained thus making these problems useful applications of SSL. Classic examples of SSL algorithms include self-training (Yarowsky, 1995) and co-training (Blum and Mitchell, 1998). Graph-based SSL algorithms are an important class of SSL techniques that have attracted much of attention of late (Blum and Chawla, 2001; Zhu et al., 2003). Here one assumes that the data (both labeled and unlabeled) is embedded within a low-dimensional manifold expressed by a graph. In other words, each data sample is represented by a vertex within a weighted graph with the weights providing a measure of similarity between vertices.

Most graph-based SSL algorithms fall under one of two categories – those that use the graph structure to spread labels from labeled to unlabeled samples (Szummer and Jaakkola, 2001; Zhu and Ghahramani, 2002) and those that optimize a loss function based on smoothness constraints derived from the graph (Blum and Chawla, 2001; Zhu et al., 2003; Joachims, 2003; Belkin et al., 2005). Sometimes the two categories are similar in that they can be shown to optimize the same underlying objective (Zhu and Ghahramani, 2002; Zhu et al., 2003). In general graph-based SSL algorithms are non-parametric and transductive.[1] A learning algorithm is said to be transductive if it is expected to work only on a closed data set, where a test set is revealed at the time of training. In practice, however, transductive learners can be modified to handle unseen data (Zhu, 2005a; Sindhwani et al., 2005). A common drawback of many graph-based SSL algorithms (e.g. (Blum and Chawla, 2001; Joachims, 2003; Belkin et al., 2005)) is that they assume binary classification tasks and thus require the use of sub-optimal (and often computationally expensive) approaches such as one vs. rest to solve multi-class problems, let alone structured domains such as strings and trees. There are also issues related to degenerate solutions (all unlabeled samples classified as belonging to a single

[1] Excluding Manifold Regularization (Belkin et al., 2005).

class) (Blum and Chawla, 2001; Joachims, 2003; Zhu and Ghahramani, 2002). For more background on graph-based and general SSL and their applications, see (Zhu, 2005a; Chapelle et al., 2007; Blitzer and Zhu, 2008).

In this paper we propose a new algorithm for graph-based SSL and use the task of text classification to demonstrate its benefits over the current state-of-the-art. Text classification involves automatically assigning a given document to a fixed number of semantic categories. Each document may belong to one, many, or none of the categories. In general, text classification is a *multi-class* problem (more than 2 categories). Training fully-supervised text classifiers requires large amounts of labeled data whose annotation can be expensive (Dumais et al., 1998). As a result there has been interest is using SSL techniques for text classification (Joachims, 1999; Joachims, 2003). However past work in semi-supervised text classification has relied primarily on one vs. rest approaches to overcome the inherent multi-class nature of this problem. We believe such an approach may be sub-optimal because, disregarding data overlap, the different classifiers have training procedures that are independent of one other. In order to address the above drawback we propose a new framework based on optimizing a loss function composed of Kullback-Leibler divergence (KL-divergence) (Cover and Thomas, 1991) terms between probability distributions defined for each graph vertex. The use of probability distributions, rather than fixed integer labels, not only leads to a straightforward multi-class generalization, but also allows us to exploit other well-defined functions of distributions, such as entropy, to improve system performance and to allow for the measure of uncertainty. For example, with a single integer, at most all we know is its assignment. With a distribution, we can continuously move from knowing an assignment with certainty (i.e., an entropy of zero) to expressions of doubt or multiple valid possibilities (i.e., an entropy greater than zero). This is particularly useful for document classification as we will see. We also show how one can use the alternating minimization (Csiszar and Tusnady, 1984) algorithm to optimize our objective leading to a relatively simple, fast, easy-to-implement, guaranteed to converge, iterative, and closed form update for each iteration.

2 Proposed Graph-Based Learning Framework

We consider the transductive learning problem, i.e., given a training set $\mathcal{D} = \{\mathcal{D}_l, \mathcal{D}_u\}$, where $\mathcal{D}_l$ and $\mathcal{D}_u$ are the sets of labeled and unlabeled samples respectively, the task is to infer the labels for the samples in $\mathcal{D}_u$. In other words, $\mathcal{D}_u$ is the "test-set." Here $\mathcal{D}_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, $\mathcal{D}_u = \{\mathbf{x}_i\}_{i=l+1}^{l+u}$, $\mathbf{x}_i \in \mathrm{X}$ (the input space of the classifier, and corresponds to vectors of features) and $y_i \in \mathrm{Y}$ (the space of classifier outputs, and for our case is the space of non-negative integers). Thus $|\mathrm{Y}| = 2$ yields binary classification while $|\mathrm{Y}| > 2$ yields multi-class. We define $n = l + u$, the total number of samples in the training set. Given $\mathcal{D}$, most graph-based SSL algorithms utilize an undirected weighted graph $\mathcal{G} = (V, E)$ where $V = \{1, \ldots, n\}$ are the data points in $\mathcal{D}$ and $E = V \times V$ are the set of undirected edges between vertices. We use $w_{ij} \in \mathbf{W}$ to denote the weight of the edge between vertices i and j. $\mathbf{W}$ is referred to as the weight (or affinity) matrix of $\mathcal{G}$. As will be seen shortly, the input features $\mathbf{x}_i$ effect the final classification results via $\mathbf{W}$, i.e., the graph. Thus graph construction is crucial to the success of any graph-based SSL algorithm. Graph construction "is more of an art, than science" (Zhu, 2005b) and is an active research area (Alexandrescu and Kirchhoff, 2007). In general the weights are formed as $w_{ij} = \mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j)\delta(j \in \mathcal{K}(i))$. Here $\mathcal{K}(i)$ is the set of i's k-nearest-neighbors ($\mathcal{K}$NN), $\mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j)$ is a given measure of similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\delta(c)$ returns a 1 if c is true and 0 otherwise. Getting the similarity measure right is crucial for the success of any SSL algorithm as that is what determines the graph. Note that setting $\mathcal{K}(i) = |V| = n$ results in a fully-connected graph. Some popular similarity measures include

$$\mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma^2}} \text{ or}$$

$$\mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j) = \cos(\mathbf{x}_i, \mathbf{x}_j) = \frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{\| \mathbf{x}_i \|_2^2 \| \mathbf{x}_j \|_2^2}$$

where $\| \mathbf{x}_i \|_2$ is the $\mathcal{L}_2$ norm, and $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ is the inner product of $\mathbf{x}_i$ and $\mathbf{x}_j$. The first similarity measure is an RBF kernel applied on the squared Euclidean distance while the second is cosine similarity. In this paper *all* graphs are constructed using cosine similarity.

We next introduce our proposed approach. For every $i \in V$, we define a probability distribution p_i over the elements of Y. In addition let r_j, $j = 1 \ldots l$ be another set of probability distributions again over the elements of Y (recall, Y is the space of classifier outputs). Here $\{r_j\}_j$ represents the labels of the supervised portion of the training data. If the label for a given labeled data point consists only of a single integer, then the entropy of the corresponding r_j is zero (the probability of that integer will be unity, with the remaining probabilities being zero). If, on the other hand, the "label" for a given labeled data point consists of a set of integers (e.g., if the object is a member of multiple classes), then r_j is able to represent this property accordingly (see below). We emphasize again that both p_i and r_j are probability distributions, with r_j fixed throughout training. The goal of learning in this paper is to find the best set of distributions $p_i, \forall i$ that attempt to: 1) agree with the labeled data r_j wherever it is available; 2) agree with each other (when they are close according to a graph); and 3) be smooth in some way. These criteria are captured in the following *new* multi-class SSL optimization procedure:

$$\min_{\mathrm{p}} C_1(\mathrm{p}), \text{ where } C_1(\mathrm{p}) = \left[\sum_{i=1}^{l} D_{KL}(r_i || p_i) + \mu \sum_{i}^{n} \sum_{j} w_{ij} D_{KL}(p_i || p_j) - \nu \sum_{i=1}^{n} H(p_i) \right], \tag{1}$$

and where $\mathrm{p} \triangleq (p_1, \ldots, p_n)$ denotes the entire set of distributions to be learned, $H(p_i) = -\sum_{\mathrm{y}} p_i(\mathrm{y}) \log p_i(\mathrm{y})$ is the standard Shannon entropy function of p_i, $D_{KL}(p_i||q_j)$ is the KL-divergence between p_i and q_j, and μ and ν are hyperparameters whose selection we discuss in section 5. The distributions r_i are derived from $\mathcal{D}_l$ (as mentioned above) and this can be done in one of the following ways: (a) if $\hat{y}_i$ is the single supervised label for input $\mathbf{x}_i$ then $r_i(y) = \delta(y = \hat{y}_i)$, which means that r_i gives unity probability for y equaling the label $\hat{y}_i$; (b) if $\hat{y}_i = \{\hat{y}_i^{(1)}, \ldots, \hat{y}_i^{(k)}\}$, $k \leq |\mathrm{Y}|$ is a *set* of possible outputs for input $\mathbf{x}_i$, meaning an object validly falls into all of the corresponding categories, we set $r_i(y) = (1/k)\delta(y \in \hat{y}_i)$ meaning that r_i is uniform over only the possible categories and zero otherwise; (c) if the labels are somehow provided in the form of a set of non-negative scores, or even a probability distribution itself, we just set r_i to be equal to those scores (possibly) normalized to become a valid probability distribution. Among these three cases, case (b) is particularly relevant to text classification as a given document many belong to (and in practice may be labeled as) many classes. The final classification results, i.e., the final labels for $\mathcal{D}_u$, are then given by $\hat{y} = \operatorname{argmax}_{\mathrm{y} \in \mathrm{Y}} p_i(\mathrm{y})$.

We next provide further intuition on our objective function. SSL on a graph consists of finding a labeling $\mathcal{D}_u$ that is consistent with both the labels provided in $\mathcal{D}_l$ and the geometry of the data induced by the graph. The first term of C_1 will penalize the solution p_i $i \in \{1, \ldots, l\}$, when it is far away from the labeled training data $\mathcal{D}_l$, but it does not insist that $p_i = r_i$, as allowing for deviations from r_i can help especially with noisy labels (Bengio et al., 2007) or when the graph is extremely dense in certain regions. As explained above, our framework allows for the case where supervised training is uncertain or ambiguous. We consider it reasonable to call our approach *soft-supervised* learning, generalizing the notion of semi-supervised learning, since there is even more of a continuum here between fully supervised and fully unsupervised learning than what typically exists with SSL. Soft-supervised learning allows uncertainty to be expressed (via a probability distribution) about any of the labels individually.

The second term of C_1 penalizes a lack of consistency with the geometry of the data and can be seen as a graph regularizer. If w_{ij} is large, we prefer a solution in which p_i and p_j are close in the KL-divergence sense. While KL-divergence is asymmetric, given that $\mathcal{G}$ is undirected implies $\mathbf{W}$ is symmetric ($w_{ij} = w_{ji}$) and as a result the second term is inherently symmetric.

The last term encourages each p_i to be close to the uniform distribution if not preferred to the contrary by the first two terms. This acts as a guard against degenerate solutions commonly encountered in SSL (Blum and Chawla, 2001; Joachims, 2003). For example, consider the case where part of the graph is almost completely disconnected from any labeled vertex (which is possible in the k-nearest neighbor case). In such situations the third term en-

sures that the nodes in this disconnected region are encouraged to yield a uniform distribution, validly expressing the fact that we do not know the labels of these nodes based on the nature of the graph. More generally, we conjecture that by maximizing the entropy of each p_i, the classifier has a better chance of producing high entropy results in graph regions of low confidence (e.g. close to the decision boundary and/or low density regions). This overcomes a common drawback of a large number of state-of-the-art classifiers that tend to be confident even in regions close to the decision boundary.

We conclude this section by summarizing some of the features of our proposed framework. It should be clear that C_1 uses the "manifold assumption" for SSL (see chapter 2 in (Chapelle et al., 2007)) — it assumes that the input data can be embedded within a low-dimensional manifold (the graph). As the objective is defined in terms of probability distributions over integers rather than just integers (or to real-valued relaxations of integers (Joachims, 2003; Zhu et al., 2003)), the framework generalizes in a straightforward manner to multi-class problems. Further, all the parameters are estimated jointly (compare to one vs. rest approaches which involve solving $|\mathrm{Y}|$ independent problems). Furthermore, the objective is capable of handling label training data uncertainty (Pearl, 1990). Of course, this objective would be useless if it wasn't possible to efficiently and easily optimize it on large data sets. We next describe a method that can do this.

3 Learning with Alternating Minimization

As long as $\mu, \nu \geq 0$, the objective $C_1(\mathrm{p})$ is convex. This follows since $D_{KL}(p_i||p_j)$ is convex in the pair (p_i, p_j) (Cover and Thomas, 1991), negative entropy is convex, and a positive-weighted linear combination of a set of convex functions is convex. Thus, the problem of minimizing C_1 over the space of collections of probability distributions (a convex set) constitutes a convex programming problem (Bertsekas, 2004). This property is *extremely* beneficial since there is a unique global optimum and there are a variety of methods that can be used to yield that global optimum. One possible method might take the derivative of the objective along with Lagrange multipliers to ensure that we stay within the space of probability distributions. This method can sometimes yield a closed form single-step analytical expression for the globally optimum solution. Unfortunately, however, our problem does not admit such a closed form solution because the gradient of $C_1(\mathrm{p})$ with respect to $p_i(\mathrm{y})$ is of the form, $k_1 p_i(\mathrm{y}) \log p_i(\mathrm{y}) + k_2 p_i(\mathrm{y}) + k_3$ (where $k_1,\ k_2,\ k_3$ are fixed constants). Sometimes, optimizing the dual of the objective can also produce a solution, but unfortunately again the dual of our objective also does not yield a closed form solution. The typical next step, then, is to resort to iterative techniques such as gradient descent along with modifications to ensure that the solution stays within the set of probability distributions (the gradient of C_1 alone will not necessarily point in the direction where p is still a valid distribution) - one such modification is called the method of multipliers (MOM). Another solution would be to use computationally complex (and complicated) algorithms like interior point methods (IPM). While all of the above methods (described in detail in (Bertsekas, 2004)) are feasible ways to solve our problem, they each have their own drawbacks. Using MOM, for example, requires the careful tuning of a number of additional parameters such as learning rates, growth factors, and so on. IPM involves inverting a matrix of the order of the number of variables and constraints during each iteration.

We instead adopt a different strategy based on alternating minimization (Csiszar and Tusnady, 1984). This approach has a single additional optimization parameter (contrasted with MOM), admits a closed form solution for each iteration not involving any matrix inversion (contrasted with IPM), and yields guaranteed convergence to the global optimum. In order to render our approach amenable to AM, however, we relax our objective C_1 by defining a new (third) set of distributions for all training samples q_i, $i = 1, \ldots, n$ denoted collectively like the above using the notation $\mathrm{q} \triangleq (q_1, \ldots, q_n)$. We define a new objective to be optimized as follows:

$$\min_{\mathrm{p,q}} C_2(\mathrm{p}, \mathrm{q}), \text{ where } C_2(\mathrm{p}, \mathrm{q}) = \left[\sum_{i=1}^{l} D_{KL}(r_i||q_i) + \mu \sum_{i=1}^{n} \sum_{j \in \mathcal{N}(i)} w'_{ij} D_{KL}(p_i||q_j) - \nu \sum_{i=1}^{n} H(p_i) \right].$$

Before going further, the reader may be wondering at this juncture how might it be desirable for us to have apparently complicated the objective function in an attempt to yield a more computationally and methodologically superior machine learning procedure. This is indeed the case as will be spelled out below. First, in C_2 we have defined a new weight matrix $[W']_{ij} = w'_{ij}$ of the same size as the original where $W' = W + \alpha \mathbf{I}_n$, where $\mathbf{I}_n$ is the $n \times n$ identity matrix, and where $\alpha \geq 0$ is a non-negative constant (this is the optimization related parameter mentioned above). This has the effect that $w'_{ii} \geq w_{ii}$. In the original objective C_1, w_{ii} is irrelevant since $D_{KL}(p||p) = 0$ for all p, but since there are now two distributions for each training point, there should be encouragement for the two to approach each other. Like C_1, the first term of C_2 ensures that the labeled training data is respected and the last term is a smoothness regularizer, but these are done via different sets of distributions, q and p respectively — this choice is what makes possible the relatively simple analytical update equations given below. Next, we see that the two objective functions in fact have identical solutions when the optimization enforces the constraint that p and q are equal:

$$\min_{(\mathrm{p},\mathrm{q}):\mathrm{p}=\mathrm{q}} C_2(\mathrm{p},\mathrm{q}) = \min_{\mathrm{p}} C_1(\mathrm{p}).$$

Indeed, as α gets large, the solutions considered viable are those only where $\mathrm{p} = \mathrm{q}$. We thus have that:

$$\lim_{\alpha\to\infty} \min_{\mathrm{p},\mathrm{q}} C_2(\mathrm{p},\mathrm{q}) = \min_{\mathrm{p}} C_1(\mathrm{p}).$$

Therefore, the two objectives should yield the same solution as long as $\alpha \geq w_{ij}$ for all i, j. A key advantage of this relaxed objective is that it is amenable to alternating minimization, a method to produce a sequence of sets of distributions $(\mathrm{p}^n, \mathrm{q}^n)$ as follows:

$$\mathrm{p}^n = \operatorname*{argmin}_{\mathrm{p}} C_2(\mathrm{p},\mathrm{q}^{n-1}), \;\; \mathrm{q}^n = \operatorname*{argmin}_{\mathrm{q}} C_2(\mathrm{p}^n,\mathrm{q}).$$

It can be shown (we omit the rather lengthy proof due to space constraints) that the sequence generated using the above minimizations converges to the minimum of $C_2(\mathrm{p},\mathrm{q})$, i.e.,

$$\lim_{n\to\infty} C_2(\mathrm{p}^{(n)},\mathrm{q}^{(n)}) = \inf_{\mathrm{p},\mathrm{q}} C_2(\mathrm{p},\mathrm{q}),$$

provided we start with a distribution that is initialized properly $q^{(0)}(y) > 0 \; \forall \; y \in \mathrm{Y}$. The update equations for $\mathrm{p}^{(n)}$ and $\mathrm{q}^{(n)}$ are given by

$$p_i^{(n)}(y) = \frac{1}{\mathcal{Z}_i} \exp^{\frac{\beta_i^{(n-1)}(y)}{\gamma_i}},$$

$$q_i^{(n)}(y) = \frac{r_i(y)\delta(i \leq l) + \mu \sum_j w'_{ji} p_j^{(n)}(y)}{\delta(i \leq l) + \mu \sum_j w'_{ji}},$$

where

$$\gamma_i = \nu + \mu \sum_j w'_{ij},$$

$$\beta_i^{(n-1)}(y) = -\nu + \mu \sum_j w'_{ij} (\log q_j^{(n-1)}(y) - 1)$$

and where $\mathcal{Z}_i$ is a normalizing constant to ensure p_i is a valid probability distribution. Note that each iteration of the proposed framework has a closed form solution and is relatively simple to implement, even for very large graphs. Henceforth we refer to the proposed objective optimized using alternating minimization as *AM*.

4 Connections to Other Approaches

Label propagation (LP) (Zhu and Ghahramani, 2002) is a graph-based SSL algorithms that performs Markov random walks on the graph and has a straightforward extension to multi-class problems. The update equations for LP (which also we use for our LP implementations) may be written as

$$p_i^{(n)}(y) = \frac{r_i(y)\delta(i \leq l) + \delta(i > l) \sum_j w_{ij} p_j^{(n-1)}(y)}{\delta(i \leq l) + \delta(i > l) \sum_j w_{ij}}$$

Note the similarity to the update equation for $q_i^{(n)}$ in our AM case. It has been shown that the squared-loss based SSL algorithm (Zhu et al., 2003) and LP have similar updates (Bengio et al., 2007).

The proposed objective C_1 is similar in spirit to the squared-loss based objective in (Zhu et al., 2003; Bengio et al., 2007). Our method, however, differs in that we are optimizing the KL-divergence over probability distributions. We show in section 5 that KL-divergence based loss significantly outperforms the squared-loss. We believe that this could be due

to the following: 1) squared loss is appropriate under a Gaussian loss model which may not be optimal under many circumstances (e.g. classification); 2) KL-divergence $D_{KL}(p||q)$ is based on a relative (relative to p) rather than an absolute error; and 3) under certain natural assumptions, KL-divergence is asymptotically consistent with respect to the underlying probability distributions.

AM is also similar to the spectral graph transducer (Joachims, 2003) in that they both attempt to find labellings over the unlabeled data that respect the smoothness constraints of the graph. While spectral graph transduction is an *approximate* solution to a discrete optimization problem (which is NP hard), AM is an *exact* solution obtained by optimizing a convex function over a continuous space. Further, while spectral graph transduction assumes binary classification problems, AM naturally extends to multi-class situations without loss of convexity.

Entropy Minimization (EnM) (Grandvalet and Bengio, 2004) uses the entropy of the unlabeled data as a regularizer while optimizing a parametric loss function defined over the labeled data. While the objectives in the case of both AM and EnM make use of the entropy of the unlabeled data, there are several important differences: (a) EnM is *not* graph-based, (b) EnM is parametric whereas our proposed approach is non-parametric, and most importantly, (c) EnM attempts to *minimize* entropy while the proposed approach aims to *maximize* entropy. While this may seem a triviality, it has catastrophic consequences in terms of both the mathematics and meaning. The objective in case of EnM is not convex, whereas in our case we have a convex formulation with simple update equations and convergence guarantees.

(Wang et al., 2008) is a graph-based SSL algorithm that also employs alternating minimization style optimization. However, it is inherently squared-loss based which our proposed approach out-performs (see section 5). Further, they do not provide or state convergence guarantees and one side of their update approximates an NP-complete optimization procedure.

The information regularization (IR) (Corduneanu and Jaakkola, 2003) algorithm also makes use of a KL-divergence based loss for SSL. Here the input space is divided into regions $\{R_i\}$ which might or might not overlap. For a given point $x_i \in R_i$, IR attempts to minimize the KL-divergence between $p_i(y_i|x_i)$ and $\hat{p}_{R_i}(y)$, the agglomerative distribution for region R_i. Given a graph, one can define a region to be a vertex and its neighbor thus making IR amenable to graph-based SSL. In (Corduneanu and Jaakkola, 2003), the agglomeration is performed by a simple averaging (arithmetic mean). While IR suggests (without proof of convergence) the use of alternating minimization for optimization, one of the steps of the optimization does *not* admit a closed-form solution. This is a serious practical drawback especially in the case of large data sets. (Tsuda, 2005) (hereafter referred to as PD) is an extension of the IR algorithm to hypergraphs where the agglomeration is performed using the geometric mean. This leads to closed form solutions in both steps of the alternating minimization. There are several important differences between IR and PD on one side and our proposed approach: (a) neither IR nor PD use an entropy regularizer, and (b) the update equation for one of the steps of the optimization in the case of PD (equation 13 in (Tsuda, 2005)) is actually a special case of our update equation for $p_i(y)$ and may be obtained by setting $w_{ij} = 1/2$. Further, our work here may be easily extended to hypergraphs.

5 Results

We compare our algorithm (AM) with other state-of-the-art SSL-based text categorization algorithms, namely, (a) SVM (Joachims, 1999), (b) Transductive-SVM (TSVM) (Joachims, 1999), (c) Spectral Graph Transduction (SGT) (Joachims, 2003), and (d) Label Propagation (LP) (Zhu and Ghahramani, 2002). Note that only SGT and LP are graph-based algorithms, while SVM is fully-supervised (i.e., it does not make use of any of the unlabeled data). We implemented SVM and TSVM using *SVM Light* (Joachims, b) and SGT using *SGT Light* (Joachims, a). In the case of SVM, TSVM and SGT we trained $|\mathrm{Y}|$ classifiers (one for each class) in a one vs. rest manner precisely following (Joachims, 2003).

5.1 Reuters-21578

We used the "ModApte" split of the Reuters-21578 dataset collected from the Reuters newswire in

1987 (Lewis et al., 1987). The corpus has 9,603 training (not to be confused with $\mathcal{D}$) and 3,299 test documents (which represents $\mathcal{D}_u$). Of the 135 potential topic categories only the 10 most frequent categories are used (Joachims, 1999). Categories outside the 10 most frequent were collapsed into one class and assigned a label "other". For each document i in the training and test sets, we extract features $\mathbf{x}_i$ in the following manner: stop-words are removed followed by the removal of case and information about inflection (i.e., stemming) (Porter, 1980). We then compute TFIDF features for each document (Salton and Buckley, 1987). All graphs were constructed using cosine similarity with TFIDF features.

For this task $\mathrm{Y} = \{$ *earn*, *acq*, *money*, *grain*, *crude*, *trade*, *interest*, *ship*, *wheat*, *corn*, *average*$\}$. For LP and AM, we use the output space $\mathrm{Y}' = \mathrm{Y} \cup \{$ *other* $\}$. For documents in $\mathcal{D}_l$ that are labeled with multiple categories, we initialize r_i to have equal non-zero probability for each such category. For example, if document i is annotated as belonging to classes $\{$ *acq*, *grain*, *wheat*$\}$, then $r_i(\textit{acq}) = r_i(\textit{grain}) = r_i(\textit{wheat}) = 1/3$.

We created 21 transduction sets by randomly sampling l documents from the training set with the constraint that each of 11 categories (top 10 categories and the class *other*) are represented at least once in each set. These samples constitute $\mathcal{D}_l$. All algorithms used the same transduction sets. In the case of SGT, LP and AM, the first transduction set was used to tune the hyperparameters which we then held fixed for all the remaining 20 transduction sets. For all the graph-based approaches, we ran a search over $\mathcal{K} \in \{2, 10, 50, 100, 250, 500, 1000, 2000, n\}$ (note $\mathcal{K} = n$ represents a fully connected graph). In addition, in the case of AM, we set $\alpha = 2$ for all experiments, and we ran a search over $\mu \in \{$1e–8, 1e–4, 0.01, 0.1, 1, 10, 100$\}$ and $\nu \in \{$1e–8, 1e–6, 1e–4, 0.01, 0.1$\}$, for SGT the search was over $c \in \{3000, 3200, 3400, 3800, 5000, 100000\}$ (see (Joachims, 2003)).

We report precision-recall break even point (PRBEP) results on the 3,299 test documents in Table 1. PRBEP has been a popular measure in information retrieval (see e.g. (Raghavan et al., 1989)). It is defined as that value for which precision and recall are equal. Results for each category in Table 1 were obtained by averaging the PRBEP over the 20 transduction sets. The final row "average" was obtained by macro-averaging (average of averages). The optimal value of the hyperparameters in case of LP was $\mathcal{K} = 100$; in case of AM, $\mathcal{K} = 2000$, $\mu =$ 1e–4, $\nu =$ 1e–2; and in the case of SGT, $\mathcal{K} = 100$, $c = 3400$. The results show that AM outperforms the state-of-the-art on 6 out of 10 categories and is competitive in 3 of the remaining 4 categories. Further it significantly outperforms all other approaches in case of the macro-averages. AM is significant over its best competitor SGT at the 0.0001 level according to the difference of proportions significance test.

Category	SVM	TSVM	SGT	LP	AM
earn	91.3	95.4	90.4	96.3	**97.9**
acq	67.8	76.6	91.9	90.8	**97.2**
money	41.3	60.0	65.6	57.1	**73.9**
grain	56.2	**68.5**	43.1	33.6	41.3
crude	40.9	**83.6**	65.9	74.8	55.5
trade	29.5	34.0	36.0	**56.0**	47.0
interest	35.6	50.8	50.7	47.9	**78.0**
ship	32.5	46.3	**49.0**	26.4	39.6
wheat	47.9	44.4	59.1	58.2	**64.3**
corn	41.3	33.7	51.2	55.9	**68.3**
average	48.9	59.3	60.3	59.7	**66.3**

Table 1: P/R Break Even Points (PRBEP) for the top 10 categories in the Reuters data set with $l = 20$ and $u = 3299$. All results are averages over 20 randomly generated transduction sets. The last row is the macro-average over all the categories. Note AM is the proposed approach.

Figure 1 shows the variation of "average" PRBEP against the number of labeled documents (l). For each value of l, we tuned the hyperparameters over the first transduction set and used these values for all the other 20 sets. Figure 1 also shows errorbars ($\pm$ standard deviation) all the experiments. As expected, the performance of all the approaches improves with increasing number of labeled documents. Once again in this case, AM, outperforms the other approaches for all values of l.

5.2 WebKB Collection

World Wide Knowledge Base (WebKB) is a collection of 8282 web pages obtained from four academic

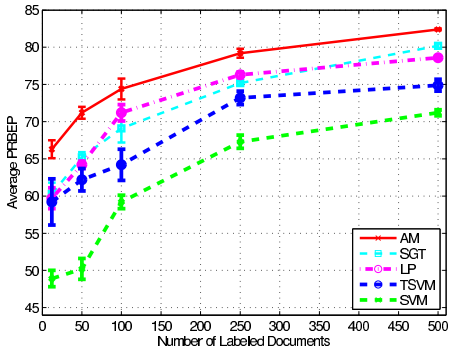

Figure 1: Average PRBEP over all classes vs. number of labeled documents (l) for Reuters data set

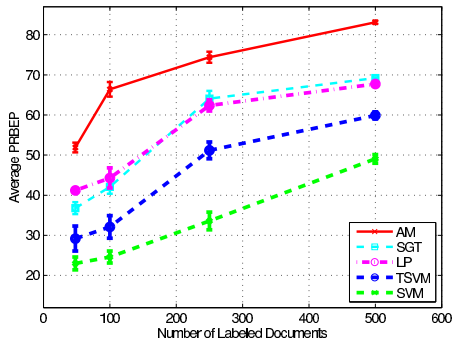

Figure 2: Average PRBEP over all classes vs. number of labeled documents (l) for WebKB collection.

domains. The web pages in the WebKB set are labeled using two different polychotomies. The first is according to topic and the second is according to web domain. In our experiments we only considered the first polychotomy, which consists of 7 categories: *course*, *department*, *faculty*, *project*, *staff*, *student*, and *other*. Following (Nigam et al., 1998) we only use documents from categories *course*, *department*, *faculty*, *project* which gives 4199 documents for the four categories. Each of the documents is in HTML format containing text as well as other information such as HTML tags, links, etc. We used both textual and non-textual information to construct the feature vectors. In this case we did not use either stop-word removal or stemming as this has been found to hurt performance on this task (Nigam et al., 1998). As in the the case of the Reuters data set we extracted TFIDF features for each document and constructed the graph using cosine similarity.

As in (Bekkerman et al., 2003), we created four roughly-equal random partitions of the data set. In order to obtain $\mathcal{D}_l$, we first randomly choose a split and then sample l documents from that split. The other three splits constitute $\mathcal{D}_u$. We believe this is more realistic than sampling the labeled web-pages from a single university and testing web-pages from the other universities (Joachims, 1999). This method of creating transduction sets allows us to better evaluate the generalization performance of the various algorithms. Once again we create 21 transduction sets and the first set was used to tune the hyperparameters. Further, we ran a search over the same grid as used in the case of Reuters. We report precision-recall break even point (PRBEP) results on the 3,148 test documents in Table 2. For this task, we found that the optimal value of the hyperparameter were: in the case of LP, $\mathcal{K} = 1000$; in case of AM, $\mathcal{K} = 1000$, $\mu =$ 1e–2, $\nu =$ 1e–4; and in case of SGT, $\mathcal{K} = 100$, $c = 3200$. Once again, AM is significant at the 0.0001 level over its closest competitor LP. Figure 2 shows the variation of PRBEP with number of labeled documents (l) and was generated in a similar fashion as in the case of the Reuters data set.

Class	SVM	TSVM	SGT	LP	AM
course	46.5	43.9	29.9	45.0	**67.6**
faculty	14.5	31.2	**42.9**	40.3	42.5
project	15.8	17.2	17.5	27.8	**42.3**
student	15.0	24.5	56.6	51.8	**55.0**
average	23.0	29.2	36.8	41.2	**51.9**

Table 2: P/R Break Even Points (PRBEP) for the WebKB data set with $l = 48$ and $u = 3148$. All results are averages over 20 randomly generated transduction sets. The last row is the macro-average over all the classes. AM is the proposed approach.

6 Discussion

We note that LP may be cast into an AM-like framework by using the following sequence of updates,

$$p_i^{(n)}(y) = \delta(i \le l) r_i(y) + \delta(i > l) q_i^{(n-1)},$$

$$q_i^{(n)}(y) = \frac{\sum_j w_{ij} p_i^{(n)}(y)}{\sum_j w_{ij}}$$

To compare the behavior of AM and LP, we applied this form of LP along with AM on a simple 5-node binary-classification SSL graph where two nodes are labeled (node 1 and 2) and the remaining nodes are unlabeled (see Figure 3, top). Since this is binary classification ($|Y| = 2$), each distribution p_i or q_i can be depicted using only a single real number between 0 and 1 corresponding to the probability that each vertex is class 2 (yes two). We show how both LP and AM evolve starting from exactly the same random starting point q^0 (Figure 3, bottom). For each algorithm, the figure shows that both algorithms clearly converge. Each alternate iteration of LP is such that the labeled vertices oscillate due to its clamping back to the labeled distribution, but that is not the case for AM. We see, moreover, qualitative differences in the solutions as well – e.g., AM's solution for the pendant node 5 is less confident than is LP's solution. More empirical comparative analysis between the two algorithms of this sort will appear in future work.

We have proposed a new algorithm for semi-supervised text categorization. Empirical results show that the proposed approach significantly outperforms the state-of-the-art. In addition the proposed approach is relatively simple to implement and has guaranteed convergence properties. While in this work, we use relatively simple features to construct the graph, use of more sophisticated features and/or similarity measures could lead to further improved results.

Acknowledgments

This work was supported by ONR MURI grant N000140510388, by NSF grant IIS-0093430, by the Companions project (IST programme under EC grant IST-FP6-034434), and by a Microsoft Research Fellowship.

References

Alexandrescu, A. and Kirchhoff, K. (2007). Data-driven graph construction for semi-supervised graph-based learnning in nlp. In *Proc. of the Human Language Technologies Conference (HLT-NAACL)*.

Bekkerman, R., El-Yaniv, R., Tishby, N., and Winter, Y. (2003). Distributional word clusters vs. words for text categorization. *J. Mach. Learn. Res.*, 3:1183–1208.

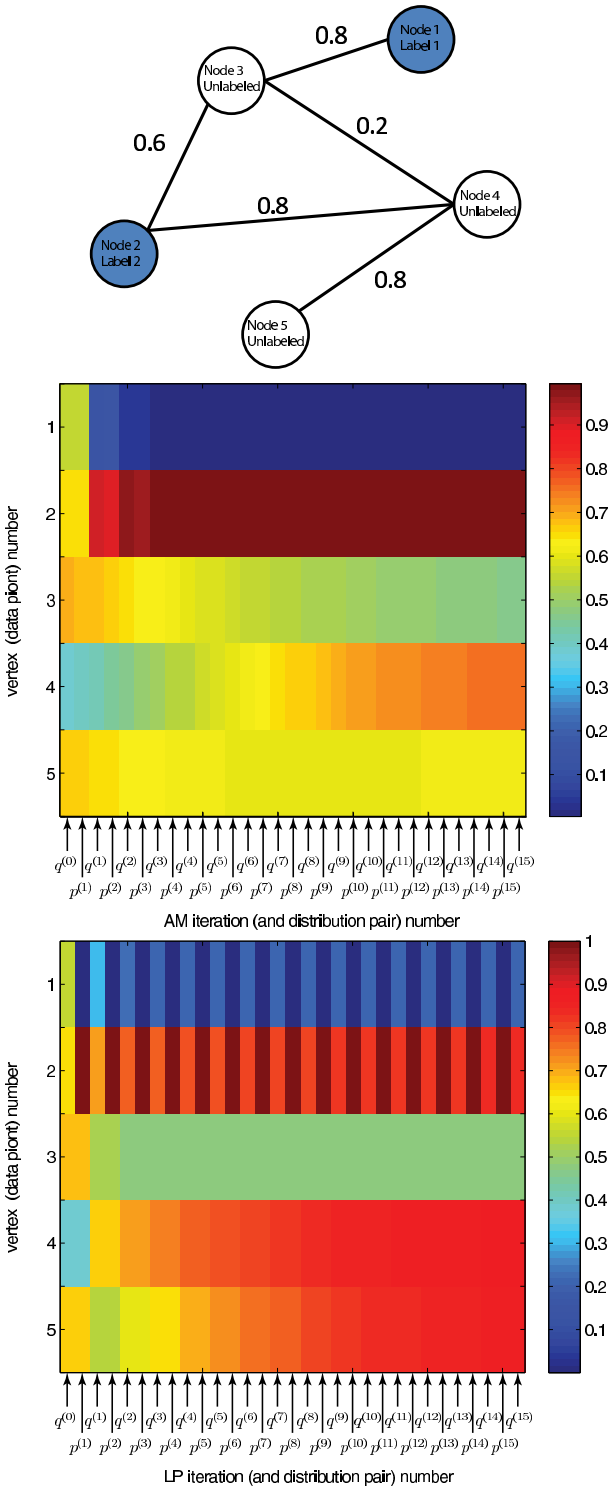

Figure 3: Graph (top), and alternating values of $\mathrm{p}^n, \mathrm{q}^n$ for increasing n for AM and LP.

Belkin, M., Niyogi, P., and Sindhwani, V. (2005). On manifold regularization. In *Proc. of the Conference on Artificial Intelligence and Statistics (AISTATS)*.

Bengio, Y., Delalleau, O., and Roux, N. L. (2007). *Semi-Supervised Learning*, chapter Label Propogation and Quadratic Criterion. MIT Press.

Bertsekas, D. (2004). *Nonlinear Programming*. Athena Scientific Publishing.

Blitzer, J. and Zhu, J. (2008). ACL 2008 tutorial on Semi-Supervised learning. `http://ssl-acl08.wikidot.com/`.

Blum, A. and Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. In *Proc. 18th International Conf. on Machine Learning*, pages 19–26. Morgan Kaufmann, San Francisco, CA.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*.

Chapelle, O., Scholkopf, B., and Zien, A. (2007). *Semi-Supervised Learning*. MIT Press.

Corduneanu, A. and Jaakkola, T. (2003). On information regularization. In *Uncertainty in Artificial Intelligence*.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley Series in Telecommunications. Wiley, New York.

Csiszar, I. and Tusnady, G. (1984). Information Geometry and Alternating Minimization Procedures. *Statistics and Decisions*.

Dumais, S., Platt, J., Heckerman, D., and Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In *CIKM '98: Proceedings of the seventh international conference on Information and knowledge management*, New York, NY, USA.

Grandvalet, Y. and Bengio, Y. (2004). Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems (NIPS)*.

Joachims, T. SGT Light. `http://sgt.joachims.org`.

Joachims, T. SVM Light. `http://svmlight.joachims.org`.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *Proc. of the International Conference on Machine Learning (ICML)*.

Joachims, T. (2003). Transductive learning via spectral graph partitioning. In *Proc. of the International Conference on Machine Learning (ICML)*.

Lewis, D. et al. (1987). Reuters-21578. `http://www.daviddlewis.com/resources/testcollections/reuters21578`.

Nigam, K., McCallum, A., Thrun, S., and Mitchell, T. (1998). Learning to classify text from labeled and unlabeled documents. In *AAAI '98/IAAI '98: Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, pages 792–799.

Pearl, J. (1990). *Jeffrey's Rule, Passage of Experience and Neo-Bayesianism in Knowledge Representation and Defeasible Reasoning*. Kluwer Academic Publishers.

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

Raghavan, V., Bollmann, P., and Jung, G. S. (1989). A critical investigation of recall and precision as measures of retrieval system performance. *ACM Trans. Inf. Syst.*, 7(3):205–229.

Salton, G. and Buckley, C. (1987). Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, USA.

Sindhwani, V., Niyogi, P., and Belkin, M. (2005). Beyond the point cloud: from transductive to semi-supervised learning. In *Proc. of the International Conference on Machine Learning (ICML)*.

Szummer, M. and Jaakkola, T. (2001). Partially labeled classification with Markov random walks. In *Advances in Neural Information Processing Systems*, volume 14.

Tsuda, K. (2005). Propagating distributions on a hypergraph by dual information regularization. In *Proceedings of the 22nd International Conference on Machine Learning*.

Wang, J., Jebara, T., and Chang, S.-F. (2008). Graph transduction via alternating minimization. In *Proc. of the International Conference on Machine Learning (ICML)*.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*.

Zhu, X. (2005a). Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison.

Zhu, X. (2005b). *Semi-Supervised Learning with Graphs*. PhD thesis, Carnegie Mellon University.

Zhu, X. and Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation. Technical report, Carnegie Mellon University.

Zhu, X., Ghahramani, Z., and Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. of the International Conference on Machine Learning (ICML)*.