Journal of Machine Learning Research 12 (2011) 3311-3370 Submitted 7/09; Revised 1/11; Published 11/11

Semi-Supervised Learning with Measure Propagation

Amarnag Subramanya ASUBRAM@EE.WASHINGTON.EDU
Jeff Bilmes BILMES@EE.WASHINGTON.EDU
Department of Electrical Engineering
University of Washington
Seattle, WA 98195, USA

Editor: Yoshua Bengio

Abstract

We describe a new objective for graph-based semi-supervised learning based on minimizing the Kullback-Leibler divergence between discrete probability measures that encode class membership probabilities. We show how the proposed objective can be efficiently optimized using alternating minimization. We prove that the alternating minimization procedure converges to the correct optimum and derive a simple test for convergence. In addition, we show how this approach can be scaled to solve the semi-supervised learning problem on very large data sets, for example, in one instance we use a data set with over 10^8 samples. In this context, we propose a graph node ordering algorithm that is also applicable to other graph-based semi-supervised learning approaches. We compare the proposed approach against other standard semi-supervised learning algorithms on the semi-supervised learning benchmark data sets (Chapelle et al., 2007), and other real-world tasks such as text classification on Reuters and WebKB, speech phone classification on TIMIT and Switchboard, and linguistic dialog-act tagging on Dihana and Switchboard. In each case, the proposed approach outperforms the state-of-the-art. Lastly, we show that our objective can be generalized into a form that includes the standard squared-error loss, and we prove a geometric rate of convergence in that case.

Keywords: graph-based semi-supervised learning, transductive inference, large-scale semi-supervised learning, non-parametric models

1. Introduction

In many applications, annotating training data is time-consuming, costly, tedious, and error-prone. For example, training an accurate speech recognizer requires large amounts of well annotated speech data (Evermann et al., 2005). In the case of document classification for Internet search, it is not feasible to accurately annotate sufficient number of web-pages for all categories of interest. The process of training classifiers with small amounts of labeled data and relatively large amounts of unlabeled data is known as semi-supervised learning (SSL). SSL lends itself as a useful technique in many machine learning applications as one only needs to annotate small amounts of data for training models.

While SSL may be used to solve a variety of learning problems, such as clustering and regression, in this paper we address only the semi-supervised classification problem—henceforth, SSL will refer to semi-supervised classification. Examples of SSL algorithms include self-training (Scudder, 1965) and co-training (Blum and Mitchell, 1998). A thorough survey of SSL algorithms is given in Seeger (2000), Zhu (2005b), Chapelle et al. (2007) and Blitzer and Zhu (2008). SSL

is also related to the problem of *transductive learning* (Vladimir, 1998). In general, a learner is transductive if it is designed only for a closed data set, where the test set is revealed at training time. In practice, however, transductive learners can be modified to handle unseen data (Sindhwani et al., 2005; Zhu, 2005b). Chapelle et al. (2007, Chapter 25) gives a nice discussion on the relationship between SSL and transductive learning.

Graph-based SSL algorithms are an important sub-class of SSL techniques that have received much attention in the recent past (Zhu, 2005b; Chapelle et al., 2007). Here one assumes that the data (both labeled and unlabeled) is embedded within a low-dimensional manifold that may be reasonably expressed by a graph. Each data sample is represented by a vertex in 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, 2002a) 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; Corduneanu and Jaakkola, 2003; Tsuda, 2005). In some cases, for example, label propagation (Zhu and Ghahramani, 2002a) and the harmonic functions algorithm (Zhu et al., 2003; Bengio et al., 2007), it can be shown that the two categories optimize a similar loss function (Zhu, 2005a; Bengio et al., 2007).

A large number of graph-based SSL algorithms attempt to minimize a loss function that is inherently based on squared-loss (Zhu et al., 2003; Bengio et al., 2007; Joachims, 2003). While squared-loss is optimal under a Gaussian noise model, it is not optimal in the case of classification problems. Another potential drawback in the case of some graph-based SSL algorithms (Blum and Chawla, 2001; Joachims, 2003) 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. While it is often argued that the use of binary classifiers within a one vs. rest framework performs as well as true multi-class solutions (Rifkin and Klautau, 2004), our results on SSL problems suggest otherwise (see Section 7.2.2).

Further, there is a lack of principled approaches to incorporate label priors in graph-based SSL algorithms. Approaches such as *class mass normalization* (CMN) and *label bidding* are used as a post-processing step rather than being tightly integrated with the inference (Zhu and Ghahramani, 2002a). In this context, it is important to distinguish label priors from balance priors. Balance priors are used in some algorithms such as Joachims (2003) and discourage the scenario where all the unlabeled samples are classified as belonging to a single class (i.e., a degenerate solution). Balance priors impose selective pressure collectively on the entire set of resulting answers. Label priors, on the other hand, select the more desirable configuration for each answer individually without caring about properties of the overall set of resulting answers. In addition, many SSL algorithms, such as Joachims (2003) and Belkin et al. (2005), are unable to handle *label uncertainty*, where there may be insufficient evidence to justify only a single label for a labeled sample.

Another area for improvement over previous work in graph-based SSL (and SSL in general) is the lack of algorithms that scale to very large data sets. SSL is based on the premise that unlabeled data is easily obtained, and adding large quantities of unlabeled data leads to improved performance. Thus practical scalability (e.g., parallelization), is important to apply SSL algorithms on large real-world data sets. Collobert et al. (2006) and Sindhwani and Keerthi (2006) discuss the application of TSVMs to large-scale problems. Delalleau et al. (2005) suggests an algorithm for improving the induction speed in the case of graph-based algorithms. Karlen et al. (2008) solve a graph transduction problem with 650,000 samples. To the best of our knowledge, the largest graph-based problem

solved to date had about 900,000 samples (includes both labeled and unlabeled data) (Tsang and Kwok, 2006). Clearly, this is a fraction of the amount of unlabeled data at our disposal. For example, on the Internet alone, we create 1.6 billion blog posts, 60 billion emails, 2 million photos and 200,000 videos every day (Tomkins, 2008). In general, graph-based SSL algorithms that use matrix inversion (Zhu et al., 2003; Belkin et al., 2005) or eigen-based matrix decomposition (Joachims, 2003) do not scale very easily.

In Subramanya and Bilmes (2008), we proposed a new framework for graph-based SSL that involves optimizing a loss function based on Kullback-Leibler divergence (KLD) between probability measures defined for each graph vertex. These probability measures encode the class membership probabilities. The advantages of this new convex objective are: (a) it is naturally amenable to multi-class (> 2) problems; (b) it can handle label uncertainty; and (c) it can integrate priors. Furthermore, the use of probability measures allows the exploitation of other well-defined functions of measures, such as entropy, to improve system performance. Subramanya and Bilmes (2008) also showed how the proposed objective can be optimized using alternating minimization (AM) (Csiszar and Tusnady, 1984) leading to simple update equations. This new approach to graph-based SSL was shown to outperform other state-of-the-art SSL algorithms for the document and web page classification tasks. In this paper we extend the above work along the following lines –

1. We prove that AM on the proposed convex objective for graph-based SSL converges to the global optima. In addition we derive a test for convergence that does not require the computation of the objective.

2. We compare the performance of the proposed approach against other state-of-the-art SSL approaches, such as manifold regularization (Belkin et al., 2005), label propagation (Zhu and Ghahramani, 2002a), and spectral graph transduction (Joachims, 2003) on a variety of tasks ranging from synthetic data sets to SSL benchmark data sets (Chapelle et al., 2007) to real-world problems such as phone classification, text classification, web-page classification and dialog-act tagging.

3. We propose a graph node ordering algorithm that is cache cognizant and makes obtaining a linear speedup with a parallel symmetric multi-processor (SMP) implementation more likely. As a result, the algorithms are able to scale to very large data sets. The node ordering algorithm is quite general and can be applied to graph-based SSL algorithms such as Zhu and Ghahramani (2002a); Zhu et al. (2003). In one instance, we solve a SSL problem over a graph with 120 million vertices (which is quite a bit more than the previous largest size of 900,000 vertices). A useful byproduct of this experiment is the *semi-supervised switchboard transcription project* (S3TP) which consists of phone level annotations of the *Switchboard-I* corpus generated in a semi-supervised manner (see Section 8.1, Subramanya and Bilmes, 2009).

4. We propose a graph-based SSL objective using Bregman divergence in Section 9.1. This objective generalizes previously proposed approaches such as label propagation (Zhu and Ghahramani, 2002a), the harmonic functions algorithm (Zhu et al., 2003), the quadratic cost criterion (Bengio et al., 2007) and our proposed approach. This objective can potentially be optimized using AM which portends well for solving general learning problems over objects for which a Bregman divergence can be defined (Tsuda et al., 2005).

5. A specific case of the Bregman divergence form is the standard squared-loss based objective, and we prove a geometric rate of convergence in this case in Appendix F

6. We discuss a principled approach to integrating label priors into the proposed objective (see Section 9.2).

7. We also show how our proposed objective can be extended to directed graphs (see Section 9.3).

2. Graph Construction

Let $\mathcal{D}_l = \{(\mathbf{x}_i, r_i)\}_{i=1}^{l}$ be the set of labeled samples, $\mathcal{D}_u = \{\mathbf{x}_i\}_{i=l+1}^{l+u}$ the set of unlabeled samples and $\mathcal{D} \triangleq \{\mathcal{D}_l, \mathcal{D}_u\}$. Here r_i is an encoding of the labeled data and will be explained shortly. We are interested in solving the transductive learning problem, that is, given $\mathcal{D}$, the task is to predict the labels of the samples in $\mathcal{D}_u$ (for inductive see Section 7.4). We are given an undirected weighted graph $\mathcal{G} = (V, E)$, where the vertices (nodes) $V = \{1, \ldots, m\}$ ($m \triangleq l + u$) are the data points in $\mathcal{D}$ and the edges $E \subseteq V \times V$. Let $V = V_l \cup V_u$ where V_l is the set of labeled vertices and V_u the set of unlabeled vertices. $\mathcal{G}$ may be represented via a matrix $\mathbf{W}$ referred to as the weight or affinity matrix.

There are many ways of constructing the graph. In some applications, it might be a natural result of relationship between the samples in $\mathcal{D}$, for example, consider the case where each vertex represents a web-page and the edges represent the links between web-pages. In other cases, such as the work of Fei and Changshui (2006), the graph is generated by performing an operation similar to local linear embedding (LLE) but constraining the LLE weights to be non-negative. In a majority of the applications, including those considered in this paper, we use k-nearest neighbor (NN) graphs. In our case here, we make use of symmetric k-NN graphs and so the edge weight $w_{ij} = [\mathbf{W}]_{ij}$ is given by

$$w_{ij} = \begin{cases} \text{sim}(\mathbf{x}_i, \mathbf{x}_j) & \text{if } j \in \mathcal{K}(i) \text{ or } i \in \mathcal{K}(j) \\ 0 & \text{otherwise} \end{cases}$$

where $\mathcal{K}(i)$ is the set of k-NN of $\mathbf{x}_i$ ($|\mathcal{K}(i)| = \text{k}, \ \forall i$) and $\text{sim}(\mathbf{x}_i, \mathbf{x}_j)$ is a measure of similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ (which are represented by nodes i and j). It is assumed that the similarity measure is symmetric, that is, $\text{sim}(x, y) = \text{sim}(y, x)$. Further $\text{sim}(x, y) \geq 0$. Some popular similarity measures include

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

where $\| \mathbf{x}_i \|_2$ is the 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 a radial-basis function (RBF) kernel of width σ applied to the squared Euclidean distance while the second is cosine similarity. Choosing the correct similarity measure and k are crucial steps in the success of any graph-based SSL algorithm as it determines the graph. At this point, graph construction "is more of an art, than science" (Zhu, 2005a) and is an active research area (Alexandrescu and Kirchhoff, 2007b). The choice of $\mathbf{W}$ depends on a number of factors such as, whether $\mathbf{x}_i$ is continuous or discrete and characteristics of the problem at hand. We discuss more about the choice of $\mathbf{W}$ in the context of the appropriate problem in Section 7.

3. Proposed Approach for Graph-based Semi-Supervised Learning

For each $i \in V$ and $j \in V_l$, we define discrete probability measures p_i and r_j respectively over the measurable space $(\mathrm{Y}, \mathcal{Y})$. That is, for each vertex in the graph, we define a measure p_i and for all the labeled vertices, in addition to the p's we also define r_i (recall, $\mathcal{D}_l = \{(\mathbf{x}_i, r_i)\}_{i=1}^{l}$). Here $\mathcal{Y}$ is the σ-field of measurable subsets of Y and $\mathrm{Y} \subset \mathbb{N}$ (the set of natural numbers) is the discrete space of classifier outputs. Thus $|\mathrm{Y}| = 2$ yields binary classification while $|\mathrm{Y}| > 2$ yields multi-class. As we only consider classification problems here, p_i and r_i are in essence multinomial distributions and so $p_i(y)$ represents the probability that the sample represented by vertex i belongs to class y. We assume that there is at least one labeled sample for every class. Note that the objective we propose is actually more general and can be easily extended to other learning problems such as regression.

The $\{r_i\}_i$'s represent the labels of the supervised portion of the training data and are derived 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 $\hat{y}_i = \{\hat{y}_i^{(1)}, \ldots, \hat{y}_i^{(t)}\}$, $t \leq |\mathrm{Y}|$ 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. As can be seen, the r_i's can handle a wide variety of inputs ranging from the most certain case where a single input yields a single output to cases where there is an *uncertainty* associated with the output for a given input. It is important to distinguish between the classical multi-label problem and the use of uncertainty in r_j. In our case, if there are two non-zero outputs during training as in $r_j(\overline{y}_1), r_j(\overline{y}_2) > 0$, $\overline{y}_1, \overline{y}_2 \in \mathrm{Y}$, it does not imply that the input $\mathbf{x}_j$ necessarily possesses the properties of the two corresponding classes. Rather, this means that there is uncertainty regarding truth, and we use a discrete probability measure over the labels to represent this uncertainty.

As p_i and r_i are discrete probability measures, we have that $\sum_y p_i(y) = 1$, $p_i(y) \geq 0$, $\sum_y r_i(y) = 1$, and $r_i(y) \geq 0$. In other words, p_i and r_i lie within a $|\mathrm{Y}|$-dimensional probability simplex which we represent using $\triangle_{|\mathrm{Y}|}$ and so $p_i, r_i \in \triangle_{|\mathrm{Y}|}$ (henceforth denoted as $\triangle$). Also $\mathrm{p} \triangleq (p_1, \ldots, p_m) \in \triangle^m$ denotes the set of measures to be learned, and $\mathrm{r} \triangleq (r_1, \ldots, r_l) \in \triangle^l$ are the set of measures that are given. Here, $\triangle^m \triangleq \triangle \times \ldots \times \triangle$ (m times). Finally let u be the uniform probability measure on $(\mathrm{Y}, \mathcal{Y})$, that is, $u(\mathrm{y}) = \frac{1}{|\mathrm{Y}|}\ \forall\, \mathrm{y} \in \mathrm{Y}$. In other words, u evenly distributes all the available probability mass across all possible assignments.

Consider the optimization problem $\mathcal{P}_{KL} : \min_{\mathrm{p} \in \triangle^m} \mathcal{C}_{KL}(\mathrm{p})$ where

$$\mathcal{C}_{KL}(\mathrm{p}) = \sum_{i=1}^{l} D_{KL}\big(r_i || p_i\big) + \mu \sum_{i=1}^{m} \sum_{j \in \mathcal{N}(i)} w_{ij} D_{KL}\big(p_i || p_j\big) - \nu \sum_{i=1}^{n} H(p_i).$$

Here $H(p) = -\sum_{\mathrm{y}} p(\mathrm{y}) \log p(\mathrm{y})$ is the Shannon entropy of p and $D_{KL}(p_i||q_j)$ is the KLD between measures p_i and q_j and is given by $D_{KL}(p||q) = \sum_{\mathrm{y}} p(\mathrm{y}) \log \frac{p(\mathrm{y})}{q(\mathrm{y})}$. (μ, ν) are hyper-parameters whose choice we discuss in Section 7. Given a vertex $i \in V$, $\mathcal{N}(i)$ denotes the set of neighbors of the vertex in the graph corresponding to w_{ij} and thus $|\mathcal{N}(i)|$ represents vertex i's degree.

Lemma 1 *If $\mu, \nu, w_{ij} \geq 0,\ \forall\, i, j$ then $\mathcal{C}_{KL}(\mathrm{p})$ is convex.*

Proof This follows as $D_{KL}(p_i||q_j)$ is convex in the pair (p_i, q_j), negative entropy is convex (Cover and Thomas, 1991), and we have a non-negative weighted combination of convex functions. ■

The goal of the above objective is to find the best set of measures p_i that attempt to: 1) agree with the labeled data r_j wherever it is available (the first term in $\mathcal{C}_{KL}$); 2) agree with each other when they are close according to a graph (the second graph-regularizer term in $\mathcal{C}_{KL}$); and 3) be smooth in some way (the last term in $\mathcal{C}_{KL}$). In essence, SSL on a graph consists of finding a labeling for $\mathcal{D}_u$ that is consistent with both the labels provided in $\mathcal{D}_l$ and the geometry of the data induced by the graph. In the following we discuss each of the above terms in detail.

The first term of $\mathcal{C}_{KL}$ 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.

The second term of $\mathcal{C}_{KL}$ 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 KLD sense. One question about the objective relates to the asymmetric nature of KLD (i.e., $D_{KL}(p||q) \neq D_{KL}(q||p)$) (see Section 9.3 for a discussion about this issue in the directed graph case).

Lemma 2 *While KLD is asymmetric, given an undirected graph $\mathcal{G}$, the second term in the proposed objective, $\mathcal{C}_{KL}(\mathrm{p})$, is inherently symmetric.*

Proof As we have an undirected graph, $\mathbf{W}$ is symmetric, that is, $w_{ij} = w_{ji}$ and for every $w_{ij} D_{KL}(p_i||p_j)$, we also have $w_{ji} D_{KL}(p_j||p_i)$. ■

The last term encourages each p_i to be close to the uniform distribution (i.e., a maximum entropy configuration) if not preferred to the contrary by the first two terms. This acts as a guard against degenerate solutions commonly encountered in graph-based SSL (Blum and Chawla, 2001; Joachims, 2003). For example, consider the case where a part of the graph is almost completely disconnected from any labeled vertex—that is, a "pendant" graph component. This occurs sometimes in the case of k-NN graphs. In such situations the third term ensures 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 (e.g., Gaussian mixture models, multi-layer perceptrons, Gaussian kernels) that tend to be confident even in regions far from the decision boundary.

Finally, while the second graph-regularizer term encourages high-entropy solutions for nodes that have high entropy neighbors, the graph regularizer alone is insufficient to yield high-entropy solutions in other cases where it may be desirable. For example, consider a connected pendant component that is "separated" from the rest of the graph by labeled nodes that have the same value. We can view this as a "lolly-pop" component, where the base of the stem is labeled, but the rest of the stem and the round portion of the lolly-pop are unlabeled. In such a configuration, the optimum configuration will set the label of all nodes to be equal to the labels of the stem. There can be cases,

however, where more uncertainty should be expressed about such a large mass of unlabeled nodes distantly situated from the nearest labeled node. The last term in the objective allows a solution where uncertainty is encouraged when a node is geodesically very distant from any label.

We conclude this section by summarizing some of the highlights and features of our framework:

1. *Manifold assumption:* $\mathcal{C}_{KL}$ uses the "manifold assumption" for SSL (see chapter 2 in Chapelle et al., 2007)—it assumes that the input data may be reasonably embedded within a low-dimensional manifold which in turn can be represented by a graph.

2. *Naturally multiclass:* As the objective is defined in terms of probability distributions over integers rather than just integers (or real-valued relaxations of integers Joachims, 2003; Zhu et al., 2003), the framework generalizes in a straightforward manner to multi-class problems. As a result, all the parameters are estimated jointly (compare to one vs. rest approaches which involve solving $|\mathrm{Y}|$ independent classification problems).

3. *Label uncertainty:* The objective is capable of handling uncertainty in the labels (encoded using r_i) (Pearl, 1990). We present an example of this in the scenario of text classification in Section 7.3.

4. *Ability to incorporate priors:* Priors can be incorporated by either

   (a) minimizing the KLD between an agglomerative measure and a prior, that is, $\mathcal{C}'_{KL}(\mathrm{p}) = \mathcal{C}_{KL}(\mathrm{p}) + \kappa D_{KL}(p_0||\tilde{p})$ where $\tilde{p}$ can for example be the arithmetic or geometric mean over p_i's or

   (b) minimizing the KLD between p_i and the prior p_0. First note that $\mathcal{C}_{KL}(\mathrm{p})$ may be re-written as $\mathcal{C}_{KL}(\mathrm{p}) = \sum_{i=1}^{l} D_{KL}\big(r_i||p_i\big) + \mu \sum_{i,j} w_{ij} D_{KL}\big(p_i||p_j\big) + \nu \sum_i D_{KL}\big(p_i||u\big)$ where u is uniform measure. This follows as $D_{KL}\big(p_i||u\big) = -H(p_i) + \text{const}$. Now if we replace the uniform measure, u, in the above by p_0 then we are asking for each p_i to be close to p_0. Even more generally, we may replace the uniform measure by a distinct fixed prior distribution for each vertex.

   While the former is more global, in the latter case, the prior effects each vertex individually. Also, the global prior is closer to the balance prior used in the case of algorithms like spectral graph transduction (Joachims, 2003). In both of the above cases, the resulting objective remains convex. It is also important to point out that using one of the above does not preclude us from using the other. We consider this to be a unique feature of our approach as we can incorporate both the balance and label priors simultaneously.

5. *Directed graphs:* The proposed objective can be used with directed graphs without any modification (see Section 9.3).

3.1 Solving $\mathcal{P}_{KL}$

As $\mathcal{C}_{KL}$ is convex and the constraints are linear, $\mathcal{P}_{KL}$ is a convex programming problem (Bertsekas, 1999). However, $\mathcal{P}_{KL}$ does not admit a closed form solution because the gradient of $\mathcal{C}_{KL}(\mathrm{p})$ w.r.t. $p_i(\mathrm{y})$ is of the form, $k_1 p_i(\mathrm{y}) \log p_i(\mathrm{y}) + k_2 p_i(\mathrm{y}) + k_3$ ($k_1,\ k_2,\ k_3$ are constants). Further, optimizing the dual of $\mathcal{P}_{KL}$ requires solving a similar equation. One of the reasons that $\mathcal{P}_{KL}$ does not admit a closed form solution is because we are optimizing w.r.t. to both variables in a KLD. Thus, we

are forced to use one of the numerical convex optimization techniques (Boyd and Vandenberghe, 2006) such as barrier methods (a type of interior point method, or IPM) or penalty methods (e.g., the method of multipliers (Bertsekas, 1999)). In the following we explain how method of multipliers (MOM) with quadratic penalty may be used to solve $\mathcal{P}_{KL}$. We choose a MOM based solver as it has been shown to be more numerically stable and has similar rates of convergence as other gradient based convex solvers (Bertsekas, 1999).

It can be shown that the update equations for $p_i(y)$ for solving $\mathcal{P}_{KL}$ using MOM are given by (see appendix A for details)

$$p_i^{(n)}(y) = \left[p_i^{(n-1)}(y) - \alpha^{(n-1)} \left(\frac{\partial \mathcal{L}_{\mathcal{C}_{KL}}(\mathrm{p}, \Lambda)}{\partial p_i(y)} \right)_{\{\mathrm{p}=\mathrm{p}^{(n-1)}, \Lambda=\Lambda^{(n-1)}, c=c^{(n-1)}\}} \right]^+$$

where $n = 1, \ldots,$ is the iteration index, $\alpha^{(n-1)}$ is the learning rate which is determined using the Armijo rule (Bertsekas, 1999), $[x]^+ = \max(x, 0)$ and

$$\frac{\partial \mathcal{L}_{\mathcal{C}_{KL}}(\mathrm{p}, \Lambda)}{\partial p_i(y)} = \mu \sum_{j \in \mathcal{N}(i)} \left[w_{ej} \big(1 + \log p_i(y) - \log p_j(y)\big) - \frac{w_{je} p_j(y)}{p_i(y)} \right] - \frac{r_i(y)}{p_i(y)} \delta(e \leq l) + \nu(\log p_i(y) + 1) + \lambda_i + 2c\big(1 - \sum_y p_i(y)\big).$$

In the above $\Lambda = \{\lambda_i\}$ are the Lagrange multipliers and c is the MOM coefficient (see appendix A).

While the MOM-based approach to solving $\mathcal{P}_{KL}$ is simple to derive, it has a number of drawbacks:

1. *Hyper(Extraneous)-Parameters:* Solving $\mathcal{P}_{KL}$ using MOM requires the careful tuning of a number of extraneous parameters including, the learning rate (α) which is obtained using the Armijo rule which has 3 other parameters, MOM penalty parameter (c), stopping criteria (ζ), and penalty update parameters (γ and β). In general, in the interest of scalability, it is advantageous to have as few tuning parameters in an algorithm as possible, especially in the case of SSL where there is relatively little labeled data available to "hold out" for use in cross validation tuning. The success of MOM based optimization depends on the careful tuning of all the 7 extraneous parameters (this is in addition to μ and ν, the hyper-parameters in the original objective). This is problematic as settings of these parameters that yield good performance on a particular data set have no generalization guarantees. In Section 7.2.1, we present an analysis that shows sensitivity of MOM to the settings of these parameters.

2. *Convergence guarantees:* For most problems, MOM lacks convergence guarantees. Bertsekas (1999) only provides a proof of convergence for cases when $c^{(n)} \to \infty$, a condition rarely satisfied in practice.

3. *Computational cost:* The termination criteria for the MOM based solver for $\mathcal{P}_{KL}$ requires that one compute the value of the objective function for every iteration leading to increased computational complexity.

4. *Lack of intuition in update equations:* While the update equations for $p_i(y)$ are easy to obtain, they lack an intuitive explanation.

As stated above, there are other alternatives for numerical optimization of convex functions. In particular, we could use an IPM for solving $\mathcal{P}_{KL}$, but barrier methods also have their own drawbacks (e.g., each step involves solving n linear equations). It is important to point out that we are not arguing against the use of gradient based approaches in general as they been quite successful for training multi-layer perceptrons, hidden conditional random fields, and so on where the objective is inherently non-convex. Sometimes even when the objective is convex, we need to rely on MOM or IPM for optimization like in our case in Section 9.2. However, as $\mathcal{P}_{KL}$ is a convex optimization problem, in this paper we explore and prefer other techniques for its optimization which do not have the aforementioned drawbacks.

4. Alternating Minimization (AM)

Given a distance function $d(p,q)$ between objects $p \in \mathcal{P}, q \in \mathcal{Q}$ where $\mathcal{P}, \mathcal{Q}$ are sets, consider the problem finding the p,q that minimizes $d(p,q)$. Sometimes solving this problem directly is hard, and in such cases the method of alternating minimization (AM) lends itself as a valuable tool for efficient optimization. AM refers to the case where we alternately minimize $d(p,q)$ with respect to p while q is held fixed and then vice-versa, that is,

$$p^{(n)} = \operatorname*{argmin}_{p \in \mathcal{P}} d(p, q^{(n-1)}) \text{ and } q^{(n)} = \operatorname*{argmin}_{q \in \mathcal{Q}} d(p^{(n)}, q).$$

Figure 1 illustrates the two steps of AM over two convex sets. We start with an initial arbitrary $Q_0 \in \mathcal{Q}$ which is held fixed while we minimize w.r.t. $P \in \mathcal{P}$ which leads to P_1. The objective is then held fixed w.r.t. P at $P = P_1$ and minimized over $Q \in \mathcal{Q}$ and this leads to Q_1. The above is then repeated with Q_1 playing the role of Q_0 and so on until (in the best of cases) convergence. The Expectation-Maximization (EM) (Dempster et al., 1977) algorithm is an example of AM. Moreover, the above objective over two variables can be extended to an objective over n variables. In such cases $n-1$ variables are held fixed while the objective is optimized with respect to the one remaining variable and the procedure iterates in a similar round-robin fashion.

An AM procedure might or might not have the following properties: 1) a closed-form solution to each of the alternating minimization steps of AM; 2) convergence to a final solution, and 3) convergence to a correct minimum of $d(p,q)$. In some cases, even when there is no closed-form solution to the direct minimization of $d(p,q)$, each step of AM has a closed form solution. In other cases, however (see Corduneanu and Jaakkola, 2003), one or both the steps of AM do not have closed form solutions.

Depending on $d(p,q)$ and on the nature of $\mathcal{P}, \mathcal{Q}$, an AM procedure might never converge. Even when AM does converge, it might not converge to the true correct minimum of $d(p,q)$. In general, certain conditions need to hold for AM to converge to the correct solution. Some approaches, such as Cheney and Goldstien (1959), Zangwill (1969) and Wu (1983), rely on the topological properties of the objective and the space over which it is optimized, while others such as Csiszar and Tusnady (1984) use geometrical arguments. Still others (Gunawardena, 2001) use a combination of the above.

In this paper, we take the *information geometry* approach proposed by Csiszar and Tusnady (1984) where the so-called *5-points property* (5-pp) is fundamental to determining whether AM on an objective converges to the global optima. 5-pp is defined as follows:

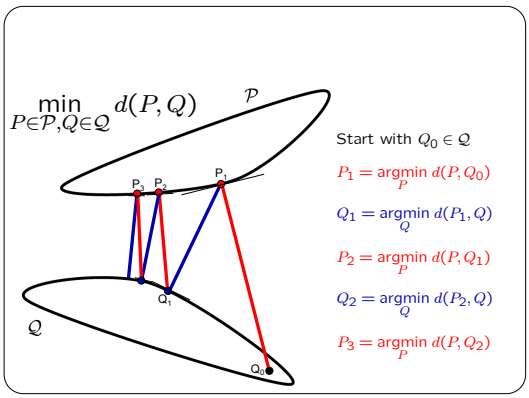

Figure 1: Alternating Minimization

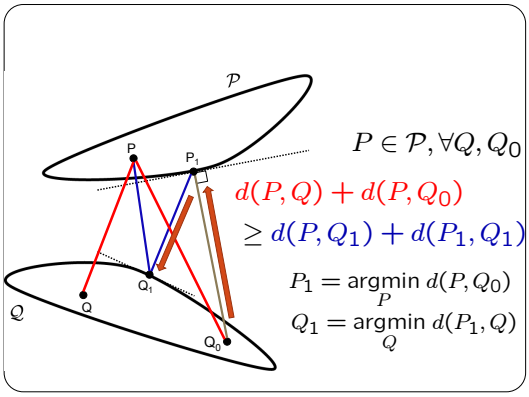

Figure 2: Illustration of the 5-point property

Definition 3 *If $\mathcal{P}$, $\mathcal{Q}$ are convex sets of finite measures, given a divergence $d(p,q)$, $p \in \mathcal{P}$, $q \in \mathcal{Q}$, then the 5-pp is said to hold for $p \in \mathcal{P}$ if $\forall\, q, q_0 \in \mathcal{Q}$ we have*

$$d(p,q) + d(p,q_0) \geq d(p,q_1) + d(p_1,q_1)$$

where $p_1 \in \operatorname{argmin}_{p\in\mathcal{P}} d(p,q_0)$ and $q_1 \in \operatorname{argmin}_{q\in\mathcal{Q}} d(p_1,q)$.

Figure 2 shows an illustration of 5-pp. Here we start with some $Q_0 \in \mathcal{Q}$, $P_1 = \operatorname{argmin}_{P\in\mathcal{P}} d(P,Q_0)$ and $Q_1 = \operatorname{argmin}_{Q\in\mathcal{Q}} d(P_1,Q)$. 5-pp is said hold for $d(P,Q)$ if for any $P \in \mathcal{P}$ and any $Q \in \mathcal{Q}$, the sum of the lengths of the red lines is greater than or equal to the sum of the lengths of the blue lines in Figure 2. Here the lengths are measured using the objective $d(P,Q)$. Csiszar and Tusnady (1984) have shown that the 5-pp holds for all p when $d(p,q) = D_{KL}(p||q)$.

So now the question is whether our proposed objective $\mathcal{C}_{KL}(\mathrm{p})$ can be optimized using AM and whether it converges to the correct optimum. This is the topic of discussion in the next section.

4.1 Graph-based SSL using AM

$\mathcal{P}_{KL}$ cannot be solved using AM and so we reformulate it in a manner amenable to AM. The following are the desired properties of such a reformulation –

1. The new (reformulated) objective should be a valid graph-based SSL criterion.
2. AM on the reformulated objective should converge to the global optimum of this objective.
3. The optimal solution in the case of the original ($\mathcal{P}_{KL}$) and reformulated problem should be identical.
4. Each step of the AM process should have a closed form and easily computable solution.
5. The resulting algorithm should scale to large data sets.

In this section, we formulate an objective that satisfies all of these properties. Consider the following reformulated objective –

$$\mathcal{P}_{MP}: \min_{\mathrm{p},\mathrm{q}\in\triangle^m} \mathcal{C}_{MP}(\mathrm{p},\mathrm{q}) \text{ where}$$

$$\mathcal{C}_{MP}(\mathrm{p},\mathrm{q}) = \sum_{i=1}^{l} D_{KL}\big(r_i||q_i\big) + \mu \sum_{i=1}^{m} \sum_{j\in\mathcal{N}'(i)} w'_{ij} D_{KL}\big(p_i||q_j\big) - \nu \sum_{i=1}^{m} H(p_i)$$

where for each vertex i in $\mathcal{G}$, we define a third discrete probability measure q_i over the measurable space $(\mathrm{Y},\mathcal{Y})$, $w'_{ij} = \left[\mathbf{W}'\right]_{ij}$, $\mathbf{W}' = \mathbf{W} + \alpha\mathbf{I}_n$, $\mathcal{N}'(i) = \{i\} \cup \mathcal{N}(i)$ and $\alpha \geq 0$. Here the q_i's play a similar role as the p_i's and can potentially be used to obtain a final classification result ($\mathrm{argmax}_y\, q_i(y)$). Thus, it would seem that we now have two classification results for each sample – one the most likely assignment according to p_i and another given by q_i. However, $\mathcal{C}_{MP}$ includes terms of the form $(w_{ii}+\alpha)D_{KL}(p_i||q_i)$ which encourage p_i and q_i to be close to each other. Thus α, which is a hyper-parameter, plays an important role in ensuring that $p_i = q_i,\ \forall\ i$. It should be clear that

$$\operatorname*{argmin}_{\mathrm{p}\in\triangle^n} \mathcal{C}_{KL}(\mathrm{p}) = \lim_{\alpha\to\infty} \operatorname*{argmin}_{\mathrm{p},\mathrm{q}\in\triangle^n} \mathcal{C}_{MP}(\mathrm{p},\mathrm{q}).$$

In the following we will show that there exists a finite α such that at a minima, $p_i(y) = q_i(y)\ \forall\ i,y$ (henceforth we will denote this as either $p_i = q_i\ \forall\ i$ or $\mathrm{p} = \mathrm{q}$).

We note that the new objective $\mathcal{C}_{MP}(\mathrm{p},\mathrm{q})$ can itself be seen as an intrinsically valid SSL criterion. While the first term encourages q_i for the labeled vertices to be close to the labels, r_i, the last term encourages higher entropy p's. The second term, in addition to acting as a graph regularizer, also acts as a glue between the p's and q's.

A natural question that arises at this point is why we choose this particular form for $\mathcal{C}_{MP}$ and not other alternatives. First note that $-H(p_i) = D_{KL}(p_i||u) + \text{const}$ where u is the uniform measure. KLD is a function of two variables (say the left and the right). In $\mathcal{C}_{MP}$, the p's always occur on the left hand side while the q's occur on the right. Recall that the reason $\mathcal{C}_{KL}$ did not admit a closed form solution is because we were attempting to optimize w.r.t. both the variables in a KLD. Thus going from $\mathcal{C}_{KL}$ to $\mathcal{C}_{MP}$ accomplishes two goals – (a) it makes optimization via AM possible, and (b) as we see shortly, it leads to closed form updates. Next we address the question of whether AM on $\mathcal{C}_{MP}$ converges to the correct optimum.

Lemma 4 *If $\mu, \nu, w'_{ij} \geq 0 \; \forall \; i, j$ then $\mathcal{C}_{MP}(\mathrm{p}, \mathrm{q})$ is convex.*

Proof This follows as $D_{KL}(p||q)$ is convex in the pair, and we have a weighted sum of convex functions with non-negative weights. ■

The previous lemma guarantees that any local minimum is a global minimum. The next theorem gives the powerful result that the AM procedure on our objective $\mathcal{C}_{MP}$ is guaranteed to converge to the true global minimum of $\mathcal{C}_{MP}$.

Theorem 5 (Convergence of AM on $\mathcal{C}_{MP}$, see appendix B) *If*

$$\mathrm{p}^{(n)} = \operatorname*{argmin}_{\mathrm{p} \in \triangle^m} \mathcal{C}_{MP}(\mathrm{p}, \mathrm{q}^{(n-1)}), \;\; \mathrm{q}^{(n)} = \operatorname*{argmin}_{\mathrm{q} \in \triangle^m} \mathcal{C}_{MP}(\mathrm{p}^{(n)}, \mathrm{q}) \text{ and } q_i^{(0)}(y) > 0 \; \forall \, y \in \mathrm{Y}, \; \forall i \text{ then}$$

(a) $\mathcal{C}_{MP}(\mathrm{p}, \mathrm{q}) + \mathcal{C}_{MP}(\mathrm{p}, \mathrm{p}^{(0)}) \geq \mathcal{C}_{MP}(\mathrm{p}, \mathrm{q}^{(1)}) + \mathcal{C}_{MP}(\mathrm{p}^{(1)}, \mathrm{q}^{(1)})$ *for all* $\mathrm{p}, \mathrm{q} \in \triangle^m$, *and*

(b) $\lim_{n \to \infty} \mathcal{C}_{MP}(\mathrm{p}^{(n)}, \mathrm{q}^{(n)}) = \inf_{\mathrm{p}, \mathrm{q} \in \triangle^m} \mathcal{C}_{MP}(\mathrm{p}, \mathrm{q})$.

Next we address the issue of showing that the solutions obtained in the case of the original and reformulated objectives are the same. We already know that if $\alpha \to \infty$ then we have equality, but we are interested in obtaining a finite lower-bound on α for which this is still the case. In the below, we let $\mathcal{C}_{MP}(\mathrm{p}, \mathrm{q}; \{w'_{ii} = 0\}_i)$ be the objective $\mathcal{C}_{MP}$ shown with the weight matrix parameterized with $w'_{ii} = 0$ for all i, and we let $\mathcal{C}_{MP}(\mathrm{p}, \mathrm{q}; \alpha)$ be the objective function shown with a particular parameterized value of α. For the proof of the next lemma and the two theorems that follow, see appendix C.

Lemma 6 *We have that*

$$\min_{\mathrm{p}, \mathrm{q} \in \triangle^m} \mathcal{C}_{MP}(\mathrm{p}, \mathrm{q}; w'_{ii} = 0) \leq \min_{\mathrm{p} \in \triangle^m} \mathcal{C}_{KL}(\mathrm{p}).$$

Theorem 7 *Given any $A, B, S \in \triangle^m$ (i.e., $A = [a_1, \ldots, a_n]$, $B = [b_1, \ldots, b_n]$, $S = [s_1, \ldots, s_n]$) such that $a_i(y), b_i(y), s_i(y) > 0, \; \forall \; i, y$ and $A \neq B$ (i.e., not all $a_i(y) = b_i(y)$) then there exists a finite α such that*

$$\mathcal{C}_{MP}(A, B) \geq \mathcal{C}_{MP}(S, S) = \mathcal{C}_{KL}(S).$$

The above theorem states that there exists a finite α that ensures $\mathcal{C}_{MP}(p, q)$ evaluated on any positive $p \neq q$ will be larger than any $\mathcal{C}_{KL}(\cdot)$. This is a stronger statement than we need, since we are interested only in what happens at the objective functions' minima. The following theorem does just this.

Theorem 8 (Equality of Solutions of $\mathcal{C}_{KL}$ and $\mathcal{C}_{MP}$) *Let*

$$\hat{\mathrm{p}} = \operatorname*{argmin}_{\mathrm{p} \in \triangle^m} \mathcal{C}_{KL}(\mathrm{p}) \text{ and } (\mathrm{p}^*_{\tilde{\alpha}}, \mathrm{q}^*_{\tilde{\alpha}}) = \operatorname*{argmin}_{\mathrm{p}, \mathrm{q} \in \triangle^m} \mathcal{C}_{MP}(\mathrm{p}, \mathrm{q}; \tilde{\alpha})$$

for an arbitrary $\tilde{\alpha} > 0$ where $\mathrm{p}^*_{\tilde{\alpha}} = (p^*_{1;\tilde{\alpha}}, \cdots, p^*_{m;\tilde{\alpha}})$ and $\mathrm{q}^*_{\tilde{\alpha}} = (q^*_{1;\tilde{\alpha}}, \cdots, q^*_{m;\tilde{\alpha}})$. Then there exists a finite $\hat{\alpha}$ such that at convergence of AM, we have that $\hat{\mathrm{p}} = \mathrm{p}^*_{\hat{\alpha}} = \mathrm{q}^*_{\hat{\alpha}}$. Further, it is the case that if $\mathrm{p}^*_{\tilde{\alpha}} \neq \mathrm{q}^*_{\tilde{\alpha}}$, then

$$\hat{\alpha} \geq \frac{\mathcal{C}_{KL}(\hat{\mathrm{p}}) - \mathcal{C}_{MP}(\mathrm{p}^*_{\tilde{\alpha}}, \mathrm{q}^*_{\tilde{\alpha}}; \alpha = 0)}{\mu \sum_{i=1}^n D_{KL}(p^*_{i;\tilde{\alpha}} || q^*_{i;\tilde{\alpha}})}$$

and if $\mathrm{p}^*_{\tilde{\alpha}} = \mathrm{q}^*_{\tilde{\alpha}}$ then $\hat{\alpha} \geq \tilde{\alpha}$.

We note that the above theorem guarantees the existence of a finite α that equates the minimum of $\mathcal{C}_{KL}$ and $\mathcal{C}_{MP}$ but it does not say how to find it since we do not know the true optimum of $\mathcal{C}_{MP}$. Nevertheless, if we use an α such that we end up with $p^* = q^*$ (or in practice, approximately so) then we are assured that this is the true optimum for $\mathcal{C}_{KL}$.

As mentioned above, AM is not always guaranteed to have closed form updates at each step, but in our case closed form updates may be achieved. The AM updates (see Appendix E for the derivation) are given by

$$p_i^{(n)}(y) = \frac{\exp\{\frac{\mu}{\gamma_i}\sum_j w'_{ij}\log q_j^{(n-1)}(y)\}}{\sum_y \exp\{\frac{\mu}{\gamma_i}\sum_j w'_{ij}\log q_j^{(n-1)}(y)\}} \quad \text{and}$$

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

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

Thus, $\mathcal{C}_{MP}$ satisfies all the desired properties of the reformulation. In addition, it is also possible to derive a test for convergence that does not require that one compute the value of $\mathcal{C}_{MP}(\mathrm{p},\mathrm{q})$ (i.e., evaluate the objective).

Theorem 9 (Test for convergence, see Appendix D) *If* $\{(\mathrm{p}^{(n)},\mathrm{q}^{(n)})\}_{n=1}^{\infty}$ *is generated by AM of* $\mathcal{C}_{MP}(\mathrm{p},\mathrm{q})$ *and* $\mathcal{C}_{MP}(\mathrm{p}^*,\mathrm{q}^*) \triangleq \inf_{\mathrm{p},\mathrm{q}\in\triangle^n} \mathcal{C}_{MP}(\mathrm{p},\mathrm{q})$ *then*

$$\mathcal{C}_{MP}(\mathrm{p}^{(n)},\mathrm{q}^{(n)}) - \mathcal{C}_{MP}(\mathrm{p}^*,\mathrm{q}^*) \le \sum_{i=1}^{n}\big(\delta(i \le l) + d_i\big)\beta_i,$$

$$\beta_i \triangleq \log\sup_y \frac{q_i^{(n)}(y)}{q_i^{(n-1)}(y)}, \quad d_j = \sum_i w_{ij}.$$

While a large number of optimization procedures resort to computing the change in the objective function with n (iteration index), in this case we have a simple check for convergence. This test does not require that one compute the value of the objective function which can be computationally expensive especially in the case of large graphs. Table 1 summarizes the advantages of the proposed AM approach to solving $\mathcal{P}_{MP}$ over that of using MOM to directly solve $\mathcal{P}_{KL}$. We also provide an empirical comparison of these approaches in Section 7.2.1. Henceforth, we refer to the process of using AM to solve $\mathcal{P}_{MP}$ as *measure propagation* (MP).

5. Squared-Loss Formulation

In this section, we show how the popular squared-loss objective may be formulated over measures. We then discuss its relationship to the proposed objective. Consider the optimization problem $\mathcal{P}_{SQ}$: $\min_{\mathrm{p}\in\triangle^m} \mathcal{C}_{SQ}(\mathrm{p})$ where

$$\mathcal{C}_{SQ}(\mathrm{p}) = \sum_{i=1}^{l} \| r_i - p_i \|^2 + \sum_{i=1}^{m}\sum_{j\in\mathcal{N}(i)} w_{ij} \| p_i - p_j \|^2 + \nu\sum_{i=1}^{m} \| p_i - u \|^2$$

and $\| p \|^2 = \sum_y p^2(y)$. $\mathcal{P}_{SQ}$ can also be seen as a multi-class extension of the *quadratic cost criterion* (Bengio et al., 2007) or as a variant of one of the objectives in Zhu and Ghahramani (2002b).

Criteria	MOM	AM
Iterative	YES	YES
Learning Rate	Armijo Rule	None
Number of Hyper-parameters	7	1 (α)
Test for Convergence	Requires Tuning	Automatic
Update Equations	Not Intuitive	Intuitive and easily Parallelized

Table 1: There are two ways to solving the proposed objective, namely, the popular numerical optimization tool method of multipliers (MOM), and the proposed approach based on alternating minimization (AM). This table compares the two approaches on various fronts.

Lemma 10 (Relationship between $\mathcal{C}_{KL}$ and $\mathcal{C}_{SQ}$) *We have that*

$$\mathcal{C}_{KL}(\mathrm{p}) \geq \frac{\mathcal{C}_{SQ}(\mathrm{p})}{\log 4} - m\nu \log |\mathrm{Y}|.$$

Proof By Pinsker's inequality we have that $D_{KL}(p||q) \geq (1/\log 4)\big(\sum_y |p(y)-q(y)|\big)^2 \geq (1/\log 4)\sum_y |p(y)-q(y)|^2$. As a result

$$\begin{aligned}
\mathcal{C}_{KL}(\mathrm{p}) &= \sum_{i=1}^{l} D_{KL}\big(r_i||p_i\big) + \mu \sum_{i=1}^{m} \sum_{j\in\mathcal{N}(i)} w_{ij} D_{KL}\big(p_i||p_j\big) - \nu \sum_{i=1}^{m} H(p_i) \\
&= \sum_{i=1}^{l} D_{KL}\big(r_i||p_i\big) + \mu \sum_{i=1}^{m} \sum_{j\in\mathcal{N}(i)} w_{ij} D_{KL}\big(p_i||p_j\big) + \nu \sum_{i=1}^{m} D_{KL}(p_i||u) - m\nu \log |\mathrm{Y}| \\
&\geq \frac{1}{\log 4}\left[\sum_{i=1}^{l} \| r_i - p_i \|^2 + \sum_{i=1}^{m} \sum_{j\in\mathcal{N}(i)} w_{ij} \| p_i - p_j \|^2 + \nu \sum_{i=1}^{m} \| p_i - u \|^2\right] - m\nu \log |\mathrm{Y}| \\
&= \frac{\mathcal{C}_{SQ}(\mathrm{p})}{\log 4} - m\nu \log |\mathrm{Y}|.
\end{aligned}$$

■

$\mathcal{P}_{SQ}$ can be reformulated as the following equivalent optimization problem $\mathcal{P}_{SQ}: \min_{\mathrm{p}\in\triangle^m} \mathcal{C}_{SQ}(\mathrm{p})$ where

$$\begin{aligned}
\mathcal{C}_{SQ}(\mathrm{p}) &= \mathrm{Tr}\big((S\mathrm{p}-\mathrm{r}')(S\mathrm{p}-\mathrm{r}')^T\big) + 2\mu \mathrm{Tr}(\mathbf{L}\mathrm{p}\mathrm{p}^T) + \nu \mathrm{Tr}((\mathrm{p}-\mathrm{u})(\mathrm{p}-\mathrm{u})^T), \\
S &\triangleq \begin{pmatrix} \mathbf{I}_l & 0 \\ 0 & 0 \end{pmatrix}, \quad \mathrm{r}' \triangleq \begin{pmatrix} \mathrm{r} & 0 \\ 0 & 0 \end{pmatrix}, \quad \mathrm{u} \triangleq (u,\ldots,u) \in \triangle^m,
\end{aligned}$$

$\mathbf{1}_m \in \mathbb{R}^m$ is a column vector of 1's, and $\mathbf{I}_l$ is the $l \times l$ identity matrix. Here $\mathbf{L} \triangleq \mathbf{D} - \mathbf{W}$ is the unnormalized graph Laplacian, $\mathbf{D}$ is a diagonal matrix given by $d_i = [\mathbf{D}]_{ii} = \sum_j w_{ij}$. $\mathcal{C}_{SQ}$ is convex if $\mu, \nu \geq 0$ and, as the constraints that ensure $\mathrm{p} \in \triangle$ are linear, we can make use of the KKT conditions (Bertsekas, 1999) to show that the solution to $\mathcal{P}_{SQ}$ is given by

$$\hat{\mathrm{p}} = (S + 2\mu\mathbf{L} + \nu\mathbf{I}_n)^{-1}\left[S\mathrm{r} + \nu\mathrm{u} + \frac{2\mu}{|\mathrm{Y}|}\mathbf{L}\mathbf{1}_n\mathbf{1}_c^T\right].$$

The above closed-form solution involves inverting a matrix of size $m \times m$. Henceforth we refer to the above closed form solution of $\mathcal{P}_{SQ}$ as *SQ-Loss-C* (C stands for closed form). Returning to the original formulation, using Lagrange multipliers, setting the gradient to zero and solving for the multipliers we get the update for p_i's to be

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

Here n is the iteration index. It can be shown that $\mathrm{p}^{(n)} \rightarrow \hat{\mathrm{p}}$ (Bengio et al., 2007). In the following we refer to the iterative method of solving $\mathcal{P}_{SQ}$ as *SQ-Loss-I*. There has not been any work in the past addressing the rate at which $\mathrm{p}^{(n)} \rightarrow \hat{\mathrm{p}}$ in the case of SQ-Loss-I. We address this issue in the following but first we define the rate of convergence of a sequence.

Definition 11 (Rate of Convergence Bertsekas, 1999) *Let $\{x_n\}$ be a convergent sequence such that $x_n \rightarrow 0$. It is said to have a linear rate of convergence if either*

$$x_n \leq q\eta^n \ \forall \ n \ \textit{or} \ \limsup_{n \rightarrow \infty} \frac{x_n}{x_{n-1}} \leq \eta$$

where $\eta \in (0,1)$ and $q > 0$.

As "geometric" rate of convergence is a more appropriate description of linear convergence, we use this term in the paper.

Theorem 12 (Rate of Convergence for SQ-Loss, see Appendix D) *If*

(a) $\nu > 0$, *and*

(b) $\mathbf{W}$ *has at least one non-zero off-diagonal element in every row (i.e.,* $\mathbf{W}$ *is irreducible)*

then the sequence of updates given in Equation 1 has a geometric rate of convergence for all i and y.

Thus we have that $\mathrm{p}^{(n)} \rightarrow \hat{\mathrm{p}}$ very quickly. It is interesting to consider a reformulation of $\mathcal{C}_{SQ}$ in a manner similar to $\mathcal{C}_{MP}$ (see Section 4.1), as we do next.

5.1 AM Amenable Formulation of $\mathcal{P}_{SQ}$

Consider the following reformulation of $\mathcal{C}_{SQ}$

$$\mathcal{C}'_{SQ}(\mathrm{p},\mathrm{q}) = \sum_{i=1}^{l} \| r_i - q_i \|^2 + \sum_{i=1}^{n} \sum_{j \in \mathcal{N}(i)} w'_{ij} \| p_i - q_j \|^2 + \nu \sum_{i=1}^{n} \| p_i - u \|^2 .$$

This form is amenable to AM and can be shown to satisfy 5-pp. Further the updates for two steps of AM have a closed form solution and are given by

$$p_i^{(n)}(y) = \frac{\nu u(y) + \mu \sum_j w'_{ij} q_j^{(n-1)}(y)}{\nu + \sum_j w'_{ij}},$$

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

We call this method *SQ-Loss-AM*. It is important to point out that for solving $\mathcal{P}_{SQ}$, one always resorts to either SQ-Loss-I or SQ-Loss-C depending on the nature of the problem. We will be using SQ-Loss-AM in the next section to provide more insights into the relationship between $\mathcal{P}_{KL}$ and $\mathcal{P}_{SQ}$.

6. Connections to Other Approaches

In this section we explore connections between our proposed approach and other previously proposed SSL algorithms.

6.1 Squared-Loss Based Algorithms

A majority of previously proposed graph-based SSL algorithms (Zhu et al., 2003; Joachims, 2003; Belkin et al., 2005; Bengio et al., 2007) are based on minimizing squared-loss. In the following we refer to the squared-loss based SSL algorithm proposed in Zhu and Ghahramani (2002a) as label propagation (LP) (this is the standard version of label propagation, see Table 2), the algorithm in Zhu et al. (2003) as the harmonic functions algorithms (HF). Also QC denotes the quadratic cost criterion (Bengio et al., 2007). While the objectives used in the case of LP, HF and QC are similar in spirit to our $\mathcal{C}_{SQ}$, there are some important differences. In the case of both HF and QC, the objective is defined over the reals whereas in our case $\mathcal{C}_{SQ}$ is defined over discrete probability measures. This leads to two important benefits – (a) it allows easy generalization to multi-class problems, (b) it allows us to exploit well-defined functions of measures in order to improve performance. Further, both the HF and LP algorithms do not have guards against degenerate solutions (i.e., the third term in $\mathcal{C}_{SQ}$). QC, on the other hand, employs a regularizer similar to the third term in $\mathcal{C}_{SQ}$ but QC is limited to only two-class problems (for multi-class problems one resorts to one vs. rest). Both the LP and HF algorithms optimize the same objective but LP uses a iterative solution while HF employs the closed form solution (it has been shown that LP converges to the solution given by HF Zhu, 2005a). QC is a generalization of HF and has been shown to outperform it (Bengio et al., 2007). Our squared-loss formulation, $\mathcal{C}_{SQ}$, is a generalization of QC for multi-class problems and as we show in Section 7.2.2, it outperforms QC. Thus, to compare against squared-loss based objectives, we simply use our formulation $\mathcal{C}_{SQ}$.

Table 2 summarizes the update equations in the case of some of the graph-based SSL algorithms. It is interesting to compare the update equations for SQ-Loss-AM and MP. It can be seen that the update equations for $q_i(y)$ in the case of SQ-Loss-AM and MP are the same. In the case of MP, the $p_i(y)$ update may be re-written as

$$p_i^{(n)}(y) = \frac{\prod_j \left(q_j^{(n-1)}(y)\right)^{\mu w'_{ij}}}{\sum_y \prod_j \left(q_j^{(n-1)}(y)\right)^{\mu w'_{ij}}}.$$

Thus, while squared loss makes use of a weighted arithmetic-mean, MP uses a weighted geometric-mean to update $p_i(y)$. In other words, while squared-error leads to additive updates, the use of KLD leads to multiplicative updates.

Algorithm	Update Equation(s)
MP	$p_i^{(n)}(y) = \frac{\exp\{\frac{\mu}{\gamma_i}\sum_j w'_{ij}\log q_j^{(n-1)}(y)\}}{\sum_y \exp\{\frac{\mu}{\gamma_i}\sum_j w'_{ij}\log q_j^{(n-1)}(y)\}}$ $q_i^{(n)}(y) = \frac{r_i(y)\delta(i\le l)+\mu\sum_j w'_{ji}p_j^{(n)}(y)}{\delta(i\le l)+\mu\sum_j w'_{ji}}$ $\gamma_i = \nu + \mu\sum_j w'_{ij}$
SQ-Loss-C	$\hat{\mathrm{p}} = (S+2\mu\mathbf{L}+\nu\mathbf{I}_m)^{-1}\left[S\mathrm{r}+\nu\mathrm{u}+\frac{2\mu}{\lvert\mathrm{Y}\rvert}\mathbf{L}\mathbf{1}_m\mathbf{1}_c^T\right]$ $\mathbf{L} \triangleq \mathbf{D}-\mathbf{W}$, $[\mathbf{D}]_{ii} = \sum_j w_{ij}$
SQ-Loss-I	$p_i^{(n)}(y) = \frac{r_i(y)\delta(i\le l)+\nu u(y)+\mu\sum_j w_{ij}p_j^{(n-1)}(y)}{\delta(i\le l)+\nu+\mu\sum_j w_{ij}}$
SQ-Loss-AM	$p_i^{(n)}(y) = \frac{\nu u(y)+\mu\sum_j w'_{ij}q_j^{(n-1)}(y)}{\nu+\sum_j w'_{ij}}$ $q_i^{(n)}(y) = \frac{r_i(y)\delta(i\le l)+\mu\sum_j w'_{ji}p_j^{(n)}(y)}{\delta(i\le l)+\mu\sum_j w'_{ji}}$
LP	$p_i^{(n)}(y) = \frac{r_i(y)\delta(i\le l)+\delta(i>l)\sum_j w_{ij}p_j^{(n-1)}(y)}{\delta(i\le l)+\delta(i>l)\sum_j w_{ij}}$

Table 2: A summary of update equations for various graph-based SSL algorithms. MP stands for our proposed measure propagation approach, SQ-Loss-C, SQ-Loss-I and SQ-Loss-AM represent the closed-form, iterative and alternative-minimization based solutions for the objective based on squared-error. LP is label propagation (Zhu and Ghahramani, 2002a). In all cases μ and ν are hyper-parameters.

Spectral graph transduction (SGT) (Joachims, 2003) is an approximate solution to the NP-hard norm-cut problem. The use of norm-cut instead of a mincut (as in Blum and Chawla, 2001) ensures that the number of unlabeled samples in each of the cuts is more balanced. SGT requires that one compute the eigen-decomposition of a $m \times m$ matrix which can be challenging for very large data sets. Manifold regularization (Belkin et al., 2005) proposes a general framework in which a parametric loss function that is defined over the labeled samples and is regularized by graph smoothness term defined over both the labeled and unlabeled samples. When the loss function satisfies certain conditions, it can be shown that the representer theorem applies and so the solution is a weighted sum over kernel computations. Thus the goal of the learning process is to discover these weights. When the parametric loss function is based on least squares, the approach is referred to as *Laplacian regularized least squares* (LapRLS) (Belkin et al., 2005) and when the loss function is based on hinge loss, the approach is called *Laplacian support vector machines* (LapSVM)) (Belkin et al., 2005). In the case of LapRLS, the weights have a closed form solution which involves inverting a $m \times m$ matrix while in the case of LapSVM, optimization techniques used for SVM training may be used to solve for the weights. In general, it has been observed that LapRLS and LapSVM give similar performance (see Chapter 11 in Chapelle et al., 2007). It very important to point out here that while LapSVM minimizes hinge loss (over the labeled samples) which is considered more appropriate than squared loss for classification, the graph regularizer is still based on squared error.

So is there a reason to prefer KLD based loss over squared-error? In this context we quote two relevant statements from Bishop (1995)

1. Page 226: *" In fact, the sum-of-squares error function is not the most appropriate for classification problems. It was derived from maximum likelihood on the assumption of Gaussian distributed target data. However, the target values for a l-of-c coding scheme are binary, and hence far from having a Gaussian distribution."*

2. Page 235: *"Minimization of the cross-entropy error function tends to result in similar relative errors on both small and large target values. By contrast, the sum-of-squares error function tends to give similar absolute errors for each pattern, and will therefore give large relative errors for small output values. This suggests that the cross-entropy error function is likely to perform better than sum-of-squares at estimating small probabilities."*

While the above quotes were made in the context of a multi-layered perceptron (MLP), they apply to learning in general. While squared-error has worked well in the case of regression problems (Bishop, 1995),[1] for classification, it is often argued that squared-loss is not the optimal criterion and alternative loss functions such as the cross-entropy (Bishop, 1995), logistic (Ng and Jordan, 2002), hinge-loss (Vladimir, 1998) have been proposed. When attempting to measure the dissimilarity between measures, KLD is said to be asymptotically consistent w.r.t. the underlying probability distributions (Bishop, 1995). The second quote above furthers the case in favor of adopting KLD based loss as it is based on relative error rather absolute error as in the case of squared-error. In addition, KLD is an ideal measure for divergence of probability distributions as it has description-length consequences (coding with the wrong distribution will lead to longer description bit length than necessary). Most importantly, as we will show in Section 7, MP outperforms the squared-error based $\mathcal{P}_{SQ}$ on a number of tasks. We also present further empirical comparison of these two objectives in Section 7.2.4.

We would like to note that Wang et al. (2008) proposed a graph-based SSL algorithm that also employs alternating minimization style optimization. However, it is inherently squared-loss based which MP outperforms (see Section 7). Further, they do not provide or state convergence guarantees and one side of their updates is not only not in the closed-form, but also it approximates an NP-complete optimization problem.

6.2 Information Regularization (Corduneanu and Jaakkola, 2003)

The information regularization (IR) (Corduneanu and Jaakkola, 2003) algorithm also makes use of a KLD based loss for SSL but is different from our proposed approach in following ways

1. IR is motivated from a different perspective. Here the input space is divided into regions $\{R_i\}$ which may or may not overlap. For a given point $x_j \in R_i$, IR attempts to minimize the KLD between $p_j(y|x_j)$ and $\hat{p}_{R_i}(y)$, the agglomerative distribution for region R_i. The intuition behind this is that, if a particular sample is a member of a region, then its posterior must be similar to the posterior of the other members. Given a graph, one can define a region to be a vertex and its neighbors thus making IR amenable to graph-based SSL. In Corduneanu and Jaakkola (2003), the agglomeration is performed by a simple averaging (arithmetic mean).

1. Assuming a Gaussian noise model in a regression problem leads to an objective based on squared-loss.

2. While IR suggests (without proof of convergence) the use of AM 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.

3. It does not make use of a entropy regularizer. But as our results show, the entropy regularizer leads to much improved performance.

Tsuda (2005) (hereafter referred to as PD) is an extension of the IR algorithm to hyper-graphs where the agglomeration is performed using the geometric mean. This leads to closed form solutions in both steps of the AM procedure. However, like IR, PD does not make use of a entropy regularizer. Further, 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 can be easily extended to hyper-graphs (see Section 9.3).

7. Results

Table 3 lists the data sets that we use in this paper. These corpora come from a diverse set of domains, including image processing (handwritten digit recognition), natural language processing (document classification, webpage classification, dialog-act tagging), and speech processing (phone classification). The sizes vary from $m = 400$ (BCI) to the largest data set, Switchboard, which has 120 million samples. The number of classes vary from $|\mathrm{Y}| = 2$ to $|\mathrm{Y}| = 72$ in the case of Dihana. The goal is to show that the proposed approach performs well on both small and large data sets, for binary and multi-class problems. Further, in each case we compare the performance of MP against the state-of-the-art algorithm for that task. Each data set is described in detail in the relevant sections.

7.1 Synthetic 2D Two-Moon Data Set

In order to understand the advantages of MP over other state-of-the-art SSL algorithms, we evaluated their performance on the synthetic 2D two-moon data set. This is a binary classification problem. We compare against SQ-Loss-I (see Section 5), LapRLS (Belkin et al., 2005), and SGT (Joachims, 2003). For all approaches, we constructed a symmetrized 10-NN graph using an RBF kernel. In the case of LapRLS and SGT, the hyper-parameter values were set in accordance to the recipe in Belkin et al. (2005) and Joachims (2003) respectively. In the case of MP, we set $\mu = 0.2$, $\nu = 0.001$ and $\alpha = 1.0$. For SQ-Loss-I, we set $\mu = 0.2$ and $\nu = 0.001$. These values were found to give reasonable performance for most data sets.

We used three different types of labelings: (a) two labeled samples from each class, (b) 4 samples from one class and 1 sample from the other class, and (c) 10 samples from one class and 1 sample from the other class. While the first represents the 'balanced' case, that is, equal number of labeled samples from the two classes, the others are 'imbalanced' conditions. In other words, (b) and (c) are representative of cases where the distribution over the labeled samples is not reflective of the underlying distribution over the classes (there are equal number of samples in each class). The results for each of the different labeling are shown in Figure 3. The first column shows the results obtained using SQ-Loss-I, the second column shows the results of LapRLS, the third is SGT and the fourth (last) column is MP. The following observations can be made from these results

Data Set	m	$\|\mathrm{Y}\|$	$H_N(p_0)$	Task
2D Two-Moon	500	2	1	Synthetic
BCI	400	2	1	Brain Computer Interface
USPS	1500	2	0.7	HandWritten Digits
Digit1	1500	2	1	Synthetic
COIL	1500	6	1	Image Recognition
Text	1500	2	1	Newsgroups Newswires
OPT-Digits	1797	10	1	HandWritten Digits
Reuters-21578	9603	10	0.8	Document Classification
WebKB	8282	4	0.9	Webpage Classification
Dihana	23,500	72	0.8	Dialog-Act Tagging
Switchboard-DA	185,000	18	0.6	Dialog-Act Tagging
TIMIT	1.4 million	48	0.9	Phone Classification
Switchboard	120 million	53	0.8	Phone Classification

Table 3: List of Data Sets we used to compare the performance of various SSL algorithms. $H_N(p_0) = H(p_0)/\log|\mathrm{Y}|$ is the normalized entropy of the prior and a value of 1 indicates a perfectly balanced data set while values closer to 0 imply imbalance. In the case of the Switchboard data set, $H_N(p_0)$ was computed over the STP data (see Section 8.1).

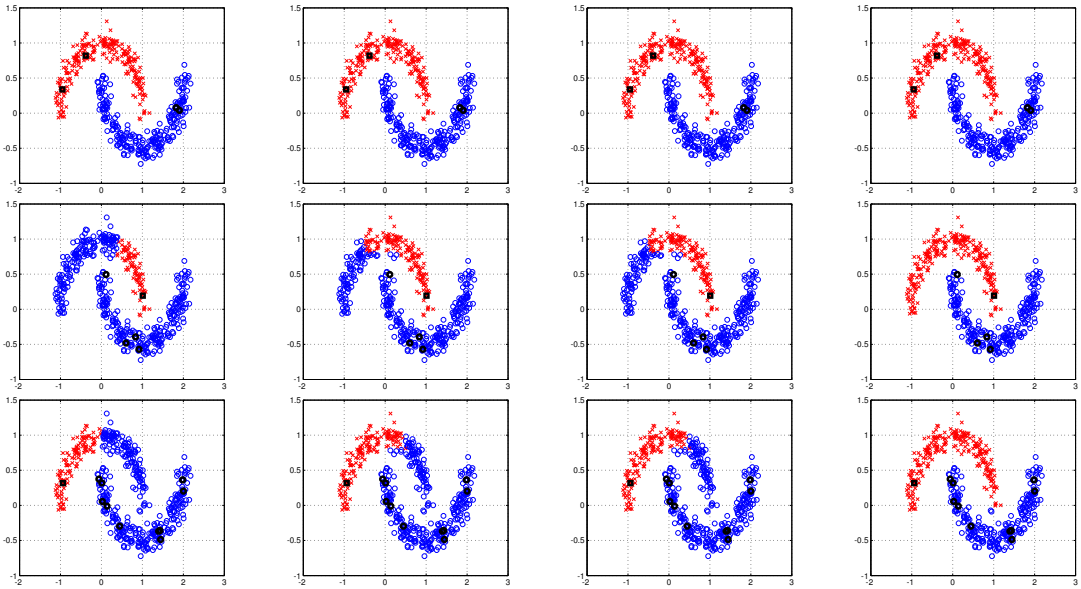

Figure 3: Results on the 2D two-moon data set. Each row shows results for different labelings and in each case the labeled points are shown in "black". The first column shows results obtained using SQ-Loss-I, the second column results were obtained using LapRLS, SGT was used for the third column and the last column shows the results in the case of MP.

1. MP is able to achieve perfect classification in the first two cases, and essentially perfect (2 errors) in the third case.

2. In the balanced case (first row), all approaches achieve perfect classification. Here, all approaches are able to correctly learn the nature of the manifold.

3. In the imbalanced cases (second and third rows), all three other approaches (SQ-Loss-I, LapRLS, and SGT) fail to correctly classify a significant portion of samples. This is not surprising and has been observed by others in the past (see Figure 1 in Wang et al., 2008).

4. Finally, in the case of SQ-Loss-I, we tried using class mass normalization (CMN) (Zhu and Ghahramani, 2002a) as a post-processing step. While the results did not change in the balanced case, CMN in fact resulted in worse error rate performance in the imbalanced cases. Note that Figure 3 for SQ-Loss-I does not include CMN.

7.2 Results on Benchmark SSL Data Sets

We also evaluated the performance of MP on a number of benchmark SSL data sets including, USPS, Text, Digit1, BCI, COIL and Opt-Digits. All the above data sets, with the exception of Opt-Digits (obtained from the UCI machine learning repository), came from `http://www.kyb.tuebingen.mpg.de/ssl-book`. Digit1 is a synthetic data set, USPS is a handwritten digit recognition task, BCI involves classifying signals obtained from a brain computer interface, COIL is a part of the Columbia object image recognition library and involves classifying objects using images taken at different orientations. Text involves classifying IBM vs. the rest for documents taken from the top 5 categories in comp.* newswire. Opt-Digits is also a handwritten digit recognition task. We note that most of these data sets are perfectly balanced (see Table 3)—further details may be found in Chapelle et al. (2007).

We compare MP against four other algorithms: 1) k-nearest neighbors; 2) Spectral Graph Transduction (SGT) (Joachims, 2003); 3) Laplacian Regularized Least Squares (LapRLS) (Belkin et al., 2005); and 4) $\mathcal{P}_{SQ}$ solved using SQ-Loss-I. Here k-nearest neighbors is the fully-supervised approach, while others are graph-based SSL approaches. We used the standard features supplied with the corpora without any further processing. For the graph-based approaches we constructed symmetrized k-NN graphs using an RBF kernel. We discuss the choice of k and the width of the kernel shortly. For each data set, we generated transduction sets with different number of labeled samples, $l \in \{10, 20, 50, 80, 100, 150\}$. In each case, we generated 11 different transduction sets. The first set was used to tune the hyper-parameters which were then held fixed over the remaining sets. In the case of the k-nearest neighbors approach, we tried $k \in \{1, 2, 4, 5, 10, 20, 30, 40, 50, 70, 90, 100, 120, 140, 150, 160, 180, 200\}$. For the graph-based approaches, k (for the k-NN graph) was tuned on the first transduction set over the following values $k \in \{2, 5, 10, 50, 100, 200, m\}$. The optimal width of the RBF kernel, σ, in the case of SQ-Loss-I, SGT and MP was determined over the following set $\sigma \in \{g_a/3 : a \in \{2, 3, \cdots, 10\}\}$ where g_a is the average distance between each sample and its a^{th} nearest neighbor over the entire data set (Bengio et al., 2007).

In the case of LapRLS, we followed the setup described in Section 21.2.5 of Chapelle et al. (2007). Here, as per the recipe in Joachims (2003), the optimal σ was determined in a slightly different manner—we tried $\sigma \in \{\frac{\sigma_0}{8}, \frac{\sigma_0}{4}, \frac{\sigma_0}{2}, \sigma_0, 2\sigma_0, 4\sigma_0, 8\sigma_0\}$ where σ_0 is the average norm of the feature vectors. In addition the hyper-parameters γ_A, r (see Belkin et al., 2005) associated with LapRLS were tuned over the following values: $\gamma_A \in \{$1e–6, 1e–4, 1e–2, 1, 100$\}$, $r \in \{0,$ 1e–4, 1e–2, 1, 100,

	USPS						**Digit1**					
l	10	20	50	80	100	150	10	20	50	80	100	150
k-NN	80.0	80.4	90.7	92.7	93.6	94.9	67.6	79.5	90.2	93.2	91.2	95.2
SGT	86.2	87.9	**94.0**	**95.7**	**96.0**	**97.0**	**92.1**	93.6	96.2	**97.1**	**97.4**	**97.7**
LapRLS	83.9	86.9	**93.7**	94.7	95.4	95.9	**92.4**	**95.3**	95.7	96.2	97.1	97.4
SQ-Loss-I	81.4	82.0	**93.6**	**95.8**	95.2	95.2	91.2	94.9	**96.9**	96.6	97.2	97.1
MP	**88.1**	**90.4**	**93.9**	95.0	**96.2**	**96.8**	**92.1**	**95.1**	96.1	**97.4**	**97.4**	**97.8**

	BCI						**Text**					
l	10	20	50	80	100	150	10	20	50	80	100	150
k-NN	48.5	52.4	53.3	50.6	53.1	53.5	60.2	64.2	71.6	72.4	72.3	74.5
SGT	49.7	50.4	**52.2**	52.4	**53.6**	54.5	**70.4**	70.9	**73.1**	**76.9**	**77.0**	**78.1**
LapRLS	**53.3**	**53.4**	**52.7**	**53.6**	**53.9**	56.1	68.2	69.1	71.2	73.4	74.2	76.2
SQ-Loss-I	51.0	51.3	50.7	**53.2**	53.3	53.1	67.9	**72.0**	**74.1**	**76.8**	**76.8**	76.6
MP	**53.0**	**53.2**	**52.8**	**53.9**	**54.0**	**57.0**	**70.3**	**72.6**	**73.0**	75.9	75.4	**77.9**

	COIL						**OPT**					
l	10	20	50	80	100	150	10	20	50	80	100	150
k-NN	34.5	53.9	66.9	77.9	79.2	83.5	79.6	83.9	85.5	90.5	92.0	93.8
SGT	40.1	61.2	78.0	88.5	89.0	89.9	90.4	90.6	91.4	94.7	**97.4**	**97.4**
LapRLS	**49.2**	61.4	78.4	80.1	84.5	87.8	89.7	**91.2**	92.3	96.1	**97.6**	**97.3**
SQ-Loss-I	**48.9**	63.0	**81.0**	87.5	89.0	90.9	**92.2**	90.2	**95.9**	**97.2**	**97.3**	**97.7**
MP	47.7	**65.7**	78.5	**89.6**	**90.2**	**91.1**	90.6	**90.8**	94.7	**96.6**	**97.0**	**97.1**

Table 4: Comparison of accuracies for different number of labeled samples (l) across USPS, Digit1, BCI, Text, COIL and Opt-Digits data sets. In each column, the best performing system and all approaches that are not significantly different at the 0.001 level (according to the difference of proportions significance test) are shown bold-faced.

1e4, 1e6$\}$. Also, as per Belkin et al. (2005), we set $p = 5$ in the case of Text data set and $p = 2$ for all the other data sets. In the case of SGT, the search was over $c \in \{$3000, 3200, 3400, 3800, 5000, 100000$\}$ (Joachims, 2003). Finally, the trade-off parameters, μ and ν (associated with both MP and SQ-Loss-I) were tuned over the following sets: $\mu \in \{$1e–8, 1e–6, 1e–4, 1e–2, 0.1, 1, 10$\}$ and $\nu \in \{$1e–8, 1e–6, 1e–4, 1e–2, 0.1$\}$. In the case of SQ-Loss-I, the results were obtained after the application of CMN as a post-processing step as this has been shown to be beneficial to the performance on benchmark data sets (Chapelle et al., 2007). For MP, we initialized $\mathrm{p}^{(0)}$ such that all assignments had non-zero probability mass as this is a required condition for convergence and set $\alpha = 1$. As LapRLS and SGT assume binary classification problems, results for the multi-class data sets (COIL and OPT) were obtained using one vs. rest.

The mean accuracies over the 10 transduction sets (i.e., excluding the set used for tuning the hyper-parameters) for each corpora is shown in Table 4. The following observations may be made from these results

1. As expected, for all approaches, an increase in number of labeled samples leads to increased accuracy.

	USPS	Text	Digit1	BCI	COIL	Opt-Digits
LP-3	77.2	65.1	70.1	51.5	31.3	81.2
MOM	88.1	70.3	91.4	53.0	46.1	91.2
MP	88.2	70.3	92.1	53.0	47.7	93.4
MOM′	81.1	67.6	79.4	51.7	41.2	90.4

Table 5: Comparison of performance of MOM and MP. Results are in accuracies for the $l = 10$ case. We also show the results obtained after three iterations of LP (LP-3) (Zhu and Ghahramani, 2002a) as this was used to initialize MOM. MOM′ are the results obtained using the MOM setup with a small change in the setting of the hyper-parameters.

2. MP performs best in 15 out of the 36 cases, SQ-Loss is best in 10 out of the 36 cases, SGT in 8 out of the 36 cases and LapRLS in 7 out of the 36 cases. In 13 of cases in which MP was not the best, it was not significantly different compared to the winner (we characterize an improvement as being significant if it is significant at the 0.001 level according to a difference of proportions significance test).

3. It can be seen that SGT does best in the case of the Text corpus for a majority of the values of l, while MP is the best in a majority of the cases in the COIL and BCI data sets. SQ-Loss does best in the case of OPT. Thus in the case of the two multi-class data sets, the two *true* multi-class approaches perform better than the SSL approaches that use one vs. rest.

4. We also tried SQ-Loss-C and SQ-Loss-AM for solving the squared-loss based objective and in a majority of the cases the performance was the same as SQ-Loss-I. In other cases, the difference was insignificant. It should however be noted that using SQ-Loss-C to solve large problems can be rather difficult.

5. While there are no silver bullets in SSL (Zhu, 2005b), our MP algorithm outperforms other approaches in a majority of the cases. We would like to point out the diversity of the data sets used in the above experiment.

6. Finally note that while we have a used a simple approach to hyper-parameter selection, there are other ways of choosing them such as Goldberg and Zhu (2009)

7.2.1 MP VS. MOM

In this section we compare the results obtained from using MP against results obtained by directly optimizing the original objective, $\mathcal{C}_{KL}$ (henceforth we refer to this as MOM). As explained in Section 3.1, implementing MOM requires the careful tuning of a number optimization related hyper-parameters (in addition to μ and ν). After extensive experimentation, we found that setting, $\gamma = 0.25$, $\beta = 5$ and $\zeta =$1e–6 gave reasonable results. Further, as MOM is gradient based, we initialized $\mathrm{p}^{(0)}$ (see Section 3.1) to the distributions obtained after 3 iterations of the label propagation algorithm described in Zhu and Ghahramani (2002a) (henceforth referred to as LP-3).

Table 5 shows average accuracies over all transduction sets for $l = 10$ (the trends were similar for other values of l) in the case of the corpora described in the previous section for (a) LP-3, (b) MOM (c) MP, and (d) MOM′. In the case of MOM′, we changed the values of the optimization

related hyper-parameters to $\gamma = 0.2$ and $\beta = 3$. The goal here is to show the sensitivity of MOM′ to the exact settings of the hyper-parameter values. The following observations can be made from these results

1. MOM outperforms LP-3. This implies MOM is able to learn over and beyond the set of distributions that result from 3 iterations of LP.

2. In the case of USPS, Digit1, COIL, Opt-Digits, MP outperforms MOM. Further, the performance gap between MP and MOM grows with the size of the data set. MP significantly outperforms MOM at the 0.0001 level in the case of the Opt-Digits. This might seem surprising because when we have that $p^* = q^*$ in the case of MP, the results obtained using MOM cannot be any worse than those obtained using MP (because the objective is convex). We conjecture that this is because MOM involves using a penalty parameter $c^{(n)}$ that tends to increase with n leading to slow convergence. This is more likely to happen in the neighborhood of p^* (Bertsekas, 1999). As a result MOM is terminated when the rate of the change of $p^{(n)}$ falls below some ζ and so it is possible that the objective has not attained the optimal value. In the case of MP, on the other hand, no such issues exist. Further we have a test for convergence (see Theorem 9).

3. The results obtained in the case of MOM′ show that this approach can be very sensitive to the settings of the hyper-parameters. While it may be possible to tune the various MOM related hyper-parameters in the case of small data sets, it is much less feasible in the case of large data sets.

7.2.2 ONE VS. REST AGAINST TRUE MULTI-CLASS

It is often argued binary classifiers when used within a one vs. rest framework perform at least as well as true multi-class solutions (Rifkin and Klautau, 2004). In this section, we test this claim in the context of SSL. We make use of the two multi-class data sets, COIL and OPT-Digits. Figure 4 shows a comparison of the performance of $\mathcal{P}_{SQ}$ (solved using SQ-Loss-C) and QC (Bengio et al., 2007). Even though SQ-Loss-I converges to SQ-Loss-C, in this case we used SQ-Loss-C as the size of the data set is small. As QC can handle only binary classification problems, the results for QC were generated using one vs. rest. Note that SQ-Loss-C is simply the closed form solution of $\mathcal{P}_{SQ}$ which is the multi-class extension of the QC objective. In the case of both the approaches, (a) the graph was generated by using an RBF kernel over the Euclidean distance, (b) we used the closed form solution, and (c) hyper-parameter search was done over exactly the same set of values. It can be seen that SQ-Loss-C outperforms QC in all cases. As the objectives are both inherently based on squared-error, the performance improvement in going from QC to $\mathcal{P}_{SQ}$ is likely because $\mathcal{P}_{SQ}$ is a true multi-class objective, that is, all the parameters are estimated jointly.

7.2.3 EFFECTS OF ENTROPY REGULARIZATION

We also wish to explore the effects of the entropy regularizer. We ran MP using the same setup described in Section 7.2 but with $\nu = 0$. The results in the $l = 10$ case are shown in Table 6. Similar trends were observed in the case of other values of l. It can be seen that entropy regularization leads to improved performance in the case of all data sets. We moreover have seen this trend in the other data sets (results not reported herein). The entropy regularizer encourages solutions closer to

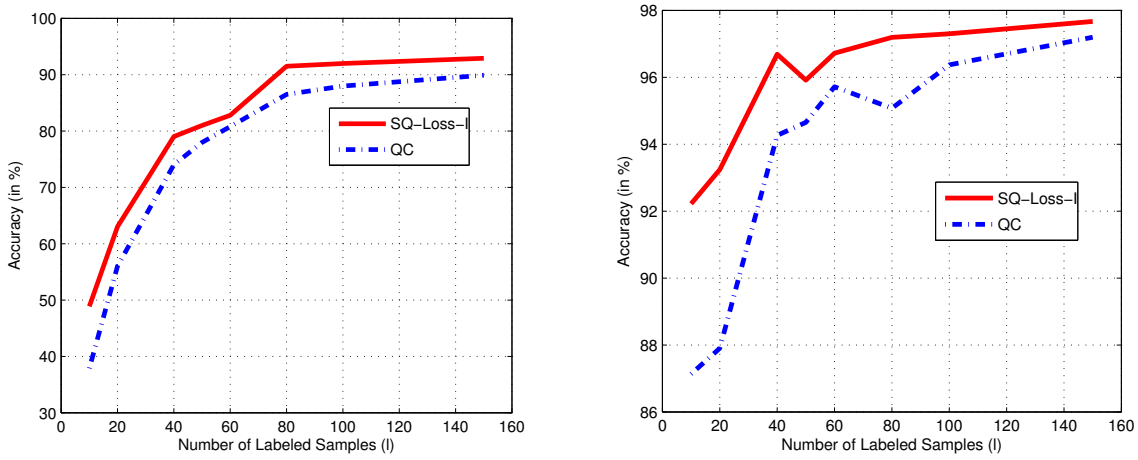

Figure 4: Comparison of the one vs. rest approach against true multi-class classifier. Figures show accuracy (in %) vs. Number of Labeled Samples (l) for (a)-left COIL and (b)-right OPT-DIGITS data sets. SQ-Loss-I is the solution to a true multi-class objective while QC makes use of one vs. rest approach for multi-class problems.

	USPS	Text	Digit1	BCI	COIL	Opt-Digits
MP ($\nu = 0$)	85.7	70.0	91.7	51.1	45.2	89.5
MP	88.2	70.3	92.1	53.0	47.7	93.4

Table 6: Comparison of performance of MP with and without ($\nu = 0$) entropy regularization. Results are in accuracies for the $l = 10$ case.

the uniform distribution, and we mentioned above that this helps to retain uncertainty in portions of graph very isolated from label information. To explain why this could lead to actual *improved* performance, however, we speculate that the entropy term is beneficial for the same reason as that of maximum entropy estimation—except for evidence to the contrary, we should prefer solutions that are as indifferent as possible.

7.2.4 SENSITIVITY OF MP AND SQ-LOSS-I TO σ

In this section, we examine the effects of change in hyper-parameters settings on the performance of $\mathcal{P}_{SQ}$ (solved using SQ-Loss-I) and MP. In particular, we look at the effects of varying the width of the RBF kernel used to generate the weighted graph. Figure 5 shows results obtained for the $l = 50$ case in the USPS and Opt-Digits data sets (in each case the value of σ at the mode of each curve is its optimal value). It can be seen that in the case of both the data sets, the performance variation is larger in the case of SQ-Loss-I while MP is more robust to the value of σ. Note that in the case of Opt-Digits, at the optimal value for σ, SQ-Loss-I outperforms MP. Similar trends were observed in the case of other data sets. As the choice of hyper-parameters in an issue in SSL, we prefer approaches that are more robust to the value of the hyper-parameters. We believe the robustness

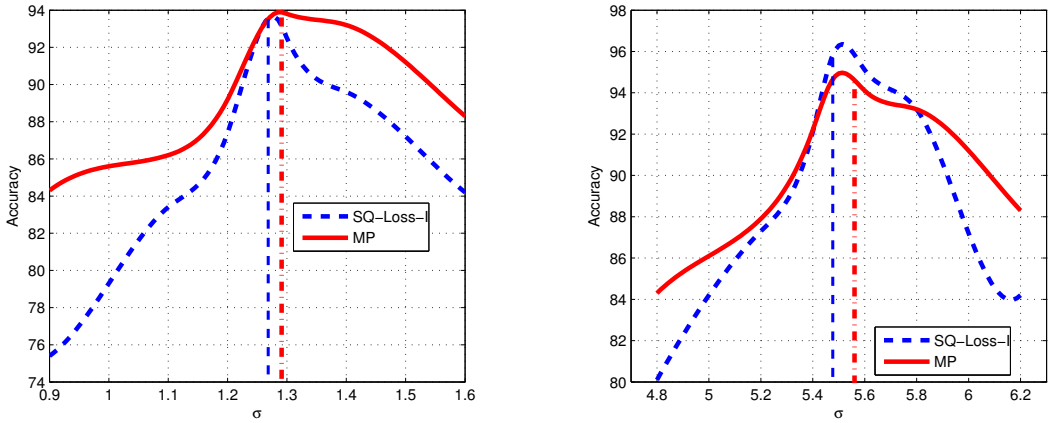

Figure 5: Figures showing the variation of accuracy with change in the width (σ) of the RBF kernel. The left figure was generated using the USPS data set for the $l = 50$ case while the right figure was generated using the Opt-Digits data set for the $l = 50$ case. The vertical lines (blue for SQ-Loss-I and red for MP) depict the σ given by the algorithm described in the previous section.

of MP is due to the fact that it is inherently based on KLD which is more suited for classification compared to squared error.

7.3 Text Classification

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, 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 compare our algorithm (MP) with other state-of-the-art text categorization algorithms, namely: (a) SVM (Joachims, 1999); (b) Transductive-SVM (TSVM) (Joachims, 1999); (c) Spectral Graph Transduction (SGT) (Joachims, 2003); and (d) $\mathcal{P}_{SQ}$ solved using SQ-Loss-I. Apart from MP, SGT and SQ-Loss-I are graph-based algorithms, while SVM is fully-supervised (i.e., it does not make use of any of the unlabeled data). As shown by the results in Joachims (2003), SGT outperforms other SSL algorithms for this task. Thus we choose to compare against SGT. We implemented SVM and TSVM using *SVM Light* (Joachims, 2002) and SGT using *SGT Light* (Joachims, 2004). 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). We used two real-world data sets: (a) Reuters-21578 and (b) WebKB. In the following we discuss the application of the above algorithms to these data sets.

7.3.1 REUTERS-21578

We used the "ModApte" split of the Reuters-21578 data set 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 data set, 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). We constructed symmetrized k-NN graphs with weights generated using cosine similarity between TFIDF features generated as explained above.

For this task $\mathrm{Y} = \{$ *earn*, *acq*, *money*, *grain*, *crude*, *trade*, *interest*, *ship*, *wheat*, *corn*, *average*$\}$. For SQ-Loss-I and MP, 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 $r_i(acq) = r_i(grain) = r_i(wheat) = 1/3$. Note that there might be other (non-uniform) ways of initializing r_i (e.g., using word counts).

We created 21 transduction sets by randomly sampling l documents from the standard Reuters 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, SQ-Loss-I and MP, the first transduction set was used to tune the hyper-parameters which we then held fixed for all the remaining 20 transduction sets. For all the graph-based approaches, we ran a search over $k \in \{2, 10, 50, 100, 250, 500, 1000, 2000, m\}$ (note $k = m$ represents a fully connected graph, i.e., a clique). In addition, in the case of MP, 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 \{$1e–8, 1e–6, 1e–4, 0.01, 0.1$\}$. In the case of SGT, the search was over $c \in \{3000, 3200, 3400, 3800, 5000, 100000\}$ (Joachims, 2003).

We report precision-recall break even point (PRBEP) results on the 3,299 test documents in Table 7. 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 7 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 hyper-parameters in case of SQ-Loss-I was $k = 100$; in case of MP, $k = 1000$, $\mu =$ 1e–4, $\nu =$ 1e–4; and in the case of SGT, $k = 100$, $c = 3400$. The results show that MP 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. MP is significantly better at the 0.001 level over its nearest competitor (SGT) according to a difference of proportions significance test.

Figure 6 shows the variation of "average" PRBEP (last row in Table 7) against the number of labeled documents (l). For each value of l, we tuned the hyper-parameters over the first transduction set and used these values for all the other 20 sets. Figure 6 also shows error-bars ($\pm$ standard deviation) for all the experiments. As expected, the performance of all the approaches improves with increasing number of labeled documents. Once again in this case, MP, outperforms the other approaches for all values of l.

Category	SVM	TSVM	SGT	SQ-Loss-I	MP
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 7: 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.

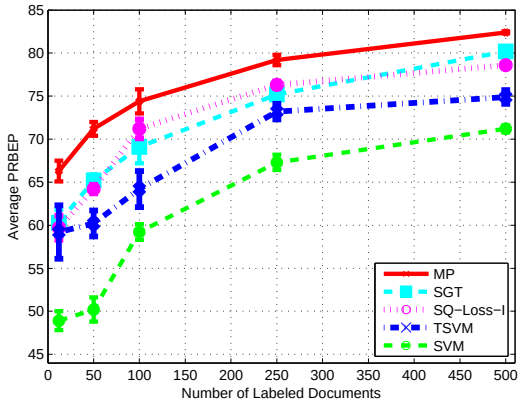

Figure 6: Average PRBEP over all classes vs. number of labeled documents (l) for Reuters data set

7.3.2 WEBKB COLLECTION

World Wide Knowledge Base (WebKB) is a collection of 8282 web pages obtained from four academic 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 case of the Reuters data set we extracted TFIDF features for each document and constructed the graph using cosine similarity.

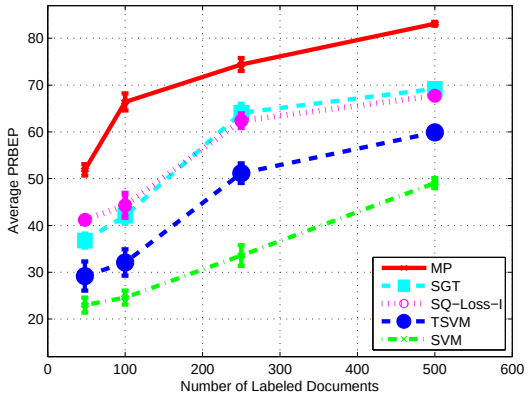

Figure 7: Average PRBEP over all classes vs. number of labeled documents (l) for WebKB collection.

Class	SVM	TSVM	SGT	SQ-Loss-I	MP
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 8: 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

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 sampled 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 hyper-parameters. 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 8. For this task, we found that the optimal value of the hyper-parameter were: in the case of SQ-Loss-I, $k = 1000$; in case of AM, $k = 1000$, $\mu =$ 1e–2, $\nu =$ 1e–4; and in case of SGT, $k = 100$, $c = 3200$. Once again, MP significantly outperforms the state-of-the-art (results are significant at the 0.0001 level). Figure 7 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.

7.4 TIMIT Phone Recognition

The TIMIT corpus of read speech was designed to provide speech data for acoustic-phonetic studies and for the development and evaluation of automatic speech recognition systems (Zue et al., 1990). TIMIT contains broadband recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences. The corpus includes time-aligned phonetic transcriptions and has standard training (3896 utterances) and test (196 utterances) sets. For hyper-parameter tuning, as TIMIT does not define a development set, we created one with 50 TIMIT utterances (independent of the training and test sets). In the past, TIMIT has been used almost exclusively to evaluate the performance of supervised learning algorithms (Halberstadt and Glass, 1997; Somervuo, 2003). Here, we use it to evaluate SSL algorithms by using fractions of the standard TIMIT training set obtained by random sampling. This simulates the case when only small amounts of data are labeled. We compare the performance of MP against that of

(a) ℓ_2 regularized 2-layer multi-layered perceptron (MLP) (Bishop, 1995), and

(b) $\mathcal{P}_{SQ}$ solved using SQ-Loss-I.

Recall that, while MLPs are fully-supervised, SQ-Loss-I and MP are both graph-based SSL algorithms. We choose ℓ_2 regularized MLPs as they have been shown to beat SVMs for the phone classification task (Li and Bilmes, 2006).

To obtain the acoustic observations, $\mathbf{x}_i$, the signal was first pre-emphasized ($\alpha = 0.97$) and then windowed using a Hamming window of size 25ms at 100Hz. We then extracted 13 mel-frequency cepstral coefficients (MFCCs) (Lee and Hon, 1989) from these windowed features. Deltas were appended to the above resulting in 26 dimensional features. As phone classification performance is improved by context information, we appended each frame with 3 frames from the immediate left and right contexts and used these 182 dimensional feature vectors as inputs to the classifier. These features were used to construct a symmetrized 10-NN graph over the entire training and development sets. This graph had 1,382,342 vertices. The weights are given by

$$w_{ij} = \text{sim}(\mathbf{x}_i, \mathbf{x}_j) = \exp\{-(\mathbf{x}_i - \mathbf{x}_j)^T \Sigma^{-1} (\mathbf{x}_i - \mathbf{x}_j)\}$$

where Σ is the covariance matrix computed over the entire TIMIT training set. We follow the standard practice of mapping the original set of 61 phones in TIMIT down to 48 phones for modeling ($|\mathrm{Y}| = 48$) and then a further mapping to 39 phones for scoring (Lee and Hon, 1989).

For each approach the hyper-parameters were tuned on the development set by running an extensive search. In the case of the MLP, the hyper-parameters include the number of hidden units and the regularization coefficient. For MP and SQ-Loss-I, the hyper-parameters were tuned over the following sets $\mu \in \{$1e–8, 1e–4, 0.01, 0.1$\}$ and $\nu \in \{$1e–8, 1e–6, 1e–4, 0.01, 0.1$\}$. We found that setting $\alpha = 1$ in the case of MP ensured that p = q at convergence. As both MP and SQ-Loss-I are transductive, in order to measure performance on an independent test set, we induce the labels using the Nadaraya-Watson estimator, that is, given an input sample, $\hat{\mathbf{x}}$, that we wish to classify, the output is given by

$$\hat{y} = \operatorname*{argmax}_{y \in \mathrm{Y}} \hat{p}(y) \text{ where } \hat{p}(y) = \frac{\sum_{j \in \mathcal{N}(\hat{\mathbf{x}})} \text{sim}(\hat{\mathbf{x}}, \mathbf{x}_j) p_j^*(y)}{\sum_{j \in \mathcal{N}(\hat{\mathbf{x}})} \text{sim}(\hat{\mathbf{x}}, \mathbf{x}_j)},$$

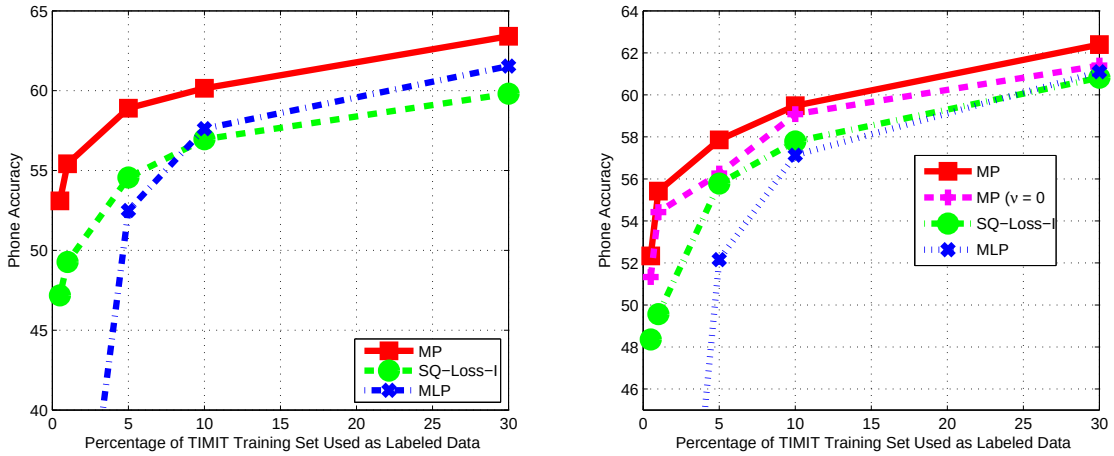

Figure 8: Phone Accuracy (PA) on the TIMIT development set (left) and TIMIT NIST core evaluation/test set (right). The x-axis shows the percentage of standard TIMIT training data that was treated as being labeled.

$\mathcal{N}(\hat{\mathbf{x}})$ is the set of nearest neighbors of $\hat{\mathbf{x}}$ in the training data (i.e., all the samples over which the graph was constructed) and p_j^* is the converged value of p_j. In our experiments we have that $|\mathcal{N}(\hat{\mathbf{x}})| = 50$.

The left plot in Figure 8 shows the phone classification results on the TIMIT development set while the right plot shows the results on the NIST Core test set. The y-axis shows phone accuracy (PA) which represents the percentage of frames correctly classified and the x-axis shows the fraction f of the training set that was treated as being labeled. We show results for $f \in \{0.005, 0.05, 0.1, 0.25, 0.3\}$. Note that in each case we use the same graph, that is, only the set of labeled vertices V_l changes depending on f. The following observations may be made from these results:

1. MP outperforms the SQ-Loss-I objective for all cases of f. This lends further weight to the claim that KLD based loss is more suitable for classification problems.

2. When little labeled training data is available, both SQ-Loss-I and MP significantly outperform the MLP. For example when 0.5% of the training set is labeled, the PA in the case of MP was 52.3% while in the MLP gave a PA of 19.6%. This is not surprising as the MLP does not make use of the unlabeled data. It remains to be tested if semi-supervised MLP training (Malkin et al., 2009) would reduce or reverse this difference.

3. Even when 10% of the original TIMIT training set is used, MP gives a PA of about 60% and outperforms both the MLP and SQ-Loss-I.

4. It is interesting to note that an MLP trained using the entire training set (i.e., it had 100% of the labeled samples) resulted in a PA of 63.1%. But using about 30% of this data, MP gives a PA of about 62.4%.

	Bigram	Trigram
SQ-Loss-I	75.6%	76.9%
MP	81.0%	81.9%

Table 9: Dialog-Act Tagging Accuracy results on the Dihana Corpus. The results are for the case of classifying user turns. The baseline DA accuracy was 76.4% (Martínez-Hinarejos et al., 2008)

5. We also found that for larger values of f (e.g., at $f = 1$), the performances of MLP and MP did not differ significantly. But those are more representative of the supervised training scenarios which is not the focus here.

6. A comparison of the curves for MP with and without entropy regularization illustrates the importance of the graph-regularizer (second term in C_{KL} and C_{MP}).

7.5 Dialog-Act Tagging

Discourse patterns in natural conversations and meetings are well known indicators of interesting and useful information about human conversational behavior. Dialog acts (DA) which reflect the functions that utterances serve in discourse are one type of such patterns. Detecting and understanding dialog act patterns can provide benefit to systems such as automatic speech recognition, machine translation and general natural language processing (NLP). In this section we present dialog-act tagging results on two tasks: (a) Dihana, and (b) SWB.

7.5.1 DIHANA DA TAGGING

Dihana is a Spanish dialog corpus. It is composed of 900 task-oriented computer-human spoken dialogs collected via a train reservation system. Typical topics include timetables, fares, and services offered on trains. The size of the vocabulary is 823 words. Dihana was acquired from 225 different speakers (153 male and 72 female). On average, each dialog consisted of 7 user turns and 10 system turns, with an average of 7.7 words per user turn. The corpus has three tasks which include classifying the DAs of the (a) user turns, (b) system turns, and (c) both user and system turns. Each of these tasks has training, test and development sets setup for 5-fold cross validation. As the system turns are more structured compared to the user turns, the task of classifying user turns is more challenging. For more information, see Martínez-Hinarejos et al. (2008).

Here we compare the performance of MP against that of SQ-Loss-I and a HMM-based DA tagging system described in Martínez-Hinarejos et al. (2008). We extracted two sets of features from the text: (a) bigram TFIDF and (b) trigram TFIDF (Salton and Buckley, 1987). We constructed symmetrized k-NN graphs using each of the above features making use of cosine similarity. The graphs were defined over the training, test and development sets for the task that involved classifying user turns. The hyper-parameters were tuned over $k \in \{2, 10, 20, 50, 100\}$, $\mu \in \{$1e–8, 1e–4, 0.01, 0.1, 1, 10, 100$\}$ and $\nu \in \{$1e–8, 1e–6, 1e–4, 0.01, 0.1$\}$ on the development set. In the case of MP, we found that setting $\alpha = 2$ gave p = q at convergence.

The DA tagging results averaged over the 5-folds for the Dihana corpus are shown in Table 9. Unlike previous experiments, in this case, we treat the entire training set as being labeled, whereas

	Bigram	Trigram
SQ-Loss-I	79.1%	81.3%
MP	83.2%	85.6%

Table 10: Dialog-Act Tagging Accuracy results on the Switchboard DA Corpus. The baseline DA accuracy was 84.2% (Ji and Bilmes, 2005)

the test set is unlabeled. This simulates the case when SSL algorithms are used for supervised learning but in a transductive manner (i.e., the test set is assumed to be given). The HMM-based DA tagger which was trained on the same set gave an accuracy of 76.4%. It can be seen from Table 9 that MP outperforms both SQ-Loss-I and the HMM based tagger in both the bigram-TFIDF and trigram-TFIDF cases. We conjecture that the performance improvement of MP over HMM is due to two reasons: (a) MP is a discriminative model while the HMM was trained in a generative fashion, (b) as MP is transductive, it is able to exploit the knowledge of the graph over the test set.

7.5.2 SWITCHBOARD DA TAGGING

The goal of the Switchboard discourse language modeling project was to annotate the utterances in the Switchboard-I (SWB) training set with their corresponding discourse acts (Jurafsky and Ess-Dykema, 1997). SWB is a collection of telephone conversations (see Section 8.1). Every utterance in a each conversation was given one of the 42 different dialog act tags (see Table 2 in Jurafsky and Ess-Dykema, 1997). For our work here we only use the 11 most frequent tags. This covers more than 86% of all the utterances in SWB. These utterances were split into training, development and test sets containing 180314, 5192 and 4832 utterances respectively.

As in the case of Dihana, we generated both bigram and trigram TFIDF features and constructed graphs in the manner described above. Here we compare the performance of MP and SQ-Loss-I against the performance of a parametric dynamic Bayesian Network (DBN) that makes use of a hidden back-off model (Ji and Bilmes, 2005). The DBN, however, made use of only bigram features. The hyper-parameters were tuned over $k \in \{2, 10, 20\}$, $\mu \in \{$1e–8, 1e–4, 0.01, 0.1, 1, 10, 100$\}$ and $\nu \in \{$1e–8, 1e–6, 1e–4, 0.01, 0.1$\}$ on the development set. In the case of MP, we found that setting $\alpha = 2$ ensured that $p = q$ at convergence.

The test set DA tagging accuracy is shown in Table 10. We see that when we use trigram TFIDF features, MP outperforms the bigram DBN. More importantly, it performs better than SQ-Loss-I in all cases.

8. Parallelism and Scalability to Large Data Sets

In this section we discuss how MP can be scaled to very large data sets. We use the Switchboard I (SWB) data set which is a collection of about 2,400 two-sided telephone conversations among 543 speakers (302 male, 241 female) from all areas of the United States (Godfrey et al., 1992). A computer-driven system handled the calls, giving the caller appropriate recorded prompts, selecting and dialing another person (the callee) to take part in a conversation, introducing a topic for discussion and recording the speech from the two subjects into separate channels until the conversation was finished. SWB is very popular in the speech recognition community and is used almost ubiq-

uitously for the training of large vocabulary conversational speech recognition systems (Evermann et al., 2005; Subramanya et al., 2007) and consists of about 300 hours of speech data.

In order to construct a graph using the SWB data, we exact features $\mathbf{x}_i$ in the following manner—the wave files were first segmented and then windowed using a Hamming window of size 25ms at 100Hz. We then extracted 13 perceptual linear prediction (PLP) coefficients from these windowed features and appended both deltas and double-deltas resulting in a 39 dimensional feature vector. As phone classification performance is improved by context, we used a 7 frame context window (3 frames in the past and 3 in the future) yielding a 273 dimensional $\mathbf{x}_i$. This procedure resulted in 120 million samples.

Due to the large size $m = 120M$ of the SWB data set, it is not currently feasible to generate the graph using the conventional brute-force search which is $O(m^2)$. Nearest neighbor search is a well researched problem with many approximate solutions. A large number of solutions to this problem are based on variations of the classic *kd-tree* data structure (Friedman et al., 1977). Here we make use of the Approximate Nearest Neighbor (ANN) library (see `http://www.cs.umd.edu/~mount/ANN/`) (Arya and Mount, 1993; Arya et al., 1998). It constructs a modified version of the kd-tree data structure which is then used to query the NNs. The query process requires that one specify an error term, ε, and guarantees that

$$\frac{d(\mathbf{x}_i, \mathcal{N}(\mathbf{x}_i))}{d(\mathbf{x}_i, \mathcal{N}_\varepsilon(\mathbf{x}_i))} \leq 1+\varepsilon$$

where $\mathcal{N}(\mathbf{x}_i)$ is a function that returns the actual NN of $\mathbf{x}_i$ while $\mathcal{N}_\varepsilon(\mathbf{x}_i)$ returns the approximate NN. Larger values of ε improve the speed of the nearest neighbor search at the cost of accuracy. For more details about the algorithm, see Arya and Mount (1993); Arya et al. (1998). In our case we constructed a symmetrized 10-NN graph with $\varepsilon = 2.0$.

Next we describe how MP can be parallelized on a shared-memory symmetric multiprocessor (SMP). The update equations in the case of MP are amenable to a parallel implementation and also to further optimizations that lead to a near linear speedup. In the MP update equations (see Section 3), we see that one set of measures is held fixed while the other set is updated without any required communication amongst set members, so there is no write contention. This immediately yields a T-threaded implementation where the graph is evenly T-partitioned and each thread operates over only a size $m/T = (l+u)/T$ subset of the graph nodes.

We used the graph constructed using the SWB data above and ran a timing test on a 16 core symmetric multiprocessor with 128GB of RAM, each core operating at 1.6GHz. We varied the number T of threads from 1 (single-threaded) up to 16, in each case running 3 iterations of MP (i.e., 3 each of p and q updates). Each experiment was repeated 10 times, and we measured the minimum CPU time over these 10 runs. CPU time does not include the time taken to load data-structures from disk. The speedup for T threads is typically defined as the ratio of time taken for single thread to time taken for T threads. The solid (black) line in Figure 9(a) represents the ideal case (a linear speedup), that is, when using T threads results in a speedup of T. The pointed (green) line shows the actual speedup of the above procedure, typically less than ideal due to inter-process communication and poor shared L1 and/or L2 microprocessor cache interaction. When $T \leq 4$, the speedup (green) is close to ideal, but for increasing T the algorithm increasingly falls away from the ideal case. Note that in the figure (and henceforth) we refer to the green pointed line as 'speech temporal ordering' as the nodes in the graph are ordered based on the sequence in which they occur in the utterance.

Our contention is that the sub-linear speedup is due to the poor cache cognizance of the algorithm. At a given point in time, suppose thread $t \in \{1,\ldots,T\}$ is operating on node i_t. The collective set of neighbors that are being used by these T threads are $\{\cup_{t=1}^{T}\mathcal{N}(i_t)\}$ and this, along with nodes $\cup_{t=1}^{T}\{i_t\}$ (and all memory for the associated measures), constitute the current *working set*. The working set should be made as small as possible to increase the chance it will fit in any shared machine caches, but this becomes decreasingly likely as T increases since the working set is monotonically increasing with T. Our goal, therefore, is for the nodes that are being simultaneously operated on to have a large amount of neighbor overlap thus minimizing the working set size. Viewed as the optimization problem, we must find a partition $(V_1,V_2,\ldots,V_{m/T})$ of V that minimizes $\max_{j\in\{1,\ldots,m/T\}}|\cup_{v\in V_j}\mathcal{N}(v)|$. With such a partition, we may also order the subsets so that the neighbors of V_i would have maximal overlap with the neighbors of V_{i+1}. We then schedule the T nodes in V_j to run simultaneously, and schedule the V_j sets successively.

Algorithm 1: Graph Node Ordering Algorithm Pseudocode, SMP Case

Input: A Graph $G=(V,E)$
Result: A node ordering, by when they are marked.
Select an arbitrary node v ;
while *There are unselected nodes remaining* **do**
  Select an unselected $v' \in \mathcal{N}^2(v)$ that maximizes $|\mathcal{N}(v)\cap\mathcal{N}(v')|$. If the intersection is empty, select an arbitrary unselected v'. ;
  Mark v' as selected.; `// v' is next node in the order`
  $v \leftarrow v'$. ;

Of course, the time to produce such a partition cannot dominate the time to run the algorithm itself. Therefore, we propose a simple fast node ordering procedure (Algorithm 1) that can be run once before the parallelization begins. The algorithm orders the nodes such that successive nodes are likely to have a high amount of neighbor overlap with each other and, by transitivity, with nearby nodes in the ordering. It does this by, given a node v, choosing another node v' (from amongst v's neighbors' neighbors, meaning the neighbors of v's neighbors) that has the highest neighbor overlap. We need not search all V nodes for this, since anything other than v's neighbors' neighbors has no overlap with the neighbors of v. Given such an ordering, the t^{th} thread operates on nodes $\{t,t+m/T,t+2m/T,\ldots\}$. If the threads proceed synchronously (which we do not enforce) the set of nodes being processed at any time instant are $\{1+jm/T,2+jm/T,\ldots,T+jm/T\}$. This assignment is beneficial not only for maximizing the set of neighbors being simultaneously used, but also for successive chunks of T nodes since once a chunk of T nodes have been processed, it is likely that many of the neighbors of the next chunk of T nodes will already have been pre-fetched into the caches. With the graph represented as an adjacency list, and sets of neighbor indices sorted, our algorithm is $O(mk^3)$ in time and linear in memory since the intersection between two sorted lists may be computed in $O(k)$ time. This is typically even better than $O(m\log m)$ since $k^3<\log m$ for large m.

We ordered the SWB graph nodes, and ran timing tests as explained above. The CPU time required for the node ordering step is included in each run along with the time for MP. The results are shown in Figure 9(a) (pointed red line) where the results are much closer to ideal, and there are no obvious diminishing returns like in the unordered case. Running times are given in Figure 9(b).

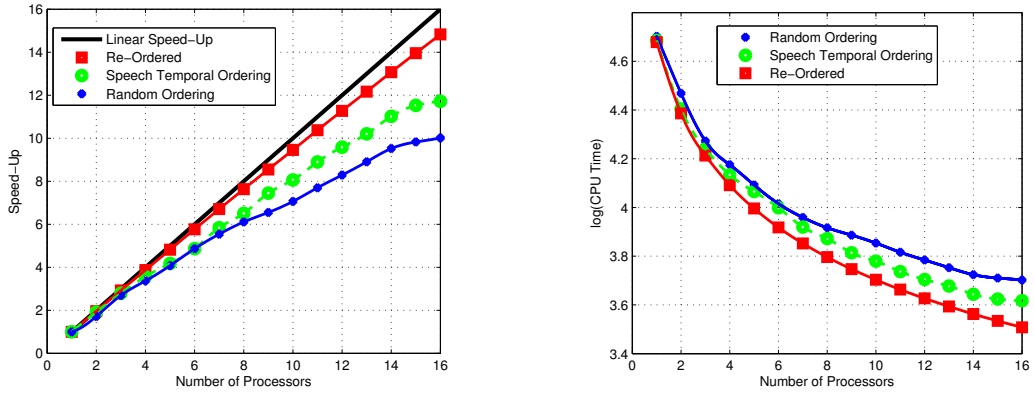

Figure 9: (a) speedup vs. number of threads for the SWB graph (see Section 7). The process was run on a 128GB, 16 core machine with each core at 1.6GHz. (b) The actual CPU times in seconds on a *log scale* vs. number of threads for with and without ordering cases. "Random" corresponds to the case where we choose a random unselected node rather than the one with maximum overlap (see Algorithm 1).

Moreover, the ordered case showed better performance even for a single thread $T = 1$. Note that since we made use of speech data to generate the graph, it is already naturally well-ordered by time. This is because human speech is a slowly changing signal, so the nodes corresponding to consecutive frames are similar, and can be expected to have similar neighbors. Therefore, we expect our "baseline" speech graph to be better than an arbitrary order, one that might be encountered in a different application domain. In order to measure performance for such arbitrarily ordered graphs, we took the original graph and reordered uniformly at random (a uniform node shuffle). We ran timing experiments on the resulting graph and the results are shown in Figure 9 as "Random". As can be seen, there is indeed a benefit from the speech order, and relative to this random baseline, our node ordering heuristic improves machine efficiency quite significantly.

We conclude this section by noting that (a) re-ordering may be considered a pre-processing (offline) step, (b) the SQ-Loss algorithm may also be implemented in a multi-threaded manner and this is supported by our implementation, (c) our re-ordering algorithm is general and fast and can be used for any graph-based algorithm where the iterative updates for a given node are a function of its neighbors (i.e., the updates are harmonic w.r.t. the graph Zhu et al., 2003), and (d) while the focus here was on parallelization across different processors on a SMP, a similar approach also applies for distributed processing across a network with a shared disk (Bilmes and Subramanya, 2011).

8.1 Switchboard Phonetic Annotation

In this section we consider how MP can be used to annotate the SWB data set. Recall that SWB consists of 300 hours of speech with word-level transcriptions. In addition, less reliable phone level annotations generated in an automatic manner by a speech recognizer with a non-zero error rate are also available (Deshmukh et al., 1998). The *Switchboard Transcription Project* (STP) (Greenberg, 1995) was undertaken to accurately annotate SWB at the phonetic and syllable levels. One of the goals was that such data could then be used to improve the performance of conversational speech

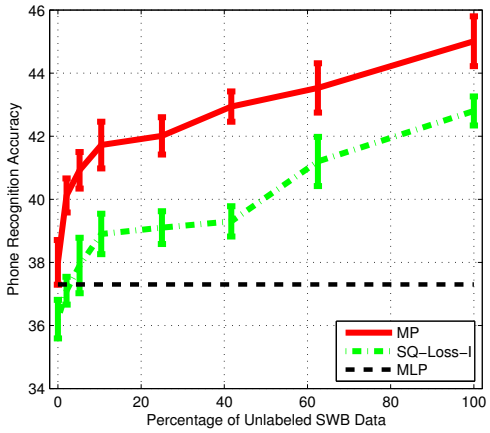

Figure 10: Phone Accuracy vs. Percentage of switchboard (SWB) I training data. STP portion of SWB was excluded. Phone Accuracy was measured on the STP data. Note that when all the Switchboard I data was added, the resulting graph had **120 million** vertices. The dashed black line shows the performance of a MLP measured using the $s = 0\%$ case over the same training, development and test sets as MP and LP.

recognition systems. As the task was time-consuming, costly, and error-prone, only 75 minutes of speech segments selected from different SWB conversations were annotated at the phone level and about 150 minutes annotated at the syllable level. Having access to such annotations for all of SWB could be useful for large vocabulary speech recognition research and speech science research in general. Thus, this an ideal real-world task for SSL.

For our experiments here we only make use of the phonetic labels ignoring the syllable annotations. Our goal here is two-fold: (a) treat the phonetically annotated portion of STP as labeled data and use it to annotate all of SWB in STP style, that is, at the phonetic level, yielding the S3TP corpus and (b) show that our approach scales to very large data sets.

We randomly split the 75 minute phonetically annotated part of STP into three sets, one each for training, development and testing containing 70%, 10% and 20% of the data respectively (the size of the development set is considerably smaller than the size of the training set). This procedure was repeated 10 times (i.e., we generated 10 different training, development and test sets by random sampling). In each case, we trained a phone classifier using the training set, tuned the hyper-parameters on the development set and evaluated the performance on the test set. In the following, we refer to SWB that is not a part of STP as *SWB-STP*. We added the unlabeled SWB-STP data in stages. The percentage, s, of unlabeled data included, 0%, 2%, 5%, 10%, 25%, 40%, 60%, and 100% of SWB-STP. We ran both MP and SQ-Loss-I in each case. When s =100%, there were about 120 million nodes in the graph. As far as we know, this is by far the largest (by about two orders of magnitude) size graph ever reported for an SSL procedure.

We constructed graphs using the STP data and s% of (unlabeled) SWB-STP data following the recipe described in the previous section. For all the experiments here we used a symmetrized 10-NN graph and $\varepsilon = 2.0$. The labeled and unlabeled points in the graph changed based on training, development and test sets used. In each case, we ran both the MP and SQ-Loss-I objectives. For each

set, we ran a search over $\mu \in \{1e{-}8, 1e{-}4, 0.01, 0.1\}$ and $\nu \in \{1e{-}8, 1e{-}6, 1e{-}4, 0.01, 0.1\}$ for both the approaches. The best value of the hyper-parameters were chosen based on the performance on the development set and the same value was used to measure the accuracy on the test set. The mean phone accuracy over the different test sets (and the standard deviations) are shown in Figure 10 for the different values of s. We would like to point out that our results at s =0% outperform the state-of-the-art. As a reference, at s =0%, an ℓ_2 regularized MLP with a 9 frame context window gave a mean phone accuracy of 37.2% and standard deviation of 0.83 (note that this MLP was trained fully-supervised). Phone classification in the case of conversational speech is a much harder task compared to phone classification of read speech (Morgan, 2009). It can be seen that MP outperforms SQ-Loss-I in all cases. More importantly, we see that the performance on the STP data improves with the addition of increasing amounts of unlabeled data, and MP seems to get a better benefit with this additional unlabeled data, although even SQ-Loss-I has not reached the point where unlabeled data starts becoming harmful (Nadler et al., 2010).

9. Discussion

In this section, we discuss possible extensions of the proposed approach.

9.1 Generalizing Graph-based Learning via Bregman Divergence

Given a strictly convex real-valued function $\phi : \triangle \rightarrow \mathbb{R}$, the Bregman divergence $\mathbf{B}_\phi(\psi_1||\psi_2)$ between two measures $\psi_1, \psi_2 \in \triangle$ is given by Lafferty et al. (1997)

$$\mathbf{B}_\phi(\psi_1||\psi_2) \triangleq \phi(\psi_1) - \phi(\psi_2) - \langle \nabla\phi(\psi_2), \psi_1 - \psi_2 \rangle.$$

It can be shown that a number of popular distance measures, such as Euclidean distance, KLD, Itakura-Satio distance are special cases of Bregman divergence (Banerjee et al., 2005). Consider the optimization problem $\mathcal{P}_{BR} : \min_{\mathrm{p} \in \triangle^m} \mathcal{C}_{BR}(\mathrm{p})$ where

$$\mathcal{C}_{BR}(\mathrm{p}) = \sum_{i=1}^{l} \mathbf{B}_\phi\big(r_i||p_i\big) + \mu \sum_{i=1}^{m} \sum_{j \in \mathcal{N}(i)} w_{ij} \mathbf{B}_\phi\big(p_i||p_j\big) + \nu \sum_{i=1}^{m} \mathbf{B}_\phi(p_i||u).$$

When $\mathbf{B}_\phi(p||q)$ is convex in the pair (p,q) (Banerjee et al., 2005), $\mathcal{C}_{BR}$ is also convex. Clearly $\mathcal{C}_{BR}$ is a valid graph-based learning objective and it can be seen that it generalizes objectives based on both squared loss ($\phi = \sum_y p^2(y)$) and KLD based loss ($\phi = \sum_y p(y) \log p(y)$). While in the case graph Laplacian-based techniques, one can generate a large family of regularizers by iterating the Laplacian or taking various transformations of its spectrum to create new ways of measuring smoothness on the graph, here in the Bregman case, the same can be achieved by using different ϕ's.

The graph regularizer is central to any graph-based SSL algorithm, and there are two factors that effect this regularizer: (a) the graph weights and (b) the loss function used to measure the disparity between the distributions. In the cases we have discussed thus far, the loss function has been either based on squared-error or KLD. Further, while there have been efforts in the past to learn the graph (and thus the graph weights) (Zhu and Ghahramani, 2002a; Zhang and Lee, 2006; Zhu et al., 2005; Alexandrescu and Kirchhoff, 2007a), to the best of our knowledge, there has been no efforts directed towards learning the loss function. So the natural question is whether it is possible to learn ϕ jointly with p?

One simple idea would be to set $\phi = \sum_y \left(\lambda p^2(y) + (1-\lambda) p(y) \log p(y)\right)$ which leads to a combination of the popular squared loss and proposed KLD based loss objectives (henceforth we refer to this as $\mathcal{C}'_{BR}(\mathrm{p}, \lambda)$). We then need to learn λ jointly with p. However, directly minimizing $\mathcal{C}'_{BR}$ w.r.t. both p and λ will always leads to $\lambda^* = 1$ as KLD is lower bounded by squared loss (by Pinsker's inequality). Thus other criteria such as those based on minimizing the leave-one-out error (Zhang and Lee, 2006) or minimum description length may be required. There might also be other convex parametrization of ϕ. This would amount to learning the loss function while the actual graph weights are held fixed.

While we have defined Bregman divergence over simplices, they are actually quite general and can be defined over other general sets of objects such as vectors or matrices (Tsuda et al., 2005). This can be used to solve general learning problems using alternating-minimization using a reformulation similar to the one suggested in Section 4. We believe that this is another contribution of our work here as our proposed objective, and the use of alternating-minimization to efficiently optimize it are in fact very general and can be used to solve other learning problems (Tsuda et al., 2005).

9.2 Incorporating Priors

As discussed in Section 1, there are two types of priors in SSL—label priors and balance priors. They are useful in the case of imbalanced data sets. We have seen that MP is less sensitive to imbalance compared to other graph-based SSL approaches (see the results in the cases of two-moon, USPS, Reuters, TIMIT and SWB data sets). However, in cases of extreme imbalance, even the performance of MP might suffer and so we show how to modify our proposed objective to handle both the above priors in a principled manner. Label priors are useful when the underlying data set is imbalanced. For example, in the case of phone classification, as a result of the nature of human speech and language production, some classes of sounds tend to occur at a higher rate compared to others. Clearly ignoring such domain knowledge can hurt performance particularly in the case of SSL where labeled data is sparse. On the other hand, balance priors are useful to prevent degenerate solutions. An extreme example of a degenerate solution would be all unlabeled samples being classified as belonging to the same class when the underlying data set has a uniform prior. This can occur due to a number of reasons such as, (a) improper graph construction, (b) improperly sampled labeled data, that is, the case where a majority of the labeled samples come from one class (similar to the scenario discussed in the case of the 2D two-moon data set).

Label Priors: This is more akin to the classical integration of priors within a Bayesian learning setting. There has been some work in the past directed towards integrating priors for parametric (non-graph-based) SSL (Mann and McCallum, 2007). In the case of graph-based SSL, class mass normalization (CMN) (Zhu and Ghahramani, 2002a; Bengio et al., 2007) and label bidding (Zhu and Ghahramani, 2002a), are the two approaches that have been used to-date. However, these are applicable only after the inference process has converged. In other words, they represent ways in which the posteriors may be influenced so that the average probability mass over all the posteriors for a given class matches that given by the prior. Ideally, like in general Bayesian learning, it is imperative that the priors are tightly integrated in to the inference process rather than influencing the results at a later point. Our proposed objective can be extended to incorporate label priors. We first remind the reader that $\mathcal{C}_{KL}(\mathrm{p})$ may be re-written as $\mathcal{C}_{KL}(\mathrm{p}) = \sum_{i=1}^{l} D_{KL}\left(r_i || p_i\right) + \mu \sum_{i,j} w_{ij} D_{KL}\left(p_i || p_j\right) + \nu \sum_i D_{KL}\left(p_i || u\right)$ where u is uniform measure.

Now consider minimizing over $\mathrm{p} \in \triangle^m$

$$C'_{KL}(\mathrm{p}) = \sum_{i=1}^{l} D_{KL}\big(r_i||p_i\big) + \mu \sum_{i,j} w_{ij} D_{KL}\big(p_i||p_j\big) + \nu \sum_i D_{KL}\big(p_i||p_0\big).$$

The above objective is convex and the last term encourages each p_i to be close to p_0 without actually insisting that $p_i(y) = p_0(y)\ \forall\ i,y$. It is possible to reformulate the above objective as

$$C'_{MP}(\mathrm{p},\mathrm{q}) = \sum_{i=1}^{l} D_{KL}\big(r_i||q_i\big) + \mu \sum_{i,j} w'_{ij} D_{KL}\big(p_i||q_j\big) + \nu \sum_i D_{KL}\big(p_i||p_0\big).$$

which can be easily solved using AM. Further each of the update equations has a closed form solution. This represents the case where the prior effects each vertex directly (i.e., a more local influence) .

Balance Priors: There has been some work in graph-based SSL for incorporating balance priors. SGT (Joachims, 2003) which is an approximation to the NP-hard norm cut problem attempts to incorporate priors by influencing the nature of the final cut. But there are other drawbacks associated with SGT such as computational complexity. We can incorporate a balance term in our objective by first defining $\tilde{p}(y)$ as the agglomerative measure over all the p's and then minimizing

$$C'_1(\mathrm{p}) = C_{KL}(\mathrm{p}) + \kappa D_{KL}(p_0||\tilde{p})$$

where $p_0(\mathrm{y})$ is the prior probability that $\mathrm{Y} = \mathrm{y}$. The above retains the nice convexity properties of the original objective. There are many ways of defining $\tilde{p}$, such as,

$$\tilde{p}(y) = \frac{1}{n}\sum_{i=1}^{n} p_i(y) \text{ or } \tilde{p}(y) \propto \prod_{i=1}^{n} (p_i(y) + \varepsilon).$$

The first case above represents the arithmetic mean while the second one is the geometric mean. Here the prior only indirectly influences the individual p's, that is, via $\tilde{p}$. Unfortunately, this form cannot be optimized in the closed form using alternating-minimization. However, the MOM approach proposed in Section 3.1 or IPMs or any other numerical convex optimization approach may be used to solve the above problem.

9.3 Directed Graphs

In some applications, the graphs are directed in nature. Examples include the Internet (a vertex might represent a web-page and directed links for hyper-links between pages), or a graph representing the routes taken by a delivery system. In such applications there is useful information that is expressed by the direction of the connection between two vertices. While we could convert any given directed graph into an undirected one, SSL algorithms in this case should exploit the information in the directed links. Thus far we have been using symmetrized k-NN graphs, but without the symmetrization step, k-NN graphs are not necessarily symmetric.

As KLD is an asymmetric measure of dissimilarity between measures, our proposed objective can very easily be extended to work for directed graphs. Note that, as in the case of an undirected graph, a directed graph can also be represented as a matrix $\mathbf{W}$, but here the matrix is asymmetric. There has been some work on graph-based SSL using directed graphs. For example, Zhou et al.

(2005), use a squared-loss based objective on directed graphs. We believe that this may not be ideal, as squared error is symmetric and as a result it might be difficult to fully exploit the information encoded by the directed links. An asymmetric measure of dissimilarity would have a better chance of correctly representing the problem of SSL on directed graphs.

It turns our that $\mathcal{C}_{MP}$ may be modified for directed graphs. We assume that if j is a NN of i then there is a directed arrow from i to j. There are in fact two scenarios that one needs to consider. Given a node $i \in V$, let $\mathcal{N}^{(in)}(i)$ be the set of nodes that have directed edges that lead into vertex i. Consider the following objective

$$\mathcal{C}_{MP}^{(D1)}(\mathrm{p},\mathrm{q}) = \sum_{i=1}^{l} D_{KL}\big(r_i||p_i\big) + \mu \sum_{i=1}^{m} \sum_{j \in \mathcal{N}^{(in)}(i)} w_{ij} D_{KL}\big(p_i||q_j\big) - \nu \sum_{i=1}^{m} H(p_i).$$

In this case, for node i, the second term in the above objective encourages p_i to be close to the q's of all its neighbors, $\mathcal{N}^{(in)}(i)$. In other words, the above form expresses the rule "each vertex should resemble its neighbors but not necessarily vice-versa." In a similar manner we can define a complementary form—let $\mathcal{N}^{(out)}(i)$ be the set of nodes which are on the other end of out-going links from node $i \in V$. Consider minimizing

$$\mathcal{C}_{MP}^{(D2)}(\mathrm{p},\mathrm{q}) = \sum_{i=1}^{l} D_{KL}\big(r_i||p_i\big) + \mu \sum_{i=1}^{m} \sum_{j \in \mathcal{N}^{(out)}(i)} w_{ij} D_{KL}\big(p_i||q_j\big) - \nu \sum_{i=1}^{m} H(p_i).$$

This form encourages, "the neighbors of a vertex should resemble it but not necessarily vice-versa."

Both $\mathcal{C}_{MP}^{(D1)}(\mathrm{p},\mathrm{q})$ and $\mathcal{C}_{MP}^{(D2)}(\mathrm{p},\mathrm{q})$ can be efficiently optimized using our alternating-minimization (the update equations are similar to MP). In a similar manner as the above, our objective can also be easily extended to hyper-graphs.

9.4 Connections to Entropy Minimization (Grandvalet and Bengio, 2005)

Entropy Minimization uses the entropy of the unlabeled data as a regularizer while optimizing a parametric loss function over the labeled data. The loss function here is given by

$$C(\Theta) = -\sum_{i=1}^{l} \log p(y_i|x_i;\Theta) + \nu \sum_{i=l+1}^{l+u} H(Y_i|X_i;\Theta)$$

where $H(Y_i|X_i;\Theta)$ is the Shannon entropy of the probability distribution $p(y_i|x_i;\Theta)$. While both our proposed approach and entropy minimization make use of the Shannon entropy as a regularizer, there are several important differences between the two approaches:

1. entropy minimization is not graph-based,
2. entropy minimization is parametric whereas our proposed approach is non-parametric
3. the objective in case of entropy minimization is not convex, whereas in our case we have a convex formulation with simple update equations and convergence guarantees.
4. most importantly, entropy minimization attempts to minimize entropy while the proposed approach aims to maximize entropy. While this may seem a triviality, it has significant consequences on the optimization problem.

It is however possible to derive an interesting relationship between the proposed objective and entropy minimization. Consider

$$\begin{aligned} C_{KL}(\mathrm{p}) &= \sum_{i=1}^{l} D_{KL}\big(r_i||p_i\big)\big) + \mu \sum_{i,j=1}^{n} w_{ij} D_{KL}\big(p_i||p_j\big) - \nu \sum_{i=1}^{n} H(p_i) \\ &\leq \sum_{i=1}^{l} D_{KL}\big(r_i||p_i\big)\big) - \mu \sum_{i,j=1}^{n} w_{ij} \sum_{y} p_i(y) \log p_j(y) \end{aligned}$$

as $w_{ij}, \nu, H(p_i) \geq 0$. Consider a degenerate graph in which $w_{ij} = \delta(i = j \wedge i > l)$ then

$$\begin{aligned} C_{KL}(\mathrm{p}) &\leq \sum_{i=1}^{l} D_{KL}\big(r_i||p_i\big)\big) - \mu \sum_{i=l+1}^{n} \sum_{y} p_i(y) \log p_i(y) \\ &= \sum_{i=1}^{l} \sum_{y} \Big(r_i(y) \log r_i(y) - r_i(y) \log p_i(y) \Big) + \mu \sum_{i=l+1}^{n} H(p_i) \\ &\leq -\sum_{i=1}^{l} \sum_{y} r_i(y) \log p_i(y) + \mu \sum_{i=l+1}^{n} H(p_i). \end{aligned}$$

Setting $w_{ij} = \delta(i = j \wedge i > l)$ amounts to not using a graph regularizer. If we assume hard labels (i.e., $H(r_i) = 0$) and that each p_i is parameterized by, say θ_i, then we can rewrite the above as

$$C_{KL}(\mathrm{p}) \leq -\sum_{i=1}^{l} \log p_i(y_i; \theta_i) + \mu \sum_{i=l+1}^{n} H(p_i; \theta_i).$$

Now if all the θ_i were tied to a single θ then we have that

$$C_{KL}(\mathrm{p}) \leq -\sum_{i=1}^{l} \log p_i(y_i; \theta) + \mu \sum_{i=l+1}^{n} H(p_i; \theta)$$

which is equal to the entropy minimization objective. Thus entropy minimization minimizes a non-convex upper bound on a special case of our proposed loss function. This is perhaps one of the reasons why graph-based approaches outperform entropy minimization on manifold-like data sets (see chapter 21 in Chapelle et al., 2007).

9.5 Rate of Convergence of MP

Recall that in Section 5 we showed that the rate of convergence of SQ-Loss-I is geometric (linear). Here we empirically compare the rate of convergence of MP and SQ-Loss-I. While we have so far been unable to derive theoretical bounds on the convergence rate of MP, our empirical analysis shows that MP convergences faster than SQ-Loss-I. The difficulties associated with analyzing the rate of convergence of MP are mostly due to the non-linear nature of the update equation for $p_i^{(n)}(y)$.

We ran both MP and SQ-Loss-I to convergence on a number of data sets taken from a variety of domains (see Table 3). For both algorithms we measured

$$f^{(n)} = \frac{C^{(n)} - C^*}{C^{(n-1)} - C^*} \tag{2}$$

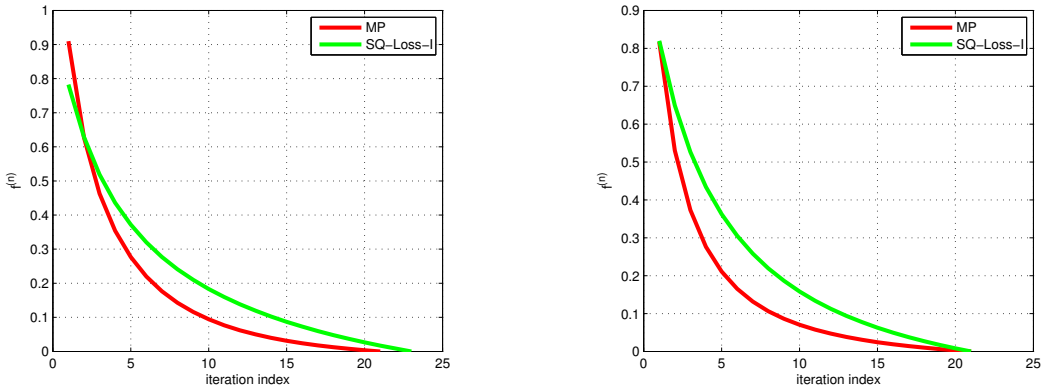

Figure 11: Plots showing the rate of convergence of MP and SQ-Loss-I in the case of the Text and USPS corpora. The x-axis represents iteration index and the y-axis rate of convergence, $f^{(n)}$ (see Equation 2).

and the plots of these quantities are shown in Figure 11 (similar trends were observed in the case of other data sets). In the above, C is the appropriate objective (i.e., $\mathcal{C}_{MP}$ in case of MP and $\mathcal{C}_{SQ}$ in the case of SQ-Loss-I) and C^* is the *corresponding* optimum value. While we have a standard test for convergence in the case of MP (see Theorem 9), for the purposes of comparison against SQ-Loss-I here, we use the following criteria: in either case we say that the algorithm has converged if the rate of change of the parameters falls below 0.5%. Figure 11 shows that MP converges faster in comparison to SQ-Loss-I. Based on these results, we make the following conjecture:

Conjecture 13 *MP has a geometric convergence rate, if not better.*

Finally a note on how to set α. Recall α is the hyperparameter that ensures that $p = q$ in the final solution in the case of MP. Recall that in theorem 8, we have shown that there exists a finite value of α such that $p^* = q^*$. In practice, we found that setting $\alpha = 2$ ensures the equality of p and q at convergence. As expected, we also found that increasing α leads to a slower rate of convergence in practice.

10. Conclusions

In this paper we presented a objective based on KLD for graph-based SSL. We have shown how the objective can be efficiently solved using alternating-minimization. In addition, we showed that the sequence of updates has a closed form solution and that it converges to the correct optima. We also derived a test for convergence of the iterative procedure that does not require the computation of the objective. A version of the squared-error graph-based SSL objective defined over measures was also presented. In this context we showed that squared-error has a geometric rate of convergence.

Our results show that MP is able to outperform other state-of-the-art graph-based SSL algorithms on a number of tasks from diverse set of domains ranging from speech to natural language to image processing. We have also shown how our algorithm can be scaled to very large data sets.

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.

Appendix A. Solving $\mathcal{P}_{KL}$ using Method of Multipliers

The first step in the application of MOM to solve $\mathcal{P}_{KL}$ is the construction of the augmented Lagrangian function for $\mathcal{C}_{KL}(\mathrm{p})$ which is given by

$$\mathcal{L}_{C_1}(\mathrm{p},\Lambda) = \mathcal{C}_{KL}(\mathrm{p}) + \sum_{i=1}^{n} \lambda_i \left(1 - \sum_y p_i(y)\right) + c \sum_{i=1}^{n} \left(1 - \sum_y p_i(y)\right)^2$$

where $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$ are the Lagrange multipliers and $c \geq 0$ is the penalty parameter. Recall that we require $\sum_y p_i(y) = 1, \ \forall\, i$ and that $p_i(y) \geq 0, \ \forall\, i, y$. Notice that the objective $\mathcal{L}_{C_1}(\mathrm{p},\Lambda)$ only penalizes deviations from the equality constraints. In order to ensure that the inequality constraints in $\mathcal{P}_{KL}$ are met we make use of the *gradient projection method* (Bertsekas, 1999). Thus the update equation is given by

$$p_i^{(n)}(y) = \left[p_i^{(n-1)}(y) - \alpha^{(n-1)} \left(\frac{\partial \mathcal{L}_{C_1}(\mathrm{p},\Lambda)}{\partial p_i(y)} \right)_{\{\mathrm{p}=\mathrm{p}^{(n-1)}, \Lambda=\Lambda^{(n-1)}\}} \right]^+ .$$

Here $n = 1, \ldots,$ is the iteration index, $\alpha^{(n-1)}$ is the learning rate, and $[x]^+ = \max(x, 0)$. Determining an appropriate learning rate is often one of the biggest challenges associated with the application of gradient descent based optimization approaches. We use the Armijo rule (Bertsekas, 1999) to compute the learning rate, α. It can be shown that

$$\frac{\partial \mathcal{L}_{C_1}(\mathrm{p},\Lambda)}{\partial p_i(y)} = \mu \sum_{j=1}^{n} \left[w_{ej} \big(1 + \log p_i(y) - \log p_j(y)\big) - \frac{w_{je} p_j(y)}{p_i(y)} \right] - \frac{r_i(y)}{p_i(y)} \delta(e \leq l) + \nu(\log p_i(y) + 1) + \lambda_i + 2c\big(1 - \sum_y p_i(y)\big).$$

Under MOM, the update equation for the Lagrange multipliers is

$$\lambda_i^{(n)} = \lambda_i^{(n-1)} + c^{(n-1)} \left(\sum_y p_i^{(n-1)}(y) - 1 \right)$$

and the penalty parameter is updated using

$$c^{(n)} = \begin{cases} \beta c^{(n-1)} \text{ if } \sum_i \left(\tau_i^{(n)} - \gamma \tau_i^{(n-1)} \right) > 0 \\ c^{(n-1)} \text{ otherwise} \end{cases}$$

where $\tau_i^{(n)} = \big(1 - \sum_y p_i^{(n)}(y)\big)^2$. Intuitively, the above update rule for the penalty parameter increases its value only if the constraint violation is not decreased by a factor γ over the previous iteration. The iterative procedure terminates when

$$\frac{\mathcal{L}_{C_1}(\mathrm{p}^{(n-1)}, \Lambda^{(n-1)}) - \mathcal{L}_{C_1}(\mathrm{p}^{(n)}, \Lambda^{(n)})}{\mathcal{L}_{C_1}(\mathrm{p}^{(n-1)}, \Lambda^{(n-1)})} \leq \zeta.$$

Appendix B. Proof of Convergence

In this section we show that AM on $\mathcal{C}_{MP}$ converges to the correct optimum. We first show that the three-and-four points properties (to be defined shortly) hold for $\mathcal{C}_{MP}$ which then implies that the five-points property holds for $\mathcal{C}_{MP}$. We note that our proof is inspired by Csiszar and Tusnady (1984).

Definition 14 *If $\mathcal{P}$, $\mathcal{Q}$ are convex sets of finite measures, given a divergence $d(p,q)$, $p \in \mathcal{P}$, $q \in \mathcal{Q}$, then the "three points property" (3-pp) is said to hold for $p \in \mathcal{P}$ if $\forall\, q, q^{(0)} \in \mathcal{Q}$ we have*

$$\delta(p,p^{(1)}) + d(p^{(1)},q^{(0)}) \leq d(p,q^{(0)}) \text{ where } p^{(1)} \in \operatorname*{argmin}_{p\in\mathcal{P}} d(p,q^{(0)}) \text{ and}$$

$\delta(p,p^{(1)}) : \mathcal{P}\times\mathcal{P} \to \mathbb{R}^+$ *is arbitrary and* $\delta(p,p) = 0$.

Lemma 15 $\mathcal{C}_{MP}(\mathrm{p},\mathrm{q})$ *satisfies the 3-pp.*

Proof Let

$$\delta(\mathrm{p},\mathrm{p}^{(1)}) \triangleq \mu \sum_{i,j=1}^{n} w'_{ij} D_{KL}(p_i||p_i^{(1)}), \quad f(t) \triangleq \mathcal{C}_{MP}(\mathrm{p}^{(t)},\mathrm{q}^{(0)})$$

where $\mathrm{p}^{(t)} = (1-t)\mathrm{p} + t\mathrm{p}^{(1)}$, $0 < t \leq 1$ and thus $p_i^{(t)} = (1-t)p_i + t p_i^{(1)}$. As $f(t)$ attains its minimum at $t = 1$, $f(1) \leq f(t)$, $\forall\, 0 < t \leq 1$ and so

$$\frac{f(1) - f(t)}{1-t} \leq 0. \tag{3}$$

We have that

$$f(t) = \sum_{i=1}^{l}\sum_{y\in\mathrm{Y}} r_i \log\frac{r_i}{q_i^{(0)}} + \mu \sum_{i,j=1}^{n} w'_{ij} \sum_{y\in\mathrm{Y}} p_i^{(t)} \log\frac{p_i^{(t)}}{q_j^{(0)}} + \nu \sum_{i=1}^{n}\sum_{y\in\mathrm{Y}} p_i^{(t)} \log\frac{p_i^{(t)}}{u}$$

where we ignore the argument y in every measure for brevity (e.g., r_i is $r_i(y)$). Using the above in Equation 3 and taking the limit as $t \to 1$, we get

$$\begin{aligned}
&\lim_{t\to 1}\left(\mu \sum_{i,j=1}^{n} w'_{ij} \sum_{y\in\mathrm{Y}} \frac{1}{1-t}\left(p_i^{(1)} \log\frac{p_i^{(1)}}{q_j^{(0)}} - p_i^{(t)} \log\frac{p_i^{(t)}}{q_j^{(0)}}\right) + \nu \sum_{i=1}^{n}\sum_{y\in\mathrm{Y}} \frac{1}{1-t}\left(p_i^{(1)} \log\frac{p_i^{(1)}}{u} - p_i^{(t)} \log\frac{p_i^{(t)}}{u}\right)\right) \\
&\overset{(a)}{=} \mu \sum_{i,j=1}^{n} w'_{ij} \sum_{y\in\mathrm{Y}} \lim_{t\to 1}\left[\frac{1}{1-t}\left(p_i^{(1)} \log\frac{p_i^{(1)}}{q_j^{(0)}} - p_i^{(t)} \log\frac{p_i^{(t)}}{q_j^{(0)}}\right)\right] \\
&\qquad + \nu \sum_{i=1}^{n}\sum_{y\in\mathrm{Y}} \lim_{t\to 1}\left[\frac{1}{1-t}\left(p_i^{(1)} \log\frac{p_i^{(1)}}{u} - p_i^{(t)} \log\frac{p_i^{(t)}}{u}\right)\right] \\
&\overset{(b)}{=} \mu \sum_{i,j=1}^{n} w'_{ij} \sum_{y\in\mathrm{Y}} \left[\frac{\partial}{\partial t}\left(p_i^{(t)} \log\frac{p_i^{(t)}}{q_j^{(0)}}\right)\right]_{t=1} + \nu \sum_{i=1}^{n}\sum_{y\in\mathrm{Y}} \left[\frac{\partial}{\partial t}\left(p_i^{(t)} \log\frac{p_i^{(t)}}{u}\right)\right]_{t=1}
\end{aligned}$$

where (a) follows as both $p_i^{(t)} \log \frac{p_i^{(t)}}{q_j^{(0)}}$ and $p_i^{(t)} \log \frac{p_i^{(t)}}{u}$ are convex in t, and thus the terms within the summations are difference quotients of convex functions which are non-increasing. As a result we can use the monotone convergence theorem (MCT) (see page 87, Theorem 6 in H.L.Royden, 1988) to exchange the limit with the summations. Finally (b) follows from the definition of the derivative. Note that (a) can also be explained via the dominated convergence theorem (DCT) (see page 84, proposition 6 in H.L.Royden, 1988). If $q_j^{(0)}(y) > 0, \; \forall \, y, j$ then there exists $\gamma < \infty$ such that $p_i^{(1)} \log \frac{p_i^{(1)}}{q_j^{(0)}} - p_i^{(t)} \log \frac{p_i^{(t)}}{q_j^{(0)}} < \gamma$ because the difference of two finite real numbers is always bounded above which implies that the DCT can be used to distribute the limits within the summations. Thus we have that

$$\begin{aligned} 0 &\geq \mu \sum_{i,j=1}^{n} w'_{ij} \sum_{y \in \mathrm{Y}} \left[\frac{\partial}{\partial t} \left(p_i^{(t)} \log \frac{p_i^{(t)}}{q_j^{(0)}} \right) \right]_{t=1} + \nu \sum_{i=1}^{n} \sum_{y \in \mathrm{Y}} \left[\frac{\partial}{\partial t} \left(p_i^{(t)} \log \frac{p_i^{(t)}}{u} \right) \right]_{t=1} \\ &= \mu \sum_{i,j=1}^{n} w'_{ij} \sum_{y \in \mathrm{Y}} \left(p_i^{(1)} \log \frac{p_i^{(1)}}{q_j^{(0)}} - p_i \log \frac{p_i^{(1)}}{q_j^{(0)}} \right) + \nu \sum_{i=1}^{n} \sum_{y \in \mathrm{Y}} \left(p_i^{(1)} \log \frac{p_i^{(1)}}{u} - p_i \log \frac{p_i^{(1)}}{u} \right). \end{aligned}$$

The last equation follows as $\sum_{y \in \mathrm{Y}} (p_i^{(1)} - p_i) = 0$. As a result we have that

$$\begin{aligned} 0 &\geq \mu \sum_{i,j=1}^{n} w'_{ij} \sum_{y \in \mathrm{Y}} \left(p_i^{(1)} \log \frac{p_i^{(1)}}{q_j^{(0)}} - p_i \log \frac{p_i^{(1)}}{q_j^{(0)}} \right) + \nu \sum_{i=1}^{n} \sum_{y \in \mathrm{Y}} \left(p_i^{(1)} \log \frac{p_i^{(1)}}{u} - p_i \log \frac{p_i^{(1)}}{u} \right) \\ &= \mu \sum_{i,j=1}^{n} w'_{ij} D_{KL}(p_i^{(1)} || q_j^{(0)}) + \nu \sum_{i=1}^{n} D_{KL}(p_i^{(1)} || u) - \left(\mu \sum_{i,j=1}^{n} w'_{ij} \sum_{y \in \mathrm{Y}} p_i \log \frac{p_i^{(1)}}{q_j^{(0)}} + \nu \sum_{i=1}^{n} \sum_{y \in \mathrm{Y}} p_i \log \frac{p_i^{(1)}}{u} \right) \end{aligned}$$

From the definition of $\mathcal{C}_{MP}(\mathrm{p}, \mathrm{q})$ we have that

$$\mu \sum_{i,j=1}^{n} w'_{ij} D_{KL}(p_i^{(1)} || q_j^{(0)}) + \nu \sum_{i=1}^{n} D_{KL}(p_i^{(1)} || u) = \mathcal{C}_{MP}(\mathrm{p}^{(1)}, \mathrm{q}^{(0)}) - \sum_{i=1}^{l} D_{KL}\big(r_i || q_i^{(0)}\big).$$

Using the above we get

$$0 \geq \mathcal{C}_{MP}(\mathrm{p}^{(1)}, \mathrm{q}^{(0)}) - \sum_{i=1}^{l} D_{KL}\big(r_i || q_i^{(0)}\big) - \left(\mu \sum_{i,j=1}^{n} w'_{ij} \sum_{y \in \mathrm{Y}} p_i \log \frac{p_i^{(1)}}{q_j^{(0)}} + \nu \sum_{i=1}^{n} \sum_{y \in \mathrm{Y}} p_i \log \frac{p_i^{(1)}}{u} \right). \quad (4)$$

Consider

$$\sum_{y \in \mathrm{Y}} p_i \log \frac{p_i^{(1)}}{q_j^{(0)}} = \sum_{y \in \mathrm{Y}} p_i \log \frac{p_i^{(1)}}{q_j^{(0)}} \frac{p_i}{p_i} = \sum_{y \in \mathrm{Y}} p_i \left(\log \frac{p_i}{q_j^{(0)}} + \log \frac{p_i^{(1)}}{p_i} \right) = D_{KL}(p_i || q_j^{(0)}) - D_{KL}(p_i || p_i^{(1)}).$$

Similarly

$$\sum_{y \in \mathrm{Y}} p_i \log \frac{p_i^{(1)}}{u} = \sum_{y \in \mathrm{Y}} p_i \log \frac{p_i^{(1)}}{u} \frac{p_i}{p_i} = \sum_{y \in \mathrm{Y}} p_i \left(\log \frac{p_i}{u} + \log \frac{p_i^{(1)}}{p_i} \right) = D_{KL}(p_i || u) - D_{KL}(p_i || p_i^{(1)}).$$

Using the above two equations in Equation 4 we have that

$$\begin{aligned}
0 &\geq \mathcal{C}_{MP}(\mathrm{p}^{(1)},\mathrm{q}^{(0)}) - \sum_{i=1}^{l} D_{KL}\big(r_i||q_i^{(0)}\big) - \mu \sum_{i,j=1}^{n} w'_{ij}\left(D_{KL}(p_i||q_j^{(0)}) - D_{KL}(p_i||p_i^{(1)})\right) \\
&\qquad - \nu \sum_{i=1}^{n}\left(D_{KL}(p_i||u) - D_{KL}(p_i||p_i^{(1)})\right) \\
&\overset{(a)}{\geq} \mathcal{C}_{MP}(\mathrm{p}^{(1)},\mathrm{q}^{(0)}) - \mathcal{C}_{MP}(\mathrm{p},\mathrm{q}^{(0)}) + \mu \sum_{i,j=1}^{n} w'_{ij} D_{KL}(p_i||p_i^{(1)}) \\
&= \mathcal{C}_{MP}(\mathrm{p}^{(1)},\mathrm{q}^{(0)}) - \mathcal{C}_{MP}(\mathrm{p},\mathrm{q}^{(0)}) + \delta(\mathrm{p},\mathrm{p}^{(1)})
\end{aligned}$$

where (a) follows as $\nu \geq 0$ and $D_{KL}(p_i||p_i^{(1)}) \geq 0$. ■

Thus we have show that 3-pp holds for $\mathcal{C}_{MP}$.

Definition 16 *If $\mathcal{P}$, $\mathcal{Q}$ are convex sets of finite measures, given a divergence $d(p,q)$, $p \in \mathcal{P}$, $q \in \mathcal{Q}$, then the "four points property" (4-pp) is said to hold for $q \in \mathcal{Q}$ if $\forall\ p, p^{(1)} \in \mathcal{P}$ we have*

$$d(p,q^{(1)}) \leq \delta(p,p^{(1)}) + d(p,q)$$

where $q^{(1)} \in \operatorname*{argmin}_{q\in\mathcal{Q}} d(p^{(1)},q)$ and $\delta(p,p^{(1)})$ should match the definition of $\delta(.,.)$ used in 3-pp.

Lemma 17 *$\mathcal{C}_{MP}(\mathrm{p},\mathrm{q})$ satisfies the 4-pp.*

Proof Let

$$g(t) \triangleq \mathcal{C}_{MP}(\mathrm{p}^{(1)},\mathrm{q}^{(t)})$$

where $\mathrm{q}^{(t)} = (1-t)\mathrm{q} + t\mathrm{q}^{(1)},\ 0 < t \leq 1$ and thus $q_i^{(t)} = (1-t)q_i + tq_i^{(1)}$ and $q^{(1)}$ is as defined above. Also recall that $\delta(\mathrm{p},\mathrm{p}^{(1)}) \triangleq \mu\Sigma_{i,j=1}^{n} w'_{ij} D_{KL}(p_i||p_i^{(1)})$. The proof for this lemma proceeds in a manner similar to the proof of lemma 15. It should be clear that $g(t)$ achieves its minimum at $t = 1$ and as a result we have that

$$\frac{g(1) - g(t)}{1-t} \leq 0 \tag{5}$$

and

$$g(t) = \sum_{i=1}^{l}\sum_{y\in\mathrm{Y}} r_i \log\frac{r_i}{q_i^{(t)}} + \mu \sum_{i,j=1}^{n} w'_{ij} \sum_{y\in\mathrm{Y}} p_i^{(1)} \log\frac{p_i^{(1)}}{q_j^{(t)}} + \nu \sum_{i=1}^{n}\sum_{y\in\mathrm{Y}} p_i^{(1)} \log\frac{p_i^{(1)}}{u}.$$

Using the above in Equation 5 and passing it to the limit we get

$$\lim_{t\to 1}\Bigg(\sum_{i=1}^{l}\sum_{y\in \mathrm{Y}}\frac{1}{1-t}\big(r_i\log\frac{r_i}{q_i^{(1)}}-r_i\log\frac{r_i}{q_i^{(t)}}\big)+\mu\sum_{i,j=1}^{n}w'_{ij}\sum_{y\in \mathrm{Y}}\frac{1}{1-t}\big(p_i^{(1)}\log\frac{p_i^{(1)}}{q_j^{(1)}}-p_i^{(1)}\log\frac{p_i^{(1)}}{q_j^{(t)}}\big)\Bigg)$$
$$\overset{(a)}{=}\sum_{i=1}^{l}\sum_{y\in \mathrm{Y}}\lim_{t\to 1}\left[\frac{1}{1-t}\big(r_i\log\frac{r_i}{q_i^{(1)}}-r_i\log\frac{r_i}{q_i^{(t)}}\big)\right]+\mu\sum_{i,j=1}^{n}w'_{ij}\sum_{y\in \mathrm{Y}}\lim_{t\to 1}\left[\frac{1}{1-t}\big(p_i^{(1)}\log\frac{p_i^{(1)}}{q_j^{(1)}}-p_i^{(1)}\log\frac{p_i^{(1)}}{q_j^{(t)}}\big)\right]$$
$$\overset{(b)}{=}\sum_{i=1}^{l}\sum_{y\in \mathrm{Y}}\left[\frac{\partial}{\partial t}\left(r_i\log\frac{r_i}{q_i^{(t)}}\right)\right]_{t=1}+\mu\sum_{i,j=1}^{n}w'_{ij}\sum_{y\in \mathrm{Y}}\left[\frac{\partial}{\partial t}\left(p_i^{(1)}\log\frac{p_i^{(1)}}{q_j^{(t)}}\right)\right]_{t=1}$$
$$=-\sum_{i=1}^{l}\sum_{y\in \mathrm{Y}}r_i+\sum_{i=1}^{l}\sum_{y\in \mathrm{Y}}\frac{r_i}{q_i^{(1)}}q_i-\mu\sum_{i,j=1}^{n}w'_{ij}\sum_{y\in \mathrm{Y}}p_i^{(1)}+\mu\sum_{i,j=1}^{n}w'_{ij}\sum_{y\in \mathrm{Y}}\frac{p_i^{(1)}}{q_j^{(1)}}q_j$$

where (a) once again follows from using DCT. This is because $\mathcal{C}_{MP}(\mathrm{p},\mathrm{q}^{(1)})$, $\mathcal{C}_{MP}(\mathrm{p},\mathrm{q})<\infty$ (else 4-pp trivially holds) and as $\mathcal{C}_{MP}(\mathrm{p},\mathrm{q})$ is the sum of all positive terms, it implies each term is finite and thus bounded above. Also (b) follows from the definition of the derivative. As a result we have that

$$\begin{aligned}0&\geq-\sum_{i=1}^{l}\sum_{y\in \mathrm{Y}}r_i+\sum_{i=1}^{l}\sum_{y\in \mathrm{Y}}\frac{r_i}{q_i^{(1)}}q_i-\mu\sum_{i,j=1}^{n}w'_{ij}\sum_{y\in \mathrm{Y}}p_i^{(1)}+\mu\sum_{i,j=1}^{n}w'_{ij}\sum_{y\in \mathrm{Y}}\frac{p_i^{(1)}}{q_j^{(1)}}q_j\\&=-l-\mu\sum_{i,j=1}^{n}w'_{ij}+\sum_{i=1}^{l}\sum_{y\in \mathrm{Y}}\frac{r_i}{q_i^{(1)}}q_i+\mu\sum_{i,j=1}^{n}w'_{ij}\sum_{y\in \mathrm{Y}}\frac{p_i^{(1)}}{q_j^{(1)}}q_j.\end{aligned}\tag{6}$$

Now consider

$$\begin{aligned}&\mathcal{C}_{MP}(\mathrm{p},\mathrm{q})-\mathcal{C}_{MP}(\mathrm{p},\mathrm{q}^{(1)})\\&\quad=\sum_{i=1}^{l}\sum_{y\in \mathrm{Y}}\left(r_i\log\frac{r_i}{q_i}-r_i\log\frac{r_i}{q_i^{(1)}}\right)+\mu\sum_{i,j=1}^{n}w'_{ij}\sum_{y\in \mathrm{Y}}\left(p_i\log\frac{p_i}{q_j}-p_i\log\frac{p_i}{q_j^{(1)}}\right)\\&\quad=\sum_{i=1}^{l}\sum_{y\in \mathrm{Y}}r_i\log\frac{q_i^{(1)}}{q_i}+\mu\sum_{i,j=1}^{n}w'_{ij}\sum_{y\in \mathrm{Y}}p_i\log\frac{q_j^{(1)}p_i}{q_jp_i^{(1)}}-\mu\sum_{i,j=1}^{n}w'_{ij}D_{KL}(p_i||p_i^{(1)}).\end{aligned}$$

Thus we have that

$$\begin{aligned}&\mathcal{C}_{MP}(\mathrm{p},\mathrm{q})-\mathcal{C}_{MP}(\mathrm{p},\mathrm{q}^{(1)})+\delta(\mathrm{p},\mathrm{p}^{(1)})\\&\qquad=\sum_{i=1}^{l}\sum_{y\in \mathrm{Y}}r_i\log\frac{q_i^{(1)}}{q_i}+\mu\sum_{i,j=1}^{n}w'_{ij}\sum_{y\in \mathrm{Y}}p_i\log\frac{q_j^{(1)}p_i}{q_jp_i^{(1)}}.\end{aligned}$$

Using the variational inequality $-\log(x) \geq (1-x)$ in the above we get

$$\begin{aligned}
&\mathcal{C}_{MP}(\mathrm{p},\mathrm{q}) - \mathcal{C}_{MP}(\mathrm{p},\mathrm{q}^{(1)}) + \delta(\mathrm{p},\mathrm{p}^{(1)}) \\
&\quad \geq \sum_{i=1}^{l} \sum_{y\in \mathrm{Y}} r_i \left(1 - \frac{q_i}{q_i^{(1)}}\right) + \mu \sum_{i,j=1}^{n} w'_{ij} \sum_{y\in \mathrm{Y}} p_i \left(1 - \frac{q_j p_i^{(1)}}{q_j^{(1)} p_i}\right) \\
&\quad = l + \mu \sum_{i,j=1}^{n} w'_{ij} - \sum_{i=1}^{l} \sum_{y\in \mathrm{Y}} \frac{r_i}{q_i^{(1)}} q_i - \mu \sum_{i,j=1}^{n} w'_{ij} \sum_{y\in \mathrm{Y}} \frac{p_i^{(1)}}{q_j^{(1)}} q_j \\
&\quad \overset{(a)}{\geq} 0
\end{aligned}$$

where (a) follows from Equation 6. ■

Which implies 4-pp holds for $\mathcal{C}_{MP}$.

Theorem 18 *$\mathcal{C}_{MP}(\mathrm{p},\mathrm{q})$ satisfies the 5-pp.*

Proof Follows as $\mathcal{C}_{MP}(\mathrm{p},\mathrm{q})$ satisfies both 3-pp and 4-pp. ■

Theorem 5 (Convergence of AM on $\mathcal{C}_{MP}$) *If*

$$\mathrm{p}^{(n)} = \operatorname*{argmin}_{\mathrm{p}\in\triangle^m} \mathcal{C}_{MP}(\mathrm{p},\mathrm{q}^{(n-1)}), \;\; \mathrm{q}^{(n)} = \operatorname*{argmin}_{\mathrm{q}\in\triangle^m} \mathcal{C}_{MP}(\mathrm{p}^{(n)},\mathrm{q}) \text{ and } q_i^{(0)}(y) > 0 \;\forall\, y \in \mathrm{Y},\; \forall i \text{ then}$$

(a) $\mathcal{C}_{MP}(\mathrm{p},\mathrm{q}) + \mathcal{C}_{MP}(\mathrm{p},\mathrm{p}^{(0)}) \geq \mathcal{C}_{MP}(\mathrm{p},\mathrm{q}^{(1)}) + \mathcal{C}_{MP}(\mathrm{p}^{(1)},\mathrm{q}^{(1)})$ *for all* $\mathrm{p},\mathrm{q} \in \triangle^m$, *and*

(b) $\lim_{n\to\infty} \mathcal{C}_{MP}(\mathrm{p}^{(n)},\mathrm{q}^{(n)}) = \inf_{\mathrm{p},\mathrm{q}\in\triangle^m} \mathcal{C}_{MP}(\mathrm{p},\mathrm{q})$.

Proof (a) follows as a result of Theorem 18. (b) is the direct result of (a) and theorem 3 in Csiszar and Tusnady (1984). ■

Appendix C. Equality of Solutions

Lemma 19 *If* $\mathrm{p} = \mathrm{q} = \tilde{\mathrm{p}}$ *then we have that* $\mathcal{C}_{MP}(\tilde{\mathrm{p}},\tilde{\mathrm{p}}) = \mathcal{C}_{KL}(\tilde{\mathrm{p}})$.

Proof Follows from the definitions of $\mathcal{C}_{KL}$ and $\mathcal{C}_{MP}$. ■

Lemma 6 *We have that*

$$\min_{\mathrm{p},\mathrm{q}\in\triangle^m} \mathcal{C}_{MP}(\mathrm{p},\mathrm{q}; w'_{ii} = 0) \leq \min_{\mathrm{p}\in\triangle^m} \mathcal{C}_{KL}(\mathrm{p}).$$

Proof Follows from the observation that

$$\min_{\mathrm{p}\in\triangle^m} \mathcal{C}_{KL}(\mathrm{p}) = \min_{\mathrm{p},\mathrm{q}\in\triangle^m,\mathrm{p}=\mathrm{q}} \mathcal{C}_{MP}(\mathrm{p},\mathrm{q};w'_{ii}=0) \geq \min_{\mathrm{p},\mathrm{q}\in\triangle^m} \mathcal{C}_{MP}(\mathrm{p},\mathrm{q};w'_{ii}=0\ \forall\ i)$$

The last step follows since the unconstrained minimum can never be larger than the constrained minimum. ■

Theorem 7 *Given any $A,B,S\in\triangle^m$ (i.e., $A=[a_1,\ldots,a_m]$, $B=[b_1,\ldots,b_m]$, $S=[s_1,\ldots,s_m]$) such that $a_i(y),b_i(y),s_i(y)>0,\ \forall\ i,y$ and $A\neq B$ (i.e., not all $a_i(y)=b_i(y)$) then there exists a finite α such that*

$$\mathcal{C}_{MP}(A,B) \geq \mathcal{C}_{MP}(S,S) = \mathcal{C}_{KL}(S).$$

Proof First

$$\begin{aligned}
\mathcal{C}_{MP}(A,B) &= \sum_{i=1}^{l} D_{KL}\big(r_i||b_i\big) + \mu\sum_{i=1}^{n}\sum_{j\in\mathcal{N}'(i)} w'_{ij} D_{KL}\big(a_i||b_j\big) - \nu\sum_{i=1}^{n} H(a_i) \\
&= \sum_{i=1}^{l} D_{KL}\big(r_i||b_i\big) + \mu\sum_{i=1}^{n}\sum_{j\in\mathcal{N}(i)} w_{ij} D_{KL}\big(a_i||b_j\big) - \nu\sum_{i=1}^{n} H(a_i) \\
&\qquad + \mu\sum_{i=1}^{m}(w_{ii}+\alpha) D_{KL}\big(a_i||b_i\big)
\end{aligned}$$

and so we want

$$\begin{aligned}
&\sum_{i=1}^{l} D_{KL}\big(r_i||b_i\big) + \mu\sum_{i=1}^{n}\sum_{j\in\mathcal{N}(i)} w_{ij} D_{KL}\big(a_i||b_j\big) - \nu\sum_{i=1}^{n} H(a_i) \\
&\qquad + \mu\sum_{i=1}^{m}(w_{ii}+\alpha) D_{KL}\big(a_i||b_i\big) - \mathcal{C}_{MP}(S,S) \geq 0
\end{aligned}$$

which holds if

$$\begin{aligned}
\alpha &\geq \frac{\mathcal{C}_{MP}(S,S) - \sum_{i=1}^{l} D_{KL}\big(r_i||b_i\big) - \mu\sum_{i,j} w_{ij} D_{KL}\big(a_i||b_j\big) + \nu\sum_i H(a_i)}{\mu\sum_i D_{KL}\big(a_i||b_i\big)} \\
&= \frac{\mathcal{C}_{MP}(S,S) - \mathcal{C}_{MP}(A,B;\alpha=0)}{\mu\sum_i D_{KL}\big(a_i||b_i\big)} = \frac{\mathcal{C}_{KL}(S) - \mathcal{C}_{MP}(A,B;\alpha=0)}{\mu\sum_i D_{KL}\big(a_i||b_i\big)}.
\end{aligned}$$

■

Theorem 8 (Equality of Solutions of $\mathcal{C}_{KL}$ and $\mathcal{C}_{MP}$) *Let*

$$\hat{\mathrm{p}} = \operatorname*{argmin}_{\mathrm{p}\in\triangle^m} \mathcal{C}_{KL}(\mathrm{p}) \text{ and } (\mathrm{p}^*_{\tilde{\alpha}},\mathrm{q}^*_{\tilde{\alpha}}) = \operatorname*{argmin}_{\mathrm{p},\mathrm{q}\in\triangle^m} \mathcal{C}_{MP}(\mathrm{p},\mathrm{q};\tilde{\alpha})$$

for an arbitrary $\alpha = \tilde{\alpha} > 0$ *where* $\mathrm{p}^*_{\tilde{\alpha}} = (p^*_{1;\tilde{\alpha}}, \cdots, p^*_{m;\tilde{\alpha}})$ *and* $\mathrm{q}^*_{\tilde{\alpha}} = (q^*_{1;\tilde{\alpha}}, \cdots, q^*_{m;\tilde{\alpha}})$. *Then there exists a finite* $\hat{\alpha}$ *such that at convergence of AM, we have that* $\hat{\mathrm{p}} = \mathrm{p}^*_{\hat{\alpha}} = \mathrm{q}^*_{\hat{\alpha}}$. *Further, it is the case that if* $\mathrm{p}^*_{\tilde{\alpha}} \neq \mathrm{q}^*_{\tilde{\alpha}}$, *then*

$$\hat{\alpha} \geq \frac{\mathcal{C}_{KL}(\hat{\mathrm{p}}) - \mathcal{C}_{MP}(\mathrm{p}^*_{\tilde{\alpha}}, \mathrm{q}^*_{\tilde{\alpha}}; \alpha = 0)}{\mu \sum_{i=1}^n D_{KL}(p^*_{i;\tilde{\alpha}} || q^*_{i;\tilde{\alpha}})}$$

and if $\mathrm{p}^*_{\tilde{\alpha}} = \mathrm{q}^*_{\tilde{\alpha}}$ *then* $\hat{\alpha} \geq \tilde{\alpha}$.

Proof First if $\mathrm{p}^*_{\tilde{\alpha}} = \mathrm{q}^*_{\tilde{\alpha}}$, this means the minimum of the unconstrained version at $\tilde{\alpha}$ resulted in equality, and since this also considers all solutions where $p = q$, and since both $\mathcal{C}_{KL}$ and $\mathcal{C}_{MP}$ are strictly convex, we must have $\mathcal{C}_{MP}(\mathrm{p}^*_{\tilde{\alpha}}, \mathrm{q}^*_{\tilde{\alpha}}; \tilde{\alpha}) = \mathcal{C}_{KL}(\hat{\mathrm{p}})$. Also, since for any $p \neq q$ we have $\mathcal{C}_{MP}(\mathrm{p}, \mathrm{q}; \hat{\alpha}) > \mathcal{C}_{MP}(\mathrm{p}, \mathrm{q}; \tilde{\alpha})$ whenever $\hat{\alpha} \geq \tilde{\alpha}$, then for all $\hat{\alpha} \geq \tilde{\alpha}$, $\mathcal{C}_{MP}(\mathrm{p}^*_{\hat{\alpha}}, \mathrm{q}^*_{\hat{\alpha}}; \hat{\alpha}) = \mathcal{C}_{KL}(\hat{\mathrm{p}})$. Next if $\mathrm{p}^*_{\tilde{\alpha}} \neq \mathrm{q}^*_{\tilde{\alpha}}$, then from Theorem 7 we have that if

$$\infty > \hat{\alpha} \geq \frac{\mathcal{C}_{KL}(\hat{\mathrm{p}}) - \mathcal{C}_{MP}(\mathrm{p}^*_{\tilde{\alpha}}, \mathrm{q}^*_{\tilde{\alpha}}; \alpha = 0)}{\mu \sum_{i=1}^n D_{KL}(p^*_{i;\tilde{\alpha}} || q^*_{i;\tilde{\alpha}})}$$

we are guaranteed that $\mathrm{p}^*_{\hat{\alpha}} = \mathrm{q}^*_{\hat{\alpha}}$, thereby making the first case applicable. ■

Appendix D. Test for Convergence

Theorem 9 (Test for Convergence) *If* $\{(\mathrm{p}^{(n)}, \mathrm{q}^{(n)})\}_{n=1}^{\infty}$ *is generated by AM of* $\mathcal{C}_{MP}(\mathrm{p}, \mathrm{q})$ *and* $\mathcal{C}_{MP}(\mathrm{p}^*, \mathrm{q}^*) \triangleq \inf_{\mathrm{p},\mathrm{q} \in \triangle^n} \mathcal{C}_{MP}(\mathrm{p}, \mathrm{q})$ *then*

$$\mathcal{C}_{MP}(\mathrm{p}^{(n)}, \mathrm{q}^{(n)}) - \mathcal{C}_{MP}(\mathrm{p}^*, \mathrm{q}^*) \leq \sum_{i=1}^n \big(\delta(i \leq l) + d_i\big)\beta_i,$$

$$\beta_i \triangleq \log \sup_y \frac{q_i^{(n)}(y)}{q_i^{(n-1)}(y)}, \quad d_j = \sum_i w_{ij}.$$

Proof As $\mathcal{C}_{MP}(\mathrm{p}, \mathrm{q})$ satisfies the 5-pp we have that

$$\mathcal{C}_{MP}(\mathrm{p}, \mathrm{q}) + \mathcal{C}_{MP}(\mathrm{p}, \mathrm{q}^{(n-1)}) \geq \mathcal{C}_{MP}(\mathrm{p}, \mathrm{q}^{(n)}) + \mathcal{C}_{MP}(\mathrm{p}^{(n)}, \mathrm{q}^{(n)}) \quad \forall \mathrm{p}, \mathrm{q} \in \mathcal{P}.$$

Rearranging the terms we have that

$$\mathcal{C}_{MP}(\mathrm{p}^{(n)}, \mathrm{q}^{(n)}) - \mathcal{C}_{MP}(\mathrm{p}, \mathrm{q}) \leq \mathcal{C}_{MP}(\mathrm{p}, \mathrm{q}^{(n-1)}) - \mathcal{C}_{MP}(\mathrm{p}, \mathrm{q}^{(n)}).$$

As the above holds for all $\mathrm{p}, \mathrm{q} \in \mathcal{P}$, it follows that

$$\mathcal{C}_{MP}(\mathrm{p}^{(n)}, \mathrm{q}^{(n)}) - \mathcal{C}_{MP}(\mathrm{p}^*, \mathrm{q}^*) \leq \mathcal{C}_{MP}(\mathrm{p}^*, \mathrm{q}^{(n-1)}) - \mathcal{C}_{MP}(\mathrm{p}^*, \mathrm{q}^{(n)}).$$

Now

$$\begin{aligned}
\mathcal{C}_{MP}(\mathrm{p}^*,\mathrm{q}^{(n-1)}) - \mathcal{C}_{MP}(\mathrm{p}^*,\mathrm{q}^{(n)}) &= \sum_{i=1}^{l}\sum_{y} r_i(y)\log\frac{q_i^{(n)}(y)}{q_i^{(n-1)}(y)} + \mu\sum_{i,j=1}^{m} w_{ij}\sum_{y} p_i^*(y)\log\frac{q_j^{(n)}(y)}{q_j^{(n-1)}(y)} \\
&= \sum_{i=1}^{l}\mathbf{E}_{r_i}\left[\log\frac{q_i^{(n)}(y)}{q_i^{(n-1)}(y)}\right] + \mu\sum_{i,j=1}^{m} w_{ij}\,\mathbf{E}_{p_i^*}\left[\log\frac{q_j^{(n)}(y)}{q_j^{(n-1)}(y)}\right] \\
&\overset{(a)}{\leq} \sum_{i=1}^{l}\sup_{y}\left[\log\frac{q_i^{(n)}(y)}{q_i^{(n-1)}(y)}\right] + \mu\sum_{i,j=1}^{m} w_{ij}\sup_{y}\left[\log\frac{q_j^{(n)}(y)}{q_j^{(n-1)}(y)}\right] \\
&= \sum_{i=1}^{l}\log\sup_{y}\left[\frac{q_i^{(n)}(y)}{q_i^{(n-1)}(y)}\right] + \mu\sum_{i,j=1}^{m} w_{ij}\,\log\sup_{y}\left[\frac{q_j^{(n)}(y)}{q_j^{(n-1)}(y)}\right] \\
&= \sum_{i=1}^{m}\big(\delta(i\leq l)+d_i\big)\log\sup_{y}\frac{q_i^{(n)}(y)}{q_i^{(n-1)}(y)}
\end{aligned}$$

where (a) follows as $E(f(x)) \leq \sup f(x)$ and recall $d_j = \sum_i w_{ij}$.

■

Appendix E. Update Equations for $\mathrm{p}^{(n)}$ and $\mathrm{q}^{(n)}$

The Lagrangian (ignoring the non-negativity constraints) for solving $\min_{\mathrm{p}\in\triangle^n} \mathcal{C}_{MP}(\mathrm{p},\mathrm{q}^{(n-1)})$ is given by

$$\mathcal{L}(\mathrm{p},\Lambda) = \sum_{i=1}^{l} D_{KL}\big(r_i||q_i\big) + \mu\sum_{i,j=1}^{n} w'_{ij} D_{KL}\big(p_i||q_j^{(n-1)}\big) - \nu\sum_{i=1}^{n} H(p_i) + \sum_i \lambda_i\left(\sum_y p_i(y) - 1\right)$$

where $\Lambda = \{\lambda_1,\ldots,\lambda_n\}$. As KKT conditions apply (since we have a convex optimization problem), we have that $\nabla_{p_i(y)}\mathcal{L}(\mathrm{p},\Lambda) = 0$ and $\mathrm{p}\in\triangle^n$ at the optimal solution. Solving the above we have

$$\log p_i(y) = \frac{-\lambda_i - \beta_i^{(n-1)}(y)}{\alpha_i}.$$

Recall $\alpha_i = \nu + \mu\sum_j w'_{ij}$, $\beta_i^{(n-1)}(y) = -\nu + \mu\sum_j w'_{ij}(\log q_j^{(n-1)}(y) - 1)$. Using the above in Equation 7 leads to the dual problem in Λ which admits a closed form solution given by

$$\lambda_i = \alpha_i \log\left(\sum_y \exp^{\frac{\beta_i^{(n-1)}(y)}{\alpha_i}}\right) \implies \mathbf{p_i^{(n)}}(\mathbf{y}) = \frac{\mathbf{1}}{\mathbf{Z_i}}\exp^{\frac{\beta_\mathbf{i}^{(\mathbf{n-1})}(\mathbf{y})}{\alpha_\mathbf{i}}}.$$

Clearly $p_i^{(n)}(y) \geq 0,\ \forall\, i,y$.

The update for $\mathrm{q}^{(n)}$ may be obtained by constructing the Lagrangian for the optimization problem $\min_{\mathrm{q}\in\triangle^n} \mathcal{C}_{MP}(\mathrm{p}^{(n)},\mathrm{q})$ which is given by

$$\mathcal{L}(\mathrm{q},\Lambda) = \sum_{i=1}^{l} D_{KL}\big(r_i||q_i\big) + \mu \sum_{i,j=1}^{n} w'_{ij} D_{KL}\big(p_i^{(n)}||q_j\big) - \nu \sum_{i=1}^{n} H(p_i^{(n)}) + \sum_i \lambda_i \left(\sum_y q_i(y) - 1 \right) + \sum_{i,y} \sigma_{iy} q_i(y)$$

where $\Lambda = \{\lambda_1,\ldots,\lambda_n,\sigma_{11},\ldots,\sigma_{n|\mathrm{Y}|}\}$. In this case KKT conditions require that $\nabla_{q_i(y)}\mathcal{L}(\mathrm{q},\Lambda) = 0$, $\sum_y q_i(y) - 1 \;\forall\, y$, $\sigma_{iy} q_i(y) = 0 \;\forall\, i,y$ solving which yields

$$\mathbf{q_i^{(n)}}(\mathbf{y}) = \frac{\mathbf{r_i}(\mathbf{y})\delta(\mathbf{i}\le\mathbf{l}) + \mu\sum_{\mathbf{j}} \mathbf{w'_{ji}}\mathbf{p_j^{(n)}}(\mathbf{y})}{\delta(\mathbf{i}\le\mathbf{l}) + \mu\sum_{\mathbf{j}} \mathbf{w'_{ji}}}.$$

Appendix F. Convergence Rate of SQ-Loss

Lemma 21 (Linear Rate of Convergence, see page 64 in Bertsekas, 1999) *If $\{x_n\}$ is a convergent sequence such that $x_n \to 0$ and $x_n > 0 \;\forall\, n$, then x_n is said to converge linearly if*

$$\limsup_{n\to\infty} \frac{x_n}{x_{n-1}} \le \eta$$

where $\eta \in (0,1)$.

Theorem 11 (Geometric Rate of Convergence for SQ-Loss) *If*

(a) $\nu > 0$, *and*

(b) $\mathbf{W}$ *has at least one non-zero off-diagonal element in every row (i.e.,* $\mathbf{W}$ *is irreducible)*

then the sequence of updates

$$p_i^{(n)}(y) = \frac{r_i(y)\delta(i\le l) + \nu u(y) + \mu\sum_j w_{ij} p_j^{(n-1)}(y)}{\delta(i\le l) + \nu + \mu\sum_j w_{ij}}$$

has a linear (geometric) rate of convergence for all i and y.

Proof

The updates can re-written in matrix form as

$$\mathrm{p}^{(n)} = [S + \nu\mathbf{I}_m + \mu\mathbf{D}]^{-1}\left(\mathrm{r}' + \frac{\nu}{|\mathrm{Y}|}\mathbf{1}_{m\times|\mathrm{Y}|} + \mu\mathbf{W}\mathrm{p}^{(n-1)}\right)$$

where

$$S \triangleq \begin{pmatrix} \mathbf{I}_l & 0 \\ 0 & 0 \end{pmatrix}, \mathrm{r}' \triangleq \begin{pmatrix} \mathrm{r} \\ \mathbf{0}_{(m-l)\times|\mathrm{Y}|} \end{pmatrix},$$

$[\mathbf{D}]_{ii} = \sum_j w_{ij}$, $\mathbf{1}_{m\times|\mathrm{Y}|}$ is a matrix of all 1's of size $m \times |\mathrm{Y}|$ and $\mathbf{0}_{(m-l)\times|\mathrm{Y}|}$ is similarly defined to be matrix of all 0's . It can be shown that $\mathrm{p}^{(n)} \to \mathrm{p}^*$ and so we have that

$$\mathrm{p}^* = [S + \nu\mathbf{I}_m + \mu\mathbf{D}]^{-1}\left(\mathrm{r}' + \frac{\nu}{|\mathrm{Y}|}\mathbf{1}_{m\times|\mathrm{Y}|} + \mu\mathbf{W}\mathrm{p}^*\right).$$

As a result

$$\mathrm{p}^{(n)} - \mathrm{p}^* = [S + \nu\mathbf{I}_m + \mu\mathbf{D}]^{-1}\left(\mu\mathbf{W}(\mathrm{p}^{(n-1)} - \mathrm{p}^*)\right)$$

which implies that

$$\| \mathrm{p}^{(n)} - \mathrm{p}^* \| = \| [S + \nu\mathbf{I}_m + \mu\mathbf{D}]^{-1}\left(\mu\mathbf{W}(\mathrm{p}^{(n-1)} - \mathrm{p}^*)\right) \|$$

where $\| A \|$ is the 2-norm (Euclidean norm) of the matrix A. Thus

$$\begin{aligned}\| \mathrm{p}^{(n)} - \mathrm{p}^* \| &= \| [S + \nu\mathbf{I}_m + \mu\mathbf{D}]^{-1}\left(\mu\mathbf{W}(\mathrm{p}^{(n-1)} - \mathrm{p}^*)\right) \| \\ &\leq \mu \| [S + \nu\mathbf{I}_m + \mu\mathbf{D}]^{-1}\mathbf{W} \| \, \| \mathrm{p}^{(n-1)} - \mathrm{p}^* \|\end{aligned}$$

and so

$$\frac{\| \mathrm{p}^{(n)} - \mathrm{p}^* \|}{\| \mathrm{p}^{(n-1)} - \mathrm{p}^* \|} \leq \mu \| [S + \nu\mathbf{I}_m + \mu\mathbf{D}]^{-1}\mathbf{W} \| .$$

Let $Z \triangleq \frac{1}{\mu}S + \frac{\nu}{\mu}\mathbf{I}_m + \mathbf{D}$ and so

$$\frac{\| \mathrm{p}^{(n)} - \mathrm{p}^* \|}{\| \mathrm{p}^{(n-1)} - \mathrm{p}^* \|} \leq \| Z^{-1}\mathbf{W} \| .$$

It should be clear that Z is a diagonal matrix.

The Perron-Frobenius theorem states that given any irreducible matrix A such that $a_{ij} \geq 0$ and a_{ij} are real then

$$\min_i \sum_j a_{ij} \leq \lambda_{max}(A) \leq \max_i \sum_j a_{ij}$$

where $\lambda_{max}(A)$ represents the maximum eigenvalue of A. If we apply the above theorem to the matrix $\mathbf{D}^{-1}\mathbf{W}$, then we have that $\lambda_{max}(\mathbf{D}^{-1}\mathbf{W}) = 1$. If we apply the same to $Z^{-1}\mathbf{W}$, then we have that

$$\min_i \sum_j \frac{w_{ij}}{\frac{1}{\mu}\delta(i \leq l) + \frac{\nu}{\mu} + \sum_k w_{ik}} \leq \lambda_{max}(Z^{-1}\mathbf{W}) \leq \max_i \sum_j \frac{w_{ij}}{\frac{1}{\mu}\delta(i \leq l) + \frac{\nu}{\mu} + \sum_k w_{ik}}.$$

But we have that

$$\sum_j \frac{w_{ij}}{\sum_k w_{ik}} = 1 \tag{7}$$

and so if $\nu > 0$ then we have that

$$\sum_j \frac{w_{ij}}{\frac{1}{\mu}\delta(i \leq l) + \frac{\nu}{\mu} + \sum_k w_{ik}} < 1.$$

As a result

$$\min_i \sum_j \frac{w_{ij}}{\frac{1}{\mu}\delta(i \leq l) + \frac{\nu}{\mu} + \sum_k w_{ik}} \leq \lambda_{max}(Z^{-1}\mathbf{W}) < 1.$$

In addition we also have that $\sum_j w_{ij} > 0$ for all i and so

$$0 < \lambda_{max}(Z^{-1}\mathbf{W}) < 1.$$

As a result

$$\| Z^{-1}\mathbf{W} \| = \sqrt{\lambda_{max}\left((Z^{-1}\mathbf{W})^T Z^{-1}\mathbf{W}\right)} = \sqrt{\lambda_{max}\left(Z^{-1}\mathbf{W}\right)^2} = \lambda_{max}(Z^{-1}\mathbf{W}).$$

The above implies that

$$\limsup_{n\to\infty} \frac{\| \mathrm{p}^{(n)} - \mathrm{p}^* \|}{\| \mathrm{p}^{(n-1)} - \mathrm{p}^* \|} \leq \| Z^{-1}\mathbf{W} \| = \lambda_{max}(Z^{-1}\mathbf{W}).$$

As $0 < \lambda_{max}(Z^{-1}\mathbf{W}) < 1$, we have that $\mathrm{p}^{(n)}$ has a linear rate of convergence.

■

References

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

A. Alexandrescu and K. Kirchhoff. Graph-based learning for statistical machine translation. In *Proc. of the Human Language Technologies Conference (HLT-NAACL)*, 2007b.

S. Arya and D. M. Mount. Approximate nearest neighbor queries in fixed dimensions. In *ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 1993.

S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching. *Journal of the ACM*, 1998.

A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with bregman divergences. *Journal of Machine Learning Research*, 2005.

R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter. Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research*, 3:1183–1208, 2003. ISSN 1533-7928.

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

Y. Bengio, O. Delalleau, and N. L. Roux. Label propogation and quadratic criterion. In O. Chapelle, B. Scholkopf, and A. Zien, editors, *Semi-Supervised Learning*. MIT Press, 2007.

D. Bertsekas. *Nonlinear Programming*. Athena Scientific Publishing, 1999.

J. Bilmes and A. Subramanya. Parallel graph-based semi-supervised learning. In R. Bekkerman, M. Bilenko, and J. Langford, editors, *Scaling Up Machine Learning*. Cambridge University Press, 2011. Forthcoming.

C. Bishop, editor. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

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

A. Blum and S. Chawla. 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, 2001.

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

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2006.

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

W. Cheney and A. Goldstien. Proximity maps for convex sets. *American Mathematical Society*, 1959.

R. Collobert, F. Sinz, J. Weston, L. Bottou, and T. Joachims. Large scale transductive svms. *Journal of Machine Learning Research*, 2006.

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

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

I. Csiszar and G Tusnady. Information geometry and alternating minimization procedures. *Statistics and Decisions*, 1984.

O. Delalleau, Y. Bengio, and N. L. Roux. Efficient non-parametric function induction in semi-supervised learning. In *Proc. of the Conference on Artificial Intelligence and Statistics (AISTATS)*, 2005.

A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

N. Deshmukh, A. Ganapathiraju, A. Gleeson, J. Hamaker, and J. Picone. Resegmentation of switchboard. In *Proceedings of the International Conference on Spoken Language Processing*, pages 1543–1546, Sydney, Australia, November 1998.

S. Dumais, J. Platt, D. Heckerman, and M. Sahami. 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, 1998.

G. Evermann, H. Y. Chan, M. J. F. Gales, B. Jia, D. Mrva, P. C. Woodland, and K. Yu. Training lvcsr systems on thousands of hours of data. In *Proc. of ICASSP*, 2005.

W. Fei and Z. Changshui. Label propagation through linear neighborhoods. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 985–992, New York, NY, USA, 2006. ACM.

J.H. Friedman, J.L. Bentley, and R.A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transaction on Mathematical Software*, 3, 1977.

J. Godfrey, E. Holliman, and J. McDaniel. Switchboard: telephone speech corpus for research and development. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 517–520, San Francisco, California, March 1992.

A. B. Goldberg and X. Zhu. Keepin' it real: Semi-supervised learning with realistic tuning. In *SemiSupLearn '09: Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *CAP*, 2005.

S Greenberg. The switchboard transcription project. Technical report, The Johns Hopkins University (CLSP) Summer Research Workshop, 1995.

A. Gunawardena. *The Information Geometri of EM Variants for Speech and Image Processing*. PhD thesis, Johns Hopkins University, 2001.

A. K. Halberstadt and J. R. Glass. Heterogeneous acoustic measurements for phonetic classification. In *Proc. Eurospeech '97*, pages 401–404, Rhodes, Greece, 1997. URL `citeseer.ist.psu.edu/article/halberstadt97heterogeneous.html`.

H.L.Royden. *Real Analysis*. Prentice Hall, 1988.

G. Ji and J. Bilmes. Dialog act tagging using graphical models. In *Proc. of ICASSP*, 2005.

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

T. Joachims. SVM Light. `http://svmlight.joachims.org`, 2002.

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

T. Joachims. SGT light. `http://sgt.joachims.org`, 2004.

D. Jurafsky and C. V. Ess-Dykema. Switchboard discourse language modeling project. Johns Hopkins Summer Workshop, 1997.

M. Karlen, J. Weston, A. Erkan, and R. Collobert. Large scale manifold transduction. In *International Conference on Machine Learning, ICML*, 2008.

J. Lafferty, S. D. Pietra, and V. D. Pietra. Statistical learning algorithms based on bregman distances. In *Proceedings of 1997 Canadian Workshop on Information Theory*, 1997.

K. F. Lee and H. Hon. Speaker independant phone recognition using hidden Markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(11), 1989.

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

X. Li and J. Bilmes. Regularized adaptation of discriminative classifiers. In *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Processing*, September 2006.

J. Malkin, A. Subramanya, and J. A. Bilmes. On the semi-supervised learning of multi-layered perceptrons. In *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Brighton, UK, September 2009.

G. S. Mann and A. McCallum. Simple, robust, scalable, semi-supervised learning via expectation regularization. In *Proc. of the International Conference on Machine Learning (ICML)*, 2007.

C. Martínez-Hinarejos, J. Benedí, and R. Granell. Statistical framework for a spanish spoken dialogue corpus. *Speech Communication*, 50(11-12), 2008.

N. Morgan. Personal communication, 2009.

B. Nadler, N. Srebro, and X. Zhou. Statistical analysis of semi-supervised learning: The limit of infinite unlabelled data. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

A. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems (NIPS)*, 2002.

K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. 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, 1998.

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

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

V. Raghavan, P. Bollmann, and G. S. Jung. A critical investigation of recall and precision as measures of retrieval system performance. *ACM Trans. Inf. Syst.*, 7(3):205–229, 1989. ISSN 1046-8188.

R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 2004.

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

H. J. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11, 1965.

M. Seeger. Learning with labeled and unlabeled data. Technical report, University of Edinburgh, U.K., 2000.

V. Sindhwani and S. Keerthi. Large scale semi-supervised linear svms. In *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR*, 2006.

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

P. Somervuo. Experiments with linear and nonlinear feature transformations in HMM based phone recognition. In *Proc. of ICASSP 2003*, 2003. URL `citeseer.ist.psu.edu/somervuo03experiments.html`.

A. Subramanya and J. Bilmes. Soft-supervised text classification. In *EMNLP*, 2008.

A. Subramanya and J. Bilmes. The semi-supervised switchboard transcription project. In *Interspeech*, 2009.

A. Subramanya, C. Bartels, J. Bilmes, and P. Nguyen. Uncertainty in training large vocabulary speech recognizers. In *Proc. of the IEEE Workshop on Speech Recognition and Understanding*, 2007.

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

A. Tomkins. Keynote speech. CIKM Workshop on Search and Social Media, 2008.

I. W. Tsang and J. T. Kwok. Large-scale sparsified manifold regularization. In *Advances in Neural Information Processing Systems (NIPS) 19*, 2006.

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

K. Tsuda, G. Rätsch, and M. K. Warmuth. Matrix exponentiated gradient updates for on-line learning and bregman projection. *J. Mach. Learn. Res.*, 6:995–1018, 2005. ISSN 1533-7928.

V. Vladimir. *Statistical Learning Theory*. Wiley Series, 1998.

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

C. F. Jeff Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1): 95–103, 1983.

W. Zangwill. *Nonlinear Programming: a Unified Approach*. Prentice-Hall International Series in Management, Englewood Cliffs: N.J., 1969.

X.H. Zhang and W.S. Lee. Hyperparameter learning for semi-supervised learning algorithms. In *NIPS*, 2006.

D. Zhou, J. Huang, and B. Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *Proceedings of the 22nd International Conference on Machine Learning*. ACM, 2005.

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

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

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

X. Zhu and Z. Ghahramani. Towards semi-supervised classification with Markov random fields. Technical Report CMU-CALD-02-106, Carnegie Mellon University, 2002b.

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

X. Zhu, J. Kandola, Z. Ghahramani, and J. Lafferty. Nonparametric transforms of graph kernels for semi-supervised learning. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems (NIPS)*, 2005.

V.W. Zue, S. Seneff, and J. Glass. Speech database development at MIT:TIMIT and beyond. In *Speech Comminication*, 1990.