Fast Semi-Supervised SVM Classifiers Using A-Priori Metric Information

Volkan Vural [*], Glenn Fung, Jennifer G. Dy and Bharat Rao

6/26/07

Abstract

This paper describes a support vector machine-based (SVM) parametric optimization method for semi-supervised classification, called LIAM (for LInear hyperplane classifier with A-priori Metric information). Our method takes advantage of similarity information to leverage the unlabeled data in training SVMs. In addition to the smoothness constraints in existing semi-supervised methods, LIAM incorporates local class similarity constraints, that we empirically show, improved the accuracies in the presence of a few labeled points. We present and discuss a general convex mathematical-programming-based formulation to solve the inductive semi-supervised problem; i.e., our proposed algorithm directly classifies test samples not present when training. This general formulation results in different variants depending on the choice of the norms that are used in the objective function. For example, when using the 1-norm the proposed formulation becomes a linear programming problem (LP) that has the advantage of generating sparse solutions depending on a minimal set of the original features (feature selection). On the other hand, one of the proposed formulations results in an unconstrained quadratic problem for which solutions can be obtained by solving a simple system of linear equations, resulting in a fast competitive alternative to state-of-the-art semi-supervised algorithms. Our experiments on public benchmarks indicate that LIAM is at least one order of magnitude faster and at least as or more accurate (in most of the cases) than other state-of-the-art semi-supervised classification methods.

1 Introduction

Supervised classification algorithms, such as support vector machines (SVM) can only use the information provided by the labeled instances to produce the optimal classifier. However, in many domains, labeled instances are typically costly to obtain. This is particularly true for the medical domains that motivate our research, where labels are assigned via time-consuming manual review by

[**]Corresponding author.Email: vvural@ece.neu.edu

physicians, or via expensive additional tests. On the other hand, unlabeled instances are often plentiful, and relatively easy to obtain. Therefore, *semi-supervised* algorithms that can use the information provided by both labeled and unlabeled instances to build classifiers are of increasing interest.

Recently, many semi-supervised learning methods have been introduced [1, 2, 3, 7, 9, 11, 12, 16, 20, 21, 23]. Comprehensive reviews are provided in [15, 22] on semi-supervised learning algorithms. One popular approach for semi-supervised learning is based on a weighted graph [1, 3, 7, 12, 21, 20, 23] where labeled and unlabeled points constitute the vertices of the graph and the similarities between the data point pairs are represented by the edge weights. A function is then used to label the unlabeled points on the graph. The method for finding the weights and the selection of the labeling function may vary. Most of these graph-based methods such as [1, 3, 7, 21, 20, 23] assume a *transductive* setting. In the transductive setting, the learner needs to observe the unlabeled or in other words the testing data while training; and therefore, although accurate, these transductive algorithms need to be retrained every time a test sample is to be classified. As a result, transductive algorithms may not satisfy the run-time requirements for many real-world applications, including computer-aided diagnosis applications where new patient cases need to be classified in real-time as part of the physician's work flow. In [2] Bennett et al. introduced a mixed integer programming (MIP) formulation that results in *inductive* classifiers (i.e., the algorithm produces a classifier that can be used directly to classify new samples without retraining). However, the method require a complex optimization solver and it is not feasible for data where the size of the unlabeled set is not small. There are methods that attempt to find efficient approximate solutions to the MIP formulation [9, 11], the drawback of these formulations is that they converge to a local minimum which may not be a sufficiently "good" solution.

In this paper, we introduce a new SVM-based algorithm called LIAM (LInear hyperplane classifier with A-priori Metric information). LIAM is an inductive semi-supervised algorithm, which makes it more efficient than transductive algorithms in terms of testing time. Additionally, training with our proposed algorithm is substantially faster and more efficient than other mathematical programming-based methods, like the ones introduced in [2, 9], which makes LIAM an option to consider when working with large datasets. We also extend the notion of graph regularization for semi-supervised learning [17, 12] (that is usually done using the 2-norm) to other norms, such as the 1-norm and the ∞-norm. The use of different norms lead to classifiers with different properties including, sparsity and low computational cost, as in one of the variants that only requires to solve a simple system of linear equations. Moreover, as opposed to most semi-supervised approaches that only consider smoothness constraints [1, 3, 16, 20, 23], we add a local class similarity constraint that helps improve the performance when the labeled examples are few.

The rest of the paper is organized as follows: The next section provides a motivation for our approach. Section 3 briefly reviews standard SVM and Section 4 provides the derivation for our semi-supervised learning algorithm, LIAM. In Section 5.3, we present several variants of LIAM for different norms

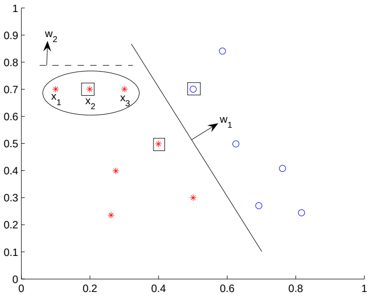

Figure 1: An example that illustrates how smoothness assumption fails. The smoothness assumption enforces a separating hyperplane parallel to the horizontal dashed line for the encircled neighboring points x_1, x_2 and x_3. However, the correct separating function that is shown as the solid line is far from being horizontal. On the other hand, the second premise works well in this example. The data points in boxes represent the labeled points in the Figure.

(1-norm, ∞-norm, and a fast 2-norm variant of LIAM that results in an unconstrained convex mathematical-programming-based formulation with a unique global solution). Experimental results on four public benchmark datasets and a medical classification problem are presented in Section 6. We conclude in Section 7 with some thoughts on future research directions. Is it important to note that even when all the algorithms presented in this work produce linear classifiers, extensions to the "kernelized" versions are relatively straightforward by using a similar approach to the one presented in [13].

2 Motivation

In a semi-supervised scenario, we are provided with a set of labeled and a set of unlabeled instances. Our goal is to devise methods that take advantage of the information available in the unlabeled data to produce more accurate classifiers. To take advantage of the information provided by the available unlabeled data, we want to enforce in an efficient way the following two premises:

1. **The classification function $f(x) = w'x - \gamma$ values should be similar for neighboring points.**

This premise is often referred to as the smoothness assumption in graph based semi-supervised algorithms and is also proposed in [1, 3, 16, 20, 23].

Although the smoothness assumption has been utilized with success in the past, it may not work well or even fail in some cases. For example, let us consider the toy example described in Figure 1. Consider the encircled neighboring points x_1, x_2 and x_3. The smoothness assumption enforces a separating function, $f(x) = w_2'x - b_2$ that produces very similar values for the points x_1, x_2 and x_3. Under this premise the "ideal" optimal hyperplane is forced to be parallel to the dashed line shown in Figure 1. However, as can be seen in the example, the correct separating function (solid line) is far from being horizontal. In order to compensate for this, we introduce the following non-smooth (only locally smooth) premise.

2. **Unlabeled points close to labeled ones should be in the same class as the labeled point.**

In this second premise, $f(x)$ does not have to produce similar values for neighboring points. This premise is complied with as long as the sign of $f(x)$ is the same for the neighboring points. Consider the example in Figure 1. The second assumption would enforce the data point x_3 to be in the same class as x_2 by pushing the separating hyperplane away, which makes x_3 behave like a support vector although it is not a labeled point. In this particular scenario the second premise corrects the first one and improves the standard SVM classifier. However, there is still a drawback of this premise: It can only be used for the unlabeled points that are in the neighborhood of a labeled point. Hence, we incorporate both premises into our proposed algorithm in the following sections.

3 Semi-supervised p-norm-SVM

Before we present our semi-supervised algorithm, we define our notations and provide a brief review of standard SVM. The notation $A \in R^{m \times n}$ signifies a real $m \times n$ matrix. For such a matrix, A' denotes the transpose of A and A_i the i-th row of A. All vectors are column vectors. For $x \in R^n$, $\|x\|_p$ denotes the p-norm, $p = 1, 2, \infty$. A vector of ones and zeros in a real space of arbitrary dimensions are denoted by e and 0 respectively. Thus, for $e \in R^m$ and $y \in R^m$, $e'y$ is the sum of the components of y.. A *separating hyperplane*, $f(x) = w'x - \gamma$, with respect to two given point sets $\mathcal{A}^+$ and $\mathcal{A}^-$, is a plane that attempts to separate R^n into two half spaces such that each open half space contains points mostly of $\mathcal{A}^+$ or $\mathcal{A}^-$. Note that $\mathcal{A}^+ \in R^{m^+ \times n}$ and $\mathcal{A}^- \in R^{m^- \times n}$ represent the positively and negatively labeled data sets respectively. In addition to the labeled data sets, $\mathcal{A}^+$ and $\mathcal{A}^-$, the matrix $U \in R^{q \times n}$ represents the unlabeled data set and $C = A^+ \cup A^- \cup U \in R^{(q+m) \times n}$ represents the entire data set including $m = m^+ + m^-$ labeled and q unlabeled instances. Using this notation, a general SVM formulation can be written as follows:

$$\begin{array}{rl} \displaystyle\min_{(w,\gamma,y^+,y^-)} & \|w\|_p + \nu(\|y^+\|_p + \|y^-\|_p) \\ \text{s.t.} & A^+w - e\gamma + y^+ \geq e \\ & A^-w - e\gamma - y^- \leq -e \\ & y^+, y^- \geq 0, \end{array} \tag{1}$$

where y^+ and y^- are slack variables.

Different choices of p would lead to different well-known SVM formulations. For example when $p = 2$, we obtain the standard quadratic programming formulation ($p = 2$) [19]. When $p = 1$ formulation (1) becomes a linear programming problem [4] that is known to produce sparse solutions and incorporates feature selection to the classification problem. A combination of different norms (one for the slack variables and one for the regularization term) is also commonly used, resulting in different variants of the formulation above. Based on the general SVM formulation presented in this section, we introduce our semi-supervised algorithm in the next section, that results in different optimization problems depending on the choice of the norm.

4 LInear hyperplane classifier with A-priori Metric information (LIAM)

In this section, we describe how to integrate the two premises presented in Section 2 into the general p-norm SVM formulation (1) to obtain different variants of our proposed semi-supervised learning algorithm.

Before describing how to incorporate premises one and two the general p-norm SVM formulation, let's consider a function $r(x_i, x_j)$ that represents similarity relations between any given data point pair. For the rest of the paper, $r(x_i, x_j)$ is assumed to be defined by the user *a-priori* and can be any kind of similarity function that maintains $0 \leq r(x_i, x_j) \leq 1$. Note that, this similarity function r defines an undirected weighted graph G where each vertex represents every point on the training set and the weight associated to each edge (i, j) is given by $r(x_i, x_j)$. We present the details of the function, r, that we used in our experiments in Section 6. Now we are ready to incorporate the premises into the SVM formulation.

Premise 1: Let us consider the first notion that was introduced earlier: namely, that the separator function should give similar values for neighboring points. In other words, we enforce the constraint that the value of the separator function should change smoothly over neighboring data points. This notion is the main basis for most of the recently proposed semi-supervised algorithms [1, 3, 16, 20, 23]. We begin by defining a set S that consists of the data points in the k-neighborhood of an arbitrary data point x in the training set. We want the classifier function $f(x) = w'x - \gamma$ to change smoothly over the set S, in other words, we want to minimize $\left\| f(x) - \frac{1}{N} \sum_{i \in S} r(x, x_i) f(x_i) \right\|_p$, where $N = \sum_{i \in S, x_i \neq x} r(x, x_i)$. Minimizing the above equation for each point $x \in C$ is equivalent to minimizing the following:

$$\left\| \tilde{L}(Cw - e\gamma) \right\|_p = \left\| \tilde{L}Cw \right\|_p \text{, since } \tilde{L}e = 0 \qquad (2)$$

Here $\tilde{L}$ is defined as: $\tilde{L}_{ij} = \begin{cases} 1 & i = j \\ -kr(C_i, C_j) & i \neq j \end{cases}$, where $k = 1/\sum_i^{m+q} r(C_i, C_j)$. In the literature, $\tilde{L}$ is referred to as the normalized Laplacian of the graph G [17].

Premise 2: We also want to consider the second notion introduced in 2. In order to improve the accuracy of our classifier by taking advantage of the unlabeled data, we propose some modifications to the linear constraints that implicitly define the margin. The main idea springs from our second intuition outlined earlier: it consists in spreading the information provided by a labeled instance ($A_j^{\mp}$) through the unlabeled instances that are in its neighborhood.

Using the previously defined similarity function r, let us define a diagonal matrix, $R^+ \in R^{q\times q}$, that represents the degree of similarity between any unlabeled data point U_j and the set of all positively labeled points in the training set, (A_i^+). An arbitrary diagonal element of R_{jj}^+ indicates the value of the similarity function r for the data pair, U_j and $A_{i^*}^+$ where $A_{i^*}^+$ is the most similar positive point to the unlabeled data point U_j. This can also be presented as: $R_{jj}^+ = max(r(A_i^+, U_j)), i \in \{1, \ldots, m^+\}, j \in \{1, \ldots, q\}$, and similarly we can define the diagonal matrix $R^- \in R^{q\times q}$ such that $R_{jj}^- = max(r(A_i^-, U_j)), i \in \{1, \ldots, m^-\}, j \in \{1, \ldots, q\}$. Using $R^{\mp}$, we utilize the unlabeled data to formulate two new constraints similar to the ones in (1):

$$\begin{array}{l} R^+Uw - R^+e\gamma + z^+ \geq e \\ R^-Uw - R^-e\gamma - z^- \leq -e \\ z^+, z^- \geq 0 \end{array} \tag{3}$$

where z^+ and z^- are the slack variables for the unlabeled data. In the formulation above, an arbitrary diagonal component of the $R^{\mp}$ matrices, $R_{jj}^{\mp}$, indicates how certain we are about the label of the corresponding unlabeled point, U_j. In the marginal case where $R_{jj}^{\mp} = 1$, U_j would be treated as a labeled point. Note that $R_{jj}^{\mp}$ may be very small ≈ 0 or 0, (depending on r) for some j, which means that there is effectively no edge between the unlabeled point U_j and the labeled set A. Therefore, a subset of the unlabeled instances would not be covered by the constraints (3) and the information provided by this subset of the unlabeled set would not be taken into account. Hence we integrate both Equations (2) and (3) into formulation (1) in order to obtain the following general p-norm semi-supervised formulation:

$$\begin{array}{rl} \min\limits_{(w,\gamma,y^{\mp},z^{\mp},)} & \|w\|_p + \nu y + \mu z + \alpha \left\| \tilde{L}Cw \right\|_p \\ \text{s.t.} & \mp(A^{\mp}w - e\gamma) + y^{\mp} \geq e \\ & \mp R^{\mp}(Uw - e\gamma) + z^{\mp} \geq e \\ & y^{\mp}, z^{\mp} \geq 0 \end{array} \tag{4}$$

where $y = \|y^+\|_p + \|y^-\|_p$ and $z = \|z^+\|_p + \|z^-\|_p$. Note that the parameters ν, μ and α in the objective function above, help us determine the trade off between the importance given to the labeled and unlabeled data according to the two premises described earlier. Figure 2 illustrates how our approach, LIAM, takes advantage of unlabeled instances to improve the decision boundary. Figure 2 contains two synthetic datasets, wherein only one point from each class is labeled and the rest of the points are unlabeled. The labeled points are displayed in squares. In the linearly separable example, the standard SVM will ignore the unlabeled instances and produce the bounding planes represented by the dashed vertical lines shown in Figure 2(a). However, as more instances are labeled, SVM will eventually learn the correct bounding planes, similar to the ones represented by the oblique solid lines in the same figure and that obtained by incorporating the extra unlabeled data. Figure 2(b) shows a similar synthetic example in a nonlinearly separable setting.

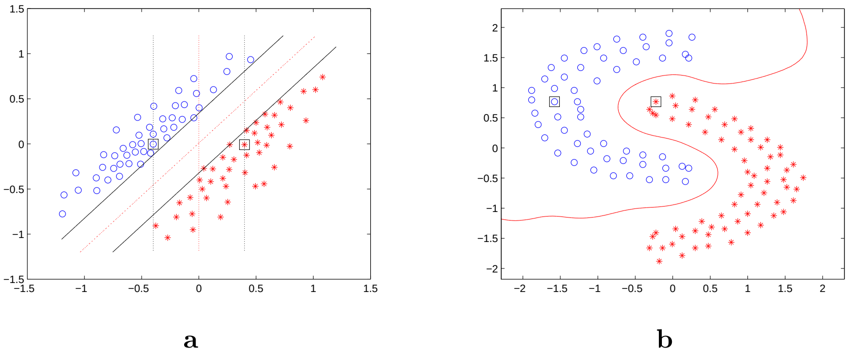

Figure 2: **a)** A comparison of standard SVM trained using the two labeled points shown in squares with the proposed semi-supervised algorithm (LIAM). The classifier boundary and the margin displayed in dashed lines are produced by standard SVM; whereas, the margin and boundary obtained by LIAM are shown in solid lines. **b)** A linearly inseparable data set classified by LIAM using radial basis function kernel. The two data points shown in squares are used as labeled data and the rest as unlabeled data.

5 LIAM with Different Norms

Equation (4) represents the general form of LIAM. We can reformulate equation (4) using different norms. The widely-used 2-norm in standard SVMs leads to a quadratic optimization problem that can be costly in terms of time. We now introduce three different versions of LIAM that result in computationally less expensive optimization problems.

5.1 1-norm: $LIAM_1$

In Section 3, we presented the 1-norm SVM formulation that results in a linear program. Similarly, we can convert the general LIAM formulation (4) into a linear program as follows:

$$\begin{array}{rl} \displaystyle\min_{(w,\gamma,y^+,y^-,t,s)} & e's + \nu(e'y^+ + e'y^-) + \mu(e'z^+ + e'z^-) + \alpha e't \\ \text{s.t.} & \mp(A^{\mp}w - e\gamma) + y^{\mp} \geq e \\ & \mp R^{\mp}(Uw - e\gamma) + z^{\mp} \geq e \\ & t \geq \tilde{L}Cw \geq -t \\ & s \geq w \geq -s \\ & y^+, y^- \geq 0, \end{array} \tag{5}$$

Formulation (5) inherits the property of generating sparse solutions, i.e. this formulation results in the normal w to the separating plane $x'w = \gamma$ having many zero components, which implies that many input space features do not play a role in determining the linear classifier. This makes this approach suitable for feature selection in classification problems. In formulation (5), we minimize

$\left\|\tilde{L}Cw\right\|_1$. This can be interpreted as minimizing $|\tilde{L}Cw|$ for every C_i separately, which means the linear program (5) needs to take into account $q+m$ separate components of the t vector. We can decrease the amount of components of the t vector by introducing the ∞ norm in the LIAM formulation.

5.2 ∞ norm: $LIAM_\infty$

In this version of LIAM we minimize $\left\|\tilde{L}Cw\right\|_\infty$ instead of $\left\|\tilde{L}Cw\right\|_1$. $\left\|\tilde{L}Cw\right\|_\infty$ can be written as $max(|(\tilde{L}C)_1 w|, |(\tilde{L}C)_2 w|, ..., |(\tilde{L}C)_{m+p} w|)$. In this case, a linear program needs to optimize the objective function for only one single scalar variable t, which makes the algorithm even more computationally efficient.

$$\begin{array}{cl} \min\limits_{(w,\gamma,y^+,y^-,t,s)} & e's + \nu(e'y^+ + e'y^-) + \mu(e'z^+ + e'z^-) + \alpha t \\ \text{s.t.} & \mp(A^\mp w - e\gamma) + y^\mp \geq e \\ & \mp R^\mp(Uw - e\gamma) + z^\mp \geq e \\ & et \geq \tilde{L}Cw \geq -et \\ & s \geq w \geq -s \\ & y^+, y^- \geq 0, \end{array} \qquad (6)$$

Next, we present a relaxed formulation that results in a fast unconstrained quadratic optimization problem whose solution can be obtained by solving a single system of linear equations of size n, where n is the number of features of the original data. This relaxed formulation will give us a tremendous computational advantage against other state-of-the-art methods, especially against the methods where a graph Laplacian, a matrix of the size $m >> n$, has to be inverted or its eigenvalues has to be calculated [6, 18, 21, 20]. Another advantage is that our proposed algorithm is inductive in nature, which means that it can classify data that was not available at the moment of training.

5.3 A fast unconstrained quadratic formulation for semi-supervised Classification: PLIAM

We can speed up the 2-norm formulation as follows: Setting $p = 2$ and following the same idea proposed in [8], we can slightly modify the inequalities in formulation (4) and substitute them by equalities to obtain:

$$\begin{array}{cl} \min\limits_{(w,\gamma,y,z)} & \|w\|_2{}^2 + \nu y + \mu z + \alpha\|\tilde{L}Cw\|_2^2 + \gamma^2 \\ \text{s.t.} & \mp(A^\mp w - e\gamma) + y^\mp = e \\ & \mp R^\mp(Uw - e\gamma) + z^\mp = e \end{array} \qquad (7)$$

where $y = \|y^+\|_2{}^2 + \|y^-\|_2{}^2$ and $z = \|z^+\|_2{}^2 + \|z^-\|_2{}^2$. Note that no explicit non negativity constraint is needed on the slack variables y, z, and that the margin is maximized with respect to both w and γ. This very simple modification changes the nature of the optimization problem significantly. Geometrically speaking, formulation (7) has an interpretation that differs from the standard SVM formulations. The planes $w'x-\gamma+1$ and $w'x-\gamma-1$ are not bounding planes anymore, instead they can be thought of as "proximal" planes, around which the points of each class are clustered and which are pushed as far apart as possible. Furthermore, the new equations introduced in the semi-supervised formulation by premise 2 also have a new and appealing meaning: we no longer require the unlabeled points that are close to the labeled ones to be strictly on the same side of the corresponding bounding plane; instead, we are asking for these unlabeled points to be closer to the plane that better fits the corresponding class. This formulation can indeed be written as an unconstrained quadratic problem by substituting the values of z and y in the objective function. As mentioned above, this formulation requires only solving a single system of linear equations; thus, it is substantially faster than standard SVMs while maintaining similar accuracy [8]. For simplicity of notations, let $L = (A^{+'}A^{+} + A^{-'}A^{-})$, $M = (A^{+} - A^{-})$, $N = (A^{+} + A^{-})$, $E = (R^{+'}R^{+} + R^{-'}R^{-})$ and $F = (R^{+} - R^{-})$. After taking derivatives and equating them to zero we can solve the optimization problem (7) by finding a solution to the following linear system of equations:

$$\begin{bmatrix} w \\ \gamma \end{bmatrix} = -2P^{-1}Q \quad \text{where } Q = \begin{bmatrix} -(\nu e'M+\mu e'FU) \\ (\nu(m^{+}-m^{-})+\mu(e'Fe)) \end{bmatrix} \tag{8}$$

and

$$P = \begin{bmatrix} I+\nu L+\mu U'EU+\alpha C'\tilde{L}'\tilde{L}C & -(\nu N'e+\mu U'Ee) \\ -(\nu e'N+\mu e'EU') & \nu m+\mu(e'Ee)+1 \end{bmatrix} \tag{9}$$

6 Experimental Results

We compare our LIAM variants with two semi-supervised approaches:

1. Transductive SVM (TSVM) introduced by [11] is an SVM-based semi-supervised algorithm, where the labels of the unlabeled points are initialized with the prediction of the SVM classifier trained on the labeled data. Then, the labels of the unlabeled points are changed as long as the margin is improved. However, TSVM may lead to a local optimum and can be time consuming. It is important to note that recently, a method to speed up TSVM has been proposed [5], however in this paper we utilized the original TSVM code included in the SVM$_{light}$ package.

2. Logistic Gaussian Random Field (LGRF) proposed by [12] is a graph-based algorithm. Unlike many other graph-based semi-supervised classifiers, LGRF is *inductive.*

6.1 The similarity function

As mentioned earlier, the similarity function $r(x_i, x_j)$ represents similarity relations between any given data point pair. For our experiments, we used the Euclidean distance between x_i and x_j to define the pairwise-similarity function: $r(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{\varsigma^2}}$. Experimentally, we found it useful to threshold the continuous similarity function, $r(x_i, x_j)$ to a new similarity function, $r^*(x_i, x_j)$ by applying a threshold such that $r^*(x_i, x_j) = \frac{r(x_i,x_j)}{\sum_{x_j \in S_k} r(x_i,x_j)}$ if x_j is an element of

	WDBC	Cleveland	Pima	Ionosphere	Lung
# of features	9	13	8	34	15
# of data points	683	297	768	351	21342

Table 1: Number of features and data points for each data set.

S_k that consists of the k points which are the most similar to x_i, $r^*(x_i, x_j) = 0$ otherwise.

6.2 Results on four publicly available datasets

We performed experiments on four UCI benchmark datasets [14] whose details are provided in Table 1 and compared LIAM to two other semi-supervised algorithms mentioned earlier: TSVM and Logistic GRF. We also present results for the standard SVM as a reference to see the improvement in accuracy by the help of unlabeled data, and the results for SVM with premise 1 alone and premise 2 alone. In our experiments, we equally divided the datasets into a training and a testing set. We randomly picked a portion of the training data as the labeled set and the rest of the training data as the unlabeled set. All the parameters were chosen by a cross-validation procedure in the available training set. For both the LGRF and LIAM algorithms, the parameter k, that corresponds to the number of considered nearby points to a labeled instance has to be tuned. We considered values of k ranging from 3 to 10 for LGRF (as suggested by one of the authors) and picked the best k value among 5, 10, 20 and 30 for LIAM. All the other parameters (ν, μ, α for LIAM and C for TSVM) were tuned in the range from 10^{-5} through 10^5 over the labeled data points.

Figures 3 a, b, c and d display the accuracies of the algorithms with respect to the number of labeled data points. We ran 10 trials on randomly selected labeled sets for each particular amount of labeled data. Each point on the plots represents the average accuracy from these 10 trials. These figures are results from inductive experiments; meaning, the test data are not treated as unlabeled data during training. We observe from Figures 3 a–d, that PLIAM outperforms TSVM, LGRF and standard SVM. The difference between PLIAM and the other algorithms is most significant when the amount of labeled data is relatively small. Note that the performance of PLIAM when only one of the premises is considered is also presented in the figures. The results empirically suggest that the combination of the two premises generally outperforms both of them when considered separately. Premise 2 performs better than premise 1 when there are few labels, whereas premise 1 is better when there are more labels, and LIAM combines both advantages. It is also interesting to note that when $\mu = 0$ in formulation (7) (only premise one is considered), the resulting optimization problem is very similar to the one proposed in [16]. As the number of labeled instances increases, LGRF and standard SVM catch up with LIAM. Note that in WDBC data set, PLIAM achieves the highest fully supervised accuracy level using only a small amount of labeled data. We observe in our

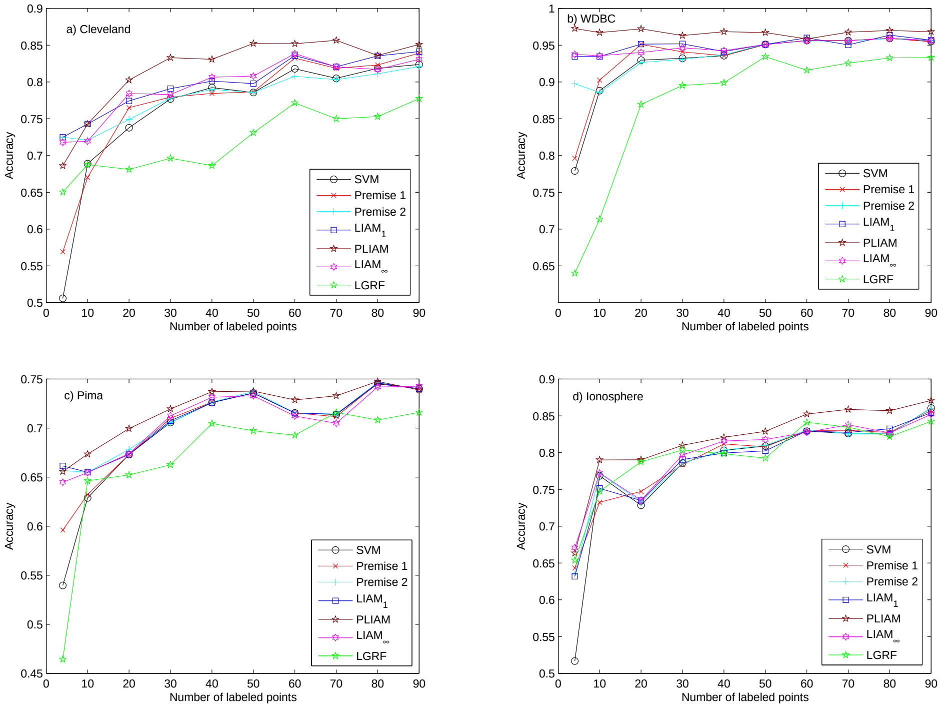

Figure 3: Results for four publicly available datasets.

experiments that PLIAM is more accurate than $LIAM_1$ and $LIAM_\infty$ in almost every case. Moreover, it is faster than those methods in training as shown in Table 2. Although $LIAM_\infty$ is faster than $LIAM_1$ in training, $LIAM_1$ was more accurate in our experiments.

6.3 Results on the LUNGCAD Dataset

LungCAD is a computer aided diagnosis system for detecting potentially cancerous pulmonary nodules from thin slice multi-detector computed tomography (CT) scans. The final output of LungCAD is provided by a classifier that classifies a set of candidates as positive or negative; obviously, high-sensitivity is critical as early detection of lung cancer is believed to greatly improve the chances of successful treatment. Furthermore, high specificity is also critical, as a large number of false positives will vastly increase physician load and lead (ultimately) to loss of physician confidence. This is a very hard classification problem: most patient lung CTs contain a few thousand structures (candidates), and only a few (≤ 5 on average) of which are potential nodules that should be

	WDBC	Cleveland	Pima	Ionosphere
$LGRF$	0.29	0.10	0.49	0.32
$TSVM$	2.75	1.39	821.54	57.84
$PLIAM$	**0.10**	**0.01**	**0.10**	**0.05**
$LIAM_1$	0.86	0.06	0.24	0.19
$LIAM_\infty$	0.73	0.05	0.21	0.14

Table 2: Average training times (in seconds) for LGRF, TSVM and LIAM on the benchmark datasets (**best is shown in bold**).

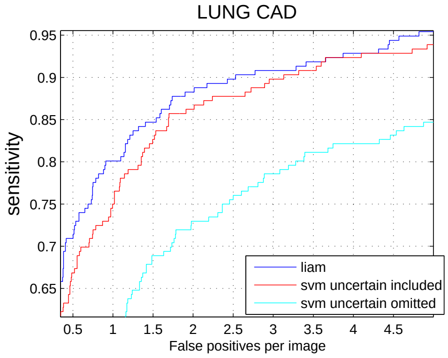

Figure 4: ROC curves for the LUNGCAD dataset.

identified as positive by LungCAD, all within the run-time requirements of completing the classification on-line during the time the physician completes their manual review.

As indicated in the introduction, getting labeled examples is hard in medical domains, and particularly difficult for lung cancer. Due to the high risks associated with lung biopsies (roughly 20% chance of complications, and a small chance of severe complications, including death), definitive diagnosis via biopsy are not obtained for most CT scans. Labels for our training data were assigned in a time-consuming manual review of potential candidates by a physician panel of leading lung radiologists. To further illustrate the difficulty of the problem, at the end there remained a number of candidates about which the panel was unable to agree upon a label. Thus, our training data set instances have one of *three* labels: nodule (positive), non-nodule (negative) and *uncertain*. (The uncertain points are candidates that are believed to have a small chance to be a nodule.)

For evaluation purposes, our test set only contained candidates that were labeled as nodules or non-nodules. The LUNGCAD dataset was split into three subsets. The training set comprised of 21342 candidates: 293 nodules, 20379 non-nodules and 670 uncertain, the validation set comprised of 2584 candidates: 36 nodules and 2548 non-nodules, and the testing set is formed by 1914 candidates: 31 nodules and 1883 non-nodules.

In clinical practice, CAD systems are evaluated on the basis of a somewhat domain-specific metric: maximize the fraction of *positives* that are correctly identified by the system while displaying at most a clinically acceptable number of false-marks per image. We report this domain-specific metric in an ROC plot, where the y-axis is a measure of sensitivity and the x-axis is the number of false-marks per patient. Sensitivity is the number of patients diagnosed as having the disease divided by the number of patients that has the disease. High sensitivity and low false-marks are desired. Our efforts to train the classifier for LungCAD were based initially on only certain data, and later by assuming that all the uncertain data points were positive. Figure 4 shows that by treating the uncertain data points as unlabeled, LIAM's performance is superior to both these approaches (LIAM's ROC curve clearly dominates the other two methods), especially in the region of interest of the ROC curve that is around 2 and 3 false positives per image. Further, because LIAM is an inductive algorithm, it has the same run-time performance as the other two classifiers (whereas, the transductive classifiers could not meet the run-time requirements).

It is also important to note that because of the size of the training data (around 20000 data points), inverting or calculating eigenvalues on the matrix $\tilde{L}$ as is needed in LGRF would be computationally very demanding and probably not feasible for CAD applications.

7 Conclusion

We have introduced a new inductive semi-supervised classifier, that in contrast with most of the recently developed semi-supervised techniques (that rely only in the smoothness assumption), LIAM combines smoothness and local class similarity constraints, which allowed LIAM to perform better compared to the smoothness assumption alone on few labeled examples. LIAM is faster than transductive algorithms with respect to testing time. Proximal LIAM presented in section 5.3 results in an unconstrained quadratic problem for which solutions can be obtained by solving a simple system of linear equations, which makes LIAM also fast in training. Another advantage of the proximal formulation is that it can be modified to efficiently solve incremental classification problems like in [10]. Experimental results confirm that LIAM is faster than TSVM and LGRF in terms of training time. Furthermore, the empirical evidence suggests that LIAM is more accurate than TSVM, LGRF and standard SVM, making our proposed algorithm a choice to consider when solving semi-supervised classification problems. One of the drawbacks of the formulations presented here is the need to tune three parameters, however we are currently working on a technique to automatically tune the parameters that shows very encouraging results. For future work, we also plan to extend LIAM to multi-class semi-supervised learning problems. We want to also explore the possibility of extending LIAM to an active learning setting.

8 Acknowledgments

The authors thank Balaji Krishnapuram for providing the code for LGRF and for discussions about semi-supervised learning. Special thanks go to Liam Fung-Jenkins for inspiring us and for providing a name to our approach. This research is supported by the Computer Aided Diagnosis Group, Siemens Medical Solutions, USA and NSF CAREER IIS-0347532.

References

[1] Belkin, Matveeva, and Niyogi. Regularization and semi-supervised learning on large graphs. In *Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann*, 2004.

[2] K. P. Bennett and A. Demiriz. Semi-supervised support vector machines. In *Advances in Neural Information Processing Systems -10-*, pages 368–374. MIT Press, 1998.

[3] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proc. 18th Int. Conf. on Machine Learning*, pages 19–26. Morgan Kaufmann, 2001.

[4] P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In J. Shavlik, editor, *Machine Learning Proceedings of (ICML '98)*, pages 82–90, San Francisco, California, 1998. Morgan Kaufmann.

[5] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Trading convexity for scalability. In *ICML '06 Proceedings*, pages 201–208, 2006.

[6] A. Corduneanu and T. Jaakkola. On information regularization. In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, 2003.

[7] A. Corduneanu and T. Jaakkola. Distributed information regularization on graphs. In *Advances in Neural Information Processing Systems*, 2004.

[8] G. Fung and O. L. Mangasarian. Proximal support vector machine classifiers. In *Knowledge Discovery and Data Mining*, pages 77–86, 2001.

[9] G. Fung and O. L. Mangasarian. Semi-supervised support vector machines for unlabeled data classification. *Optimization Methods and Software*, 15:29–44, 2001.

[10] G. Fung and O. L. Mangasarian. Incremental support vector machine classification. In H. M. R. Grossman and R. Motwani, editors, *Proceedings of SIAM International Conference on Data Mining*, 2002.

[11] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of ICML 1999*, pages 200–209. Morgan Kaufmann, 1999.

[12] B. Krishnapuram, D. Williams, Y. Xue, A. Hartemink, L. Carin, and M. Figueiredo. On semi-supervised classification. In *NIPS 17*, pages 721–728. MIT Press, 2005.

[13] O. L. Mangasarian. Generalized support vector machines. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 135–146, Cambridge, MA, 2000. MIT Press.

[14] D. Newman, S. Hettich, C. Blake, and C. Merz. UCI repository of machine learning databases, 1998.

[15] M. Seeger. Learning with labeled and unlabeled data. Technical report, Institute for ANC, Edinburgh, UK, 2000.

[16] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *Proceedings of the International Conference on Machine Learning*, pages 824–831, 2005.

[17] A. Smola and R. Kondor. Kernels and regularization on graphs. In *Learning Theory and Kernel Machines, Berlin - Heidelberg, Germany, 2003. Springer Verlag.*, 2003.

[18] M. Szummer and T. Jaakkola. Kernel expansions with unlabeled examples. In *Advances in Neural Information Processing Systems*, volume 13, pages 626–632, 2001.

[19] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.

[20] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems.* MIT Press, 2003.

[21] D. Zhou and B. Scholkopf. Learning from labeled and unlabeled data using random walks. In *German Pattern Recognition Symposium*, pages 237–244, 2004.

[22] X. Zhu. Semi-supervised learning literature survey. Technical Report tr-1530, Computer Sciences Department,University of Wisconsin Madison, 2006.

[23] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. 20th Int. Conf. on Machine Learning*, pages 912–919, 2003.