

On Manifold Regularization

Mikhail Belkin
Department of Computer Science

Partha Niyogi
Departments of Computer Science and Statistics

Vikas Sindhwani
Department of Computer Science

The University of Chicago
Chicago, IL 60637
{misha, niyogi, vikass}@cs.uchicago.edu

June 19, 2004

Abstract

We propose a family of learning algorithms based on a new form of regularization which allows us to incorporate both labeled and unlabeled data in a general-purpose learner. Transductive graph learning algorithms and standard methods including SVM and Regularized Least Squares can be obtained as special cases of our framework.

1 Introduction

The problem of learning from labeled and unlabeled data (*semi-supervised* and *transductive* learning) has attracted considerable attention in recent years (cf. [9, 5, 6, 13, 17, 18]). In this paper, we consider this problem within a new framework for data-dependent regularization.

The idea of regularization has a rich mathematical history going back to [15], where it is used for solving ill-posed inverse problems. Regularization is a key idea in the theory of splines (e.g., [16]) and has been used in machine learning (e.g., [8]). Many machine learning algorithms, including Support Vector Machines, can be interpreted as instances of regularization.

Our framework exploits the geometry of the probability distribution that generates the data and incorporates it as an additional regularization term. There are two regularization terms — one controlling the complexity of the classifier in the *ambient space* and the other controlling the complexity as measured by the *geometry* of the distribution. We consider in some detail the special case where this probability distribution is supported on a submanifold of the ambient space.

Within this general framework, we propose two specific families of algorithms: the Laplacian Regularized Least Squares (hereafter LapRLS) and the Laplacian Support Vector Machines (hereafter LapSVM). These are natural extensions of RLS and SVM respectively. In addition, several recently proposed transductive methods (e.g., [18, 1]) are also seen to be special cases of this general approach. Our solution for the semi-supervised case can be expressed as an expansion over labeled and unlabeled data points. Finally, it is worth noting that the problem of out-of-sample extension (see also [4]) is naturally resolved in our setting.

This paper provides a short account of the framework, theorems, algorithms, and experiments. A more extensive account with proofs and more detailed elaborations of algorithms is contained in a forthcoming longer paper that will be available as a technical report from The University of Chicago (Computer Science Department).

2 The Semi-Supervised Learning Framework

Recall the standard framework of learning from examples. There is a probability distribution P on $X \times \mathbb{R}$ according to which examples are generated for function learning. Labeled examples are (x, y) pairs generated according to P . Unlabeled examples are simply $x \in X$ drawn according to the marginal distribution \mathcal{P}_X of P .

One might hope that knowledge of the marginal \mathcal{P}_X can be exploited for better function learning (e.g. in classification or regression tasks). Of course, if there is no identifiable relation between \mathcal{P}_X and the conditional $\mathcal{P}(y|x)$, the knowledge of \mathcal{P}_X is unlikely to be of much use. Therefore, we will make a specific assumption about the connection between the marginal and the conditional. We will assume that if two points $x_1, x_2 \in X$ are *close* in the *intrinsic* geometry of \mathcal{P}_X , then the conditional distributions $\mathcal{P}(y|x_1)$ and $\mathcal{P}(y|x_2)$ are similar. In other words, the conditional probability distribution $\mathcal{P}(y|x)$ varies smoothly along the geodesics in the intrinsic geometry of \mathcal{P}_X .

We utilize these geometric ideas to extend an established framework for

function learning. A number of popular algorithms such as SVM, Ridge regression, splines, Radial Basis Functions may be broadly interpreted as regularization algorithms with different empirical cost functions and complexity measures in an appropriately chosen Reproducing Kernel Hilbert Space (RKHS).

For a Mercer kernel $K : X \times X \rightarrow \mathbb{R}$, there is an associated RKHS \mathcal{H}_K of functions $X \rightarrow \mathbb{R}$ with the corresponding norm $\|\cdot\|_K$. Given a set of labeled examples $(x_i, y_i), i = 1, \dots, l$ the standard framework estimates an unknown function by minimizing

$$f^* = \operatorname{argmin}_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^l V(x_i, y_i, f) + \gamma \|f\|_K^2 \quad (1)$$

where V is some loss function, such as squared loss $(y_i - f(x_i))^2$ for RLS or the soft margin loss function for SVM. Penalizing the RKHS norm imposes smoothness conditions on possible solutions. The classical Representer Theorem states that the solution to this minimization problem exists in \mathcal{H}_K and can be written as

$$f^*(x) = \sum_{i=1}^l \alpha_i K(x_i, x) \quad (2)$$

Therefore, the problem is reduced to optimizing over the finite dimensional space of coefficients α_i , which is the algorithmic basis for SVM, Regularized Least Squares and other regression and classification schemes.

Our goal is to extend this framework by incorporating additional information about the geometric structure of the marginal \mathcal{P}_X . We would like to ensure that the solution is smooth with respect to both the ambient space and the marginal distribution \mathcal{P}_X . To achieve that, we introduce an additional regularizer:

$$f^* = \operatorname{argmin}_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^l V(x_i, y_i, f) + \gamma_A \|f\|_K^2 + \gamma_I \|f\|_I^2 \quad (3)$$

where $\|f\|_I^2$ is an appropriate penalty term that should reflect the intrinsic structure of \mathcal{P}_X . Here γ_A controls the complexity of the function in the *ambient* space while γ_I controls the complexity of the function in the *intrinsic* geometry of \mathcal{P}_X . Given this setup one can prove the following representer theorem:

Theorem 2.1. Assume that the penalty term $\|f\|_I$ is sufficiently smooth with respect to the RKHS norm $\|f\|_K$. Then the solution f^* to the optimization problem in Eqn 3 above exists and admits the following representation

$$f^*(x) = \int_{\mathcal{M}} \alpha(y)K(x, y) d\mathcal{P}_X(y) + \sum_{i=1}^l \alpha_i K(x_i, x) \quad (4)$$

where $\mathcal{M} = \text{supp}\{\mathcal{P}_X\}$ is the support of the marginal \mathcal{P}_X .

To get a sense of why the theorem is true, let S be the closure of $\text{span}\{K_x \mid x \in \mathcal{M}\}$ in \mathcal{H}_K . For any $f \in \mathcal{H}_K$, its projection f_S to S satisfies (i) $\|f\|_I = \|f_S\|_I$ (ii) $f(x_i) = f_S(x_i)$ and (iii) $\|f\|_K \geq \|f_S\|_K$. Therefore it is easy to see that $f^* \in S$. We observe that functions as in right-hand side of Eqn 4 lie in S . Additional steps are required to show that f^* has that form.

However, in most applications we do not know \mathcal{P}_X . Therefore we must attempt to get empirical estimates of $\|f\|_I$. Note that in order to get such empirical estimates it is sufficient to have *unlabeled* examples.

A case of particular recent interest (e.g., see [11, 14, 2, 7] for a discussion on dimensionality reduction) is when the support of \mathcal{P}_X is a compact submanifold $\mathcal{M} \subset X = \mathbb{R}^n$. In that case, a natural choice for $\|f\|_I$ is $\int_{\mathcal{M}} \langle \nabla_{\mathcal{M}} f, \nabla_{\mathcal{M}} f \rangle$. The optimization problem becomes

$$f^* = \underset{f \in \mathcal{H}_K}{\text{argmin}} \frac{1}{l} \sum_{i=1}^l V(x_i, y_i, f) + \gamma_A \|f\|_K^2 + \gamma_I \int_{\mathcal{M}} \langle \nabla_{\mathcal{M}} f, \nabla_{\mathcal{M}} f \rangle$$

The term $\int_{\mathcal{M}} \langle \nabla_{\mathcal{M}} f, \nabla_{\mathcal{M}} f \rangle$ may be approximated on the basis of labeled and unlabeled data using the graph Laplacian ([1]). Thus, given a set of l labeled examples $\{(x_i, y_i)\}_{i=1}^l$ and a set of u unlabeled examples $\{x_j\}_{j=l+1}^{l+u}$, we consider the following optimization problem :

$$\begin{aligned} f^* &= \underset{f \in \mathcal{H}_K}{\text{argmin}} \frac{1}{l} \sum_{i=1}^l V(x_i, y_i, f) + \gamma_A \|f\|_K^2 + \frac{\gamma_I}{(u+l)^2} \sum_{i,j=1}^{l+u} (f(x_i) - f(x_j))^2 W_{ij} \\ &= \underset{f \in \mathcal{H}_K}{\text{argmin}} \frac{1}{l} \sum_{i=1}^l V(x_i, y_i, f) + \gamma_A \|f\|_K^2 + \frac{\gamma_I}{(u+l)^2} \hat{f}^T L \hat{f} \end{aligned} \quad (5)$$

where W_{ij} are edge weights in the data adjacency graph, $\hat{f} = [f(x_1), \dots, f(x_{l+u})]^T$, and L is the graph Laplacian given by $L = D - W$. Here, the diagonal matrix D is given by $D_{ii} = \sum_{j=1}^{l+u} W_{ij}$. The normalizing coefficient $\frac{1}{(u+l)^2}$ is the

natural scale factor for the empirical estimate of the Laplace operator. On a sparse adjacency graph it may be replaced by $\sum_{i,j=1}^{l+u} W_{ij}$.

The following simple version of the representer theorem shows that the minimizer has an expansion in terms of both labeled and unlabeled examples and is a key to our algorithms.

Theorem 2.2. *The minimizer of optimization problem 5 admits an expansion*

$$f^*(x) = \sum_{i=1}^{l+u} \alpha_i K(x_i, x) \quad (6)$$

in terms of the labeled and unlabeled examples.

The proof is a variation of the standard orthogonality argument.

Remarks : (a) Other natural choices of $\|\cdot\|_I$ exist. Examples are (i) heat kernel (ii) iterated Laplacian (iii) kernels in geodesic coordinates. The above kernels are geodesic analogs of similar kernels in Euclidean space. (b) Note that K restricted to \mathcal{M} (denoted by $K_{\mathcal{M}}$) is also a kernel defined on \mathcal{M} with an associated RKHS $\mathcal{H}_{\mathcal{M}}$ of functions $\mathcal{M} \rightarrow \mathbb{R}$. While this might suggest $\|f\|_I = \|f|_{\mathcal{M}}\|_{K_{\mathcal{M}}}$ ($f|_{\mathcal{M}}$ is f restricted to \mathcal{M}) as a reasonable choice for $\|f\|_I$, it turns out, that for the minimizer f^* of the corresponding optimization problem we get $\|f^*\|_I = \|f^*\|_K$, yielding the same solution as standard regularization, although with a different γ .

3 Algorithms

We now present solutions to the optimization problem posed in Eqn (5). To fix notation, we assume we have l labeled examples $\{(x_i, y_i)\}_{i=1}^l$ and u unlabeled examples $\{x_j\}_{j=l+1}^{l+u}$. We use K interchangeably to denote the kernel function or the Gram matrix.

3.1 Laplacian Regularized Least Squares (LapRLS)

The Laplacian Regularized Least Squares algorithm solves Eqn (5) with the squared loss function: $V(x_i, y_i, f) = (y_i - f(x_i))^2$. Since the solution is of the form given by (6), the objective function can be reduced to a convex differentiable function of the $(l + u)$ -dimensional expansion coefficient vector $\alpha = [\alpha_1, \dots, \alpha_{l+u}]^T$:

$$\alpha^* = \operatorname{argmin}_{\alpha \in \mathbb{R}^{l+u}} \frac{1}{l} (Y - K\alpha)^T J (Y - K\alpha) + \gamma_A \alpha^T K \alpha + \frac{\gamma_I}{(u+l)^2} \alpha^T K L K \alpha$$

where K is the $(l + u) \times (l + u)$ Gram matrix over labeled and unlabeled points; Y is an $(l+u)$ dimensional label vector given by $Y = [y_1, \dots, y_l, 0, \dots, 0]$ and J is an $(l+u) \times (l+u)$ diagonal matrix given by $J = \text{diag}(1, \dots, 1, 0, \dots, 0)$ with the first l diagonal entries as 1 and the rest 0. The minimizer can be easily obtained by solving the linear system.

$$\alpha^* = (JK + \gamma_A I + \frac{\gamma_I}{(u+l)^2} LK)^{-1} Y \quad (7)$$

Note that when $\gamma_I = 0$, Eqn (7) gives zero coefficients over unlabeled data. The coefficients over labeled data are exactly those for standard RLS.

3.2 Laplacian Support Vector Machines (LapSVM)

For standard SVM classification, the optimization problem 1 is solved with the soft margin loss function $V(x_i, y_i, f) = \max(0, 1 - y_i f(x_i))$, $y_i \in \{-1, +1\}$. Introducing slack variables, using standard Lagrange Multiplier techniques, and the form of the solution given by 2, we can arrive at the following quadratic program in dual variables β :

$$\beta^* = \max_{\beta \in \mathbb{R}^l} \sum_{i=1}^l \beta_i - \frac{1}{2} \beta^T Q \beta \quad \text{subject to:} \quad \sum_{i=1}^l y_i \beta_i = 0, \quad 0 \leq \beta_i \leq \frac{1}{l} \quad i = 1, \dots, l$$

where $Q = Y(\frac{K}{2\gamma})Y$, Y is the diagonal matrix $Y_{ii} = y_i$ and K is the Gram matrix over the labeled data. The optimal expansion coefficient vector can be obtained by the relation $\alpha^* = \frac{Y\beta^*}{2\gamma}$. SVM practitioners may be more familiar with a re-parametrized formulation that uses a C parameter as the weight on the hinge loss. In this formulation, C appears as an upper bound on β_i in the quadratic program; the relationship between the expansion coefficients and the Lagrange Multipliers is simpler: $\alpha^* = Y\beta^*$ and $Q = YKY$. The two parameterizations are related by $C = \frac{1}{2\gamma l}$.

LapSVM solves the optimization problem 5 with the soft margin loss function. Utilizing the same techniques used for deriving SVM, we can arrive at a similar quadratic program in l variables with

$$Q = YJK(2\gamma_A I + 2\frac{\gamma_I}{(u+l)^2} LK)^{-1} J^T Y \quad (8)$$

where Y , as before, is the diagonal matrix $Y_{ii} = y_i$, K is the Gram matrix over both the labeled and the unlabeled data; L is the data adjacency graph Laplacian; and J is an $l \times (l + u)$ matrix given by $J_{ij} = 1$ if $i = j$ and x_i is a labeled example, and $J_{ij} = 0$ otherwise.

To obtain the optimal expansion coefficient vector $\alpha^* \in \mathbb{R}^{(l+u)}$, one has to solve the following linear system :

$$\alpha^* = (2\gamma_A I + 2\frac{\gamma_I}{(u+l)^2} LK)^{-1} J^T Y \beta^* \quad (9)$$

Note that when $\gamma_I = 0$, the SVM QP and Eqns (8,9), give zero expansion coefficients over the unlabeled data. The expansion coefficients over the labeled data and the Q matrix are as in standard SVM, in this case.

The Manifold Regularization algorithms and some connections are presented in the table below. For Graph Regularization and Label Propagation see [10, 3, 18].

<i>Manifold Regularization Algorithms</i>	
Input:	l labeled examples $\{(x_i, y_i)\}_{i=1}^l$, u unlabeled examples $\{x_j\}_{j=l+1}^{l+u}$
Output:	Estimated function $f : \mathbb{R}^n \rightarrow \mathbb{R}$
Step 1	► Construct data adjacency graph with $(l + u)$ nodes using, e.g, k nearest neighbors. Choose edge weights W_{ij} , e.g. binary weights or heat kernel weights $W_{ij} = e^{-\ x_i - x_j\ ^2 / 4t}$.
Step 2	► Choose a kernel function $K(x, y)$. Compute the Gram matrix $K_{ij} = K(x_i, x_j)$.
Step 3	► Compute graph Laplacian matrix : $L = D - W$ where D is a diagonal matrix given by $D_{ii} = \sum_{j=1}^{l+u} W_{ij}$.
Step 4	► Choose γ_A and γ_I .
Step 5	► Compute α^* using Eqn (7) for squared loss (Laplacian RLS) or using Eqns (8,9) together with the SVM QP solver for soft margin loss (Laplacian SVM).
Step 6	► Output function $f^*(x) = \sum_{i=1}^{l+u} \alpha_i^* K(x_i, x)$.
Connections to other algorithms	
$\gamma_A \geq 0 \quad \gamma_I \geq 0$	<i>Manifold Regularization</i>
$\gamma_A \geq 0 \quad \gamma_I = 0$	<i>Standard Regularization (RLS or SVM)</i>
$\gamma_A = 0 \quad \gamma_I > 0$	<i>Out-of-sample extension for Graph Regularization (RLS or SVM)</i>
$\gamma_A = 0 \quad \gamma_I \rightarrow 0$	<i>Out-of-sample extension for Label Propagation (RLS or SVM)</i>
$\gamma_A \rightarrow 0 \quad \gamma_I = 0$	<i>Hard margin (RLS or SVM)</i>

4 Experiments

We performed experiments on a synthetic dataset and two real world classification problems arising in visual and speech recognition. Comparisons are made with inductive methods (SVM, RLS) and Transductive SVM (e.g., [9]). All software and datasets used for these experiments will be made available at:

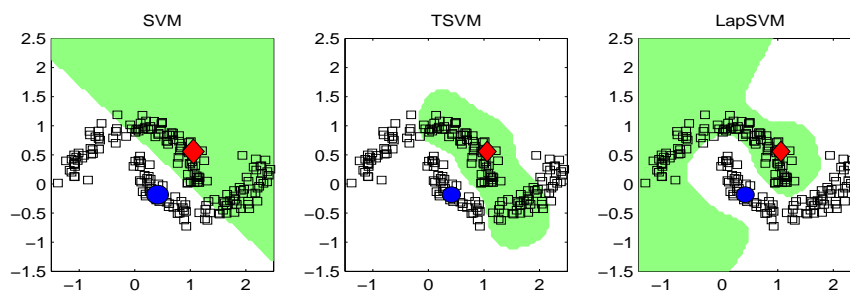
http://manifold.cs.uchicago.edu/manifold_regularization/manifold.html.

For all experiments, we constructed adjacency graphs with 6 nearest neighbors.

4.1 Synthetic Data : Two Moons Dataset

The two moons dataset is shown in Figure 1. The best decision surfaces across a wide range of parameter settings are also shown for SVM, Transductive SVM and Laplacian SVM. The dataset contains 200 examples with only 1 labeled example for each class. The SVM solution is fixed by the location of the two labeled points. Transductive SVM use the inductive SVM to label the unlabeled data and then iteratively solve SVM quadratic programs, at each step switching labels so that the margin improves. Figure 1 demonstrates how TSVM fails to find the optimal solution. The Laplacian SVM decision boundary seems to be intuitively most satisfying.

Figure 1: Two Moons Dataset: Best decision surfaces using RBF kernels for SVM, TSVM and Laplacian SVM. Labeled points are shown in color, other points are unlabeled.



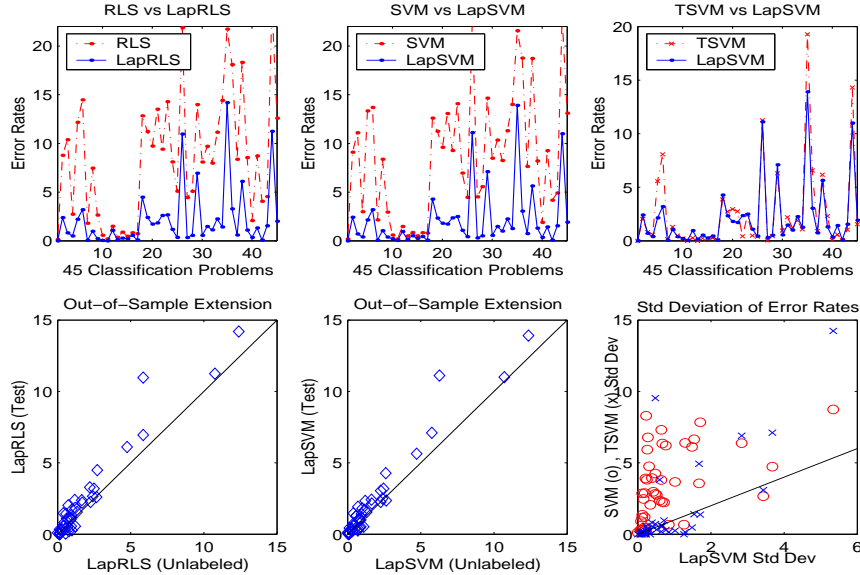
4.2 Handwritten Digit Recognition

In this set of experiments we applied Laplacian SVM and Laplacian RLSC algorithms to 45 binary classification problems that arise in pairwise classification of handwritten digits. The first 400 images for each digit in the USPS training set (preprocessed using PCA to 100 dimensions) were taken to form the training set. The remaining images formed the test set. 2 images for each class were randomly labeled ($l=2$) and the rest were left unlabeled ($u=398$). Following [12], we chose to train classifiers with polynomial kernels of degree 3, and set the weight on the regularization term for inductive methods as $\gamma l = 0.05 (C = 10)$. For manifold regularization, we chose to split the same weight in the ratio 1 : 9 so that $\gamma_{Al} = 0.005$, $\frac{\gamma l}{(u+l)^2} = 0.045$. The observations reported in this section hold consistently across a wide choice of parameters. In Figure 2, we compare the error rates of Laplacian algorithms, SVM and TSVM, at the precision-recall breakeven points in the ROC curves for the 45 binary classification problems. These results are averaged over 10 random choices of labeled examples. The following comments can be made: (a) Manifold regularization results in significant improvements over inductive classification, for both RLS and SVM, and either compares well or significantly outperforms TSVM across the 45 classification problems. Note that TSVM solves multiple quadratic programs in the size of the labeled and unlabeled sets whereas LapSVM solves a single QP in the size of the labeled set, followed by a linear system. This resulted in substantially faster training times for LapSVM in this experiment. (b) Scatter plots of performance on test and unlabeled data sets confirm that the out-of-sample extension is good for both LapRLS and LapSVM. (c) Finally, we found Laplacian algorithms to be significantly more stable with respect to choice of the labeled data than the inductive methods and TSVM, as shown in the scatter plot in Figure 2 on standard deviation of error rates.

4.3 Spoken Letter Recognition

This experiment was performed on the Isolet database of letters of the English alphabet spoken in isolation (available from the UCI machine learning repository). The data set contains utterances of 150 subjects who spoke the name of each letter of the English alphabet twice. The speakers are grouped into 5 sets of 30 speakers each, referred to as isolet1 through isolet5. For the purposes of this experiment, we chose to train on the first 30 speakers (isolet1) forming a training set of 1560 examples, and test on isolet5 containing 1559 examples (1 utterance is missing in the database due to poor record-

Figure 2: USPS Experiment - Error Rates at Precision-Recall Breakeven points for 45 binary classification problems



ing). We considered the task of classifying the first 13 letters of the English alphabet from the last 13. The experimental set-up is meant to simulate a real-world situation: we considered 30 binary classification problems corresponding to 30 splits of the training data where all 52 utterances of one speaker were labeled and all the rest were left unlabeled. The test set is composed of entirely new speakers, forming the separate group isolet5.

We chose to train with RBF kernels of width $\sigma = 10$ (this was the best value among several settings with respect to 5-fold cross-validation error rates for the fully supervised problem using standard SVM). For SVM and RLS we set $\gamma l = 0.05$ ($C = 10$) (this was the best value among several settings with respect to mean error rates over the 30 splits). For Laplacian RLS and Laplacian SVM we set $\gamma_{Al} = \frac{\gamma l}{(u+l)^2} = 0.005$. In Figure 3, we compare these algorithms. The following comments can be made: (a) LapSVM and LapRLS make significant performance improvements over inductive methods and TSVM, for predictions on unlabeled speakers that come from the same group as the labeled speaker, over all choices of the labeled speaker. (b) On Isolet5 which comprises of a separate group of speakers, performance improvements are smaller but consistent over the choice of the la-

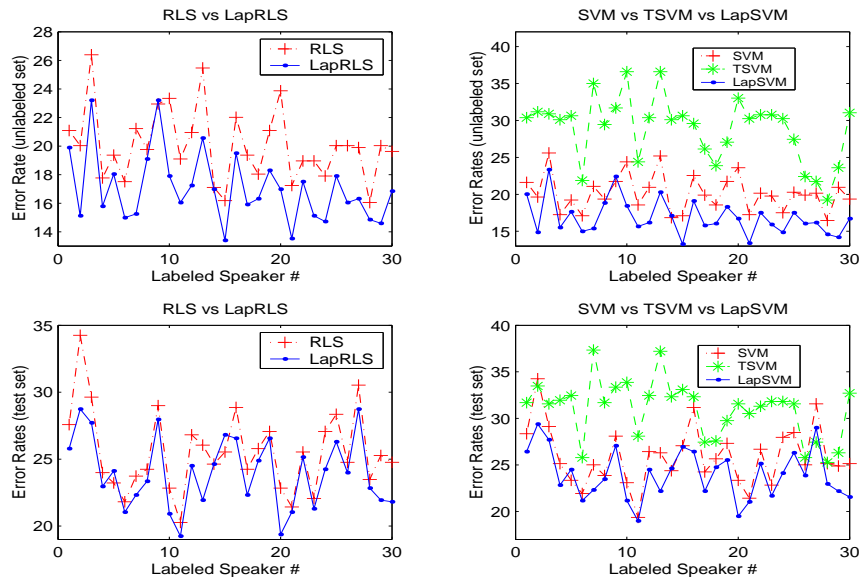


Figure 3: Isolet Experiment - Error Rates at precision-recall breakeven points 30 binary classification problems

beled speaker. This can be expected since there appears to be a systematic bias that affects all algorithms, in favor of same-group speakers.

Acknowledgments. We are grateful to Marc Coram, Steve Smale and Peter Bickel for intellectual support and to NSF funding for financial support.

References

- [1] M. Belkin, P. Niyogi, *Using Manifold Structure for Partially Labeled Classification*, NIPS 2002.
- [2] M. Belkin, P. Niyogi. (2003). *Laplacian Eigenmaps for Dimensionality Reduction and Data Representation*, Neural Computation, Vol. 15, No. 6, 1373-1396.
- [3] M. Belkin, I. Matveeva, P. Niyogi, *Regression and Regularization on Large Graphs*, COLT 2004.
- [4] Y. Bengio, O. Delalleau and N. Le Roux, *Efficient Non-Parametric Function Induction in Semi-Supervised Learning*, Technical Report 1247, DIRO, University of Montreal, 2004.

- [5] A. Blum, S. Chawla, *Learning from Labeled and Unlabeled Data using Graph Min-cuts*, ICML 2001.
- [6] Chapelle, O., J. Weston and B. Schoelkopf, *Cluster Kernels for Semi-Supervised Learning*, NIPS 2002.
- [7] D. L. Donoho, C. E. Grimes, *Hessian Eigenmaps: new locally linear embedding techniques for high-dimensional data*, Proceedings of the National Academy of Arts and Sciences vol. 100 pp. 5591-5596.
- [8] T. Evgeniou, M. Pontil and T. Poggio, *Regularization Networks and Support Vector Machines*, Advances in Computational Mathematics, Vol. 13, 1-50, 2000.
- [9] T. Joachims, *Transductive Inference for Text Classification using Support Vector Machines*, ICML 1999.
- [10] A. Smola and R. Kondor, *Kernels and Regularization on Graphs*, COLT/KW 2003.
- [11] Sam T. Roweis, Lawrence K. Saul. (2000). *Nonlinear Dimensionality Reduction by Locally Linear Embedding*, Science, vol 290.
- [12] B. Schoelkopf, C.J.C. Burges, V. Vapnik, *Extracting Support Data for a Given Task*, KDD95.
- [13] Martin Szummer, Tommi Jaakkola, *Partially labeled classification with Markov random walks*, NIPS 2001.
- [14] J.B.Tenenbaum, V. de Silva, J. C. Langford. (2000). *A Global Geometric Framework for Nonlinear Dimensionality Reduction*, Science, Vol 290.
- [15] A.N. Tikhonov, *Regularization of Incorrectly Posed Problems*, Soviet Math. Doklady 4, 1963 (English Translation).
- [16] G. Wahba. (1990). *Spline Models for Observational Data*, Society for Industrial and Applied Mathematics.
- [17] D. Zhou, O. Bousquet, T.N. Lal, J. Weston and B. Schoelkopf, *Learning with Local and Global Consistency*, NIPS 2003.
- [18] X. Zhu, J. Lafferty and Z. Ghahramani, *Semi-supervised learning using Gaussian fields and harmonic functions*, ICML 2003.