Technical Report No. 141, Max Planck Institute for Biological Cybernetics

Semi-supervised Induction with Basis Functions

Kai Yu, Volker Tresp
Corporate Technology, Siemens AG
Otto-Hahn-Ring 6, 81739, Munich, Germany
KAI.YU, VOLKER.TRESP@SIEMENS.COM

Dengyong Zhou
Max Planck Institute for Biological Cybernetics,
Spemann str. 38, 72076 Tübingen
DENGYONG.ZHOU@TUEBINGEN.MPG.DE

June 12, 2004

Abstract

Considerable progress was recently made on semi-supervised learning, which differs from the traditional supervised learning by additionally exploring the information of the unlabeled examples. However, a disadvantage of many existing methods is that it does not generalize to unseen inputs. This paper suggests a space of basis functions to perform semi-supervised inductive learning. As a nice property, the proposed method allows efficient training and can easily handle new test points. We validate the method based on both toy data and real world data sets.

1. Introduction

Recent years have seen considerable attention on semi-supervised learning, which differs from traditional supervised learning by making use of *unlabeled* data. In many applications, like text categorization, collecting labeled examples costs human efforts, while vast amounts of unlabeled data are often readily available and offer some additional information. This is the situation, in which semi-supervised learning becomes very useful. In the paradigm, the function of interest is regularized to be *a priori* consistent with the inherent structure of input density $p(\mathbf{x})$. Several advances were recently achieved, like Markov random walks (Szummer and Jaakkola, 2002), cluster kernels (Chapelle et al., 2003), Gaussian random fields (Zhu et al., 2003), and regularization on graphs (Belkin and Niyogi, 2003; Zhou et al., 2004).

So far most of the efforts have been invested in a transductive setting that predicts only for observed inputs. Yet, in many applications there is a clear need for inductive learning, for example, in hand-written zip code recognition or in document classification. Unfortunately, most existing semi-supervised learners do not readily generalize to new test data. A brute force approach is to incorporate the new test points and re-estimate the function using semi-

supervised learning, but this is very inefficient. Chapelle et al. (2003) suggest to approximate new test points with seen data points, which is however an indirect way. Another problem of semi-supervised transduction is the computational complexity. Since an $n \times n$ matrix needs either to be inverted (Zhu et al., 2003; Zhou et al., 2004) or digitalized (Chapelle et al., 2003; Belkin and Niyogi, 2003), semi-supervised transduction scaling as $O(n^3)$. As potentially a vast amount of unlabeled points are involved, the computational cost becomes prohibitive.

This paper extends the approach suggested in (Zhou et al., 2004) to realize a semi-supervised *inductive* learning method with $m \leq n$ finite basis functions (RBF function). The methods learn a function defined on the whole input space by solving a linear system of size m (Sec. 2). We then justify the adopted basis function expansion via the view of learning eigenfunctions. Finally we present results of an empirical study in Sec. 3.

2. Semi-supervised Function Induction

The general supervised learning problem considers a predictive function space $\mathcal{H} = \{f : \mathcal{X} \to \mathbb{R}\}$ defined on an input space $\mathcal{X} \subseteq \mathbb{R}^d$. In order to learn a function $f \in \mathcal{H}$ based on training data $\{\mathbf{x}_i, y_i\}_{i=1}^l$, where $\mathbf{x}_i$ are the inputs and y_i the measurements of outputs $f(\mathbf{x}_i)$, one solves the following problem

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \sum_{i=1}^{l} \left[y_i - f(\mathbf{x}_i)\right]^2 + \lambda \Omega(f) \tag{1}$$

where $\Omega : f \to \mathbb{R}^+ \cup 0$ measures the complexity of functions in $\mathcal{H}$. The first term of the cost function enforces f to explain well the observations. The second term, called the *regularizer*, ensures $f(\mathbf{x})$ to be sufficiently smooth, i.e. with low complexity. A common smoothness assumption behind the regularizer (e.g. in ridge regression and spline interpolation) states that, similar inputs should have similar function values. This assumption often reflects the learner's prior knowledge about what kind of functions are preferred *a priori*.

2.1 Graph-based Transduction

In the supervised setting, the notion of input similarity is usually independent to the distribution of input data $\mathbf{x}$. Instead, semi-supervised learning employs a stronger assumption that there may exist some relationship between the distribution $p(\mathbf{x})$ of inputs and the target function f. A well-known example is the so called "cluster assumption" (e.g. (Chapelle et al., 2003)): inputs $\mathbf{x}$ in the same cluster are likely to have similar function values $f(\mathbf{x})$. Therefore, in the situation where labeled data $\{\mathbf{x}_i, y_i\}_{i=1}^l$ are limited while vast amount of unlabeled data $\{\mathbf{x}_i\}_{i=l}^n$ exist[1], semi-supervised learning employs the input density, explored from both labeled and unlabeled inputs $\{\mathbf{x}_i\}_{i=1}^n$, as additional knowledge in learning the input-output relations.

Recently an elegant semi-supervised learning framework was introduced, which explores the unlabeled data via a *similarity graph* (e.g. (Zhu et al., 2003; Zhou et al., 2004)). Formally, let $G(\mathbf{V}, \mathbf{E}, \mathbf{W})$ be an undirected and weighted graph, which represents seen inputs $\mathbf{x}_i$

1. Without loss of generality, we set the first l data as labeled and the rest $n - l$ inputs unlabeled.

as vertices $v_i \in \mathbf{V}$, and the similarities as the weights $W_{ij} \in \mathbf{W}$ of edges $[v_i, v_j] \in \mathbf{E}$ connecting input pairs $(\mathbf{x}_i, \mathbf{x}_j)$, where $\mathbf{W}$ can be viewed as an $n \times n$ matrix, its entries W_{ij} should be nonnegative and symmetric (i.e. $W_{ij} = W_{ji}$), and additionally $W_{ii} = 0$ (to remove self similarities). In this paper we mainly apply the RBF function $W_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^{-2})$. If only interested in predictions on seen inputs, i.e. vertices V, then one considers a function space $\mathcal{G}\{\mathbf{f} : V \to \mathbb{R}\}$, $\mathcal{G} \subseteq \mathbb{R}^n$, and solves the following problem

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f} \in \mathcal{G}} \sum_{i=1}^{l} (y_i - f_i)^2 + \lambda \Omega(\mathbf{f}) \tag{2}$$

where $\mathbf{f} = [f_1, \ldots, f_n]^\top$, f_i is the function value on v_i, and $\Omega(\mathbf{f})$ is a complexity measure of $\mathbf{f}$, indicating how smooth $\mathbf{f}$ is with respect to the somehow the *geometric* structure of graph G. One reasonable assumption is that, function values should changes slowly over closely connected vertices. One way to realize this is to apply the *combinatorial Laplacian* (Zhu et al., 2003)

$$\Omega_c(\mathbf{f}) = \frac{1}{2} \sum_{i,j=1}^{n} W_{ij} (f_i - f_j)^2, \tag{3}$$

It is easy to see that, penalizing the quantity ensures $\mathbf{f}$ to vary slowly between strongly connected vertices. Another version of the regularizer was adopted in (Zhou et al., 2004):

$$\Omega_g(\mathbf{f}) = \frac{1}{2} \sum_{i,j=1}^{n} W_{ij} \left(\frac{f_i}{\sqrt{D_i}} - \frac{f_j}{\sqrt{D_j}} \right)^2, \tag{4}$$

where $D_i = \sum_j W_{ij}$ is the degree of v_i (analog to the density of $\mathbf{x}_i$). This regularizer also penalizes the functions that change rapidly across nearby vertices, but behaves somehow different to the former one, due to the normalization by degrees. Let $\mathbf{D}$ be the diagonal matrix with the diagonal entries as vertex degrees. Then it is not difficult to have $\Omega_c = \mathbf{f}^\top (\mathbf{D} - \mathbf{W}) \mathbf{f}$ and $\Omega_g = \mathbf{f}^\top (\mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}}) \mathbf{f}$. In particular, $\Delta = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}}$ is called *graph Laplacian* in spectral graph theory (Chung, 1997). In this paper, we mainly apply the regularizer in Eq. (4).

2.2 Inductive Generalization with Basis Functions

The regularization on graphs falls into a *transductive* setting, since the learned $\mathbf{f}$ only predicts those unlabeled inputs involved as vertices of graph G. However, in practice it is often required to do *induction* so that the learned function $f : \mathbf{X} \to \mathbb{R}$ can predict any new inputs. Therefore, we consider the class of approximating functions to be

$$f(\mathbf{x}) = \sum_{j=1}^{m} w_j \varphi_j(\mathbf{x}) = \varphi(\mathbf{x})^\top \mathbf{w} \tag{5}$$

where $\varphi(\mathbf{x}) = [\varphi_1(\mathbf{x}), \ldots, \varphi_m(\mathbf{x})]^\top$ are basis functions, and $\mathbf{w} = [w_1, \ldots, w_m]^\top$ the weights. In this paper we will only consider RBF basis functions $\varphi_j(\mathbf{x}) = \exp(-\|\mathbf{x} - \mathbf{x}_j\|^2 / 2\sigma_b^{-2})$, $\mathbf{x}_j \in \{\mathbf{x}_j\}_{j=1}^n$, centered on the whole set of seen inputs (then $m = n$) or a subset ($m < n$). In

general, f can be seen as a form of neural networks with a set of basis functions, meanwhile it can also be cast as a kernel machine since the basis function applied here are actually RBF kernels. Now let $\boldsymbol{\varphi}_n \in \mathbb{R}^{m \times n}$ be the matrix with $\{\boldsymbol{\varphi}_n\}_{ji} = \varphi_j(\mathbf{x}_i)$, i.e. basis functions evaluated on those seen inputs. By plugging $\mathbf{f} = \boldsymbol{\varphi}_n^\top \mathbf{w}$ into Eq. (2), we obtain a learning problem with the form

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{i=1}^{l} \left[y_i - \varphi(\mathbf{x}_i)^\top \mathbf{w} \right]^2 + \lambda \mathbf{w}^\top \boldsymbol{\varphi}_n \Delta \boldsymbol{\varphi}_n^\top \mathbf{w}. \tag{6}$$

By setting the derivatives of the cost function with respect to $\mathbf{w}$ to be zero, the optimal weights are estimated by

$$\hat{\mathbf{w}} = (\boldsymbol{\varphi}_l \boldsymbol{\varphi}_l^\top + \lambda \boldsymbol{\Gamma})^{-1} \boldsymbol{\varphi}_l \mathbf{y}_l, \tag{7}$$

where $\boldsymbol{\Gamma} = \boldsymbol{\varphi}_n \Delta \boldsymbol{\varphi}_n^\top$, $\mathbf{y}_l = [y_1, \ldots, y_l]^\top$, and $\boldsymbol{\varphi}_l \in \mathbb{R}^{m \times l}$ are the first l columns of $\boldsymbol{\varphi}_n$ corresponding to responses of basis functions on the labeled data. The approximated function is then given by

$$\hat{f}(\mathbf{x}) = \varphi(\mathbf{x})^\top \hat{\mathbf{w}} \tag{8}$$

The proposed method has certain advantages. First, it builds an inductive learner able to handle new inputs. The computation for prediction only scales linearly as $O(m)$, while transduction has to re-compute the predictor whenever new data arrive, which scales as $O(n^3)$ each time. Second, for training the algorithm inverts an $m \times m$ matrix $\boldsymbol{\varphi}_l \boldsymbol{\varphi}_l^\top + \lambda \boldsymbol{\Gamma}$, which can be much more efficient if $m \ll n$. Finally, nonlinear functions can be modeled since the basis functions are nonlinear.

2.3 Implicit Feature Mapping

It is well-known that the regularization framework for learning can be equivalently explained by an view of *implicit feature mapping* (see Girosi et al., 1995). The regularization on graphs can also be cast in this way, namely, the graph Laplacian induces a set of *eigenfunctions* and *eigenvalues* that project inputs into another vector space and then perform normal linear least square regression there. Accordingly, the key point to do induction is whether one can make the feature mappings defined not just on $\{\mathbf{x}_i\}_{i=1}^n$ but the whole input space $\mathcal{X}$.

Since the system contains m basis functions, there are accordingly m eigenfunctions $\phi_1(\mathbf{x}), \ldots, \phi_m(\mathbf{x})$, and each of them has the form

$$\phi_k(\mathbf{x}) = \sum_{j}^{m} w_{jk} \varphi_j(\mathbf{x}) = \varphi(\mathbf{x})^\top \mathbf{w}_k \quad \text{for} \quad k = 1, \ldots, m. \tag{9}$$

These eigenfunctions should be mutually orthogonal and satisfy $\int \phi_k(\mathbf{x})^2 p(\mathbf{x}) dx = 1$. The later condition is approximated by empirical averaging on i.i.d. data, giving rise to $\frac{1}{n} \boldsymbol{\phi}_k^\top \boldsymbol{\phi}_k = 1$, where $\boldsymbol{\phi}_k = [\phi_k(\mathbf{x}_1), \ldots, \phi_k(\mathbf{x}_n)]^\top$. The first eigenfunction should be the one with lowest complexity, thus its coefficients is obtained by

$$\mathbf{w}_1 = \arg\min_{\mathbf{w}} \quad \mathbf{w}^\top \boldsymbol{\varphi}_n \Delta \boldsymbol{\varphi}_n^\top \mathbf{w}, \quad \text{subject to:} \qquad \frac{1}{n} \mathbf{w}^\top \boldsymbol{\varphi}_n \boldsymbol{\varphi}_n^\top \mathbf{w} = 1 \tag{10}$$

The result is actually independent to the scaling factor $\frac{1}{n}$. The objective function at the optimum gives the corresponding eigenvalue λ_1. Though this just solves the first eigenfunction, the optimization problem's Lagrangian suggests a generalized eigenvalue problem, whose solutions gives all the eigenfunctions' coefficients and eigenvalues

$$(\boldsymbol{\varphi}_n \triangle \boldsymbol{\varphi}_n{}^\top)\mathbf{w}_k = \lambda_k(\boldsymbol{\varphi}_n \boldsymbol{\varphi}_n{}^\top)\mathbf{w}_k, \quad \text{for} \quad k = 1, \ldots, m \tag{11}$$

Those eigenfunctions present m orthogonal basis functions, corresponding to different levels of smoothness. Smaller eigenvalues means more smooth. For a function $f(\mathbf{x}) = \varphi(\mathbf{x})^\top \mathbf{w}$, its complexity is

$$\mathbf{w}^\top \boldsymbol{\varphi}_n \Delta \boldsymbol{\varphi}_n^\top \mathbf{w} = \sum_{k=1}^{m} \lambda_k \langle \boldsymbol{\varphi}_n^\top \mathbf{w}, \boldsymbol{\phi}_k \rangle = \sum_{k=1}^{m} \lambda_k c_k$$

where $\langle \cdot, \cdot \rangle$ is vector inner product. It is clear from above equation that minimizing the complexity of f enforces f close to the smooth eigenfunctions (i.e. those with smaller eigenvalues) as much as possible.

3. Empirical Study

3.1 Toy Data

We test the proposed algorithms on the two-moon toy problem (Zhou et al., 2004). As shown in Fig. 1, 120 inputs are generated from two underlaying classes and each class has only *one* labeled example. The performance of transduction has been shown in (Zhou et al., 2004), which predicts for only seen inputs. In contrast, the induction learns a function defined in the whole space and gives a classification boundary. We also estimate the eigenfunctions based on the two-moon data, using 120 RBF basis functions, and illustrate the 6 smoothest ones in Fig. 2. The eigenfunctions expose the the structure of input density in different resolutions, i.e. the first eigenfunction reflects the density of inputs, the second one exactly reflects the two different classes, the third one describes the isolated "island" in the density, and the following ones indicate more details, behaving like the Fourier transformation to describe signals in different frequency bands.

3.2 Digit Recognition

We test the performance of algorithms in a digit recognition task based on the USPS benchmark. We follow the setting in (Zhou et al., 2004) and pick up the digits 1, 2, 3, and 4, with a total of 3874 examples. As comparison, we also test support vector machines, as the baseline, and semi-supervised transduction described by (Zhou et al., 2004). We test induction learners with randomly selected m inputs to form RBF basis functions, where m is 100%, or 10% of seen inputs. The parameter λ in Eq. (6) is set to be 100, which corresponds to $w = 0.99$ in (Zhou et al., 2004). We split the data into *seen* (including labeled and unlabeled data) and *unseen* sets, 90% vs. 10%, and examine the predictive accuracy on the unseen set given a number of labeled examples in the seen set. For the transductive learner, each time we have to include one test point into the affinity matrix and then predict its label. Note it is unfair to include the whole "unseen" sets (to make

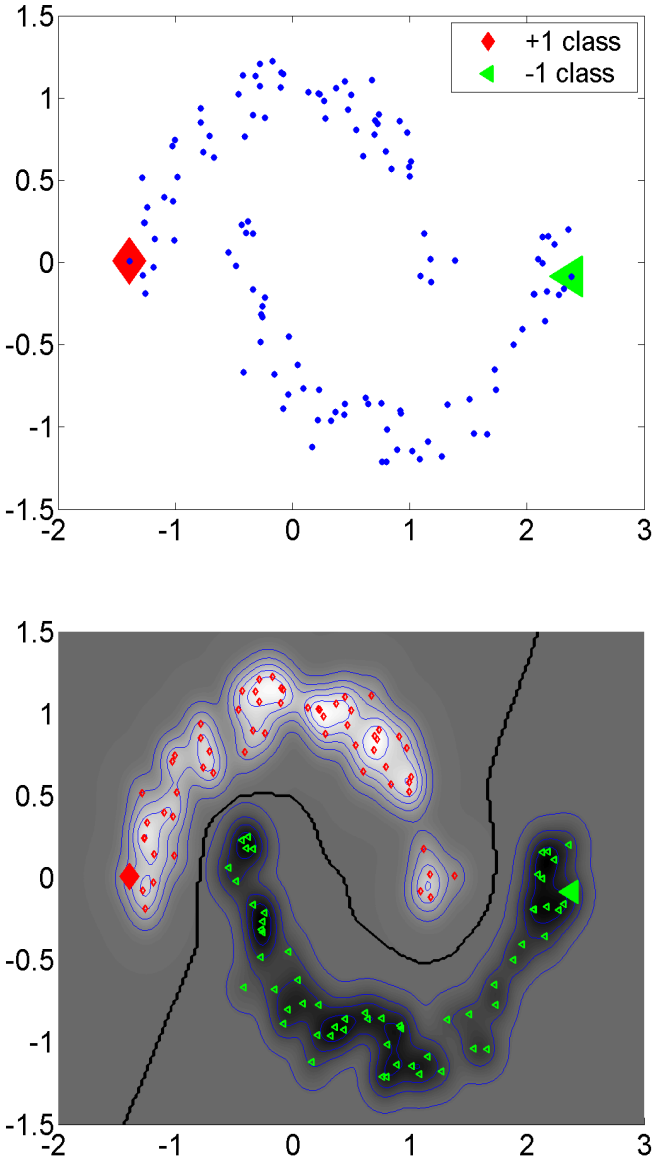

Figure 1: Semi-supervised induction on the two-moon data. Top:, each class has only one labeled example; Bottom, induction with 120 basis functions, the black bold curve gives the classification boundary and the gray level indicates the function value.

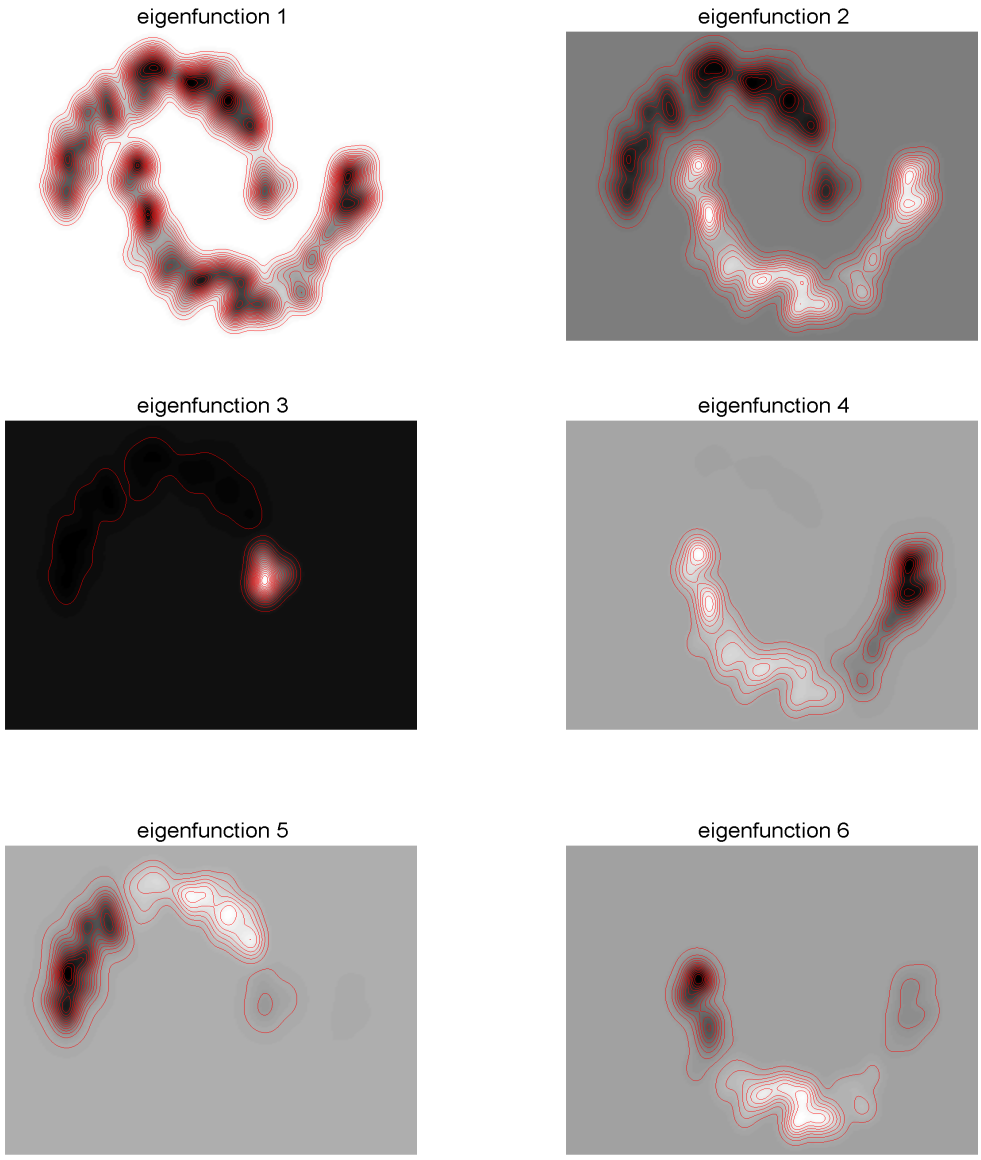

Figure 2: The six eigenfunctions with smallest eigenvalues, estimated with 120 basis functions. The eigenfunctions not only expose the structure of input density, but also help to understand semi-supervised learning: choosing the smooth eigenfunctions that also explain the labeled examples well, which gives the second eigenfunction in the case shown in Fig. 1.

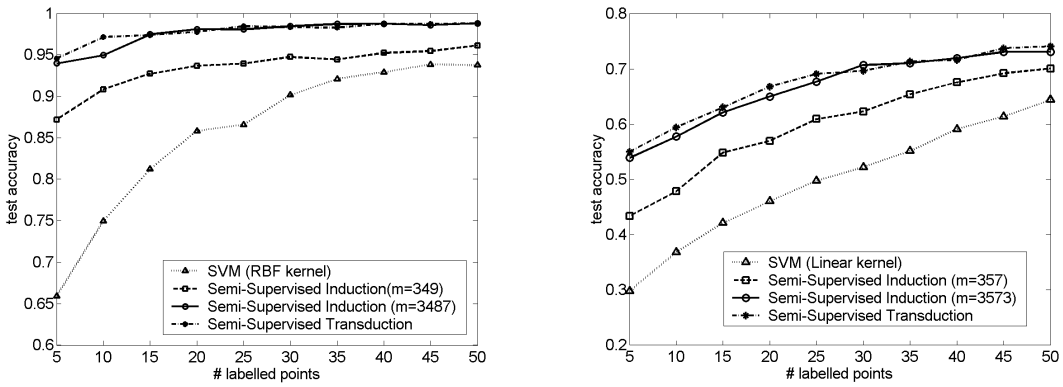

Figure 3: Left panel: Test results for digit recognition based on USPS data. Right panel: Test results for text categorization based on 20-newsgroup data.

computation cheaper) because then transduction has a much larger affinity matrix than induction. The setting makes the test computationally expensive, but highlights the point that induction can cheaply handle new test points. We repeat all the tests for 50 times, i.e. each time a different seen/unseen split and a different random set of m seen inputs for basis functions. As shown in Fig. 3-(a), the induction taking the whole seen set as basis functions gives the accuracy almost as excellent as transduction. The functions formed by 10% basis functions perform a bit worse than the tranductive learner but still much better than SVMs, and is computationally much cheaper than the transduction.

3.3 Text Categorization

In this experiment we test the algorithms for text categorization based on the 20-newsgroup data set. We take the same setting as in (Zhou et al., 2004), i.e. choosing the four topics *autos*, *motorcycles*, *baseball* and *hockey* and taking the same preprocessing steps to finally get 3970 TFIDF vectors. The distance between documents $d(\mathbf{x}_i, \mathbf{x}_j) = 1 - \langle \mathbf{x}_i, \mathbf{x}_j \rangle / \|\mathbf{x}_i\| \|\mathbf{x}_j\|$ is applied to form RBF functions for affinity matrix (with width 0.15), basis functions for induction (width 0.15) (Zhou et al., 2004). We then perform 50 trials with random 90% *seen* and 10% *unseen* split and report the average performance of each algorithm in Fig. 3-(b). We find that the induction with basis functions formed by 100% seen inputs (m=3573) performs very closely to the transduction learner. The computationally cheaper inductive learner with $m = 357$ basis functions trades off the accuracy, but still outperforms SVMs.

4. Conclusion

This paper realizes a semi-supervised *inductive* algorithm by extending previous graph-based transductive approaches. The idea is to use basis function expansion to form a regularizer induced by the normalized graph Laplacian. Finally the effectiveness of the proposed algorithm is illustrated on both toy problem and digit recognition. In the near future it is interesting to identify the optimal set of basis functions in this framework.

References

M. Belkin and P. Niyogi. Using manifold structure for partially labeled classification. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*. MIT Press, 2003.

O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*. MIT Press, 2003.

F. Chung. *Spectral Graph Theory*. Number 92 in Regional Conference Series in Mathematics. American Mathematical Society, 1997.

Federico Girosi, Michael Jones, and Tomaso Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, 1995. URL `citeseer.ist.psu.edu/girosi95regularization.html`.

M. Szummer and T. Jaakkola. Partially labeled classification with markov random walks. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.

D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In L. Saul Thrun and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.

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