Semi-Supervised Induction

Kai Yu[†]**, Volker Tresp**[†]**, Dengyong Zhou**[‡]
[†]Siemens Corporate Technology, 81730 Munich, Germany
kai.yu, volker.tresp@siemens.com
[‡]Max Planck Institute for Biological Cybernetics, 72076 Tübingen, Germany
dengyong.zhou@tuebingen.mpg.de

Abstract

Considerable progress was recently achieved on semi-supervised learning, which differs from the traditional supervised learning by additionally exploring the information of the unlabelled examples. However, a disadvantage of many existing methods is that it does not generalize to unseen inputs. This paper investigates learning methods that effectively make use of both labelled and unlabelled data to build predictive functions, which are defined on not just the seen inputs but the whole space. 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 *unlabelled* data. In many applications, like text categorization, collecting labelled examples costs human efforts, while vast amounts of unlabelled 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 $p(\boldsymbol{x})$. Several advances were recently achieved, like Markov random walks [8], cluster kernels [4], Gaussian random fields [11], and regularization on graphs [1, 10].

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. [4] 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 [11, 10] or diagonalized [4, 1], semi-supervised transduction scaling as $O(n^3)$. As potentially a vast amount of unlabelled points are involved, the computational cost becomes prohibitive.

This paper extends the approach suggested in [10] to realize a family of semi-supervised *inductive* learners with $m \leq n$ finite basis functions. The methods learn a function defined on the whole input space by solving only a linear system of size m (Sec. 2). In Sec. 3 we introduce the adopted regularizer induced by the *normalized graph Laplacian*, and connect its limit to the expected squared gradient of the function weighted by $p(\boldsymbol{x})$, giving rise to a natural *density-dependent* smoothness penalty. We then justify the adopted basis function expansion via the view of learning eigenfunctions in a Hilbert space. The connection clarifies which representation of functions is suitable for a good approximation (Sec. 4). Finally we present results of an empirical study in Sec. 5.

2 Semi-Supervised Function Induction

Suppose that, given n inputs $\{\boldsymbol{x}_i\}_{i=1}^n$ i.i.d. sampled from a density $p(\boldsymbol{x})$, one observes responses $\{y_i\}_{i=1}^l$ of an underlying function $f(\boldsymbol{x})$ on the first $l \leq n$ inputs (without loss of generality) plus some stationary additive noises. The goal is to estimate the underlying function f. *Supervised* induction ignores the existence of unlabelled data $\{\boldsymbol{x}_i\}_{i=l+1}^n$, and seeks for a f that minimizes the cost

$$\mathcal{Q}(f) \quad = \quad \frac{1}{2}\sum_{i=1}^{l}\big[y_i - f(\boldsymbol{x}_i)\big]^2 + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2 \tag{1}$$

where $\|f\|^2$ is the norm defined in a Hilbert space $\mathcal{H}$. The first term of $\mathcal{Q}(f)$ enforces f to be close to observations. The second term, called the *regularizer*, ensures the smoothness of f. A reasonable assumption behind the regularizer is that close inputs should have similar function values (as in ridge regression). The notion of *closeness* between two inputs usually does not regard their *context* of input density: for example, the two points might be separated by a low-density region.

Semi-supervised learning employs a different assumption, in which the smoothness of f is not evenly ensured but depends on the input density, i.e. f should change slowly in a dense region if compared to a low-density region. A possible approach to realize such an assumption is given by [10]:

$$\mathcal{S}(\boldsymbol{f}_n) = \sum_{i,j=1}^{n} W_{ij}\Big[\frac{f(\boldsymbol{x}_i)}{\sqrt{D_{ii}}} - \frac{f(\boldsymbol{x}_j)}{\sqrt{D_{jj}}}\Big]^2, \tag{2}$$

where $\boldsymbol{f}_n \in \mathbb{R}^{1\times n}$ denotes function values on the seen inputs, and the matrix W satisfying $W_{ij} \geq 0$ and $W_{ii} = 0$ for all $i \leq n$ can be viewed as a symmetric similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ (e.g., $W_{ij} = \exp(-\|x_i - x_j\|^2/2\sigma^{-2})$, and is enforced to be zero if $i = j$), and $D_{ii} = \sum_j W_{ij}$ reflects the local density of $\boldsymbol{x}_i$ (analog to the Parzen density). The regularizer penalizes the functions that change rapidly across nearby inputs. It is not hard to see that

$$\mathcal{S}(\boldsymbol{f}_n) = \boldsymbol{f}_n^T(\boldsymbol{I} - \boldsymbol{S})\boldsymbol{f}_n$$

where $\boldsymbol{S} \in \mathbb{R}^{n\times n}$ is the normalized similarity matrix with $\boldsymbol{S} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{W}\boldsymbol{D}^{-\frac{1}{2}}$ with $\boldsymbol{D}$ being a diagonal matrix $\{\boldsymbol{D}\}_{ii} = D_{ii}$. The matrix $\Delta = \boldsymbol{I} - \boldsymbol{S}$ is called *normalized graph Laplacian* in spectral graph theory [5]. $\mathcal{S}(\boldsymbol{f}_n)$ defines a regularizer for only the functions defined at discrete points. To carry out induction, we consider the class of approximating fucntions to be

$$f(\boldsymbol{x}) = \sum_{j=1}^{m} w_j\varphi_j(\boldsymbol{x}) \tag{3}$$

where $\{\varphi_j(\boldsymbol{x})\}_{j=1}^m$ are basis functions, which are not necessary orthogonal, and $\boldsymbol{w} = [w_1, \ldots, w_m]$ are the weights. In this paper we will only consider radial basis functions (RBF) $\varphi_j(\boldsymbol{x}) = \exp(-\|\boldsymbol{x} - \boldsymbol{x}_j\|^2/2\sigma_b^{-2})$, defined either for the whole set of seen inputs $\{\boldsymbol{x}_j\}_{j=1}^n$ or for a subset. In general, f describes a large class of functions, including neural networks with a finite number of hidden nodes. Now let $\boldsymbol{\varphi}_n \in \mathbb{R}^{m\times n}$ be the matrix with $\{\boldsymbol{\varphi}_n\}_{ji} = \varphi_j(\boldsymbol{x}_i)$, 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 labelled data. By plugging the representation of f into the regularizer Eq. (2), we obtain a cost function which is similar to Eq. (1)

$$\mathcal{Q}(\boldsymbol{w}) = \frac{1}{2}(\boldsymbol{y}_l - \boldsymbol{\varphi}_l\boldsymbol{w})^T(\boldsymbol{y}_l - \boldsymbol{\varphi}_l\boldsymbol{w}) + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{\varphi}_n\Delta\boldsymbol{\varphi}_n^T\boldsymbol{w} \tag{4}$$

By setting the derivatives of the cost function with respect to $\boldsymbol{w}$ to be zero, we obtain as optimal weights $\hat{\boldsymbol{w}} = (\boldsymbol{\varphi}_l\boldsymbol{\varphi}_l^T + \lambda\boldsymbol{\varphi}_n\Omega\boldsymbol{\varphi}_n^T)^{-1}\boldsymbol{\varphi}_l\boldsymbol{y}_l$ where $\boldsymbol{\Omega} = \boldsymbol{\varphi}_n\Delta\boldsymbol{\varphi}_n^T$. Let $\varphi(\boldsymbol{x}) = [\varphi_1(\boldsymbol{x}), \ldots, \varphi_m(\boldsymbol{x})]^T$. The approximated function is then given by

$$\hat{f}(\boldsymbol{x}) = \boldsymbol{\varphi}^T(\boldsymbol{x})(\boldsymbol{\varphi}_l\boldsymbol{\varphi}_l^T + \lambda\Omega)^{-1}\boldsymbol{\varphi}_l\boldsymbol{y}_l \tag{5}$$

The proposed method has certain advantages. First, it builds an inductive learner able to handle new test points. The computation for prediction only scales linearly as $O(m)$, while transduction has to re-compute the predictor whenever new test points arrive, which scales as $O(n^3)$. Second, for training the algorithm inverts an $m \times m$ matrix $\boldsymbol{\varphi}_l^T\boldsymbol{\varphi} + \lambda\Omega$, which can be more efficient than dealing with the $n \times n$ matrix in the transductive setting, assuming $m \ll n$.

3 Density-Dependent Regularizer and Graph Laplacian

A different regularizer was applied in [1, 11, 6],

$$\mathcal{L}(\boldsymbol{f}_n) = \sum_{i,j} W_{ij}\left[f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\right]^2 = \boldsymbol{f}_n^T(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{f}_n \tag{6}$$

Instead, we choose the normalized graph Laplacian as in [10], because the normalization by D_{ii} in Eq. (2) makes the smoothness constraint adapted to the local context of input density. Intuitively, the penalty strength with respect to a certain distance in a dense input region should be equal to the penalty strength for a relatively longer distance in a low-density region. Bousquet et al. [3] showed that the limiting case of $\mathcal{L}(\boldsymbol{f}_n)$ gives a regularizer $\int \|\nabla f(\boldsymbol{x})\|^2 p^2(\boldsymbol{x})d\boldsymbol{x}$. We can make a similar proof and derive the following proposition:

Proposition 1. For any function $f \in C^2(\mathbb{R}^d)$ with bounded Hessian, then

$$\lim_{\substack{\sigma\to 0\\ n\to\infty}} \frac{1}{n-1}\mathcal{S}(\boldsymbol{f}_n) \propto \int \|\nabla f(\boldsymbol{x})\|^2 p(\boldsymbol{x})d\boldsymbol{x} \tag{7}$$

The proposition states that the limiting case of $\mathcal{S}(\boldsymbol{f}_n)$ gives the *expected* smoothness $\mathbb{E}(\|\bigtriangledown f(\boldsymbol{x})\|^2)$ with respect to $p(\boldsymbol{x})$. Due to space limitations, we will provide the proof in a coming technical report. Now we compare the three regularization terms: (1) $\int \|\nabla f(\boldsymbol{x})\|^2 d\boldsymbol{x}$: This *density-free* constraint has been widely applied in supervised learning, e.g. in spline smoothing. For a linear model, it gives the maximum-margin criterion; (2) $\int \|\nabla f(\boldsymbol{x})\|^2 p^2(\boldsymbol{x})d\boldsymbol{x}$: This is the regularizer approximated by $\mathcal{L}(\boldsymbol{f}_n)$; (3) $\int \|\nabla f(\boldsymbol{x})\|^2 p(\boldsymbol{x})d\boldsymbol{x}$: This case corresponds to the regularizer $\mathcal{S}(\boldsymbol{f}_n)$ induced by the normalized graph Laplacian.

The second and third regularizers are *density-dependent*, giving rise to semi-supervised learning. Though $\mathcal{L}(\boldsymbol{f}_n)$ might be considered to be more intuitive than $\mathcal{S}(\boldsymbol{f}_n)$, this is not the case in the limiting case, while Eq. (2) converges to the expected squared gradient of f with respect to $p(\boldsymbol{x})$. The difference between both regularizers can be well illustrated by the toy problem shown in Fig. 1, where we can see that $\mathcal{L}(\boldsymbol{f}_n)$ over-emphasizes the smoothness on dense region while Eq. (2) imposes a balanced smoothness constraint over $p(\boldsymbol{x})$.

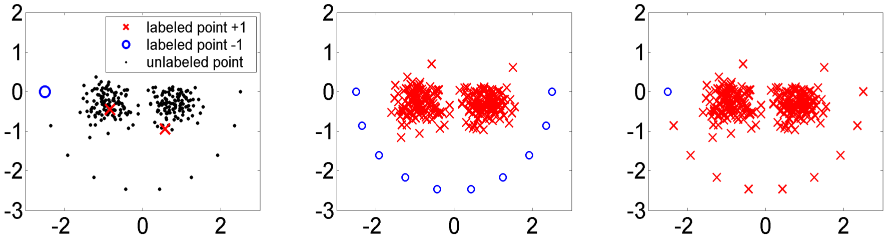

Figure 1: Classification on the doll toy data. Left panel: toy data; middle panel: normalized; right panel: non-normalized.

4 Optimal Basis Expansion for Semi-Supervised Induction

The regularizer $\int \|\nabla f(\boldsymbol{x})\|^2 p(\boldsymbol{x}) d\boldsymbol{x}$ defines a norm $\|f\|^2_{\mathcal{H}}$ in a reproducing kernel Hilbert space $\mathcal{H}$ with infinite dimensions, while Eq. (3) restricts our approximated function class in an m-dimensional Hilbert space $\mathcal{H}_m$. In this section we further study the properties of $\mathcal{H}_m$ and consider the problem of finding optimal basis functions in Eq. (3) that lead to optimal $\mathcal{H}_m$.

4.1 The Spectrum of $\mathcal{H}_m$

Suppose $\mathcal{H}$ is associated with a set of eigenfunctions $\phi_k(\boldsymbol{x})$ and eigenvalues λ_k such that $\|f\|^2_{\mathcal{H}} = \sum_{k=1}^{\infty} \lambda_k c_k^2$, where c_k is the inner product $\langle f, \phi_k \rangle$. Recall that the norm penalizes functions with large projections on the eigenfunctions with the leading eigenvalues. Assuming that the null space of the Hilbert space as rank zero, this means that functions are favored which have a large projection onto the space defined by the eigenfunctions with the smallest eigenvalues. In the following we call those simply the set of smoothest eigenfunctions since reasonable regularizers would favor smooth eigenfunction. Thus, λ_k indicates the smoothness of eigenfunction ϕ_k, i.e. smaller λ_k means smoother ϕ_k. Following our discussion, minimizing $\|f\|^2_{\mathcal{H}}$ enforces f to be close to the set of smooth eigenfunctions ϕ_k with smallest λ_k.

The difficulty of transforming semi-supervised transduction to induction is a result of the fact that eigenfunctions that construct $\mathcal{H}$ are unknown, since we can only infer the discrete realizations of eigenfunctions on the finite i.i.d. inputs $\{\boldsymbol{x}_i\}_{i=1}^n$, i.e. eigenvectors of Δ[1], giving rise to a regularizer $\boldsymbol{f}_n^T \Delta \boldsymbol{f}_n$ on finite function values rather than $\|f\|^2_{\mathcal{H}}$. Here we will point out that the proposed induction in Sec. 2 implicitly reconstructs eigenfunctions and eigenvalues in $\mathcal{H}_m$ that give the same smoothness measure as in $\mathcal{H}$. Let the eigenfunctions in $\mathcal{H}_m$ have the form

$$\tilde{\phi}_k(\boldsymbol{x}) = \sum_{j}^{m} \alpha_{jk} \varphi_j(\boldsymbol{x}) \quad \text{for} \quad k = 1, \ldots, m. \tag{8}$$

which satisfy the unitary condition $\int \tilde{\phi}_k(\boldsymbol{x})^2 p(\boldsymbol{x}) dx = 1$. Following our previous discussion, the m eigenfunctions in $\mathcal{H}_m$ should be related to the m eigenfunctions in $\mathcal{H}$ with the m smallest eigenvalues. We shall let $\tilde{\phi}_k(\boldsymbol{x})$ preserve the smoothest eigenfunctions of $\mathcal{H}$ reflected by the eigenvectors of Δ with the smallest eigenvalues based on the empirical data, giving rise to the constrained minimization problem:

$$\min_{\boldsymbol{\alpha}_k = [\alpha_{1k}, \ldots, \alpha_{mk}]^T} \quad \tilde{\boldsymbol{\phi}}_k^T \Delta \tilde{\boldsymbol{\phi}}_k, \quad \text{subject to:} \quad \frac{1}{n} \tilde{\boldsymbol{\phi}}_k^T \tilde{\boldsymbol{\phi}}_k = 1 \tag{9}$$

[1] The $n \times n$ matrix induces n eigenvectors that approximate the original infinite eigenfunctions induced by $\|f\|^2_{\mathcal{H}}$. See the relation between eigenvectors and eigenfunctions in [2].

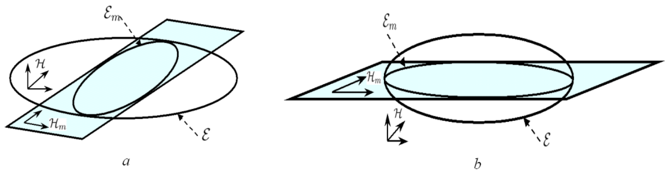

Figure 2: Geometry interpretation: The right case gives a better $\mathcal{H}_m$ to preserve the structure of $\mathcal{H}$.

where $\tilde{\boldsymbol{\phi}}_k = [\tilde{\phi}_k(\boldsymbol{x}_1),\ldots,\tilde{\phi}_k(\boldsymbol{x}_n)]^T$, and the constraint is the the unitary condition approximated by empirical averaging on i.i.d. samples. The minimization restricts $\tilde{\boldsymbol{\phi}}_k$ to lie on a hyper sphere and enforces it close to the smoothest eigenvectors of Δ. Its Lagrangian formulism suggests the equivalence to a generalized eigendecomposition problem which has m solutions

$$(\boldsymbol{\varphi}_n \triangle {\boldsymbol{\varphi}_n}^T)\boldsymbol{u}_k = \tilde{\lambda}_k(\boldsymbol{\varphi}_n{\boldsymbol{\varphi}_n}^T)\boldsymbol{u}_k, \quad \text{for} \quad k = 1,\ldots,m \tag{10}$$

where $\boldsymbol{\varphi}_n \in \mathbb{R}^{m\times n}$ is as defined in Eq. (4), $\boldsymbol{u}_k \in \mathbb{R}^{m\times 1}$ are generalized eigenvectors, and $\tilde{\lambda}_k$ are generalized eigenvalues. Finally, the estimated eigenfunctions are

$$\tilde{\phi}_k(x) = \sqrt{n}\boldsymbol{u}_k^T\boldsymbol{\varphi}(\boldsymbol{x}), \quad \text{for} \quad k = 1,\ldots,m \tag{11}$$

where $\varphi(\boldsymbol{x}) = [\varphi_1(\boldsymbol{x}),\ldots,\varphi_m(\boldsymbol{x})]^T$. The above equation gives m orthogonal basis functions in $\mathcal{H}_m$ that preserve the smoothest eigenfunctions $\phi_k(\boldsymbol{x})$ which are discretely realized as the eigenvectors of Δ with the smallest eigenvalues. The corresponding $\{n\tilde{\lambda}_k\}_{k=1}^m$ reflect the smoothness of $\{\tilde{\phi}_k(\boldsymbol{x})\}_{k=1}^m$ in the way that a smaller value indicates a smoother function. Then the smoothness of an $f \in \mathcal{H}_m$ in Eq. (3) is given by $\|f\|^2_{\mathcal{H}_m} = \sum_{k=1}^m n\tilde{\lambda}_k\tilde{c}_k^2$, where $\tilde{c}_k = \langle f, \tilde{\phi}_k\rangle$. The cost function Eq. (4) is thus interpreted as an empirical loss plus the norm $\|f\|^2_{\mathcal{H}_m}$.

4.2 Geometry Interpretations: What is a Good $\mathcal{H}_m$

As illustrated in Fig. 2, the unit norm $\|f\|^2_{\mathcal{H}} = 1$ restricts f to lie on a hyper ellipsoid $\mathcal{E}$ with the axes corresponding to the eigenfunctions $\{\phi_k(\boldsymbol{x})\}_{k=1}^\infty$ and the span along each axis is scaled by ${\lambda_k}^{-\frac{1}{2}}$ (i.e. smooth eigenfunctions correspond to the principle axes of $\mathcal{E}$). In addition, $\|f\|^2_{\mathcal{H}_m} = 1$ restrict f to lie on an m-dimensional ellipsoid $\mathcal{E}_m$ which is the *intersection* of $\mathcal{E}$ in $\mathcal{H}_m$. The eigenfunctions Eq. (11) are the axes of $\mathcal{E}_m$ and the spans of axes are scaled by $\{\tilde{\lambda}_k^{-\frac{1}{2}}\}_{k=1}^m$. It is clear that a good $\mathcal{H}_m$ should preserve the principle axes of $\mathcal{E}$, i.e. approximating the smoothest eigenfunctions in $\mathcal{H}$. Therefore we use the *volume* of $\mathcal{E}_m$ to define the optimal Hilbert space $\mathcal{H}_m$:

$$\mathcal{H}_m^{opt} = \arg\max_{\mathcal{H}_m \in \mathcal{A}} \sum_{k=1}^m \frac{1}{\tilde{\lambda}_k^2 + \tilde{\sigma}^2} \tag{12}$$

where $\mathcal{A}$ is the considered domain of $\mathcal{H}_m$, and $\tilde{\sigma}$ is a small number to avoid the difficulties when $\tilde{\lambda}_k \approx 0$. Eq. (12) defines how we should choose the optimal basis-function set in practice, for example, by tuning the width, number, and centers of RBF basis functions.

Finally we point out that learning the eigenfunctions of an RKHS has already been discussed in various contexts, like *kernel PCA* for non nonlinear dimensionality reduction [7], *Nyström* method for speeding up kernel methods [9], and *out-of-sample extension for manifold learning* [2]. However, our method is derived in a different context where the kernel function is assumed unknown.

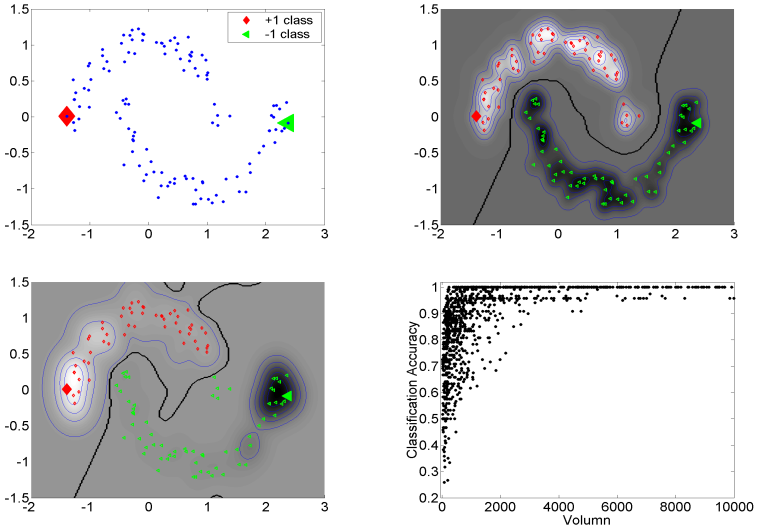

Figure 3: Semi-supervised induction on the two-moon data: top-left, each class has only one labelled example; top-right, induction with 120 basis functions; bottom-left, induction with 60 basis functions; bottom-right, 1200 random trials of function settings showing the connection between predictive accuracy and the performance indicator suggested in Eq. (12). For the illustration of induction, the black bold curve gives the classification boundary and the gray level indicates the function value.

5 Empirical Study

5.1 Toy Data

We test the proposed algorithms on the two-moon toy problem [10]. As shown in Fig. 3, 120 inputs are generated from two underlaying classes and each class has only *one* labelled example. The performance of transduction has been shown in [10], 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. 4. 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. In the next, we repeat 1200 trials by randomizing the number of RBF basis functions (between 10 and 110) and the width (between 0.1 and 0.2), and for each trial get the classification accuracy and the volume of corresponding $\mathcal{E}_m$ Eq. (12). We plot all the 1200 accuracy-volume dots in the bottom-right of Fig. 3, which indicates that a larger volume leads to a better and stabler classifier.

5.2 Digit Recognition

We test the performance of algorithms in a digit recognition task based on the USPS benchmark. We follow the setting in [10] 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

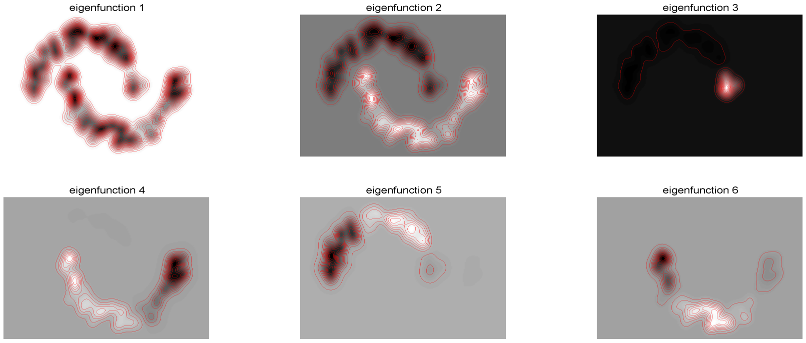

Figure 4: The six eigenfunctions with smallest eigenvalues, estimated with 120 basis functions. The eigenfunctions not only expose the structure of input density, but also help to understand semi-supervised learning: choosing the smooth eigenfunctions that also explain the labelled examples well, which gives the second eigenfunction in the case shown in Fig. 3. (Note: the figure is more informative if enlarged on the screen.)

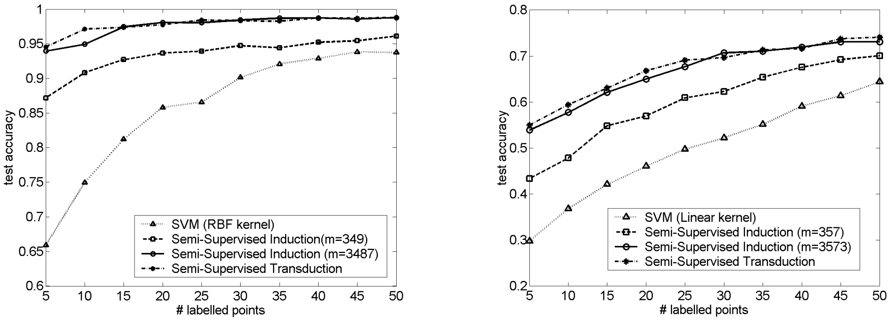

Figure 5: Left panel: Test results for digit recognition based on USPS data. Right panel: Test results for text categorization based on 20-newsgroup data.

the baseline, and semi-supervised transduction described by [10]. 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. (4) is set to be 100, which corresponds to $\alpha = 0.99$ in [10]. We split the data into *seen* (including labelled and unlabelled data) and *unseen* sets, 90% vs. 10%, and examine the predictive accuracy on the unseen set given a number of labelled 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 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. 5-(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.

5.3 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 [10], 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 $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 - \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle / \|\boldsymbol{x}_i\| \|\boldsymbol{x}_j\|$ is applied to form RBF functions for affinity matrix (with width 0.15), basis functions for induction (width 0.15) [10]. We then perform 50 trials with random 90% *seen* and 10% *unseen* split and report the average performance of each algorithm in Fig. 5-(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.

6 Conclusion

This paper realizes a semi-supervised *inductive* algorithm by extending previous transductive approaches. The idea is to use basis function expansion to form a regularizer induced by the normalized graph Laplacian. We clarify the reason of choosing the adopted smoothness regularizer and discuss what are the desired approximating functions in terms of eigenfunction estimation. Finally the effectiveness of the proposed algorithm is illustrated on both toy problem and digit recognition. An unsolved problem for semi-supervised learning is the model selection when little labelled examples are known, which should be an interesting future work.

References

[1] Belkin, M. and Niyogi, P. Using manifold structure for partially labeled classification. In *NIPS 15*. 2003.

[2] Bengio, Y., Paiement, J.-F., Vincent, P., Delalleau, O., Roux, N. L., and Ouimet, M. Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering. In *NIPS 16*. 2004.

[3] Bousquet, O., Chappelle, O., and Hein, M. Measure based regularization. In *NIPS 16*. 2004.

[4] Chapelle, O., Weston, J., and Schölkopf, B. Cluster kernels for semi-supervised learning. In *NIPS 15*. 2003.

[5] Chung, F. *Spectral Graph Theory*. No. 92 in Regional Conference Series in Mathematics. American Mathematical Society, 1997.

[6] He, X. and Niyogi, P. Locality Preserving Projections. In *NIPS 16*. 2004.

[7] Schölkopf, B., Smola, A., and Müller, K.-R. Kernel principal component analysis. In *Advances in Kernel Methods - Support Vector Learning*, pp. 327–352. 1999.

[8] Szummer, M. and Jaakkola, T. Partially labeled classification with markov random walks. In *NIPS 14*. 2002.

[9] Williams, C. and Seeger, M. Using the nyström method to speed up kernel machines. In *NIPS 13*. 2001.

[10] Zhou, D., Bousquet, O., Lal, T., Weston, J., and Schölkopf, B. Learning with local and global consistency. In *NIPS 16*. 2004.

[11] Zhu, X., Ghahramani, Z., and Lafferty, J. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML-2003*. 2003.