Linear Neighborhood Propagation and Its Applications

Jingdong Wang, *Member, IEEE,* Fei Wang, *Student member, IEEE,*
Changshui Zhang, *Member, IEEE,* Helen C. Shen, and Long Quan, *Senior member, IEEE*

Abstract—In this paper, a novel graph-based transductive classification approach, called Linear Neighborhood Propagation, is proposed. The basic idea is to predict the label of a data point according to its neighbors in a linear way. This method can be cast into a second-order intrinsic Gaussian Markov random field framework. Its result corresponds to a solution to an approximate inhomogeneous biharmonic equation with Dirichlet boundary conditions. Different from existing approaches, our approach provides a novel graph structure construction method by introducing multiple-wise edges instead of pairwise edges, and presents an effective scheme to estimate the weights for such multiple-wise edges. To the best of our knowledge, these two contributions are novel for semi-supervised classification. The experimental results on image segmentation and transductive classification demonstrate the effectiveness and efficiency of the proposed approach.

Index Terms—Gaussian Markov random fields, linear neighborhood propagation, transductive classification, image segmentation.

1 Introduction

Many real-world pattern classification and data mining applications usually suffer from a lack of sufficient labeled data since labeling requires extensive human effort and much time. However, in many cases, a large amount of unlabeled data can be more easily available. For example, in text classification, the user may have easy access to a large database of documents (e.g., by crawling the web), but only a small part of them are manually classified.

To leverage both the labeled and unlabeled data, *semi-supervised classification* (SSC) methods were proposed. There are many SSC methods, such as the Expectation-Maximization algorithm for semi-supervised generative mixture models [15], [30], self-training [32], co-training [11], transductive support vector machines [20], [21], and graph-based methods [43], [47]. A detailed survey of the literature on SCC methods is presented in [46]. In general, these methods can be divided into two categories: *inductive classification* methods and *transductive classification* methods. Inductive classification methods try to induce a decision function that has a low classification error rate on the whole sample space, whereas transductive classification methods directly estimate the labels of the unlabeled data. In the following, we will give a review of graph-based methods for transductive classification, which are closely related to the proposed approach.

1.1 Related Work

The common assumption of graph-based methods is *label consistency* [43] (also called *cluster assumption* [13]), i.e., nearby points are prone to have the same labels. Most existing graph-based methods essentially estimate a labeling on a graph that is constructed by connecting the neighboring data points, expecting that such a labeling satisfies two properties: 1) it should be as close as possible to (or the same as) the given labeling on the labeled data points, and 2) it should be smooth on the whole graph. The former property can be formulated as a *loss function* penalizing the deviation of the predicted labeling from the given labeling. The latter property can be expressed as a *regularizer* enforcing label consistency. All the existing methods are very similar, and differ only slightly in the loss function and the regularizer.

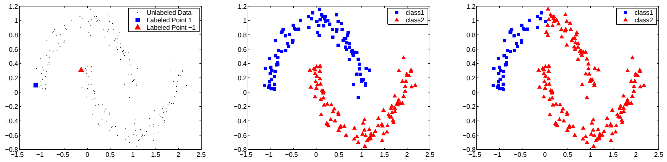
The loss function basically consists of two parts: the penalty for deviation of the predicted labeling from the given labeling on the labeled data points and the penalty for deviation of the predicted labeling from the labeling bias on the unlabeled data points. It can be formulated as an energy function

$$\begin{aligned} E_{\text{loss}} &= \lambda_{\mathcal{L}} E_{\mathcal{L}} + \lambda_{\mathcal{U}} E_{\mathcal{U}} \\ &= \lambda_{\mathcal{L}} \sum\nolimits_{i \in \mathcal{L}} (y_i - \bar{y}_i)^2 + \lambda_{\mathcal{U}} \sum\nolimits_{j \in \mathcal{U}} (y_j - \bar{y}_j)^2, \end{aligned} \quad (1)$$

where $\mathcal{L}$ is a set of indices for the labeled points, $\mathcal{U}$ is a set of indices for the unlabeled points, $\bar{y}_i$ is the given label when $\mathbf{x}_i$ is a labeled point, and the biased label when $\mathbf{x}_i$ is an unlabeled point. The harmonic approach [47] only penalizes the deviation from the given labels, i.e., the first term of the right-hand side in Eqn. (1) is involved,

- J. Wang is with the Internet Media Group, Microsoft Research Asia, Beijing, China.
E-mail: i-jingdw@microsoft.com.
- F. Wang and C. Zhang are with the Department of Automation, Tsinghua University, Beijing, China.
E-mail: feiwang03@gmail.com, zcs@mail.tsinghua.edu.cn.
- H.C. Shen and L. Quan are with the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology, Hong Kong.
E-mail: {helens, quan}@cse.ust.hk.

Manuscript received March 29, 2007; revised October 09, 2007.



(a) (b) (c)

Fig. 1. Illustration of the sensitivity of the consistency method [43] to the parameter setting. The edge weight is computed by a Gaussian function $\exp(-\frac{1}{2\sigma^2}||\mathbf{x}_i - \mathbf{x}_j||^2)$. (a) shows the points with two labeled points, (b) shows the classification result with $\sigma = 0.15$, and (c) shows the classification result with $\sigma = 0.25$. It can be seen that a small variation of σ will lead to a different classification result.

while the consistency method [43] additionally considers the second term and prefers a compromise in assigning the labels on the unlabeled data. Alternatively, one may train an external classifier from the labeled data, and then for each data point compute the compatibilities between it and its possible associate labels to obtain an estimation of the label bias.

The regularizer is usually formulated based on the Laplacian operator and its variants. In [6], [47], the combinatorial graph Laplacian, $\mathbf{\Delta}$, is adopted for the regularizer, which is formulated as

$$E_{\text{reg}} = \frac{1}{2}\sum\nolimits_{i,j\in\mathcal{L}\cup\mathcal{U}} w_{ij}(y_i - y_j)^2 = \mathbf{y}^T\mathbf{\Delta}\mathbf{y}, \qquad (2)$$

where $\mathbf{y}$ is a label vector, w_{ij} is the similarity between data points i and j, and $\mathbf{\Delta} = \mathbf{D} - \mathbf{W}$. Here, $\mathbf{W} = [w_{ij}]_{n\times n}$, where $n = |\mathcal{L}\cup\mathcal{U}|$ is the number of the data points, $\mathbf{D} = \text{Diag}(d_1, \cdots, d_n)$ is a diagonal degree matrix, and $d_i = \sum_j w_{ij}$. It is mentioned in [6] that a p-Laplacian, $\mathbf{\Delta}^p$, can be used, but unfortunately the choice of p is not analyzed. The normalized combinatorial Laplacian operator is used in [43]:

$$\begin{aligned} E_{\text{reg}} &= \frac{1}{2}\sum\nolimits_{i,j\in\mathcal{L}\cup\mathcal{U}} w_{ij}\Big(\frac{y_i}{\sqrt{d_i}} - \frac{y_j}{\sqrt{d_j}}\Big)^2 \\ &= \mathbf{y}^T\mathbf{D}^{-\frac{1}{2}}\mathbf{\Delta}\mathbf{D}^{-\frac{1}{2}}\mathbf{y}. \qquad (3)\end{aligned}$$

In semi-supervised classification, the label vector is integer-valued. Hence, the task is in essence a discrete optimization problem. For two-class classification problems, it can be formulated as a graph min-cut problem, and a global optimal solution can be obtained as shown in [9]. Multi-class problems are generally NP-hard (if the graph contains cycles). Therefore, an approximation procedure is necessary. There are two types of approximation methods: 1) adopting combinatorial optimization or statistical inference methods to get an approximate solution and 2) relaxing the discrete optimization problem to a continuous one to obtain an approximate solution. Combinatorial optimization methods, such as graph min-cut and random min-cut [9], [10], can be adopted. Those methods are also widely used in computer vision, such as stereo matching [23], and in computer graphics, such as texture synthesis [24]. Relaxation approaches obtain real label values as the intermediate results, which reflect the ordinal degrees to which the points belong to the possible labels. They may be applied to other problems, such as matting in image editing [18] and ranking in image retrieval [19].

However, there are several drawbacks to existing graph-based semi-supervised classification methods in the following three aspects:

1) Graph structure construction. It is difficult because there is little information for learning the structure. Most existing methods only construct a pairwise graph structure, which apparently has the disadvantage that only the pairwise relations between the data points are taken into account.
2) Edge weight estimation. The majority of existing approaches set the weight between two data points using a Gaussian function. In practice, the labeling is very sensitive to the parameter setting in the Gaussian function. An illustrative example in Fig. 1 shows the sensitivity. It is expected that the weight is adaptively learned from the data and little human effort is required.
3) Noisy data handling. The penalty weights for different data points are homogeneous in nearly all the approaches except [21]. Homogeneous penalization is sensitive to noisy data points. A more robust way is to take an inhomogeneous penalty with less confidence in the labeling of possible noisy data points.

The first two aspects are related to the regularizer design. Few existing approaches consider them. Practice shows that it is more important to construct a good graph (including structure and weight) than to choose a good inference algorithm. This is also pointed out in [46]. The last aspect is related to the loss function definition. Previous methods rarely consider the possible noise in the data points, and hence the classification results may be sensitive to noisy data points.

1.2 Contributions

In this paper, we address two little-studied issues: graph structure construction and edge weight estimation. Moreover, in order to unify existing graph-based semi-supervised classification approaches, we present a probabilistic framework.

We propose a novel method, called *Linear Neighborhood Propagation* (LNP), under the framework of an intrinsic Gaussian Markov random field. First, a hypergraph is constructed with a series of overlapped neighborhood patches being hyperedges. The weight for each hyperedge is estimated by solving a standard quadratic program. Then, all the weights are aggregated together to obtain a regularization matrix over the whole graph, which is essentially an approximation of the *biharmonic* matrix on a weighted undirected graph. Thus a regularizer is constructed based on the biharmonic matrix. Finally, the labeling is obtained by solving a biharmonic equation with Dirichlet boundary conditions. In summary, this paper offers the following contributions:

1) We build a hypergraph (multiple-wise graph) instead of a pairwise graph. In a pairwise graph, only the relations between two data points are explored. However, a hypergraph provides a way to explore the multiple-wise (or high-order) relations among more data points, and hence is more powerful.
2) We present an automatic and effective weight estimation scheme. Generally speaking, the weights on the hyperedges can be arbitrary as long as they can reflect the requirement of label consistency. However, it is more practical if an efficient and effective estimation approach exists. We estimate the edge weight based on a linear construction assumption, which is very similar to locally linear embedding (LLE) developed in [34]. Differently, we constrain the weights to be nonnegative in order to guarantee that the estimated weights are meaningful and reflect the relations among the points. A similar weight estimation scheme was also used to improve the harmonic approach in [40], but this paper gives a natural derivation from a probabilistic view and offers a weight estimation scheme for hypeedges instead of pairwise edges.
3) We present some analysis to compare our approach and existing graph-based methods. Our approach can capture high-order relations among data points. This difference implies that our method is more powerful than other methods.

1.3 Organization

The remainder of this paper is organized as follows. Section 2 reviews the background of Gaussian Markov random fields. Linear Neighborhood Propagation is proposed in Section 3, and the analysis is presented in Section 4. Section 5 discusses some extensions. Section 6 illustrates the effectiveness and robustness of our approach on toy examples. The image segmentation results are given in Section 7. The experiments on transductive pattern recognition are given in Section 8. Finally, Section 9 concludes this paper.

2 Preliminaries

2.1 Markov Random Fields

A *Markov random field* (MRF), also known as an *undirected graphical model* or a *Markov network*, is a graph in which the nodes represent the variables and the edges represent the compatibility constraints between the variables. According to the Hammersley-Clifford theorem, it can be guaranteed that the probability distribution can be factorized into a product of the compatibility functions over the maximal cliques of the graph. Denoting by $\mathbf{x}$ the values of all the variables in the graph, such a factorization takes the form:

$$p(\mathbf{x}) = \frac{1}{Z}\prod\nolimits_c \psi(\mathbf{x}_c), \tag{4}$$

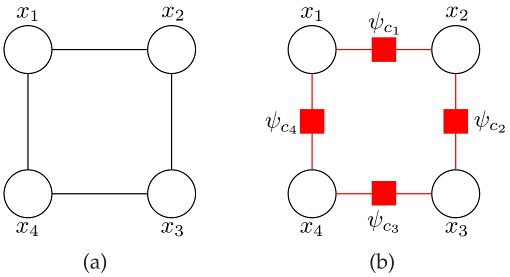


Fig. 2. A sample Markov random field. (a) shows a simple MRF, (b) shows its factor graph representation.

where $\mathbf{x}_c$ is a sub-vector of $\mathbf{x}$ corresponding to maximal clique c in the graph, and Z is a normalization constant.

As an example, Fig. 2(a) shows a first-order MRF, in which there are four maximal cliques. An MRF can also be represented as a factor graph as shown in Fig. 2(b) by adding a function node for each maximal clique.

2.2 Gaussian Markov Random Fields

A *Gaussian Markov random field* (GMRF) is an MRF in which the joint distribution is Gaussian. Mathematically, the compatibility function $\psi(\mathbf{x}_c)$ in Eqn. (4) is an exponential distribution

$$\psi(\mathbf{x}_c) = \exp\{-E(\mathbf{x}_c)\},$$

where

$$E(\mathbf{x}_c) = \frac{1}{2}(\mathbf{x}_c - \boldsymbol{\mu}_c)^T \mathbf{Q}_c (\mathbf{x}_c - \boldsymbol{\mu}_c).$$

Thus the joint probability is also a Gaussian distribution

$$p(\mathbf{x}) = \frac{1}{Z}\exp\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{Q}(\mathbf{x}-\boldsymbol{\mu})\}, \tag{5}$$

where $Z = (2\pi)^{n/2}|\mathbf{Q}|^{-1/2}$, $\boldsymbol{\mu}$ is the mean and $\mathbf{Q}$ is the precision matrix, i.e., the inverse covariance matrix.

If there is no edge between nodes i and j, or $\mathbf{x}_i$ and $\mathbf{x}_j$ are assumed to be conditionally independent, then $q_{ij} = 0$. For example, the precision matrix of the GMRF shown in Fig. 2(a) has the following form:

$$\mathbf{Q} = \begin{bmatrix} \times & \times & 0 & \times \\ \times & \times & \times & 0 \\ 0 & \times & \times & \times \\ \times & 0 & \times & \times \end{bmatrix}.$$

2.3 Intrinsic Gaussian Markov Random Fields

A GMRF is called an *intrinsic Gaussian Markov random field* (IGMRF) if the precision matrix $\mathbf{Q}$ is not of full rank [35]. The *Laplacian* matrix of the graph in Fig. 2(a),

$$\mathbf{L} = \begin{bmatrix} 1 & -0.5 & 0 & -0.5 \\ -0.5 & 1 & -0.5 & 0 \\ 0 & -0.5 & 1 & -0.5 \\ -0.5 & 0 & -0.5 & 1 \end{bmatrix},$$

can be set as a precision matrix of an IGMRF. The IGMRF with the Laplacian matrix or its variants as the precision matrix has wide applications in real-world problems. More descriptions about IGMRFs are presented in [35].

3 Linear Neighborhood Propagation

Suppose there are n data points $\mathcal{X} = \{\mathbf{x}_i\}_{i\in\mathcal{N}}$, and l points $\{\mathbf{x}_i\}_{i\in\mathcal{L}}$ are labeled as $\{\bar{y}_i\}_{i\in\mathcal{L}}$, $\mathcal{L} = \{1, \cdots, l\}$. The task is to assign the labels $\{y_i\}_{i\in\mathcal{U}}$ to the remaining points $\{\mathbf{x}_i\}_{i\in\mathcal{U}}$ with $\mathcal{U} = \{l+1, \cdots, n\}$. Let $u = n - l$ be the number of the unlabeled points. We organize the points in the form of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$. The node set $\mathcal{V}$ corresponds to n data points, $\mathcal{V} = \mathcal{L} \cup \mathcal{U}$. $\mathcal{E}$ is the edge set, and $\mathcal{E} = \{e_1, \cdots, e_k\}$ with edge e_i corresponding to a set of points. $\mathcal{W} = \{w_1, \cdots, w_k\}$ with w_i equal to the weight of edge e_i. In the following, we will formulate this problem in an intrinsic Gaussian Markov random field framework.

3.1 First-Order IGMRFs

An IGMRF is called first-ordered if the precision matrix $\mathbf{Q}$ satisfies $\mathbf{Qe} = \mathbf{0}$, where $\mathbf{e}$ is a n-dimensional vector of ones. Here, we give a widely-used method to construct a first-order IGMRF. Suppose each edge e in $\mathcal{E}$ is composed of a pair of vertices (i, j). Following the terminology in [35], we define an *increment* for each edge as

$$d_{ij} = y_i - y_j. \tag{6}$$

We assume that the increment satisfies a Gaussian distribution

$$d_{ij} \sim \mathcal{N}(0, 1/w_{ij}), \tag{7}$$

where w_{ij} is the edge weight. Assuming the increments are independent, the joint probability over $\mathbf{y}$ becomes the normalized product of the probabilities over all the increments:

$$\begin{aligned} p(\mathbf{y}) &\propto \prod\nolimits_{(i,j)\in\mathcal{E}} p(d_{ij}) \\ &\propto \prod\nolimits_{(i,j)\in\mathcal{E}} \exp(-\frac{1}{2} w_{ij}(y_i - y_j)^2) \\ &= \exp(-\frac{1}{2} \sum\nolimits_{(i,j)\in\mathcal{E}} w_{ij}(y_i - y_j)^2) \\ &= \exp(-\frac{1}{2} \mathbf{y}^T(\mathbf{D} - \mathbf{W})\mathbf{y}). \end{aligned} \tag{8}$$

Here, the precision matrix is just the *Laplacian* matrix, $\mathbf{Q} = \mathbf{\Delta} = \mathbf{D} - \mathbf{W}$, i.e.,

$$q_{ij} = d_i \delta_{[i=j]} - w_{ij},$$

where $\delta_{[i=j]} = 1$ if $i = j$, and 0 otherwise. From Eqn. (8), we can obtain the conditional expectation

$$\mathrm{E}(\mathbf{y}_u | \mathbf{y}_l = \bar{\mathbf{y}}_l) = -\mathbf{Q}_{uu}^{-1}\mathbf{Q}_{ul}\bar{\mathbf{y}}_l, \tag{9}$$

where $\mathbf{Q}_{uu} = [q_{ij}]_{i,j\in\mathcal{U}}$ is a submatrix of $\mathbf{Q}$, and $\mathbf{Q}_{ul}$ is similarly defined. In the following notation, the submatrices in such forms are similarly defined. It should be noted that the conditional expectation of $\mathbf{y}_u$ is also the mode (maximum) of the conditional probability $p(\mathbf{y}_u|\mathbf{y}_l)$. In essence, the approaches in [16], [17], [43] and [47], including the harmonic approach, the consistency approach and so on, can be cast into this first-order IGMRF framework.

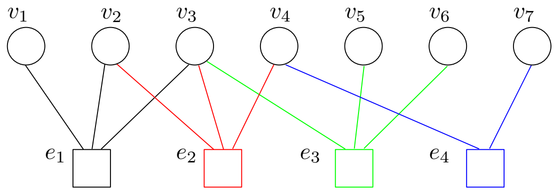


Fig. 3. A factor graph representation of a hypergraph. e represents a hyperedge vertex. v represents a data vertex. The data vertices connecting a hyperedge vertex correspond to a hyperedge.

3.2 Second-Order IGMRFs

Starting from the intuition of label consistency among the neighboring points, we construct a *hypergraph* $\mathcal{G}' = (\mathcal{V}, \mathcal{E}', \mathcal{W})$, where $\mathcal{V}$, a set of data vertices, corresponds to all the points, which is the same as in the pairwise graph $\mathcal{G}$, while each edge $e \in \mathcal{E}'$ is a *hyperedge* that may include more than two vertices. Specifically, we construct a hyperedge by connecting a vertex i and its neighboring vertices $\mathcal{N}_i$. For convenience, we index such a hyperedge as e_i using its center vertex i, $e_i = \{i\} \cup \mathcal{N}_i$. Such a graph with hyperedges is called a *hypergraph*. It is not easy to depict a hypergraph in a similar way as a pairwise graph is depicted since a hyperedge is not easily illustrated. We use a factor graph to represent it. We introduce additional vertices, called *hyperedge vertices* (or *function vertices* in the terminology of factor graph theory), to represent the hyperedges, and connect a hyperedge vertex and the data vertices belonging to the corresponding hyperedge. An example of a hypergraph is shown in Fig. 3.

Similarly, we define an *increment* for hyperedge $e_i = \{i\} \cup \mathcal{N}_i$ as

$$d_i = y_i - \sum\nolimits_{j\in\mathcal{N}_i} w_{ij} y_j, \tag{10}$$

where the weights satisfy $\sum_{j\in\mathcal{N}_i} w_{ij} = 1$. The weights should reflect the local similarity of the labels. We suppose that the increment satisfies an i.i.d. (independent and identically-distributed) standard normal distribution

$$d_i \sim \mathcal{N}(0, 1). \tag{11}$$

Then we can obtain the joint distribution over $\mathbf{y}$ as the normalized product of the probabilities over all the increments:

$$\begin{aligned} p(\mathbf{y}) &\propto \prod\nolimits_i p(d_i) \\ &\propto \exp(-\frac{1}{2} \sum\nolimits_i (\sum\nolimits_{j\in\mathcal{N}_i} w_{ij} y_j - y_i)^2) \\ &= \exp(-\frac{1}{2} \mathbf{y}^T(\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W})\mathbf{y}). \end{aligned} \tag{12}$$

Here, the precision matrix is $\mathbf{Q} = (\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W})$, called a *biharmonic* matrix, with

$$q_{ij} = \delta_{[i=j]} - w_{ij} - w_{ji} + \sum\nolimits_{k\in\mathcal{N}_i\cap\mathcal{N}_j} w_{ki} w_{kj},$$

where $\delta_{[i=j]} = 1$ if $i = j$, and 0 otherwise. This precision matrix is similar to the one of a second-order IGMRF in two dimensions that is presented in [35]. Hence, the above described probability distribution can be viewed as a second-order IGMRF over a graph. It can also be considered as a second-order random walk model over a graph.

In the above probabilistic formulation, the labels $\mathbf{y}_u$ on the unlabeled data points can also be estimated by computing the conditional expectation

$$\mathrm{E}(\mathbf{y}_u|\mathbf{y}_l = \bar{\mathbf{y}}_l) = -\mathbf{Q}_{uu}^{-1}\mathbf{Q}_{ul}\bar{\mathbf{y}}_l. \tag{13}$$

3.3 Learning the Weights

The weight w_{ij} is very important for forming the biharmonic matrix. Straightforwardly, we can define a weight between two data points as a Gaussian function, $\bar{w}_{ij} = \exp(-\frac{1}{2\sigma^2}||\mathbf{x}_i - \mathbf{x}_j||^2)$, where σ is the length scale parameter. This setting sounds fine, and has been applied in pairwise graph-based methods [43], [47]. However, as pointed out in [43], there is no reliable approach for model selection if only very few labeled points are available, i.e., it is hard to determine an optimal σ. Moreover, we observed in our experiments that even a small perturbation of σ could make the classification results dramatically different. Thus we derive a more reliable and stable way to construct the weights.

Let's consider the increment d_i of a hyperedge satisfying a standard normal distribution as shown in Eqn. (11) individually. When d_i is 0, i.e., the mode of its probability distribution, the following equation holds:

$$y_i = \sum\nolimits_{j\in\mathcal{N}_i} w_{ij}y_j. \tag{14}$$

This means intuitively that the label can be linearly constructed by its neighbors. We impose the local label consistency assumption in a linear way, i.e., the label space and the data space share the same local linear reconstruction weights. Hence, we estimate the linear construction weights by minimizing

$$E_{\mathcal{W}} = \sum\nolimits_{i\in\mathcal{V}} E_i = \sum\nolimits_{i\in\mathcal{V}} \left\|\mathbf{x}_i - \sum\nolimits_{j\in\mathcal{N}_i} w_{ij}\mathbf{x}_j\right\|^2. \tag{15}$$

This objective function is similar to the one used in locally linear embedding [34], in which it is assumed that the low dimensional coordinates share the same linear construction weights with the high dimensional coordinates. Differently, we assume that the sharing relation exists between the label space and the feature space. To avoid the negative contribution, we further constrain $\sum_{j\in\mathcal{N}_i} w_{ij} = 1$ and $w_{ij} \geqslant 0$. Usually, the more similar $\mathbf{x}_j$ is to $\mathbf{x}_i$, the larger w_{ij} will be. Thus w_{ij} can be used to measure the similarity degree from $\mathbf{x}_j$ to $\mathbf{x}_i$.

It can easily be inferred that

$$\begin{aligned} E_i &= \left\|\mathbf{x}_i - \sum\nolimits_{j\in\mathcal{N}_i} w_{ij}\mathbf{x}_j\right\|^2 \\ &= \left\|\sum\nolimits_{j\in\mathcal{N}_i} w_{ij}\left(\mathbf{x}_i - \mathbf{x}_j\right)\right\|^2 \\ &= \sum\nolimits_{j,k\in\mathcal{N}_i} w_{ij}w_{ik}\left(\mathbf{x}_i - \mathbf{x}_j\right)^T\left(\mathbf{x}_i - \mathbf{x}_k\right) \\ &= \sum\nolimits_{j,k\in\mathcal{N}_i} w_{ij}g_{jk}^i w_{ik}, \end{aligned} \tag{16}$$

where g_{jk}^i represents the (j,k)-th entry of the local Gram matrix $\mathbf{G}^i$, and $g_{jk}^i = \left(\mathbf{x}_i - \mathbf{x}_j\right)^T\left(\mathbf{x}_i - \mathbf{x}_k\right)$ at the data point $\mathbf{x}_i$. Thus the reconstruction weights of each data point can be estimated by solving the following n standard *quadratic programming* (QP) problems

$$\begin{aligned} \min_{w_{ij}} \quad & \sum\nolimits_{j,k\in\mathcal{N}_i} w_{ij}g_{jk}^i w_{ik} \\ \mathrm{s.\,t.} \quad & \sum\nolimits_{j\in\mathcal{N}_i} w_{ij} = 1, \\ & w_{ij} \geqslant 0. \end{aligned} \tag{17}$$

There exist many standard algorithms to solve these QP problems as introduced in [31]. To solve them efficiently, we adopt the *active set* algorithm [31] with a *warm start*, which usually converges in several iterations. We first compute a relaxed solution without involving nonnegative constraints using the algorithm as presented in [34]. Then, we compute a warm start by replacing the negative elements of this relaxed solution with 0. Finally, we run the active set algorithm on this warm start.

One issue that should be addressed here is that the local Gram matrix $\mathbf{G}^i$ may be singular or nearly singular when the neighbor size is larger than the dimension of a data point. In this case an extra regularization as used in [37] is required:

$$\mathbf{G}^i = \mathbf{G}^i + \epsilon\,\mathrm{trace}(\mathbf{G}^i)\mathbf{I}, \tag{18}$$

where ϵ is a small number satisfying $\epsilon \ll 1$, $\mathrm{trace}(\cdot)$ is the matrix trace operator, and $\mathbf{I}$ is an identity matrix.

3.4 Linear Neighborhood Propagation

The proposed transductive classification approach consists of the following steps: 1) building the hypergraph, 2) estimating the hyperedge weight, and 3) inferring the labeling for the unlabeled data points. Intuitively, the first two steps shear the whole graph into a series of overlapped *neighborhood* patches with the *linear* reconstruction weights, and then paste them together to form a biharmonic matrix. As mentioned before, the labeling is obtained by computing the conditional expectation as shown in Eqn. (13). Essentially, it is also equivalent to solving a biharmonic equation

$$(\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W})\mathbf{y} = \mathbf{0}, \tag{19}$$

with Dirichlet boundary conditions $\mathbf{y}_l = \bar{\mathbf{y}}_l$. Intuitively our method aims to *propagate* the labels from boundary points to the unlabeled points until an equilibrium state is reached. Based on the above intuitions, the proposed transductive classification approach is called *Linear Neighborhood Propagation* (LNP).

4 Analysis and Connections

4.1 Relation to Manifold Regularization

Most existing graph-based semi-supervised classification approaches can be cast into the *manifold regularization* framework [8]. Those approaches build a graph with pairwise edges, and impose a pairwise regularization. For example, the approaches in [8], [47] adopted the *combinatorial graph Laplacian*, and the approach in [43] used the *normalized graph Laplacian*. However, the proposed approach, LNP, goes beyond the pairwise regularization. It defines the regularizer over the hyperedges, and results in the *biharmonic regularization*. In the case that the Laplacian matrix is symmetric, it holds that $\mathbf{\Delta}^T\mathbf{\Delta} = \mathbf{\Delta}^2$, and the biharmonic operator will be the same as the p-Laplacian operator $\mathbf{\Delta}^p$ when $p = 2$. It is mentioned in [8] that the 2-Laplacian operator works well in practice, and this paper presents theoretical analysis of why it works, i.e., the biharmonic operator comes from the multiple-wise relation or the second-order relation.

4.2 Connection to Hypergraph

The second-order IGMRF, constructed for LNP, is based on the concept of the hypergraph by defining the joint weights on the hyperedges. Recently, the applications of the hypergraph in machine learning have been investigated in [3], [44]. However, there is no general approach to set the weight for the hyperedge, and the inference on the hypergraph is often difficult. In our approach, we provide a practical way to adaptively estimate the weights on the hyperedges. Moreover, we define a quadratic energy function over all the labels, which can be easily optimized.

4.3 Biharmonic and Harmonic

The harmonic approach in [47] aims at solving a Laplacian equation $\mathbf{\Delta y} = 0$ with boundary conditions given on the labeled data points. In the form of the differential operator, the combinatorial Laplacian operator $\mathbf{\Delta}$ comes from a continuous differential operator. In the two dimensional space, it is written as

$$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}. \tag{20}$$

Analogously, the construction of the second-order IGMRF can be interpreted from the biharmonic differential operator. In the two dimensional space, it is written as

$$(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})^2 = \frac{\partial^4}{\partial x^4} + 2\frac{\partial^4}{\partial x^2 \partial y^2} + \frac{\partial^4}{\partial y^4}. \tag{21}$$

On general Riemannian manifolds, the exact biharmonic operator involves an extra curvature related term, but in practice (e.g., on a local flat manifold) this term is very small and hence can be omitted. Then, we can square the discrete Laplacian operator in order to obtain the discrete biharmonic operator. Therefore, the operator, $\mathbf{Q} = \mathbf{\Delta}^T\mathbf{\Delta} = (\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W})$, can be viewed as an approximate biharmonic operator. For simplicity, we just call it a combinatorial *biharmonic* operator in this paper. By comparison, the harmonic approach is related to the equation $\mathbf{\Delta y} = \mathbf{0}$, while the biharmonic approach is related to the minimization of $||\mathbf{\Delta y}||^2$. Hence the biharmonic approach may be more relaxant. This relaxation property makes LNP potentially have the power to obtain the superior performance.

It is pointed out in [17] that the harmonic approach can guarantee the property that each connected component generated by the final classification result must contain at least one labeled point bearing the label. Unfortunately, it is not theoretically guaranteed when the biharmonic operator is used, but this property is usually observed in practice, such as in the image segmentation application.

4.4 Relation to Matting Laplacian

The *matting Laplacian* operator, a variant of graph Laplacian, is proposed in [26]. This method constructs a regularized local linear classifier for each data point using its neighborhood. The matting Laplacian approach is different from our approach in that the linear assumption of our method is on the data level, as we assume that the label space and the feature space share the same locally linear reconstruction weights. The linear assumption of matting Laplacian is on the classifier level, i.e., it assumes that the label of each data point can be predicted by a local linear classifier that is estimated using its neighborhood. From another perspective, the matting Laplacian operator is essentially an operator between the Laplacian operator and the biharmonic (biLaplacian) operator in that the order of the relation it captures is between the two operators.

4.5 Connection to Mean Field

The *mean field* approach [22] is a variational method for a complicated probabilistic inference problem. Basically, it approximates a probability $p(\mathbf{y})$ using the product of several factor probabilities,

$$q(\mathbf{y}) = \prod\nolimits_i q_i(y_i). \tag{22}$$

Then the mode of $p(\mathbf{y})$ can approximately be estimated by maximizing $q(\mathbf{y})$. For the factor probability, it can be shown that there is a closed-form solution

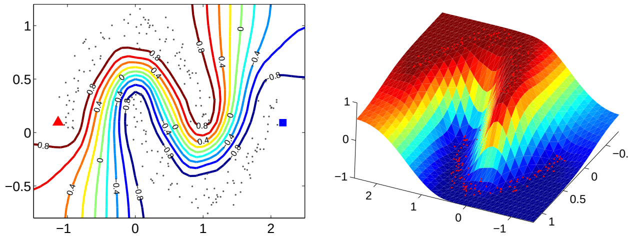
$$q_i(y_i) \sim \mathcal{N}(\mu_i, q_{ii}^{-1})$$

if $p(\mathbf{y}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{Q}^{-1})$. In this case, the mode of $p(\mathbf{y})$ is just $\boldsymbol{\mu}$. From this sense, the solution of our approach is equivalent to the mean field approach since the conditional expectation computed in Eqn. (13) is also the mean of the conditional probability.

5 Extensions

5.1 Extension to Multi-Class

In this subsection, we extend LNP to multi-class classification problems. Suppose there are c classes and the label set becomes $\mathcal{C} = \{1, 2, \cdots, c\}$. Let $\mathcal{M}$ be a set of $u \times c$ matrices with non-negative real-value entries. Any matrix $\mathbf{F}^u = [\mathbf{f}_1\ \mathbf{f}_2\ \cdots\ \mathbf{f}_c] \in \mathcal{M}$ corresponds to a specific



(a) (b)

Fig. 4. Illustration of the induction scheme for the out-of-sample data points. (a) shows the in-sample data points with two points labeled into different classes. The contour lines of the label values for the out-of-sample data points, i.e., all the data points in $\{(x, y)|x \in [-1.5, 2.5], y \in [-0.8, 1.2]\}$ are also depicted. In (b), the z-axis indicates the predicted label values associated with different colors over the region. We can see that the induced decision boundary is intuitively satisfactory since it is in accordance with the intrinsic structure of the in-sample data points.

classification on the unlabeled data points $\mathcal{U}$. We adopt the one-to-the-rest methodology to calculate $\mathbf{F}^u$. First, we precompute the matrix $\mathbf{P} = -\mathbf{Q}_{uu}^{-1}\mathbf{Q}_{ul}$. Second, we build a matrix $\bar{\mathbf{F}}^l$ for the labeled data points such that $\bar{f}_{ik}^l = 1$ if the corresponding data point is assigned a label k, and $\bar{f}_{ik}^l = 0$ otherwise. Then, we obtain $\mathbf{F}^u = \mathbf{P}\bar{\mathbf{F}}^l$. Finally, $y_i = \arg\max_{k \in \mathcal{C}} f_{ik}^u$.

5.2 Extension to Out-of-Sample Data

In this subsection, we propose a method to extend LNP to the out-of-sample data points. According to [14], we need to address two issues: 1) use the same type of regularizer for a new testing point $\mathbf{x}_o$, and 2) the inclusion of $\mathbf{x}_o$ should not affect the original labeling of the in-sample data points. Therefore, we formulate the labeling problem of the out-of-sample data point as finding the conditional expectation

$$\mathrm{E}(y_o|\mathbf{y}) = -q_o^{-1}\mathbf{q}_{on}\mathbf{y}, \tag{23}$$

where $\mathbf{y}$ is the label vector over the in-sample data points, $q_o = 1 + \sum_{k \in \mathcal{N}_o} w_{ko}w_{ko}$ is a scalar, i.e., a degraded submatrix of the joint precision matrix over the in-sample and out-of-sample data, $\mathcal{X} \cup \{\mathbf{x}_o\}$, and $\mathbf{q}_{on} = [q_{ij}]_{i=o, j \in \mathcal{L} \cup \mathcal{U}}$ is a row vector.

We validate this induction scheme in the classification problem shown in Fig. 4(a). This is a two-class classification problem with only two points labeled, one for each class. We first adopt the LNP procedure to predict the labels of the unlabeled points. Then, Eqn. (23) is used for inducing the labels of all the points in the region $\{(x, y)|x \in [-1.5, 2.5], y \in [-0.8, 1.2]\}$. The results are depicted in Fig. 4.

5.3 Incorporating the Bias

We may have the label bias for the unlabeled points. For example, we can build an external classifier from the labeled data points, and compute the compatibilities between each unlabeled data point and its possible associate labels. Those compatibilities are then combined into the proposed second-order IGMRF models. For each unlabeled data point y_u, we attach an observable node that is assigned a bias value b_u. An edge is built to connect such two nodes y_u and b_u. We define an independent increment on this edge such that

$$y_u - b_u \sim \mathcal{N}(0, 1/\lambda). \tag{24}$$

Accordingly we make the increments on the hyperedges satisfy

$$d_i \sim \mathcal{N}(0, 1/(1-\lambda)). \tag{25}$$

It is easy to show that the new graph, with the observable, labeled and unlabeled nodes, and the hyperedges and augmented edges, still forms an IGMRF $p(\mathbf{y}_l, \mathbf{y}_u, \mathbf{b}_u)$. Its precision matrix is in a form of

$$\mathbf{Q} = \begin{bmatrix} (1-\lambda)\mathbf{Q}_{ll} & (1-\lambda)\mathbf{Q}_{lu} & \mathbf{0} \\ (1-\lambda)\mathbf{Q}_{ul} & (1-\lambda)\mathbf{Q}_{uu} + \lambda\mathbf{I} & -\lambda\mathbf{I} \\ \mathbf{0} & -\lambda\mathbf{I} & \lambda\mathbf{I} \end{bmatrix}. \tag{26}$$

Then, we can compute the conditional expectation $\mathrm{E}(\mathbf{y}_u|\mathbf{y}_l, \mathbf{b}_u)$ by solving the augmented linear system $\mathbf{Q}[\mathbf{y}_l^T\ \mathbf{y}_u^T\ \mathbf{b}_u^T]^T = 0$ with $\mathbf{y}_l$ and $\mathbf{b}_u$ given.

5.4 Handling the Noisy Labeling

In some cases, the labeling on the labeled data may include some noise possibly due to the carefulness of the annotator. It would not be reasonable if we still constrain the labels on the labeled data that are definitely unchangeable. Instead, we relax such hard constraints to soft constraints, and reformulate the problem. We attach an observable node to each labeled point $\mathbf{x}_l$, denoted as n_l. n_l is valued as the corresponding label $\bar{y}_l$. Then we build an edge to connect them. An increment is defined on it such that

$$y_l - n_l \sim \mathcal{N}(0, 1/\lambda). \tag{27}$$

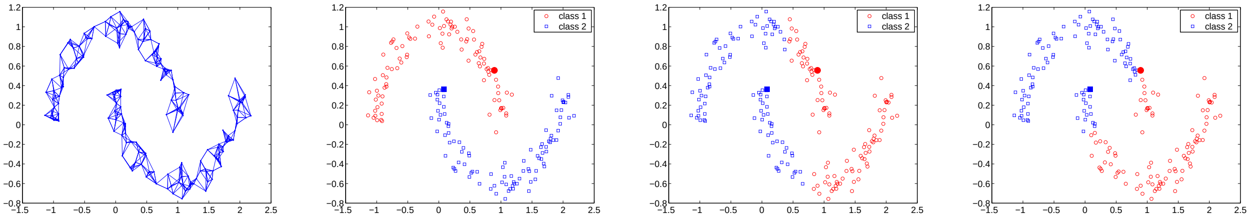
Accordingly we make the increments on the hyperedges satisfy

$$d_i \sim \mathcal{N}(0, 1/(1-\lambda)). \tag{28}$$

Then, we can easily get a new IGMRF $p(\mathbf{y}_l, \mathbf{y}_u, \mathbf{n}_l)$ over all the nodes. The precision matrix $\mathbf{Q}$ is written as

$$\begin{bmatrix} (1-\lambda)\mathbf{Q}_{ll} + \lambda\mathbf{I} & (1-\lambda)\mathbf{Q}_{lu} & -\lambda\mathbf{I} \\ (1-\lambda)\mathbf{Q}_{ul} & (1-\lambda)\mathbf{Q}_{uu} & \mathbf{0} \\ -\lambda\mathbf{I} & 0 & \lambda\mathbf{I} \end{bmatrix}. \tag{29}$$

Here, both $\mathbf{y}_l$ and $\mathbf{y}_u$ are required to be estimated with $\mathbf{n}_l$ is given. They are still estimated as the conditional expectation $\mathrm{E}(\mathbf{y}|\mathbf{n}_l)$.



(a) 4-NN graph (b) LNP (c) Nearest neighbor (d) TSVM

Fig. 5. Comparison with inductive classifiers. (a) shows the original data points with the 4-NN connected graph, (b) shows the classification result of LNP, (c) shows the result of the nearest neighbor classifier, and (d) shows the result of transductive SVM that is essentially an inductive classifier.

6 Illustration by Toy Examples

In this section, we illustrate the power of our approach with three toy examples. Subsection 6.1 tests the basic LNP method. Subsection 6.2 presents the experiment to illustrate the ability of the weight estimation scheme. Subsection 6.3 validates the noisy labeling handling method based on the precision matrix as shown in Eqn. (29) with $\lambda = 0.1$.

There are many optimization algorithms for solving the biharmonic equation with Dirichlet boundary conditions, such as the power method, the biconjugate gradients stabilized method and the coupled iterative method [31]. The optimization algorithm is not the focus of this paper. Hence, we adopt the widely-used Gaussian elimination method in the experiments.

6.1 Comparison with Transductive SVM

The data set with its four-nearest neighbor graph is shown in Fig. 5(a), which is also used in [8], [43]. This task is to separate the two moons. There are only one labeled point and 99 unlabeled points in each moon. The classification results of LNP with each point and its four-nearest neighbor points forming a hyperedge, the nearest neighbor classifier based on the Euclidean distance and transductive support vector machines (TSVMs) [20], are shown in Figs. 5(b), 5(c) and 5(d). From this example, it can be observed that LNP is capable of classifying the points correctly while the inductive approaches do not obtain accurate classifications. This shows that LNP has the capability to outperform the inductive classifiers.

6.2 Comparison with the Graph Laplacian and Harmonic Methods

This subsection presents the comparison results with the harmonic method on the glass data set as shown in Fig. 6. This data set contains 311 points with two hidden clusters shown in Fig. 6(a). The outside cluster, shaped like a parabola and composed of 11 points, is uniformly sampled from a half circle with the radius 1 and the center at point $(0, 0)$. The inside cluster, consisting of 300 points, includes two subclusters that are generated from a mixture of two Gaussians:

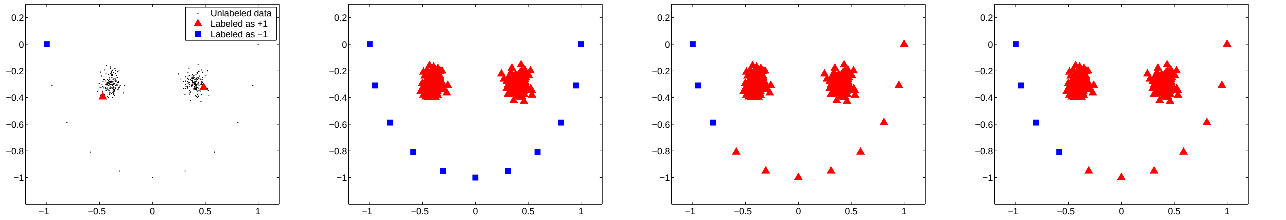
$$p(\mathbf{x}) = \frac{1}{2}\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \sigma\mathbf{I}) + \frac{1}{2}\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \sigma\mathbf{I}), \tag{30}$$

where $\boldsymbol{\mu}_1 = [-0.4, 0.3]^T$, $\boldsymbol{\mu}_2 = [0.4, 0.3]^T$, and $\sigma^2 = 0.05$. Note that the points in the inside cluster are denser than the ones in the outside cluster. The task is to separate the two hidden clusters by labeling partial points. Here, we manually labeled only three points as shown in Fig. 6(a), where the point with ■ is labeled as one cluster, and the two points with ▲ are labeled as the other cluster.

We compare our approach with the graph Laplacian method [7] and the harmonic method [47]. We tune the parameters in these two methods to obtain the best results. Figs. 6(b), 6(c) and 6(d) show the results of the three approaches, respectively. It can be observed that LNP can successfully discover the latent clusters, while the other two methods fail to capture the outside parabola shaped cluster. The success of LNP comes from the weight estimation scheme. The weight matrix $\mathbf{W}$ estimated using the linear reconstruction technique has the ability to discriminate between the clusters with different densities. This similar problem has also been addressed in [45], in which a symmetrically normalized Laplacian matrix is adopted.

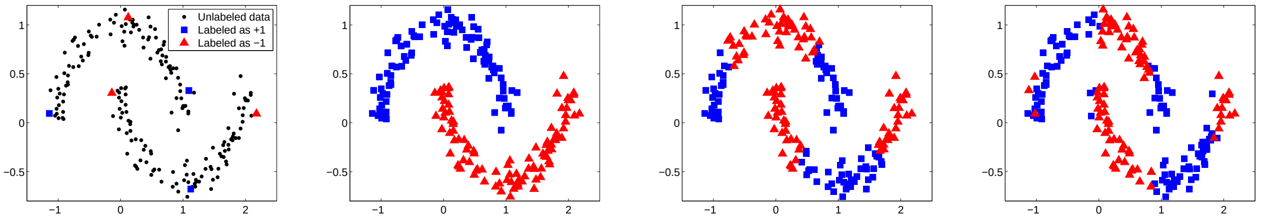
6.3 Robustness Comparison

The annotator may be prone to get the labeling on the labeled data points with some noise due to the tiredness or carelessness. But inspecting such noisy labeling needs extra human efforts and time, and hence designing a robust classifier is necessary. In this toy example, the robustness of LNP to such labeling noise is illustrated. For example, Fig. 7(a) shows the data points of two moons. Ideally, these points should be classified into two clusters. Suppose the labeling is given as shown in Fig. 7(a). It is obvious that two labeled points are not expected as our common sense would tell us, which can be viewed as labeling noise. Hence it is expected that the classifiers can handle the noisy labeling. Figs. 7(b), 7(c) and 7(d) show the results of our approach with noisy labeling handling, the nearest neighbor classifier and the harmonic approach, respectively. We can observe that all the points are correctly classified only by our approach. It is still difficult for NN to detect the noisy data since it is essentially similar to an inductive approach. The harmonic method is also unsatisfactory here in that the labels of the labeled points remain fixed and unchangeable during its optimization.



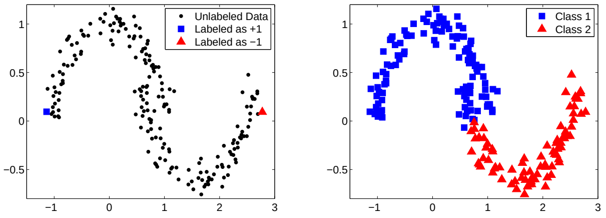
(a) Original points (b) LNP (c) Graph Laplacian (d) Harmonic

Fig. 6. Illustration of the capability of the weight estimation scheme. (a) shows the original data points with ■ and ▲ representing two classes, note that the densities of the two classes are different, (b) shows the classification result by LNP, (c) shows the result using the graph Laplacian approach, and (d) shows the result using the harmonic approach.



(a) Original points (b) LNP (c) Nearest neighbor (d) Harmonic

Fig. 7. Illustration of the robustness to the noisy labeling. (a) shows the two-moon data points, in which initially there is one point wrongly labeled on each moon, (b) shows the classification result of LNP, (c) shows the result of the nearest neighbor classifier, and (d) shows the result of the harmonic approach.



(a) (b)

Fig. 8. A failure example. We adopt a novel two-moon example, in which the points in the upper and lower moons are too ambiguous. As shown in (a), the lower moon comes closer to the upper moon. LNP fails to separate the two moons correctly.

6.4 Failure Example

In the case that the data points in different classes are too ambiguous, our method may fail. For example, a new two-moon example is shown in Fig. 8(a). In this example, the upper and low moons are too closer. The points in the right end of the upper moon and the left end of the lower moon are interweaved together. Without the moon shape prior, humans may not separate them correctly. Our algorithm has little ability to handle this case with too ambiguous points. This is because the weight estimation scheme and the biharmonic operator have no ability to discriminate between very ambiguous points.

7 Application to Image Segmentation

Image segmentation is a fundamental problem in computer vision and image processing. Although in recent years many fully automatic image segmentation methods have seen great success, the segmentation results are not as desirable as expected. However, interactive or semi-automatic image segmentation is much more practical, and has attracted more attention. The interactive image segmentation methods borrow users' assistance to perform image segmentation. Representative works include graph cuts [12], iterative graph cuts (grabcut) [33], random walker [16], [17] and lazy snapping [27]. In essence, interactive image segmentation could be cast into the semi-supervised classification framework. Naturally the proposed approach can be applied to interactive image segmentation. In this section, we present the visual and quantitative image segmentation results.

7.1 Visual Comparisons

We first give the visual comparison results on medical and natural image segmentation. We allow the user to label a few pixels (as seeds) to respectively indicate the foreground and the background. Then, we propagate the labels of the seeds to the remaining pixels. We compare our approach with graph cuts [12] and random walker [17]. Note that we use the spatially neighboring pixels as the neighborhood of a pixel to build the hyperedge, and in this experiment we use a 5×5 patch. We adopt the RGB (red, green, and blue) color values as the feature of a pixel.

Fig. 9 shows the comparison results on medical images. The original images in the first five columns are from DICOM image samples [4], and the last column is from [41]. Fig. 10 shows the comparison results on natural images. Those images are from the examples in [27] and the Berkeley segmentation dataset [28]. In the two figures, (b) shows the strokes drawn by the user in

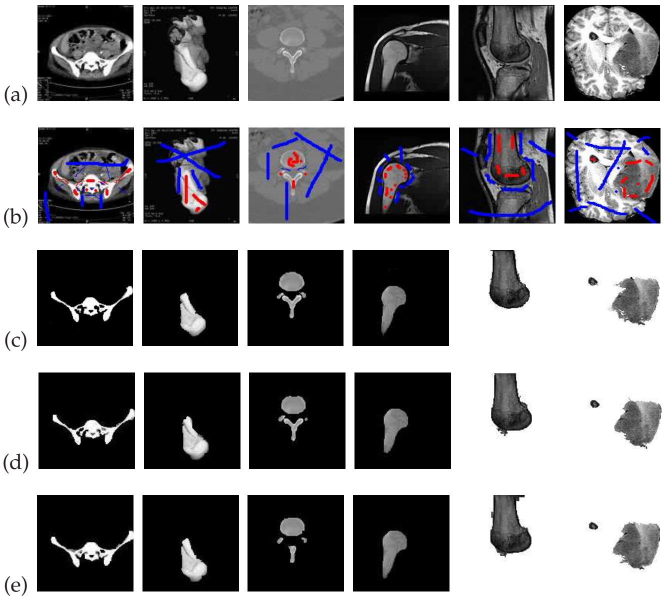


Fig. 9. Medical image segmentation results. (a) shows the original images, (b) shows the partially labeled images with the red pixels representing the foreground and the blue pixels representing the background, (c), (d) and (e) show the segmentation results from LNP, graph cuts and random walker, respectively.

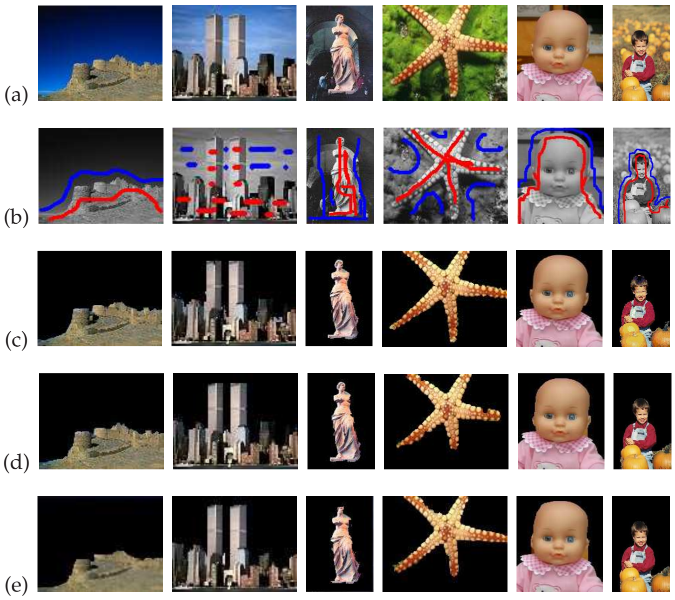


Fig. 10. Natural color image segmentation results. (a) shows the original images, (b) shows the partially labeled images with the red pixels representing the foreground and the blue pixels representing the background, (c), (d) and (e) show the segmentation results from LNP, graph cuts and random walker, respectively.

which blue and red colors indicate different segments, (c) shows the result of our approach, and (d) and (e) show the results of graph cuts and random walker. It can be seen that our results appear visually comparable and even better than the ones of graph cuts and random walker, especially on the boundaries.

7.2 Quantitative Comparisons

To obtain a quantitative comparison with existing algorithms, we perform an experiment on the image data set available from the grabcut web page [2]. This data set, including 50 images, consists of four parts: original images, manual segmentation, lasso-based labeling, and rectangle-based labeling. The manual segmentation is represented by a trimap, i.e., the definite foreground and background pixels, and a few ambiguous pixels around the boundary. The lasso-based labeling consists of a region with pixels of gray color 128, remaining for segmentation, and our task is to classify those pixels. The rectangle-based labeling is only designed for grabcut [33], and hence does not apply to our comparison. Some examples, including the original images, lasso-based labeling, and manual segmentation, are shown in Fig. 11(a), (b) and (c), respectively. We use the manual segmentation as the ground truth, evaluate the segmentation accuracy only on the definite foreground and background pixels, and compute the error ratio as the score of a segmentation algorithm.

We compare the performance of five algorithms: graph cuts (GC), random walker (RW), random walker with the weight estimation scheme described in Subsection 3.3 (RW-LLE) (equivalent to the approach in [40]), linear neighborhood propagation with the weights directly set as normalized similarities based on well tuned radius basis functions (LNP-RBF), and the proposed approach (LNP). The neighborhood size is set to one, and the parameters in RW are set as the default values in the online matlab implementation [17]. Some representative segmentation results are shown in Fig. 11. Visually, most of our results are better than the others. The boundaries of our results are more accurate and smooth, and they are much closer to the ground truths.

In addition, the *biharmonic* operator has the ability to label an ambiguous thin blob accurately. Fig. 13 shows such an example. It can be noticed that the thin white blob in the red rectangle is not segmented as the foreground with the harmonic operator (RW-LLE) as shown in Fig. 13(b) while LNP can segment this blob accurately as shown in Fig. 13(c). Another advantage is that LNP does not require that other extra parameters (except the neighborhood size) be tuned and hence leads to stable results, while the parameter for RBF should be well tuned to obtain satisfactory results. Figs. 14(b) and 14(c) show the results with different RBF parameters for LNP-RBF. We can observe that the result is very sensitive to the parameters. Fig. 14(d) shows the result of LNP without the necessity to tune any extra parameters.

We also present the error rates, and their median and mean statistics over all the 50 images of the five algorithms as shown in Tab. 1. The minimum error rate for each image is highlighted in bold font. We can observe that LNP is the best in 25 examples, and LNP-RBF is the best in 12 examples. This shows that the biharmonic

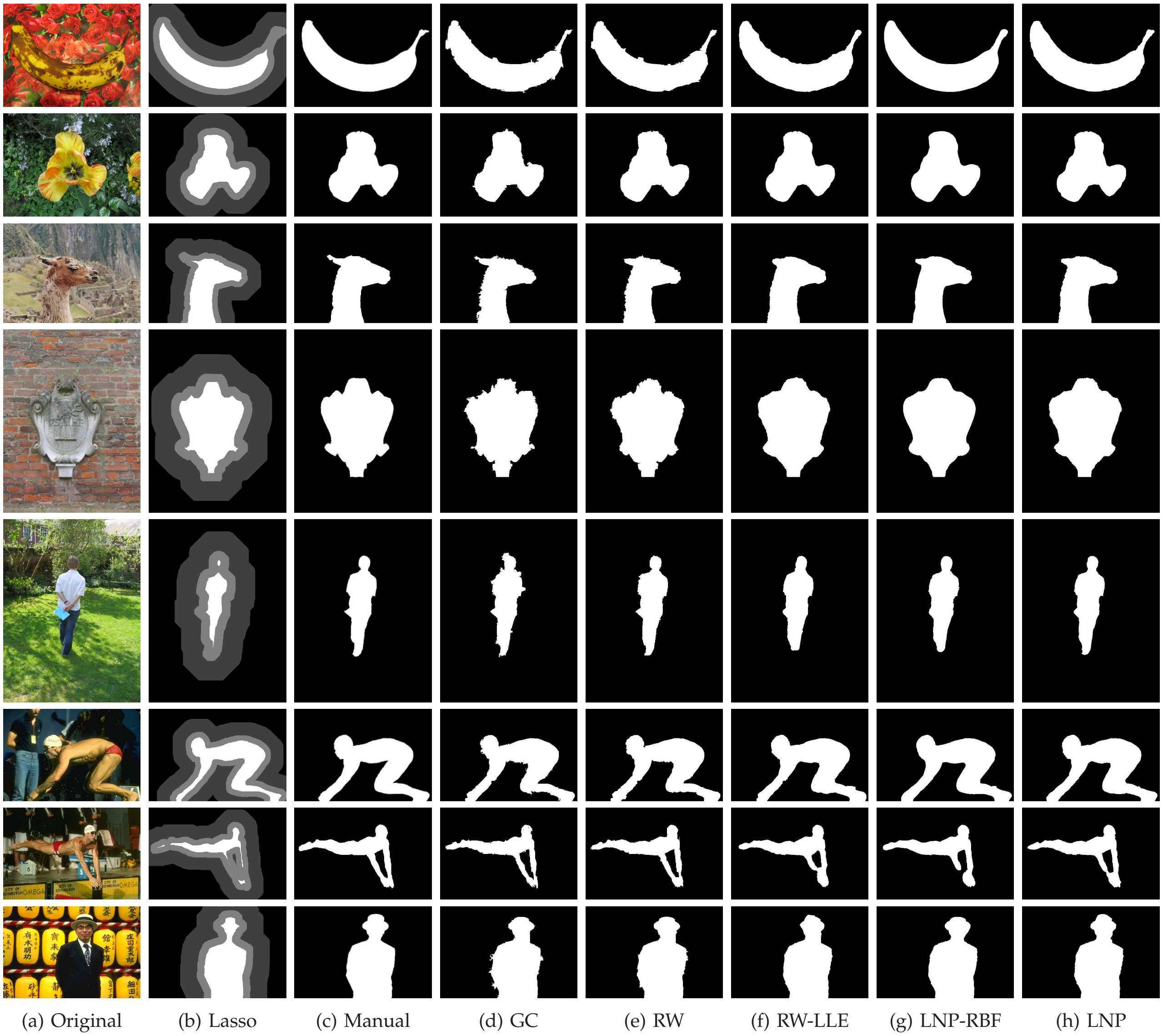


Fig. 11. Representative visual segmentation results for quantitative comparisons.

operator has the ability to obtain better performance and that the weight estimation scheme is successful.

Furthermore, we present the quantitative comparisons to illustrate the affects of the *biharmonic* operator and the *weight estimation* scheme. Given the error rates e_α and e_β of two methods α and β, we define a relative reduction ratio as $r_{\alpha\beta} = 2 \times \frac{e_\beta - e_\alpha}{e_\beta + e_\alpha}$. The larger $r_{\alpha\beta}$ is, the better method α is compared with method β. Fig. 12 shows the comparisons of the four pairs of methods over all the 50 images. The median and mean relative reduction ratios are given in Tab. 2. It can be observed that the biharmonic operator is the main factor that leads to the superiority of our approach from the two comparisons: LNP vs. RW-LLE and LNP-RBF vs. RW. The weight estimation scheme also has the ability to improve the performance.

Similar to existing approaches, LNP fails to segment the object satisfactorily when the object and the background are very similar. Fig. 15 shows such a failure case. It should be honestly pointed out that LNP is time-consuming. This is because a lot of quadratic programs need to be optimized in estimating the weights. On average, the consumed time is about 16 minutes over all the 50 testing images. Solving Eqn. (19) takes about 3 seconds for each image. However, our method may benefit from parallel computing since these quadratic programs can be optimized independently.

8 Applications to Transductive Classification

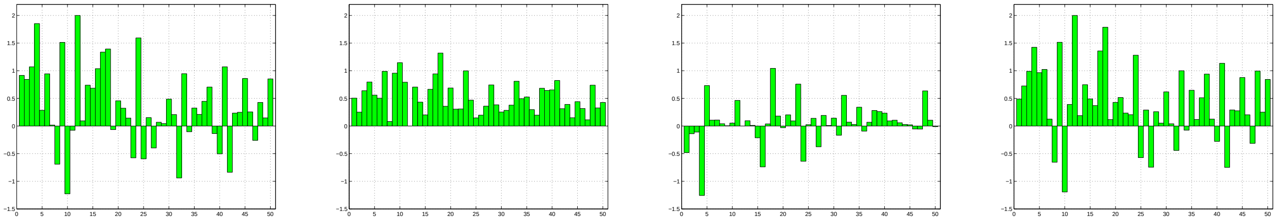
In this section, we test our approach on several transductive classification problems.

8.1 Transductive Face Recognition

LNP is performed on the Olivetti-Oracle Research Lab (ORL) face image database [36], in which there are 40 distinct faces and each object has ten gray images of size

TABLE 1
A quantitative comparison. We present the error rates of all the five algorithms on all the 50 images. The unit of the error rate is %. The minimum error rate for each image is highlighted in bold font. The abbreviated terms, GC, RW, RW-LLE, LNP-RBF, LNP, mean graph cuts, random walker, random walker with the weights estimated using the proposed weight estimation scheme, linear neighborhood propagation with the weights directly set as the normalized similarities based on the radius basis functions, and the proposed approach in this paper. The *median* and *mean* error rates over all the 50 images are also reported.

	GC	RW	RW-LLE	LNP-RBF	LNP
banana1	7.471	5.393	5.483	**2.006**	3.276
banana2	6.037	4.846	2.919	**1.971**	2.268
banana3	6.619	6.515	4.264	**1.971**	2.200
book	17.09	15.86	6.186	**0.6076**	2.664
bool	2.794	6.478	4.008	4.866	**2.258**
bush	13.93	14.07	7.603	5.052	**4.545**
ceramic	2.473	2.455	6.408	2.414	**2.167**
cross	9.954	**9.669**	20.71	19.89	19.09
doll	4.730	4.839	1.893	0.6711	**0.6682**
elefant	0.5766	**0.2922**	4.261	1.220	1.157
flower	2.567	0.8886	1.386	0.9601	**0.5992**
fullmoon	0.8387	0.2669	0.0000	0.0000	**0.0000**
grave	**7.072**	8.668	14.98	7.892	7.176
llama	14.98	15.08	10.66	6.947	**6.872**
memorial	9.286	6.499	4.823	**3.184**	3.942
music	2.742	2.355	3.232	**0.7459**	1.621
person1	12.39	13.69	7.292	2.720	**2.619**
person2	6.110	7.945	2.179	1.420	**0.4466**
person3	8.016	4.755	6.073	5.073	**4.234**
person4	11.39	6.820	9.055	**4.286**	4.420
person5	**3.582**	17.44	14.00	12.61	10.30
person6	13.00	11.42	12.32	9.890	**9.019**
person7	6.324	1.564	3.813	2.828	**1.274**
person8	8.207	6.158	2.175	**0.6967**	1.349
scissors	**6.863**	8.299	17.31	15.31	14.95
sheep	9.819	8.120	7.392	6.973	**6.068**
stone1	**0.9141**	1.022	3.217	1.528	2.236
stone2	2.324	1.089	1.833	1.018	**0.8397**
teddy	**3.435**	4.915	6.876	4.705	4.665
tennis	**10.78**	20.43	13.92	12.44	10.79
106024	12.48	12.06	15.42	**9.799**	11.59
124084	4.172	**3.252**	7.475	9.021	5.100
153077	6.301	6.216	4.894	2.224	**2.073**
153093	8.319	**6.742**	12.06	7.480	7.275
181079	13.18	12.95	11.33	9.337	**6.619**
189080	5.253	5.091	6.100	**4.123**	4.526
208001	4.879	4.216	3.047	2.678	**2.500**
209070	7.158	5.506	4.048	2.637	**1.987**
21077	4.964	4.430	7.609	5.081	**3.909**
227092	**0.9033**	3.721	9.714	6.225	4.927
24077	8.715	12.27	8.129	3.714	**3.385**
271008	**3.081**	3.518	10.60	8.586	7.732
304074	14.78	13.96	15.47	11.04	**10.42**
326038	20.54	18.83	16.58	14.67	**14.27**
37073	4.375	6.395	3.906	2.550	**2.496**
376043	13.62	10.95	12.29	**8.471**	8.935
388016	13.32	**6.606**	10.15	8.583	9.071
65019	7.813	5.989	4.367	3.885	**2.011**
69020	12.86	12.45	13.46	10.75	**9.679**
86016	5.286	7.875	4.940	**3.172**	3.208
median	6.967	6.488	6.642	4.204	**3.925**
mean	7.686	7.617	7.757	5.518	**5.108**



(a) LNP-RBF vs. RW (b) LNP vs. RW-LLE (c) LNP vs. LNP-RBF (d) LNP vs. RW

Fig. 12. Illustration of the error reduction ratios. The horizontal axis corresponds to the image index. The vertical axis represents the error reduction ratio.

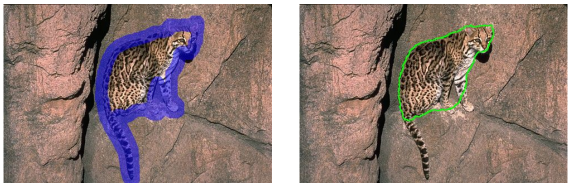
TABLE 2
Quantitative illustration of the error reduction ratios. The error reduction ratios with the unit % between LNP (LNP-RBF) and GC (RW, RW-LLE, LNP-RBF) are reported. The larger the value is, the better the method is. (a) shows the median error reduction ratios over all the 50 images, and (b) shows the mean error reduction ratios.

	LNP-RBF	LNP
GC	-	26.97
RW	26.97	37.92
RW-LLE	-	45.44
LNP-RBF	-	5.61

(a) Median error reduction ratio.

	LNP-RBF	LNP
GC	-	47.01
RW	36.82	43.49
RW-LLE	-	51.26
LNP-RBF	-	5.891

(b) Mean error reduction ratio.

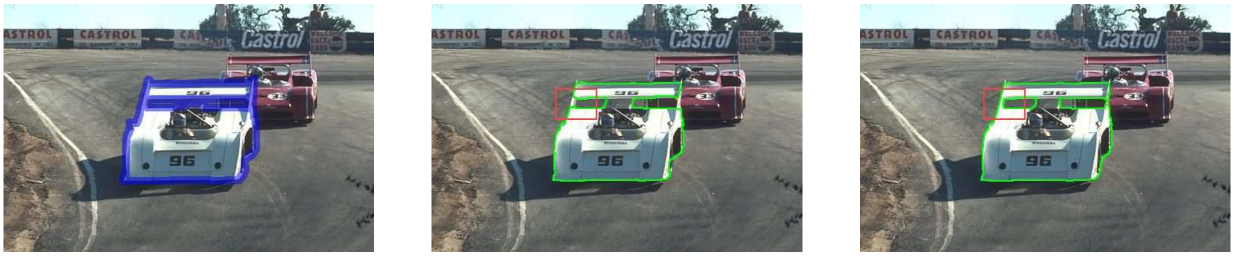


(a) Lasso (b) LNP

Fig. 15. A failure example. LNP may fail when the foreground and the background are very similar.

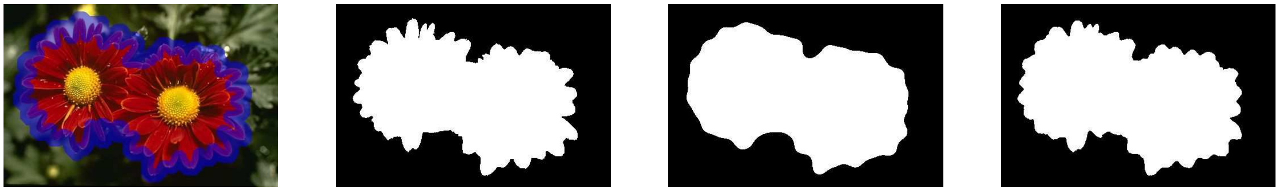
92 × 112. For computational efficiency, all the face images are downsampled to 23 × 28. Fig. 16 shows some sample face images.

By comparison, we also report the results from some other methods, including the consistency method [43] and three commonly used face recognition methods,



(a) Original image (b) RW-LLE (c) LNP

Fig. 13. Visual illustration of the superiority of the biharmonic operator over the harmonic operator. (a) shows the original image with lasso labeling colored by the blue color, (b) shows the segmentation result of RW-LLE that uses the harmonic operator, and (c) shows the result of LNP that adopts the biharmonic operator. The segmentation of the region in the red rectangle shows the performance difference. The biharmonic operator based approach, LNP, can label the thin blob accurately.



(a) Original image (b) LNP-RBF, $\sigma = 10$ (c) LNP-RBF, $\sigma = 100$ (d) LNP

Fig. 14. Comparison between LNP-RBF and LNP. (a) shows the original image, (b) and (c) show the segmentation results from LNP-RBF with the RBF parameter valued as 10 and 100, respectively. The two results are very different as LNP-RBF is very sensitive to the length scale in RBF, and (d) shows the result of LNP without the necessity to tune any extra parameter except the neighborhood size.

Fig. 16. Sample ORL face images.

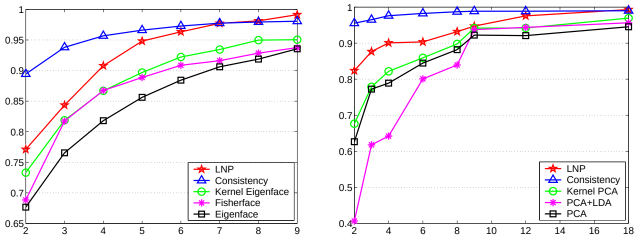
Fig. 17. Sample images in the COIL-20 database.

eigenface [38], fisherface [5] and kernel eigenface [42]. The neighborhood size in LNP and the consistency method is set to 5, and the final dimensionality of the three other methods is set to 10. The length scale of the Gaussian function in the consistency method is adjusted to achieve the best performance. The comparison results are shown in Fig. 18(a). The horizontal axis represents the number of randomly labeled face images per subject. The vertical axis is the corresponding recognition accuracy, which is an average value of 50 independent runs. The results show that LNP outperforms traditional face recognition methods and is comparable to the consistency method. It should be noted that the result of the consistency method is obtained by best tuning the parameter.

8.2 Transductive Visual Object Recognition

In this subsection, LNP is applied to a visual object recognition task on the Columbia University Image Library (COIL-20) image database [29], which consists of a set of gray-scale images with 20 objects. For each object, there are 72 images of size 128×128. The images were taken around the object at the pose interval of 5 degrees. Fig. 17 shows 20 sample images.

For comparison, we test the recognition accuracy from the consistency method and three other recognition methods [25], including Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Kernel PCA. The parameter settings of all these methods are the same as in Subsection 8.1. The only difference is that we select the images for each subject uniformly for labeling since the images for each subject are also taken uniformly around the object. The detailed procedure of the selection of the training set and the description of the three methods, PCA, LDA and Kernel PCA, can be found in [39]. Fig. 18(b) shows the recognition results. The results show that LNP outperforms traditional visual recognition methods and is comparable to the consistency method. It should also be noted that the result of the consistency method is obtained by best tuning the parameter.



(a) ORL (b) COIL

Fig. 18. Recognition accuracies on ORL and COIL. The vertical axis indicates the accuracy. The horizontal axis indicates the number of the randomly-labeled face images.

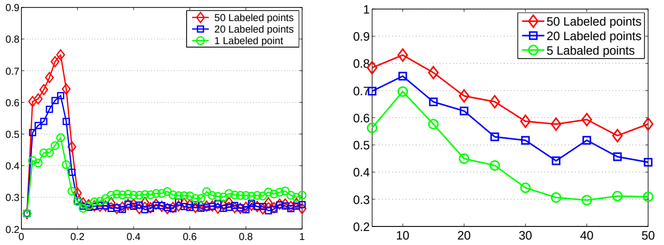
8.3 Transductive Text Classification

In this experiment, we address the task of text classification on the 20-newsgroups dataset [1]. The topic *rec*, containing autos, motorcycles, baseball and hockey, is selected from the version 20news-18828. The articles are preprocessed using the same procedure as described in [45]. Then the 3970 document vectors are of dimension 8014. Finally the document vectors are transformed into the term frequency-inverse document frequency (tf-idf) representations.

We use the inner-product based distance to find the k nearest neighbors, i.e., $d(\mathbf{x}_i, \mathbf{x}_j) = 1 - \mathbf{x}_i^T\mathbf{x}_j/(\|\mathbf{x}_i\|\|\mathbf{x}_j\|)$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are document vectors. The value of k is set to 10. For the consistency and the harmonic methods, the affinity matrices are all computed as $w_{ij} = \exp(-\frac{d(\mathbf{x}_i,\mathbf{x}_j)}{2\sigma^2})$ with $\sigma = 0.15$. The SVM and Nearest Neighbor classifiers are served as the baseline algorithms, and the length scale of the RBF kernel in SVM is set to 1.5. The accuracy vs. number of labeled points plot is shown in Fig. 19(a), where the accuracy values are averaged over 200 independent trials. From this figure, we can clearly see that LNP performs better.

We also conduct a set of experiments to compare the classification performance using different variants of graph Laplacians. The algorithms consist of LNP, the harmonic method, the consistency method, the harmonic (or random walker) method with the weight estimation scheme described in Subsection 3.3 (RW-LLE), and other biharmonic methods with different weight settings, including the square of the normalized Laplacian $\mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ (BiNLap) and the asymmetrically normalized Laplacian $\mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$ (BiANLap). From Fig. 19(b), we can see that LNP is better than other methods.

Fig. 19(c) illustrates the induction performance of LNP. We fix the number of the in-sample data points to be 1000, 2000 and 3000, and the remaining data are viewed as out-of-sample data points. It can be observed that 1000 data points are not enough for describing the structure of the whole data points as the classification accuracies are dramatically poor. However, 3000 points are sufficient for discovering this structure in that the classification accuracies are approximately equal to the accuracies achieved by LNP when all the data points are used as the in-sample data points.



(a) (b)

Fig. 20. Parameter stability. In both figures, the vertical axis represents the average recognition accuracy of 50 independent runs. (a) shows the results achieved by the consistency method, where the horizontal axis is the length scale of the RBF kernel, and (b) shows the results achieved by our LNP method, where the horizontal axis represents the size of the neighborhood.

In addition, we test the parameter stability in the consistency and the LNP methods. The experimental results are shown in Fig. 20. We may find that the consistency method is very unstable in this experiment, since it can achieve high classification accuracy only when σ falls into a very small range (between 0.1 and 0.2). By contrast, our LNP method is much more stable, and it has high classification accuracy as long as the neighborhood size is not too large and the local label consistency property can be guaranteed.

9 Conclusions

In this paper, we present a novel transductive classification approach, called Linear Neighborhood Propagation, which is novel in two aspects: graph structure construction and weight estimation. This approach can be cast into the second-order intrinsic Gaussian Markov random field framework. It is equivalent to solving a biharmonic equation with Dirichlet boundary conditions. The experiments on interactive image segmentation and transductive pattern classification demonstrate its effectiveness.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive and helpful comments. Fei Wang and Changshui Zhang were supported by NSFC Grant No. 60721003. This work was also supported by RGC/NSFC project N_HKUST602/05 and Hong Kong RGC project 619107.

References

[1] "http://people.csail.mit.edu/jrennie/20newsgroups/."
[2] "http://research.microsoft.com/vision/cambridge/i3l/segmentation/grabcut.htm."
[3] S. Agarwal, K. Branson, and S. Belongie, "Higher Order Learning with Graphs," *Proc. Twenty-Third Int'l Conf. Machine Learning*, pp. 17–24, 2006.
[4] S. Barré, "http://www.barre.nom.fr/medical/samples."
[5] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

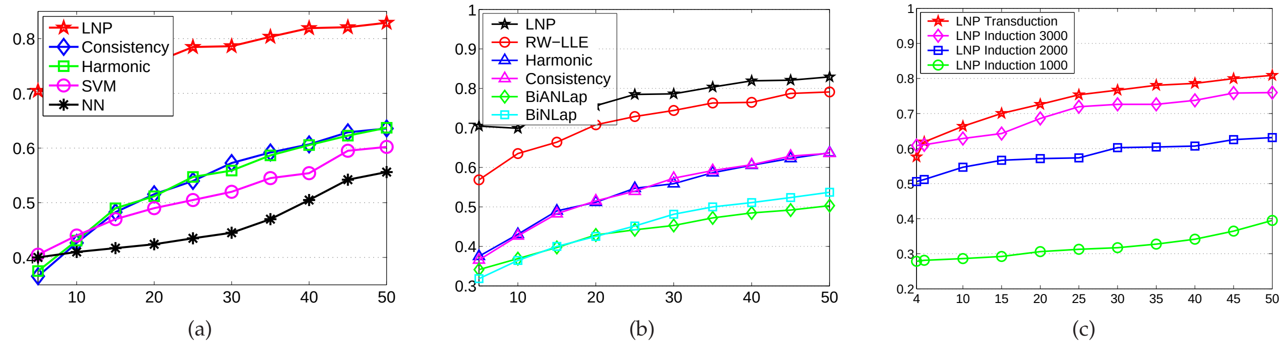


Fig. 19. Classification accuracies on the 20-newsgroup dataset. A subset of topic *rec* is adopted. (a) shows the classification accuracies for different algorithms, (b) shows the transductive classification accuracies with different types of graph Laplacians, and (c) shows the induction accuracies with 1000, 2000 and 3000 points selected as the in-sample data points and the remaining data points used as the out-of-sample data points. In the three figures, the horizontal axis represents the number of the randomly labeled data points. We guarantee that there are at least one labeled point in each class. The vertical axis is the total recognition accuracy averaged over 200 independent runs.

[6] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and Semi-supervised Learning on Large Graphs," *Proc. 17th Ann. Conf. Learning Theory*, pp. 624–638, 2004.

[7] M. Belkin and P. Niyogi, "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering," *Advances in Neural Information Processing Systems 14*, pp. 585–591, 2001.

[8] M. Belkin, P. Niyogi, and V. Sindhwani, "On Manifold Regularization," *Proc. Tenth Int'l Workshop Artificial Intelligence and Statistics*, pp. 17–24, 2005.

[9] A. Blum and S. Chawla, "Learning from Labeled and Unlabeled Data using Graph Mincuts," *Proc. Eighteenth Int'l Conf. Machine Learning*, pp. 19–26, 2001.

[10] A. Blum, J. D. Lafferty, M. R. Rwebangira, and R. Reddy, "Semi-supervised Learning using Randomized Mincuts," *Proc. Twenty-First Int'l Conf. Machine Learning*, 2004.

[11] A. Blum and T. M. Mitchell, "Combining Labeled and Unlabeled Sata with Co-Training," *Proc. Eleventh Ann. Conf. Learning Theory*, pp. 92–100, 1998.

[12] Y. Boykov and M.-P. Jolly, "Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images," *Proc. Sixth Int'l Conf. Computer Vision*, pp. 105–112, 2001.

[13] O. Chapelle, J. Weston, and B. Schölkopf, "Cluster Kernels for Semi-Supervised Learning," *Advances in Neural Information Processing Systems 15*, pp. 585–592, 2002.

[14] O. Delalleau, Y. Bengio, and N. L. Roux, "Efficient Non-Parametric Function Induction in Semi-Supervised Learning," *Proc. Tenth Int'l Workshop Artificial Intelligence and Statistics*, pp. 96–103, 2005.

[15] A. Fujino, N. Ueda, and K. Saito, "A Hybrid Generative/Discriminative Approach to Semi-Supervised Classifier Design," *Proc. Twentieth Nat'l Conf. Artificial Intelligence*, pp. 764–769, 2005.

[16] L. Grady, "Multilabel Random Walker Image Segmentation using Prior Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 763–770, 2005.

[17] ——, "Random Walks for Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.

[18] L. Grady, T. Schiwietz, S. Aharon, and R. Westermann, "Random Walks for Interactive Alpha-Matting," *Proc. Fifth IASTED Int'l Conf. Visualization, Imaging and Image Processing*, pp. 423–429, 2005.

[19] J. He, M. Li, H. Zhang, H. Tong, and C. Zhang, "Manifold-Ranking based Image Retrieval," *Proc. 12th ACM Int'l Conf. Multimedia*, pp. 9–16, 2004.

[20] T. Joachims, "Transductive Inference for Text Classification using Support Vector Machines," *Proc. Sixteenth Int'l Conf. Machine Learning*, pp. 200–209, 1999.

[21] ——, "Transductive Learning via Spectral Graph Partitioning," *Proc. Twentieth Int'l Conf. Machine Learning*, pp. 290–297, 2003.

[22] M. I. Jordan, Z. Ghahramani, T. Jaakkola, and L. K. Saul, "An Introduction to Variational Methods for Graphical Models," *Machine Learning*, vol. 37, no. 2, pp. 183–233, 1999.

[23] V. Kolmogorov and R. Zabih, "Multi-Camera Scene Reconstruction via Graph Cuts," *Proc. 7th European Conf. Computer Vision*, pp. 82–96, 2002.

[24] V. Kwatra, A. Schödl, I. A. Essa, G. Turk, and A. F. Bobick, "Graphcut Textures: Image and Video Synthesis using Graph Cuts," *ACM Trans. Graphics*, vol. 22, no. 3, pp. 277–286, 2003.

[25] J. Lee, J. Wang, C. Zhang, and Z. Bian, "Visual Object Recognition using Probabilistic Kernel Subspace Similarity," *Pattern Recognition*, vol. 38, no. 7, pp. 997–1008, 2005.

[26] A. Levin, D. Lischinski, and Y. Weiss, "A Closed Form Solution to Natural Image Matting." *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 61–68, 2006.

[27] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy Snapping," *ACM Trans. Graphics*, vol. 23, no. 3, pp. 303–308, 2004.

[28] D. R. Martin, C. Fowlkes, D. Tal, and J. Malik, "A Database of Human Segmented Natural Images and Its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics," *Proc. Eighth Int'l Conf. Computer Vision*, pp. 416–425, 2001.

[29] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia Object Image Library: COIL-20," Department of Computer Science, Columbia University, Tech. Rep.

[30] K. Nigam, "using Unlabeled Data to Improve Text Classification," Ph.D. dissertation, Department of Computer Science, Carnegie Mellon University, Pittsburgh, US.

[31] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Serials in Operations Research. London: Springer-Verlag, 2006, vol. 104.

[32] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-Supervised Self-Training of Object Detection Models," *Proc. Seventh IEEE Workshops Application of Computer Vision*, pp. 29–36, 2005.

[33] C. Rother, V. Kolmogorov, and A. Blake, ""GrabCut": Interactive Foreground Extraction using Iterated Graph Cuts," *ACM Trans. Graphics*, vol. 23, no. 3, pp. 309–314, 2004.

[34] S. T. Roweis and L. K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[35] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications*, Monographs on Statistics and Applied Probability. London: Chapman & Hall, 2005, vol. 104.

[36] F. Samaria and A. Harter, "Parameterisation of a Stochastic Model for human Face Identification," *Proc. IEEE Workshop Applications of Computer Vision*, Sarasota (Florida), December 1994.

[37] L. K. Saul and S. T. Roweis, "Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifold," *J. Machine Learning Research*, vol. 4, pp. 119–155, 2003.

[38] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[39] F. Wang, J. Wang, and C. Zhang, "Spectral Feature Analysis," *Proc. Int'l Joint Conf. Neural Networks*, pp. 1971–1976, 2005.

[40] F. Wang and C. Zhang, "Label Propagation through Linear Neigh-

borhoods," *IEEE Trans. Knowledge and Data Engineering*, vol. 20, no. 1, pp. 55–67, 2008.

[41] Q. Wu, W. Dou, Y. Chen, and J.-M. Constans, "Fuzzy Segementaion of Cerebral Tumorous Tissues in MR Images via Support Vector Machine and Fuzzy Clustering," *Proc. 11th World Congress of Int'l Fuzzy Systems Association*, 2005.

[42] M.-H. Yang, "Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition using Kernel Methods," *Proc. 5th IEEE Int'l Conf. Automatic Face and Gesture Recognition*, pp. 215–220, 2002.

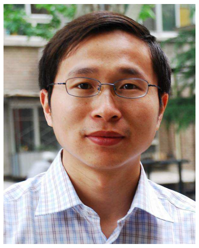
[43] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with Local and Global Consistency," *Advances in Neural Information Processing Systems 16*, 2003.

[44] D. Zhou, J. Huang, and B. Schölkopf, "Learning with Hypergraphs: Clustering, Classification, and Embedding," *Advances in Neural Information Processing Systems 19*, pp. 1601–1608, 2006.

[45] D. Zhou and B. Schölkopf, "Learning from Labeled and Unlabeled Data using Random Walks," *Proc. 26th DAGM Symposium Pattern Recognition*, pp. 237–244, 2004.

[46] X. Zhu, "Semi-Supervied Learning Literature Survey," Department of Computer Sciences, University of Wisconsin-Madison, Tech. Rep. 1530, 2006.

[47] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-Supervised Learning using Gaussian Fields and Harmonic Functions," *Proc. Twentieth Int'l Conf. Machine Learning*, pp. 912–919, 2003.



Jingdong Wang received the BSc and MSc degrees in Automation from Tsinghua University, Beijing, China, in 2001 and 2004, respectively, and the PhD degree in Computer Science from the Hong Kong University of Science and Technology, Hong Kong, in 2007. He is currently an associate researcher at the Internet Media Group, Microsoft Research Asia. His areas of interest include machine learning, pattern recognition, multimedia computing, and computer vision. In particular, he has worked on kernel methods, semi-supervised learning, data clustering, image segmentation, and image and video presentation, management and search.

Fei Wang received the PhD degree from Tsinghua University, Beijing, China, in 2008. His main research interests include graph-based learning, semi-supervised learning, and image segmentation. He served as the reviewers of many top journals and conferences, such as IEEE TPAMI, IEEE TKDE, IEEE TNN, CVPR, SIGKDD, etc. He will join the data mining group of Florida International University in September, 2008.

Changshui Zhang received the BSc degree in Mathematics from Peking University, Beijing, China, in 1986, and the PhD degree from Tsinghua University, Beijing, China, in 1992. In 1992, he joined the Department of Automation, Tsinghua University, and is currently a Professor. His interests include pattern recognition, machine learning, etc. He has authored over 200 papers. He currently serves on the editorial board of Pattern Recognition journal.

Helen C. Shen received the BMath and PhD Degrees from the University of Waterloo, Canada, in 1973 and 1982, respectively. Before joining the Hong Kong University of Science and Technology in 1992, she had been a faculty member at the University of Waterloo since 1982. Her research interests include pattern recognition, computer vision and biometrics.

Long Quan received the PhD degree in Computer Science from INPL, France, in 1989. Before joining the Department of Computer Science at the Hong Kong University of Science and Technology in 2001, he has been a French CNRS senior research scientist at INRIA in Grenoble since 1990. His research interests focus on 3D Reconstruction, Structure from Motion, and Image-based Modeling. He has served as an Associate Editor of IEEE TPAMI and a Regional Editor of IVC, and is currently an editorial board member of IJCV, ELCVIA, MVA and Foundations and Trends in Computer Graphics and Vision. He has been actively involved in conference committees of ICCV, ECCV, CVPR, and ICPR. He served as a Program Chair of ICPR 2006 Computer Vision and Image Analysis, and is a General Chair of the coming ICCV 2011 in Barcelona.