Neurocomputing 153 (2015) 41–53

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Label propagation based on local information with adaptive determination of number and degree of neighbor's similarity

Seyed Alireza Saffari *, Abbas Ebrahimi-Moghadam

Electrical Engineering Department, Ferdowsi University of Mashhad, Iran

ARTICLE INFO

Article history:
Received 14 April 2014
Received in revised form
21 November 2014
Accepted 22 November 2014
Available online 2 December 2014

Keywords:
Semi-supervised learning algorithms
Graph-based algorithms
Sparse representation
Local information

ABSTRACT

In many practical applications of machine vision, a small number of samples are labeled and therefore, classification accuracy is low. On the other hand, labeling by humans is a very time consuming process, which requires a degree of proficiency. Semi-supervised learning algorithms may be used as a proper solution in these situations, where ε-neighborhood or k nearest neighborhood graphs are employed to build a similarity graph. These graphs, on one hand, have a high degree of sensitivity to noise. On the other hand, optimal determination of ε and k parameters is a complex task. In some classification algorithms, sparse representation (SR) is employed in order to overcome these obstacles. Although SR has its own advantages, SR theory in its coding stage does not reflect local information and it requires a time consuming and heavy optimization process. Locality-constrained Linear Coding (LLC) addresses these problems and regards the local information in the coding process. In this paper we examine the effectiveness of using local information in form of label propagation algorithm and present three new label propagation modifications. Experimental results on three UCI datasets, two face databases and a biometric database show that our proposed algorithms have higher classification rates compared to other competitive algorithms.

1. Introduction

Data classification is of interest to machine learning researchers. Many classification algorithms are offered and developed by the researchers. Usually, in these classification algorithms for test or unlabeled samples, we use training or labeled samples. Unfortunately, when the number of training samples is far less than the test samples, these methods perform poorly. In many practical applications in machine learning, the number of labeled samples is quite low, while a large number of samples are unlabeled. Therefore, a large number of samples must be labeled and used as training samples. Labeling process by human is a time consuming task, which requires skilled hand work. In this condition an appropriate approach is to employ both labeled and unlabeled samples for data classification. In semi-supervised learning, which is an active topic in machine vision [1–10], labeled and unlabeled samples are both employed. Since many unlabeled samples can be gathered only by measuring them without interpretation, semi-supervised learning methods are very useful. These methods are divided into two main groups. In the first group, we only estimate labels of unlabeled samples [11,12]. These methods are known as "*transductive algorithms*". In the second group of methods, known as "*inductive algorithms*" [13], a decision function with very low error-rate for all samples (labeled and unlabeled) is sought. Another semi-supervised learning method, which has been studied widely, is graph-based semi-supervised learning. In these algorithms, the knowledge of the mutual data similarity is represented by graphs. In this regard, graph $G=(V,E)$ in which vertex set V includes all labeled and unlabeled samples and edge set E which contains similarity between data corresponding to vertex set of that edge set is considered. The graph is called similarity graph. Different types of graph-based methods by defining different similarity graphs can be introduced all of which have the same goal of modeling the relationship between sample point and its neighbors. Two conventional similarity graphs are ε-neighborhood graph and k nearest neighborhood graph. In ε-neighborhood graph, vertices of each pair of samples that have distance less than ε are connected to each other. In k nearest neighborhood graph, corresponding vertex of samples that belong to one of k nearest neighborhood are connected to each other. A semi-supervised learning method can be defined as a mincut problem [14].

Label propagation methods, which propagate labels of the training samples to test samples [15–20], are among semi- supervised learning methods. In consistency method [15] Gaussian kernel is employed to determine edge weights. In fact, in this algorithm, edge weights are determined using $e_{ij}=\exp(-||x_i-x_j||^2/2\sigma^2)$, $i\neq j$ and $e_{ii}=0$. In [16,17] k nearest neighborhood graph is used as similarity graph.

* Corresponding author.
E-mail address: arsaffari@yahoo.com (S.A. Saffari).

http://dx.doi.org/10.1016/j.neucom.2014.11.053
0925-2312/

After determining k nearest neighbors for each sample, that sample is expressed as a linear combination of its neighbors and the weight vector obtained by this method is considered as edge weights. Usually, in semi-supervised learning methods, center of attention is on graph structure and weights of edges are defined separately. ε-neighborhood and k nearest neighbors' graphs, which are usually used in these methods, have the following disadvantages: 1- These graphs are constructed using pair-wise Euclidean distance which is very sensitive to noise. 2- Considering different sample distributions, to determine proper neighbors for each sample, ε and k must be defined adaptively for each sample, but in ε-neighborhood and k nearest neighbors' graphs, a fixed parameter is considered for all samples so the accuracies of these graphs are very low.

Recently sparse representation (SR) has found various applications in machine vision and statistical pattern recognition [21–23]. In [24], l_1-graph, which is based on SR, is employed to produce a graph based algorithm. In l_1-graph, graph structure and weights of edges are found simultaneously using l_1-minimization. SR graph based algorithms have the following attributes compared to the other graph-based algorithms: First, in SR graph-based algorithms, graph structure and edge weights are found simultaneously by l_1-norm minimization. Second, since Euclidean distance is not employed, SR graph-based algorithms have lower degrees of sensitivity to noise. And third, in SR graph-based algorithms the number and degree of similarity of samples to each other are determined adaptively, hence there is no metric to determine the number and the degree of similarity between each sample and its neighbors. So efficiency of SR graph-based algorithms is much higher than other graph based methods which are explained so far in this paper.

We can group graph based semi-supervised learning algorithms which use SR theory in two categories. First category is the graph reduction semi-supervised learning methods [25,26]. Usually when the size of test samples grow, graph based semi-supervised learning methods have two major weaknesses: (1) Possible outliers and noisy samples have negative effect on the construction of the similarity graph. (2) The evaluation of predictors learned from the graph for new samples can be time-consuming if the predictors involve computations on all the samples in the original graph. To solve these problems graph reduction semi-supervised learning methods were introduced. In [25] a graph reduction method based on manifold-preserving sparse graphs has been proposed, where the number of vertices is reduced while the edge weights from the original graph are remained unchanged. In [26] a sparse semi-supervised learning framework using Fenchel-Legendre conjugates is proposed. The main focus of [26] is to reduce the number and to choose the appropriate unlabeled samples. The purpose of the second category of the graph based semi-supervised learning algorithms which use SR theory is graph construction [27,28]. Although there has been a numerous graph based semi-supervised learning methods, there are still much to do about neighbor selection and the degree of their similarities for each sample. In [3] a semi-supervised classification algorithm called the Sparse Regularized Least Square Classification (S-RLSC) algorithm. In [4] a semi-supervised classification approach through kernel-based sparse representation is proposed. This method computes the sparse representation of data in the feature space, and then the learner is subject to a cost function which aims to preserve the sparse representing coefficients. Our proposed method in this paper can also be categorized into graph construction methods but based on LLC and not the SR method.

Although numerous works have been done in machine vision based on the SR theory [29–32], little is said about the shortcomings of SR based graphs. Recently, a type of signal representation namely Locality-constrained Linear Coding (LLC) [33] is introduced in which local information is utilized, instead of sparsity constraint. Various studies make use of local information in order to enhance the learning efficiency, like feature reduction [34,35], density estimation [36], anomaly detection [37] and data classification [38,39]. k nearest neighbor classifier is the most familiar instance for local information usage. In SR-based algorithm, due to over completeness of the dictionary matrix, sometimes the samples selected as neighbors are not actually close to the related sample; the sparsity constraint in SR is what that has forced choosing those samples as neighboring samples. In fact, SR does not preserve the samples' local information during the coding process. The second problem of SR-based algorithm is the absence of an analytical trouble-free solution. Solving SR requires a time consuming optimization process. In LLC, local information constraint is employed instead of sparsity constraint and simple analytical solution exists. In this paper, we study the efficiency of using local information, like in LLC, in a form of label propagation algorithm. Experimental results show that our proposed algorithms have better classification rate compared to the other label propagation methods.

The rest of this paper is organized as follows: in Section 2, after a brief review on SR theory and LLC coding, some label propagation algorithms are investigated. Our proposed algorithms are introduced in the third section of this paper. The fourth section contains results of our experiments. Finally in Section 5 conclusion and future works are discussed.

2. Review on related works

In this section first a brief review on SR theory is presented. Then LLC, in which the local information is used instead of sparsity constraint, is briefly reviewed and finally a number of label propagation algorithms are introduced.

2.1. SR theory in machine vision

In recent years, sparse representation has caught researcher's attention in different fields. Based on this representation, several algorithms have been introduced. One of the first algorithms, introduced in statistical pattern recognition based on SR theory, is Sparse Representation based Classification (SRC) [29], which provides excellent accuracy in face data classification. SRC is based on a simple assumption that samples of a specific class are in the same sub-space. As a result, a test sample can be well represented by the training samples of its own class. In SRC, each test sample is expressed by a sparse linear combination of all training samples, and the non-zero elements of a coefficient vector are expected to point to a specific class.

Assume that dictionary matrix

$$\mathbf{D} = \left[\mathbf{x}_1, \ldots, \mathbf{x}_\ell, \mathbf{x}_{\ell+1}, \ldots, \mathbf{x}_n\right] \in \mathscr{R}^{d\times n}$$

contains all labeled and unlabeled samples; first ℓ samples contain labeled and the rest are unlabeled samples. Each sample can be expressed as linear combination of other samples. When the samples are numerous, the resulting weight vectors are sparse, i.e., many of their elements are zero. In this condition, optimal weight vector can be found with the aid of SR theory. SR theory can be defined as the following optimization problem:

$$\underset{\mathbf{w}_i}{\operatorname{argmin}} \|\mathbf{w}_i\|_0 \ \ s.t. \ \ \|\tilde{\mathbf{D}}\mathbf{w}_i - \mathbf{x}_i\|_2^2 \le \varepsilon \tag{1}$$

where $\mathbf{w}_i$ is weight vector for i^{th} sample and $\tilde{\mathbf{D}} \in \mathscr{R}^{d\times(n-1)}$ is a dictionary matrix in which i^{th} sample is omitted. Moreover $\|\cdot\|_0$ represents ℓ_0- norm of $\mathbf{w}_i$ that gives the number of non-zero elements of $\mathbf{w}_i$. In fact, Eq. (1) finds $\mathbf{w}_i$ vector such that in addition to satisfying $\|\tilde{\mathbf{D}}\mathbf{w}_i - \mathbf{x}_i\|_2^2 \le \varepsilon$, number of its non-zero elements are minimum and $\mathbf{w}_i$ vector is sparse. But since $\|\cdot\|_p$

for $0 \le p < 1$ is neither convex nor differentiable, solving Eq. (1) is an NP-hard problem [40]. Since ℓ_1- norm ($\|\cdot\|_1$) is closest convex form to zero norm, replacing zero norm by one norm looks a reasonable estimation and Eq. (1) can be rewritten as:

$$\underset{\mathbf{w}_i}{\operatorname{argmin}} \|\mathbf{w}_i\|_1 \ \ s.t. \ \ \|\tilde{\mathbf{D}}\mathbf{w}_i - \mathbf{x}_i\|_2^2 \le \varepsilon \tag{2}$$

where there are many algorithms to solve it [35–43]. The above relationship can be rewritten as:

$$\underset{\mathbf{w}_i}{\operatorname{argmin}} \|\tilde{\mathbf{D}}\mathbf{w}_i - \mathbf{x}_i\|_2^2 + \lambda\|\mathbf{w}_i\|_1 \ , \ i = 1, 2, \ldots, n \tag{3}$$

Expression $\|\tilde{\mathbf{D}}\mathbf{w}_i - \mathbf{x}_i\|_2^2$ suggests a type of signal representation in which each sample is represented in terms of other samples. $\|\mathbf{w}_i\|_1$ is the sparsity constraint of the weight and λ is the sparsity adjustment parameter of the weight vector. In case a similar assumption like SRC is considered, i.e., samples of a specific class are located in the same sub-space, $\mathbf{w}_i$ (optimum weight vector) includes positive and negative elements and coefficient related to the samples of other classes (other than the class that contains i^{th} sample) are zero. Considering the constraint of optimum weight vector coefficient being positive, we can consider these coefficients as similarity degree of samples to each other. As a result, number and degree of similarity of neighbors are determined by SR theory, adaptively. This means:

$$\underset{\mathbf{w}_i}{\operatorname{argmin}} \|\tilde{\mathbf{D}}\mathbf{w}_i - \mathbf{x}_i\|_2^2 + \lambda\|\mathbf{w}_i\|_1 \quad s.t. \quad \mathbf{w}_{i,j} \ge 0$$
$$i = 1, 2, \ldots, n \ , \quad j = 1, 2, \ldots, n-1 \tag{4}$$

where $\mathbf{x}_i$is i^{th} sample, $\widehat{\mathbf{w}_i} = [\mathbf{w}_{i,1}, \ldots, \mathbf{w}_{i,n-1}] \in \mathscr{R}^{n-1}$ is weight vector corresponding to $\mathbf{x}_i$ and $\tilde{\mathbf{D}} \in \mathscr{R}^{d\times(n-1)}$ is dictionary matrix in which i^{th} sample is omitted. This equation can be solved through non-negative sparse representation algorithm (NSR). Weight vector resulted from solving Eq. (3) can be used as edge weight in graph based algorithms. ℓ_1- graph algorithm employs this idea to assign weight to edges.

2.2. Locality-constrained linear coding (LLC)

Although many applications of SR are found in machine learning, the SR theory conveys some shortcomings. First, due to over-completeness of the SR dictionary matrix and weight vector's sparsity constraint, it is likely that samples selected by SR as neighbors are not in the neighborhood of that sample. Second, SR solution algorithms are very computationally expensive. Third, SR algorithm does not preserve local information in the coding process. In LLC, to consider local information, sparsity constraint in SR algorithm, is modified by a locality adaptor. LLC is defined as:

$$\underset{\mathbf{w}_i}{\operatorname{argmin}} \|\tilde{\mathbf{D}}\mathbf{w}_i - \mathbf{x}_i\|_2^2 + \lambda\|\mathbf{p}_i \ \odot \ \mathbf{w}_i\|_2^2$$
$$s.t. \ \ \mathbf{1}^T\mathbf{w}_i = 1 \ , \quad i = 1, 2, \ldots, n \tag{5}$$

$\odot$Sign in (5) indicates element-wise multiplication and $\mathbf{p}_i$ is a local adaptor in a vector form in which its kth element is the distance between ith and kth samples. Weight vector obtained by LLC is sparse in some extent. Elements of the weight vector from LLC are zero when the corresponding sample to those elements is far from the ith sample and as a result this weight vector becomes sparse. The shift invariance constraint of $1^T\mathbf{w}_i$ is considered in order to cancel the effect of changing the coordinate center on the selection of weight vector$\mathbf{w}_i$.

2.3. Related label propagation algorithms

Among graph based semi-supervised learning algorithms are label propagation methods, which directly propagate labels of the training samples to the test samples. Linear neighborhood propagation (LNP) algorithm is the simplest label propagation algorithm in which weights of the edges are calculated using k nearest neighborhood graph. Due to disadvantages of this graph, which were briefly explained in the introduction section, a good way to determine the weight of the edges is to exploit SR theory. ℓ_1- graph and LPSN [44] are two graph-based algorithms based on the SR theory. In this part of the paper, after introducing LNP, LPSN algorithm is briefly reviewed.

2.3.1. LNP algorithm

Assume that all the samples are contained in the dictionary matrix of $\mathbf{D} = [\mathbf{x}_1, \ldots, \mathbf{x}_\ell, \mathbf{x}_{\ell+1}, \ldots, \mathbf{x}_n] \in \mathscr{R}^{d\times n}$. The first ℓ samples($\mathbf{x}_i, i \le \ell$), are labeled sample with labels y_i and the rest of samples ($\mathbf{x}_u, \ell+1 \le u \le n$)are unlabeled samples.

In LNP, first k – nearest neighbors of each sample are determined using a distance metric. This means$\forall \ \mathbf{x}_i \in \mathbf{D}$, the set of $N\{\mathbf{x}_i\}$ containing k nearest neighbors of the sample $\mathbf{x}_i$ is defined. In fact in LNP, k nearest neighbor graph is used to determine graph formation. In this graph, the vertices corresponding to the k nearest neighbors of a sample are connected. To determine edge weights in LNP the following minimization is used:

$$\min_{\mathbf{w}_i} \varepsilon_i = \min_{\mathbf{w}_i} \|\mathbf{x}_i - \sum_{j:\mathbf{x}_j \in N\{\mathbf{x}_i\}} \mathbf{w}_{ij}\mathbf{x}_j\|_2^2$$
$$s.t. \begin{cases} \mathbf{w}_{ij} = 0 \ ; \ \forall \mathbf{x}_j \notin N\{\mathbf{x}_i\} \\ \sum_{j:\mathbf{x}_j \in N\{\mathbf{x}_i\}} \mathbf{w}_{ij} = 1 \ , \ \mathbf{w}_{ij} \ge 0 \end{cases} \tag{6}$$

where $\mathbf{x}_j \in N\{\mathbf{x}_i\}$ includes all k nearest neighbors of sample $\mathbf{x}_i$. After determining the edge weights, label propagation process is done as follows:

$$\min_{\mathbf{F}} \sum_{i=1}^{n} \|\mathbf{f}_i - \sum_{j:\mathbf{x}_j \in N\{\mathbf{x}_i\}} \mathbf{w}_{ij}\mathbf{f}_j\|_2^2$$
$$s.t. \ \ \mathbf{f}_i = \mathbf{y}_i \ \ 1 \le i \le \ell \ , \ \ 1 \le j \le c \tag{7}$$

In which, $\mathbf{F} = \left(\mathbf{f}_1{}^T, \mathbf{f}_2{}^T, \ldots, \mathbf{f}_n{}^T\right) \in \mathscr{R}^{n\times c}$ and c is the number of classes. If the label of the labeled sample is j, $f_{ij} = 1$, otherwise $f_{ij} = 0$. For unlabeled samples$f_{uj} = 0$, $1 \le u \le c$, then the label propagation process is repeated c time and the resulting $\mathbf{f}_i$ in each stage is considered as the input to the next stage.

Similar to LNP, graph-based semi-supervised learning methods usually focus on graph structure and weights obtained for edges are defined separately. ε -Neighborhood and k nearest neighbors graphs which are usually used in these methods are constructed using pair-wise Euclidean distance and therefore highly sensitive to noise. Besides, considering different sample distributions, to determine proper neighbors for each sample, ε and k must be determined adaptively and different for each sample, but in $\varepsilon-$ Neighborhood and k nearest neighbors graphs, a fixed parameter for each sample is considered and as a result the accuracy of the graphs is very little.

For example, when k is small, only local information of sample space is reflected and the global information is not considered and when $k = k_{\max}$, global information is used and the local information is not reflected. As a result, finding an optimal parameter for the neighbor numbers is different and requires a complete search. The most recent method to solve this problem is using SR method.

In the next section, a label propagation algorithm based on SR is presented.

2.3.2. LPSN algorithm

In graph-based algorithms using SR theory, graph structure and edge weights are obtained simultaneously by l_1-norm minimization. l_1-graph and LPSN are two examples of these algorithms. In comparison to the other graph-based algorithms, algorithms constructed based on graph-based sparse representation have the following distinctions: First, in these methods, graph structure and

edge weights are obtained simultaneously by solving l_1-optimization problem. Second, due to Euclidean distance usage, these algorithms have higher sensitivity to noise. Third, in these algorithms, number and degree of similarity of samples, are adaptively calculated using SR, hence, the efficiency of these algorithms is much higher than other graph-based algorithms.

In l_1-graph each sample (labeled and unlabeled) is represented as a linear combination of other samples. So, with the assumption of high number of samples, the weight vector is sparse and can be calculated using l_1-norm optimization problem. The sparse vector has negative and positive components. If we confine the vector components to be positive in the optimization process, this vector can be looked as a level of mutual similarity of samples. This is attainable by Non-negative Sparse Representation or NSR in short.

Contrary to l_1-graph, in LPSN, which is a label propagation method based on SR, each sample is initially represented by a linear combination of other samples. Then, after determining the neighbors, that sample is represented as a linear combination of the neighbors. The weight vector attained in this stage is introduced as the edge weights in the similarity graph. Neighbors determined at the first stage are known as Sparse Neighborhood (SN). To obtain sparse neighborhood, the following equation is considered:

$$\underset{\boldsymbol{\alpha}_i}{\operatorname{argmin}} \|\tilde{\mathbf{D}}\boldsymbol{\alpha}_i - \mathbf{x}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_1 \qquad i = 1, 2, \ldots, n \tag{8}$$

where $\mathbf{x}_i$ is the ith sample, $\widehat{\boldsymbol{\alpha}_i} = [\boldsymbol{\alpha}_{i,1}, \ldots, \boldsymbol{\alpha}_{i,n-1}] \in \mathscr{R}^{n-1}$ is weight vector of $\mathbf{x}_i$ and $\tilde{\mathbf{D}} \in \mathscr{R}^{d\times(n-1)}$ is a dictionary matrix in which the ith training sample is omitted. Considering $\varepsilon > 0$, sparse neighborhood of $\mathbf{x}_i$ is defined as: if $\alpha_{i,j} > \varepsilon$ then sample $\mathbf{x}_j$ is sparse neighborhood of$\mathbf{x}_i$. This way we define a set of $\mathbf{D}^N = N\{\mathbf{x}_i\}_{i=1}^n \in R^{k\times d}$ including all sparse neighborhood of ith sample in which k is the number of these neighbors.

To find the edge weights the following expression is used:

$$\underset{\mathbf{w}_i}{\operatorname{argmin}} \|\mathbf{D}^N\mathbf{w}_i - \mathbf{x}_i\|_2^2 + \lambda\|\mathbf{w}_i\|_1 \qquad i = 1, 2, \ldots, n \tag{9}$$

where $\mathbf{x}_i$ is the ith sample, and $\mathbf{D}^N \in \mathscr{R}^{k\times d}$ is the dictionary matrix containing all the sparse neighborhoods. So, at this stage, number of neighbors and their extent of similarity are adaptively determined by the SR theory.

After edge weights calculation in LPSN, in order to estimate the labels of the unlabeled samples, the subsequent objective function is used, similar to the one used in LNP:

$$\min_{\mathbf{Y}} \sum_{i=1}^{n} \|\mathbf{y}_i - \sum_{j:\mathbf{x}_j \in N\{\mathbf{x}_i\}} \mathbf{w}_{ij}\mathbf{y}_j\|_2^2 \qquad 1 \le i \le \ell \quad , \quad 1 \le j \le c \tag{10}$$

In methods based on the SR theory, as a consequence of the over-completeness of the dictionary matrix, it is possible that some samples determined as neighbors are not really in the neighborhood of the corresponding sample due to the force of the sparsity constraint. In fact, SR does not preserve local information of the samples; therefore, the sparse neighborhoods may not be properly effective. The second problem of the SR based algorithms is the absence of a simple analytic solution for them. SR solution requires a heavy and time consuming optimization process. In LLC, instead of sparsity, local information constraint is utilized and a simple analytic solution is obtained. In this work, we investigate the efficiency of using local information, like LLC, in a form of label propagation algorithm.

3. Proposed algorithm

In this section, construction of a graph using LLC is investigated, and two linear algorithms and a kernel algorithm for label propagation base on LLC are presented.

3.1. Similarity graph construction based on LLC

Assume $\mathbf{D} = [\mathbf{x}_1, \ldots, \mathbf{x}_\ell, \mathbf{x}_{\ell+1}, \ldots, \mathbf{x}_n] \in \mathscr{R}^{d\times n}$, dictionary matrix, includes all samples in such a way that first ℓ samples $\mathbf{x}_i, i \le \ell$, are labeled and the rest $\mathbf{x}_u, \ell+1 \le u \le n$ are unlabeled samples. A label matrix of all samples is defined as $\mathbf{Y} = [\mathbf{Y}_\ell; \mathbf{Y}_u]$, where, $\mathbf{Y}_\ell$ and $\mathbf{Y}_u$are label sub matrices of labeled and unlabeled samples, respectively. Our goal is to properly estimate$\mathbf{Y}_u$. To do so, we use the follow minimization objective function:

$$\min_{\mathbf{Y}} \sum_{i=1}^{n} \|\mathbf{y}_i - \sum_{j:\mathbf{x}_j \in N\{\mathbf{x}_i\}} \mathbf{w}_{ij}\mathbf{y}_j\|_2^2 \tag{11}$$

where, $\mathbf{y}_i \in \mathscr{R}^{1\times c}$ is the label vector whose components are the probability of the i^{th} sample association to different classes. Based on this objective function, each sample's label can be a linear combination of its neighbors' labels. Using linear algebra, we can find a simple solution to the above optimization problem:

$$\begin{aligned}\min_{\mathbf{Y}} \sum_{i=1}^{n} \|\mathbf{y}_i - \sum_{j:\mathbf{x}_j \in N\{\mathbf{x}_i\}} \mathbf{w}_{ij}\mathbf{y}_j\|_2^2 &= \|\mathbf{Y}(\mathbf{I}-\mathbf{W})\|_2^2 \\ &= tr\left(\mathbf{Y}^T(\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})\mathbf{Y}\right) = tr\left(\mathbf{Y}^T\mathbf{M}\mathbf{Y}\right)\end{aligned} \tag{12}$$

where $\mathbf{I}$ is an identity matrix and $\mathbf{M} = (\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})$ is a symmetric matrix. Matrix $\mathbf{M}$ is divided to 4 sub matrices as:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{LL} & \mathbf{M}_{LU} \\ \mathbf{M}_{UL} & \mathbf{M}_{UU} \end{bmatrix} \tag{13}$$

Hence Eq. (12) is written as:

$$tr\left(\mathbf{Y}^T\mathbf{M}\mathbf{Y}\right) = tr\left(\begin{bmatrix} \mathbf{Y}_\ell \\ \mathbf{Y}_u \end{bmatrix} \begin{bmatrix} \mathbf{M}_{LL} & \mathbf{M}_{LU} \\ \mathbf{M}_{UL} & \mathbf{M}_{UU} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_\ell & \mathbf{Y}_u \end{bmatrix}\right) \tag{14}$$

By taking derivative with respect to $\mathbf{Y}$ and putting equal to zero we have:

$$\begin{cases} \mathbf{M}_{LL}\mathbf{Y}_\ell + \mathbf{M}_{LU}\mathbf{Y}_u = 0 \\ \mathbf{M}_{UL}\mathbf{Y}_\ell + \mathbf{M}_{UU}\mathbf{Y}_u = 0 \end{cases} \Rightarrow \mathbf{Y}_u = -\mathbf{M}_{UU}^{-1}\mathbf{M}_{UL}\mathbf{Y}_\ell \tag{15}$$

So, by knowing labels of the labeled samples ($\mathbf{Y}_\ell$), an estimation of $\mathbf{Y}_u$ labels are attainable. To make similarity graph using LLC we exploit LLC objective function as follows:

$$\underset{\mathbf{w}_i}{\operatorname{argmin}} \|\mathbf{x}_i - \tilde{\mathbf{D}}\mathbf{w}_i\|_2^2 + \lambda\|\mathbf{p}_i \odot \mathbf{w}_i\|_2^2 \qquad s.t.\ \mathbf{1}^T\mathbf{w}_i = 1 \tag{16}$$

where $\mathbf{p}_i = \left\{p_{ij}\right\}_{j\neq i, j=1,\ldots,n}$ is the local adaptor. To solve the above optimization problem, we use Lagrangian multiplier:

$$\begin{aligned}L(\mathbf{w}_i;\eta) = {} & \|\mathbf{x}_i - \tilde{\mathbf{D}}\mathbf{w}_i\|_2^2 \\ & + \lambda_1\|\mathbf{p}_i \odot \mathbf{w}_i\|_2^2 + \lambda_2\left(\mathbf{1}^T\mathbf{w}_i - 1\right)\end{aligned} \tag{17}$$

where λ_2 is Lagrange parameter and λ_1 is LLC regularization parameter. Eq. (17) can be modified to:

$$\begin{aligned}L(\mathbf{w}_i;\eta) = {} & \mathbf{w}_i^T\mathbf{C}\mathbf{w}_i + \lambda_1\mathbf{w}_i^T\left\{diag(\mathbf{p}_i)\right\}^2\mathbf{w}_i \\ & + \lambda_2\left(\mathbf{1}^T\mathbf{w}_i - 1\right)\end{aligned} \tag{18}$$

where $\mathbf{C} = \left(\mathbf{x}_i\mathbf{1}^T - \tilde{\mathbf{D}}\right)^T\left(\mathbf{x}_i\mathbf{1}^T - \tilde{\mathbf{D}}\right)$ and $diag(\mathbf{p}_i)$ is a diagonal matrix whose components are elements of $\mathbf{p}_i$ vector. By Derivation from

(18) and making it equal to zero we have:

$$\frac{\partial}{\partial \mathbf{w}_i} L(\mathbf{w}_i;\eta)=0 \Rightarrow \mathbf{S}\mathbf{w}_i+\lambda_2\mathbf{1}=0 \tag{19}$$

where $\mathbf{S}=2\left(\mathbf{C}+\lambda_1\{diag(\mathbf{p}_i)\}^2\right)$. By multiplying both sides of (19) by $\mathbf{1}^T\mathbf{S}^{-1}$ we have:

$$\mathbf{1}^T\mathbf{S}^{-1}(\mathbf{S}\mathbf{w}_i+\lambda_2\mathbf{1})=0$$
$$\Rightarrow \mathbf{1}^T\mathbf{w}_i+\mathbf{1}^T\mathbf{S}^{-1}\lambda_2\mathbf{1}=0 \tag{20}$$

Considering LLC constraint $\mathbf{1}^T\mathbf{w}_i=1$ we have:

$$\lambda_2=-\left(\mathbf{1}^T\mathbf{S}^{-1}\mathbf{1}\right)^{-1} \tag{21}$$

By placing Eq. (21) in Eq. (19) we have:

$$\mathbf{w}_i=\frac{\mathbf{S}^{-1}\mathbf{1}}{\mathbf{1}^T\mathbf{S}^{-1}\mathbf{1}} \tag{22}$$

So $\mathbf{w}_i$ coefficients are calculated by:

$$\begin{cases}\tilde{\mathbf{w}}_i=\left(\mathbf{C}+\{diag(\mathbf{p}_i)\}^2\right)\backslash\mathbf{1}\\ \mathbf{w}_i=\tilde{\mathbf{w}}_i/\mathbf{1}^T\tilde{\mathbf{w}}_i\end{cases} \tag{23}$$

Later, using the above mentioned expressions, two linear and a kernel algorithms for label propagation based on LLC are presented.

3.2. Label propagation through locality constrained linear coding (LPLLC)

We use two kinds of local adapters. The first local adaptor is considered as:

$$\mathbf{p}_i=\left\{p_{ij}\right\}_{j\neq i,j=1,\dots,n}=\left\{\|\mathbf{x}_i-\mathbf{x}_j\|_2\right\}_{j\neq i,j=1,\dots,n} \tag{24}$$

This represents the Euclidean distance between ith and jth samples. We call algorithm based on this local adaptor LPLLC-L2.

The second local adaptor used in this paper has an exponential form of:

$$\mathbf{p}_i=\left\{p_{ij}\right\}_{j\neq i,j=1,\dots,n}=\left\{\exp\left(\frac{\|\mathbf{x}_i-\mathbf{x}_j\|_2}{\sigma}\right)\right\}_{j\neq i,j=1,\dots,n} \tag{25}$$

where σ is a positive parameter. Since the above local adaptor grows exponentially, when ith and jth samples are far from each other, the corresponding $\mathbf{p}_i$will be very high. When we want to emphasis on local information importance, we use this local adaptor. We call algorithm constructed based on this local adaptor, LPLLC-exp.

LPLLC-L2 and LPLLC-exp methods are presented as follows:

Algorithm 1. LPLLC-L2 and LPLLC-exp

Input: $\mathbf{D}=[\mathbf{x}_1,\dots,\mathbf{x}_\ell,\mathbf{x}_{\ell+1},\dots,\mathbf{x}_n]\in\mathscr{R}^{d\times n}$dictionary matrix including all samples and the labels of training samples $\mathbf{Y}_\ell$.

1. Defining local adaptor l_2-norm for LPLLC-L2 and exponential adaptor for LPLLC-exp by (24) and (25).
2. Calculating graph structure and edge weights in similarity graph by (23).
3. Estimating the labels of the unlabeled samples$\mathbf{Y}_u$by (15).

Output: labels of the unlabeled samples$\mathbf{Y}_u$.

Second and third stages which are main stages of LPLLC-L2 and LPLLC-exp algorithms have very simple and analytical solution. So these two algorithms are linear and extremely simple.

3.3. Label propagation through kernel locality constrained Linear Coding (LPKLLC)

In this section, by soling LLC algorithm in kernel field, we introduce a label propagation algorithm based on LLC in kernel field, which we call LPKLLC. Eq. (16) is rewritten in kernel field as:

$$\underset{\mathbf{w}_i}{\operatorname{argmin}}\ \|\varphi(\mathbf{x}_i)-\tilde{\mathbf{D}}^\phi\mathbf{w}_i^\phi\|_2^2+\lambda\|\mathbf{p}_i^\phi\odot\mathbf{w}_i^\phi\|_2^2$$
$$s.t.\ \mathbf{1}^T\mathbf{w}_i^\phi=1 \tag{26}$$

where φ is a nonlinear mapping which maps samples from input space $\mathscr{X}$ into high dimensional feature space$\mathscr{H}$. $\mathbf{w}_i^\phi$ represents weight vector obtained in the kernel field and $\tilde{\mathbf{D}}^\phi=[\varphi(\mathbf{x}_1),\varphi(\mathbf{x}_2),\dots,\varphi(\mathbf{x}_n)]\in\mathscr{R}^{L\times n}$ represents dictionary matrix, once information is mapped into high dimensional feature space $\mathscr{H}$, where $L\gg d$ is $\mathscr{H}$ dimensionality. Also

$\mathbf{p}_i^\phi=\left\{p_{ij}^\phi\right\}_{j\neq i,j=1,\dots,n}=\left\{\|\varphi(\mathbf{x}_i)-\varphi(\mathbf{x}_j)\|_2\right\}_{j\neq i,j=1,\dots,n}$ is a local adaptor in kernel field. $\mathbf{w}_i^\phi$ is calculated in the same manner as for LPLLC, hence:

$$\begin{cases}\tilde{\mathbf{w}}_i^\phi=\mathbf{S}^{-1}\mathbf{1}=\left(\mathbf{C}^\phi+\left\{diag\left(\mathbf{p}_i^\phi\right)\right\}^2\right)^{-1}\mathbf{1}\\ \mathbf{w}_i^\phi=\tilde{\mathbf{w}}_i^\phi/\mathbf{1}^T\tilde{\mathbf{w}}_i^\phi\end{cases} \tag{27}$$

Since nonlinear mapping φ is unknown, (27) is not directly solvable. So we do the following:

$$\begin{aligned}\mathbf{C}^\phi&=\left\{\varphi(\mathbf{x}_i)\mathbf{1}^T-\tilde{\mathbf{D}}^\phi\right\}^T\left\{\varphi(\mathbf{x}_i)\mathbf{1}^T-\tilde{\mathbf{D}}^\phi\right\}\\&=\left\{\mathbf{1}\varphi^T(\mathbf{x}_i)-\tilde{\mathbf{D}}^{\phi^T}\right\}\left\{\varphi(\mathbf{x}_i)\mathbf{1}^T-\tilde{\mathbf{D}}^\phi\right\}\\&=\mathbf{1}K(i,i)\mathbf{1}-K(:,i)\mathbf{1}^T-\mathbf{1}K(i,:)+K(:,:)\end{aligned} \tag{28}$$

in which $K(i,j)$ is defines by ith row and jth column, $K(i,:)$ represents ith row and $K(:,j)$ represents jth column of Gram matrix which is usually known. Moreover we have:

$$\begin{aligned}\left\{diag\left(\mathbf{p}_i^\phi\right)\right\}^2&=diag\left(\left\{\|\varphi(\mathbf{x}_i)-\varphi(\mathbf{x}_j)\|_2^2\right\}_{j\neq i,j=1,\dots,n}\right)\\&=diag\left(\left\{K(i,i)+K(j,j)-2K(i,j)\right\}_{j\neq i,j=1,\dots,n}\right)\end{aligned} \tag{29}$$

Using (27) to (29) we can have $\mathbf{w}_i^\phi$(weigh vector in kernel field).

Algorithm 2. LPKLLC

Input: $\mathbf{D}=[\mathbf{x}_1,\dots,\mathbf{x}_\ell,\mathbf{x}_{\ell+1},\dots,\mathbf{x}_n]\in\mathscr{R}^{d\times n}$ dictionary matrix including all samples and training samples labels$\mathbf{Y}_\ell$.

1. Calculating graph structure and edge weights in similarity graph by (27) to (29).
2. Estimating the labels of the unlabeled samples,$\mathbf{Y}_u$ using (15).

Output: labels of the unlabeled samples$\mathbf{Y}_u$.

4. Experimental results

In this part of the paper, in order to evaluate the performance of the proposed algorithms, experimental results on some data bases are presented. Evaluations are performed on a number of manual data sets, 3 benchmark datasets of UCI [45], ORL face data base [46], Extended Yale B (B+) face data base [47] and PolyU 2D-3D Palmprint data base [48]. Also, to make better comparison for the performance of the proposed algorithms, a number of classification algorithms are employed. Among these classification algorithms there are three graph-based semi-supervised learning algorithms LNP06 [16,17], LNP08 [18] and consistency [15]. Also we have employed four graph-based semi-supervised learning algorithms

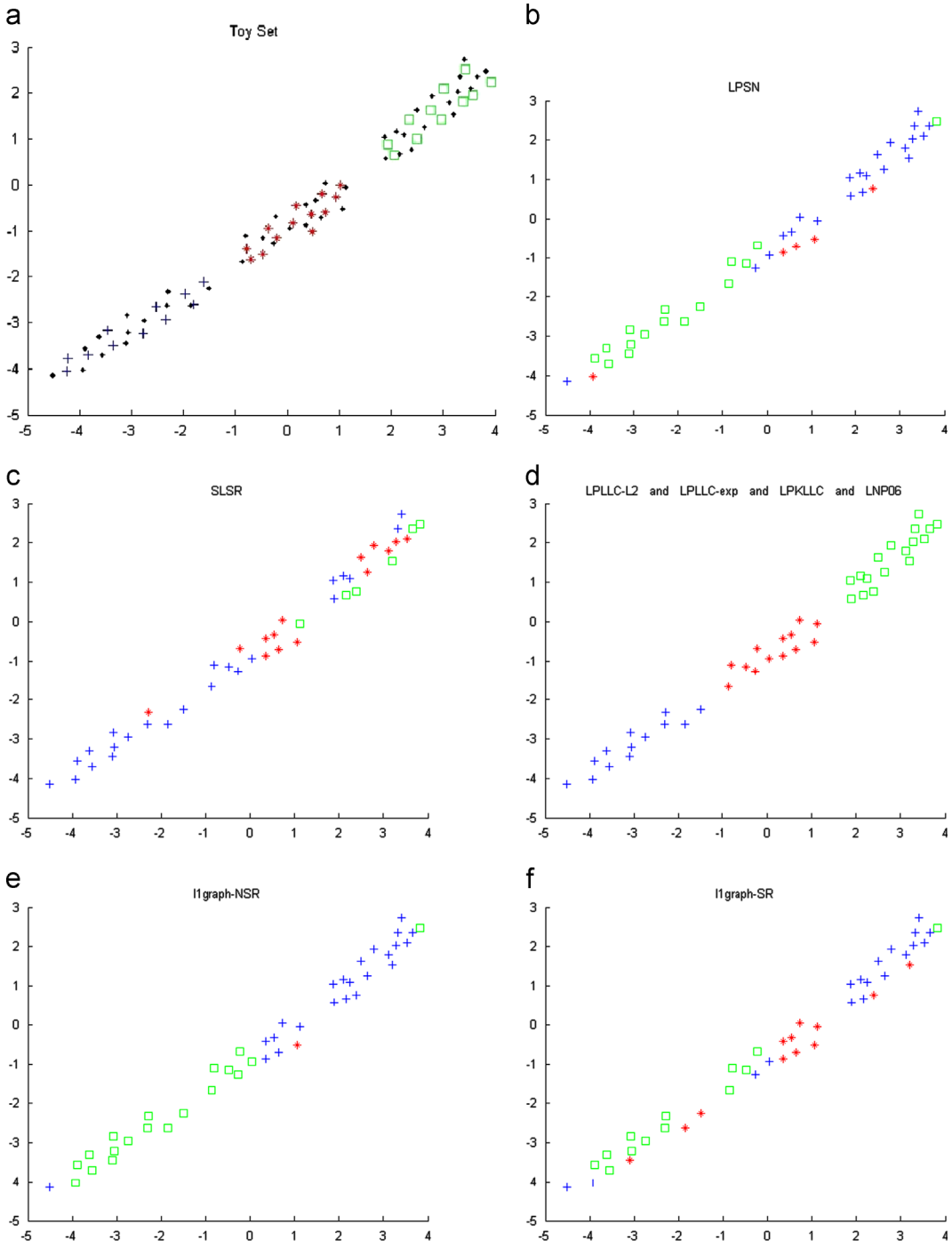

Fig. 1. Toy dataset 1 (data with same direction distribution): in (a) samples of 3 classes are shown. Test sample marked with **"."**. (b–f) show results of test sample classification using algorithms addressed in this article.

based on SR including l_1-graph-SR, l_1-graph-NSR [49] and LPSN [44]. We have also used SRC, KSRC and K-NNC algorithms, where the first two are the basic SR based classification algorithms and the third is one of the most fundamental classification algorithms. It is noticeable that in l_1-graph algorithm presented in [24], in order to make edge weights, the constraint of coefficients being non-negative is hidden inside the cost function. Besides, NSR algorithm is employed to solve them. We have defined this algorithm in the form of l_1-graph-NSR. Also l_1-graph-SR is constructed as follows: after calculating SR coefficients consisting of positive and negative elements, negative coefficients are considered zero and we have used the vector as edge weights. This is different from l_1-graph main idea and we call this algorithm, l_1-graph-SR.

Since algorithms discussed in this paper are used when training samples (labeled samples) are small in numbers, we try to consider the least number of training samples for each class while experimenting. Also, due to labeled samples being few, it is not possible to use methods like cross-validation to find optimal parameters. So an algorithm with lower degree of sensitivity to its own parameters is more suitable. Our approach toward the determination of the optimum parameters is to initially assume that the labels for all test samples are known, then to execute all the algorithms and to find their optimum parameters and finally to exploit these parameters as the optimum parameters for next experiment. This way an algorithm that performs better than others not only has better classification performance, but has less sensitivity to parameters adjustment. These

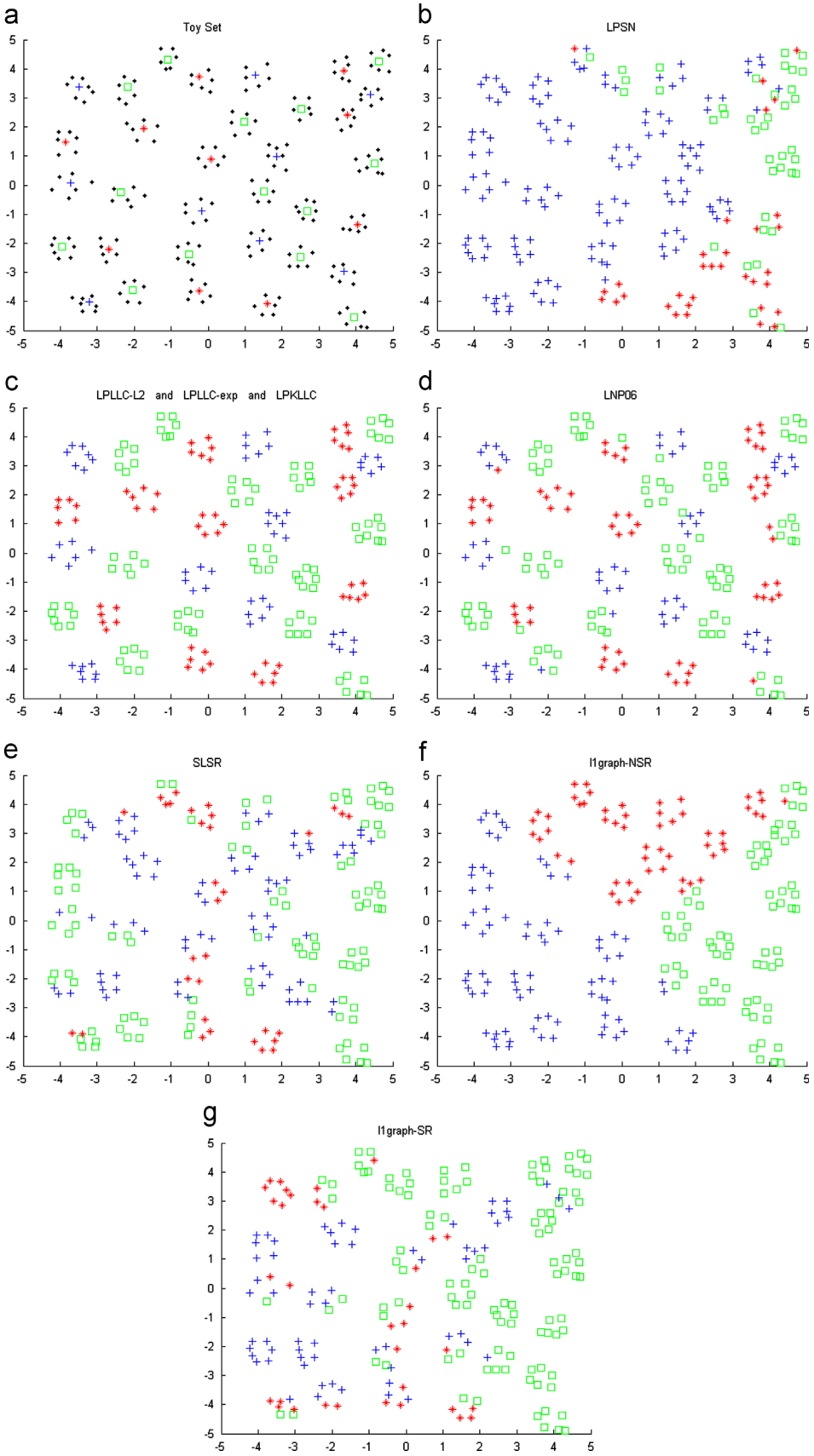

Fig. 2. Toy data set 2: in (a) samples of 3 classes are shown. Test sample marked with **"."** . (b-g) sow results of test sample classification with aim of proposed algorithms in this paper.

points are two main advantages for that algorithm. ε parameter as in LPSN is assumed 10^{-4}. Also for LNP08 and consistency α parameter ($0 \prec \alpha \prec 1$) is considered0.99.

We use Principal Component Analysis (PCA) [50] for feature reduction and MATLAB functions "spgl1" [51] and "l1-ls-nonneg" [52] respectively to calculate SR and NSR coefficients. Also famous RBF kernel is used to map samples to high dimensional feature space.

In all experiments, interval [1–9] with distance 1, interval [0.1–0.9] with distance 0.1, interval [0.01–0.09] with distance 0.01 and interval [0.001–0.009] with distance 0.001are intervals used to determine parameters.

After choosing each interval, to find better parameter, search in smaller distance in its interval is carried. This search method is known as global to local search strategy [53].

Table 1
Information for data sets used from UCI.

Data set	Number of samples	Number of features	Number of classes
Wine	178	13	3
Iris	150	4	3
Sonar	208	60	2

Table 2
Mean and standard deviations of recognition rate (%) for 13 algorithms on Wine, Iris and sonar data sets when 10%, 5% and 20% of samples respectively are chosen for training and the rest are chosen for testing.

Number of training samples Algorithms	Wine	Iris	Sonar
LPLLC-L2	**75.10** ± 4.48	92.87 ± 4.58	83.16 ± 5.92
LPLLC-exp	74.94 ± 3.60	**96.16** ± 1.26	**83.66** ± 4.93
LPKLLC	74.55 ± 4.12	94.04 ± 4.32	68.93 ± 8.49
LPSN	72.69 ± 4.07	90.69 ± 4.71	73.54 ± 8.57
LNP06	68.90 ± 4.22	75.32 ± 9.06	63.50 ± 13.50
LNP08	50.14 ± 6.19	78.70 ± 4.74	58.77 ± 4.17
Consistency	55.41 ± 4.18	83.33 ± 4.37	51.56 ± 2.77
L1graph-SR	73.21 ± 3.70	92.78 ± 5.44	68.48 ± 2.70
L1graph-NSR	74.01 ± 4.80	90.05 ± 7.23	70.62 ± 7.69
SLSR	73.07 ± 3.39	92.78 ± 5.44	68.48 ± 2.70

4.1. Evaluation on some toy datasets

We introduce two toy datasets and in order to assess the performance of our proposed algorithms. Figs. 1a and 2a show the datasets. These datasets include 3 classes of samples, which are all in the interval of $[-5, 5]$. In all the experiments, the best result for each algorithm is displayed as the result of that experiment. Data set 1 show disadvantages of algorithms based on sparse representation. In these algorithms, local information is not considered in the coding process and this causes the algorithms show weak performance in data classification of the first dataset, although the classification of these datasets is very simple. Data set 1 shows a very simple example of data with the same directional distribution. This means samples of different classes are in the same vector direction. In these conditions, each sample is represented in term of almost all samples, which result in less accuracy for classification algorithms based on SR. For this case, the resulting coefficients vector is no longer sparse and the application of SR theory is pointless. Data set 2 is mostly used to show disadvantages of graph based algorithms in which there is a need to find a proper neighbor's number parameter K. In data set 2, samples of different classes are sporadic and in a constant parameter K, no accurate classification is an attainable. So LNP06 algorithm has less accuracy than our proposed algorithms. Also in this data set, each sample is not represented only by a linear combination of samples in its own class, so SR algorithms are not efficient. LPLLC-L2, LPLLC-exp, and LPKLLC, by considering local information in coding process and adaptive determination of the number of neighbors (K), have better classification accuracy compared to SR based algorithms and algorithms like LNP06.

4.2. Evaluation on some benchmark datasets from UCI

In this section of experiments, we evaluate efficiency of proposed algorithms on 3 benchmark data sets of UCI machine learning set. Information of data sets used is brought in Table 1.

For data sets Wine, Iris and sonar, 10%, 5% and 20% of samples are chosen randomly for training and the rest are chosen for testing. Algorithms are repeated 15 times. Table 2 shows result of this experiment.

Table 2 shows for Wine data set LPLLC-L2 has better performance compared to other 12 algorithms. Also, it is seem that for Iris and sonar data sets LPLLC-exp das better classification rate. Moreover,

Fig. 3. Sample images of one person in ORL face data base.

Table 3
Mean and standard deviations of recognition rate (%) for algorithms on ORL face data base versus different size of training samples.

Number of training samples Algorithms	2	3	4	5	6
LPLLC-L2	87.64 ± 3.18	91.86 ± 2.39	95.22 ± 1.73	96.70 ± 1.01	97.29 ± 1.49
LPLLC-exp	88.94 ± 3.17	93.21 ± 2.30	**95.81** ± 1.32	**97.43** ± 1.05	**97.75** ± 1.41
LPKLLC	84.65 ± 3.58	90.29 ± 3.19	93.86 ± 1.88	96.00 ± 1.91	97.04 ± 1.35
LPSN	74.69 ± 4.42	81.19 ± 2.56	85.11 ± 1.87	86.93 ± 1.64	87.63 ± 1.64
LNP06	58.87 ± 4.13	66.88 ± 3.95	83.67 ± 2.59	88.90 ± 2.38	91.88 ± 2.23
L1graph-SR	88.62 ± 2.99	92.83 ± 2.05	95.39 ± 1.50	96.20 ± 1.58	96.96 ± 1.33
L1graph-NSR	85.23 ± 2.79	90.29 ± 2.67	93.06 ± 2.05	95.30 ± 1.71	96.63 ± 1.87
SLSR	**89.40** ± 2.81	**93.64** ± 1.99	95.78 ± 1.79	96.77 ± 1.13	97.00 ± 1.36
SRC	82.44 ± 2.59	89.48 ± 2.78	93.00 ± 1.45	94.87 ± 1.45	95.96 ± 1.27
KSRC	84.21 ± 2.28	90.19 ± 2.78	93.47 ± 1.57	95.13 ± 1.55	96.08 ± 1.22

Table 4
Mean and standard deviations of recognition rate (%) for algorithms on ORL face database versus different noise variances.

Noise variances Algorithms	0.1	0.2	0.25	0.3	0.4	0.5
LPLLC-L2	87.36 ± 1.85	78.64 ± 2.20	74.08 ± 1.83	64.33 ± 2.69	55.64 ± 2.51	43.14 ± 1.41
LPLLC-exp	**89.22** ± 1.61	**80.44** ± 2.16	**74.78** ± 2.59	64.14 ± 2.63	55.58 ± 2.84	43.94 ± 1.38
LPKLLC	87.78 ± 2.21	79.53 ± 1.77	74.50 ± 1.80	**65.36** ± 2.76	**56.92** ± 2.75	**44.08** ± 1.33
LPSN	76.00 ± 2.97	58.39 ± 3.40	49.81 ± 1.95	39.00 ± 3.18	31.53 ± 2.79	21.22 ± 1.89
LNP06	87.56 ± 1.73	77.25 ± 2.74	69.39 ± 2.71	57.36 ± 3.19	42.69 ± 2.88	30.56 ± 1.81
L1graph-SR	84.61 ± 2.15	70.64 ± 3.01	62.06 ± 2.72	48.67 ± 2.60	39.89 ± 2.13	28.08 ± 1.40
L1graph-NSR	86.06 ± 2.27	80.11 ± 1.81	73.56 ± 1.74	63.97 ± 3.20	55.34 ± 2.50	43.17 ± 1.83
SLSR	77.97 ± 2.29	57.56 ± 2.78	46.36 ± 2.42	31.56 ± 1.75	24.69 ± 2.42	14.69 ± 2.08
SRC	82.69 ± 2.34	67.22 ± 2.46	58.53 ± 2.69	44.92 ± 2.29	33.03 ± 2.23	20.42 ± 2.06
KSRC	85.31 ± 1.91	73.89 ± 2.12	66.61 ± 2.74	54.50 ± 2.37	41.11 ± 2.19	30.33 ± 1.99

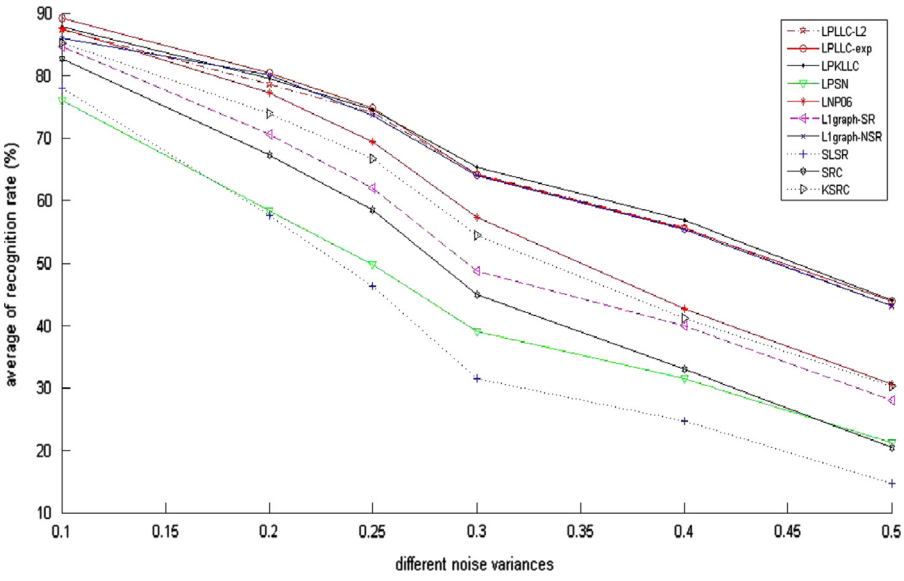

Fig. 4. Average of recognition rate (%) for algorithms on ORL face database versus different noise variances.

among SR based algorithms, for Wine, Iris and sonar data sets l_1-graph-NSR, SRC and LPSN respectively have better classification accuracy.

Considering the way required parameters for each algorithm are determined, it is concluded that algorithms with highest efficiency rate among other methods, have less sensibility to adjusting their own parameters, which is considered an advantage.

4.3. Evaluation on face data bases

To evaluate proposed algorithms, in this part of experiment, ORL and Extended Yale B (B+) face data bases are used.

4.3.1. Evaluate on ORL face data base.

ORL face database consist of 400 face images of 40 people. The size of each image is 92 × 112 pixels, with 256 gray levels per pixel. All images are taken against a dark homogeneous background but vary in sampling time, light conditions, facial expression (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). Fig. 3 shows sample images of one person.

L images of each person (L ∈ [2–6]) are chosen randomly for training and the rest are chosen for testing. Due to high dimension of samples, PCA is used for dimension reduction and number of feature for each sample is reduced to 110 features. Table 3 shows average recognition rates and its standard deviations across 15 runs.

Fig. 5. Sample images of one person in Extended Yale B (B+) face data base.

Table 5
Mean and standard deviations of recognition rate (%) for algorithms versus dimension, on Extended Yale B (B+) face data base for executing program 15 times when 3 images of each person are chosen randomly for training and the rest are chosen for testing.

Number of features Algorithms	10	60	110	160
LPLLC-L2	44.84 ± 2.65	69.13 ± 2.57	71.87 ± 3.57	72.41 ± 3.63
LPLLC-exp	**45.36** ± 2.85	**69.56** ± 2.92	**72.22** ± 3.10	**72.54** ± 3.46
LPKLLC	42.62 ± 2.78	64.12 ± 4.16	69.48 ± 2.93	69.71 ± 3.08
LPSN	36.71 ± 2.15	62.88 ± 3.03	67.28 ± 3.44	68.29 ± 3.19
LNP06	12.10 ± 2.08	42.98 ± 3.37	57.10 ± 2.72	58.21 ± 3.35
L1graph-SR	38.65 ± 1.02	64.44 ± 3.13	68.21 ± 2.79	69.90 ± 3.58
L1graph-NSR	41.55 ± 2.74	68.26 ± 2.38	71.21 ± 3.02	71.63 ± 2.89
SLSR	34.80 ± 2.75	63.65 ± 2.44	67.06 ± 2.78	69.13 ± 3.57
SRC	31.47 ± 2.33	46.23 ± 2.64	49.64 ± 2.61	50.07 ± 3.39
KSRC	36.55 ± 2.35	48.89 ± 2.21	51.75 ± 3.01	52.06 ± 3.52

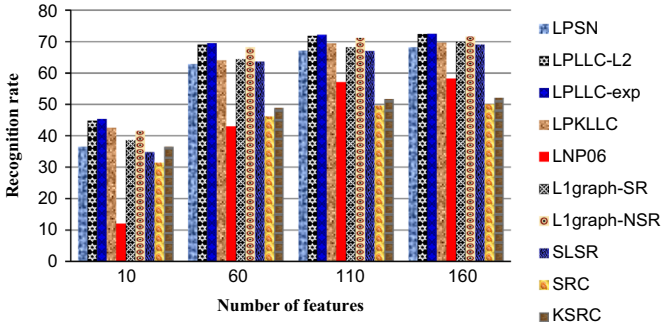

Fig. 6. Average recognition rate for each method versus dimension, on Extended Yale B (B+) face data base when executing program 15 times, and when 3 images of each person are chosen randomly for training and the rest are chosen for testing.

Results of this experiment show that while number of training sample for each class increases, classification efficiency for all algorithms improve. Also, in fixed and low number of training samples, SLSR has better efficiency than other algorithms. And as number of training samples increases, LPLLC-exp outperforms other algorithms. In all experiments, LNP06 and LPSN have the least classification accuracy. Since LPKLLC algorithm is kernel based, it is expected that its classification rate is higher than other algorithms, but as optimum kernel parameter needs to be defined properly in this algorithm and we consider a constant parameter for it, efficiency is lower. On the whole, based algorithm this experiment, it is concluded that these algorithms have less sensibility to adjusting their own parameters, which is an advantage for them.

In order to verify the performance of our proposed methods in the presence of noise a set of experiments is done on ORL face database. 4 images of each person are chosen randomly for training and the rest are chosen for testing. A zero mean Gaussian noise with different variances (0.1, 0.2, 0.25, 0.3, 0.4, and 0.5) is added to the data. Experiments on 100-dimensional PCA subspaces are repeated 15 times. Average and standard deviation of recognition rates obtained from this experiment are listed in Table 4 and Fig. 4.

According to this table for given variance our algorithms have better performance quality with regard to other methods. This table also shows that by increasing the noise variance, the performance of our methods involve relatively fewer reduction compared to other algorithms, therefore the suggested methods have less sensitivity to the sample noise.

4.3.2. *Evaluate on extended yale B (B+) face data base*

The Extended Yale B (B+) Face database contains 16,128 images of 28 human subjects under 9 poses and 64 illumination conditions. All the face images are manually aligned, cropped, and then re-sized to 80 × 80 images. Fig. 5 shows sample images of one person.

We randomly choose 3 images of each person for training and the rest for testing. Experiments on m-dimensional PCA subspaces are repeated 15 times where m is 10, 60, 110 and 160. Average and standard deviation of recognition rates obtained from this experiment are listed in Table 5. Fig. 6 illustrates the average recognition rates of each method versus dimension.

Below results are taken from Table 5:

(a) Not considering sample dimension and under sane condition, LLLC-exp and LPLLC-L2 have the best performance. (b) When feature space has low dimension, LNP06 is worse than other methods. (c) l_1-graph-NSR is the most efficiency algorithm a among SR based algorithms. (d) LPLLC-exp and LPLLC-L2 algorithms have less sensibility to adjusting their own parameters.

4.4. *Evaluation PolyU 2D-3D palmprint data base*

The PolyU 3D Palmprint Database contains 8000 samples collected from 400 different palms. Twenty samples from each of these palms were collected in two separated sessions, where 10 samples were

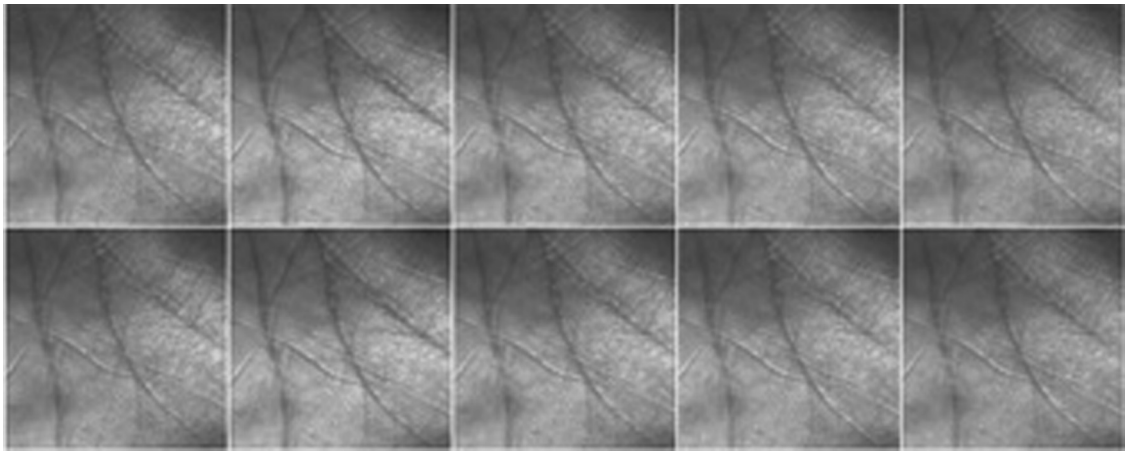

Fig. 7. Sample images of one person's palmprints in PolyU 2D-3D Palmprint data base.

Table 6
Mean and standard deviations of recognition rate (%) for 8 algorithms on two groups of PolyU 2D-3D Palmprint data base after executing programs for 15 times.

Data Number of training samples for each class	Group of images 1 %10 of sample of each class	Group of images 2 One sample of each class	Group of images 3 Two samples of each class
LPLLC-L2	93.63 ± 4.04	**94.84** ± 0.62	**96.40** ± 0.83
LPLLC-exp	**93.85** ± 2.70	93.38 ± 1.19	95.73 ± 1.15
LPKLLC	87.81 ± 4.14	81.11 ± 2.29	88.93 ± 1.56
LPSN	82.15 ± 3.38	85.02 ± 0.79	88.07 ± 1.93
LNP06	67.09 ± 5.07	17.02 ± 2.25	95.77 ± 1.18
L1graph-SR	92.83 ± 3.37	89.00 ± 1.05	93.00 ± 1.70
L1graph-NSR	92.04 ± 3.31	86.98 ± 0.87	91.37 ± 1.32
SLSR	91.93 ± 3.70	87.16 ± 0.04	91.47 ± 1.68

captured in each session, respectively. The average time interval between the two sessions is one month. Each sample contains a 3D ROI (region of interest) and its corresponding 2D ROI. However, almost all the current palmprint recognition techniques use the two dimensional (2D) image of the palm surface for feature extraction and matching. Hence, in this work, samples of 2D palmprint images for experimentation. All the face images are manually aligned, cropped, and then re-sized to 128 × 128 images. Few sample palmprint images are shown in Fig. 7.

Of this data base 2 group of images are selected. The first group of images include images of first 20 people of data base where all 20 images (10 from first session and 10 from second session) are chosen and totally 400 image are considered. Second group of images include 100 last people in data base where 4 image of each person is chosen so that 2 images are belonging to first session and 2 images are belonging to second session.

In an experiment on images belonging to first group, 10% of sample of each class chosen randomly are considered for training and the rest of samples are for testing.

In experiment on second group, we randomly choose L images of each person (L ∈ [2,3]) for training and the rest of images for testing. Experiment on 100-dimensional PCA subspaces, are repeated 15 times. Table 6 lists average and standard deviation of recognition rate for each classification method.

Table 6 shows that in experiments on first group of images, LPLLC-exp and LPSN algorithms are best and worst algorithms. Also it is seen that in experiment on second group of images LPLLC-L2 is most efficiency algorithm. On the other hand, LPLLC-exp and LPLLC-L2 have best classification rate and least sensibility to adjusting their own parameters.

Of this data base 2 group of images are selected. The first group of images include images of the first 20 people of data base where all 20 images (10 from first session and 10 from the second session) are chosen and totally 400 image are considered. The second group of images include 100 last people in data base where 4 image of each person is chosen so that 2 images are belonging to first session and 2 images are belonging to second session.

Of this data base 2 group of images are selected. The first group of images include images of first 20 people of data base where all 20 images (10 from the first session and 10 from the second session) are chosen and totally 400 image are considered. Second group of images include 100 last people in data base where 4 image of each person is chosen so that 2 images are belonging to first session and 2 images are belonging to second session.

Of this data base 2 group of images are selected. The first group of images include images of first 20 people of data base where all 20 images (10 from the first session and 10 from the second session) are chosen and totally 400 image are considered. Second group of images include 100 last people in data base where 4 image of each person is chosen so that 2 images are belonging to first session and 2 images are belonging to second session.

In an experiment on images belonging to first group, %10 of sample of each class chosen randomly are considered for training and the rest of samples are for testing.

In experiment on second group, we randomly choose L images of each person (L ∈ [2,3]) for training and the rest of images for testing. Experiment on 100-dimensional PCA subspaces, are repeated 15 times. Table 6 lists average and standard deviation of recognition rate for each classification method.

Table 6 shows that in experiments on first group of images, LPLLC-exp and LPSN algorithms are best and worst algorithms. Also it is seen that in experiment on second group of images LPLLC-L2 is most efficiency algorithm. On the other hand, LPLLC-exp and LPLLC-L2 have best classification rate and least sensibility to adjusting their own parameters.

5. Conclusions

In graph-based algorithms based on sparse representation, graph structure and edge weights are found simultaneously using l_1-norm optimization. Also since Euclidean distance is not used in these algorithms, they have less sensibility to noise. On the other hand, in these algorithms number and degree of similarity of samples to each other using SR theory is determined adaptively and there is no parameter to determine number and degree of similarity between neighbors, so efficiency of these algorithms are much higher than graph-based algorithms. Although sparse representation has advantages, does not consider local information in coding process and its solution requires a time consuming and heavy optimization process. In this paper 3 new label propagation algorithms LPLLC-L2, LPLLC-exp and LPKLLC are presented where the first 2 are linear algorithm and the third is a nonlinear algorithm in kernel space. In our algorithms unlike sparse representation based algorithms local information of samples are used in coding process. Also in this algorithms like SR based algorithms number and degree of similarity for neighbors of each sample are determined adaptively hence classification accuracy is higher.

To make better compression for proposed algorithms efficiency, a large number of related algorithms are investigated. Result of experiments on 3 toy data sets, 3 benchmark data sets from UCI, two face data bases and a biometric data base show that our algorithms have higher classification rate compared to other algorithms. Moreover due to the method used in choosing parameters in each algorithm, it is fair to say that our algorithm have less sensibility to these parameters and this is another advantage for our algorithms. In future we will focus on our algorithms in applications such as target detection and visual tracking.

References

[1] X. Zhu, Z. Ghahramani, J. Lafferty, Semi-supervised learning using gaussian fields and harmonic functions, Proc. 20th Int. Conf. Mach. Learn. (2003) 912–919.
[2] A. Blum, T.M. Mitchell, Combining labeled and unlabeled data with Co-Training, Proc. 11th Ann. Conf. Learn. Theory (1998) 92–100.
[3] A. Fujino, N. Ueda, K. Saito, A hybrid generative/discriminative approach to semi-supervised classifier design, Proc. 20th Natl. Conf. Artif. Intell. (2005) 764–769.
[4] C. Rosenberg, M. Hebert, H. Schneiderman, Semi-Supervised self-training of object detection models, in: Proceedings of the Seventh IEEE Workshops Application of Computer Vision, pp. 29–36, 2005.
[5] G. Fung, O. Mangasarian, Semi-supervised support vector machines for unlabeled data classification, Optim. Methods Software 15 (2001) 29–44.
[6] C. Rosenberg, M. Hebert, H. Schneiderman, Semi-supervised self-training of object detection models, in: Proceedings of the 7th Workshop on Applications of Computer Vision, 1, pp. 29–36, 2005.
[7] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.
[8] Y. Song, F. Nie, C. Zhang, S. Xiang, A unified framework for semi-supervised dimensionality reduction, Pattern Recognit. 41 (9) (2008) 2789–2799.
[9] Chapelle O., Weston J., Scholkopf B., Cluster kernels for semi-supervised learning, in: Proceedings of the Neural Information Processing Systems Conference, 15, pp. 585–592, 2003.
[10] N.D. Lawrence, M.I. Jordan, Semi-supervised learning via gaussian processes, Proc. Neural Inf. Process. Syst. Conf. (2005) 753–760.
[11] T. Joachims, Transductive inference for text classification using support vector machines, in: Proceedings of the 16th International Conference Machine Learning, 1999.
[12] T. Joachims, Transductive learning via spectral graph partitioning, Proc. 20th Intl. Conf. Mach. Learn. (2003) 290–297.
[13] M. Belkin, P. Niyogi, V. Sindhwani, On manifold regularization, Proc. 10th Int. Worksh. Artif. Intell. Stat. (2005) 17–24.
[14] A. Blum, S. Chawla, Learning from labeled and unlabeled data using graph mincuts, ICML (2001) 19–26.
[15] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, Adv. Neural Inf. Process. Syst. 16 (2003).
[16] F. Wang, J.D. Wang, C.S. Zhang, H.C. Shen, Supervised classification using linear neighborhood propagation, IEEE Conference on CVPR'06, 1, pp. 160–167, 2006.
[17] J.D. Wang, F. Wang, C.S. Zhang, H.C. Shen, L. Quan, Linear neighborhood propagation and its applications, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 1600–1615.
[18] F. Wang, C. Zhang, Label propagation through linear neighborhoods, IEEE Trans. Knowl. Data Eng. 20 (1) (2008) 55–67.
[19] Y. Bengio, O.B. Alleau, N. Le Roux, Label propagation and quadratic criterion, in: O. Chapelle, B. Scholkopf, A. Zien (Eds.), Semi-Supervised Learning, MIT Press, 2006, pp. 193–216.
[20] Z.E. Tian, R.U.I. Kuang, Global linear neighborhood for efficient label propagation, Proc. SIAM Int. Conf. Data Min. (SDM) (2012) 863–872.
[21] M. Elad, M. Figueiredo, Y. Ma, On the role of sparse and redundant representations in image processing, Proc. IEEE 98 (2010) 972–982.
[22] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T.S. Huang, S. Yan, Sparse recognition for computer vision and pattern recognition, Proc. IEEE 98 (6) (2010) 1031–1044.
[23] H. Cheng, Z. Liu, L. Yang, X. Chen, Sparse representation and learning in visual recognition: Theory and applications, Signal Process. 93 (6) (2013) 1408–1425.
[24] B.I.N. Cheng, Jianchao Yang, Shuicheng Yan, Y.U.N. Fu, Thomas Huang, Learning with L1-graph for image analysis, IEEE Trans. Image Process. (TIP) 19 (4) (2010) 858–866.
[25] S. Sun, Z. Hussain, J. Shawe-Taylor, Manifold-preserving graph reduction for sparse semi-supervised learning, Neurocomputing 124 (2013) 13–21.
[26] S. Sun, J. Shawe-Taylor, Sparse semi-supervised learning using conjugate functions, J. Mach. Learn. Res. 11 (2010) 2423–2455.
[27] Mingyu Fan, Nannan Gu, Hong Qiao, et al., Sparse regularization for semi-supervised classification, Pattern Recognit. 44 (8) (2011) 1777–1784.
[28] Nannan Gu, D.I. Wang, Mingyu Fan, Deyu Meng, A kernel-based sparsity preserving method for semi-supervised classification, Neurocomputing 139 (2014) 345–356.
[29] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.
[30] J. Yin, Z. Liu, Z. Jin, W. Yang, Kernel sparse representation based classification, Neurocomputing 77 (2012) 120–128.
[31] J. Xu, J. Yang, A nonnegative sparse representation based fuzzy similar neighbor classifier, Neurocomputing 99 (2013) 76–86.
[32] T. Bai, Y.F. Li, Robust visual tracking with structured sparse representation appearance model, Pattern Recognit. 45 (6) (2012) 2390–2404.
[33] J.Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3360–3367, 2010.
[34] S.T. Roweis, L.K. Saul, Nonlinear dimensionality analysis by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[35] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality Reduction, Science 290 (5500) (2000) 2319–2323.
[36] A. Elgammal, R. Duraiswami, L. Davis, Efficient kernel density estimation using the fast gauss transform with applications to color modeling and tracking, IEEE Trans. Pattern Anal. Mach. Intell. 25 (11) (2003) 1499–1504.
[37] M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander, LOF: identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD international conference on Management of data, pp. 93–104, 2000.
[38] K. Yu, T. Zhang, Y. Gong, Nonlinear learning using local coordinate coding, Adv. Neural Inf. Process. Syst. 22 (2009) 2223–2231.
[39] Y.-W. Chao, Y.-R. Yeh, Y.-W. Chen, Y.-J. Lee, Y.-C.F. Wang, Locality-constrained group sparse representation for robust face recognition, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), pp. 761–764, 2011.
[40] E. Amaldi, V. Kann, On the approximation of minimizing non zero variables or unsatisfied relations in linear systems, Theor. Comput. Sci. 209 (1998) 237–260.
[41] S. Chen, D. Donoho, M. Saunders, Atomic decomposition by basis pursuit, SIAM Review 43 (1) (2001) 129–159.
[42] D. Needell, J.A. Tropp, R. Vershynin, Greedy signal recovery review, in: Proceedings of the 42nd Asilomar Conference on Signals, Systems and Computers, pp. 1048–1050, 2008.
[43] J.A. Tropp, Greed is good: algorithmic results for sparse approximation, IEEE Trans. Inf. Theory 50 (10) (2004) 2231–2242.
[44] F. Zang, J.S. Zhang, Label propagation through sparse neighborhood and its applications, Neurocomputing 97 (2012) 267–277.
[45] C. Blake, C. Merz, UCI Repository of Machine Learning Databases, 1998.
[46] F.S. Samaria, A.C. Harter, Parameterisation of a stochastic model for human face identification, in: Proceedings of the Second IEEE Workshop on Applications of Computer Vision, pp. 138–142, 1994.
[47] A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: Illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2001) 643–660.
[48] D. Zhang, G. Lu, W. Li, L. Zhang, N. Luo, Palmprint recognition using 3-D information, IEEE Trans. Syst., Man, Cybern., Part C: Appl. Rev. 39 (5) (2009) 505–519.
[49] S.C. Yan, H. Wang, Semi-supervised learning by sparse representation, Proc. SIAM Int. Conf. Data Min. (SDM) (2009) 792–801.
[50] M.A. Turk, A.P. Pentland, Face Recognition Using Eigenfaces, IEEE Conf. Comput. Vis. Pattern Recognit. (1991) 586–591.
[51] E.V. Berg, M.P. Friedlander, Probing the Pareto frontier for basis pursuit solutions, SIAM J. Sci. Comput. 31 (2) (2008) 890–912.
[52] S.J. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, A method for large-scale ℓ1-regularized least squares problems with applications in signal processing and statistics, IEEE J. Sel. Top. Signal Process 1 (4) (2007) 606–617.
[53] K.R.M. uller, S. Mike, G.R. atsch, K. Tsuda, B.Sch olkopf, An introduction to kernel-based learning algorithms, IEEE Trans. Neural Netw. 12 (2) (2001) 181–201.

Seyyed Alireza Saffari received a B.Sc. degree in Electrical Engineering from Semnan University, Semnan, Iran, and an M.Sc. in Electrical Engineering from Ferdowsi University of Mashhad, Mashhad, Iran, respectively in 2011, and 2014. He is currently a Ph.D. candidate in Electrical Engineering at Ferdowsi University of Mashhad, Mashhad, Iran.

Abbas Ebrahimi-Moghadam received a B.Sc. degree in Electrical Engineering from Sharif University of Technology, Tehran, Iran, and an M.Sc. degree in Electrical Engineering from K. N. Toosi University of Technology, Tehran, Iran in 1991 and 1995, respectively. He also received a Ph.D. degree in Electrical and Computer Engineering form McMaster University, Hamilton, ON, Canada. Dr. Ebrahimi-Moghadam is currently with Electrical Engineering Department at Ferdowsi University of Mashhad, Mashhad, Iran, as an assistant professor since 2011.