Pattern Analysis and Applications (2019) 22:1481–1492
https://doi.org/10.1007/s10044-018-0745-9

THEORETICAL ADVANCES

L1-norm orthogonal neighbourhood preserving projection and its applications

Purvi A. Koringa[1] · Suman K. Mitra[1]

Received: 11 July 2017 / Accepted: 27 July 2018 / Published online: 4 August 2018

Abstract

Dimensionality reduction techniques based on manifold learning are becoming very popular for computer vision tasks like image recognition and image classification. Generally, most of these techniques involve optimizing a cost function in L2-norm and thus they are susceptible to outliers. However, recently, due to capability of handling outliers, L1-norm optimization is drawing the attention of researchers. The work documented here is the first attempt towards the same goal where orthogonal neighbourhood preserving projection (ONPP) technique is performed using optimization in terms of L1-norm to handle data having outliers. In particular, the relationship between ONPP and PCA is established theoretically in the light of L2-norm and then ONPP is optimized using an already proposed mechanism of PCA-L1. Extensive experiments are performed on synthetic as well as real data for applications like classification and recognition. It has been observed that when larger number of training data is available L1-ONPP outperforms its counterpart L2-ONPP.

Keywords L1-norm · L2-norm · Outliers · Dimensionality reduction

1 INTRODUCTION

Being very high-dimensional data, images produce many challenges while handling them in tasks like machine learning, computer vision. Though image appears to be high-dimensional data, it is proved that it lies in comparatively very low-dimensional linear or nonlinear manifold [7, 10]. Thus, dimensionality reduction techniques are very much applicable in these fields and much research is being done. The fundamental philosophy is to seek a nonlinear or linear transformation to map the data from high-dimensional data space to a lower-dimensional subspace which makes the same class of data more compact for applications like recognition and classification, in addition to that also reduces computational burden. Such manifold learning-based dimensionality reduction techniques have drawn considerable interests in recent years. Some of the examples are principal component analysis (PCA) [21], linear discriminant analysis (LDA) [17], locality-preserving projection (LPP) [6, 20] and neighbourhood preserving embedding (NPE) [7, 11], and some of their 2D variants are discussed in [14, 23, 24]. Techniques such as PCA and LDA preserve global geometry of data in the lower-dimensional space also, whereas techniques such as LPP and NPE tend to preserve global geometry as well as local geometry by a graph structure using neighbourhood information.

The linear dimensionality reduction technique such as orthogonal neighbourhood preserving projection (ONPP) proposed in [10] preserves global geometry of data and captures local relationship of neighbourhood also. A modified version of the same that deals with nonlinearity present in the local neighbourhood is given in [11]. ONPP is a linear extension of Locally Linear Embedding (LLE) proposed in [19]. LLE assumes that the data lie on or near a low-dimensional manifold and can be approximated as a linear combination its neighbours. LLE represents this relation using a weighted neighbourhood graph and tries to find embeddings that preserve this linear relationship in lower-dimensional space also. Because of the nonlinear nature of LLE technique, it cannot be used as a tool for finding embeddings of out-of-sample data. ONPP projects the sample data onto a linear subspace using

✉ Purvi A. Koringa
purvi.koringa@gmail.com

Suman K. Mitra
mitrasumank@gmail.com

[1] Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, India

the same philosophy of locally linear patches and also accommodates out-of-sample data into lower-dimensional space.

All these dimensionality reduction techniques try to optimize error functions based on some criteria imposed either on original data points in higher-dimensional space and/or on its embeddings in lower-dimensional space. Most of these error functions are formulated using L2-norm, which are not robust to outliers [16]. L1-norm, on the other hand, is known for its robustness to outliers [8]. In recent times, many dimensionality reduction techniques involve L1-norm optimization [12, 22, 25]. This article proposes one such algorithm used in PCA-L1[12] to achieve L1-norm-based ONPP (now will be denoted as L1-ONPP). This work firstly documents the experiments performed on synthetic data using L2-ONPP, showing susceptibility of it towards outliers. Secondly, the relationship between ONPP and PCA is established and proved theoretically, and the experiments performed on synthetic as well as real data support the claim that ONPP basis can be obtained using PCA. PCA-L1 is used to obtain L1-ONPP, and performance of L2-ONPP and L1-ONPP in the presence of outliers is compared. Experimental outcomes imply that L1-ONPP outperforms L2-ONPP while dealing with the data having outliers.

In Sect. 2, L1-norm-based PCA is explained in detail, followed by Sect. 3 in which a relation between ONPP and PCA is established theoretically, with supporting experiments and results documented in Sect. 4, followed by conclusion in Sect. 5.

2 L1-NORM FOR DIMENSIONALITY REDUCTION

As discussed in Sect. 1, all conventional dimensionality reduction techniques employ optimization of a cost function expressed using L2-norm. Conventional ONPP proposed in [10] is also based on L2-norm optimization. Despite the fact that it has been employed successfully in many problems like face recognition, etc., it is prone to the presence of outliers because the effect of the outliers with a large norm is magnified by the use of the L2-norm. In order to mitigate this problem and achieve robustness against outliers, research has been performed on dimensionality reduction techniques based on L1-norm. Many works have been done in PCA and LDA based on the use of L1-norm [1, 3, 9, 12, 22, 25]. Not much efforts have been put into the use of L1-norm-based methods in recently proposed dimensionality reduction techniques such as LPP and ONPP.

In [1, 9], instead of assuming that each component of error between the original data point and its projection follows Gaussian distribution, it is assumed to follow a Laplacian distribution and maximum likelihood estimation was used to formulate L1-norm PCA (L1- PCA) basis for the given data. In [1], a heuristic estimation approach for general L1-norm problem was applied to solve L1-PCA optimization, whereas in [9], convex programming methods and the weighted median method were proposed for L1-norm PCA. Despite being robust, L1-PCA has several disadvantages, it is computationally expensive because it is based on linear or quadratic programming. [15] discussed 2D variants of L1-norm PCA. [3] proposed R1-PCA, which bands together the merits of L2-PCA and those of L1-PCA. R1-PCA is rotational invariant like L2-PCA, and it also overcomes the effect of outliers as L1-PCA does. However, these methods are highly dependent on the dimension d of a subspace to be found. For example, the projection vector obtained when $d = 1$ may not be in a subspace obtained when $d = 2$. Moreover, as it is an iterative algorithm so for a large-dimensional input space, it takes a lot of time to achieve convergence. Let us now discuss the work done on L1-norm-based PCA.

2.1 L2-PCA and L1-PCA

Let each data point $\mathbf{x_i}$ be a column of $\mathbf{X}$ such that $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_N}] \in \mathcal{R}^{m \times N}$ be the given data matrix, where m denotes dimensions of the original input space and N denotes number of data samples. Without the loss of generality, data are assumed to be centred at origin i.e. $\bar{\mathbf{x}} = \mathbf{0}$. L2-PCA tries to search a $d(< m)$-dimensional linear subspace such that the basis vectors capture the direction of maximum variances by minimizing the error function in terms of L2-norm:

$$\arg\max \mathcal{E}(\mathbf{y}) = \arg\max_{\mathbf{y}} \sum_{i=1}^{N} \| \mathbf{y_i} - \bar{\mathbf{y}} \|_2 \qquad \text{where, } \mathbf{y_i} = \mathbf{V}^{\mathrm{T}} \mathbf{x_i} \tag{1}$$

$$\begin{aligned} \arg\max \mathcal{E}(\mathbf{V}) &= \arg\max_{\mathbf{V}} \sum_{i=1}^{N} \| \mathbf{V}^{\mathrm{T}} \mathbf{x_i} - \mathbf{V}^{\mathrm{T}} \bar{\mathbf{x}} \|_2 \\ &= \arg\max_{\mathbf{V}} \sum_{i=1}^{N} \| \mathbf{V}^{\mathrm{T}} \mathbf{x_i} \|_2 \\ \arg\max \mathcal{E}(\mathbf{V}) &= \arg\max_{\mathbf{V}} \| \mathbf{V^T X} \|_2 \\ s.t.\ \mathbf{V}^{\mathrm{T}} \mathbf{V} &= \mathbf{I_d} \end{aligned} \tag{2}$$

where $\mathbf{V} \in \mathcal{R}^{\mathbf{m \times d}}$ is the projection matrix and its d columns are the bases of the d-dimensional linear subspace.

In PCA-L1 proposed in [12], instead of finding bases in the original data space that capture the direction of maximum variances which is based on the L2-norm, a method that maximizes the dispersion in terms of L1-norm in the feature space is presented to achieve robust and rotation invariant PCA. The approach presented in [12] for L1-norm optimization is iterative and also proven to find a locally maximal solution.

Maximizing dispersion in the feature space using L1-norm can be formulated as

$$\arg\max \mathcal{E}(\mathbf{V}) = \arg\max_{\mathbf{V}} \| \mathbf{V}^{\mathrm{T}}\mathbf{X} \|_1 \tag{3}$$

Since the closed form solution of problems involving L1-norm is not possible, the basis is sought iteratively as follows:

For $d = 1$

$$v_1 = \arg\max_{\mathbf{v}} \| \mathbf{v}^{\mathrm{T}}\mathbf{X} \|_1 = \arg\max_{\mathbf{v}} \sum_{i=1}^{N} |\mathbf{v}^{\mathrm{T}}\mathbf{x_i}| \quad s.\,t.\ \| \mathbf{v} \|_{\mathbf{2}} = 1 \tag{4}$$

For $d > 1$:

Once the basis in the direction of ith maximum variance $\mathbf{v_j}$ ($\mathbf{v_1}$ for first basis) is sought by solving Eq. 4, the data are projected on this newly found basis vector. For the rest of the basis vectors $\mathbf{v_j}$ ($2 \leq j \leq d$) the same maximization problem given in 4 is solved for projected data ($\mathbf{X_j} = \mathbf{X_{j-1}} - \mathbf{v_{j-1}}(\mathbf{v}^{\mathrm{T}}_{\mathbf{j-1}}\mathbf{X_{j-1}})$) iteratively, which essentially means in every iteration, direction of maximum variance in feature space is sought, until desirable $d(d < m)$dimensional space is achieved.

Algorithm to compute bases of PCA-L1[12]:

```
For $d = 1$:

 1. Initialization:
    Pick any $\mathbf{v(0)}$
    Set $\mathbf{v(0)} \leftarrow \mathbf{v(0)} / \| \mathbf{v(0)} \|_{\mathbf{2}}$
    Set $t = 0$.
 2. Polarity Check:
    $\forall i \in 1, ..., N$,
    if $\mathbf{v^T(t)x_i} < 0$, $p_i(t) = -1$,
    otherwise $p_i(t) = 1$
 3. Flipping and maximization:
    Set $t \leftarrow t + 1$
    Set $\mathbf{v(t)} = \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{p_i(t)x_i}$
    Set $\mathbf{v(t)} \leftarrow \mathbf{v(t)} / \| \mathbf{v(t)} \|_{\mathbf{2}}$
 4. Convergence Check:
    a. if $\mathbf{v(t)} \neq \mathbf{v(t-1)}$, go to Step 2.
    b. Else if there exists $i$ such that $\mathbf{v^T(t)x_i} = \mathbf{0}$,
    set $\mathbf{v(t)} \leftarrow (\mathbf{v(t)} + \mathit{\Delta}\mathbf{v}) / \| \mathbf{v(t)} + \mathit{\Delta}\mathbf{v} \|_2$ and go to step 2. (Here, $\mathit{\Delta}\mathbf{v}$ is a small
    nonzero random vector.)
    c. Otherwise, set $\mathbf{v}^* = \mathbf{v(t)}$ and stop.

For $d$>1:
    For $j = 2$ to $d$,

 1. Projecting Data:
    $\mathbf{X_j} = \mathbf{X_{j-1}} - \mathbf{v_{j-1}}(\mathbf{v^T_{j-1}X_{j-1}})$
 2. Finding PCA-L1 basis:
    in order to find $\mathbf{v_j}$, apply PCA-L1 procedure to $\mathbf{X_j}$

end
```

3 L1-ONPP USING PCA-L1

As stated in Sect. 1, ONPP [10] is a linear extension of LLE and thus inherits the sensitivity of LLE towards outliers. The degradation in manifold learning when the data have outliers, inspired the use of L1-norm minimization in ONPP to tackle the outliers. In order to use PCA-L1 explained in Sect. 2 to achieve L1-ONPP, a relationship between PCA and ONPP in established in this Section. Firstly, ONPP and a modified variant of ONPP, namely MONPP, is explained in detail in Sect. 3.1, followed by an theoretical explanation of a relation between PCA and ONPP. Section 3.3 explains how PCA-L1 can be used to compute L1-ONPP bases.

3.1 L2-ONPP and L2-MONPP

ONPP is a linear extension of LLE, which is a nonlinear dimensionality reduction technique that finds lower-dimensional embeddings of high-dimensional data samples, but the disadvantage of this embedding is a non-explicit mapping, in the sense that embedding is data dependent. In LLE, the inclusion or exclusion of any data point will result in the learning of entirely different manifold. Hence, in tasks such as recognition or classification of out-of-sample data point, LLE fails. ONPP solves the problem of out-of-sample data and finds the explicit mapping of the data points in lower-dimensional subspace through a linear orthogonal projection matrix. This orthogonal projection matrix can embed the new test data point into the lower-dimensional subspace making tasks such as recognition or classification of out-of-sample data possible. Another variant of ONPP is modified ONPP [11], that suggests the use of piece-wise nonlinear weights to reconstruct the data and represent local nonlinearity present in the neighbourhood patch more effectively. ONPP and MONPP both use L2-norm optimization to find projection matrix, but they differ in the mechanism to assign weights to neighbours of a data points as explained below. However, ONPP and MONPP still inherit the susceptibility to the presence of outliers.

Let $\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_N}$ be the N data points from m-dimensional space and $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_N}] \in \mathcal{R}^{m \times N}$ be the data matrix. The basic task of the dimentionality reduction techniques is to find a non-orthogonal or an orthogonal projection matrix $\mathbf{V} \in \mathcal{R}^{m \times d}$ which projects the data point $\mathbf{x_i} \in \mathcal{R}^m$ in the higher-dimensional space to the embeddings $\mathbf{y_i} \in \mathcal{R}^d$ in the lower-dimensional space (as d is assumed to be less than m) such that $\mathbf{y_i} = \mathbf{V}^{\mathrm{T}}\mathbf{x_i}$.

ONPP algorithm achieves the projection matrix in three basic steps. The first step involves finding neighbours of a data point $\mathbf{x_i}$. In unsupervised mode, neighbours are either decided by k-NN method or by using ϵ-NN method, whereas in the supervised mode, neighbours are decided based on knowledge of class label. The second step considers local patch of a data point, where a linear relationship between a data point and its neighbours is expressed using reconstruction weights. In the third step, ONPP tries to achieve compactness in the lower-dimensional space through a minimization problem such that this linear relationship in the high-dimensional neighbourhood is preserved.

Let set of k neighbours of data point $\mathbf{x_i}$ be $\mathcal{N}_{x_i}$. First, data point $\mathbf{x_i}$ is approximated as a linear combination of its neighbours as $\sum_{j=1}^{k} w_{ij}\mathbf{x_j}$ where $\mathbf{x_j} \in \mathcal{N}_{x_i}$ and the weight w_{ij} indicates $\mathbf{x_j}$'s contribution in reconstructing $\mathbf{x_i}$. The optimum weights w_{ij} are computed by minimizing the sum of the reconstruction errors, i.e. sum of errors between all $\mathbf{x_i}$ and linear combination of its neighbours $\mathbf{x_j} \in \mathcal{N}_{x_i}$. The minimization problem can be posed as:

$$
\begin{aligned}
\arg\min \mathcal{E}(\mathbf{W}) = \arg\max_{W} \sum_{i=1}^{N} \parallel \mathbf{x_i} - \sum_{j=1}^{k} w_{ij}\mathbf{x_j} \parallel_2 \\
s.\,t. \ \sum_{j=1}^{k} w_{ij} = 1
\end{aligned}
\tag{5}
$$

The problem corresponding to each data point $\mathbf{x_i}$ can be solved individually as a least square problem. Let matrix $\mathbf{X_{N_i}}$ be a neighbourhood matrix such that each neighbour $\mathbf{x_j} \in \mathcal{N}_{x_i}$ constitutes its columns. Note that $\mathbf{x_i}$ is also included in $\mathbf{X_{N_i}}$ as one of its own neighbour, making dimension of $\mathbf{X_{N_i}}$ is $m \times k + 1$. Now, for each $\mathbf{x_i} \in \mathbf{X}$ Eq. (5) can be written as an individual least square problem $(\mathbf{X_{N_i}} - \mathbf{x_i}\mathbf{e}^{\mathrm{T}})\mathbf{w_i} = \mathbf{0}$ for a data point $\mathbf{x_i}$ with a constraint $\mathbf{e}^{\mathrm{T}}\mathbf{w_i} = \mathbf{1}$, which results in a closed form solution for $\mathbf{w_i}$ as shown in Eq. (6). Here, $\mathbf{w_i}$ is a reconstruction weight vector of dimension $k \times 1$ and $\mathbf{e}$ is a vector of ones of dimension $k \times 1$.

$$
\mathbf{w_i} = \frac{\mathbf{G_i}^{-1}\mathbf{e}}{\mathbf{e}^{\mathrm{T}}\mathbf{G_i}^{-1}\mathbf{e}}
\tag{6}
$$

where $\mathbf{G_i}$ is Gramiam matrix of dimension $k \times k$. Each element of $\mathbf{G}$ is calculated as $\mathbf{g_{pl}} = (\mathbf{x_i} - \mathbf{x_p})^{\mathrm{T}}(\mathbf{x_i} - \mathbf{x_l})$, for $\forall\, \mathbf{x_p}, \mathbf{x_l} \in \mathcal{N}_{x_i}$. Detailed discussion on reconstruction weights can be found in [10, 19].

On the other hand, a variant of ONPP, modified orthogonal neighbourhood preserving projections (MONPP) stresses on the fact that the local neighbourhood patch assumed to be lying on or near a linear manifold may have some inherent nonlinearity. To take this nonlinearity into account while approximating a data point using its neighbours, MONPP uses nonlinear weights incorporating Z-shaped function [11] in place of linear weights obtained using least square solution. Equation (7) is used to assign weight to each neighbour $\mathbf{x_j} \in \mathcal{N}_{\mathbf{x_i}}$ using Z-shaped function based on the distance d_{ij} between data points $\mathbf{x_i}$ and $\mathbf{x_j}$. Note that this equation is same as Eq. (6), where $\mathbf{G}^{-1}$ is replaced by $\mathbf{Z}$. The new weights are

$$\mathbf{w_i} = \frac{\mathbf{Z_i e}}{\mathbf{e^T Z_i e}} \tag{7}$$

Third and the last step finds the projection matrix $\mathbf{V} \in \mathcal{R}^{\mathbf{m \times d}}$ to reduce the dimensionality, using $\mathbf{V}$ the data point $\mathbf{x_i} \in \mathbf{R}^m$ is projected on lower-dimensional space as $\mathbf{y_i} \in \mathbf{R}^d$ $(d << m)$ assuming that the neighbours $\mathbf{x_j}$s used to approximate data point $\mathbf{x_i}$ using reconstruction weights w_{ij} can be used to reconstruct data point $\mathbf{y_i}$ in lower-dimensional space using the lower-dimensional embeddings $\mathbf{y_j}$s and corresponding weights w_{ij}. Problem of finding such embeddings can be posed as a minimization problem with cost function

$$\begin{aligned} \arg\min \mathcal{E}(\mathbf{Y}) &= \arg\min_{Y} \sum_{i=1}^{N} \| \mathbf{y_i} - \sum_{j=1}^{N} w_{ij}\mathbf{y_j} \|_2 \\ \arg\min \mathcal{E}(\mathbf{V}) &= \arg\min_{V} \sum_{i=1}^{N} \| \mathbf{V^T x_i} - \sum_{j=1}^{N} w_{ij}\mathbf{V^T x_j} \|_2 \\ &s.\,t.\ \mathbf{V^T V} = \mathbf{I_d} \end{aligned} \tag{8}$$

This optimization problem results in a eigenvalue problem with the closed form solution. The eigenvectors corresponding to the smallest d eigenvalues of matrix $\mathbf{M} = \mathbf{X}(\mathbf{I} - \mathbf{W})(\mathbf{I} - \mathbf{W}^{\mathrm{T}})\mathbf{X}^{\mathrm{T}}$ constitute the basis of the low-dimensional ONPP space. ONPP explicitly maps $\mathbf{X}$ to $\mathbf{Y}$, which is of the form $\mathbf{Y} = \mathbf{V}^{\mathrm{T}}\mathbf{X}$, where each column of $\mathbf{V}$ is an eigenvector of $\mathbf{M}$. Once the ONPP projection space is obtained, any test data point $\mathbf{x_l}$ can be embedded into the space using a simple matrix-vector product.

3.2 ONPP as a PCA on reconstruction error

Rewriting Eq. (8) in a matrix form, to establish the relationship between ONPP and PCA:

$$\begin{aligned} \arg\min \mathcal{E}(\mathbf{Y}) &= \arg\min_{\mathbf{Y}} \| \mathbf{Y} - \mathbf{YW} \|_2 \\ \arg\min \mathcal{E}(\mathbf{V}) &= \arg\min_{\mathbf{V}} \| \mathbf{V^T X} - \mathbf{V^T XW} \|_2 \\ &= \arg\min_{\mathbf{V}} \| \mathbf{V^T(X - XW)} \|_2 \\ \arg\min \mathcal{E}(\mathbf{V}) &= \arg\min_{\mathbf{V}} \| \mathbf{V^T Er} \|_2 \\ &s.\,t.\ \mathbf{V^T V} = \mathbf{I} \end{aligned} \tag{9}$$

Now, comparing the optimization problems of PCA (Eq. 2) and the optimization problem of ONPP (Eq. 9), both result in an eigenvalue problems and have closed form solutions in terms of eigenvectors. Equation (2) is maximization problem and thus the bases vectors of PCA are eigenvectors corresponding to largest d eigenvalues, whereas Eq. (9) is minimization problem and thus the desired ONPP bases are eigenvectors corresponding to smallest d eigenvalues.

In other words, ONPP can be stated as a PCA of reconstruction errors, and conventional ONPP algorithm is essentially finding bases vectors $\mathbf{V}$ such that it actually captures the directions of minimum variances of reconstruction error. Thus, finding the strongest ONPP basis is same as finding weakest basis of PCA when performed on the reconstruction errors $\mathbf{Er}$. Each column of $\mathbf{Er}$ is a reconstruction error for ith data point, which is calculated using its neighbours and corresponding weights found using (6) or (7) using

$$\mathbf{er_i} = \mathbf{x_i} - \sum_{j=1}^{k} w_{ij}\mathbf{x_j} \tag{10}$$

This relationship between PCA and ONPP bases is verified in experiments performed on the synthetic data, and it is observed that the ONPP bases obtained using conventional L2-ONPP algorithm and ONPP bases obtained using L2-PCA on reconstruction error are same. Details of this experiment are documented in Sect. 4.

3.3 L1-ONPP using PCA on reconstruction error

Once the relationship between L2-PCA bases and L2-ONPP bases is in place, it is evident that PCA algorithm can also be used to find ONPP bases. This led to the use of existing L1-norm-based PCA algorithms to solve L1-ONPP optimization problem. Rewriting L2-ONPP optimization problem in Eq. (8) using L1-norm minimization, we have

$$\begin{aligned} \arg\min \mathcal{F}(\mathbf{Y}) &= \arg\min_{\mathbf{Y}} \sum_{i=1}^{N} \| \mathbf{y_i} - \sum_{j=1}^{N} w_{ij}\mathbf{y_j} \|_1 \\ &s.\,t.,\ \mathbf{V^T V} = I \end{aligned} \tag{11}$$

In matrix form,

$$\begin{aligned} \arg\min \mathcal{F}(\mathbf{Y}) &= \arg\min_{\mathbf{Y}} \| \mathbf{Y} - \mathbf{YW} \|_1 \\ \arg\min \mathcal{F}(\mathbf{V}) &= \arg\min_{\mathbf{V}} \| \mathbf{V^T X} - \mathbf{V^T XW} \|_1 \\ &= \arg\min_{\mathbf{V}} \| \mathbf{V^T(X - XW)} \|_1 \\ &= \arg\min_{\mathbf{V}} \| \mathbf{V^T Er} \|_1 \end{aligned} \tag{12}$$

the problem stated in Eq. (12) is similar to problem stated in PCA-L1 (Eq. 4), and thus the solution of (Eq. 4) can be used to solve Eq. (12). L1-ONPP bases can be found using any L1-norm-based PCA algorithm when performed on reconstruction error matrix $\mathbf{Er}$. As discussed in Sect. 1, many L1-norm-based PCA methods have been developed which

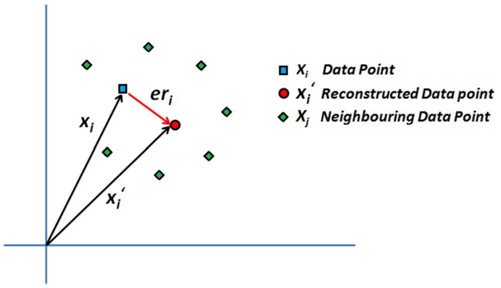

x_i Data Point
x_i' Reconstructed Data point
x_j Neighbouring Data Point
er_i
x_i
x_i'

Fig. 1 Illustration of data point $\mathbf{x_i}$ (represented by a blue square), its reconstruction $\mathbf{x_i'}$ using neighbours (represented by a red circle) $\mathcal{N}_{\mathbf{x_i}}$ and error vector $\mathbf{er_i}$ (represented by a green diamond). i^{th} reconstruction error vector is denoted by $\mathbf{er_i}$

find bases vectors through linear or quadratic programming. These methods are computationally expensive. The PCA-L1 algorithm [12] used here is a robust as well as fast L1-norm-based method. PCA-L1 converts the L1-norm variance into a direct sum of signed training points into projection space. The bases vectors are updated by the sum of resigned training points. As a result, the convergence procedure is fast. Refer to [12] for the proof.

L2-ONPP involves closed form solution which involves eigenvalue problem of matrix size $m \times m$. L1-ONPP is computationally costly because it involves an iterative procedure because each basis vector v_k starts from a random m-dimensional vector and polarity check, flipping and maximization are performed iteratively until v_k converges.

Comparing Eq. (12) of L1-ONPP with Eq. (4) of L1-PCA, we can intuitively state that the component in the direction of minimum variance gives the strongest ONPP basis. Considering reconstruction error between a data point $\mathbf{x_i}$ and its approximation $\mathbf{x_i'}$ as a vector $\mathbf{er_i}$, which is also a point in m-dimensional space (as shown in Fig. 1) L1-PCA can be performed on the reconstruction errors to search for the d-dimensional space such that the bases vectors are in the direction of minimum variances of these reconstruction errors. Such bases can be computed using PCA-L1 algorithm, and the detailed algorithm is given in Table 1.

4 EXPERIMENTS

To validate the theoretical conclusion on the relationship between ONPP and PCA, experiments were performed on the synthetic as well as real data as documented in this section.

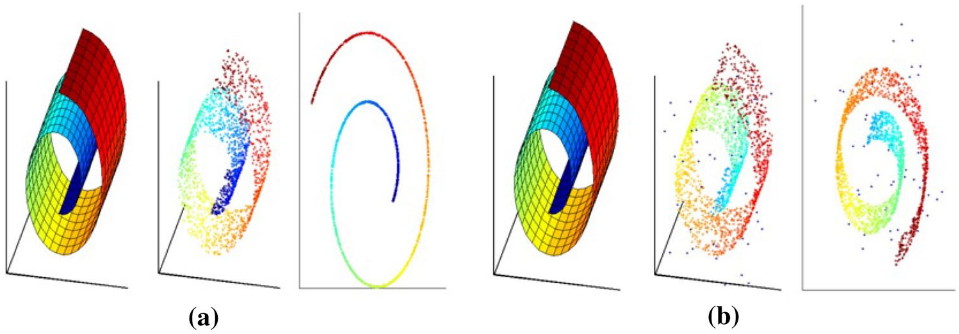
4.1 A small problem with Swiss-role data

In the literature, definition of outlier is given as a data point that seems to be taken from an entirely different distribution.

Table 1 L1-ONPP Algorithm

```
    Input: Dataset $X \in R^{m \times N}$and number of reduced dimension $d$
    Output: Lower dimension projection $Y \in R^{d \times N}$
1:  Compute NN with class label information (in supervised mode) or using $k$-NN algorithm (in unsupervised mode).
2:  Compute the weight $w_{ij}$ for each neighbor of $\mathbf{x_i}$ data point $\mathbf{x_j} \in \mathcal{N}_{\mathbf{x_i}}$ as given in equation (6) or (7)
3:  Compute reconstruction error matrix $\mathbf{Er}$ using equation (11)
4:  Set $\mathbf{v_0} = \mathbf{0}$, $\mathbf{Er^0} = \mathbf{Er}$
5:  for $(j = 1, j \leq d, ++j)$ do
6:      Set $\mathbf{Er^j} = \mathbf{Er^{j-1}} - \mathbf{v_{j-1}}(\mathbf{v_{j-1}^T}\mathbf{Er^{j-1}})$
7:      Initialize $\mathbf{v(0)}$
8:      Set $t = 0$
9:      for$(i = 1; i \leq N; ++i)$ do
10:         if $\mathbf{v(t)^T er_i{}^j} < 0$ then
11:             $\mathbf{p_i(t)} = -1$
12:         else
13:             $\mathbf{p_i(t)} = 1$
14:         end if
15:     end for
16:     Set $t = t + 1$
17:     Set $\mathbf{v(t)} = \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{p_i(t) er_i{}^j}$
18:     Set $\mathbf{v(t)} \leftarrow \mathbf{v(t)} / \parallel \mathbf{v(t)} \parallel_\mathbf{2}$
19:     if $\mathbf{v(t)} \neq \mathbf{v(t-1)}$ then
20:         Go to Step 9.
21:     else if There exists $i$ such that $\mathbf{v^T(t) er_i{}^j} = \mathbf{0}$ then
22:         Set $\mathbf{v(t)} \leftarrow (\mathbf{v(t)} + \mathit{\Delta}\mathbf{v}) / \parallel \mathbf{v(t)} + \mathit{\Delta}\mathbf{v} \parallel_2$ and go to step 9.
            (Here, $\mathit{\Delta}\mathbf{v}$ is a small nonzero random vector.)
23:     else
24:         Set $\mathbf{v_j} = \mathbf{v(t)}$
25:     end if
26: end for
27: Project data $\mathbf{X}$ on L1-ONPP projection space using $\mathbf{V}$ to get embeddings $\mathbf{Y} = \mathbf{V^T X}$.
```

Fig. 2 L2-ONPP performed on Swiss-role data. **a** Continuous manifold (left), sampled 3D data (middle) and its 2D representation using the strongest 2 basis of ONPP(right). **b** Continuous manifold (left), sampled 3D data corrupted with additional outliers from uniform distribution (middle) and its 2D representation using the strongest 2 basis of ONPP (right)



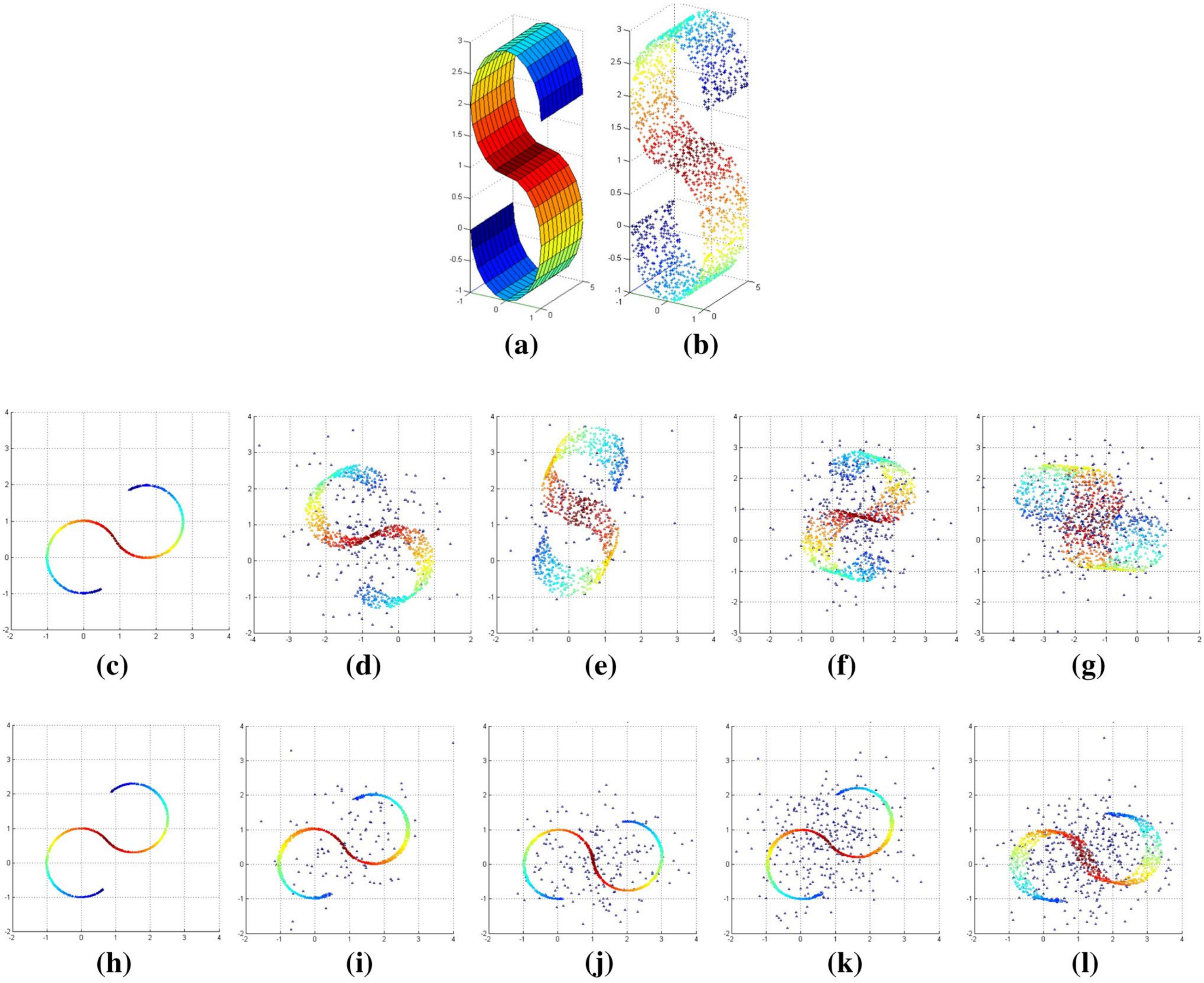


Fig. 3 Manifold learning on S-curve data. **a** Continuous manifold. **b** Sampled 1000 3D clean data points. Its 2D representation using the strongest 2 basis of L2-ONPP starting with (**c**–**g**) with clean data, 100, 200, 300 and 400 outliers, respectively. Its 2D representation using strongest 2 basis of L1-ONPP starting with (**h**–**l**) with clean data, 100, 200, 300 and 400 outliers, respectively

To observe the effect of outliers on L2-ONPP algorithm, an experiment was performed on Swiss-role data. Over 2000 three-dimensional data points were randomly sampled from a continuous Swiss-role manifold. Two-dimensional embeddings of clean data were found using L2-ONPP as shown in Fig. 2a right. Now, 50 data points (nearly 2.5% of clean data) from a normal distribution are added to these 2000 clean data point as outliers, two-dimensional embeddings of this data are also found using L2-ONPP as shown in 2b right. Comparing embeddings from clean data (Fig. 2a) and embeddings from data having outliers (Fig. 2b), it can be observed that global structure as well as local geometry is well preserved in the case of clean data, whereas in the case of noisy data (Fig. 2b), two-dimensional representation is distorted. The reason is all neighbours of the clean data point may not lie on locally linear patch of a manifold in the presence of outliers, which leads to the biased reconstruction. On the other hand, the neighbourhood patch of the outlier will be comparatively very large and thus does not capture local geometry very well, as the effect of large distance is exaggerated by the use of L2-norm. It has been known that L2-norm-based techniques are not robust, in the sense that the presence of outliers can arbitrarily skew the solution from the desired solution.

Another experiment was performed to analyse the effectiveness of the proposed algorithm in the presence of different amounts of outliers. As shown in Fig. 3b, 1000 3D clean data points were sampled from S-shaped continuous manifold shown in Fig. 3a. Figure 3c, h shows 2D representation of clean data using L2-ONPP and L1-ONPP, respectively. The clean data are then corrupted with 100, 200, 300 and 400 outliers sampled from an uniform distribution. Figure 3d–g shows 2D representation of noisy data using L2-ONPP, and Fig. 3i–l shows 2D representation of noisy data using L1-ONPP. As it can be seen from this experiment L1-ONPP very well handles outliers by preserving intrinsic neighbourhood relations as well as global geometry of data. On the contrary, increasing density of outliers distorts the learned manifold in increasing manner, the presence of

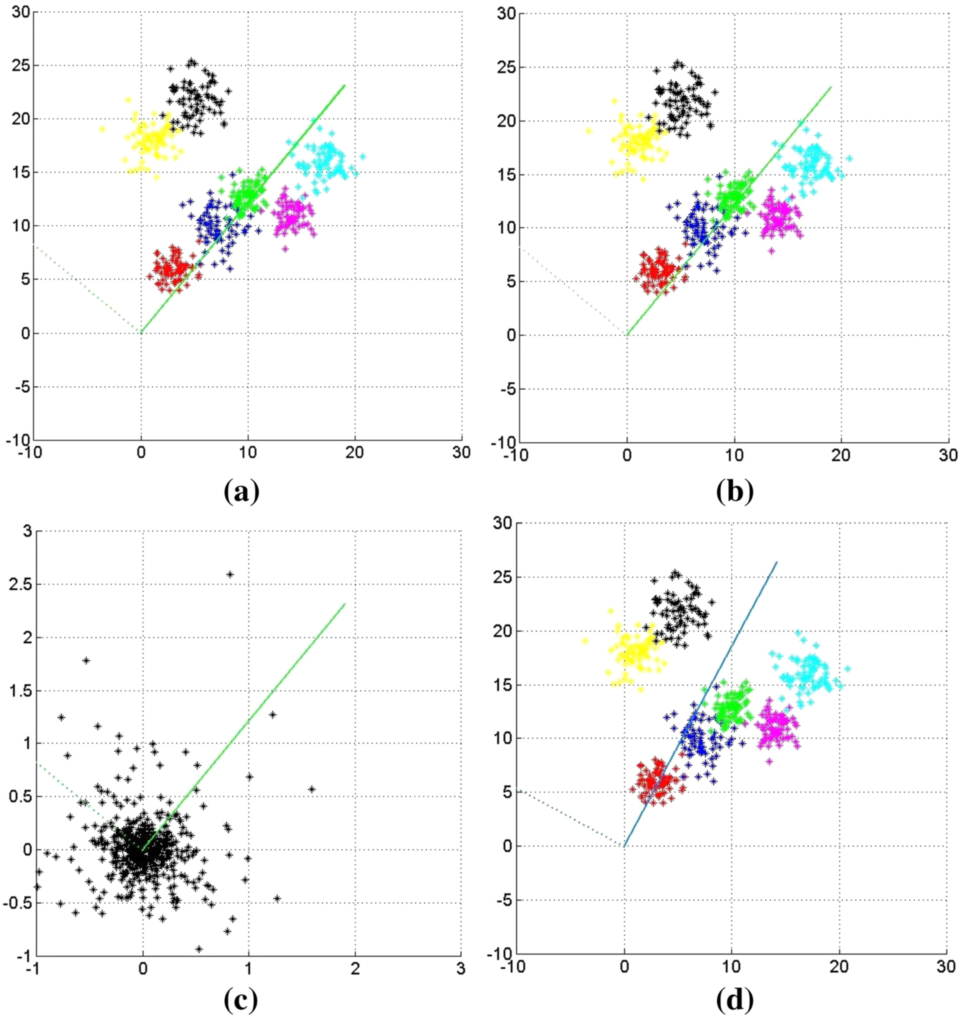


Fig. 4 A toy example with 700 data samples from 7 clusters. Solid line represents first projection basis and dotted line represents second projection basis **a** Projection basis using conventional L2-ONPP. **b** Projection basis using L2-PCA on reconstruction basis. **c** Projection basis overlapped on reconstruction errors. **d** Projection basis using proposed L1-ONPP

outliers even affects the orientation of data as can be seen in Fig. 3d–e.

4.2 Comparing bases of L2-PCA, L2-ONPP and L1-ONPP

To validate the relationship between PCA and ONPP as described in Sect. 3.1, the experiment was performed on synthetically generated data. 2D data were randomly generated to form 7 clusters with 100 data point each resulting in 700 data points. The clusters are closely placed and slightly overlapping, and 2 out of 7 were slightly separated as shown in Fig. 4a. L2-ONPP bases were found using the conventional algorithm, and another set of bases vectors were computed by performing L2-PCA on reconstruction error. The bases found using both methods are same.

L2-norm ONPP basis [Fig. 4a]

1st basis : $[0.6361, 0.7716]^T$ 2nd basis: $[-0.7716, 0.6361]^T$

PCA basis on reconstruction errors [Fig. 4b]

1st basis : $[0.6360, 0.7717]^T$ 2nd basis: $[-0.7717, 0.6360]^T$

Figure 4c shows that L2-ONPP bases are essentially pointing the direction in which the variance of reconstruction error is minimum. Note that the reconstruction error of all data point is centred at origin (same as the assumption in PCA that the data points are mean centred). For this data, L1-ONPP bases were computed using L1-PCA algorithm. As it can be seen from Fig. 4d, the projection bases are tilted towards the outlier data.

L1-norm ONPP basis [Fig. 4b]

1st basis : $[0.4741, 0.8805]^T$ 2nd basis: $[-0.8805, 0.4741]^T$

In this experiment, the residual error was observed for both, L2-ONPP and L1-ONPP. Residual error is a measure of how well the information is preserved while projecting data on the lower-dimensional space using few strongest bases, while discarding other dimensions. In this case, the data were projected using only one dimension using the strongest basis vector. The average residual error was calculated using

$$\mathbf{e}_{\mathbf{avg}} = \frac{\mathbf{1}}{\mathbf{N}} \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{x}_{\mathbf{i}} - \mathbf{v}_{\mathbf{1}}\left(\mathbf{v}_{\mathbf{1}}^{\mathbf{T}}\mathbf{x}_{\mathbf{i}}\right) \qquad (13)$$

Table 2 Comparison of performance in terms of residual error and classification error (in %) of L2-ONPP and L1-ONPP with varying number of dimensions on IRIS data

Dim	Residual error		Classification error (%)	
	L2-ONPP	L1-ONPP	L2-ONPP	L1-ONPP
1	7.8614	7.8546	16.00	13.00
2	7.7450	7.6730	6.67	4.00
3	7.0055	5.6545	5.33	2.67
4	1.44e−15	1.45e−15	6.67	6.67

The average residual errors of L2-ONPP and L1-ONPP are 2.3221 and 0.7894, respectively. Thus. it can be concluded that L1-ONPP is less susceptible to outliers compared to L2-ONPP. The same behaviour related to residual error is observed with real data also as stated in the following experiment.

4.3 Experiment with IRIS dataset

To further compare behaviour of L2-ONPP and L1-ONPP regarding residual error and performance in classification task, Iris data form UCI Machine Learning Repository [4] are used. The data set contains 150 instances of 4-dimensional data belonging to three different classes. The residual error obtained while reconstructing the data using varying number of dimensions is shown in Fig. 5. Table 2 lists the residual errors using different numbers of dimensions; as it can be seen, the residual error is less in L1-ONPP as compared to L2-ONPP which significantly improves classification accuracy at lower dimensions. When all 4 dimensions are used in projection space, the projection of data spans

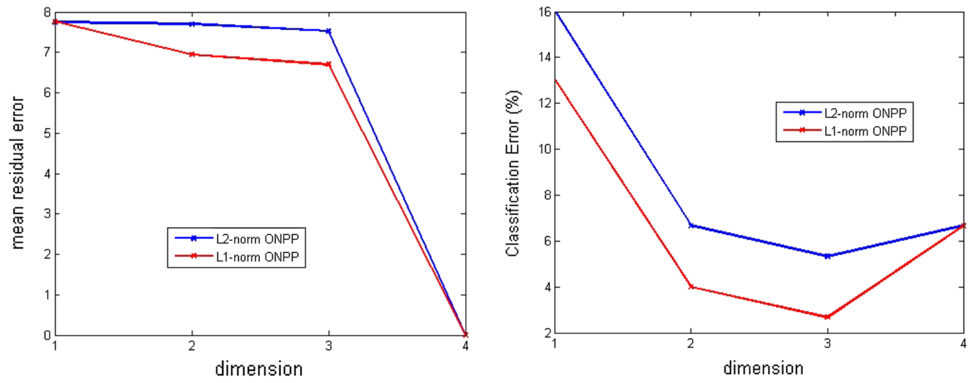


Fig. 5 Performance comparison of L2-ONPP and L1-ONPP with respect to varying number of dimensions used to reconstruct the IRIS data in terms of Residual Error (left) Classification Error (right)

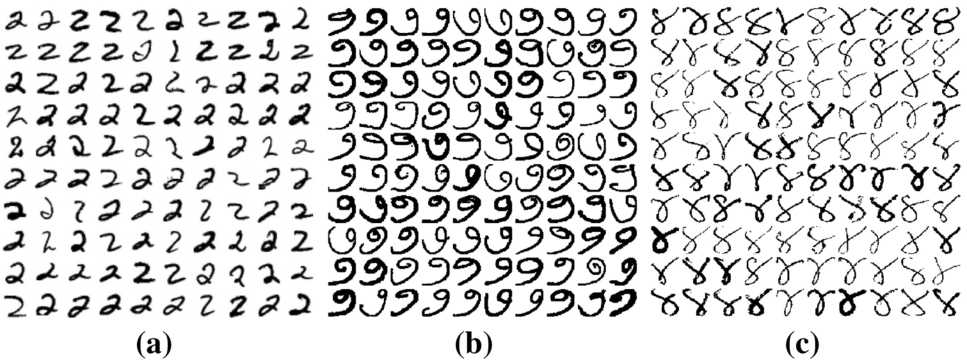


Fig. 6 **a** Examples of 2 s in the MNIST database first 100 examples. **b** Examples of 7 s in the Gujarati database first 100 examples. **c** Examples of 4 s in the Devnagari numerals database first 100 examples. Notice the very diverse shape, stroke width, orientation and pattern of different digits

entire original space, and thus the residual error drops to almost zero for both methods, L2-ONPP and L1-ONPP. Similar behaviours expected for classification also at 4 dimensions; as can be seen from Table 2, the classification error at 4 dimensions yields greater than the lower dimension representation because it includes the redundant details present in higher dimensions. The same behaviour at higher dimensions can be observed in all dimensionality reduction techniques. Here, nearest neighbour (NN) is used as a classifier.

4.4 Experiment with handwritten numerals

Handwritten text or numerals have huge variations in terms of shape, stroke width, orientations and pattern, thus making an ideal data to test the capacity of L1-ONPP to handle outliers. To compare performance of L1-ONPP and L2-ONPP with real-world data having outliers, three handwritten numeral databases, one English and two different Indian languages Gujarati and Devnagari, are used. The images in each database are resized to 30 × 30 to maintain uniformity across all three databases.

English numerals

The MNIST database [13] is a large database of handwritten English digits which contains nearly 70,000 images of each digit. Some of the examples of digit ′2′ are shown in Fig. 6a. 1000 samples were selected randomly such that each digit is equally present in training, while remaining samples were used for testing. Average recognition accuracy of 20 randomization are reported here. Performance of L2-ONPP and L2-MONPP is compared with L1-ONPP and L1-MONPP with varying number of dimensions as shown in Fig. 7. As it can be seen, L1-ONPP and L1-MONPP outperform their L2-norm counterparts with large difference. Best average recognition accuracy of L1-ONPP and L1-MONPP is almost same, nearly 87.52% achieved at 210 dimensions. As it can be seen from Fig. 7, the performance of L2-ONPP and L2-MONPP is poor compared to its L1-norm counterparts.

Gujarati numerals

The Gujarati Handwritten numeral dataset [5, 18] have nearly 1300 samples for each digit. Some of the examples of digit ′7′ are given in Fig. 6b to show the large variations in the dataset. Randomly, 1000 samples were selected such that each digit is well represented in training data and the

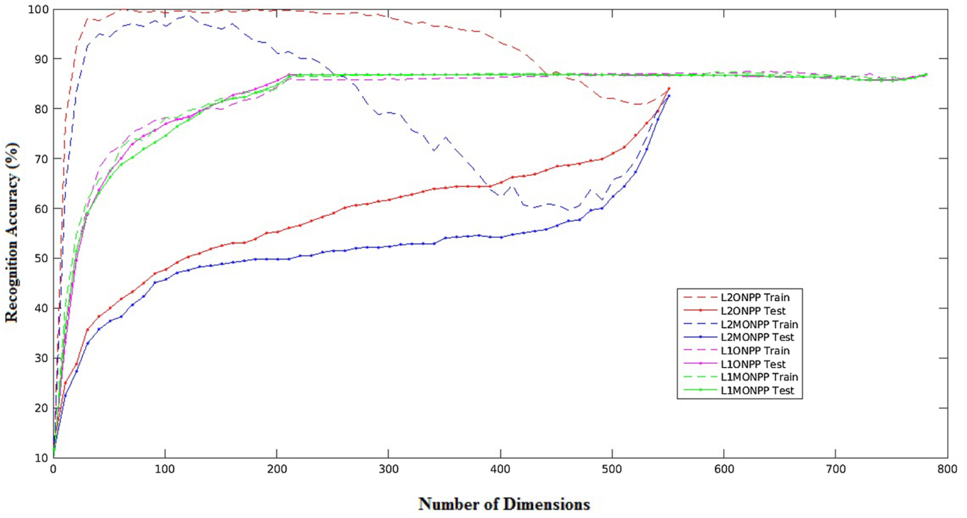


Fig. 7 Performance comparison in terms of recognition accuracy for L2-ONPP, L1-ONPP, L2-MONPP and L1-MONPP for MNIST handwritten numerals database

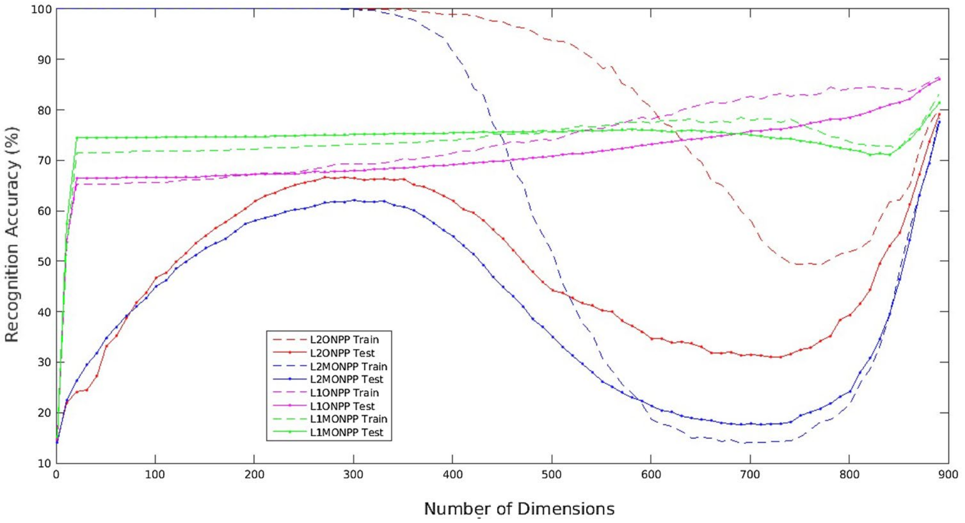


Fig. 8 Performance comparison in terms of recognition accuracy for L2-ONPP, L1-ONPP, L2-MONPP and L1-MONPP for Gujarati handwritten numerals

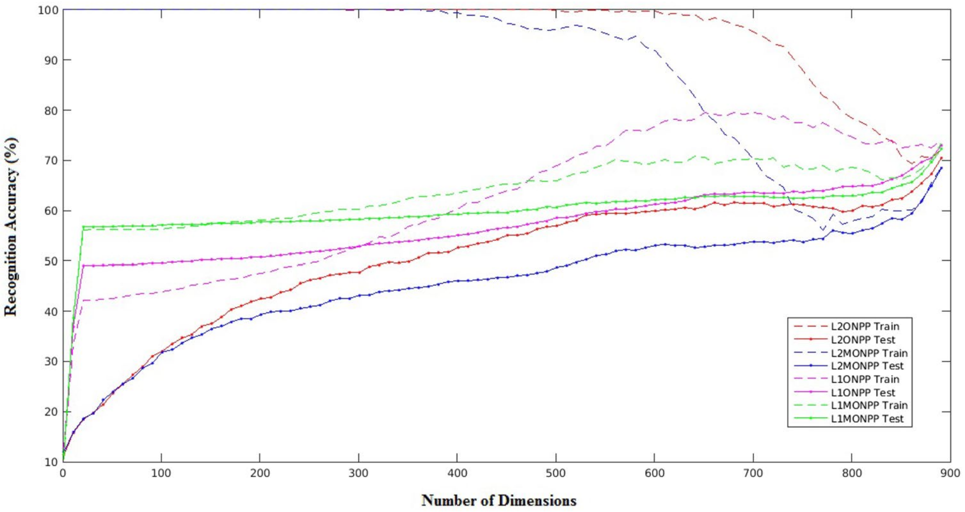


Fig. 9 Performance comparison in terms of recognition accuracy for L2-ONPP, L1-ONPP, L2-MONPP and L1-MONPP for Devnagari handwritten numerals database

remaining samples were used as testing data. Recognition accuracy of L2-ONPP, L2-MONPP, L1-ONPP and L1-MONPP with varying number of dimensions are compared in Fig. 8. Average recognition accuracy achieved by L1-ONPP and L1-MONPP is 75.20% and 67.32% nearly at 20 dimensions, whereas that of L2-ONPP and L2-MONPP is 67.16% and 61.28% nearly at 290 dimensions, respectively. As it can be seen, the performance of L1-ONPP and L1-MONPP stabilizes nearly at 30 dimensions, but the performance of L2-ONPP and L2-MONPP deteriorates after 300 dimensions due to the presence of redundant information present at higher dimensions, but the use of all dimensions again leads to almost similar recognition as that of L1-norm counterparts as observed with IRIS data.

Devnagari numerals

The Devnagari handwritten database [2] has approximately 1800 sample of each digit. Randomly selected samples of digit ‘4’ are shown in Fig. 6c. Randomly 900 samples are used for training data, and the remaining are used for testing. As it can be seen from Fig. 9, L1-ONPP and L1-MONPP achieve nearly 50% and 60% recognition accuracy at 30 dimensions, respectively, whereas the performance of L2-ONPP and L2-MONPP is consistently poor compared to L1-norm counterpart.

The purpose of this experiment is not to show recognition performance of L1-ONPP and compare it with other state-of-the-art OCR techniques, but to compare L2-ONPP and L1-ONPP when data are very diverse and have large variability and to show the capacity of L1-ONPP and L1-MONPP of handling such diverse data. With a larger number of training data, L1-ONPP proves to be a better at recognizing digits compared to L2-ONPP. L2-MONPP and L1-MONPP also perform at par with L2-ONPP and L1-ONPP. Here, nearest neighbour (NN) is used for classification, the use of sophisticated classifier like SVM can lead to improved recognition accuracy. The proposed L2-ONPP algorithm converges in about 16 iterations for a single basis vector when nearly 1000 samples of images having size 30 × 30 are used for training and the procedure of learning L1-ONPP bases took average 3 min. On the other hand, having an closed form solution, L2-ONPP takes on an average 7 s to learn the bases. The configuration of the machine used is as follows: Intel Xeon® E5-2620 @ 2.40GB, 24 core, 64-bit with 2GB RAM allocation.

5 CONCLUSION

Linear dimensionality reduction techniques such as PCA, LDA, LPP and ONPP solve an optimization problem based on some criteria. Usually, the optimization problem is defined using L2-norm. However, use of L2-norm makes these techniques susceptible to outliers present in the data. The present work is first attempt to compute bases vectors for ONPP using L1-norm. In particular, a relation is established to show that ONPP bases can be obtained by performing PCA on reconstruction error. These phenomenon is established both theoretically and experimentally. An existing algorithm of finding PCA bases using L1-norm optimization is applied to compute the L1-ONPP bases. It has also been proved experimentally that the residual error calculated after discarding few dimensions in projection space and reconstructing data with less number of dimensions is comparatively low in the case of L1-ONPP than that of L2-ONPP. Experiments are performed for synthetic as well as real data, and the same conclusion as mentioned above is observed. Performance of L1-ONPP is compared with L2-ONPP on numeral recognition task, and it is observed that with larger number of training data, L1-ONPP outperforms L2-ONPP with huge margin, but being an iterative method, L1-ONPP is computationally expensive compared to L2-ONPP.

References

1. Baccini A, Besse P, De Falguerolles A (1996) A L1-norm PCA and a heuristic approach. Ordinal Symb Data Anal 1:359–368
2. Bhattacharya U, Chaudhuri B (2005) Databases for research on recognition of handwritten characters of indian scripts. In: Proceedings eighth international conference on document analysis and recognition, 2005. IEEE, pp 789–793
3. Ding C, Zhou D, He X, Zha H (2006) R1-PCA: rotational invariant l 1-norm principal component analysis for robust subspace factorization. In: Proceedings of the 23rd international conference on machine learning, pp 281–288
4. Dua D, Karra Taniskidou E (2017) UCI machine learning repository. University of California, School of Information and Computer Science, Irvine, CA. http://archive.ics.uci.edu/ml. Accessed 03 Aug 2018
5. Goswami MM, Mitra SK (2015) Offline handwritten gujarati numeral recognition using low-level strokes. Int J Appl Pattern Recognit 2(4):353–379
6. He X, Niyogi P (2004) Locality preserving projections. Adv Neural Inf Process Syst 16:153–160
7. He X, Cai D, Yan S, Zhang HJ (2005) Neighborhood preserving embedding. In: Tenth IEEE international conference on computer vision, ICCV 2005, vol 2. IEEE, pp 1208–1213
8. Ke Q, Kanade T (2005a) Robust L1 norm factorization in the presence of outliers and missing data by alternative convex programming. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol 1, pp 739–746
9. Ke Q, Kanade T (2005b) Robust L1 norm factorization in the presence of outliers and missing data by alternative convex programming. In: IEEE computer society conference on computer vision and pattern recognition, 2005. CVPR 2005, vol 1. IEEE, pp 739–746
10. Kokiopoulou E, Saad Y (2007) Orthogonal neighborhood preserving projections: a projection-based dimensionality reduction technique. IEEE Trans Pattern Anal Mach Intell 29(12):2143–2156
11. Koringa P, Shikkenawis G, Mitra SK, Parulkar S (2015) Modified orthogonal neighborhood preserving projection for face recognition. In: Kryszkiewicz M, Bandyopadhyay S, Rybinski H, Pal SK (eds) Pattern recognition and machine intelligence. Springer, Berlin, pp 225–235
12. Kwak N (2008) Principal component analysis based on L1-norm maximization. IEEE Trans Pattern Anal Mach Intell 30(9):1672–1680
13. Lecun Y, Cortes C (2009) The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/. Accessed 03 Aug 2018
14. Li M, Yuan B (2005) 2D-LDA: a statistical linear discriminant analysis for image matrix. Pattern Recognit Lett 26(5):527–532
15. Li X, Pang Y, Yuan Y (2010a) L1-norm-based 2DPCA. IEEE Trans Syst Man Cybern Part B (Cybern) 40(4):1170–1175
16. Li X, Pang Y, Yuan Y (2010b) L1-norm-based 2DPCA. IEEE Trans Syst Man Cybern Part B (Cybern) 40(4):1170–1175
17. Lu J, Plataniotis KN, Venetsanopoulos AN (2003) Face recognition using LDA-based algorithms. IEEE Trans Neural Netw 14(1):195–200
18. Nagar R, Mitra SK (2015) Feature extraction based on stroke orientation estimation technique for handwritten numeral. In: 2015 eighth international conference on advances in pattern recognition (ICAPR), pp 1–6
19. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323–2326
20. Shikkenawis G, Mitra SK (2012) Improving the locality preserving projection for dimensionality reduction. In: Third international conference on emerging applications of information technology (EAIT), 2012. IEEE, pp 161–164
21. Turk M, Pentland A (1991) Eigenfaces for recognition. J Cognit Neurosci 3(1):71–86
22. Wang H, Lu X, Hu Z, Zheng W (2014) Fisher discriminant analysis with L1-norm. IEEE Trans Cybern 44(6):828–842. https://doi.org/10.1109/TCYB.2013.2273355
23. Zhang D, Zhou ZH (2005) (2D) 2PCA: two-directional two-dimensional pca for efficient face representation and recognition. Neurocomputing 69(1):224–231
24. Zhang H, Wu QJ, Chow TW, Zhao M (2012) A two-dimensional neighborhood preserving projection for appearance-based face recognition. Pattern Recognit 45(5):1866–1876
25. Zhong F, Zhang J (2013) Linear discriminant analysis based on L1-norm maximization. IEEE Trans Image Process 22(8):3018–3027