Pattern Recognition Letters 34 (2013) 679–685

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

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

Generalization of linear discriminant analysis using L_p-norm

Jae Hyun Oh[1], Nojun Kwak[*,2]

Department of Electrical & Computer Engineering, Ajou University, San 5, Woncheon-dong, Yeongtong-gu, Suwon 443-749, Republic of Korea

ARTICLE INFO

Article history:
Received 16 July 2012
Available online 4 February 2013

Keywords:
LDA
Norm
Outlier
LDA-L_p

ABSTRACT

In this paper, the linear discriminant analysis (LDA) is generalized by using an L_p-norm optimization technique. Although conventional LDA based on the L_2-norm has been successful for many classification problems, performances can degrade with the presence of outliers. The effect of outliers which is exacerbated by the use of the L_2-norm can cause this phenomenon. To cope with this problem, we propose an LDA based on the L_p-norm optimization technique (LDA-L_p), which is robust to outliers. Arbitrary values of p can be used in this scheme. The experimental results show that the proposed method achieves high recognition rate for many datasets. The reason for the performance improvements is also analyzed.

1. Introduction

During the last few decades, numerous feature extraction methods have been proposed for data analysis and object classification in the computer vision and pattern recognition communities. Principal component analysis (PCA) (Fukunaga, 1990; Turk et al., 1991), independent component analysis (ICA) (Bell and Sejnowski, 1995; Kwak and Choi, 2003) and linear discriminant analysis (LDA) (Belhumeur et al., 1997; Martinez and Kak, 2001) are successful representatives of linear subspace-based feature extraction methods, and many further improvements continue to be researched. Unlike PCA and ICA, LDA is designed for supervised learning and has been widely used for classification problems. The goal of LDA is to find a series of projections that maximize the ratio of between class and within class variance, both of which are based on the L_2 norm. It is known that conventional L_2-norm based LDA is optimal if each class has the same Gaussian distribution. Although conventional LDA, based on the L_2-norm, has been successful for many problems, there are numerous problems whose class-specific distributions are far from Gaussian. For these problems, the performances of LDA could degrade with the presence of outliers because L_2-norm-based methods are dominated by samples with large norms.

As a generalized version of LDA, Yang et al. (2011) introduced a new concept of designing a discriminant analysis method and Yang and Yang (2003) suggested a complete PCA plus LDA algorithm. A new kernel Fisher discriminant analysis framework was also proposed to implement the KPCA plus LDA strategy (Yang et al., 2005). An extension of LDA to regression problems and its kernel version were also proposed in (Kwak and Lee, 2010; Kwak, 2012), respectively.

There are many studies aimed at enhancing the performance of the conventional L_2-norm-based feature extraction methods. In particular, many studies have focused on PCA algorithms based on the L_1-norm instead of the L_2-norm. L_1-norm-based PCA (L1-PCA) Ke and Kanade (2005) finds the optimal projection vectors that minimize the L_1-norm-based reconstruction error in the input space through linear or quadratic programming which is computationally expensive. Another drawback of L1-PCA is that it is not rotational invariant. Ding et al. (2006) proposed R1-PCA, which combines the merits of L2-PCA and those of L1-PCA. Unlike L1-PCA, it is rotation-invariant while it successfully suppresses the effect of outliers, as L1-PCA does. On the other hand, PCA-L1 (Kwak, 2008) maximizes L_1-norm-based dispersion in the feature space, instead of maximizing L_2-norm-based variance, to achieve robust and rotation-invariant PCA. Several extensions of PCA-L1 have been introduced recently. 2DPCA-L1 (Li et al., 2009) is an L_1-norm version of 2DPCA that is robust to outliers with very simple iteration process. In addition, Kwak and Oh (2009) proposed SL1-BDA, an L_1-norm version of biased discriminant analysis that was originally developed for one-class classification problems. It tries to reduce the negative effect of extracting features due to negative samples that are very far from the center of positive samples and utilizes the L_1-norm instead of the L_2-norm.

There are also studies that try to extend LDA using other norms than the L_2-norm. The novel rotation-invariant L_1-norm (R_1-norm)-based discriminant criterion called DCL_1, which better characterizes intra-class compactness and inter-class separability by using the rotation-invariant L_1-norm, was proposed in (Li et al., 2010). In addition, robust L1-norm based tensor analysis (TPCA-L1)

[*] Corresponding author. Tel.: +82 (0) 31 219 2480; fax: +82 (0) 31 212 9531.
E-mail addresses: hyunsda@ajou.ac.kr (J.H. Oh), nojunk@ajou.ac.kr, nojunk@ieee.org (N. Kwak).

[1] Jae Hyun Oh is pursuing a Ph.D. degree at the Department of Electrical & Computer Engineering, Ajou University, Republic of Korea.

[2] Nojun Kwak is an associate professor at the Department of Electrical & Computer Engineering, Ajou University, Republic of Korea.

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.01.016

formulates the reconstruction error with the L_1-norm (Pang et al., 2010). The use of the L_1 norm makes tensor analysis robust to outliers. Moreover, the algorithm converges well in several iterations. Fast Haar transform (FHT) based PCA and FHT-based spectral regression discriminant analysis have also been proposed to solve the problem of the computationally expensive processing time of the projection process (Pang et al., 2009). Recently, we studied the generalization of the L_1 norm to an L_p norm with an arbitrary p value for PCA (Kwak, 2013). This algorithm uses a new L_p-norm optimization technique using the gradient search method.

In this paper, a method is proposed for classification, which is based on the L_p-norm optimization technique as a generalized version of LDA. We address a novel method of LDA that uses the L_p-norm instead of the L_2-norm to obtain a robust and rotation-invariant version of LDA. The objective function is formulated using the general L_p-norm in both the numerator and denominator and the optimal solution is found using the steepest-gradient method. The effect of outliers for each method is analyzed, and it is shown that the proposed LDA based on the L_p-norm is more robust to outliers. In doing so, a novel methodology for measuring the effect of outliers is also presented.

This paper is organized as follows. In Section 2, conventional LDA is overviewed, and the new algorithm LDA-Lp which uses the L_p-norm instead of the L_2-norm is presented. Section 3 shows the experimental results with an analysis on the effect of outliers. Finally, conclusions are presented in Section 4.

2. Methods

2.1. LDA (based on the L_2-norm)

LDA is one of the well-known methods of supervised dimensionality reduction for classification problems. It tries to find transformations that maximize the ratio of the between-class and the within-class scatter matrices. Consider a dataset $\{(x_i, c_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^d$ and $c_i \in \{1, \dots, C\}$ are an input and the corresponding class, respectively. The between-class scatter matrix S_B and the within-class scatter matrix S_W are defined, respectively, as:

$$S_B = \sum_{c=1}^{C} N_c (m_c - m)(m_c - m)^T,$$
$$S_W = \sum_{i=1}^{N} (x_i - m_{c_i})(x_i - m_{c_i})^T, \tag{1}$$

where N_c is the number of samples belonging to class c, and $m \triangleq \frac{1}{N}\sum_{i=1}^{N} x_i$ and $m_c \triangleq \frac{1}{N_c}\sum_{i\in\{j|c_j=c\}} x_i$ are the total mean and the class mean of the input data.

The LDA is formulated to find M projection vectors $\{w_i\}_{i=1}^{M}$ that maximize Fisher's criterion, as follows:

$$W_{LDA} = \underset{W}{argmax} \frac{|W^T S_B W|}{|W^T S_W W|}. \tag{2}$$

Here, the ith column of W corresponds to w_i. Maximizing the above Fisher's criterion is equivalent to solving the following eigenvalue decomposition problem:

$$S_B w_i = \lambda_i S_W w_i \quad \lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_m. \tag{3}$$

Then, the linear projections $\{w_i\}_{i=1}^{M}$ can be obtained. However, conventional LDA is very sensitive to the presence of outliers, because both S_B and S_W in (1) are dominated by a set of outliers with large norms. To alleviate this problem, we propose a novel method that utilize the L_p-norm instead of the L_2-norm in the subsequent subsection.

2.2. Algorithm: LDA-L_p

It is well known that an algorithm based on the L_p-norm is less sensitive to the samples with large norms compared to the corresponding algorithm based on the L_2-norm.

Therefore, we define a new maximization problem for the design of an L_p-norm-based LDA. Consider the following L_p-norm maximization problem with the constraint $||w||_2 = 1$.

$$F_p(w) = \frac{\sum_{c=1}^{C} N_c |w^T(m_c - m)|^p}{\sum_{i=1}^{N} |w^T(x_i - m_{c_i})|^p}. \tag{4}$$

This can be solved by taking the gradient of $F_p(w)$ with respect to w. An important point to note here is that because of the absolute value operator in (4), the gradient of $F_p(w)$ is not well defined on some singular points. To avoid this technical difficulty, a sign function below is introduced.

$$sgn(a) = \begin{cases} 1 & \text{if } a > 0, \\ 0 & \text{if } a = 0, \\ -1 & \text{if } a < 0. \end{cases} \tag{5}$$

With the help of this sign function, (4) can be rewritten as follows:

$$F_p(w) = \frac{\sum_{c=1}^{C} N_c [\text{sgn}(w^T(m_c - m))w^T(m_c - m)]^p}{\sum_{i=1}^{N} [\text{sgn}(w^T(x_i - m_{c_i}))w^T(x_i - m_{c_i})]^p}. \tag{6}$$

Now, in order to get an optimal w which maximizes (6), we can take a gradient of $F_p(w)$ in (6) with respect to w as follows:

$$\nabla_w = \frac{dF_p(w)}{dw} = \frac{A \times B}{E} - \frac{C \times D}{E},$$
$$\text{where} \quad A = p\sum_{c=1}^{C} N_c \text{sgn}(w^T(m_c - m))|w^T(m_c - m)|^{p-1}(m_c - m),$$
$$B = \sum_{i=1}^{N} [\text{sgn}(w^T(x_i - m_{c_i}))w^T(x_i - m_{c_i})]^p,$$
$$C = \sum_{c=1}^{C} N_c [\text{sgn}(w^T(m_c - m))w^T(m_c - m)]^p,$$
$$D = p\sum_{i=1}^{N} \text{sgn}(w^T(x_i - m_{c_i}))|w^T(x_i - m_{c_i})|^{p-1}(x_i - m_{c_i}),$$
$$E = (\sum_{i=1}^{N} [\text{sgn}(w^T(x_i - m_{c_i}))w^T(x_i - m_{c_i})]^p)^2. \tag{7}$$

The above gradient is well defined when $w^T(m_c - m) \neq 0$ and $w^T(x_i - m_{c_i}) \neq 0$ for all x_i. Furthermore, it is also well defined if $p > 1$ on singular points where $w^T(m_c - m) = 0$ or $w^T(x_i - m_{c_i}) = 0$ for some x_i's. On the other hand, if $p = 1$ the term A or D in (7) is not well defined on the singular points because 0^0 is hard to define, and if $p < 1$, A or D diverges at the singular points. To avoid this problem, we add a singularity check step before computing the gradient.

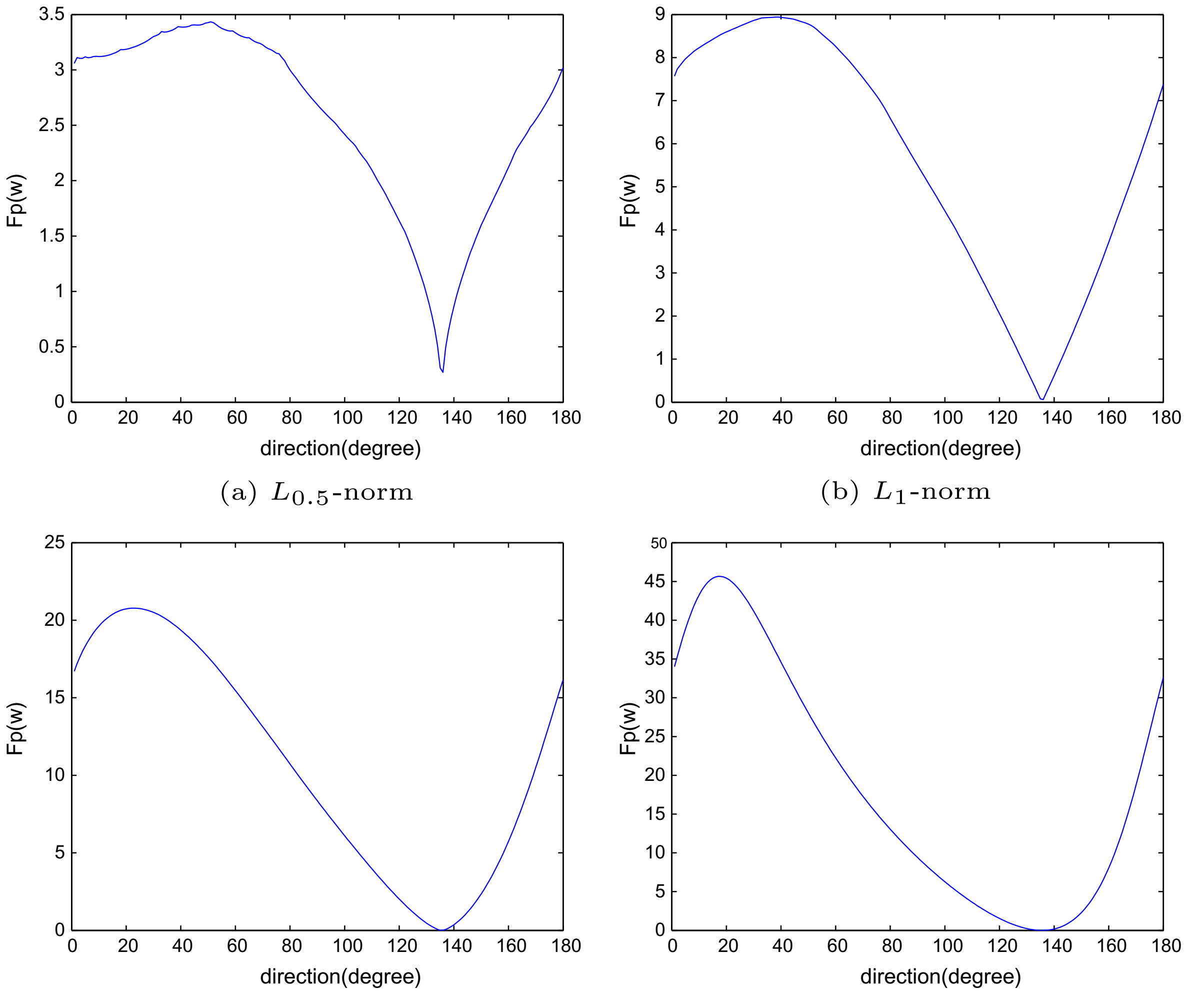
The optimal solution to this problem can be obtained using the steepest-gradient method as follows:

i. Initialization
   - $t \leftarrow 0$. Set $w(0)$ such that $||w(0)||_2 = 1$.

ii. Singularity check (applies only when $p \leqslant 1$)
   - If $w(t)^T(m_c - m) = 0$ or $w(t)^T(x_i - m_{c_i}) = 0, w(t) \leftarrow \frac{(w(t)+\delta)}{||w(t)+\delta||_2}$ where δ is a small random vector.

iii. Computation of ∇_w in (7)

iv. Gradient search
   - $w(t+1) \leftarrow w(t) + \alpha\nabla_w$ where α is a learning rate.



(a) $L_{0.5}$-norm

(b) L_1-norm

(c) $L_{1.5}$-norm

(d) L_2-norm

Fig. 1. $F_p(w)$ obtained by Eq. 4.

```
v. Normalization
    • $w(t) \leftarrow \frac{w(t)}{||w(t)||_2}$
vi. Convergence check
    • If $||w(t) - w(t-1)|| \geqslant \epsilon$, goto Step ii.
    • Else, $w^* \leftarrow w(t)$. Stop iteration.
```

First, we initialize the projection vector $w(0)$ such that $||w(0)||_2 = 1$. Then, if $p \leqslant 1$, we check whether w is a singular point or not. After the singularity check, we compute the gradient ∇_w using (7). Then, the steepest-gradient method is applied: $w(t+1) \leftarrow w(t) + \alpha\nabla_w$ where α is a learning rate. We normalize the obtained projection vector to make it a unit vector. Finally, the convergence check is performed on the obtained projection vector $w(t)$. In all the experiments in the next section, we set ϵ to 0.01, which, in this study, is a very small value.

In the above L_p-norm maximization problem, only one projection vector w can be obtained. When more than one vectors $\{w_i\}_{i=1}^m$ are needed, the proposed method can be easily extended to extract arbitrary number of features by applying the same procedure greedily to the remainder of the projected samples as follows:

- $w_0 = 0$, $\{x_i^0 = x_i\}_{i=1}^N$.
- For $j = 1$ to m,
    - For all $i \in \{1, \ldots, N\}$, $x_i^j = x_i^{j-1} - w_{j-1}(w_{j-1}^T x_i^{j-1})$.
    - In order to find w_j, apply the above LDA-Lp procedure to $\{(x_i^j, c_i)\}_{i=1}^N$.
- end

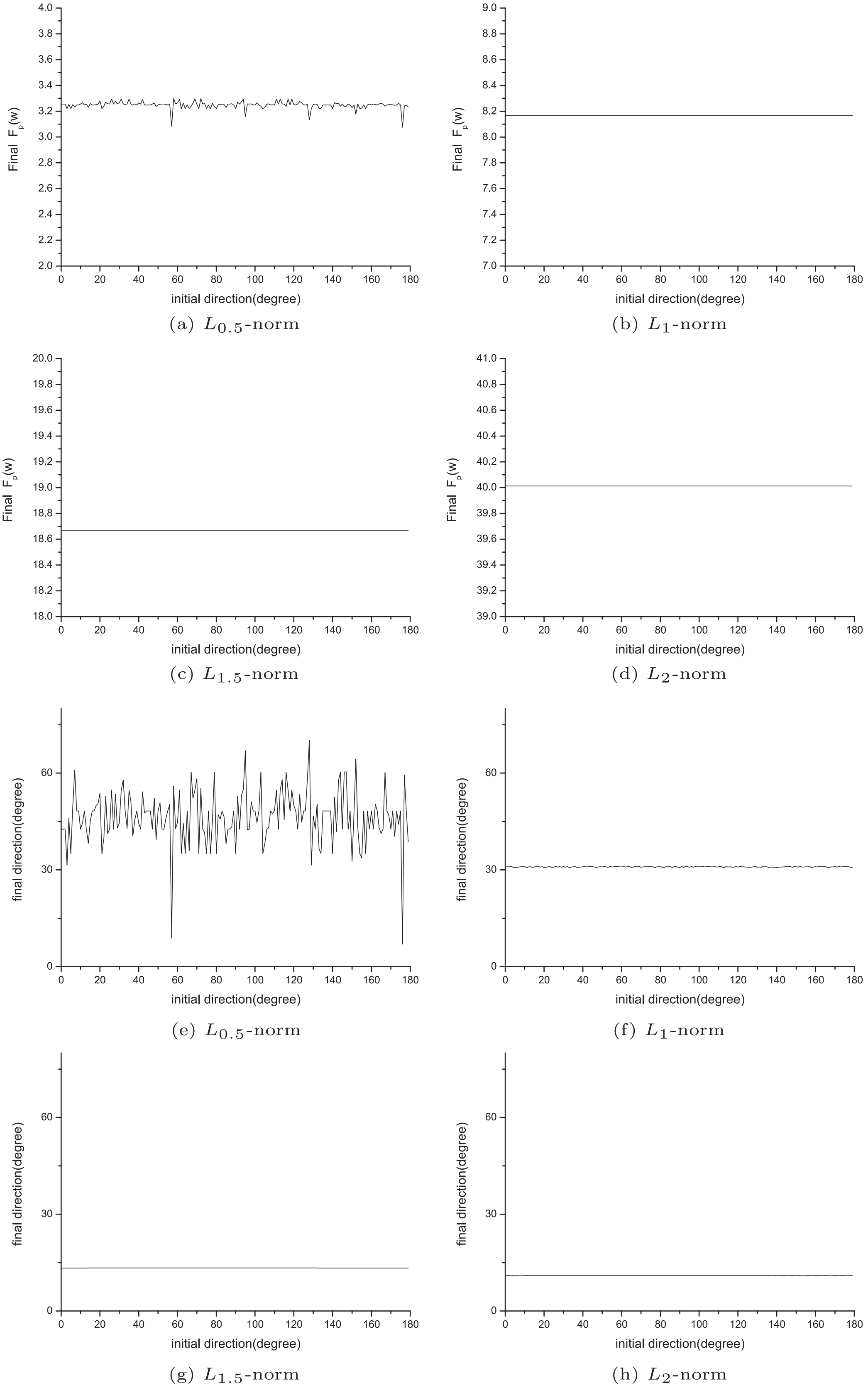
3. Experimental results and analysis

3.1. Experimental results

3.1.1. A toy problem

Consider the simple binary classification problem in a two-dimensional input space. Twenty data points of the first class are randomly generated from a Gaussian distribution with mean $(-5, -5)$ and standard deviation $(1, 1)$. Another 20 data points of the second class are generated in the same way with mean $(5, 5)$ and standard deviation $(1, 1)$. Finally, an outlier of the second class is positioned at $(5, 15)$.

Fig. 1 shows the values of objective function $F_p(w)$ for various values of p while varying the direction of w from 0° to 180°. We can estimate that the optimal direction of w is around 45° for above randomly generated dataset if the proposed method is robust to the presence of outliers. The maximum $F_p(w)$ were obtained in 51°, 39°, 23°, and 17° for $L_{0.5}$-norm, L_1-norm, $L_{1.5}$-norm and L_2-norm, respectively. As shown in Fig. 1, the $F_p(w)$ is locally



(a) $L_{0.5}$-norm

(b) L_1-norm

(c) $L_{1.5}$-norm

(d) L_2-norm

(e) $L_{0.5}$-norm

(f) L_1-norm

(g) $L_{1.5}$-norm

(h) L_2-norm

Fig. 2. Final $F_p(w)$ and final direction using steepest gradient method.

Table 1
Properties of UCI dataset.

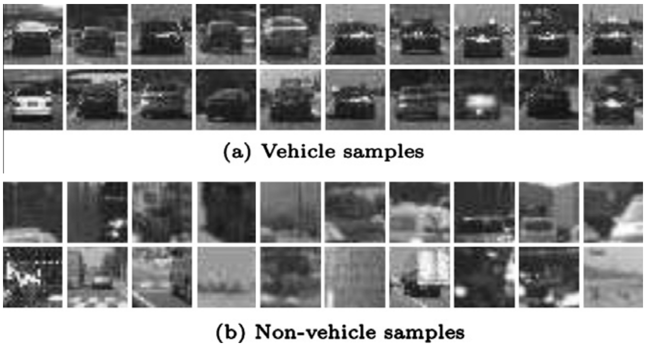
Dataset	No. of variables	No. of classes	No. of instances
Australian	14	2	690
Breast cancer	9	2	683
Heart disease	13	2	297
Bupa	6	2	345
Pima	8	2	768
Sonar	60	2	208
Balance	4	3	625
Waveform	21	3	4999
Iris	4	3	150
Yeast	8	10	1484

concave function for L_1-norm, $L_{1.5}$-norm and L_2-norm. Otherwise, the $F_p(w)$ is non-concave function for $L_{0.5}$-norm which contains many sharp peak. The final solution has much chance to stick to local maximum when we use steepest gradient method for $L_{0.5}$-norm. To avoid this problem, we used the projection obtained by the conventional LDA for the initial projection of LDA-L_p in the following experiments for UCI datasets and vehicle dataset. The final direction of w was more close to the optimal 45° when $L_{0.5}$- or $L_{1.0}$-norm was used instead of L_2-norm.

Fig. 2 shows the final $F_p(w)$ while varying initial projection vector $w(0)$ which is used in intialization step. The final direction indicates the angle of final projection of vector w of LDA-L_p. We select α to obtain a stable $F_p(w)$, while varying the initial direction in initialization step. We select α as 0.01 and the average of final directions is 46.5°, 30.9°, 13.3°, and 10.9°, respectively. LDA-L_1 and LDA-$L_{1.5}$ converge to a certain direction, as shown in Fig. 2. However, LDA-$L_{0.5}$ finds it difficult to converge because LDA-$L_{0.5}$ could converge to the local maximum, as shown in Fig. 1(a).

3.1.2. UCI datasets

We applied LDA-L_p to several datasets in the UCI machine learning repositories, and compared their performance with the conventional LDA. Table 1 shows the number of variables, classes, and instances of each dataset. In Table 2, the classification rates using tenfold cross-validation are shown with their standard deviations in parentheses. We extracted one feature for LDA and LDA-L_p to compare performance except for "Yeast" dataset. For "Yeast" dataset, one, three, and five features were extracted to compare the performances of LDA-L_p with various values of p. One-nearest neighbor classifier was used throughout this study. The initial projection of LDA-L_p was set to the projection obtained by LDA. We also set the maximum number of iterations to 100. The performances shown in Table 2 are the best classification performance among the results obtained by varying the learning rate α. We can see that LDA-$L_{1.5}$, LDA-L_1, and LDA-$L_{0.5}$ outperformed LDA by more than 3.6037%, 4.8515%, and 4.9472% on average. We obtained better classification performance using LDA-$L_{1.5}$ compared to LDA, but the performance of LDA-L_1 and LDA-$L_{0.5}$ was slightly better than LDA-$L_{1.5}$. Above all, the performance for the "Bupa" and "Sonar" dataset is improved by more than 10% in LDA-L_1 and LDA-$L_{0.5}$. For the same number of features, LDA-L_1 outperformed LDA-$L_{0.5}$ and LDA-$L_{1.5}$ for the datasets "Breast cancer", "Heart disease", "Bupa", "Sonar", and "Balance". LDA-$L_{0.5}$ outperformed the other methods for the "Pima", "Waveform", "Iris" and "Yeast" datasets. This shows that the effect of outliers can be reduced and the classification performance can be enhanced by using L_p norm instead of L_2 norm.


(a) Vehicle samples

(b) Non-vehicle samples

Fig. 3. Vehicle and non-vehicle samples for tenfold cross-validation.

Table 2
Classification rates, average time, and average number of iterations for UCI dataset.

	Classification rates (%)				Average time (s) and average number of iterations			
	LDA	LDA-$L_{1.5}$	LDA-L_1	LDA-$L_{0.5}$	LDA	LDA-$L_{1.5}$	LDA-L_1	LDA-$L_{0.5}$
Australian	76.6667	**81.8841**	81.8840	81.0145	0.5507	0.2625	1.6717	1.7036
	(3.5136)	(4.1134)	(5.4224)	(5.1783)	–	12.0	100.0	100.0
Breast cancer	96.0507	96.0465	**97.2251**	96.4898	0.5425	0.2063	1.6492	1.1893
	(2.9230)	(2.1941)	(2.5263)	(2.3001)	–	9.1	100.0	70.4
Heart disease	75.7586	80.1264	**80.8506**	79.4713	0.1240	0.0779	0.1268	0.2243
	(6.2433)	(6.7818)	(6.7557)	(9.4803)	–	7.8	15.2	28.8
Bupa	54.2185	63.1765	**65.8655**	65.2941	0.1575	0.3537	0.4132	0.8444
	(7.7966)	(5.3936)	(8.3002)	(9.1411)	–	38.9	47.9	100.0
Pima	65.8749	70.1487	71.0919	**72.3941**	0.6621	0.3705	1.1014	1.8621
	(5.6777)	(7.7649)	(4.9059)	(6.6174)	–	16.5	58.8	100.0
Sonar	65.3810	76.9286	**76.8333**	76.4762	0.0783	0.2707	0.5850	0.5948
	(9.1894)	(9.4524)	(7.1040)	(9.5560)	–	35.4	100.0	100.0
Balance	88.9683	89.2806	**90.0794**	86.5745	0.4493	0.0863	1.0717	1.5084
	(5.2415)	(11.9901)	(7.0860)	(7.9325)	–	3.0	71.1	99.0
Waveform	52.9111	55.0308	56.0905	**56.2717**	26.5034	4.3742	4.3259	15.0706
	(2.3254)	(1.7044)	(2.1430)	(1.5705)	–	11.4	11.3	100.0
Iris	96.6667	**97.3333**	96.6667	**97.3333**	0.0527	0.0206	0.0156	0.0163
	(4.7140)	(4.6614)	(4.7140)	(4.6614)	–	2.0	2.0	2.0
Yeast feature: 1	32.3535	34.3683	35.7722	**39.7610**	3.4150	3.7925	3.7118	3.8615
	(3.6347)	(2.5082)	(1.6705)	(2.2452)	–	15.3	29.3	29.3
Yeast feature: 3	44.7469	47.5789	49.4631	**51.8189**	2.0129	10.7295	10.6311	10.9018
	(3.6180)	(2.9475)	(3.0155)	(2.4699)	–	21.8	25.4	29.3
Yeast feature: 5	49.2667	50.2054	55.2603	**55.3310**	1.9804	17.0258	16.9107	17.6461
	(6.3659)	(2.6493)	(2.2815)	(3.0648)	–	21.2	27.8	29.3

The best classification rate is denoted in bold face.

Table 3
Classification rates, average time, and average number of iterations for vehicle data.

Classification rates (%)				Average time (s) and average number of iterations			
LDA	LDA-$L_{1.5}$	LDA-L_1	LDA-$L_{0.5}$	LDA	LDA-$L_{1.5}$	LDA-L_1	LDA-$L_{0.5}$
85.7000	**86.4750**	85.9000	86.0250	18.0792	2.2662	2.3312	10.7075
(1.5581)	(1.4599)	(1.4916)	(0.7945)	–	9.7	10.8	100.0

The best classification rate is denoted in bold face.

The right side of Table 2 shows the average time for each UCI dataset on a 2.66 GHz CPU with 2 GB RAM. For LDA-L_p, the average numbers of iterations for steepest-gradient method are also shown. For example, LDA-$L_{1.5}$ took 0.2625 (s) with 12 iterations for the "Australian" dataset. Note that the average time increases as the number of samples increases. For LDA-L_p, the average time is also in proportional to the number of iterations.

3.1.3. Vehicle dataset

The samples of data that were used for classifying vehicles and non-vehicles are shown in Fig. 3. In total, 4000 images were collected for tenfold cross-validation, which consisted of 2000 vehicle images and 2000 non-vehicle images. Each sample was scaled and aligned to a base resolution of 24 × 24 pixels. The experimental results for vehicle data are summarized in Table 3. We also extracted one feature and a one-nearest neighbor classifier was used. For this dataset, LDA-L_p is slightly better than LDA. Among LDA-L_p, the classification rates of LDA-$L_{1.5}$ was 86.4760%, which is better than those of the others. We also set the maximum number of iterations to 100 for vehicle data. The average time for LDA-$L_{0.5}$ was longer than that for LDA-$L_{1.5}$ and LDA-L_1 because of the large number of iterations. However, it took less time than LDA. The LDA-L_p method did not include the eigenvalue decomposition process, which takes up a large portion of the processing time. This is the main reason that LDA-L_p took less time than LDA.

Table 4
Effect of outliers for UCI datasets using LDA and LDA-L_p.

Dataset		Percentage of data		
		1%	2%	3%
Australian	LDA	0.1111	0.1619	0.2026
	LDA-$L_{1.5}$	0.0313	0.0521	0.0709
	LDA-L_1	0.0260	0.0468	0.0653
	LDA-$L_{0.5}$	0.0202	0.0367	0.0531
Breast cancer	LDA	0.0851	0.1467	0.2017
	LDA-$L_{1.5}$	0.0415	0.0780	0.1108
	LDA-L_1	0.0271	0.0519	0.0747
	LDA-$L_{0.5}$	0.0178	0.0347	0.0510
Heart disease	LDA	0.0512	0.0891	0.1211
	LDA-$L_{1.5}$	0.0392	0.0709	0.1011
	LDA-L_1	0.0250	0.0475	0.0695
	LDA-$L_{0.5}$	0.0168	0.0328	0.0486
Bupa	LDA	0.2623	0.3301	0.3967
	LDA-$L_{1.5}$	0.1766	0.2314	0.2877
	LDA-L_1	0.0912	0.1279	0.1725
	LDA-$L_{0.5}$	0.0375	0.0586	0.0838
Pima	LDA	0.1060	0.1697	0.2249
	LDA-$L_{1.5}$	0.0637	0.1105	0.1513
	LDA-L_1	0.0409	0.0737	0.1036
	LDA-$L_{0.5}$	0.0226	0.0429	0.0623
Sonar	LDA	0.1342	0.1908	0.2422
	LDA-$L_{1.5}$	0.0825	0.1273	0.1644
	LDA-L_1	0.0455	0.0745	0.1014
	LDA-$L_{0.5}$	0.0282	0.0461	0.0637
Balance	LDA	0.0717	0.1201	0.1642
	LDA-$L_{1.5}$	0.0508	0.0865	0.1196
	LDA-L_1	0.0345	0.0605	0.0850
	LDA-$L_{0.5}$	0.0212	0.0382	0.0547
Waveform	LDA	0.0496	0.0908	0.1276
	LDA-$L_{1.5}$	0.0357	0.0670	0.0961
	LDA-L_1	0.0253	0.0484	0.0704
	LDA-$L_{0.5}$	0.0168	0.0328	0.0484
Iris	LDA	0.0383	0.0571	0.0920
	LDA-$L_{1.5}$	0.0330	0.0490	0.0790
	LDA-L_1	0.0271	0.0399	0.0642
	LDA-$L_{0.5}$	0.0210	0.0311	0.0506
Yeast	LDA	0.0906	0.1497	0.1988
	LDA-$L_{1.5}$	0.0504	0.0899	0.1239
	LDA-L_1	0.0317	0.0571	0.0808
	LDA-$L_{0.5}$	0.0186	0.0354	0.0516

3.2. Analysis

In this part, the effect of outliers for UCI datasets and vehicle data is analyzed to show the reason for improvement in classification performance. A novel method for measuring the effect of outliers is proposed. To evaluate the effect of outliers in LDA-L_p, the following factor is defined.

$$\mathit{Effect\ on\ scatter\ matrix}_{x_i} = (|x_i - m|)^p + (|x_i - m_j|)^p, \tag{8}$$

$$\mathit{Outlier\ effect}^k_{x_i} = \frac{\sum_{i=1}^{ceil(\frac{k}{100}n)} \mathit{Effect\ on\ scatter\ matrix}_{x_i}}{\sum_{i=1}^{n} \mathit{Effect\ on\ scatter\ matrix}_{x_i}}. \tag{9}$$

The first and second terms of (8) denote the effect of each datum x_i on the between- and within-class scatter matrices, respectively, as shown in (1). The data that have a large *Effect on scatter matrix*$_{x_i}$ value can be regarded as outliers. We calculated (8) for each datum in several UCI datasets and reordered them in decreasing order. Then, we chose 1%, 2%, and 3% of the data that had a significant *Effect on scatter matrix*$^k_{x_i}$ value, as shown in (9). Here, *ceil* is the ceiling function. For example, $k = 1$ corresponds to the effect of 1% outlier.

We compared the effect of outliers in LDA and LDA-L_p in Table 4 using the *Outlier effect*$_{x_i}$ value. The *Outlier effect*$_{x_i}$ is small when we apply LDA-L_p. This means that the outlier effect of LDA-L_p is reduced compared to that of LDA. The outlier effects for the "Bupa" and "Sonar" datasets in LDA are 0.2623 and 0.1342, respectively, which are larger than for the other datasets. This means that we can reduce the outlier effect more effectively for these datasets if we use LDA-L_p instead of LDA. This is the reason for improved performance for the "Bupa" and "Sonar" datasets as compared to other datasets, as shown in Table 2. Further, we also focus on the outlier effect for "Iris" using LDA. The value is 0.0383, which is the smallest *Outlier effect*$_{x_i}$ value among the ten datasets. Thus, we can conclude that the "Iris" dataset does not contain many outliers. This is why the performance difference between LDA and LDA-L_p is smaller for "Iris" than for the other datasets.

Among LDA-L_p, we can reduce the outlier effect when we use smaller norm as shown in Table 4. Although the outlier effect is decreased consistently when we use small norm, the classification performance is slightly different for each LDA-L_p, as shown in Table 2. This depends on how well the data distribution is assumed by each norm. For example, we assume that the data distribution is Laplacian when we use LDA-L_1. If the data distribution is similar to a Laplacian distribution, the performance of LDA-L_1 is better than that of the other methods. Otherwise, if the data distribution is more similar to that of the $L_{0.5}$-norm, the performance of LDA-$L_{0.5}$ is better than that of other methods. We conclude that the performance of LDA-L_p is better than conventional LDA, and that the data distribution also affects the classification performance with varying norms.

4. Conclusions

This paper described a method for an LDA based on the L_p-norm optimization technique. Although conventional LDA based on the L_2-norm has been successfully solved many problems, the performance can be degraded with the presence of outliers. The reason for this phenomenon could be the effect of outliers, which is exacerbated by the use of a large norm. LDA-L_p is more robust to the presence of outliers than conventional LDA. By introducing a new method of measuring the effect of outliers, we analyzed the effect of outliers being reduced when we used LDA-L_p compared to LDA. The experimental results show that the proposed method achieves a high recognition rate for UCI datasets and vehicle data.

Acknowledgment

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (KRF-2010-0004908).

References

Fukunaga, K., 1990. Introduction to Statistical Pattern Recognition, second ed. Academic Press, Ch. 9–10, pp. 399–507.

Turk, M., Pentland, A., 1991. Face recognition using Eigenfaces. In: Proc. IEEE Conf. on Computer Vision and, Pattern Recognition, pp. 586–591.

Bell, A., Sejnowski, T., 1995. An information-maximization approach to blind separation and blind deconvolution. Neural Comput. 7 (6), 1129–1159.

Kwak, N., Choi, C.-H., 2003. Feature extraction based on ICA for binary classification problems. IEEE Trans. Knowl. Data Eng. 15 (6), 1374–1388.

Belhumeur, P., Hespanha, J., Kriegman, D., 1997. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. IEEE Trans. Pattern Anal. Machine Intell. 19 (7), 711–720.

Martinez, A., Kak, A., 2001. PCA versus LDA. IEEE Trans. Pattern Anal. Machine Intell. 23 (2), 228–233.

Yang, J., Zhang, L., Yang, J.-Y., Zhang, D., 2011. From classifiers to discriminators: A nearest neighbor rule induced discriminant analysis. Pattern Recognition 44 (7), 1387–1402.

Yang, J., Yang, J.-Y., 2003. Why can LDA be performed in PCA transformed space? Pattern Recognition 36 (2), 563–566.

Yang, J., Frangi, A.F., Yang, J.-Y., Zhang, D., Zhong, J., 2005. Kpca plus LDA: A complete Kernel Fisher discriminant framework for feature extraction and recognition. IEEE Trans. Pattern Anal. Machine Intell. 27 (2), 230–244.

Kwak, N., Lee, J.-W., 2010. Feature extraction based on subspace methods for regression problems. Neurocomputing 73 (10), 1740–1751.

Kwak, N., 2012. Kernel discriminant analysis for regression problems. Pattern Recognition 45 (5), 2019–2031.

Ke, Q., Kanade, T., 2005. Robust L1 norm factorization in the presence of outliers and missing data by alternative convex programming. In: Proc. IEEE Internat. Conf. on Computer Vision and Pattern Recognition, pp. 739–746.

Ding, C., Zhou, D., He, X., Zha, X., 2006. R1-PCA: Rotational invariant L1-norm principal component analysis for fobust subspace factorization. In: Proc. Internat. Conf. on Machine Learning, Pittsburgh, PA.

Kwak, N., 2008. Principal component analysis based on L1-norm maximization. IEEE Trans. Pattern Anal. Machine Intell. 30 (9), 1672–1680.

Li, X., Pang, Y., Yuan, Y., 2009. L1-norm based 2DPCA. IEEE Trans. Systems Man Cybern. Part B: Cybern. 40 (4), 1170–1175.

Kwak, N., Oh, J., 2009. Feature extraction for one-class classification problem: Enhancements to biased discriminant analysis. Pattern Recognition 42 (1), 17–26.

Li, X., Hu, W., Wang, H., Zhang, Z., 2010. Linear discriminant analysis using rotational invariant L1 norm. Neurocomputing 73, 2571–2579.

Pang, Y., Li, X., Yuan, Y., 2010. Robust tensor analysis with L1-norm. IEEE Trans. Circuits Systems Video Technol. 20 (2), 172–178.

Pang, Y., Li, X., Yuan, Y., Tao, D., Pan, J., 2009. Fast haar transform based feature extraction for face representation and recognition. IEEE Trans. Inf. Forensic Secur. 4 (3), 441–450.

Kwak, N., 2013. Principal component analysis by Lp-norm maximization. IEEE Trans. SMC-B.