Learning Latent Variable Models with Discriminant Regularization

Jing Peng[1(✉)] and Alex J. Aved[2]

[1] Department of Computer Science, Montclair State University, Montclair, NJ 07043, USA
pengj@montclair.edu
[2] Information Directorate, AFRL, Rome, NY 13441, USA
alexander.aved@us.af.mil

Abstract. In many machine learning applications, data are often described by a large number of features or attributes. However, too many features can result in overfitting. This is often the case when the number of examples is smaller than the number of features. The problem can be mitigated by learning latent variable models where the data can be described by a fewer number of latent dimensions. There are many techniques for learning latent variable models in the literature. Most of these techniques can be grouped into two classes: techniques that are informative, represented by principal component analysis (PCA), and techniques that are discriminant, represented by linear discriminant analysis (LDA). Each class of the techniques has its advantages. In this work, we introduce a technique for learning latent variable models with discriminant regularization that combines the characteristics of both classes. Empirical evaluation using a variety of data sets is presented to verify the performance of the proposed technique.

Keywords: Classification · Dimensionality reduction · Latent variable models

1 Introduction

In many machine learning applications, a large number of features or attributes is often used to describe examples. However, too many features can cause overfitting, resulting in poor generalization performance. This is the case when there are more features than examples. Poor generalization performance can be attributed to the curse of dimensionality [6], which implies that to avoid overfitting, the number of examples must increase exponentially with the number of features. For example, it is shown that there are $O(2^q)$ unknowns that must be estimated to learn a binary distribution in a space with q correlated features [8].

The above problem can be mitigated by learning latent variable models where the data can be described by a fewer number of latent dimensions. In fact, learning latent variable models has been one of the key building blocks in machine learning, which in turn will benefit many practical applications [4,20,24,27,40].

One of the goals of learning latent variable models is to compute the intrinsic dimensionality of the input space represented by high dimensional input examples. There are both linear and non-linear techniques for learning latent variable models in the literature. In this work, we are concerned with linear techniques for their simplicity. Many linear techniques can be extended to non-linear cases.

A. P. Rocha et al. (Eds.): ICAART 2020, LNAI 12613, pp. 378–398, 2021.
https://doi.org/10.1007/978-3-030-71158-0_18

Many techniques for learning latent variable models have been developed over the years [3,5,15,21,22,28,29,43]. There are two major categories of techniques for learning latent variable models. The first category of techniques is represented by principal component analysis (PCA), where the objective is to minimize information loss. The second category of techniques is represented by linear discriminant analysis (LDA), where the objective is to maximize class separation. Each has its advantages. For example, latent positions computed by PCA do not rely on class label information, while latent positions computed by LDA do. And as such, one expects LDA to be able to perform better than PCA in classification applications. However, when there are insufficient training examples per class in face recognition problems, empirical evidence shows that PCA can perform better [26].

In this paper, we propose a technique for learning latent variable models that combine some of the characteristics of both PCA and LDA. The proposed technique draws upon ideas from probabilistic latent variable models [25,33,38], where the negative log prior can be viewed as regularization (or penalty) in a non-Bayesian context. The technique minimizes a minimum information loss objective with discriminant regularization. As a result, the technique represents a trade-off between PCA and LDA, resulting in better generalization performance in many applications. Empirical evaluation using a number of data sets is presented to verify the proposed technique for learning latent variable models.

Note that an earlier version of the current work appeared in [30]. While the technical idea presented here is along the lines of the one described in [30], the technical discussion is carried out in a general probabilistic context, rather than Gaussian processes. Thus the technical presentation is more refined in the present paper. Furthermore, we have included more examples in the empirical evaluation section in the present paper. These diverse examples, ranging from biometric (such as iris and fingerprint) and image classification problems to hyperspectral image analysis, have provided strong empirical evidence to support the technique proposed in the present paper.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 introduces a probabilistic framework for learning latent variable models to motivate the introduction of out proposal. Section 4 introduces a linear technique for learning latent variable models with discriminant regularization. The technique can be interpreted as regularized PCA, where discriminant analysis is the regularizer. It can also be viewed as regularized discriminant analysis, where the regularizer is PCA. Section 5 provides the empirical evaluation of the proposed technique against several competing techniques. Finally, Sect. 6 summarizes our contributions and points out future research directions.

2 Related Work

There are many techniques in the literature that aim to exploit the inherent low dimensional nature of the data [12,17,34,41]. Linear techniques for learning latent variable models can be broadly categorized into two classes, represented by PCA and LDA, respectively. These techniques can learn the intrinsic geometry of the input space, along with its global Euclidean structure.

Note that PCA is closely related to auto-encoders in neural networks. In its very basic form (one hidden layer with linear outputs), the q hidden units span the same latent space as the first q components found by PCA [7,18]. The components of PCA are orthogonal, while the weight vectors of the basic auto-encoder may not. A deep auto-encoder can learn a non-linear subspace, which can be desirable in many applications. The challenge is that it can be very difficult to optimize deep auto-encoders using backpropagation. Techniques have been proposed to address this challenge in the literature [18]. In addition, studies have been done comparing deep auto-encoders with kernel PCA, which can also produce a non-linear subspace.

Locality preserving projection (LPP) is a linear technique for learning the locality structure of input space [19]. The technique constructs an adjacency matrix from input examples that describes the local neighborhood information of the input space. The optimal projection can then be computed that preserves the neighborhood information in the latent space. It has been noted that the basis functions, resulting from LPP, may not be orthogonal [9]. Thus, the data reconstruction can be a challenge in many applications.

Orthogonal locality preserving projection (OLPP) is a linear technique for learning latent variable models [9]. It is proposed to address some of the problems associated with LPP [19]. As in LPP, OLPP first constructs an adjacency matrix that contains locality information. OLPP then computes a latent subspace, where its basis functions are orthogonal. These orthogonal basis functions preserve the metric structure of latent space. OLPP has been shown to perform better than LPP in several applications [9].

Gaussian Process (GP) latent variable models are probabilistic techniques for learning latent variable models from high dimensional input examples [16,23,25,38]. GP latent variable models have been shown to be successful in a number of problems such as image reconstruction and facial expression recognition [1,10,14,35].

GP latent variable models are generative techniques [25,33,38]. Similar to PCA, these techniques are unsupervised [25]. GP latent variable models are useful in many applications, such as data visualization and regression analysis. However, GP latent variable models may not be suitable for classification applications. One possible solution is to introduce priors over latent variables to bias their positions in latent space [38]. One potential problem associated with GP latent variable models require an inference process for a test example in order to estimate its position in latent space. This separate inference process can complicate GP latent variable model computation, due to increased computational complexity.

Techniques for combining PCA and LDA for dimensionality reduction have been introduced in the literature [42,44]. The objective function is formulated as a linear combination of the objectives of PCA and LDA in these techniques. In this work, we aim to learn latent variable models with discriminant regularization. This formulation is closely related to the maximum a posteriori estimation in a Gaussian framework, where the negative log prior can be viewed as regularization (or penalty) [33,38].

3 Latent Variable Models

In this work, we use $\mathbf{x}$ to represent the input, and use y to represent the output or target. We let

$$D = \{(\mathbf{x}_i, y)|i = 1, \cdots, n\}^t \tag{1}$$

be a set of n centered examples, where $\mathbf{x}_i \in \Re^q$. The vector inputs are aggregated in the $n \times q$ matrix X

$$X = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]^t, \tag{2}$$

where t represents transpose. We denote the corresponding latent variables as $\mathbf{h} \in \Re^d$. The vector latent variables are aggregated in the $n \times d$ matrix H

$$H = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n]^t.$$

We note that $d \ll q$.

The input $\mathbf{x}$ and its latent variable $\mathbf{h}$ can be described in the following way

$$\mathbf{x} = F\mathbf{h} + \varepsilon, \tag{3}$$

where F is a $q \times d$ matrix of weights or parameters of the model, and ε represents the error term. We assume that this error term follows a Gaussian distribution with zero mean and uniform variance

$$p(\varepsilon) = N(0, \beta^{-1}\mathbf{I}),$$

where β is a constant. We further assume that the error term is independent and identically distributed (i.i.d.). The model (3) along with the Gaussian error term gives us the following likelihood, conditional probability density of the input examples

$$p(\mathbf{x}|\mathbf{h}, F, \beta) = N(F\mathbf{h}, \beta^{-1}\mathbf{I}).$$

The above shows that X (2) follows a matrix variate normal distribution

$$\begin{aligned} p(X|H, F, \beta) &= \prod_{i=1}^{n} p(\mathbf{x}_i|\mathbf{h}_i, F, \beta) \\ &= \frac{\beta^{\frac{qn}{2}}}{(2\pi)^{\frac{qn}{2}}} \exp(-\frac{1}{2} tr(\beta(X^t - FH^t)(X^t - FH^t)^t)). \end{aligned} \tag{4}$$

Here the underlying assumption is that input examples $\mathbf{x}_i$ are independent and identically distributed. If we integrate out the latent variables H, we obtain the probabilistic PCA solution for F [36].

An alternative approach is to integrate out F and optimize with respect to H to obtain a solution for the latent variables. This dual approach has been studied in [25,38]. In this approach, $\mathbf{f}_i$, the ith row of F, is assumed to follow a Gaussian distribution with zero mean and uniform variance

$$p(\mathbf{f}_i) = N(0, \alpha^{-1}\mathbf{I})$$

where α is a constant. It follows that $p(F)$ is also Gaussian and given by

$$p(F) = \prod_{i=1}^{q} p(\mathbf{f}_i) = \frac{1}{C_q} \exp(-\frac{1}{2} tr(\alpha F^t F))$$
$$= \frac{\alpha^{\frac{dq}{2}}}{(2\pi)^{\frac{dq}{2}}} \exp(-\frac{1}{2} tr(\alpha F^t F)). \quad (5)$$

where C_q is a normalization factor. Combining (4) and (5) and integrating out F, we obtain the marginalized likelihood of X

$$p(X|H, \beta) = \int p(X|H, F, \beta) p(F) dF$$
$$\propto \frac{1}{|K|^{q/2}} \exp(-\frac{1}{2} tr(\Sigma^{-1} X X^t)), \quad (6)$$

where

$$\Sigma = (\alpha^{-1} H H^t + \beta^{-1} \mathbf{I}),$$

and $|\Sigma|$ denotes the determinant of matrix Σ.

The above (6) shows that the likelihood of the input examples X is Gaussian, given the latent variables H. The log likelihood of X is

$$L = -\frac{qn}{2} \ln(2\pi) - \frac{q}{2} |\Sigma| - \frac{1}{2} tr(\Sigma^{-1} X X^t).$$

It is shown that optimization of the log likelihood respect to latent variables H results in a solution that is equivalent to the PCA solution [25,36].

We note that it is possible to further constrain the latent variables H by introducing priors over H. For example, if we introduce an uninformed prior on H, we obtain the following log prior

$$\ln p(H) = -\frac{1}{2} \sum_{i=1}^{n} \mathbf{h}_i^t \mathbf{h}_i.$$

This simple prior constrains the latent variables to be closer to the origin [38]. For classification problems, class labels can be incorporated into priors [14,35]. For example, priors can be based on linear discriminant analysis [15]. If Σ_w and Σ_b represent the sample between- and within-class matrices in the latent space, respectively, the LDA based criterion $J(H) = tr(\Sigma_w^{-1} \Sigma_b)$ can be implemented. We use tr to represent the matrix trace operator. This leads us to the following prior [38]

$$p(H) = C \exp(-J^{-1}).$$

One of the problems with the latent variable models discussed above is that to estimate the latent position for an unseen test example, a separate inference process is required. And as such, additional uncertainties can be introduced in the estimate with increased computational complexity.

4 Latent Variable Models with Discriminant Regularizers

In the previous section, we discussed a general technique for learning latent variable models. In this section, we introduce an algorithm for learning latent variable models for classification problems without a separate process and increased computational complexity.

As discussed in the previous section, the optimization of the likelihood (6) with respect to latent variables H gives rise to the probabilistic PCA solution to the latent variables H. Additional constraints can be placed on the latent variables H by introducing priors. The introduction of priors over latent variables $p(H)$ results in the log posterior (terms that the posterior depends on)

$$L = \frac{q}{2} \ln |\Sigma| + \frac{1}{2} tr(\Sigma^{-1} X X^t) - \ln p(H). \tag{7}$$

In a non-Bayesian setting, the negative log prior $-\ln p(H)$ can often be regarded as a penalty term [33,38]. This is also related to ridge regression [28] and weight decay [39]. If the prior $p(H)$ is discriminant, optimization of (7) produces a solution to the latent variables Z that is both informative, as in PCA, and discriminant, as in LDA. While the idea is appealing for many applications, an inference must be made for each test example, which potentially introduces uncertainty and additional computational complexity [25,38].

We address this problem by describing a simple algorithm for learning latent variable models with discriminant regularization. The algorithm achieves the desired representation balance shown in (7), without separate inference for test examples.

We begin with PCA. Recall that PCA computes linear projection P by optimizing

$$J_{PCA}(P) = tr(P^t X X^t P), \tag{8}$$

where XX^t represents the sample covariance matrix, assuming that the examples are centered (2). The resulting linear projection P has the following property that

$$\sum_{i}^{n} \|\mathbf{x}_i - PP^t \mathbf{x}_i\|^2$$

is minimum. That is, the latent representations of the examples X estimated from PCA are optimal in terms of information loss.

We note that PCA is entirely unsupervised. For classification problems, we want to leverage class label information to compute latent variable models. To do so, we explore the idea behind the joint distribution of the latent variables (7), where the prior distribution over the latent variables imposes conditions on their positions in the resulting latent space. As discussed in the previous section, the negative log prior can be simply interpreted as a penalty term, or regularization. Therefore, we can introduce a discriminant regularization or penalty term in (8)

$$J_{PCA_r}(P) = tr(P^t X X^t P) + \lambda r(P), \tag{9}$$

where $r(\cdot)$ denotes a regularization term, and λ is a regularization parameter. In this work, we examine two discriminant regularization schemes: Locality Preserving regularizer and Linear Discriminant regularizer.

4.1 Locality Preserving Regularizer

The locality preserving projection (LPP) is a technique introduced in [19]. The technique first constructs a graph of the input examples (2). LPP then computes a linear projection from the graph that preserves the locality information.

Suppose that A is a $n \times n$ matrix, where the entry A_{ij} (ith row and jth column) is computed according to

$$A_{ij} = \begin{cases} \exp(-\eta \|\mathbf{x}_i - \mathbf{x}_j\|^2) & i \neq j \text{ and } l(\mathbf{x}_i) = l(\mathbf{x}_j) \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

In the above, $\mathbf{x}_i$ denotes the ith training example, $l(\mathbf{x})$ is the label of $\mathbf{x}$, and η is a parameter. That is, A represents the adjacency matrix of the input examples. Let $\mathbf{p} \in \Re^q$ such that $h_i = \mathbf{p}^t \mathbf{x}_i$. Also, let

$$J_{LPP} = \sum_{i,j} (h_i - h_j)^2 A_{ij}. \tag{11}$$

As can be seen, when examples $\mathbf{x}_i$ and $\mathbf{x}_j$ that are in the same class are projected far apart by $\mathbf{p}$, they contribute to J_{LPP}. On the other hand, J_{LPP} completely ignores examples that are in different classes. The locality preserving technique computes a linear projection $\mathbf{p}$ by minimizing (11).

We rewrite the above objective (11) by simple algebraic manipulation

$$\begin{aligned} J_{LPP} &= \frac{1}{2} \sum_{i,j} (h_i - h_j)^2 A_{ij} \\ &= \frac{1}{2} \sum_{i,j} (\mathbf{p}^t \mathbf{x}_i - \mathbf{p}^t \mathbf{x}_j)^2 A_{ij} \\ &= \mathbf{p}^t X^t L X \mathbf{p}, \end{aligned} \tag{12}$$

where $L = \Lambda - A$ is the graph Laplacian, and Λ is a diagonal matrix with diagonal entries $\lambda_{ii} = \sum_j A_{ij}$. Often Λ_{ii} can be regarded as the volume of h_i. Thus, LPP aims to solve the following constraint optimization problem

$$\begin{aligned} &\min_{\mathbf{p}} \mathbf{p}^t X^t L X \mathbf{p} \\ &\text{s.t. } \mathbf{p}^t X^t \Lambda X \mathbf{p} = 1 \end{aligned} \tag{13}$$

Therefore, the optimal $\mathbf{p}$ can be obtained by solving the following generalized eigenvalue problem

$$X^t L X \mathbf{p} = \lambda X^t \Lambda X \mathbf{p}, \tag{14}$$

where λ denotes the eigenvalue corresponding to $\mathbf{p}$. In many applications, LPP has been demonstrated to be effective [9,19].

The above discussion naturally suggests that locality preserving can be exploited as a regularizer to the PCA objective (8). Therefore, the proposed PCA with locality preserving regularization becomes

$$J(\mathbf{p}) = tr(\mathbf{p}^t X X^t \mathbf{p}) + \lambda tr(\mathbf{p}^t (X^t L X)^{-1} (X^t \Lambda X) \mathbf{p}). \tag{15}$$

It follows that the optimal projection $\mathbf{p}$ can be computed by maximizing

$$J_{PCA-LPP} = tr(XX^t + \lambda((X^tLX)^{-1}(X^t\Lambda X)). \tag{16}$$

The resulting linear projection algorithm is denoted as *P-Lpp*. It is interesting to note that we can interpret (16) as regularized PCA, where locality preserving is the regularizer. We can also interpret (16) as regularized locality preserving projection, where PCA is the regularizer.

4.2 Linear Discriminant Regularizer

In this section, we consider an alternate regularizer-linear discriminant analysis (LDA) [15]. Recall that LDA finds a linear projection $\mathbf{p}$ by optimizing

$$J(\mathbf{p}) = tr((\mathbf{p}^t\Sigma_w\mathbf{p})^{-1}(\mathbf{p}^t\Sigma_b\mathbf{p})), \tag{17}$$

where

$$\Sigma_w = \sum_{c=1}^{C}\sum_{i=1,\mathbf{x}_i\in c}^{n_c}(\mathbf{x}_i - \mathbf{m}_c)(\mathbf{x}_i - \mathbf{m}_c)^t \tag{18}$$

and

$$\Sigma_b = \sum_{c=1}^{C}(\mathbf{m}_c - \mathbf{m})(\mathbf{m}_c - \mathbf{m})^t \tag{19}$$

are the within and between class matrices, $\mathbf{m}$ is the overall mean of the input examples, and $\mathbf{m}_c$ denotes the mean of class c. It turns out that maximizing (17) is equivalent to maximizing

$$J_{LDA} = tr(\Sigma_w^{-1}\Sigma_b).$$

This allows us to propose PCA (8) with linear discriminant regularization

$$J_{IP-LDA} = tr(XX^t + \lambda\Sigma_w^{-1}\Sigma_b). \tag{20}$$

We call the resulting linear projection algorithm *P-Lda*. Note that similar to P-Lpp (16), we can interpret (20) as regularized PCA, where the regularizer is LDA. We can also view (20) as regularized LDA. In this case, PCA is the regularizer.

5 Empirical Evaluation

In this section, we provide empirical evaluation using a number of problems that validates performance of the proposed technique. We also include several competing techniques for comparison.

5.1 Competing Methods

The following competing methods are evaluated in our empirical evaluation.

1. P-Lpp–Regularized PCA, where locality preserving is the regularizer (Eq. 16).
2. P-Lda–Regularized PCA, where LDA is the regularizer (Eq. 20).
3. PCA–Laten variable model that maximizes (Eq. 8)

$$J(\mathbf{p}) = tr(\mathbf{p}^t X X^t \mathbf{p}).$$

4. LDA–Latent variable model that maximizes

$$J_{LDA} = tr(\Sigma_w^{-1} \Sigma_b),$$

   where Σ_w and Σ_b are given by (18) and (19), respectively.
5. OLPP–Orthogonal Locality Preserving Projection (OLPP) proposed in [9].

We state that OLPP is developed to address some of the problems associated with LPP (13). It has been shown that the eigenvectors resulting from optimizing (14) may not be orthogonal. OLPP addresses this problem by projecting the input examples onto the PCA subspace, from which it computes the solution to (14) so that orthogonality can be preserved. OLPP has been shown to perform better than LPP in a number of problems [9]. Therefore, we compare the proposed techniques P-Lpp and P-Lda against OLPP in the experiments.

5.2 Data Sets

Several data sets are used to demonstrate the generalization performance by each of the competing techniques. They are described below.

Fig. 1. Sample AR-face images, adapted from [30].

1. **AR-Face Image Data (ARFace).** This data set comes from the AR-face database [26]. A detailed description of the AR-face image data set is provided in [26]. For this data set, we randomly selected 50 different subjects (25 males and 25 females) from this AR-face database. All the face images used here were normalized to 85×60 pixel arrays of intensity values. Figure 1 shows some sample images from the AR-face data set. In the AR-face experiment we follow the setup of the Small Training Data set experiment detailed in [26]. When there are insufficient training examples per class, PCA has been shown to provide better performance than LDA [26]. Our goal here is to examine how well the proposed techniques P-Lpp and P-Lda perform against PCA in such a setting.

For this example, we selected the first seven images from each subject. This gives us a total of 350 face images. To emphasize the problems often associated with insufficient training examples, two instances from each subject were chosen as training examples, and the remaining five images were used as test examples. This gives rise to 21 different ways to split face images into training and testing. The results averaged over 21 runs are reported. Note that, as in [26], we apply PCA to transform the original face images of 85×60 pixels into vectors of 350 dimensions. These vectors are input to all the competing techniques examined here.

2. **MNIST Data (MNIST).** This dataset consists of handwritten digit images from the US National Institute of Standards and Technology (NIST)[1]. Each digit image is a array of 28 by 28 pixels of intensity values. Therefore, each example is a vector of 784 intensity values. 100 examples were randomly selected from each digit class in this experiment. Thus, The MNIST dataset has a total of 1000 examples. Sample MNIST digit images are shown in Fig. 2.

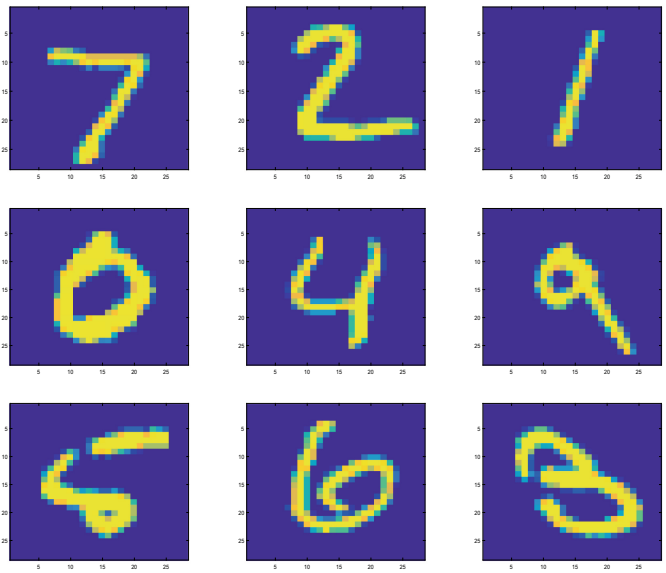

Fig. 2. Sample MNIST digit images.

3. **Cat and Dog data (CatDog).** The CatDog dataset consists of two hundred cat and dog face images. Each face image is an array of 64×64 pixels. These images have been normalized by aligning the eyes. Figure 3 shows some sample cat and dog face images.
4. **Multilingual Text (MText).** The MText dataset is a text data set consisting of multiple languages [2]. The data set is a collection of Reuters' RCV1 and RCV2. The text documents have six categories: (1) *Economics*, (2) *Equity Markets*,

[1] yann.lecun.com/exdb/mnist/.

Fig. 3. Sample images of the cat and dog face data, adapted from [30].

(3) *Government Social*, (4) *Corporate/Industrial*, (5) *Performance*, and (6) *Government Finance*. Each English document in the dataset has been translated into French, German, Italian and Spanish, using PORTAGE [37]. For this experiment, only English texts are used. Also, each text is described by a bag of words model, resulting in 21531 dimensions. 100 examples from each category were randomly selected for this experiment. Thus, the resulting dataset has a total of 600 examples described by 21531 features.

5. **Iris Data (Iris).** The Iris dataset is part of a large multimodal biometric data collection created by researchers at West Virginia University (WVU) [11]. The collection is available upon request. The Iris dataset consists of iris images from people of different age, gender, and ethnicity. A detailed description can be found in [11]. The Iris dataset is challenging because the quality of these images are low due to blur, occlusion, and noise. Figure 4 shows some sample iris images in the data set.
   Each iris image is segmented into a 25×240 template [32]. It has been shown that Gabor features are suitable for representing iris images [13]. Therefore, each image template is convolved with a log-Gabor filter at a single scale to obtain a vector of 6000 features. For this experiment, a pair of subjects was randomly selected. One subject has 27 examples, and the other subject has 36 examples, resulting in a total of 63 examples described by 6000 features.

Fig. 4. Sample Iris images, adapted from [30].

6. **Fingerprint Data (Finger).** The Fingerprint dataset consists of fingerprint images of people of different age, gender, and ethnicity. Similar to the Iris dataset, it is part of a large multimodal biometric data collection created by researchers at WVU [11]. The fingerprint dataset is difficult because many examples are of low quality, as a result of blur, occlusion, and noise. Figure 5 shows sample fingerprint images used in this experiment.

Fig. 5. Sample fingerprint images, adapted from [30].

This dataset has a total of 124 fingerimages from randomly chosen pair of subjects. One of the subjects has 61 instances, and the other has 63. Ridge and bifurcation features are computed to represent the fingerprint images, using code that is publically available (sites.google.com/site/athisnarayanan/). As a result, each fingerprint image is represented by a feature vector of 7241 dimensions.

7. **Feret Face Data (FeretFace).** The FERET face image dataset has 400 facial images. There are 50 subjects of both males and females, selected randomly from the Feret face database [31]. There are 8 examples per subject. The images vary in terms of facial expressions and illumination. Each image has 150×130 pixels, resulting in an image vector of 19500 intensity values. Figure 6 shows the sample images used in this experiment.

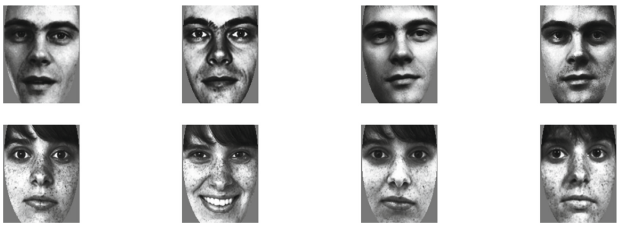

Fig. 6. Normalized Feret sample images, adapted from [30].

8. **Cooke City Hyperspectral Data (CookeCity).** The Cooke City data set consists of a hyperspectral image of Cooke City, Montana, a library of target spectral reflectances, and target (class) location information in the image (or regions of interest). Figure 7 shows a false color representation of the Cooke city scene. The self-test set of the Cooke City data set contains the ground truth information. Therefore, we only used the self-test set in this experiment. The Cooke city hyperspectral image contains 7 target classes, and the number of instances in each class varies from 9 to 34. Each instance is represented by 126 bands, resulting in a vector of 126 dimensions. The detailed information about the Cooke city hyperspectral image is provided in dirsapps.cis.rit.edu/blindtest/.
9. **Pavia University Hyperspectral Data(Pavia).** The Pavia University hyperspectral image data was captured by the ROSIS sensor during a flight over the Pavia University in northern Italy. There are 103 spectral bands, and the number of pixels is 610 by 610. The image has a resolution of 1.3 m. There are 9 target classes: Asphalt, Meadows, Gravel, Trees, Painted metal sheets, Bare soil, Bitumen, Self-Blocking

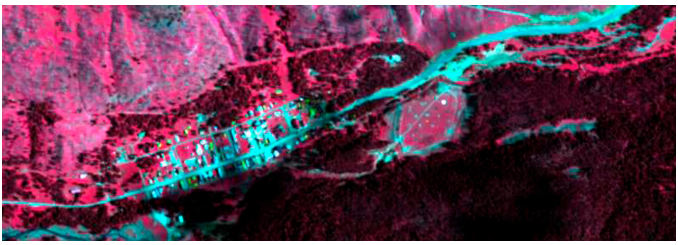

Fig. 7. False color representation of Cooke City.

bricks, and Shadows. Also, the data set has 42,776 examples, and the number of examples per target class varies from 947 to 18,649. Figure 8 shows a sample band image of the scene and the corresponding ground truth map.

10. **Indian Pines Hyperspectral Data (Pines).** This data set contains a hyperspectral image, covering the Indian Pines test site in north-western Indiana. The image is acquired by the AVIRIS sensor, and consists of 145×145 pixels. There are 224 spectral reflectance bands. In this experiment, there are only 200 bands after removing bands that cover the water absorption region: [104–108], [150–163], and 220. The data set has 10,249 examples and 16 target classes, which are not all mutually exclusive: Alfalfa, Corn-notill, Corn-mintill, Corn, Grass-pasture, Grass-trees, Grass-pasture-mowed, Hay-windrowed, Oats, Soybean-notill, Soybean-mintill, Soybean-clean, Wheat, Woods, Buildings-Grass-Trees-Drives, and Stone-Steel-Towers. And the number of examples per target class varies from 20 to 2,455. Figure 9 shows the image and the corresponding ground truth map.

Fig. 8. Sample band representation of Pavia University scene in Northern Italy and the corresponding ground truth map.

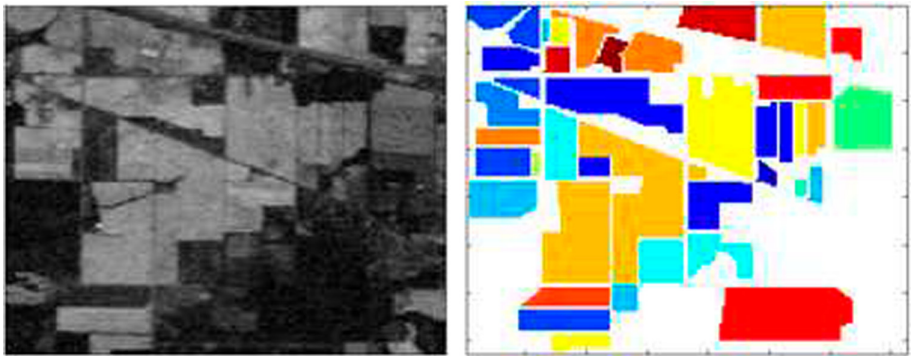

Fig. 9. Sample band representation of the Indian Pines test site and the ground truth map.

Table 1. Average error rates obtained by the competing methods on the 10 diverse data sets.

	PCA	P-Lpp	P-Lda	LDA	OLPP
ARFace	0.307	0.245	0.245	0.394	0.314
MNIST	0.143	0.152	0.143	0.402	0.148
CatDog	0.492	0.216	0.210	0.457	0.286
MText	0.405	0.217	0.232	0.365	0.305
Iris	0.480	0.133	0.133	0.141	0.136
Finger	0.466	0.378	0.387	0.444	0.467
FeretFace	0.088	0.042	0.042	0.092	0.098
CookeCity	0.155	0.045	0.045	0.121	0.164
Pavia	0.224	0.248	0.180	0.220	0.214
Pines	0.408	0.395	0.436	0.405	0.400
Ave	0.317	0.207	0.205	0.304	0.253

5.3 Empirical Results

In this section, we report the empirical performance by each method. We have normalized all the training data to have zero mean and unit variance along each feature. We have also normalized all the test data using the corresponding training mean and variance. Since we want to highlight the techniques for learning latent variable models, we prefer simple methods for classification in latent space. Thus, we used the one nearest neighbor rule for classification in the resulting latent space. Note that all the procedural parameters such as the regularization constant λ and the kernel parameter η in the graph Laplacian (13) were selected through cross validation. Table 1 shows the 10-fold crossed validated error rates achieved by the five competing methods on the 10 data sets described above.

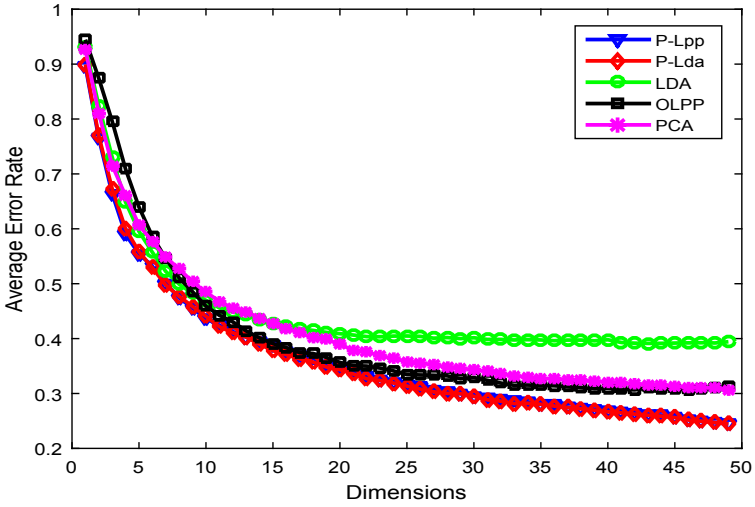

Fig. 10. Average error rates obtained by P-Lpp, P-Lda, LDA, OLPP, and PCA as a function of subspace dimensionality on the AR face data set (adapted from [30]).

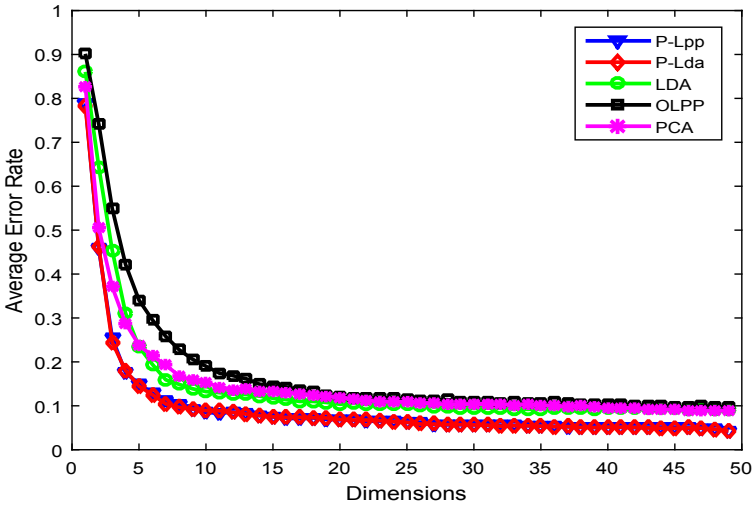

Fig. 11. Average error rates obtained by P-Lpp, P-Lda, LDA, OLPP, and PCA as a function of subspace dimensionality on the Feret face data.

The table shows that on average both P-Lpp and P-Lda performed well, compared to the competing methods for learning latent variable models. The table also shows that P-Lpp and P-Lda performed similarly on these datasets. The results show that performing classification in a latent space that is both information preserving and discriminant gives rise to better generalization than in either latent space alone.

Figure 10 shows the average error rates achieved by the competing techniques over 21 runs on the AR face dataset as a function of increasing dimensions. The plots show clearly that on average both P-Lpp and P-Lda performed better than PCA across the 49 subspaces, and consistently performed better than the remaining techniques. Also, the average error rates over the 21 runs in a subspace with 49 dimensions are shown in the first row in Table 1.

Figures 11, 12, 13, and 14 plot the 10-fold error rates computed by the competing methods on the Feret face, the Cooke City, Pavia University, and Indian Pines datasets with increasing subspace dimensionality, respectively. For the Feret face data, both P-Lpp and P-Lda consistently outperformed the competing methods across the 49 subspaces. P-Lda achieved consistently better performance in the hyperspectral datasets.

We can observe that overall the performance of P-Lpp correlates well with that of OLPP on the Indian Pines (Fig. 14). Likewise, the performance of P-Lda in general correlates well with that of LDA on the Cooke City and Pavia University data (Figs. 12 and 13). It is interesting to note that PCA performed competitively in these experiments. By placing constraints on latent variable positions, the proposed technique for learning latent variable models with discriminant regularization seems to provide better generalization.

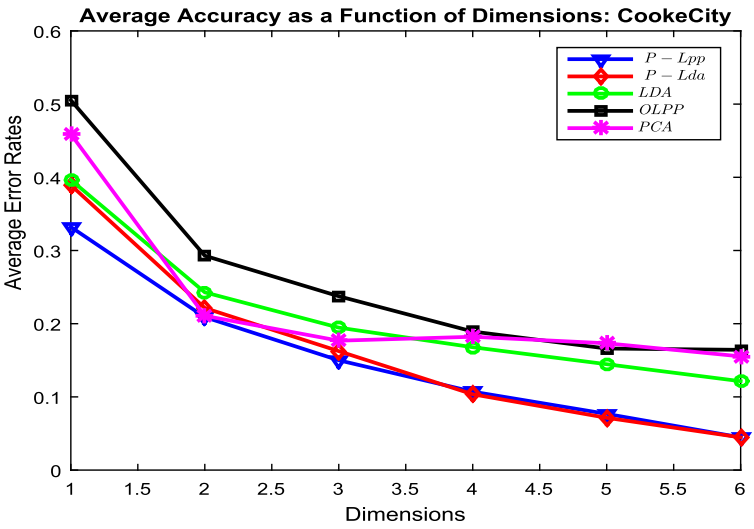

Fig. 12. Average error rates obtained by P-Lpp, P-Lda, LDA, OLPP, and PCA as a function of subspace dimensionality on the Cooke City hyperspectral data.

5.4 Performance Robustness

The empirical results show that P-Lpp and P-Lda achieved overall the best performance across the 10 diverse data sets, followed by OLPP. We ask the question of performance robustness. It is the question of how well a particular method m performs in problems

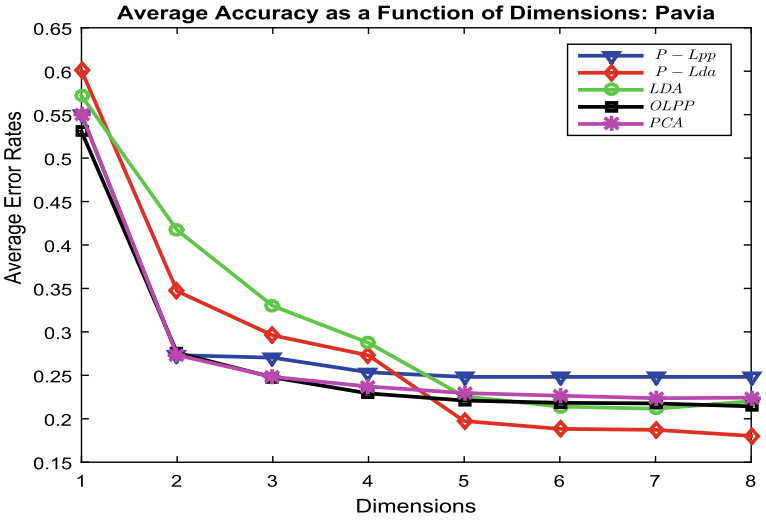

Fig. 13. Average error rates obtained by P-Lpp, P-Lda, LDA, OLPP, and PCA as a function of subspace dimensionality on the Pavia University hyperspectral data.

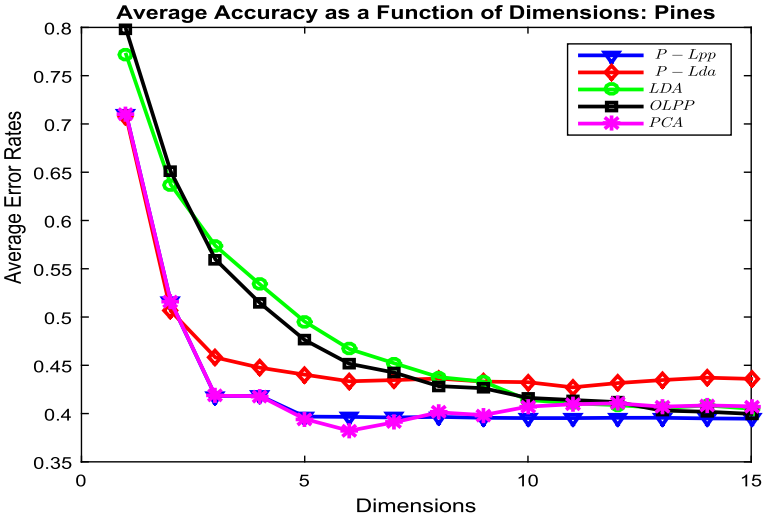

Fig. 14. Average error rates obtained by P-Lpp, P-Lda, LDA, OLPP, and PCA as a function of subspace dimensionality on the Indian Pines hyperspectral data.

that are most favorable to other methods. One possible measure of performance robustness can be described by computing the ratio r_m of the error rate err_m by method m to the smallest error rate over all methods in a particular problem. That is, we compute the following

$$r_m = err_m / \min_{1 \le k \le 5} err_k. \tag{21}$$

According to this measure, the best method m^* for the problem has $r_{m^*} = 1$. And for all other methods other than the best, their r_m values should be greater than one. Clearly, the larger the value of r_m, the worse the performance of method m is in relation to the best method for the problem. Therefore, the distribution of the r_m values for each method m over all the problems provides a good measurement for its performance robustness.

The distributions of the r_m values for each method over the 10 data sets are shown in Fig. 15. The boxed areas show the lower and upper quartiles of the distribution. They are separated by the median (red horizontal line). The entire range of values for each distribution is represented by the outer vertical lines.

Figure 15 shows that the performance of P-Lda was most robust over the 10 data sets. Tts error rates were the best (median = 1.0) in 7/10 of the data sets. It was no worse than 10.0% higher than the best error rate in the worst case. The next is P-Lpp. Its error rates were the best (median = 1.0) in 7/10 of the data sets. In 2/10 of the data sets, its error rates were no worse than 6% higher than the best error rate. Also, P-Lpp was no worse than 38.0% in the worst case.

Figure 15 also shows that PCA's worst error rate was 260.9% higher than the best error rate, which was worse than LDA's worst error rate (181.1% higher than the best error rate).

On the other hand, the median r_{PCA} value for PCA was 1.56. In contrast, the median r_{LDA} value for LDA was 1.645, which was worse than that of PCA. PCA achieved the best error rate on one of the data sets. LDA did not. Overall, PCA and LDA performed similarly, in terms of the distributions of the values of r_{PCA} and r_{LDA}.

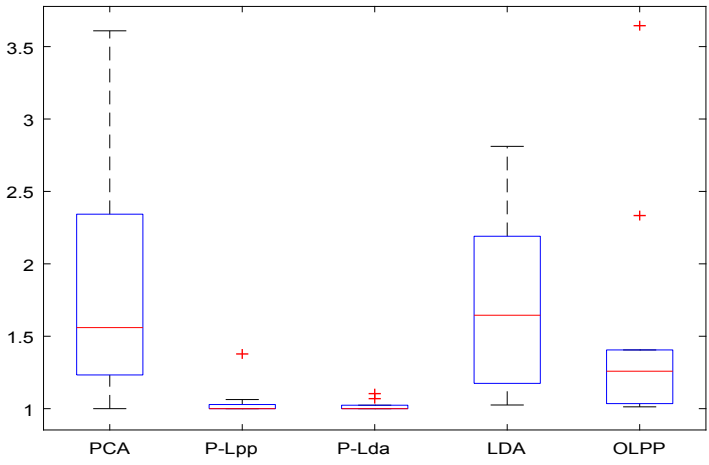

Fig. 15. Distributions of r_m (robustness) values for PCA, P-Lpp, P-Lda, LDA, and OLPP over the 10 data sets. (Color figure online)

6 Conclusion

A technique for learning latent variable models with discriminant regularization has been presented. By incorporating discriminant regularization into PCA, the proposed technique is capable of learning latent variable models that achieve better generalization. An empirical evaluation of the proposed technique against competing techniques using a variety of examples has been provided. The empirical results show that the proposed technique is competitive in the examples that have been experimented with.

References

1. Abolhasanzadeh, B.: Gaussian process latent variable model for dimensionality reduction in intrusion detection. In: 2015 23rd Iranian Conference on Electrical Engineering, pp. 674–678 (2015)
2. Amini, M., Usunier, N., Goutte, C.: Learning from multiple partially observed views-an application to multilingual text categorization. In: Advances in Neural Information Processing Systems, pp. 28–36 (2009)
3. Aved, A., Blasch, E., Peng, J.: Regularized difference criterion for computing discriminants for dimensionality reduction. IEEE Trans. Aerosp. Electron. Syst. **53**(5), 2372–2384 (2017)
4. Banerjee, B., Peng, J.: Efficient learning of multi-step best response. In: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2005, pp. 60–66. ACM, New York (2005)
5. Belhumeur, V., Hespanha, J., Kriegman, D.: Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 711–720 (1997)
6. Bellman, R.E.: Adaptive Control Processes: A Guided Tour. Princeton University Press, Princeton (1961)
7. Bourlard, H., Kamp, Y.: Auto-association by multilayer perceptrons and singular value decomposition. Biol. Cybern. **59**(4–5), 291–294 (1988)
8. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth and Brooks, Monterey (1984)
9. Cai, D., He, X., Han, J., Zhang, H.: Orthogonal Laplacianfaces for face recognition. IEEE Trans. Image Process. **15**(11), 3608–3614 (2006)
10. Cai, L., Huang, L., Liu, C.: Age estimation based on improved discriminative Gaussian process latent variable model. Multimedia Tools Appl. **75**(19), 11977–11994 (2016)
11. Crihalmeanu, S., Ross, A., Schukers, S., Hornak, L.: A protocol for multibiometric data acquisition, storage and dissemination. Technical report, WVU, Lane Department of Computer Science and Electrical Engineering (2007)
12. Darnell, G., Georgiev, S., Mukherjee, S., Engelhardt, B.: Adaptive randomized dimension reduction on massive data. J. Mach. Learn. Res. **18**(140), 1–30 (2017)
13. Daugman, J.: How iris recognition works. IEEE Trans. Circ. Syst. Video Technol. **14**(21), 21–30 (2004)
14. Eleftheriadis, S., Rudovic, O., Pantic, M.: Discriminative shared Gaussian processes for multiview and view-invariant facial expression recognition. IEEE Trans. Image Process. **24**(1), 189–204 (2015)
15. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press, San Diego (1990)

16. Gao, X., Wang, X., Tao, D., Li, X.: Supervised Gaussian process latent variable model for dimensionality reduction. IEEE Trans. Syst. Man Cybern. Part B Cybern. **41**(2), 425–434 (2011)
17. Harandi, M., Salzmann, M., Hartley, R.: Joint dimensionality reduction and metric learning: a geometric take. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 1404–1413. International Convention Centre, Sydney. PMLR (2017)
18. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
19. He, X., Niyogi, P.: Locality preserving projections. In: Proceedings of the 16th International Conference on Neural Information Processing Systems, NIPS 2003, pp. 153–160. MIT Press (2003)
20. Heisterkamp, D., Peng, J., Dai, H.: Feature relevance learning with query shifting for content-based image retrieval. In: Proceedings 15th International Conference on Pattern Recognition, ICPR 2000, vol. 4, pp. 250–253 (2000)
21. Howland, P., Park, H.: Generalizing discriminant analysis using the generalized singular Valuke decomposition. IEEE Trans. Pattern Anal. Mach. Intell. **26**(8), 995–1006 (2004)
22. Huo, X., et al.: Optimal reduced-rank quadratic classiers using the Fukunaga-Koontz transform, with applications to automated target recognition. In: Proceedings of SPIE Conference (2003)
23. Jiang, X., Gao, J., Wang, T., Zheng, L.: Supervised latent linear Gaussian process latent variable model for dimensionality reduction. IEEE Trans. Syst. Man Cybern. Part B Cybern. **42**(6), 1620–1632 (2012)
24. Kaluarachchi, A.C., et al.: Incorporating terminology evolution for query translation in text retrieval with association rules. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 1789–1792 (2010)
25. Lawrence, N.: Probabilistic non-linear principal component analysis with Gaussian process latent variable models. J. Mach. Learn. Res. **6**, 1783–1816 (2005)
26. Martinez, A.M., Kak, A.: PCA versus LDA. IEEE Trans. Pattern Anal. Mach. Intell. **23**(2), 228–233 (2001)
27. Peng, J.: Efficient memory-based dynamic programming. In: Proceedings of the 12th International Conference on Machine Learning, pp. 438–446 (1995)
28. Peng, J., Zhang, P., Riedel, N.: Discriminant learning analysis. IEEE Trans. Syst. Man Cybern. Part B Cybern. **38**(6), 1614–1625 (2008)
29. Peng, J., Seetharaman, G., Fan, W., Varde, A.: Exploiting fisher and Fukunaga-Koontz transforms in chernoff dimensionality reduction. ACM Trans. Knowl. Disc. Data **7**(2), 8:1–8:25 (2013)
30. Peng, J., Aved, J.A.: Information preserving discriminant projections. In: Proceedings of the 12th International Conference on Agents and Artificial Intelligence (ICAART), pp. 162–171 (2020)
31. Phillips, P.J., et al.: The FERET database and evaluation procedure for face recognition algorithms. Image Vis. Comput. **16**(6), 295–306 (1998)
32. Pundlik, S., Woodard, D., Birchfield, S.: Non-ideal iris segmentation using graph cuts. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1–6 (2008)
33. Rasmussen, C., Williams, C.: Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press, Cambridge (2005)
34. Sarveniazi, A.: An actual survey of dimensionality reduction. Am. J. Comput. Math. **4**(2), 55–72 (2014)
35. Song, G., Wang, S., Huang, Q., Tian, Q.: Similarity Gaussian process latent variable model for multi-modal data analysis. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 4050–4058 (2015)

36. Tipping, M., Bishop, C.: Probabilistic principal component analysis. J. Roy. Stat. Soc. Ser. B **61**(3), 611–622 (1999)
37. Ueffing, N., Simard, M., Larkin, S., Johnson, J.: NRC's PORTAGE system for WMT 2007. In: ACL-2007 Second Workshop on SMT, pp. 185–188 (2007)
38. Urtasun, R., Darrell, T.: Discriminative Gaussian process latent variable model for classification. In: Proceedings of the 24th International Conference on Machine Learning, ICML 2007, pp. 927–934. ACM, New York (2007)
39. Williams, R.J., Peng, J.: Function optimization using connectionist reinforcement learning algorithms. Connect. Sci. **3**(3), 241–268 (1991)
40. Xie, S., Fan, W., Peng, J., Verscheure, O., Ren, J.: Latent space domain transfer between high dimensional overlapping distributions. In: Proceedings of the 18th International Conference on World Wide Web, pp. 91–100 (2009)
41. Xie, P., et al.: Learning latent space models with angular constraints. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 3799–3810. International Convention Centre, Sydney. PMLR (2017)
42. Yu, J., Tian, Q., Rui, T., Huang, T.S.: Integrating discriminant and descriptive information for dimension reduction and classification. IEEE Trans. Circ. Syst. Video Technol. **17**(3), 372–377 (2007)
43. Zhang, P., Peng, J., Domeniconi, C.: Kernel pooled local subspaces for classification. IEEE Trans. Syst. Man Cybern. Part B Cybern. **35**(3), 489–502 (2005)
44. Zhao, N., Mio, W., Liu, X.: A hybrid PCA-LDA model for dimension reduction. In: The 2011 International Joint Conference on Neural Networks, pp. 2184–2190 (2011)