Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Latent Discriminant Analysis with Representative Feature Discovery

Gang Chen
Department of Computer Science and Engineer
SUNY at Buffalo, Buffalo, NY 14260
gangchen@buffalo.edu

Abstract

Linear Discriminant Analysis (LDA) is a well-known method for dimension reduction and classification with focus on discriminative feature selection. However, how to discover discriminative as well as representative features in LDA model has not been explored. In this paper, we propose a latent Fisher discriminant model with representative feature discovery in an semi-supervised manner. Specifically, our model leverages advantages of both discriminative and generative models by generalizing LDA with data-driven prior over the latent variables. Thus, our method combines multi-class, latent variables and dimension reduction in an unified Bayesian framework. We test our method on MUSK and Corel datasets and yield competitive results compared to baselines. We also demonstrate its capacity on the challenging TRECVID MED11 dataset for semantic keyframe extraction and conduct a human-factors ranking-based experimental evaluation, which clearly demonstrates our proposed method consistently extracts more semantically meaningful keyframes than challenging baselines.

Introduction

Linear Discriminant Analysis (LDA) (Fisher 1936) is a powerful tool for dimensionality reduction and classification that projects high dimensional data into a low-dimensional space where the data achieves maximum class separability (Duda, Hart, and Stork 2000; Fukunaga 1990; Wu, Wipf, and Yun 2015). The basic idea in classical LDA, known as Fisher Linear Discriminant Analysis (FDA) is to obtain the projection matrix by minimizing the within-class distance and maximizing the between-class distance simultaneously to yield the maximum class discrimination. It has been proved analytically that the optimal transformation is readily computed by solving a generalized eigenvalue problem (Fukunaga 1990). In order to deal with multi-class scenarios (Rao 1948; Duda, Hart, and Stork 2000), LDA can be easily extended from binary case to multi-class problems, which finds a subspace with $d-1$ dimensions, where d is the number of classes in the training dataset. Because of its effectiveness and computational efficiency, it has been applied successfully in many applications, such as face recognition (Belhumeur, Hepanha, and Kriegman 1997) and microarray gene expression data analysis. Moreover, LDA was shown to compare favorably with other supervised dimensionality reduction methods through extensive experiments (Sugiyama et al. 2010).

However, as a supervised approach, LDA expects manually annotated training sets, e.g., instance/label pairs. As we known, it is labor-intensive and time-consuming to label each instance, which is surprisingly prohibitive especially for large scale data. Correspondently, it is reasonable to extend supervised LDA into a semi-supervised method, and many approaches (Joachims 1999; Cai, He, and Han 2007; Zhang and Yeung 2008; Sugiyama et al. 2010) have been proposed. Unfortunately, most of these methods still need instance/label pairs, i.e. training a classifier with a few labeled instances. In practice, many real applications require bag-level labels (Andrews, Tsochantaridis, and Hofmann 2002), such as molecule activity (Maron and Lozano-Prez 1998), image classification (Maron and Ratan 1998) and event detection (Perera et al. 2011). Recently, MI-SVM or latent SVM (Andrews, Tsochantaridis, and Hofmann 2002; Felzenszwalb et al. 2010) has been widely used for classification tasks, such as object detection. In a sense, MI-SVM can learn discriminative features effectively under maximum margin framework. However, MI-SVM does not consider data distribution while inferring latent variables. On the contrary, LDA leverages data distribution by computing between-class and within-class covariances to learn a discriminant projection. Thus it is possible to incorporate the data driven prior into LDA to discover both representative and discriminative features.

In this paper, we propose a Latent Fisher Discriminant Analysis model (or LFDA in short) with representative feature discovery. On the one hand, we hope our model can handle semi-supervised learning problems. On the other hand, we can generalize discriminative FDA to select representative features as well. More specifically, our method unifies the discriminative nature of FDA with a data driven Gaussian mixture prior over the training data under the Bayesian framework. By combining these two terms into one model, we infer latent variables and learn projection matrix in an alternative manner until convergence. To further leverage the compactness of each component with Gaussian mixture model, we assume that all instances in each component have the same label. Thus, our model relaxes the instance

level inference into component level inference by maximizing a joint likelihood, which can capture representative features effectively. To sum up, our method combines multi-class, latent variables and dimension reduction in an unified bayesian framework. We demonstrate the advantages of our model on MUSK and Corel datasets for classification problems, and on TRECVID MED11 dataset for semantic keyframe extraction on five video events (Perera et al. 2011).

Related Work

LDA has been a popular method for dimension reduction and classification. It searches a projection matrix that simultaneously maximizes the between-class dissimilarity and minimizes the within-class dissimilarity to increase class separability, typically for classification applications. And many methods (Belhumeur, Hepanha, and Kriegman 1997; Chen et al. 2000; Baudat and Anouar 2000; Merchante, Grandvalet, and Govaert 2012) have been proposed to either leverage or extend LDA because of its effectiveness and computational efficiency. Belhumeur et al proposed PCA+LDA (Belhumeur, Hepanha, and Kriegman 1997) for face recognition. Recently, sparsity induced LDA is also proposed (Merchante, Grandvalet, and Govaert 2012; Wu, Wipf, and Yun 2015).

However, many real-world applications only provide labels on bag-level, such as object detection (Felzenszwalb et al. 2010) and image classification (Maron and Ratan 1998). In the last decades, semi-supervised methods have been proposed to utilize unlabeled data to aid classification or regression tasks under situations with limited labeled data, such as Transductive SVM (TSVM) (Vapnik 1998; Joachims 1999) and Co-Training (Blum and Mitchell 1998). One of the main trend is to extend LDA to handle semi-supervised problems (Cai, He, and Han 2007; Zhang and Yeung 2008; Sugiyama et al. 2010) in a transductive manner, which attempts to utilize unlabeled data to aid classification or regression tasks under situations with limited labeled data. For example, Semi-supervised Discriminant Analysis (Cai, He, and Han 2007) was proposed, which made use of both labeled and unlabeled samples. Sugiyama et. al. proposed a semi-supervised dimensionality reduction method (Sugiyama et al. 2010), which can preserve the global structure of unlabeled samples in addition to separating labeled samples in different classes from each other. Chen and Corso proposed a semi-supervised approach (Chen and Corso 2012) to learn discriminative codebooks and classify instances with nearest neighbor voting. Recenly, latent SVM or MI-SVM has attracted great attention for semi-supervised problems, such as multiple instance learning and object detection (Zhang and Yeung 2008; Felzenszwalb et al. 2010). It basically infers the latent variables by maximizing a posterior probability and shows great improvement on object detection (Felzenszwalb et al. 2010).

Another trend prefers to extent LDA into an unsupervised scenarios. For example, Ding and Li proposed to combine LDA and K-means clustering into the LDA-Km algorithm (Ding and Li 2007) for adaptive dimension reduction. In this algorithm, K-means clustering was used to generate class labels and LDA is utilized to perform subspace selection. However, directly casting LDA as a semi-supervised method to handle bag-level labels is still a challenge for multi-class problems.

Latent Fisher discriminant analysis

Let $\boldsymbol{\mathcal{X}} = \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}\}$ represent n bags, with the corresponding labels $\boldsymbol{\mathcal{L}} = \{l_1, l_2, ..., l_n\}$ as the training data. For each bag $\mathbf{x_i} \in \boldsymbol{\mathcal{X}}$, it can have one or multiple instances (Andrews, Tsochantaridis, and Hofmann 2002), and its label l_i is categorical and assumes values in a finite set, e.g. $\{1, 2, ..., C\}$. Let $\mathbf{x_i} \in \mathbb{R}^{d \times n_i}$, which means it contains n_i instances (or frames), denoted as $\mathbf{x_i} = \{x_i^1, x_i^2, ..., x_i^{n_i}\}$ with its j^{th} instance $x_i^j \in \mathbb{R}^d$ (however, its label is not given). Given the data $\boldsymbol{\mathcal{X}}$ and its corresponding instance level labels $Z(\boldsymbol{\mathcal{X}})$, LDA searches for a discriminative feature transformation $f : \boldsymbol{\mathcal{X}} \rightarrow \boldsymbol{\mathcal{Y}}$ to maximize the ratio of between-class variance to the within-class variance, where $y \in \mathbb{R}^{d'}$ and $d' \leq d$. In general, d' is decided by C, namely $d' = C - 1$. However, we do not know the instance-level labels $Z(\boldsymbol{\mathcal{X}})$ for the data $\boldsymbol{\mathcal{X}}$. In our case, only the bag-level labels $\boldsymbol{\mathcal{L}}$ are available. Thus, we think for any instance $\forall x \in \boldsymbol{\mathcal{X}}$, it has a corresponding label $z(x)$, which can be inferred from the training pairs $(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}})$.

Latent Fisher discriminant analysis model generalizes LDA with latent variables. Suppose the projection matrix is $\boldsymbol{\mathcal{P}}$, and $y = f(x) = \boldsymbol{\mathcal{P}}x$, then our latent Fisher LDA proposes to minimize the following ratio:

$$\begin{aligned}(\boldsymbol{\mathcal{P}}^*) &= \operatorname*{argmin}_{\boldsymbol{\mathcal{P}}, Z} J(\boldsymbol{\mathcal{P}}, Z) \\ &= \operatorname*{argmin}_{\boldsymbol{\mathcal{P}}, Z} \text{trace}\left(\frac{\boldsymbol{\mathcal{P}}^T \Sigma_w(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, Z)\boldsymbol{\mathcal{P}}}{\boldsymbol{\mathcal{P}}^T \Sigma_b(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, Z)\boldsymbol{\mathcal{P}}} + \beta \boldsymbol{\mathcal{P}}^T \boldsymbol{\mathcal{P}}\right)\end{aligned} \tag{1}$$

where Z are the latent variables for the data $\boldsymbol{\mathcal{X}}$, and β is a weighing parameter for regularization term. The variable $z \in Z(\boldsymbol{\mathcal{X}})$ defines the possible latent values for a sample $x \in \boldsymbol{\mathcal{X}}$. In our case, $z \in \{1, 2, ..., C\}$. $\Sigma_b(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, Z)$ is between-class scatter matrix and $\Sigma_w(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, Z)$ is within-class scatter matrix, defined respectively as follows:

$$\Sigma_w(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, Z) = \Sigma_{k=1}^C \Sigma_{\{x \in \boldsymbol{\mathcal{X}} | \delta(z(x)=k)\}} (x - \bar{x}_k)(x - \bar{x}_k)^T \tag{2}$$

$$\Sigma_b(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, Z) = \Sigma_{k=1}^C m_k (\bar{x}_k - \bar{x})(\bar{x}_k - \bar{x})^T \tag{3}$$

where $\delta(z(x) = k)$ is the indicator function, m_k is the number of training samples for each class k, $\bar{x}_k = \frac{\sum_{\{x \in \boldsymbol{\mathcal{X}} | \delta(z(x)=k)\}} x}{m_k}$ is the mean of the k-th class and $\bar{x}$ is the total mean vector given by $\bar{x} = \frac{1}{\sum_{k=1}^C m_k} \Sigma_{k=1}^C m_k \bar{x}_k$. Note that LDA (Fisher 1936) is dependent on a categorical variable z (i.e. the class label) for each instance x to compute Σ_b and Σ_w. If z is given for any x, we can use LDA to find the discriminative transform $\boldsymbol{\mathcal{P}}$, using eigenvectors of $\Sigma_w^{-1}\Sigma_b$ to capture both compactness of each class and separations between classes.

However, in our case, given the training data $\boldsymbol{\mathcal{X}}$, we only know bag-level labels $\boldsymbol{\mathcal{L}}$, not the instance-level labels Z. To minimize $J(\boldsymbol{\mathcal{P}}, Z)$, we need to infer $z(x)$ for any given x. This problem is a chicken and egg problem, and can be

solved by alternating algorithms, such as EM (Dempster, Laird, and Rubin 1977). In other words, solve $\boldsymbol{\mathcal{P}}$ in Eq. (1) with fixed z, and vice versa in an alternating strategy.

Updating z

Suppose we have found the projection matrix $\boldsymbol{\mathcal{P}}$. Then, we can project $\boldsymbol{\mathcal{X}}$ into its corresponding subspace $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{X}}$, where $\boldsymbol{\mathcal{Y}} = \{\mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_n}\}$ is the one to one mapping of $\boldsymbol{\mathcal{X}} = \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}\}$. Instead of inferring latent variables at instance-level as latent SVM, we propose to infer latent variable z at clustering-level in the projected space $\boldsymbol{\mathcal{Y}}$. That means all elements in the same cluster have the same label. Such assumption is reasonable because elements in the same cluster are close to each other. On the other hand, cluster-level inference can speed up the learning process. We extend mixture discriminative analysis model in (Hastie and Tibshirani 1996) by incorporating latent variables over all instances for an given class. Thus, we assume each class i is a K components of Gaussians,

$$p(t|\lambda_i) = \sum_{j=1}^{K} \pi_i^j g(y|\mu_i^j, \Sigma_i^j) \tag{4}$$

where t, $y \in \mathbb{R}^{d'}$ (i.e. vector or feature); $\pi_i = \{\pi_i^j\}_{j=1}^K$ are the mixture weights, and $g(y|\mu_i^j, \Sigma_i^j)$ is the j-th component Gaussian with μ_i^j as mean and Σ_i^j as covariance. $\lambda_i = \{\pi_i, \mu_i, \Sigma_i\}$ are the parameters for class i with $\mu_i = \{\mu_i^j\}_{j=1}^K$ and $\Sigma_i = \{\Sigma_i^j\}_{j=1}^K$ which we need to estimate.

Given the training data $(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{L}})$, we can compute the data-driven prior using Gaussian mixture model (GMM) in Eq. (4). In our case, for each class $i \in \{1, 2, \ldots, C\}$, we collect all instances whose $\mathbf{y}_i$ belongs to this class i, then we use mixtures of Gaussians in Eq. (4) for clustering analysis. As a result, we can estimate λ_i and get its K components $S_i = \{S_i^1, S_i^2, ..., S_i^K\}$ with EM algorithm, as well as its data-driven prior distribution $\pi_i = \{\pi_i^1, \pi_i^2, ..., \pi_i^K\}$. With a little abuse of symbols, we may μ_i^j to indicate the j-th cluster of i-th class, and each cluster S_i^j has n_i^j elements, which satisfies $\sum_{j=1}^K n_i^j = n_i$.

Note that the basic idea here is to select the most discriminative or representative cluster in each class, and then infer its latent variables by maximizing a posterior probability or a joint likelihood. Suppose we have the discriminative weights (or posterior probability) corresponding to its K components in each class, $w_i = \{w_i^1, w_i^2, ..., w_i^K\}$, which are the posterior probability determined by LFDA and will be discussed later in the next part. We maximize one of the following two objectives:

Maximizing a posterior probability:

$$\mu_i^j = \underset{\mu_i^j \in \mu_i, j \in [1,K]}{\operatorname{argmax}} \quad w_i = \underset{\mu_i^j \in \mu_i, j \in [1,K]}{\operatorname{argmax}} \quad p(z_i|\mu_i, \boldsymbol{\mathcal{P}}) \tag{5a}$$

Maximizing the joint probability with prior:

$$\mu_i^j = \underset{\mu_i^j \in \mu_i, j \in [1,K]}{\operatorname{argmax}} \quad (\pi_i \circ w_i) \tag{5b}$$

where z_i is the latent label assignment, π_i is the data-driven prior for each class i, w_i is the posterior (or weight) determined by kNN voting (see further) in the subspace and $\circ$ is the pointwise production or Hadamard product. We treat Eq. (5a) as the latent Fisher discriminant analysis model (LFDA), because it takes the same strategy as latent SVM model (Andrews, Tsochantaridis, and Hofmann 2002; Felzenszwalb et al. 2010). As for Eq. (5b), we extend LFDA by combining both representative and discriminative factors together, and find the cluster S_i^j in class i by maximizing Eq. (5b). In a sense, Eq. (5b) considers the prior distribution from the training dataset, thus, we treat it as the joint latent Fisher discriminant analysis model (JLFDA) or LFDA with prior. In a nutshell, we propose a way to formulate discriminative and generative models together under Bayesian framework. We comparatively analyze both of these models in experiments.

Consequently, if we select the cluster S_i^j with the mean μ_i^j which maximizes the above equation (for example, Eq. (5a)) for class i, we can relabel all samples $y \in S_i^j$ positive for class i and the rest negative, subject to $y = \boldsymbol{\mathcal{P}}x$ and $y \in S_i^j$. And further we can decide the label of x with the assumption $z(y) = z(x)$. Thus, for any $x \in \boldsymbol{\mathcal{X}}$, we decide its label $z(x)$ based on the following

$$z(x) = l_i, \text{ if } y = \boldsymbol{\mathcal{P}}x \in S_i^j \text{ which maximizes Eq. (5)} \tag{6}$$

Then, we update the training data $\boldsymbol{\mathcal{X}}^+ = \{\mathbf{x_1^+}, \mathbf{x_2^+}, ..., \mathbf{x_n^+}\}$, with labels $\boldsymbol{\mathcal{L}}^+ = \{\mathbf{z_1^+}, \mathbf{z_2^+}, ..., \mathbf{z_n^+}\}$, where $\mathbf{x_i^+} = S_i^j$ for class i with $n_i^{j'}$ elements, and its labels $\mathbf{z_i^+} = \{z_i^1, z_i^2, ..., z_i^{n_i^{j'}}\}$ on instance level. In a sense, we are doing a kind of selection, which finds the most discriminative and representative component for each class from mixture of Gaussians. Obviously, $\mathbf{x_i^+} \subseteq \mathbf{x_i}$ and $\boldsymbol{\mathcal{X}}^+$ is a subset of $\boldsymbol{\mathcal{X}}$. The difference between $\boldsymbol{\mathcal{X}}^+$ and $\boldsymbol{\mathcal{X}}$ lies that every element $x_i^+ \in \mathbf{x_i^+}$ has label $z(x_i^+)$ decided by Eq. (6), while $\mathbf{x_i} \subset \boldsymbol{\mathcal{X}}$ only has bag level label.

Updating projection $\boldsymbol{\mathcal{P}}$

After we decide labels for the new training data $\boldsymbol{\mathcal{X}}^+$, we can use LDA to minimize $J(\boldsymbol{\mathcal{P}}, Z)$. Note that Eq. (1) is invariant to the scale of the vector $\boldsymbol{\mathcal{P}}$. Hence, we can always choose $\boldsymbol{\mathcal{P}}$ such that the denominator is simply $\boldsymbol{\mathcal{P}}^T \Sigma_b \boldsymbol{\mathcal{P}} = \mathbf{1}$. For this reason we can transform the problem of minimizing Eq. (1) into the following constrained optimization problem (Duda, Hart, and Stork 2000; Fukunaga 1990; Ye 2007):

$$\begin{aligned} \boldsymbol{\mathcal{P}}^* = \underset{\boldsymbol{\mathcal{P}}}{\operatorname{argmin}} & \operatorname{trace}\left(\boldsymbol{\mathcal{P}}^T \Sigma_w(\boldsymbol{\mathcal{X}}^+, \boldsymbol{\mathcal{L}}, Z)\boldsymbol{\mathcal{P}} + \beta \boldsymbol{\mathcal{P}}^T \boldsymbol{\mathcal{P}}\right) \\ s.t. \quad & \boldsymbol{\mathcal{P}}^T \Sigma_b(\boldsymbol{\mathcal{X}}^+, \boldsymbol{\mathcal{L}}, Z)\boldsymbol{\mathcal{P}} = \mathbf{1} \end{aligned} \tag{7}$$

where $\mathbf{1}$ is the identity matrix in $\mathbb{R}^{d' \times d'}$. The optimal Multi-class LDA consists of the top eigenvectors of $\left(\Sigma_w(\boldsymbol{\mathcal{X}}^+, \boldsymbol{\mathcal{L}}, Z) + \beta\right)^\dagger \Sigma_b(\boldsymbol{\mathcal{X}}^+, \boldsymbol{\mathcal{L}}, Z)$ corresponding to the nonzero eigenvalues (Fukunaga 1990), here $(\Sigma_w(\boldsymbol{\mathcal{X}}^+, \boldsymbol{\mathcal{L}}, Z) + \beta)^\dagger$ denotes the pseudo-inverse of $(\Sigma_w(\boldsymbol{\mathcal{X}}^+, \boldsymbol{\mathcal{L}}, Z) + \beta)$. After we calculated $\boldsymbol{\mathcal{P}}$, we can project $\boldsymbol{\mathcal{X}}^+$ into subspace $\boldsymbol{\mathcal{Y}}^+$. Note that in the subspace $\boldsymbol{\mathcal{Y}}^+$, any

$y^+ \in \boldsymbol{\mathcal{Y}}^+$ preserves the same labels as in the original space. In other words, $\boldsymbol{\mathcal{Y}}^+$ has corresponding labels $\boldsymbol{\mathcal{L}}^+$ at element level, namely $z(y^+) = z(x^+)$.

In general, multi-class LDA (Ye 2007) uses kNN to classify new input data. We compute w_i using the following kNN strategy: for each sample $x \in \boldsymbol{\mathcal{X}}$, we get $y = \boldsymbol{\mathcal{P}}x$ by projecting it into subspace $\boldsymbol{\mathcal{Y}}$. Then, for $y \in \boldsymbol{\mathcal{Y}}$, we choose its N nearest neighbors from $\boldsymbol{\mathcal{Y}}^+$, and use their labels as a vote for each cluster S_i^j in each class i. Then, we compute the following posterior probability:

$$\begin{aligned} w_i^j &= p(z_i = 1|\mu_i^j) \propto p(\mu_i^j|z_i = 1)p(z_i = 1) \\ &= p(z_i = 1)\frac{p(\mu_i^j, z_i = 1)}{\sum_{i=1}^{C} p(\mu_i^j, z_i = 1)} \end{aligned} \tag{8}$$

It counts all $y \in \boldsymbol{\mathcal{Y}}$ that fall into N nearest neighbor of μ_i^j with label z_i. Note that kNN is widely used as the classifier in the subspace after LDA transformation. Thus, Eq. (8) consider all training data to vote the weight for each discriminative cluster S_i^j in every class i. Hence, we can find the most discriminative cluster S_i^j, s.t. $w_i^j > w_i^k, k \in [1, K], k \neq j$.

Algorithm

The basic idea behind our method is that we use Gaussian mixture model to partition all instances in each class, which can get the data driven prior for each component. Then, we infer the latent variable at the component level, which can be used further to label each instance in that cluster. Finally, given the labeled data, we can use LDA to find a discriminative transformation. Such processes repeat until convergence. We summarize the above discussion in pseudo code in Algorithm 1. To put simply, we update $\boldsymbol{\mathcal{P}}$ and z in an alternative manner, and accept the new projection matrix $\boldsymbol{\mathcal{P}}$ with LDA on the relabeled instances. Such algorithm can always converge very fast (e.g. 10 iterations). After we learned matrix $\boldsymbol{\mathcal{P}}$ and $\{\lambda_i\}_{i=1}^C$ by maximizing Eq. (5), we can use them to select representative and discriminative features (i.e. frames from video datasets) by nearest neighbor searching.

Convergence analysis

Our method updates latent variable z and then $\boldsymbol{\mathcal{P}}$ in an alternating manner. Such strategy can be attributed to the hard assignment of EM algorithm. Recall the EM approach:

$$\begin{aligned} \boldsymbol{\mathcal{P}}^* &= \underset{\boldsymbol{\mathcal{P}}}{\operatorname{argmax}}\, p(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}|\boldsymbol{\mathcal{P}}) = \underset{\boldsymbol{\mathcal{P}}}{\operatorname{argmax}} \sum_{i=1}^{C} p(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, z_i|\boldsymbol{\mathcal{P}}) \\ &= \underset{\boldsymbol{\mathcal{P}}}{\operatorname{argmax}} \sum_{i=1}^{C} p(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}|z_i)p(z_i|\boldsymbol{\mathcal{P}}) \end{aligned} \tag{9}$$

then the likelihood can be optimized using iterative use of the EM algorithm.

Theorem 1. *Assume the latent variable z is inferred for each instance in $\boldsymbol{\mathcal{X}}$, then maximizing the above function is equivalent to maximizing the following auxiliary function*

$$\boldsymbol{\mathcal{P}} = \underset{\boldsymbol{\mathcal{P}}}{\operatorname{argmax}} \sum_{i=1}^{C} p(z_i|\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{P}}')ln\Big(p(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}|z_i)p(z_i|\boldsymbol{\mathcal{P}})\Big) \tag{10}$$

Algorithm 1

Input: training data $\boldsymbol{\mathcal{X}}$ and its labels $\boldsymbol{\mathcal{L}}$ at video level, β, K, N, T and ϵ.
Output: $\boldsymbol{\mathcal{P}}$, $\{\lambda_i\}_{i=1}^C$

```
Initialize P and w_i;
for Iter = 1 to T do
    for i = 1; i <= C; i++ do
        Project all the training data X into subspace Y using Y=PX;
        For each class, using the Gaussian mixture model to partition its elements in the subspace, and learn λ_i = {π_i, μ_i, Σ_i};
        Maximize Eq. (5) to find S_i^j with center μ_i^j;
        Relabel all elements positive in the cluster S_i^j for class i according to Eq. (6);
    end for
    Update z and construct the new subset X^+ and its labels L^+ for all C classes;
    Do Fisher linear discriminant analysis and update P;
    if P converge (change less than ε), then break;
    Compute N nearest neighbors for each training data, and calculate discriminative weight w_i for each class i according to Eq. (8).
end for
Return P and cluster centers {λ_i}_{i=1}^C learned respectively for all C classes;
```

where $\boldsymbol{\mathcal{P}}'$ is the old $\boldsymbol{\mathcal{P}}$. This proof can be shown using Jensen's inequality.

Lemma 1.1. *The hard assignment of latent variable z by maximizing Eq. (5) is a special case of EM algorithm.*

Proof.

$$\begin{aligned} &\sum_{i=1}^{C} p(z_i|\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{P}}')ln\Big(p(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}|z_i)p(z_i|\boldsymbol{\mathcal{P}})\Big) \\ =&\sum_{i=1}^{C} p(z_i|\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{P}}')ln(p(z_i|\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{P}})) \\ +&\sum_{i=1}^{C} p(z_i|\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{P}}')ln(p(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}|\boldsymbol{\mathcal{P}})) \\ =&\sum_{i=1}^{C} p(z_i|\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{P}}')ln(p(z_i|\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{P}})) + ln(p(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}|\boldsymbol{\mathcal{P}})) \end{aligned} \tag{11}$$

Given $\boldsymbol{\mathcal{P}}'$, we can infer the latent variable z. Because the hard assignment of z, the first term in the right hand side of Eq. (11) assigns z_i into one class. Note that $p(z|\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{P}})ln(p(z|\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{P}}))$ is a monotonically increasing function, which means that by maximizing the posterior likelihood $p(z|\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{P}})$ for each instance, we can maximize Eq. (11) for the hard assignment case in Eq. (5). Thus, the updating strategy in our algorithm is a special case of EM algorithm, and it can converge into a local maximum as EM algorithm. Note that in our implementation, we infer the latent variable in cluster level. In other words, to maximize $p(z_i|\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{P}}')$, we can include another latent

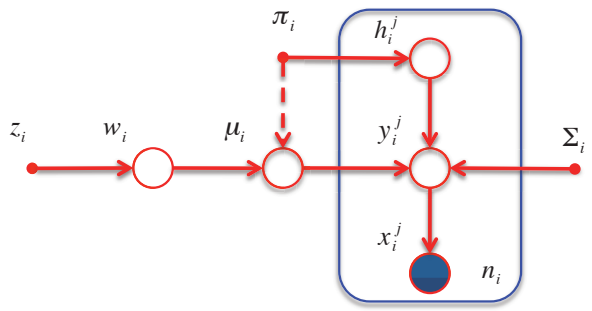

Figure 1: Example of graphical representation only for one class (event) in our model. h_i^j is the hidden variable, x_i^j is the observable input, y_i^j is the projection of x_i^j in the subspace, $j \in [1, n_i]$, and n_i is the number of total training data for class i. The K cluster centers $\mu_i = \{\mu_i^1, \mu_i^2, ..., \mu_i^K\}$ is determined by both π_i and w_i. The graphical model of our method is similar to GMM model in vertical. By adding z_i into LDA, the graphical model can handle latent variables.

variable $\pi_i^j, j \in [1, K]$. Specifically, we need to maximize $\sum_{j=1}^{K} p(z_i, \pi_i^j | \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{P}}')$, which we can recursively determine the latent variable π_i using an embedded EM algorithm. Hence, our algorithm uses two steps of EM algorithm, and it can converge to a local maximum. Refer to (Wu 1983) for more details about the convergence of EM algorithm. □

Probabilistic understanding for the model

Latent SVM model (Felzenszwalb et al. 2010; Andrews, Tsochantaridis, and Hofmann 2002) attempts to label instance x_i in positive bag, by maximizing $p(z(x_i) = 1|x_i)$, which is the optimal Bayes decision rule. Similarly, Eq. (5a) takes the same strategy as latent SVM to maximize a posterior probability. Moreover, instead of only maximizing $p(z = 1|x)$, we also maximize the joint probability $p(z = 1, x)$, using the Bayes rule, $p(z = 1, x) = p(x)p(z = 1|x)$. In this paper, we use Gaussian mixture model to approximate the prior $p(x)$. We argue that to maximize a joint probability is reasonable, because it considers both discriminative (posterior probability) and representative (prior) property in the video dataset. We give the graphical representation of our model in Fig. 1.

Experiments and results

In this section, we perform experiments on various data sets to evaluate the proposed techniques and compare it to other baseline methods. For all the experiments, we set T = 20 and $\beta = 40$ if there is no other specification; and initialize uniformly weighted w_i and projection matrix $\boldsymbol{\mathcal{P}}$ with LDA.

Classification on toy data sets

The MUSK data sets[1] are the benchmark data sets used in virtually all previous approaches and have been described in detail in the landmark paper (Dietterich, Lathrop, and Lozano-Pérez 1997). Both data sets, MUSK1 and MUSK2, consist of descriptions of molecules using multiple low-energy conformations. Each conformation is represented by a 166-dimensional feature vector derived from surface properties. MUSK1 contains on average approximately 6 conformation per molecule, while MUSK2 has on average more than 60 conformations in each bag. The Corel data set consists three different categories ("elephant", "fox", "tiger"), and each instance is represented with 230 dimension features, characterized by color, texture and shape descriptors. The data sets have 100 positive and 100 negative example images. The latter have been randomly drawn from a pool of photos of other animals. We first use PCA reducing its dimension into 40 for our method. For parameter setting, we set K=3, T = 20 and $N = 4$ (namely the 4-Nearest-Neighbor (4NN) algorithm is applied for classification) on all datasets except Elephant (we set $K = 2$ for it). The averaged results of 10-fold cross-validation runs are summarized in Table (1). We set LDA[2] and MI-SVM as our baselines. We can observe that JLFDA outperforms LDA and MI-SVM on MUSK1, Elephant and Fox data sets, especially our method shows significantly better result on MUSK1 data set. We also show the average accuracy over the five datasets, and it demonstrates that our model is better than MI-SVM with higher accuracy.

Data set	inst/Dim	MI-SVM	LDA	LFDA	JLFDA
MUSK1	476/166	77.9	70.4	81.4	**87.1**
MUSK2	6598/166	**84.3**	51.8	76.4	81.3
Elephant	1391/230	81.4	70.5	74.5	**82.2**
Fox	1320/230	57.8	53.5	**61.5**	**59.5**
Tiger	1220/230	**84.0**	71.5	74.0	80.5
Average	-	77.08	63.54	73.56	**78.1**

Table 1: Accuracy results for various methods on MUSK and Corel datasets. Our approach outperforms LDA significantly, and we get better result than MI-SVM on MUSK1, Elephant and Fox datasets. On average, our method outperforms MI-SVM, which indicates that our model is stable for the semi-supervised classification.

Semantic keyframe extraction

Keyframe defines the starting and ending points of any smooth transition in a video. Semantic keyframe in our paper refers the keyframes which have semantic meanings. In other words, we call tell what event or topic happened in the video when we observe the keyframes. We conduct experiments on the challenging TRECVID MED11 dataset[3] with five events (or classes): attempting a board trick, feeding an animal, landing a fish, wedding ceremony and working on a woodworking project. As for parameters, we set $K = 10$ and $N = 10$. We learned the representative clusters for each class (event), and then use them to find semantic frames in

[1] www.cs.columbia.edu/ andrews/mil/datasets.html

[2] we use the bag label as the instance label to test its performance

[3] http://www.nist.gov/itl/iad/mig/med11.cfm

videos with the same labels. Then we evaluate the semantic frames for each video through human-factors analysis—the semantic keyframe extraction problem demands a human-in-the-loop for evaluation.

Video representation. For all videos, we extract HOG3D descriptors (Klaser, Marszalek, and Schmid 2008) every 25 frames (about sampling a frame per second). To represent videos using local features we apply a bag-of-words model, using all detected points and a codebook with 1000 elements.

Benchmark methods. We make use of SVM as the benchmark method in the experiment. We take the one-vs-all strategy to train a linear SVM classifier using SVM^{light} (Joachims, Finley, and Yu 2009), for each kind of event. Then we choose 10 frames for each video which are far from the margin and close to the margin on positive side. For the frames chosen farthest away from the margin, we refer it SVM(1), while for frames closest to the margin we refer it SVM(2). We also randomly select 10 frames from each video, and we refer it RAND in our experiments.

Experimental setting. Ten highly motivated graduate students (range from 22 to 30 years' old) served as subjects in all of the following human-in-the-loop experiments. Each novel subject to the annotation-task paradigm underwent a training process. Two of the authors gave a detailed description about the dataset and problem, including its background, definition and its purpose. In order to indicate what representative and discriminative means for each event, the two authors showed videos for each kind of event to the subjects, and make sure all subjects understand what semantic keyframes are. The training procedure was terminated after the subject's performance had stabilized. We take a pairwise ranking strategy for our evaluation. We extract 10 frames per video for 5 different methods (SVM(1), SVM(2), LFDA, JLFDA and RAND) respectively. For each video, we had about 1000 image pairs for comparison. We had developed an interface using Matlab to display the two image pair and three options (Yes, No and Equal) to compare an image pair each time. The students are taught how to use the software; a trial requires them to give a ranking: If the left image is better than the right, then choose 'Yes'; if the right is better than the left, choose 'No'. If the two images are same, then choose 'Equal'. The subjects are again informed that better means a better semantic keyframe. The ten subjects each installed the software to their computers, and conducted the image pair comparison independently.

Experimental Results. We have scores for each image pair. Then, by sampling 10 videos from each event, we at last had annotations of 104 videos. It means our sampling videos got from 10 subjects almost cover all test data (105 videos). Table 2 shows the win-loss matrix between five methods by counting the pairwise comparison results on all 5 events. It shows that JLFDA and LFDA always beat the three baselines. Furthermore, JLFDA is better than LFDA because it considers data-driven prior, which will help JLFDA to find more representative frames. Refer to supplementary material for keyframes extracted with JLFDA. We compared the five methods on the basis of Condorcet voting method. We treat 'Yes', 'No' and 'Equal' as voters for each method in the pairwise comparison. If 'Yes', we cast one ballot to the left method; else if 'No', we add a ballot to the right method; else do nothing to the two methods. Fig. 2 shows ballots for each method on each event. It demonstrates our method JLFDA always beat other methods, except for the E004 dataset.

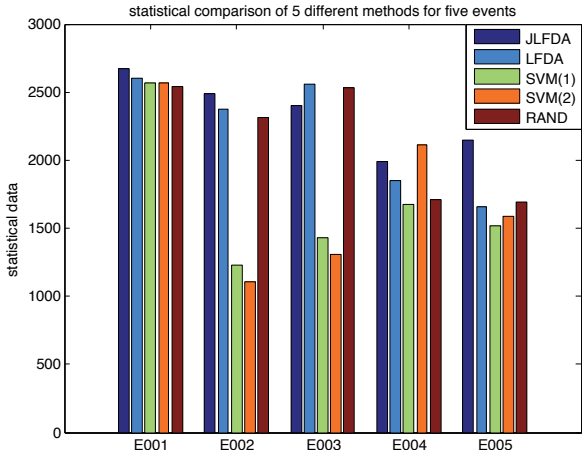

Figure 2: Comparison of 5 methods for five events. Higher value, better performance.

Method	Win-Loss matrix				
	JLFDA	FLDA	RAND	SVM(1)	SVM(2)
JLFDA	-	3413	2274	**2257**	**3758**
LFDA	2957	-	**2309**	2230	**3554**
RAND	2111	2175	-	1861	2274
SVM(1)	2088	2270	2010	-	2314
SVM(2)	3232	3316	2113	2125	-

Table 2: Win-Loss matrix for five methods. It represents how many times methods in each row win methods in column.

Method	the number of No.1 method in each event				
	E001	E002	E003	E004	E005
JLFDA	**6**	**7**	**7**	3	**7**
LFDA	**6**	4	4	5	1
SVM(1)	4	4	4	2	4
SVM(2)	6	3	1	**7**	6
RAND	2	2	4	3	2

Table 3: Higher value, better results. It demonstrates that our method is more capable at extracting semantically meaningful keyframes.

We also compared the five methods based on Elo rating system[4]. For each video, we ranked the five methods according to Elo ranking system. Then, we counted the number of No.1 method on video level in each event. The results in Table 3 show that our method is better than others, except E004. For example, E002 has total 20 videos (column summation), where JLFDA has ranked first on 7 videos, while RAND ranks first on only 2 videos. Such results is consistent

[4]https://en.wikipedia.org/wiki/Elo_rating_system

with that using Condorcet ranking method in Fig. 2. E004 is the wedding ceremony event and our method is consistently outperformed by the SVM baseline method. We believe this is due to the distinct nature of the E004 videos in which the video scene context itself distinguishes it from the other four events (the wedding ceremonies typically have very many people and are inside). Hence the discriminative component of the methods are taking over, and the SVM is able to outperform our model.

Conclusion

In this paper, we have presented a latent Fisher discriminant analysis model with representative feature learning, which combines the latent variable, multi-class and dimension reduction in an unified framework. To the best of our knowledge, this is the first paper to generalize LDA and study the extraction of semantically representative and discriminative features together rather than in a separate manner (either representative or discriminative). We conduct a thorough experiments on MUSK, Corel and TRECVID MED11 datasets, and demonstrate that our method is able to consistently outperform competitive baselines.

Acknowledgement

The authors would like to thank Dr. Jason J. Corso for inspiring discussions on the whole idea and valuable feedback on the paper. Also thank all VPML members in SUNY at Buffalo conducting human-factors ranking-based experimental evaluation on TRECVID MED11 datasets. In addition, we also like to thank the anonymous reviewers for their helpful comments.

References

Andrews, S.; Tsochantaridis, I.; and Hofmann, T. 2002. Support vector machines for multiple-instance learning. In *NIPS*, 561–568.

Baudat, G., and Anouar, F. 2000. Generalized discriminant analysis using a kernel approach. *Neural Computation* 12:2385–2404.

Belhumeur, P. N.; Hepanha, J. P.; and Kriegman, D. J. 1997. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE TPAMI* 19:711–720.

Blum, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the Workshop on Computational Learning Theory*, 92–100.

Cai, D.; He, X.; and Han, J. 2007. Semi-supervised discriminant analysis. In *ICCV*. IEEE.

Chen, G., and Corso, J. J. 2012. Greedy multiple instance learning via codebook learning and nearest neighbor voting. *Arxiv preprint*.

Chen, L.; Liao, H.; Ko, M.; Lin, J.; and Yu, G. 2000. A new lda-based face recognition system which can solve the small sample size problem. *Pattern Recognition* 33:1713–1726.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B* 39(1):1–38.

Dietterich, T. G.; Lathrop, R. H.; and Lozano-Pérez, T. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.* 89:31–71.

Ding, C., and Li, T. 2007. Adaptive dimension reduction using discriminant analysis and k-means clustering. In *ICML*, 521–528. New York, NY, USA: ACM.

Duda, R. O.; Hart, P. E.; and Stork, D. G. 2000. *Pattern Classification (2nd Edition)*. Wiley-Interscience.

Felzenszwalb, P. F.; Girshick, R. B.; McAllester, D.; and Ramanan, D. 2010. Object detection with discriminatively trained part based models. *TPAMI* 32:1627–1645.

Fisher, R. A. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 7(7):179–188.

Fukunaga, K. 1990. *Introduction to statistical pattern recognition (2nd ed.)*. San Diego, CA, USA: Academic Press Professional, Inc.

Hastie, T., and Tibshirani, R. 1996. Discriminant analysis by gaussian mixtures. *Journal of the Royal Statistical Society, Series B* 58:155–176.

Joachims, T.; Finley, T.; and Yu, C.-N. J. 2009. Cutting-plane training of structural svms. *JMLR* 77:27–59.

Joachims, T. 1999. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, 200–209.

Klaser, A.; Marszalek, M.; and Schmid, C. 2008. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*.

Maron, O., and Lozano-Prez, T. 1998. A framework for multiple-instance learning. In *NIPS*.

Maron, O., and Ratan, A. L. 1998. Multiple-instance learning for natural scene classification. In *ICML*, 341–349. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Merchante, L. F. S.; Grandvalet, Y.; and Govaert, G. 2012. An efficient approach to sparse linear discriminant analysis. In *ICML*.

Perera, A.; Oh, S.; Leotta, M.; Kim, I.; Byun, B.; Lee, C.-H.; McCloskey, S.; Liu, J.; Miller, B.; Huang, Z. F.; Vahdat, A.; Yang, W.; Mori, G.; Tang, K.; Koller, D.; Fei-Fei, L.; Li, K.; Chen, G.; Corso, J. J.; Fu, Y.; and Srihari, R. K. 2011. GENIE TRECVID2011 multimedia event detection: Late-fusion approaches to combine multiple audio-visual features. In *NIST TRECVID Workshop*.

Rao, C. R. 1948. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society - Series B* 10(2):159–203.

Sugiyama, M.; Idé, T.; Nakajima, S.; and Sese, J. 2010. Semi-supervised local fisher discriminant analysis for dimensionality reduction. In *Machine Learning*, volume 78, 35–61.

Vapnik, V. N. 1998. *Statistical learning theory*. John Wiley & Sons.

Wu, Y.; Wipf, D. P.; and Yun, J. 2015. Understanding and evaluating sparse linear discriminant analysis. In *AISTATS*.

Wu, C. F. J. 1983. On the convergence properties of the EM algorithm. *The Annals of Statistics* 11(1):95–103.

Ye, J. 2007. Least squares linear discriminant analysis. In *ICML*.

Zhang, Y., and Yeung, D.-Y. 2008. Semi-supervised discriminant analysis via cccp. In *ECML*. IEEE.