OSU Dept. of Computer Science and Engineering Technical Report OSU-CISRC-2/06-TR23.

Emphasizing Key Features in LDA using Bootstrap Bumping

Hui Gao and James W. Davis
Dept. of Computer Science and Engineering
The Ohio State University
2015 Neil Ave, Columbus, OH 43220, USA
{gaoh, jwdavis}@cse.ohio-state.edu

Abstract

We present a novel LDA-based method for dimensionality reduction and recognition by emphasizing key features. The approach is based on a statistical framework called Bootstrap Bumping LDA (BB-LDA), which specifically deals with the Small Sample Size (SSS) problem in LDA by sampling examples from the training set to hypothesize different representations and selecting the one yielding the best discrimination. In our new approach, a feature weight is calculated for each input dimension to indicate its importance for discrimination. Multiple weight maps are generated from the feature weights to control the scaling and are applied to each representation in BB-LDA to create multiple hypothesis with different emphasis of input dimensions. By selecting both key examples and key features, our new approach shows clear performance improvements over BB-LDA and a significant performance gain over traditional LDA methods.

1. Introduction

In the image-based approach of Computer Vision, a large input space (*e.g.*, rasterized images) is often represented by a relatively small number of examples. To avoid the curse of dimensionality [16] and to speed up the classification process, feature extraction is usually employed at the first step to reduce the dimensionality. As a statistical method, Linear Discriminant Analysis (LDA) has been widely employed for feature extraction and classification (*e.g.*, in face and gait recognition [2, 18, 5, 15]). It assumes multiple Gaussians with equal covariance and is optimal under Bayesian decision theory.

However, LDA does not directly address the curse of dimensionality. Like any statistical method, it requires a large amount of examples ($N \gg D$) relative to the input dimension D to accurately estimate the probability distribution (*e.g.*, model parameters such as the class means μ_i and common covariance Σ in LDA) with Maximum Likelihood (ML) estimates. This results in unstable or even singular solutions (the common covariance estimate $\hat{\Sigma}$ is singular when $N < D$) of LDA in Computer Vision applications, which is the so-called Small Sample Size (SSS) problem.

The traditional LDA methods [17, 2, 18, 29, 12] focus only on the singularity problem of $\hat{\Sigma}$ (when $N < D$), but ignore the issue of accurately estimating of the true model parameters. Various assumptions have been made to invert the singular $\hat{\Sigma}$. However, even if $N > D$, as long as N is not much larger than D, the estimation problem persists due to the curse of dimensionality.

In our recent approach [13], we introduced a variant of the general statistical framework of bootstrap bumping [26] to specifically deal with the SSS problem in LDA without imposing explicit assumptions. We refer to it as Bootstrap Bumping LDA (BB-LDA). A linear representation is hypothesized from each bootstrap sample (subset of examples) and the final model (representation) is selected having the best recognition performance. This extension not only preserves the asymptotical property in the original bumping procedure, but now improves the estimation accuracy and implicitly handles the singularity problem of $\hat{\Sigma}$ in the SSS problem. Experiments on both synthetic and real datasets showed advantages over traditional LDA methods.

Since BB-LDA uses key examples to directly hypothesize a linear representation, each input dimension (*e.g.*, pixel) is treated equally in the projection. However, certain dimensions may be more discriminative than others. If the representation can be biased more towards those important dimensions, the discriminative information can be better kept within the hypothesized subspace. Therefore, it is possible to further improve the performance of BB-LDA.

Motivated by this idea, in this paper we introduce the concept of key features in BB-LDA. Each input dimension (*e.g.*, pixel) is associated with a feature weight, which locally measures the discrimination power. Multiple weight maps are formed by controlling the scale of the feature weights and are employed in BB-LDA to create multiple representation hypotheses from each bootstrap sample, which have different a emphasis on input dimensions. The final representation is selected that yields the best recognition performance. Therefore, the new procedure seeks both *key examples* and *key features* to create the best representation for recognition. We demonstrate this new approach using the same datasets as in [13], and show clear performance improvements over the original BB-LDA method and a significant performance gain over traditional LDA methods.

In the remainder of this paper, we first discuss the background and related work of LDA and BB-LDA in Sect. 2. Then we describe the key feature extension to BB-LDA in Sect. 3. Experimental results are presented in Sect. 4, followed by conclusion in Sect. 5.

2. Background and Related Work

There are two different perspectives of LDA. Fisher's LDA is defined by maximizing the ratio of the between-class and within-class scatter matrices (S_b and S_w) in a linear subspace [9, 21]. In Bayesian decision theory, LDA is defined for multiple Gaussians with equal covariance. The two approaches were shown to be equivalent in [4] with S_w being the ML estimate $\hat{\Sigma}$ and S_b being derived from the ML estimates of the class means. The mathematical description of both approaches can be found in detail in [7].

2.1. LDA and the SSS Problem

Although well-grounded in theory, LDA faces the challenge of the SSS problem in real applications. The traditional methods only aim to solve the singularity problem of $\hat{\Sigma}$. The simple approach PINV-LDA [17] substitutes the inverse operation with the pseudoinverse. The two-stage method PCA+LDA [2] projects the data in the nearly complete PCA subspace to make the $\hat{\Sigma}$ projection just full rank. However, with a small amount of examples, $\hat{\Sigma}$ is unstable especially in components with small eigenvalues which are emphasized in the inverse operation. Both PINV-LDA and PCA+LDA are sensitive to noise and small perturbations.

As one improvement, Enhanced Fisher's Linear Discriminant (EFLD) [18] varies the number of PCA components to regulate the projection of $\hat{\Sigma}$ by assuming the small components being non-informative for classification (potentially limiting the performance). Direct LDA (D-LDA) [29] assumes the null space of S_b contains no useful information for classification. However, as shown in [14], D-LDA is equivalent to directly taking the linear space of class means as the LDA solution. It has severe limitations by ignoring the common covariance estimate $\hat{\Sigma}$ (or S_w). Lastly, $\hat{\Sigma}$ can be directly modified to avoid the singularity problem, such as $\hat{\Sigma} + \sigma I$ in Regularized LDA (R-LDA). With σ usually being a small scalar, R-LDA heavily relies on the small components and even null components for recognition, which is neither stable nor supported by the existing examples. Furthermore, due to the high computational cost to invert a full-rank $D \times D$ matrix $\hat{\Sigma} + \sigma I$ (for a large input dimension D), R-LDA may not be computationally feasible in real applications (see Sect. 4).

Additionally, there are efforts to address the model limitations of LDA by extracting non-linear features (*e.g.*, Quadratic Discriminant Analysis (QDA) [7], kernel-based Generalized LDA [1]), finding linear features for multiple Gaussians with non-equal covariance [27], and allowing classifiers other than thresholding (assumed by LDA) [19]. However since more complex models usually require more examples to constrain the solution, these extensions are often more sensitive to the SSS problem.

2.2. Bootstrap Bumping LDA (BB-LDA)

As the traditional LDA methods only focus on the singularity problem of $\hat{\Sigma}$, they lack the systematic attempts to improve the accuracy of the model parameter estimation (class means and common covariance). In our recent approach (BB-LDA), we introduced a variate of the general statistical framework of bootstrap bumping to specifically deal with the SSS problem in LDA.

2.2.1 Bootstrap Bumping

As a method for model search and inference, the original bumping procedure was proposed in [26] based on bootstrap resampling theory [8]. The bumping procedure follows the paradigm of hypothesize and test. Each bootstrap sample $\mathbf{z}^{*b} \in [\mathbf{z}^{*1}, \mathbf{z}^{*2}, \cdots, \mathbf{z}^{*B}]$ is is a "subset of examples" randomly drawn *with replacement* (at a sampling ratio α) from the original set of training examples $\mathbf{z} = (z_1, z_2, \cdots, z_N)$. A candidate model $\hat{\theta}^{*b}$ is hypothesized from each bootstrap sample $\mathbf{z}^{*b}$ by minimizing a *working* criteria R_0. The best model $\hat{\theta}^{\mathcal{BB}}$ is selected according to a *target* criteria R.

$$\hat{\theta}^{*b} = \text{argmin}_{\theta}\ R_0(\mathbf{z}^{*b}, \theta), \tag{1}$$

$$\hat{\theta}^{\mathcal{BB}} = \hat{\theta}^{*\hat{b}}, \text{ where } \hat{b} = \text{argmin}_{b}\ R(\mathbf{z}, \hat{\theta}^{*b}). \tag{2}$$

The criteria R and R_0 may be the same, or with R_0 being a more convenient, but compatible criteria [26] to R for minimization, which ensures the asymptotic convergence of the procedure to the true model parameters.

Bumping is closely related to other bootstrap-based techniques, such as *bagging* and *boosting*. *Bagging* [3] produces a new estimator, which often has a smaller variance, by averaging the model estimates from multiple bootstrap samples. *Boosting* [24, 10] improves classification by combining multiple weak classifiers, individually trained from a subset of examples (bootstrap sample). As an enhanced version of boosting, AdaBoosting [11, 28] employs adaptive sampling and weighted voting. However, when a LDA-based classifier is desired, the bagged (averaged) linear classifier from subsets may not perform well on the entire dataset, and the boosted classifier results in complex decision boundaries, which is non-linear. Both bagged and boosted LDA [25, 20] are no longer true "LDA". Bumping avoids this issue by selecting the hypothesis that gives the *best* classification rate. The procedure is capable of reducing the variance of the estimates, while preserving the LDA model structure and interpretation.

2.2.2 Bootstrap Bumping LDA

The original bumping procedure [26] is not directly applicable to the SSS problem. Since each bootstrap sample $\mathbf{z}^{*b}$ only contains a subset of training examples, directly estimating the model parameters $\hat{\theta}^{*b}$ from $\mathbf{z}^{*b}$ is more severely influenced by the SSS problem (*e.g.*, boosting only obtains weak classifiers from bootstrap samples). Furthermore, the singularity problem of $\hat{\Sigma}$ in LDA is not yet addressed.

Instead, BB-LDA [13] addresses the SSS problem by first hypothesizing a representation space L^{*b} from $\mathbf{z}^{*b}$ as

$$L^{*b} = \text{argmin}_L \ R_{rep}(\mathbf{z}^{*b}, L). \quad (3)$$

The new *working* criteria R_{rep} measures the capacity of a given representation L (*e.g.*, linear, quadratic, etc.), which is to be minimized and compatible with the model assumption (*representation* criteria). With regards to LDA, a linear subspace defined by $\mathbf{z}^{*b}$ is minimum in terms of capacity among all compatible representations. Therefore we have

$$L^{*b} = LinearSpace(\mathbf{z}^{*b}). \quad (4)$$

For the other models, the representation should be chosen accordingly (*e.g.*, quadratic representation for QDA).

The discrimination performance of each hypothesized representation L^{*b} is evaluated over the entire dataset $\mathbf{z}$, with the best selected as the one with the minimum misclassification rate

$$L^{\mathcal{BB}-\mathcal{LDA}} = L^{*\hat{b}}, \text{ where } \hat{b} = \text{argmin}_b \ R_{dis}(\mathbf{z}, L^{*b}). \quad (5)$$

The new *target* criteria R_{dis} measures the misclassification rate of $\mathbf{z}$ with regards to the representation space L^{*b} (*discrimination* criteria), which can be evaluated by first projecting $\mathbf{z}$ into L^{*b} (*e.g.*, correlating with a linear basis), estimating the model parameters (*e.g.*, using an ML estimator), and lastly calculating the misclassification rate.

Since the performance of LDA is invariant to the basis selection of L^{*b} (the information loss occurs only at the subspace level), we simply choose A_b containing all the examples in $\mathbf{z}^{*b}$ as the linear basis of L^{*b} for simplicity.

$$A_b = [z_1^{*b}, z_2^{*b}, \cdots, z_k^{*b}] \quad (6)$$

The solution of BB-LDA is obtained by reconstructing the model parameters learned in the representation subspace $L^{\mathcal{BB}-\mathcal{LDA}}$ (*e.g.*, multiplying the LDA discrimination vector(s) with the basis). In essence, the approach seeks out the key prototype examples that best represent the space of $\mathbf{z}$ for the purpose of discrimination. Any new example z_{new} is classified by projecting it onto the reconstructed LDA discrimination vector(s) and thresholding as in classic LDA.

Since duplicate examples do not affect the linear representation, bootstrap samples are drawn at a fixed rate αN from $\mathbf{z}$ *without replacement* in BB-LDA. The number of bootstrap samples B can be determined for a particular percentage p of training examples $\mathbf{z}$ covered by all bootstrap samples (*e.g.*, $p = 99.9\%$) with

$$B = log(1-p)/log(1-\alpha). \quad (7)$$

For a fixed coverage p, it has been shown in [13] that the worse case time complexity of BB-LDA is in the same order as traditional subspace LDA [17, 2, 18]. But BB-LDA has the asymptotic property of convergence to the true model parameters (with R_{rep} and R_{dis} being compatible) [13].

The approach of BB-LDA is significant in that it directly addresses the SSS problem in a general statistical framework without imposing specific assumptions (as in the traditional LDA methods). At a particular sampling ratio α, only a portion of examples are used to hypothesize a representation, which can ensure $\hat{\Sigma}$ being full rank in the projection space L^{*b} (under the basis A_b) for the entire dataset $\mathbf{z}$. By appropriately selecting the sampling ratio α (*e.g.*, in cross-validation) to balance the representation and discrimination, BB-LDA improves the accuracy of the model parameter estimation for LDA under the SSS problem. Moreover, the bumping procedure ensures BB-LDA maintains the original LDA interpretation by avoiding averaging (bagging) or voting (boosting).

3. Bootstrap Bumping LDA with Key Features

The approach of BB-LDA emphasizes key examples to hypothesize linear representations for recognition. Since there exists information loss when projecting the entire dataset into the representation subspace, each input dimension should not be equally treated in the projection. Emphasizing key features/dimensions which are more important for discrimination should better preserve the information for recognition and hence improve the performance. In this work, we propose to extend the BB-LDA framework with key features. A feature weight is employed on each input dimension (*e.g.*, pixel) to indicate its importance for discrimination. Multiple weight maps (variants of a weight vector by controlling the scaling) are applied to each hypothesized representation in BB-LDA to emphasize key features for recognition. The new procedure selects both key *examples* and key *features* to achieve the best recognition.

3.1. Embedding Feature Weights

Let w be a $D \times 1$ vector containing all of the feature weights. We apply w to the entire dataset $\mathbf{z}$ by directly scaling each input dimension with the corresponding feature weight

$$\tilde{\mathbf{z}} = diag(w) \cdot \mathbf{z}, \tag{8}$$

where $diag(w)$ denotes a diagonal matrix with w along its diagonal. Then the bootstrap samples $\tilde{\mathbf{z}}^{*b}$ are drawn from the scaled dataset $\tilde{\mathbf{z}}$ to hypothesize each linear subspace $\tilde{L}^{*b}$. The projection of $\tilde{\mathbf{z}}$ into $\tilde{L}^{*b}$ (using A_b) is

$$\begin{aligned} \tilde{y}_b &= [diag(w) \cdot A_b]^T \cdot \tilde{\mathbf{z}} & (9)\\ &= A_b^T \cdot diag(w^2) \cdot \mathbf{z}, & (10) \end{aligned}$$

where w^2 is a $D \times 1$ vector containing the squared feature weights. This indicates that only the magnitude of the feature weights influence the projection. Each element of w can be required to be $0 \leq w_i \leq 1$.

In relation to the previous BB-LDA approach, the new framework is capable of altering the linear subspace by embedding feature weights to emphasize certain input dimensions. Let $\hat{A}_b = diag(w^2)A_b$. Eqn. 10 is equivalent to

$$\tilde{y}_b = [diag(w^2)A_b]^T \cdot \mathbf{z} = \hat{A}_b\mathbf{z} \tag{11}$$

with $\hat{A}_b$ being a basis of a new linear subspace. Except when w is a vector containing all one elements, $\hat{A}_b$ represents a different linear subspace than A_b in capturing information of the entire dataset $\mathbf{z}$ in the reduced subspace. Thus it is possible to improve the recognition performance by properly emphasizing certain features.

Moreover, this key feature approach seamlessly integrates into the BB-LDA framework. Without the subspace representation (*e.g.*, A_b) in BB-LDA, the $D \times D$ dimensional weighting matrix $diag(w)$ is at most a full rank transformation (non-zero weights) for $\mathbf{z}$. Thus w alone has no performance benefits for LDA directly trained in the original input space since LDA is invariant to any full rank transformation (or basis selection).

3.2. Calculating Feature Maps

To obtain the weight vector w, we measure the discriminating power of each input dimension (*e.g.*, pixel) with a Bayesian classifier (multiple Gaussians with equal variance) over the entire training set $\mathbf{z}$. We calculate each feature weight w_i as the Bayesian classification rate after subtracting the chance level performance ($\frac{1}{c}$ for c classes). The vector is then divided by its maximal element to be normalized to between 0 and 1.

However, the calculated weight vector w may not be the best weighting scheme if directly applied in the framework. The internal scaling within w has not yet been considered. Perhaps a weight of 0.5 is as important as 1.0 for one dataset, but not for another. Motivated by the occurrence of w^2 in Eqn. 10, we regulate the weight vector w into multiple weight maps m_q with

$$m_q = w^q \tag{12}$$

where $q \in [0, .125, .25, .5, 1, 2, 4, 8]$. We treat each weight map m_q as a hypothesis for the weight vector to be employed in the BB-LDA framework to bias the representation. Note that m_0 corresponds to uniform weights (as implicitly employed in BB-LDA). A value of $q < 1$ increases the importance of the smaller weights and for $q > 1$ the larger weights receive more strength. The weight map m_1 is equivalent to the original weight vector w. Illustrations of the range of weight maps are shown in Sect. 4 (Fig. 3).

3.3. Algorithm

In the modified BB-LDA algorithm with key features, multiple weight maps are tested with each bootstrap to create multiple hypotheses, which are distinct representations as oppose to the single representation with uniform weights (m_0) as in the original BB-LDA. The bumping procedure selects the representation/model (key examples and key features) having the best recognition performance. The new algorithm is summarized in Alg. 1.

Algorithm 1 BB-LDA Algorithm with Key Features

```
Calculate weight vector $w$ and multiple weight maps $m_q$ with $q = [0, .125, .25, .5, 1, 2, 4, 8]$.
Randomly draw $B$ bootstrap samples (at sampling ratio $\alpha$,) $\mathbf{z}^{*1}, \mathbf{z}^{*2}, \cdots, \mathbf{z}^{*B}$ from entire training set $\mathbf{z}$
for $b = 1$ to $B$ do
    Let $A_b = [z_1^{*b}, z_2^{*b}, \cdots, z_k^{*b}]$.
    for each $m_q$ do
        Let $\hat{A}_b = diag(m_q^2)A_b$.
        Project $\mathbf{z}$ in $\hat{A}_b$ as $y_{bq} = \hat{A}_b^T\mathbf{z}$. Run LDA (ML estimates) on $y_{bq}$ to obtain model parameters (LDA discrimination vector(s) $v_{bq}$ and threshold(s) $t_{bq}$).
        Calculate the misclassification rate on $y_{bq}$ based on the estimated model parameters.
    end for
end for
Choose $A_b$ and $m_q$ which has the minimum misclassification rate. Obtain the solution by reconstructing the LDA discrimination vector(s) $diag(m_q^2)A_b v_{bq}$ and keeping the same threshold $t_{bq}$.
```

4. Experiments

To demonstrate the advantages of emphasizing key features for recognition in BB-LDA, we present experimental results on the same face and gait databases (Yale face database [2], ORL face dataset [23], and the CMU gait database

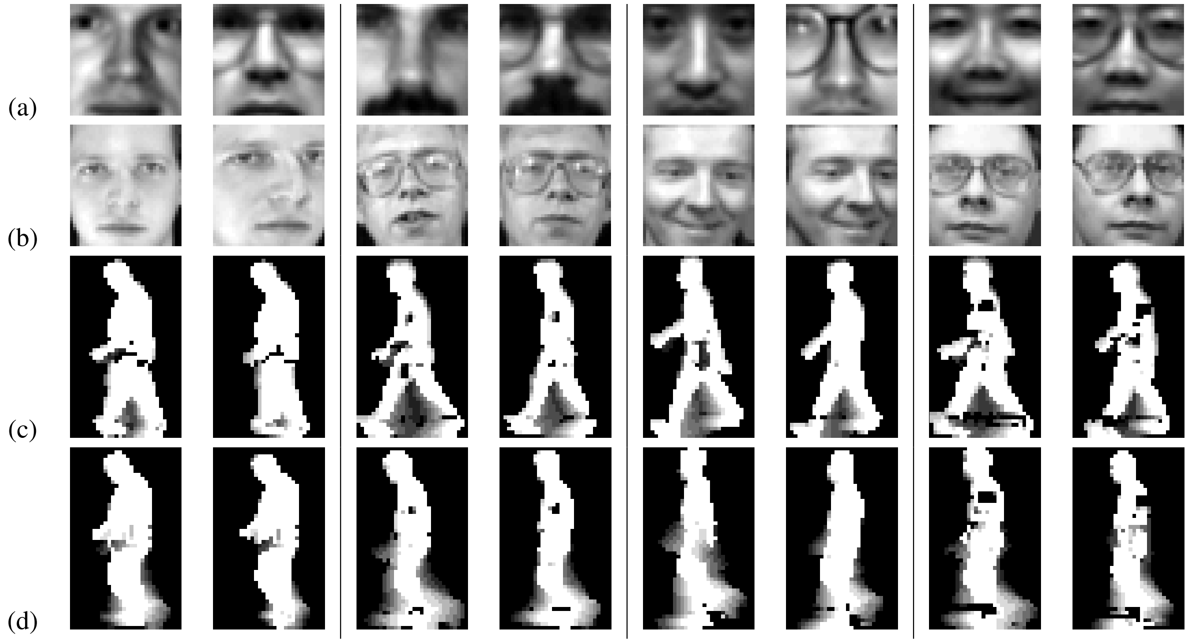


Figure 1. Sample images of 3 datasets. (a) Yale face database (15 subjects, glasses vs. no glasses). (b) ORL face dataset (40 subjects). (c) CMU Gait database (25 subjects, fast vs. slow walk) in Type-1 MHI representation. (d) Corresponding Type-2 MHI Gait representation.

[22]) as used in [13] (see sample images in Fig. 1). We first compare our results to the original BB-LDA approach to demonstrate the advantages of emphasizing key features in BB-LDA. Then we compare the recognition results with the traditional LDA algorithms.

4.1. Datasets

The Yale face database includes 15 subjects and 11 images of each person across various conditions (e.g., lighting, expressions, etc.). In addition to face recognition, we examined the task of distinguishing people with glasses from people without glasses (36 with and 129 without). This is a much larger set than the case of 36 images studied in [2]. We next examined face recognition using the ORL face dataset with 40 subjects and 10 images per person. Lastly, we tested at the CMU gait database of 25 subjects with 16 cycles extracted for each person (8 slow and 8 fast). Both identity and walking speed (slow, fast) recognition were performed over two types of MHIs [6] for two phases in each walking cycle.

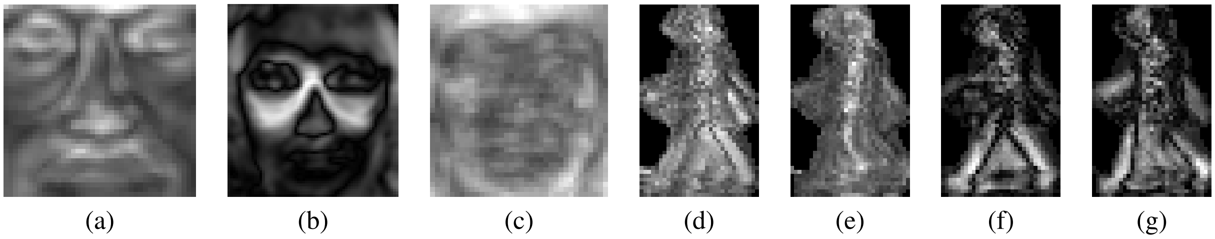
For each dataset, images were aligned to control position and scaling. Then they were down-sampled and cropped to the region of interest (except for the glasses vs. no-glasses case). The average intensity of each image (foreground region only) was also removed. The classification tasks were made slightly more difficult than [13] with 20% of the examples used for testing in cross-validation.

4.2. Improvements to BB-LDA with Key Features

In the BB-LDA extension with key features, we first calculate the weight vector w (from Bayesian classifier) as shown in Fig. 2 for one cross-validation set. As we can see, the weight vector w may not have large internal variations for certain recognition tasks, such as Yale-ID and ORL-ID. This suggest that these tasks may naturally favor uniform emphasis in representations (holistic). However for the others (*e.g.*, Yale-Glasses, CMU-Speed), w has clear emphasis over certain areas (*e.g.*, glasses frame area in Yale-Glasses, stride and arm swing in CMU-Speed), which are intuitively important for recognition. Thus by properly emphasizing those areas with appropriate weight maps in the hypothesized representation, it is possible to improve the recognition performance of BB-LDA.

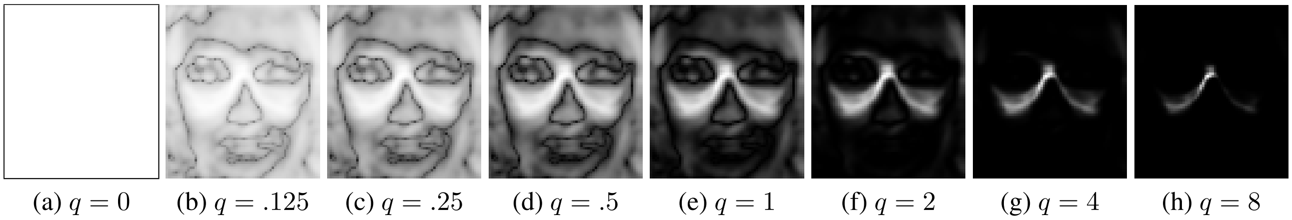
We selected the case of Yale-Glasses to further demonstrate the key feature approach. The range of weight maps are illustrated in Fig. 3. As q increases, there is more emphasis in the glasses frame area. The best weight map is chosen in the bumping procedure along with the key examples to yield the best recognition performance.

Next we compare the extracted LDA discrimination vector(s) between BB-LDA and the new key feature extension. At the sampling ratio $\alpha = 0.3$ (see Table 1), although the original BB-LDA extracts a LDA discrimination vector (Fig. 4a) covering the regions of the glasses frame, a large amount of pixels in unrelated areas are also included (*e.g.*, chin and nose). But with the key feature extension using weight map m_1 (Fig. 3e), our new approach success-



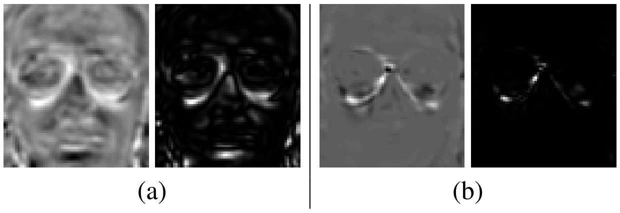
(a) (b) (c) (d) (e) (f) (g)

Figure 2. Weight vector w of each task. (a) Yale-ID, (b) Yale-Glasses, (c) ORL-ID, (d) CMU-ID-Type-1, (e) CMU-ID-Type-2, (f) CMU-Speed-Type-1, (g) CMU-Speed-Type-2.



(a) $q = 0$ (b) $q = .125$ (c) $q = .25$ (d) $q = .5$ (e) $q = 1$ (f) $q = 2$ (g) $q = 4$ (h) $q = 8$

Figure 3. Regulated weight maps m_q for Yale-Glasses.



(a) (b)

Figure 4. Extracted LDA discrimination vector and its corresponding energy image (square of the vector) for Yale-Glasses. (a) BB-LDA, (b) BB-LDA with key features.

fully latches down to the glasses frame region (Fig. 4b) as desired. This results in improved recognition performance over BB-LDA (see Table 1), which clearly shows the benefits of emphasizing key features in the BB-LDA framework.

4.3. Recognition Performance

We quantitatively evaluated the recognition performance of BB-LDA and its key feature extension with the traditional LDA methods, which include PINV-LDA [17], PCA+LDA [2], EFLD [18], and D-LDA [29]. The method of R-LDA [12] was not examined due to its inherit high time complexity (*e.g.*, $D = 1600$ in Yale face database). The optimal model parameters of BB-LDA (the sampling ratio $\alpha \in [0.1 : 0.1 : 0.9]$ and the best representation/classifier with key examples and key features) and EFLD (the number of PCA components) were adjusted in cross-validation. The same bootstrap coverage was kept at $p = 99.9\%$.

The recognition results across the different datasets are summarized in Table 1. Again our previous approach of BB-LDA outperformed the traditional LDA methods by sampling key representative examples for recognition, as demonstrated in [13]. PINV-LDA and PCA+LDA are sensitive to noise due to their overemphasis over small components. D-LDA is a limited case of taking the linear space of class means or S_b as the LDA solution. EFLD gives the best performance among these traditional methods by adjusting the number of PCA components. With our current key feature extension to BB-LDA, the recognition performance has been further improved in all cases due to the proper emphasis of key features in the hypothesized representation. This gives the new BB-LDA framework a significant performance advantages over the traditional LDA methods in dealing with the SSS problem.

5. Conclusion

We presented a novel method for dimensionality reduction and recognition based on Bootstrap Bumping LDA (BB-LDA). Our extension further improves the performance by emphasizing key features in the representation. The method seamlessly integrates into the BB-LDA framework to seek both key examples and key features to hypothesize a representation for the best discrimination. Experiments show clear advantages of the new approach over the original BB-LDA method and significant performance gain over the traditional LDA approaches in dealing with the SSS problem. In future work, we plan to investigate non-linear representation (*e.g.*, quadratic) in the bootstrap bumping framework to extend the approach to non-linear cases (*e.g.*, QDA).

6. Acknowledgments

This research was supported by the NSF Faculty Early Career Development (CAREER) Award IIS-0236653.

	Yale-ID	Yale-Glasses	ORL-ID	CMU-ID (24 sets)		CMU-Speed (30 sets)	
	(11 sets)	(18 sets)	(10 sets)	Type-1	Type-2	Type-1	Type-2
PINV-LDA	83.8	79.4	89.1	93.0	91.7	83.6	83.6
PCA+LDA	38.0	83.5	34.3	37.6	36.8	83.4	85.5
EFLD	88.1 (32 PCs)	90.5 (56 PCs)	91.9 (37 PCs)	97.0 (264 PCs)	96.1 (300 PCs)	90.9 (245 PCs)	91.0 (255 PCs)
D-LDA	70.7	70.8	79.4	72.5	65.7	76.9	79.0
BB-LDA	91.1 ($\alpha = 0.3$)	94.2 ($\alpha = 0.3$)	93.6 ($\alpha = 0.3$)	97.5 ($\alpha = 0.5$)	97.1 ($\alpha = 0.6$)	91.3 ($\alpha = 0.5$)	91.3 ($\alpha = 0.6$)
BB-LDA (Key Features)	**93.3** ($\alpha = 0.4$)	**97.6** ($\alpha = 0.3$)	**94.6** ($\alpha = 0.3$)	**98.2** ($\alpha = 0.5$)	**98.1** ($\alpha = 0.6$)	**92.6** ($\alpha = 0.5$)	**92.8** ($\alpha = 0.6$)

Table 1. Classification results on multiple datasets. The new BB-LDA approach with key features showed clear performance improvements over the original BB-LDA, while it significantly outperformed the traditional methods.

References

[1] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404, 2000. 2

[2] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 19(7):711–720, 1997. 1, 2, 3, 4, 5, 6

[3] L. Breiman. Bagging predictors. *Machine Learning Journal*, 24(2):123–140, 1996. 2

[4] N. Campbell. Canonical variate analysis - a general model formulation. *Australian J. Statistics*, 26:86–96, 1984. 2

[5] Y. Cui, D. Swets, and J. Weng. Learning-based hand sign recognition using SHOSLIF-M. In *Proc. Int. Conf. Comp. Vis.*, pages 631–636. IEEE, 1995. 1

[6] J. Davis and A. Bobick. The representation and recognition of action using temporal templates. In *Proc. Comp. Vis. and Pattern Rec.*, pages 928–934. IEEE, 1997. 5

[7] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2001. 2

[8] B. Efron. Bootstrap methods: another look at the jackknife. *Annals of Statistics*, 7:1–26, 1979. 2

[9] R. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7 Part II:179–188, 1936. 2

[10] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995. 2

[11] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proc. of the 13th Int. Conf.*, pages 148–156, 1996. 2

[12] J. Friedman. Regularized discriminant analysis. *J. Am. Statistical Assoc.*, 84(405):165–175, 1989. 1, 6

[13] H. Gao and J. Davis. Sampling representative examples for dimensionality reduction and recognition - Bootstrap Bumping LDA. In *European Conference on Computer Vision*, Graz, Austria, May 7-13 2006. 1, 2, 3, 5, 6

[14] H. Gao and J. Davis. Why Direct LDA is not equivalent to LDA. *to appear in Pattern Recognition*, 2006. 2

[15] P. Huang, C. Harris, and M. Nixon. Human gait recognition in canonical space using temporal templates. In *Proc. Vis. Image Signal Process.*, pages 93–100. IEE, 1999. 1

[16] A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: a review. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 22(1):4–37, 2000. 1

[17] W. Krzanowski, P. Jonathan, W. McCarthy, and M. Thomas. Discriminant analysis with singular covariance matrices:methods and applications to spectroscopic data. *Applied Statistics*, 44:101–115, 1995. 1, 2, 3, 6

[18] C. Liu and H. Wechsler. Enhanced Fisher linear discriminant models for face recognition. In *Proc. Int. Conf. Pat. Rec.*, pages 1368–1372. IEEE, 1998. 1, 2, 3, 6

[19] X. Liu, A. Srivastava, and K. Gallivan. Optimal linear representations of images for object recognition. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 26(5):662–666, 2004. 2

[20] X. Lu and A. K. Jain. Resampling for face recognition. In *Int. Conf. on Audio and Video Based Biometric Person Auth.*, pages 869–877, 2003. 3

[21] C. Rao. The utilization of multiple measurements in problems of biological classification. *J. Royal Statistical Soc., B*, 10:159–203, 1948. 2

[22] R.Gross and J. Shi. The CMU motion of body (MoBo) database. Technical Report cmu-ri-tr-01-18, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 2001. 5

[23] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *2nd IEEE Workshop on Applications of Computer Vision*, Dec. 1994. 4

[24] R. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990. 2

[25] M. Skurichina and R. Duin. Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis & Applications*, 5:121–135, 2002. 3

[26] R. Tibshirani and K. Knight. Model search by bootstrap “bumping”. *J. of Computational and Graphical Statistics*, 8(4):671–686, 1999. 1, 2, 3

[27] F. Torre and T. Kanade. Oriented discriminant analysis (ODA). In *Brit. Mach. Vis. Conf.*, pages 132–141, 2004. 2

[28] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc. Int. Conf. Comp. Vis.*, pages 734–741, 2003. 2

[29] H. Yu and J. Yang. A direct LDA algorithm for high-dimensional data - with application to face recognition. *Pattern Recognition*, 34:2067–2070, 2001. 1, 2, 6