SPLITTING FACTOR ANALYSIS AND MULTI-CLASS BOOSTING

Xiuwen Liu

Department of Computer Science
Florida State University
Tallahasse, FL 32306-4530 USA

Washington Mio

Department of Mathematics
Florida State University
Tallahassee, FL 32306-4510 USA

ABSTRACT

We develop *Splitting Factor Analysis* (SFA), a novel linear model selection technique for dimension reduction that seeks to optimize the discriminative ability of the nearest neighbor classifier for data classification and labeling. We also discuss methodology for data kernelization that can be used in conjunction with any model selection technique. Applied to SFA, it leads to KSFA, a powerful new technique for the analysis of datasets with essential nonlinearities underlying their structures. For computational efficiency in the analysis of large datasets, we combine weak KSFA classifiers with multi-class boosting techniques. Several applications to image-based classification are discussed.

Index Terms— Factor analysis, kernel methods, machine learning, model selection

1. INTRODUCTION

The development of model and feature selection techniques to address data classification and labeling problems using information contained in training sets is of fundamental importance in machine learning and data mining, in particular, to applications in image processing and analysis. The classification performance and generalization ability of a proposed model are key elements to be assessed and optimized during a model selection process. The computational feasibility of both model and selection process are also considerations of basic importance. In particular, the investigation of dimension reduction in data representation is a natural companion problem to effective model selection. For large-scale and real-time applications, the development of mechanisms for rapid feature extraction and fast decisions is also essential.

The main goals of this paper are to develop: (i) a novel linear method, referred to as *Splitting Factor Analysis* (SFA), for simultaneous dimension reduction and optimal feature selection for data classification and labeling based on the K-nearest-neighbor (KNN) classifier; (ii) a general data kernelization strategy designed to be used in conjunction with any model selection technique. If the input data is represented by feature vectors in $\mathbb{R}^m$ and $k < m$, the goal of SFA is to find a linear transformation $A\colon \mathbb{R}^m \to \mathbb{R}^k$ that optimizes the discriminative ability of the KNN classifier on the transformed data; in typical applications, k is very small relative to m. The idea is that the optimal mapping A will reduce dimension and restructure the data so that it becomes more amenable to classification. Applied to SFA, the kernelization methodology yields *Kernel Splitting Factor Analysis* (KSFA), which has the ability to cope with nonlinearity in data structure.

To address scalability issues, we present a discussion on the use of multi-class boosting techniques [1] to construct an effective classifier by merging weak SFA or KSFA-based classifiers constructed by appropriately sampling the training set so as to sequentially enhance the classification performance. Several experiments are carried out with SFA, KSFA, and Boost-KSFA and classification results are compared with those obtained using other methods.

2. SPLITTING FACTOR ANALYSIS

We introduce *Splitting Factor Analysis* (SFA), a linear feature selection technique whose goal is to find a linear transformation that reduces the dimension of data representation while optimizing the predictive ability of the K-nearest neighbor (KNN) classifier as measured by its performance on given training data. We assume that a given ensemble of data in Euclidean space $\mathbb{R}^m$ is divided into *training* and *cross-validation* sets, each consisting of labeled representatives from P different classes of objects. For an integer c, $1 \leq c \leq P$, we denote by $x_{c,1}, \ldots, x_{c,t_c}$ and $y_{c,1}, \ldots, y_{c,v_c}$ the training and cross-validation elements, resp., that belong to class c.

If $A\colon \mathbb{R}^m \to \mathbb{R}^k$ is a linear transformation and $x, y \in \mathbb{R}^m$, we let $d(x, y; A) = \|Ax - Ay\|$ denote the distance between the transformed points Ax and Ay. The quantity

$$\rho(y_{c,i}; A) = \frac{\min_{c \neq b, j} d^p(y_{c,i}, x_{b,j}; A)}{\min_j d^p(y_{c,i}, x_{c,j}; A) + \epsilon} \tag{1}$$

provides a measurement of how well the *nearest-neighbor classifier* applied to the transformed data identifies the cross-validation element $y_{c,i}$ as belonging to class c. Here, $\epsilon > 0$ is

This work was supported in part by NSF grants CCF-0514743 and IIS-0307998, and ARO grant W911NF-04-01-0268.

a small number used to prevent vanishing denominators and $p > 0$ is an exponent that can be adjusted to regularize ρ in different ways. A large value $\rho(y_{c,i}; A)$ indicates that, after the transformation A is applied, $y_{c,i}$ lies much closer to a training sample of the class it belongs than to those of other classes; $\rho(y_{c,i}; A) \approx 1$ indicates a transition between correct and incorrect decisions by the nearest neighbor classifier. The function ρ was used in the development of OCA with $p = 1$ [2]. Note that expression (1) can be easily modified to reflect the performance of the *KNN* classifier.

The idea is to choose a transformation A that maximizes the average value of $\rho(y_{c,i}; A)$ over the cross-validation set. To control bias with respect to particular classes, we scale $\rho(y_{c,i}; A)$ with a sigmoid of the form $\sigma(x) = 1/(1 + e^{-\beta x})$ before taking the average. We identify linear maps $A\colon \mathbb{R}^m \to \mathbb{R}^k$ with $k \times m$ matrices and define a performance function $F\colon \mathbb{R}^{k\times m} \to \mathbb{R}$ by

$$F(A) = \frac{1}{P}\sum_{c=1}^{P}\left(\frac{1}{v_c}\sum_{i=1}^{v_c}\sigma\left(\rho(y_{c,i}; A) - 1\right)\right). \tag{2}$$

For a given A, the limit value of $F(A)$, as $\beta \to \infty$ and $\epsilon \to 0$, is the recognition performance of the nearest neighbor classifier applied to the transformed data.

Scaling an entire dataset does not change decisions based on the nearest neighbor classifier. This is reflected in the fact that F is (nearly) scale invariant; that is, $F(A) \approx F(rA)$, for $r > 0$. Equality does not hold if $\epsilon \neq 0$, but in practice, ϵ is negligible. Thus, we restrict F to transformations of unit norm. Let

$$\mathbb{S} = \left\{A \in \mathbb{R}^{k\times m} : \|A\|^2 = \operatorname{tr}(AA^T) = 1\right\}$$

be the unit sphere in $\mathbb{R}^{k\times m}$. The goal of splitting factor analysis is to maximize the performance function F over $\mathbb{S}$; that is, to find $\hat{A} = \operatorname{argmax} F(A)$.

The existence of a maximum of F is guaranteed by the simple facts that the sphere $\mathbb{S}$ is a compact space and F is continuous. This is in contrast with *Neighborhood Component Analysis* (NCA), developed in [3], where no such assurance can be provided.

Due to the existence of multiple local maxima of F, the numerical estimation of $\hat{A}$ is carried out with a stochastic gradient search. We omit the details since the search strategy is similar to that employed in OCA [2], but much simpler since the search is performed over a sphere instead of a Grassmann manifold.

3. KERNEL METHODS

Kernel methods are commonly used as a strategy to account for nonlinearity in data structure. For data represented by feature vectors $x_1, \ldots, x_M$ in $\mathbb{R}^n$, instead of developing models and classifiers based directly on the given feature vectors, one maps the entire ensemble into a Hilbert space $\mathbb{H}$ using a nonlinear map $\Phi\colon \mathbb{R}^n \to \mathbb{H}$, and then develops models based on the kernelized data $\Phi(x_1), \ldots, \Phi(x_M)$. The typical assumption is that Φ is not known explicitly, only the kernel function

$$k(x, y) = \langle \Phi(x), \Phi(y) \rangle,$$

where $\langle\, ,\, \rangle$ denotes the inner product in $\mathbb{H}$. This means that explicit knowledge of $\Phi(x_1), \ldots, \Phi(x_M)$ is not assumed, only the inner products $k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$.

Kernel methods are frequently studied targeting one specific model at a time. In this paper, we fully decouple data kernelization from the model. Let

$$V = \operatorname{span}\{\Phi(x_1), \ldots, \Phi(x_M)\} \subseteq \mathbb{H}. \tag{3}$$

As in [4], we first argue that one can introduce an orthonormal coordinate system in V and calculate the coordinates of the orthogonal projection onto V of any vector of the form $\Phi(x)$, $x \in \mathbb{R}^n$, only using the kernel k. Once such coordinate system is available, the coordinate vectors yield a new representation of the data as points in $\mathbb{R}^m$, $m = \dim V$, to which we can apply any model or feature selection technique, in particular, splitting factor analysis to obtain *Kernel Splitting Factor Analysis* (KSFA).

3.1. A Coordinate System in V

Each $a = (a_1, \ldots, a_M)^T \in \mathbb{R}^{M\times 1}$ defines a vector $v \in V$ given by $v = \sum_{i=1}^{M} a_i \Phi(x_i)$. Form the $M \times M$ symmetric Gram matrix K, whose entries are $K_{ij} = k(x_i, x_j)$. If $a, b \in \mathbb{R}^{M\times 1}$ represent $v, w \in V$, then

$$\langle v, w \rangle = a^T K b. \tag{4}$$

To find the a-coordinates of an orthonormal basis of V, first diagonalize the Gram matrix K, and let $\{\eta_1, \ldots, \eta_m\} \subset \mathbb{R}^M$ be an orthonormal set associated with the nonzero eigenvalues $\lambda_1 \geq \ldots \geq \lambda_m$ of K, where $m = \dim V = \operatorname{rank} K$. For each $1 \leq j \leq m$, write the vector $\eta_j/\sqrt{\lambda_j}$ as $\eta_j/\lambda_j = (\alpha_{1j}, \ldots, \alpha_{Mj})^T$ and let $v_j = \sum_{i=1}^{M} \alpha_{ij}\Phi(x_i)$. It follows from Eqn. 4 that $\mathcal{B} = \{v_1, \ldots, v_m\}$ is an orthonormal basis of V.

3.2. Kernel Representation

For $x \in \mathbb{R}^n$, let $\Phi_V(x) \in V$ be the orthogonal projection of $\Phi(x)$ onto V. The inner product $\langle \Phi_V(x), v_j \rangle = \langle \Phi(x), v_j \rangle$ can be calculated as

$$\langle \Phi(x), v_j \rangle = \sum_{i=1}^{M} \alpha_{ij} \langle \Phi(x), \Phi(x_i) \rangle = \sum_{i=1}^{M} \alpha_{ij} k(x, x_i).$$

Thus, the coordinate vector of $\Phi_V(x)$ with respect to the orthonormal basis $\mathcal{B}$ is

$$[\Phi_V(x)]_{\mathcal{B}} = \sum_{i=1}^{M} \begin{bmatrix} \alpha_{i1} k(x, x_i) \\ \vdots \\ \alpha_{im} k(x, x_i) \end{bmatrix} = \alpha^T \begin{bmatrix} k(x, x_1) \\ \vdots \\ k(x, x_M) \end{bmatrix}.$$

Table 1. Recognition performance of different representations on the full 40-class ORL dataset

Set	Nearest Neighbor (%)	3-Nearest Neighbor (%)
Original 10,304-D feature space		
Cross validation	96.7	71.7
Test	94.2	73.3
KPCA with $k = 10$		
Cross validation	91.7	78.3
Test	95.8	74.2
KSFA with $k = 10$		
Cross validation	100.0	100.0
Test	**99.17**	**98.3**

Thus, the projected and kernelized data set $x_1, \ldots, x_M$ is represented by their coordinate vectors $y_i = [\Phi_V(x_i)]_{\mathcal{B}} \in \mathbb{R}^m$.

4. BOOSTING WEAK KSFA CLASSIFIERS

KSFA provides a powerful new dimension reduction and model selection tool. The results of several data classification experiments in the context of image analysis and comparisons with other methods are provided in the next section. A drawback of the general kernelization techniques of Sec. 3 is scalability – as the training set becomes large, learning optimal features for classification and labeling might become computationally very costly. To address this problem, we propose to use a series of weak KSFA classifiers in conjunction with *Stagewise Additive Modeling using a Multi-class Exponential loss function* (SAMME), a multi-class boosting procedure developed in [1]. This will lead to an effective and computationally efficient classifier, which we refer to as Boost-KSFA, or simply, β-KSFA. Using the methodology introduced in Secs. 2 and 3, a weak learner constructs KSFA classifiers sequentially using subsets of the training set chosen according to the sampling techniques of [5, 1], which emphasize "the hard cases" at each stage of the construction. Details of the SAMME algorithm can be found in [1].

5. EXPERIMENTAL RESULTS

We demonstrate the effectiveness of the algorithms introduced in this paper on two datasets: (i) the ORL face dataset[1], which consists of 40 classes with 10 images in each class; (ii) a dataset of handwritten digits, which contains ten classes with an average of 729 training samples per class and a separate test set with 2009 samples. In the first experiment, we divided each class in the ORL dataset into training, cross-validation, and test sets consisting of 4, 3, and 3 images, respectively and

[1]http://www.uk.research.att.com/facedatabase.html

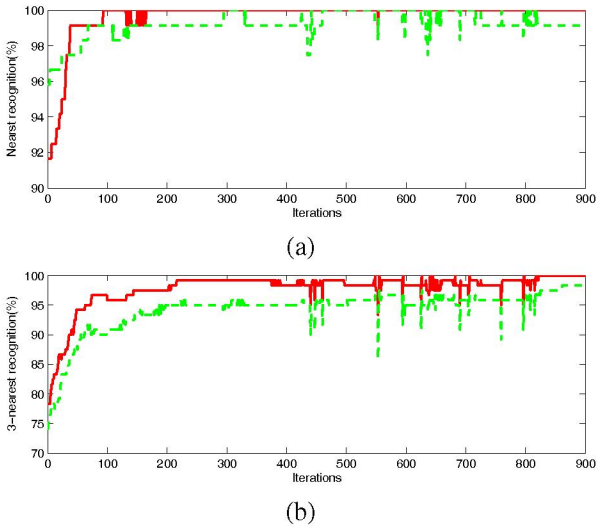

Fig. 1. Recognition performance of KSFA with a Gaussian kernel versus the number of iterations on the ORL dataset. In each panel, solid and dashed lines correspond to performance on the cross-validation and test sets, respectively. (a) Performance with the nearest neighbor classifier. (b) Performance with the 3-nearest neighbor classifier.

Table 2. Comparison of classification accuracy (%) applying different methods to the ORL data set.

PCA	97.25	β-KSFA (Polynomial)	98.5
QR/LDA	98.25	β-KSFA (Gaussian)	99.0
PCA+LDA	95.00	β-SFA	98.0

used a Gaussian kernel with $k = 10$. The results obtained and comparisons with KPCA are reported in Table 1 and plots of recognition performance are shown in Fig. 1.

In the second experiment, we divided the ORL dataset into disjoint training and test sets with 200 images in each. In the experiments, we use a Gaussian kernel with $k = 8$. Fig. 2(a) shows the recognition performance on the entire training set and on the test set of 25 different weak KSFA classifiers. As expected, the individual weak classifiers do not perform well, but the β-KSFA classifier improves the performance on both training and test sets dramatically. Fig. 2(b) shows the performance of the strong classifier versus the number of weak classifiers used. To further demonstrate the effectiveness of the proposed methods, we compare the results with those obtained with PCA, LDA (linear discriminant analysis), and QR/LDA [6] used in conjunction with the nearest neighbor classifier. The results are shown in Table 2. With an appropriate choice of kernels, the present methods yield the best performance.

To demonstrate the scalability of β-KSFA, we apply it to a handwritten digit recognition dataset. Fig. 3 shows the results on the training and the test sets using a Gaussian kernel.

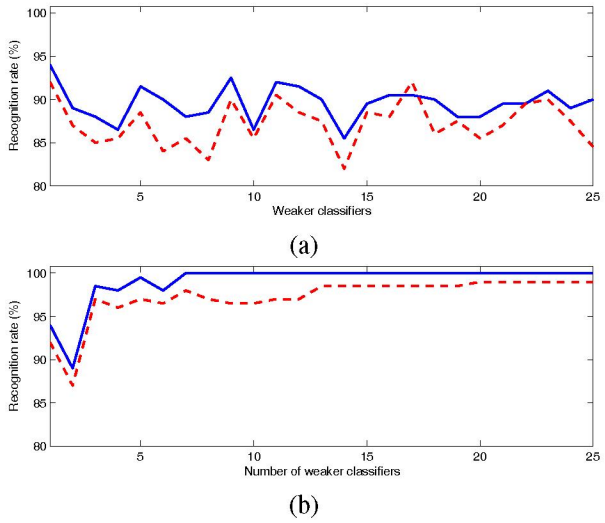

Fig. 2. Recognition performance with a Gaussian kernel on disjoint ORL training and test sets: (a) performance of various weak KSFA classifiers labeled 1–25; (b) performance of a β-KSFA classifier versus the number of weak classifiers used. Solid and dashed lines correspond to performance on the training and test sets, respectively.

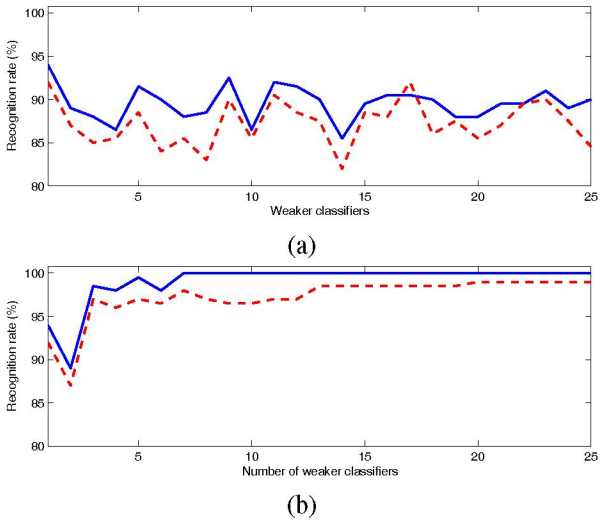

Fig. 3. Recognition performance with a Gaussian kernel on USPS handwritten digit training and test sets: (a) performance of 25 weak KSFA classifiers; (b) performance of a β-KSFA classifier versus the number of weak classifiers used. Solid and dashed lines correspond to performance on the training and test sets, respectively.

A recognition performance of 94.2% was achieved on the test set. The result is comparable to those obtained with some other methods [7]. The highest performance reported on this dataset is 97.5% [7]. Note, however, that the best result is obtained using a tangent vector distance, which can be incorporated to our methods as well.

6. SUMMARY

We developed a new linear feature selection technique termed *Splitting Factor Analysis* (SFA) that optimizes the ability of the K-nearest neighbor classifier to discriminate data while performing dimension reduction. A general data kernelization procedure was adopted that decouples data kernelization from classifiers. Combined with SFA, the technique yields KSFA, a kernel analogue of SFA that can cope with nonlinearities in data structure. For scalability and computational efficiency, weak KSFA classifiers were used in conjunction with multi-class boosting techniques to produce β-KSFA, a novel model selection method for data classification and labeling. Results of several experiments were reported and recognition performance was compared to those obtained using various different methods.

7. REFERENCES

[1] J. Zhu, S. Rosset, H. Zou, and T. Hastie, "A multi-class forward stagewise generalization of AdaBoost," Technical report, University of Michigan, Department of Statistics, 2005.

[2] X. Liu, A. Srivastava, and K. Gallivan, "Optimal linear representations of images for object recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, pp. 662–666, 2004.

[3] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighborhood component analysis," in *Advances in Neural Information Processing Systems*, L. K. Saul, Y. Weiss, and L. Bottou, Eds., Cambridge, MA, 2005, vol. 17, pp. 513–520, MIT Press.

[4] B. Schölkopf, A. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[5] Y. Freund and R. Schapire, "A decision theoretic generalization of online learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[6] J. Ye and Q. Li, "A two-stage linear discriminant analysis via QR-decomposition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 929–941, 2005.

[7] D. Keysers, W. Macherey, H. Ney, and J. Dahmen, "Adaptation in statistical pattern recognition using tangent vectors," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 269–274, 2004.