Original Article
International Journal of Fuzzy Logic and Intelligent Systems
Vol. 14, No. 3, September 2014, pp. 162-170
http://dx.doi.org/10.5391/IJFIS.2014.14.3.162

ISSN(Print) 1598-2645
ISSN(Online) 2093-744X

Greedy Learning of Sparse Eigenfaces for Face Recognition and Tracking

Minyoung Kim
Department of Electronics and IT Media Engineering, Seoul National University of Science and Technology, Seoul 139-743, Korea

Ijfis

Abstract

Appearance-based subspace models such as eigenfaces have been widely recognized as one of the most successful approaches to face recognition and tracking. The success of eigenfaces mainly has its origins in the benefits offered by principal component analysis (PCA), the representational power of the underlying generative process for high-dimensional noisy facial image data. The sparse extension of PCA (SPCA) has recently received significant attention in the research community. SPCA functions by imposing sparseness constraints on the eigenvectors, a technique that has been shown to yield more robust solutions in many applications. However, when SPCA is applied to facial images, the time and space complexity of PCA learning becomes a critical issue (e.g., real-time tracking). In this paper, we propose a very fast and scalable greedy forward selection algorithm for SPCA. Unlike a recent semidefinite program-relaxation method that suffers from complex optimization, our approach can process several thousands of data dimensions in reasonable time with little accuracy loss. The effectiveness of our proposed method was demonstrated on real-world face recognition and tracking datasets.

Keywords: Sparse learning, Principal component analysis, Face recognition/tracking

Received: Jun. 17, 2014
Revised : Sep. 19, 2014
Accepted: Sep. 20, 2014

Correspondence to: Minyoung Kim
(mikim@seoultech.ac.kr)

1. Introduction

Principal component analysis (PCA) is one of the most popular techniques for high-dimensional data analysis and dimensionality reduction [1–3]. The PCA formulation can be obtained from factor analysis when a small set of projection vectors is sought to preserve the variation in data as much as possible. This effectively reduces the problem to that of an eigenvalue/vector for the data covariance matrix (see Section 2.1 for details).

One of the most successful applications of PCA is the use of *eigenfaces* in computer vision and neuroscience, an appearance-based subspace model for facial image data [4–7]. It is widely believed that although facial images are high-dimensional, they lie on a low-dimensional manifold or subspace. In addition, the eigenfaces, referring to the PCA eigenvectors for facial image data, are likely to capture salient features from data (e.g., inter-subject appearance differences) while suppressing noise in images (e.g., illumination effects or occlusion). This results in superior performance for face recognition [6, 7], as well as in face tracking tasks [8], in which case the eigenface model is updated incrementally in accordance with changes in the appearance of the target (e.g., pose or illumination changes).

Despite the successful use of PCA in many data analysis problems, one known drawback of PCA is that nearly all data dimensions can potentially contribute to principal components, whereas many real-world applications including facial images are of a sparse nature (i.e., only a few dimensions are actually effective). To address this problem, the sparse extension of PCA (SPCA) was recently developed, and it has since received significant attention in the research community [9–11]. SPCA imposes sparseness constraints on the eigenvectors, which has been shown to yield more robust solutions in many applications.

However, most existing SPCA approaches are considerably slow, difficult to optimize, and do not scale gracefully with data dimension. In [10], for instance, an L1- and L2-penalized regression-like optimization was formulated for SPCA learning that requires repetitive solving of the iterative optimization problem until it converges. In [11], the original problem is relaxed to a semidefinite program (SDP) that can be solved by off-the-shelf SDP solvers; however, most SDP solvers are relatively slow, and are not even capable of processing several hundreds of data dimensions.

The facial image data are high-dimensional, and the related tasks (e.g., real-time tracking) require near real-time, fast learning of PCA loadings. Application of SPCA to facial images requires the time and space complexity to be addressed. In this paper, we propose a very fast and scalable greedy forward selection algorithm for SPCA. Unlike existing methods that suffer from complex optimization, our approach is able to process several thousands of data dimensions in reasonable time with little accuracy loss.

Our approach involves maintaining an active list (initially empty) of non-zero eigenvector entries, and the use of a greedy fashion to select a new non-zero entry that yields the maximal improvement of the PCA objective function at each stage. Once a new entry is chosen, the eigenvector is refined by solving the eigenvalue/vector problem on the current active list. Note that as the active list size is usually small (typically no more than several hundreds), the refinement (finding an eigenvector) can be performed very fast.

By using facial image data, it is demonstrated that the learned sparse eigenfaces have non-zero loadings that correspond to the most important feature points of the face (e.g., eyes/eyebrows, nose). As these are regarded as the most salient features for face recognition, our sparse eigenface learning algorithm has the ability to discover a few of these most discriminative features very effectively. Our experiment also demonstrates a significant improvement in face recognition accuracy compared to standard (full) PCA.

In addition to face recognition, the proposed SPCA learning algorithm was also applied to the visual face tracking problem. One crucial issue in tracking is to establish how to update the target appearance model effectively and efficiently in accordance with the actual target that usually changes over time. Similar to increment visual tracking (IVT) [8], new eigenfaces are learnt based on the accumulated historic tracked face images. It is shown that, under regular conditions, our SPCA learning is only five times slower than IVT's sophisticated incremental SVD algorithm for full PCA, that is capable of handling approximately 20–30 frames per second in standard computing environments, which is near real-time.

The paper is organized as follows: After briefly reviewing PCA and related work in Section 2, the proposed greedy SPCA learning algorithm is described in Section 3 together with its application to face tracking. The effectiveness of the proposed method is demonstrated on selected real-world face recognition and tracking datasets in Section 4. The conclusion appears in Section 5.

2. Background and Previous Work

In this section, PCA and the formulation of its sparse extension are briefly reviewed. This is followed by a description of the application of the recent SDP relaxation approach [11] to the SPCA problem.

2.1 Principal Component Analysis

PCA was initially proposed by Pearson in 1901 [1], while a more general procedure was subsequently developed [2]. PCA is based on finding a small set of projection vectors that preserves the variation in data as much as possible. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \Re^{p \times n}$ be a data matrix comprised of n data samples of dimension p in the columns and let $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$ be the sample mean of the data.

PCA then aims to find a subspace $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_q]$ with q basis vectors $\mathbf{b}_j \in \mathbb{R}^p$ that best reconstructs the data covariance $\boldsymbol{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top$). The typical assumption is that $q \ll p$ (i.e., low-dimensional subspace). This can be formulated as follows.

$$\min_{\mathbf{B},\boldsymbol{\Lambda}} \left\| \boldsymbol{\Sigma} - \sum_{j=1}^{q} \lambda_j \mathbf{b}_j \mathbf{b}_j^\top \right\|_F^2, \text{ s.t. } ||\mathbf{b}_j||^2 = 1,\ \lambda_j \geq 0,\ \forall j, \quad (1)$$

where $||\cdot||_F$ is the Frobenius norm, and $\boldsymbol{\Lambda}$ is a diagonal matrix with entries $\{\lambda_j\}$ indicating the impact/weight of j-th basis in reconstruction. One can further impose the usual orthogonal constraints of $\mathbf{b}_i^\top \mathbf{b}_j = 0$ for all $i \neq j$.

The optimization (1) can be performed in a recursive manner: 1) solving the rank-one approximation problem (2) to have $(\lambda, \mathbf{b})$,

$$\min_{\mathbf{b},\lambda} \left\| \mathbf{\Sigma} - \lambda \mathbf{b}\mathbf{b}^\top \right\|_F^2, \text{ s.t. } ||\mathbf{b}||^2 = 1, \ \lambda \geq 0, \tag{2}$$

and 2) compute the residual and set it as a new covariance, $\mathbf{\Sigma} \leftarrow \mathbf{\Sigma} - \lambda \mathbf{b}\mathbf{b}^\top$. This procedure is repeated until q eigenvectors are obtained. Notice that the optimal solution $\mathbf{b}$ of (2) is the eigenvector of $\mathbf{\Sigma}$ corresponding to the largest eigenvalue. This can easily be shown by: minimizing (2) over λ first yields $\lambda = \mathbf{b}^\top \mathbf{\Sigma} \mathbf{b}$ as an optimal solution, and the resulting problem becomes the famous Rayleigh quotient problem (3) that has the largest eigenvector as a solution.

$$\max_{\mathbf{b}} \mathbf{b}^\top \mathbf{\Sigma} \mathbf{b}, \text{ s.t. } ||\mathbf{b}||^2 = 1. \tag{3}$$

2.2 Sparse PCA

SPCA additionally imposes the sparseness constraints on the eigenvectors. Formally, letting $\text{Card}(\mathbf{u})$ be the number of non-zero entries of vector $\mathbf{u}$, the SPCA can be formulated as follows.

$$\max_{\mathbf{b}} \mathbf{b}^\top \mathbf{\Sigma} \mathbf{b}, \text{ s.t. } ||\mathbf{b}||^2 = 1, \ \text{Card}(\mathbf{b}) \leq R, \tag{4}$$

where R is a sparsity level constant assumed to be known a priori. The sparse eigenvectors obtained from the sparseness constraints can be beneficial in several aspects, for instance, many real-world data have sparse relationships with underlying factors (e.g., gene interaction in biological models), and the resulting sparse basis vectors are more robust to noise/outliers by focusing on a few of the most informative dimensions.

However, solving (4) incurs additional difficulty due to the non-convex, non-differentiable cardinality constraints. Several approximate approaches have recently been proposed. The two most popular methods are: 1) iteratively optimizing the L1- and L2-regularized regression problem (also referred to as the elastic-net problem) [10], and 2) relaxing the original problem into a tractable SDP (hence, a convex problem) [11]. Here, we briefly summarize the latter approach.

In [11], a new semidefinite rank-one matrix variable $\mathbf{X} = \mathbf{b}\mathbf{b}^\top$ is introduced. Removing the rank-one constraint and considering that it is possible to further relax $||\mathbf{b}||^2 = 1$ as the L1-norm (the sum of the absolute entries of $\mathbf{X}$) constraint $||\mathbf{X}||_1 \leq R$, a relaxed problem can be formulated:

$$\max_{\mathbf{X} \succeq 0} \text{Trace}(\mathbf{\Sigma}\mathbf{X}), \text{ s.t. } \text{Trace}(\mathbf{X}) = 1, \ ||\mathbf{X}||_1 \leq R. \tag{5}$$

This is an instance of an SDP [12]. Any off-the-shelf SDP solvers (e.g., [13, 14]) can be utilized; however, most of the current packages simply fail to run when the data dimension p is large, as is the case with the facial image data. In the next section, our fast and scalable SPCA learning algorithm is introduced.

3. Greedy Learning of Sparse Eigenfaces

The original SPCA problem (4) is directly attempted. The approach that is followed is to add a non-zero entry to $\mathbf{b}$ one at a time in a greedy fashion. In particular, we maintain an active list I of non-zero entry indices in $\mathbf{b}$. As the objective for the current $\mathbf{b}$ is $\mathbf{b}^\top \mathbf{\Sigma} \mathbf{b}$, the gradient direction that yields maximal improvement in the objective is $\mathbf{\Sigma}\mathbf{b}$. If only one dimension is selected to turn on from zero to non-zero (i.e., the steepest coordinate ascent), it should be the largest entry of the gradient vector, namely $j^* = \arg\max_{j \notin I} (\mathbf{\Sigma}\mathbf{b})_j$, whereupon j^* is added to I. Furthermore, the current non-zero entries of $\mathbf{b}$ can be refined to yield a more desirable objective; the best solution for the current I would be to set $\mathbf{b}_I$ (the non-zero parts of $\mathbf{b}$) to the largest eigenvector of $\mathbf{\Sigma}_{I,I}$ (the sub-matrix of $\mathbf{\Sigma}$ by using the row/column indices in I). Note that this can be obtained very fast compared to finding the eigenvector of $\mathbf{\Sigma}$ since I usually has a small cardinality (no greater than R). This procedure is repeated while $|I| \leq R$. The overall algorithm is summarized below.

1) **(Initialization)** Start with $\mathbf{b} = 0$ and $I = \phi$. Then, find the j such that $\mathbf{b} = \mathbf{e}_j$ maximizes the objective where $\mathbf{e}_j$ has zero entries except when the j-th entry is equal to 1. As the objective is $\mathbf{e}_j^\top \mathbf{\Sigma} \mathbf{e}_j = \mathbf{\Sigma}_{j,j}$, j is the index of the largest diagonal entry of $\mathbf{\Sigma}$. Set $I = \{j\}$.
2) **(Steepest coordinate ascent)** For the current $\mathbf{b}$ and I, find $j^* \notin I$ such that the coordinate ascent gradient $(\mathbf{\Sigma}\mathbf{b})_j$ is maximized. That is, $j^* = \arg\max_{j \notin I} (\mathbf{\Sigma}\mathbf{b})_j$. Then, set $I = I \cup \{j^*\}$.
3) **(Eigenvector refinement)** Compute $\mathbf{b}_I$ as the largest eigenvector of $\mathbf{\Sigma}_{I,I}$, the sub-matrix of $\mathbf{\Sigma}$ by taking row/column indices in I. Copy $\mathbf{b}_I$ into the indices I in $\mathbf{b}$ and set all the other entries of $\mathbf{b}$ to 0.
4) **(Cardinality check)** Go to step 2 if $|I| < R$ and the

objective $\mathbf{b}^\top \mathbf{\Sigma} \mathbf{b}$ is improved from the previous stage. Otherwise, stop.

Note that the algorithm repeats at most R stages, and that each stage only requires a simple maximum number search over $(p-|I|)$ entries, and finding the largest eigenvector of the small $(|I| \times |I|)$ matrix. This results in highly accelerated SPCA learning. It is worth noting that a similar forward selection type SPCA algorithm has been proposed [15]. However, new entries are selected by solving $(p-|I|)$ different eigenvalue problems, which may require a prohibitively long time for high-dimensional facial image data.

3.1 Face Tracking with Adaptive Sparse Eigenfaces

We also consider applying the sparse eigenface model to the visual face tracking problem. Tracking is an online estimation problem where given the sequence of image frames up to the current time t, denoted by $F_0, \ldots, F_t$, the location (and also the size and the in-plane rotation angle of the tight bounding box around face) of the face image is estimated in F_t. One way to formalize the problem is to use the online temporal filtering formulation [16], that involves the estimation of $P(u_t|F_t)$ at each time $t = 1, 2, \ldots$. Here u_t is the tracking state specifying the face bounding box with respect to the current frame F_t. Typically, $u_t = [c_x, c_y, \rho, \phi]^\top$ where the first two values are the center position, ρ is the relative size of the bounding box (1 for the reference patch size), and ϕ is the rotation angle. It is straightforward to extract the face image $\mathbf{h}_t$ from u_t and F_t by a simple affine warping $\mathbf{h}_t = \omega(u_t, F_t)$.

The underlying probabilistic model is the Markov chain over tracking states in which each state is related to an emission model that measures the goodness of track (the extent to which $\mathbf{h}_t = \omega(u_t, F_t)$ resembles a face). A 1st-order Gaussian random-walk model for state dynamics, namely $P(u_t|u_{t-1}) = \mathcal{N}(u_t; u_{t-1}, V_0)$, with some covariance V_0 is common. The crucial part is the emission model, for which a generic energy model,

$$P(F_t|u_t) \propto \exp(-E(\mathbf{h}_t;\theta)/\sigma_0^2), \text{ where } \mathbf{h}_t = \omega(u_t, F_t), \tag{6}$$

can be considered, where $E(\mathbf{h}_t;\theta)$ is the energy function that has a lower (higher) value when $\mathbf{h}_t$ more (less) closely resembles the target face model θ. The online filtering can then be written as the following recursion.

$$P(u_t|F_{0\ldots t}) \propto P(F_t|u_t) \cdot \int P(u_t|u_{t-1}) \cdot P(u_{t-1}|F_{0\ldots t-1})\, du_{t-1}, \tag{7}$$

which can be typically solved by using a sampling-based method (e.g., particle filtering [16]).

The eigenface model can be employed as a target face model. As it is possible for the appearance of a face to change over time (mainly due to changes in the facial pose/expression, illumination, or distance from the camera), a more sensible strategy would be to adaptively change the target model. In [8], the IVT builds an up-to-date eigenface model each time using the historically tracked facial image data. This can be performed considerably efficiently by using the sophisticated incremental SVD algorithm [17].

Although our proposed algorithm is not similarly incremental, it is possible to plug the proposed greedy SPCA learning algorithm directly into the adaptive emission model. More formally, to obtain the current data mean and covariance $(\boldsymbol{\mu}, \mathbf{\Sigma})$, once a new tracked result $\mathbf{h}_t$ is available, the mean and covariance are updated by using the following equations

$$\boldsymbol{\mu}^{new} = \frac{t}{t+1}\boldsymbol{\mu} + \frac{1}{t+1}\mathbf{h}_t, \tag{8}$$

$$\mathbf{\Sigma}^{new} = \frac{t}{t+1}\mathbf{\Sigma} + \frac{1}{(t+1)^2}(\mathbf{h}_t - \boldsymbol{\mu})(\mathbf{h}_t - \boldsymbol{\mu})^\top. \tag{9}$$

Using the newly updated covariance, the sparse eigenvectors are learnt in a greedy fashion as before. The emission model can optionally be scheduled to be updated in a stepwise manner (e.g., update every third frames). There is a trade-off regarding the choice of the update frequency: frequent updates would enable the model to adapt to appearance changes instantly, but at the cost of a delay in the tracking time, and vice versa. In practice, an update frequency of 3–5 steps is known to work well without causing a significant delay.

4. Experimental Results

The proposed SPCA learning algorithm was tested using face recognition and tracking problems, but first, it was demonstrated that our method is capable of efficiently and accurately recovering the underlying sparse eigenvectors for a specified synthetic dataset.

(a)
$$A=\begin{bmatrix} 0.7071 & 0 & 0 \\ 0 & 0 & 0.7071 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0.7071 \\ 0 & 0.7071 & 0 \\ 0 & 0.7071 & 0 \\ 0.7071 & 0 & 0 \end{bmatrix}$$

(b)
$$A^{Full}=\begin{bmatrix} 0.5966 & -0.1126 & -0.3625 \\ 0.2290 & -0.4317 & 0.5111 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.2290 & -0.4317 & 0.5111 \\ 0.3027 & 0.5486 & 0.3278 \\ 0.3027 & 0.5486 & 0.3277 \\ 0.5966 & -0.1126 & -0.3625 \end{bmatrix}$$

(c)
$$A^{SDP}=\begin{bmatrix} 0.7070 & 0 & 0 \\ 0 & 0 & 0.7071 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0.7071 \\ 0 & 0.7071 & 0 \\ 0 & 0.7071 & 0 \\ 0.7072 & 0 & 0 \end{bmatrix}$$

(d)
$$A^{Grd}=\begin{bmatrix} 0.7071 & 0 & 0 \\ 0 & 0 & 0.7071 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0.7071 \\ 0 & 0.7071 & 0 \\ 0 & 0.7071 & 0 \\ 0.7071 & 0 & 0 \end{bmatrix}$$

Figure 1. Synthetic sparse extension of principal component analysis (PCA) basis recovery. (a) True basis matrix, (b) estimated basis by standard PCA, (c) semidefinite program relaxation, (d) our greedy method.

4.1 Synthetic Data

A synthetic experimental setup was devised to demonstrate that our SPCA learning algorithm is much faster than the SDP relaxation method and that it is able to recover the underlying sparse subspace basis vectors accurately. Using the probabilistic extension of PCA [18] (referred to as PPCA) as motivation, observable data $\mathbf{x} \in \mathbb{R}^p$ was generated using an underlying latent low-dimensional vector $\mathbf{z} \in \mathbb{R}^q$ with $q \ll p$, based on the linear Gaussian-noise model, namely

$$\mathbf{x} = \mathbf{A}\mathbf{z} + \boldsymbol{\epsilon}, \tag{10}$$

where $\mathbf{A}$ is a $(p \times q)$ matrix that relates $\mathbf{z}$ to $\mathbf{x}$, and $\boldsymbol{\epsilon}$ represents independent Gaussian noise.

Specifically, we set $p = 10$, $q = 3$, and defined $\mathbf{A}$ to have sparse column basis vectors with only two non-zero entries (Figure1a). The noise variance was chosen as 10^{-4}. As in the PPCA process, the latent vectors $\mathbf{z}$ were regarded as independent and identical isotropic Gaussian random vectors with zero mean and identity covariance. The number of samples that was generated was $n = 1,000$ samples $\mathbf{x}$, which were used to estimate the covariance matrix from the samples.

The recovered basis results are shown in Figure 1. Three different methods were compared: the standard PCA that only finds the directions of the highest variations without considering the sparsity of the basis vectors (Figure 1b), the SPCA obtained by SDP relaxation [11] (Figure 1c), and our greedy method (Figure 1d). As shown, the full PCA recovers the right subspace, but is not able to find the correct PCA loadings. On the other hand, by exploiting the sparsity constraints of the basis vectors, the SPCA methods were able to discover a nearly perfect embedding matrix. Although not very significant, our greedy method yielded a slightly smaller error, with the average L2 difference between true and estimated eigenvectors being 3.94×10^{-6}, compared to that of the SDP relaxation, which was 7.24×10^{-5} (c.f., the full PCA has an L2 difference of 0.6578).

Finally, the actual running time was measured by using an SDP relaxation method, for which purpose the MATLAB SDP solver package [13, 14] was used. Our greedy SPCA learning algorithm was implemented in MATLAB running on a 2.0 GHz Dual Core machine, on which the SDP-based method required 3.85 s to run to completion, whereas our greedy method only required 0.03 s, approximately 100 times faster than a difficult SDP optimization. In the next section, it is shown that our approach is scalable for even higher ambient data dimension, which, in the case of image data, may number several thousands.

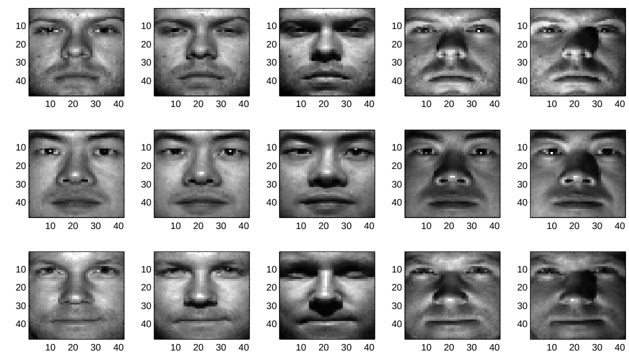


Figure 2. Sample face images for face recognition. Each row contains images of a single subject with appearance variations, mostly achieved by using diverse illuminations.

4.2 Face Recognition

This experiment was designed to test the degree of accuracy with which the proposed sparse eigenface learning method is able to recognize a face. The evaluation used the extended Yale-B dataset [19], which comprises approximately 2,400 frontal face images of a few dozen subjects. A 3-way classification was considered for the three chosen subjects by randomly selecting 15 images for each subject. The selected images displayed

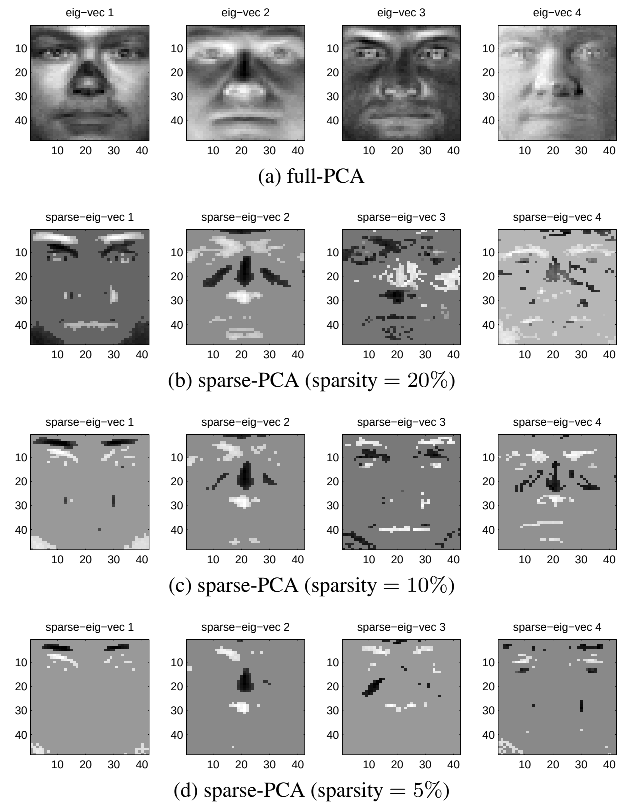


Figure 3. Learned eigenfaces for a full-principal component analysis (PCA) and the proposed sparse-PCA with different sparsity levels. In each sparse eigenvector image, the dominant gray pixels (of which the brightness may vary across different images) indicate 0 values, while darker/brighter pixels are interpreted as non-zero entries.

a variety of considerably diverse illuminations. Subsequent to applying proper sub-sampling, (48×42) tightly cropped face images were obtained (hence, the data dimension was $p = 2016$). Selected sample images are shown in Figure 2.

For the face recognition and eigenface learning task, the images were randomly divided into 10/5 training/test images for each subject, and this procedure was repeated for 10 random runs. The performance of our SPCA approach was contrasted against the standard PCA (denoted by full-PCA) that simply finds q largest eigenvectors. In contrast, as a PCA subspace dimension, our approach considered $q = 4$ and $q = 8$. It should be noted that the SDP relaxation based SPCA method simply failed to run as a result of memory and computational issues (Most of the current SDP solver packages (e.g., [13, 14]) are unable to deal with $p^2 \approx 4 \times 10^6$ variables.). First, the learned eigenvectors of the full-PCA and our sparse-PCA for $q = 4$ were visualized in Figure 3. In our sparse model, three different sparsity levels $r \in \{5\%, 10\%, 20\%\}$ were considered, in which the cardinality bound was specified as $R = floor(r \cdot p)$.

Table 1. Running time (in seconds) of the PCA subspace learning

	Full-PCA	SPCA (5%)	SPCA (10%)	SPCA (20%)
$q = 4$	0.18	1.02	2.41	7.52
$q = 8$	0.23	1.84	4.40	13.85

SPCA, sparse extension of principal component analysis.

Interestingly, the non-zero entries in the sparse eigenvectors mostly correspond to the points around eyes/eyebrows and nose, and it becomes more evident as the significance of the sparsity level is increased from 20% to 5%. Those non-zero points are typically considered the most important facial feature points that are used as salient features for face recognition. Our sparse eigenface learning algorithm thus discovers a few most discriminative features very effectively.

The superior performance of the proposed sparse eigenface learning of face recognition, was verified by conducting nearest neighbor (NN) classification on the learned PCA subspaces. Formally, the class label (i.e., subject ID) of the test data (image) $\mathbf{x}^*$ is determined as the class label of the training image $\mathbf{x}_j$ where $j = \arg\min_i ||\mathbf{z}^* - \mathbf{z}_i||_2$. Here, the minimization is performed across the entire range of training images, and $\mathbf{z}$ indicates the subspace coordinates for $\mathbf{x}$, namely $\mathbf{z} = \mathbf{B}^\top(\mathbf{x} - \boldsymbol{\mu})$ where $\mathbf{B}$ and $\boldsymbol{\mu}$ are the learned eigenvectors in the column and the training data mean, respectively. This minimization procedure was repeated for 10 random training/test folds, and the average test errors are reported in Figure 4.

First it can be seen that increasing the subspace dimension improves the prediction accuracy of both the full-PCA and sparse-PCA (for all sparsity levels). Interestingly, the NN classifiers on the SPCA subspaces are less sensitive to the chosen subspace dimension. Considering that a model with smaller dimensions is beneficial especially from a computational point of view, our SPCA method would be more useful for real-time applications and in situations with limited computing resources (e.g., real-time face tracking on mobile devices). Furthermore, for the same subspace dimension, the most aggressive 5%-level SPCA model achieved the smallest test errors (even the 4-dim sparse-PCA outperformed the 8-dim full-PCA).

The learning time was measured for the proposed greedy SPCA learning, using a 2.0 GHz Dual Core machine. As Table 1 demonstrates, the running time is quite fast and comparable to standard PCA methods especially for the most sparse model.

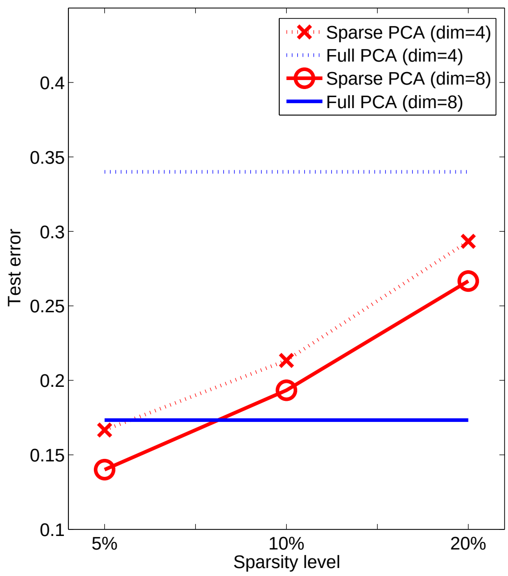


Figure 4. Face recognition results. The nearest neighbor classification was conducted on the full-principal component analysis (PCA) subspaces and the sparse-PCA subspaces with different sparsity levels. The average test errors are shown.

4.3 Face Tracking

Our SPCA learning algorithm was applied to the adaptive face tracker (details described in Section 3.1). The update frequency was specified as 3 (re-learn the sparse eigenfaces every third frame). Video recordings were made of presentations by two lecturers at the British Machine Vision Conference that was held in 2009. Each video recording comprises a lecture with a duration of length 1–2 min, taking up approximately 700 frames. The frame size is (352×288). The dataset is especially challenging as a result of abrupt changes in pose (with many widely varying poses: profile/up/down), size, and illumination conditions caused by the dynamic motions of the lecturers.

Our approach contrasted with that of the IVT technique that essentially adopts the *full* eigenfaces for the adaptive target model. Selected tracker parameters (e.g., the initial tracking state, image patch size, number of particles, and the scale σ_0 in the emission model), were set identical for both sparse and full PCA models to enable a fair comparison. In particular, the image patch size for both datasets was chosen as (24×24), that is, the data dimension was $p = 576$).

The initial-frame states were marked manually. The subspace dimension was set to $q = 4$ for both models, and the number of samples in the particle filtering process was fixed at 500. The sparsity level constraint for our model was set to

Table 2. Average tracking errors (in pixels) for the lecture video datasets

Video	IVT (full-PCA)	Tracker w/SPCA
Lecturer-1	3.8743	2.7396
Lecturer-2	5.8970	3.1989

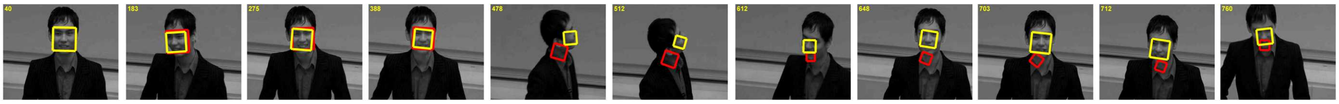
IVT, increment visual tracking; SPCA, sparse extension of principal component analysis.

$R = floor(p \times 0.05)$, that is, only 5% of the entries in the eigenvector were permitted to be non-zero. The tracking results that were obtained with both models are shown in Figure 5. The tracking states are superimposed in selected frames. Upon visual inspection, it was clear that our sparse-PCA achieved near successful tracking for both videos, whereas IVT often failed to track the target.

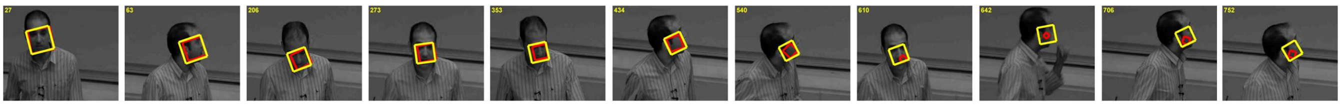
The quantitative errors are also reported. This was carried out by manually marking the center position of the face on every fifth frame, following which the Euclidean distance between the ground-truth position and the predicted center positions was recorded. The tracking errors that were averaged over all video frames are provided in Table 2. As shown, on average, the errors generated using our approach were significantly smaller for both videos than those obtained with IVT using full PCA for both videos. This indicates that the sparse target model would be a more robust tracker, attributed to its ability to keep track of the most salient features, while capturing the largest variation in the latest changes in the target. These enhanced abilities are highlighted by the results obtained using this video dataset that contains diverse target changes and noise levels. Finally, the time required to update the target model update time was also compared: 0.5 ms for IVT vs. 34.3 ms for the proposed SPCA. The latter method would therefore be able to handle approximately 20–30 frames per second, allowing for near real-time tracking.

5. Conclusions

In this paper, we proposed a very fast and scalable greedy forward selection algorithm for learning SPCA. Compared to recent methods based on iterative regression estimation or SDP-relaxation, our approach is able to process several thousands of data dimensions gracefully in reasonable time with little accuracy loss. The sparse eigenface model was tested using facial image data and was shown to be computationally efficient and capable of recovering the most salient and discriminative features from data, leading to superior face recognizer and



(a) Lecturer-1



(b) Lecturer-2

Figure 5. Tracking results for the British Machine Vision Conference 2009 lecture videos. Selected frames are highlighted where the yellow (brighter) box indicates our sparse extension of principal component analysis (PCA), while the red (darker) box is the increment visual tracking (full PCA). (a) Lecturer-1, and (b) Lecturer-2.

tracker performance that is robust to diverse forms of noise.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgments

This study was supported by the Research Program funded by the Seoul National University of Science and Technology.

References

[1] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *Philosophical Magazine Series 6*, vol. 2, no. 11, pp. 559-572, Nov. 1901. http://dx.doi.org/10.1080/14786440109462720

[2] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, no. 6, pp. 417-441, Sep. 1933. http://dx.doi.org/10.1037/h0071325

[3] I. T. Jolliffe, *Principal Component Analysis*, New York: Springer-Verlag, 1986.

[4] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Journal of the Optical Society of America A*: Optics and Image Science, vol. 4, no. 3, pp. 519-524, Mar. 1987.

[5] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for the characterization of human faces," *IEEE Transactions on Pattern analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103–108, 1990.

[6] M. Turk and A. Pentland, "Face recognition using eigenfaces," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–591, 1991. http://dx.doi.org/10.1109/34.41390

[7] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991. http://dx.doi.org/10.1162/jocn.1991.3.1.71

[8] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125-141, May 2008. http://dx.doi.org/10.1007/s11263-007-0075-7

[9] I. T. Jolliffe, N. T. Trendafilov, and M. Uddin, "A modified principal component technique based on the LASSO," *Journal of Computational and Graphical Statistics*, vol. 12, no. 3, pp. 531-547, Sep. 2003. http://dx.doi.org/10.1198/1061860032148

[10] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, pp. 265-286, Jun. 2006. http://dx.doi.org/10.1198/106186006X113430

[11] A. d'Aspremont, L. El Ghaoui, M. Jordan, and G. Lanckriet, "A direct formulation of sparse PCA using semidefinite programming," *SIAM Review*, vol. 49, no. 3, pp. 434-448, 2007. http://dx.doi.org/10.1137/050645506

[12] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Review*, vol. 38, no. 1, pp. 49-95, 1996. http://dx.doi.org/10.1137/1038003

[13] K. C. Toh, M. J. Todd, and R. Tütüncü, "SDPT3: a Matlab software package for semidefinite programming, version 1.3," *Optimization Methods and Software*,vol. 11,

no. 1-4, pp. 545-581, Jan. 1999. http://dx.doi.org/10.1080/10556789908805762

[14] R. Tütüncü, K. C. Toh, and M. J. Todd, "Solving semidefinite-quadratic-linear programs using SDPT3," *Mathematical Programming*, , vol. 95, no. 2, pp. 189-217, Feb. 2003. http://dx.doi.org/10.1007/s10107-002-0347-5

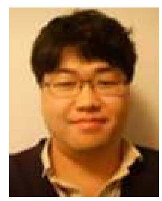
[15] B. Moghaddam, Y. Weiss, and S. Avidan, "Spectral bounds for sparse PCA: exact and greedy algorithms," *Advances in Neural Information Processing Systems*, vol. 18, pp. 915-922, 2005.

[16] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," ," in *Computer Vision: ECCV '96, Lecture Notes in Computer Science Vol. 1064*, B. Buxton and R. Cipolla, Eds. Heidelberg: Springer Berlin, 1996, pp. 343-356. http://dx.doi.org/10.1007/BFb0015549

[17] A. Levey and M. Lindenbaum, "Sequential Karhunen-Loeve basis extraction and its application to images," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1371-1374, Aug. 2000. http://dx.doi.org/10.1109/83.855432

[18] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, vol. 61, no. 3, pp. 611-622, 1999.http://dx.doi.org/10.2307/2680726

[19] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643-660, Jun. 2001. http://dx.doi.org/10.1109/34.927464



Minyoung Kim received his B.S. and M.S. degrees both in computer science and engineering from Seoul National University, South Korea. He earned a Ph.D. degree in computer science from Rutgers University in 2008. From 2009 to 2010 he was a postdoctoral researcher at the Robotics Institute of Carnegie Mellon University. He is currently an assistant professor in the Department of Electronics and IT Media Engineering at Seoul National University of Science and Technology in Korea. His primary research interest is machine learning and computer vision. His research focus includes graphical models, motion estimation/tracking, discriminative models/learning, kernel methods, and dimensionality reduction.