Machine Learning

Contents

1. Introduction
2. K-Nearest Neighbor Algorithm
3. LDA(Linear Discriminant Analysis)
4. Perceptron
5. Feed-Forward Neural Networks
6. RNN(Recurrent Neural Networks)
7. SVM(Support Vector Machine)
8. Ensemble Learning
9. CNN(Convolutional Neural Network)
10. **PCA(Principal Component Analysis)**
11. ICA(Independent Component Analysis)
12. Clustering
13. GAN(Generative Adversarial Network)

10.1. Dimensionality Reduction

Data is often high dimensional - images, bag-of-word descriptions, gene-expressions etc. Provided there is some `structure', data will typically lie close to a much lower dimensional `manifold'.

- What
  - Given d input variables (*d*-dimensional data), reduce it to *k* input variables (*k*-dimensional data), $k < d$, without loss of information

- Why
  - Reduces time complexity: Less computation
  - Reduces space complexity: Less parameters
  - Saves the cost of observing the feature
  - Simpler models are more robust on small datasets
  - More interpretable; simpler explanation
  - Data visualization (structure, groups, outliers, etc.) easy if plotted in 2 or 3 dimensions

Feature Selection vs. Extraction

- Feature selection
  - Choose $k<d$ important features, ignoring the remaining (d-k) features
  - Subset selection algorithms
- Feature extraction
  - Project the original x_i, $i=1,\ldots,d$ dimensions to new $k<d$ dimensions, z_j, $j=1,\ldots,k$
  - Principal component analysis (PCA), linear discriminant analysis (LDA), factor analysis (FA)

Subset Selection

of subset selection of d features = 2^d

- Cannot enumerate all of them!

Forward search: Add the best feature at each step iteratively

- Set of features F initially $\varnothing$
- At each iteration, find the best new feature
  - $j = \text{argmin}_i \ \text{Err}(F \cup X_i)$
- Add X_j to F if $\text{Err}(F \cup X_j) < \text{Err}(F)$ else stop

- Requires $O(d^2)$ times of training and testing
- Optimal?

Sequential backward selection: Start with all features and remove one at a time, if possible

Floating search: Add k' features and remove m' features at each step

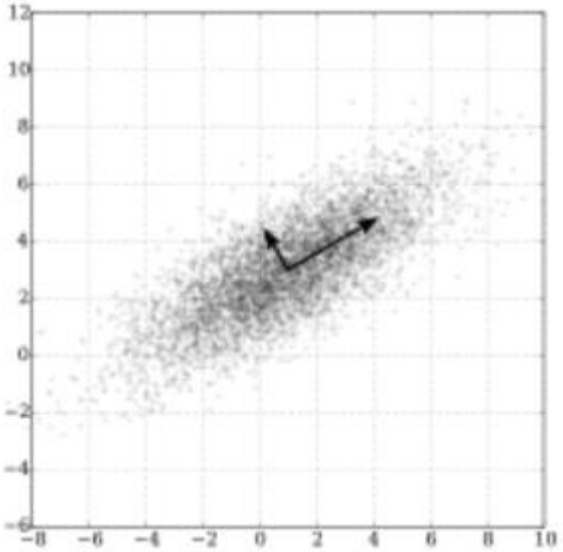
10.2. Principal Components Analysis (PCA)

- Idea:
  - Given data points in *d*-dimensional space, project onto lower dimensional space while preserving as much information as possible
    - E.g., find best planar approximation to 3D data
    - E.g., find best planar approximation to 104D data
  - In particular, choose projection that minimizes the squared error in reconstructing original data

- Learned encoding is a linear combination of inputs
- Given *d*-dimensional data **x**, learns the top *m*-dimensions where
  - the dimensions are orthogonal
  - the re-representational as a linear combination of the top *m*-dimensions minimizes reconstruction error (sum of squared errors)

Maximum co-variance and orthogonality

- Select a direction in *m*-dimensional space along which the variance in **x** is maximized.
- Find another direction along which variance is maximised, but restrict the search to all directions orthonormal to all previous selected directions.
- Repeat this until *m* vectors are selected.



PCA of a multivariate Gaussian distribution centered at (1,3) with a standard deviation of 3 in roughly the (0.866, 0.5) direction and of 1 in the orthogonal direction. The vectors shown are the eigenvectors of the covariance matrix scaled by the square root of the corresponding eigenvalue, and shifted so their tails are at the mean.

10.3. Eigenstructure of Principal Component Analysis

zero - mean random vector :

$$\mathbf{x} = [x_1, x_2, \cdots, x_d]^T$$

projection unit vector :

$$\mathbf{w} = [w_1, w_2, \cdots, w_d]^T \text{ and } \quad \|\mathbf{w}\| = \left(\mathbf{w}^T\mathbf{w}\right)^{1/2} = 1$$

Projection :

$$z = \mathbf{w}^T\mathbf{x} \text{ and } E[z] = E[\mathbf{w}^T\mathbf{x}] = \mathbf{w}^T E[\mathbf{x}] = 0$$

Variance :

$$\sigma^2 = E[z^2] - E^2[z] = E[(\mathbf{w}^T\mathbf{x})(\mathbf{x}^T\mathbf{w})] = \mathbf{w}^T E[\mathbf{x}\mathbf{x}^T]\mathbf{w} = \mathbf{w}^T\mathbf{S}\mathbf{w}$$

Correlation Matrix :

$$\mathbf{S} = E[\mathbf{x}\mathbf{x}^T]$$

The variance of the projection z is a function of the unit vector $\mathbf{w}$

$$\psi(\mathbf{w}) = \sigma^2 = \mathbf{w}^T\mathbf{S}\mathbf{w}$$

If $\mathbf{w}$ is a unit vector such that the variance probe $\psi(\mathbf{w})$ has an extremal value, for any small perturbation $\delta\mathbf{w}$ of the unit vector $\mathbf{w}$

$$\psi(\mathbf{w} + \delta\mathbf{w}) = \psi(\mathbf{w})$$

$$\psi(\mathbf{w} + \delta\mathbf{w}) = (\mathbf{w} + \delta\mathbf{w})^T \mathbf{S}(\mathbf{w} + \delta\mathbf{w}) = \mathbf{w}^T \mathbf{S}\mathbf{w} + 2(\delta\mathbf{w})^T \mathbf{S}\mathbf{w} + (\delta\mathbf{w})^T \mathbf{S}\delta\mathbf{w}$$

Ignoring the second - order term $(\delta\mathbf{w})^T \mathbf{S}\delta\mathbf{w}$

$$\psi(\mathbf{w} + \delta\mathbf{w}) = \mathbf{w}^T \mathbf{S}\mathbf{w} + 2(\delta\mathbf{w})^T \mathbf{S}\mathbf{w} = \psi(\mathbf{w}) + 2(\delta\mathbf{w})^T \mathbf{S}\mathbf{w}$$

Hence

$$(\delta\mathbf{w})^T \mathbf{S}\mathbf{w} = 0$$

Restriction

$$\|\mathbf{w} + \delta\mathbf{w}\| = (\mathbf{w} + \delta\mathbf{w})^T (\mathbf{w} + \delta\mathbf{w}) = 1$$

$$(\delta\mathbf{w})^T \mathbf{w} = 0$$

$$(\delta\mathbf{w})^T \mathbf{S}\mathbf{w} - \lambda(\delta\mathbf{w})^T \mathbf{w} = 0 \quad \text{equivalently} \quad (\delta\mathbf{w})^T (\mathbf{S}\mathbf{w} - \lambda\mathbf{w}) = 0$$

Eigenvalue Problem $\mathbf{S}\mathbf{w} = \lambda\mathbf{w}$

Eigenvalue Problem $\mathbf{S}\mathbf{w}_j = \lambda_j \mathbf{w}_j, \quad j = 1,2,...,d$

Let the corresponding eigenvalues be arranged in decreasing order

$$\lambda_1 > \lambda_2 > \lambda_3 ..... > \lambda_d$$

Associated eigenvectors

$$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, ... \mathbf{w}_d]$$

Orthonormality

$$\mathbf{w}_i^T \mathbf{w}_j = 1 \text{ (if } i=j \text{) and 0 otherwise} \quad \mathbf{W}^T \mathbf{W} = \mathbf{I}$$

Basic data representation

$$z_j = \mathbf{w}_j^T \mathbf{x}, \qquad j=1,2,...,d : \text{principal component}$$

$$\mathbf{z} = [z_1, z_2, z_3, ... z_d]^T = \mathbf{W}^T \mathbf{x} \quad \text{and} \quad \mathbf{x} = \mathbf{W}\mathbf{z} = \sum_{j=1}^{d} z_j \mathbf{w}_j$$

Dimensionality Reduction

$$\mathbf{x}' = \sum_{j=1}^{k} z_j \mathbf{w}_j \qquad \mathbf{e} = \mathbf{x} - \mathbf{x}' = \sum_{j=k+1}^{d} z_j \mathbf{w}_j$$

참조: 차원축소 오차

식 (10.3.25)를 유도해보자. (10.3.24)를 대입하면

$$E[\boldsymbol{e}^T\boldsymbol{e}] = E[\sum_{j=k+1}^{d} z_j \boldsymbol{w}_j^T \boldsymbol{w}_j z_j] = \sum_{j=k+1}^{d} E[z_j^2] = \sum_{j=k+1}^{d} \boldsymbol{w}_j^T E[\boldsymbol{x}\boldsymbol{x}^T]\boldsymbol{w}_j \qquad (10.3.26)$$

$$= \sum_{j=k+1}^{d} \boldsymbol{w}_j^T \boldsymbol{S}\boldsymbol{w}_j = \sum_{j=k+1}^{d} \boldsymbol{w}_j^T \lambda_j \boldsymbol{w}_j = \sum_{j=k+1}^{d} \lambda_j$$

를 얻게 된다.

예제 10.3-1

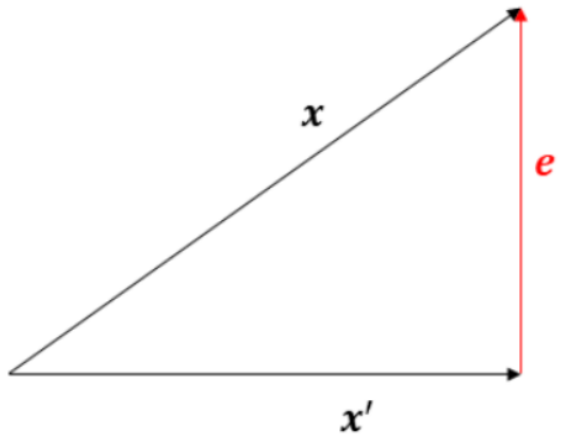
PCA에 의해 차원 축소 후 복원된 벡터 $\boldsymbol{x}'$과 복원 오차 $\boldsymbol{e}$가 각각 식 (10.3.23)과 (10.3.24)로 주어졌다. 두 벡터 $\boldsymbol{x}'$과 $\boldsymbol{e}$가 직교함을 보이고, 이들 사이의 관계를 $\boldsymbol{x}$와 함께 그림으로 나타내어라.

풀이

먼저, 두 벡터 $\boldsymbol{x}'$과 $\boldsymbol{e}$가 직교함을 보이겠다. 이를 위하여 두 벡터의 내적을 계산하면

$$\boldsymbol{e}^T\boldsymbol{x}' = (\sum_{j=k+1}^{d} z_j\boldsymbol{w}_j^T)(\sum_{i=1}^{k} z_i\boldsymbol{w}_i) = \sum_{j=k+1}^{d}\sum_{i=1}^{k} z_j\boldsymbol{w}_j^T\boldsymbol{w}_i z_i = 0$$

이 된다. 그 이유는 $\boldsymbol{w}_i^T\boldsymbol{w}_j = \begin{cases} 1 \text{ if } i=j \\ 0 \text{ otherwise} \end{cases}$와 같이 정규직교하기 때문이다. $\boldsymbol{x}$, $\boldsymbol{x}'$, $\boldsymbol{e}$의 관계를 그림으로 그리면 아래와 같다.



PCA Algorithm

Algorithm Principal Components Analysis to form an M-dimensional approximation of a dataset $\{\mathbf{x}^n, n = 1, \ldots, N\}$, with $\dim(\mathbf{x}^n) = D$.

1: Find the $D \times 1$ sample mean vector and $D \times D$ covariance matrix

$$\mathbf{m} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}^n, \quad \mathbf{S} = \frac{1}{N-1}\sum_{n=1}^{N}(\mathbf{x}^n - \mathbf{m})(\mathbf{x}^n - \mathbf{m})^\mathsf{T}$$

2: Find the eigenvectors $\mathbf{e}^1, \ldots, \mathbf{e}^D$ of the covariance matrix $\mathbf{S}$, sorted so that the eigenvalue of $\mathbf{e}^i$ is larger than $\mathbf{e}^j$ for $i < j$. Form the matrix $\mathbf{E} = [\mathbf{e}^1, \ldots, \mathbf{e}^M]$.

3: The lower dimensional representation of each data point $\mathbf{x}^n$ is given by

$$\mathbf{y}^n = \mathbf{E}^\mathsf{T}(\mathbf{x}^n - \mathbf{m})$$

4: The approximate reconstruction of the original datapoint $\mathbf{x}^n$ is

$$\mathbf{x}^n \approx \mathbf{m} + \mathbf{E}\mathbf{y}^n$$

5: The total squared error over all the training data made by the approximation is

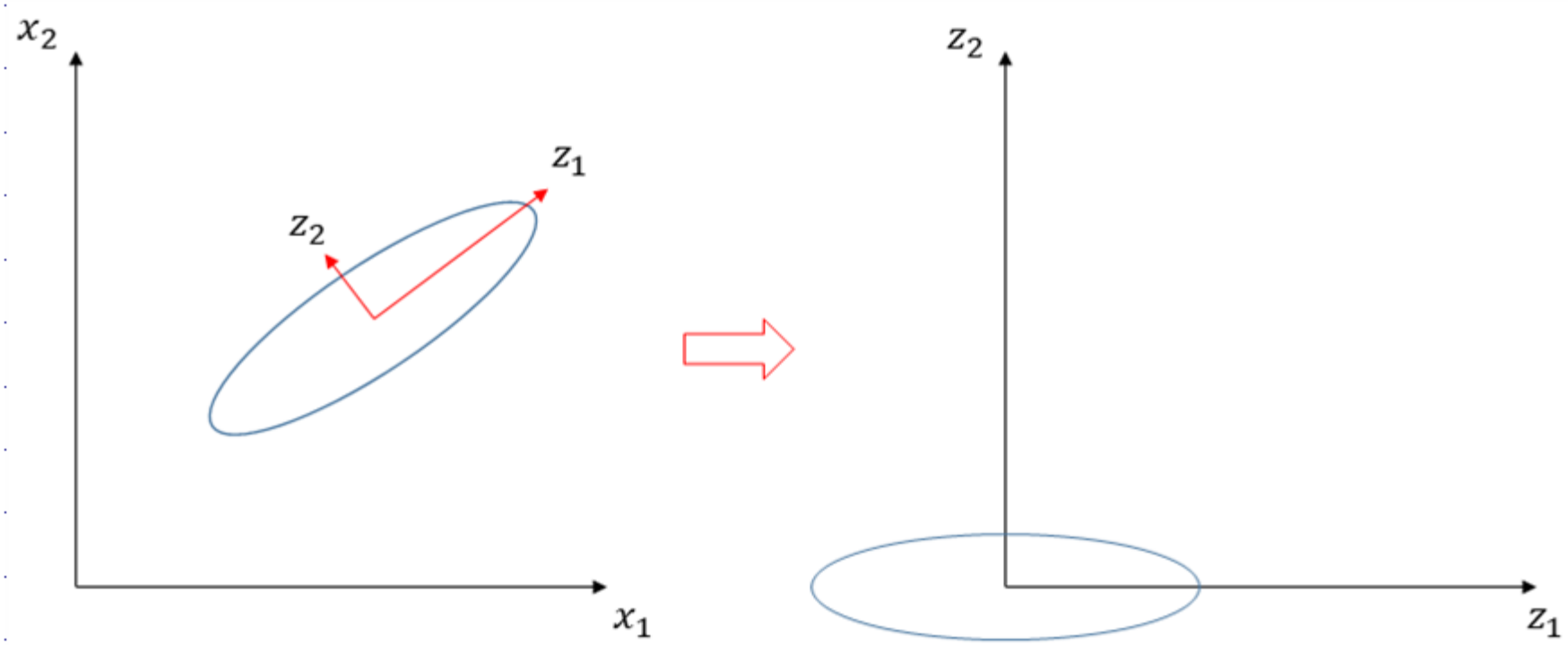
$$\sum_{n=1}^{N}(\mathbf{x}^n - \tilde{\mathbf{x}}^n)^2 = (N-1)\sum_{j=M+1}^{D}\lambda_j$$

where $\lambda_{M+1} \ldots \lambda_N$ are the eigenvalues discarded in the projection.

10.4. Understanding PCA

Define $\mathbf{z} = W^T(\mathbf{x} - \mathbf{m})$

- W: Columns as eigenvectors of S (estimator to Σ)
- **m**: Sample mean
- Centers the data at the origin and rotates the axis



Choosing the Size of Reduced Dimension

$|S| = \prod_{i=1}^{d} \lambda_i$ measures how much Z varies

- Larger eigenvalues contribute more to variance of Z

Proportion of Variance (PoV) explained by k principal components:

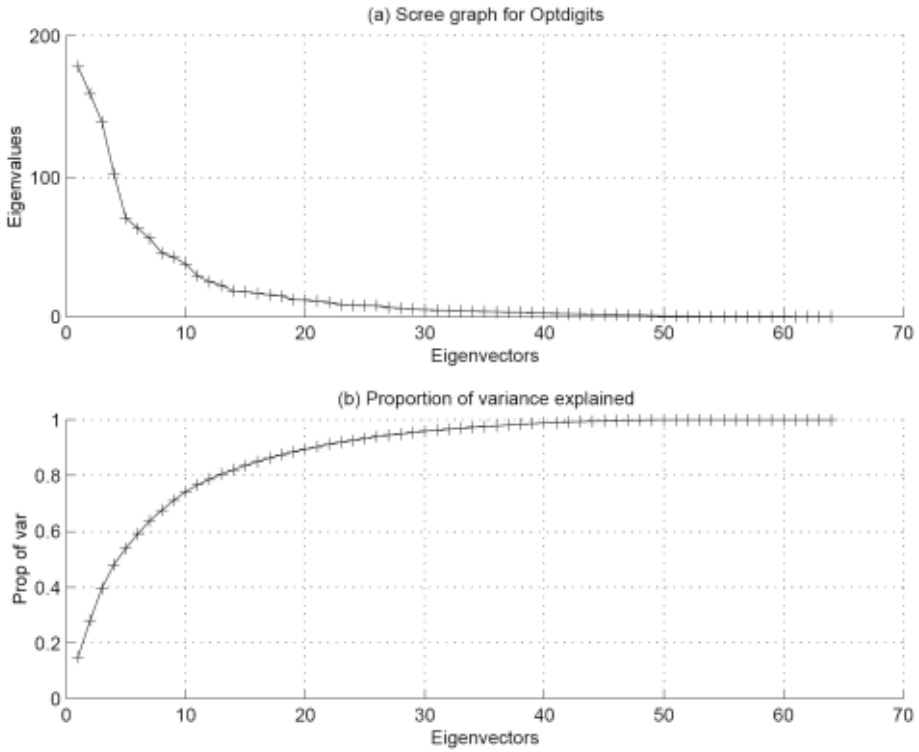
- POV: $$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d}$$

- Rule of thumb: stop at PoV > 0.9

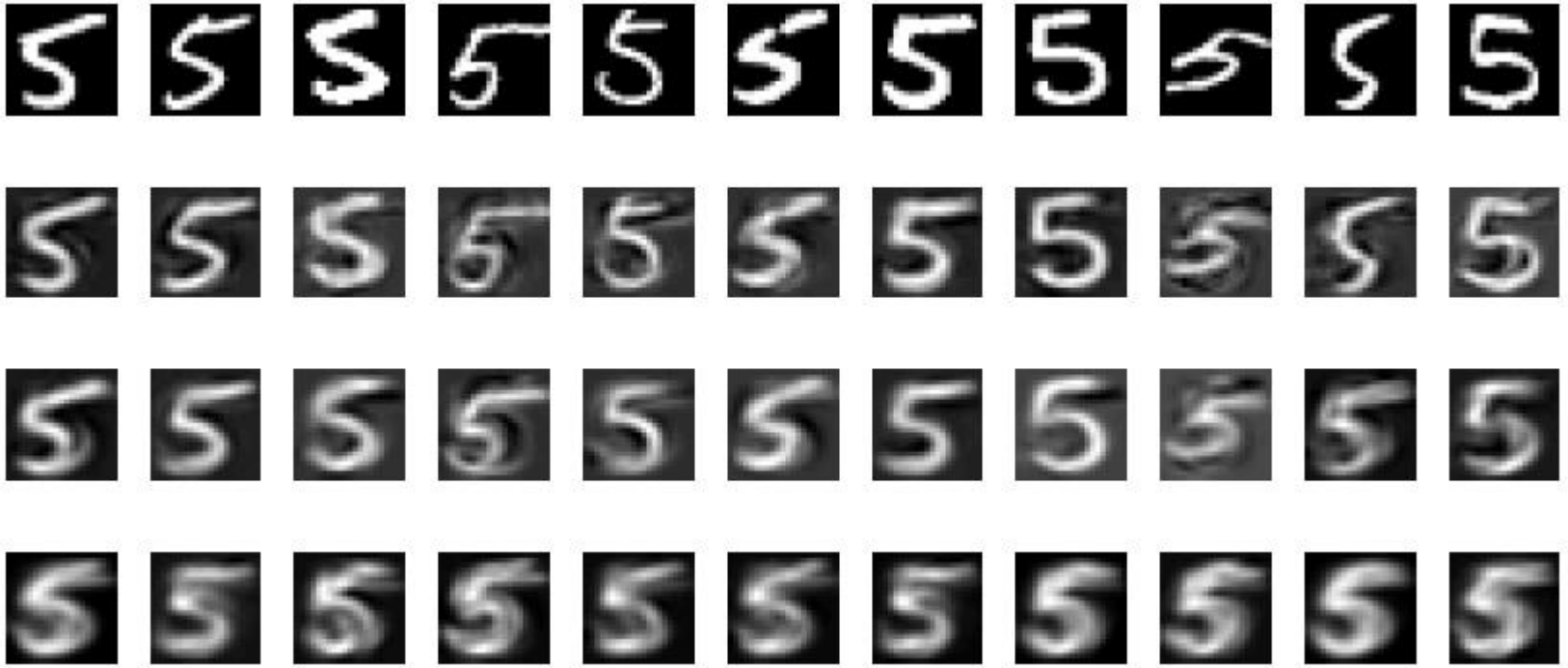
Scree graph: plots of λ_k vs. k

- Stop at elbow

Only keep the eigenvectors whose eigenvalues are larger than average input variance (i.e. average of eigenvalue)



Reducing the dimension of digits



Top row : a selection of the digit 5 taken from the database of 892 examples. Plotted beneath each digit is the reconstruction using 100, 30 and 5 eigenvectors (from top to bottom). Note how the reconstructions for fewer eigenvectors express less variability from each other, and resemble more a mean 5 digit.

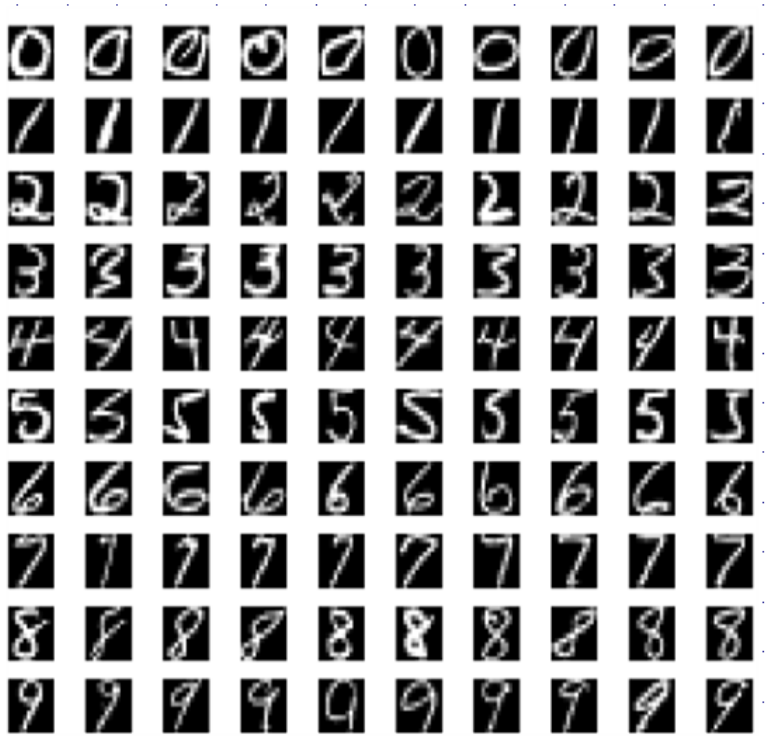


그림 10.4. 숫자 영상 예시

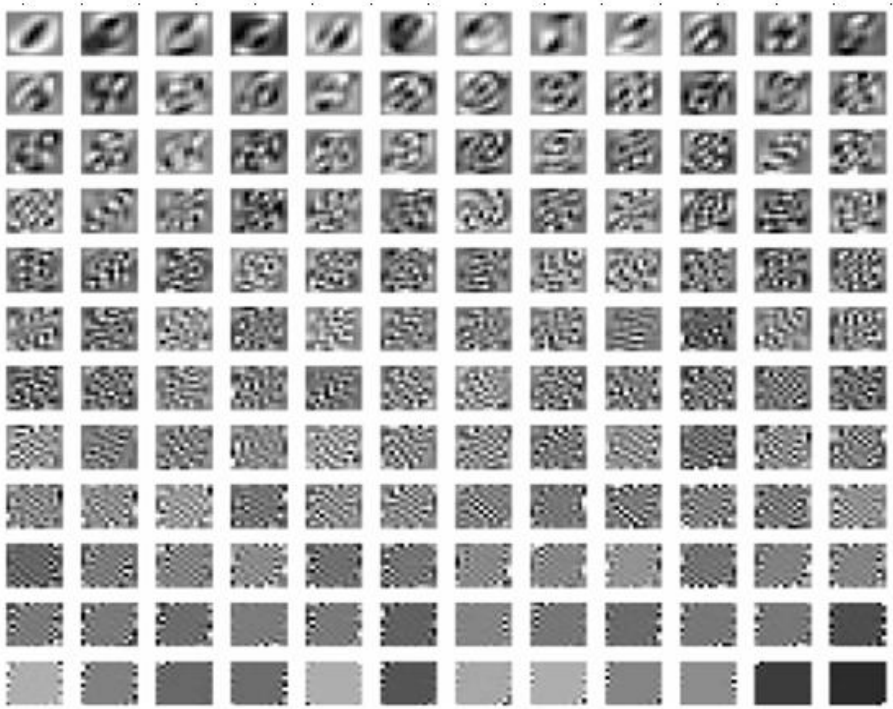


그림 10.5. 숫자영상에서 얻어진 고유벡터

그림 10.6. 주성분 분석에 의한 영상복원 예시

Reducing the dimension of faces

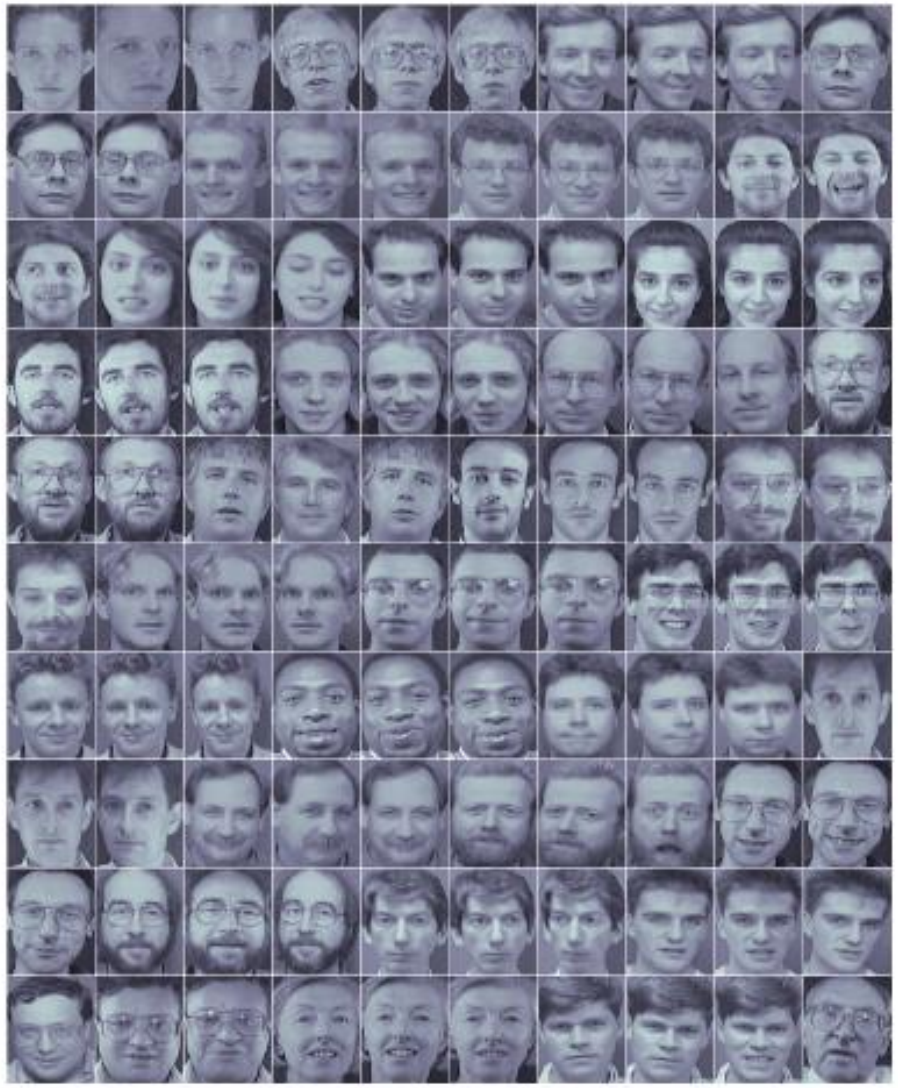
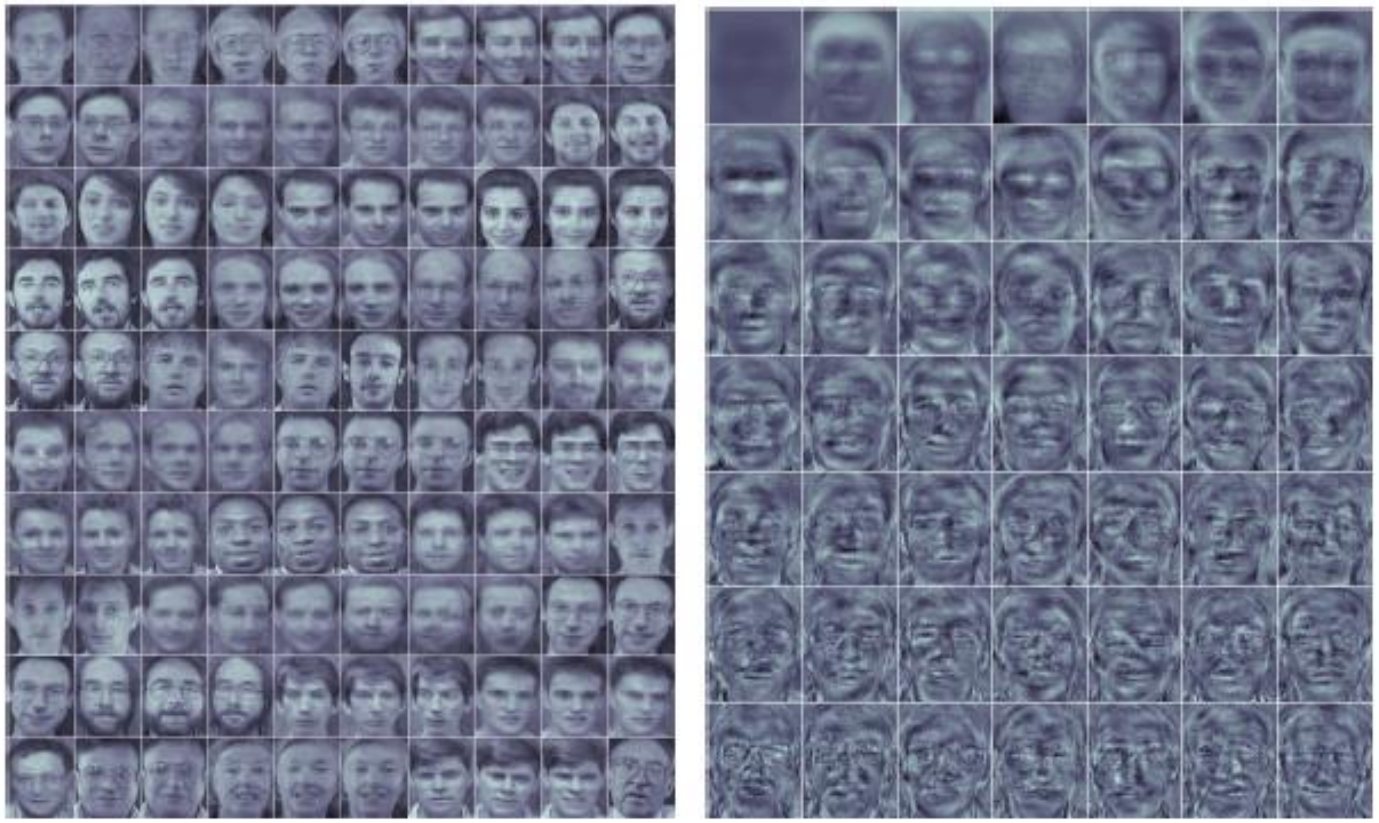


Figure : 100 of the 120 training images (40 people, with 3 images of each person). Each image consists of $92 \times 112 = 10304$ non-negative greyscale pixels. The data is scaled so that, represented as an image, the components of each image sum to 1. The average value of each pixel across all images is 9.70×10^{-5}. This is a subset of the 400 images in the full Olivetti Research Face Database

Reducing the dimension of faces



(a) (b)

Figure : **(a)**: SVD reconstruction of the images using a combination of the 49 eigen-images. **(b)**: The eigen-images are found using SVD of the above data and taking the 49 eigenvectors with largest eigenvalue. The images corresponding to the largest eigenvalues are contained in the first row, and the next 7 in the row below, *etc.*