Linear Dimensionality Reduction

Hong Chang

Institute of Computing Technology, Chinese Academy of Sciences

Machine Learning (Spring 2015)
Outline I

1. Introduction
2. Principal Component Analysis
3. Factor Analysis
4. Multidimensional Scaling
5. Linear Discriminant Analysis
Why Dimensionality Reduction?

- The number of inputs (input dimensionality) often affects the time and space complexity of the learning algorithm.
- Eliminating an input saves the cost of extracting it.
- Simpler models are often more robust on small data sets.
- Simpler models are more interpretable, leading to simpler explanation.
- Data visualization in 2 or 3 dimensions facilitates the detection of structure and outliers.
Feature Selection vs. Extraction

- **Feature selection:**
  - Choosing $K < D$ important features and discarding the remaining $D - K$.
  - Subset selection algorithms

- **Feature extraction:**
  - Projecting the original $D$ dimensions to $K (< D)$ new dimensions.
  - **Unsupervised** methods (without using output information):
    - Principal component analysis (PCA)
    - Factor analysis (FA)
    - Multidimensional scaling (MDS)
  - **Supervised** methods (using output information):
    - Linear discriminant analysis (LDA)

The linear methods above also have nonlinear extensions.
Principal Component Analysis

- PCA finds a linear mapping from the $D$-dimensional input space to a $K$-dimensional space ($K < D$) with minimum information loss according to some criterion.
- Projection of $\mathbf{x}$ on the direction of $\mathbf{w}$:

$$z = \mathbf{w}^T \mathbf{x}$$

- Finding the first principal component $\mathbf{w}_1$ s.t. $\text{var}(z_1)$ is maximized:

$$\text{var}(z_1) = \text{var}(\mathbf{w}_1^T \mathbf{x}) = E[(\mathbf{w}_1^T \mathbf{x} - \mathbf{w}_1^T \mu)^2] = E[\mathbf{w}_1^T (\mathbf{x} - \mu)(\mathbf{x} - \mu)^T \mathbf{w}_1] = \mathbf{w}_1^T \mathbf{\Sigma} \mathbf{w}_1$$

where

$$\text{cov}(\mathbf{x}) = E[(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T] = \mathbf{\Sigma}$$
Optimization Problem for First Principal Component

- Maximization of $\text{var}(z_1)$ subject to $\|w_1\| = 1$ can be solved as a constrained optimization problem using a Lagrange multiplier.

- Maximization of Lagrangian:

$$w_1^T \Sigma w_1 - \alpha (w_1^T w_1 - 1)$$

- Taking the derivative of the Lagrangian w.r.t. $w_1$ and setting it to 0, we get an eigenvalue equation for the first principal component $w_1$:

$$\Sigma w_1 = \alpha w_1$$

- Because we have

$$w_1^T \Sigma w_1 = \alpha w_1^T w_1 = \alpha$$

we choose the eigenvector with the largest eigenvalue for the variance to be maximum.

Hong Chang (ICT, CAS)
The second principal component $\mathbf{w}_2$ should also maximize the variance $\text{var}(z_2)$, subject to the constraints that $\|\mathbf{w}_2\| = 1$ and that $\mathbf{w}_2$ is orthogonal to $\mathbf{w}_1$.

**Maximization of Lagrangian:**

$$\mathbf{w}_2^T \Sigma \mathbf{w}_2 - \alpha (\mathbf{w}_2^T \mathbf{w}_2 - 1) - \beta (\mathbf{w}_2^T \mathbf{w}_1 - 0)$$

Taking the derivative of the Lagrangian w.r.t. $\mathbf{w}_2$ and setting it to 0, we get the following equation:

$$2 \Sigma \mathbf{w}_2 - 2\alpha \mathbf{w}_2 - \beta \mathbf{w}_1 = 0$$

We can show that $\beta = 0$ and hence have this eigenvalue equation:

$$\Sigma \mathbf{w}_2 = \alpha \mathbf{w}_2$$

implying that $\mathbf{w}_2$ is the eigenvector of $\Sigma$ with the second largest eigenvalue.
What PCA Does

- **Transformation** of data:

  \[ z = W^T(x - m) \]

  where the columns of \( W = [w_1, w_2, \ldots] \) are the eigenvectors of \( \Sigma \) and \( m \) is the sample mean.

- **Centering** the data at the origin and **rotating** the axes:

  If the variance on \( z_2 \) is too small, it can be ignored to reduce the dimensionality from 2 to 1.
**How to Choose $K$**

- **Proportion of variance (PoV) explained:**
  \[
  \frac{\lambda_1 + \lambda_2 + \ldots + \lambda_K}{\lambda_1 + \lambda_2 + \ldots + \lambda_D}
  \]

  where $\lambda_i$ are sorted in descending order.

- Typically, stop at PoV $> 0.9$

- Scree graph plotting PoV against $K$; stop at “elbow”.

Hong Chang (ICT, CAS)
Scree Graph

(a) Scree graph for Optdigits

(b) Proportion of variance explained
Scatterplot in Lower-Dimensional Space
**Eigenfaces**: a set of eigenvectors as basis features for face images.

**Computing eigenfaces**:

1. A set of $N$ face images, each being represented as a $D$-dimensional vector $\mathbf{x}^{(i)}$, $\mathbf{S} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}]$. 
Computing eigenfaces:

2 Compute the mean face image $\mathbf{m} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^{(i)}$.

3 Centralize the face images by subtracting the mean face $\mathbf{m}$:

$$\hat{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} - \mathbf{m}$$

$$\hat{\mathbf{S}} = [\hat{\mathbf{x}}^{(1)}, \ldots, \hat{\mathbf{x}}^{(N)}]$$
PCA for Face Recognition: Eigenface (3)

Computing eigenfaces:

1. Compute the eigenvectors and eigenvalues of the covariance matrix \( C = \hat{S}\hat{S}^T \). The eigenvectors are therefore called eigenfaces. They are the directions in which the images differ from the mean face.
Computing eigenfaces:

5. Choose the principal components. E.g., choose $k$ principal components according to

$$\frac{\lambda_1 + \ldots + \lambda_k}{\lambda_1 + \ldots + \lambda_D} \geq \epsilon.$$

6. The eigenfaces can be used to represent both existing and new faces. A new centralized face image can be projected on the eigenfaces.
Eigenface for Face Reconstruction

Reconstruction procedure
Comments on Eigenface

**Advantages:**
- Completely automatic training process and easy coding
- The complexity of face representation is reduced
- Real time face recognition
- Handling large databases

**Disadvantages:**
- Very sensitive to lighting, scale and translation
- Difficult to capture expression changes
- The most significant eigenfaces are mainly about illumination encoding, with less useful information regarding the actual face.
Factor Analysis

- FA assumes that there is a set of latent factors $z_j$ which when acting in combination to generate the observed variables $x$.
- The goal of FA is to characterize the dependency among the observed variables by means of a smaller number of factors.

Problem settings:
- Sample $\mathcal{X} = \{x^{(i)}\}$: drawn from some unknown probability density with $E[x] = \mu$ and $\text{cov}(x) = \Sigma$.
- Factors $z_k$ with unit normals:
  
  $$E[z_k] = 0, \text{var}(z_k) = 1, \text{cov}(z_j, z_k) = 0, j \neq k.$$  

- Noise sources $\epsilon_j$:
  
  $$E[\epsilon_j] = 0, \text{var}(\epsilon_j) = \Psi_j, \text{cov}(\epsilon_i, \epsilon_j) = 0, i \neq j, \text{cov}(\epsilon_j, z_k) = 0.$$
Each of the $D$ input dimensions $x_j$ is expressed as a weighted sum of the $K (< D)$ factors $z_k$ plus some residual error term:

$$x_j - \mu_j = \sum_{k=1}^{K} v_{jk} z_k + \epsilon_j,$$

where $v_{jk}$ are called factor loadings.

Without loss of generality, we assume that $\mu = 0$. 

Hong Chang (ICT, CAS)

Linear Dimensionality Reduction
The factor $z_k$ are **independent unit normals** that are stretched, rotated and translated to generate the inputs $x$. 
The direction of FA is opposite to that of PCA:

- PCA (from $\mathbf{x}$ to $\mathbf{z}$): $\mathbf{z} = \mathbf{W}^T (\mathbf{x} - \mu)$
- FA (from $\mathbf{z}$ to $\mathbf{x}$): $\mathbf{x} - \mu = \mathbf{Vz} + \epsilon$
Given that $\text{var}(z_k) = 1$ and $\text{var}(\epsilon_j) = \Psi_j$,

$$\text{var}(x_j) = \sum_{k=1}^{K} v_{jk}^2 \text{var}(z_k) + \text{var}(\epsilon_j) = \sum_{k=1}^{K} v_{jk}^2 + \Psi_j$$

where the first part ($\sum_{k=1}^{K} v_{jk}^2$) is the variance explained by the common factors and the second part ($\Psi_j$) is the variance specific to $x_j$.

**Covariance matrix:**

$$\Sigma = \text{cov}(x) = \text{cov}(Vz + \epsilon) = \text{cov}(Vz) + \text{cov}(\epsilon) = V \text{cov}(z) V^T + \Psi = VV^T + \Psi$$

where $\Psi = \text{diag}([\Psi_1, \ldots, \Psi_D])$
2-Factor Example

Let

\[ \mathbf{x} = (x_1, x_2)^T \quad \mathbf{V} = \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{pmatrix} \]

Since

\[ \Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{pmatrix} = \mathbf{V} \mathbf{V}^T + \Psi = \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{pmatrix} \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{pmatrix} + \begin{pmatrix} \psi_1 & 0 \\ 0 & \psi_2 \end{pmatrix} \]

we have

\[ \sigma_{12} = \text{cov}(x_1, x_2) = v_{11}v_{21} + v_{12}v_{22} \]

If \( x_1 \) and \( x_2 \) have high covariance, then they are related through a factor:
- If it is the first factor, then \( v_{11} \) and \( v_{21} \) will both be high.
- If it is the second factor, then \( v_{12} \) and \( v_{22} \) will both be high.

If \( x_1 \) and \( x_2 \) have low covariance, then they depend on different factors:
- In each of the products \( v_{11}v_{21} \) and \( v_{12}v_{22} \), one term will be high and the other will be low.
Because

\[ \text{cov}(x_1, z_1) = \text{cov}(v_{11}z_1 + v_{12}z_2 + \epsilon_1, z_1) = \text{cov}(v_{11}z_1, z_1) = v_{11}\text{var}(z_1) = v_{11} \]

and similarity,

\[ \text{cov}(x_1, z_2) = v_{12} \]
\[ \text{cov}(x_2, z_1) = v_{21} \]
\[ \text{cov}(x_2, z_2) = v_{22} \]

so we have

\[ \text{cov}(x, z) = V \]

i.e., the factor loadings \( V \) represent the covariance between the variables and the factors.
Given $S$ as the estimator of $\Sigma$, we want to find $V$ and $\Psi$ such that

$$S = VV^T + \Psi$$

If there are only a few factors (i.e., $K \ll D$), then we can get a simplified structure for $S$.

The number of parameters is reduced from $D(D + 1)/2$ (for $S$) to $DK + D$ (for $VV^T + \Psi$).

Special cases:
- Probabilistic PCA (PPCA): $\Psi = \Psi I$ (i.e., all $\Psi_j$ are equal)
- PCA: $\Psi_j = 0$

For dimensionality reduction, FA offers no advantage over PCA except the interpretability of factors allowing the identification of common causes, a simple explanation, and knowledge extraction.
Problem formulation:

- Given the **pairwise distances** between pairs of points in some space (but the exact coordinates of the points and their dimensionality are unknown).
- We want to **embed** the points in a **lower-dimensional space** such that the pairwise distances in this space are as close as possible to those in the original space.

The projection to the lower-dimensional space is not unique because the pairwise distances are invariant to such operations as translation, rotation and reflection.
MDS Embedding of Cities

Hong Chang (ICT, CAS)

Linear Dimensionality Reduction
Derivation

Sample $\mathcal{X} = \{\mathbf{x}^{(i)} \in \mathbb{R}^{D}\}_{i=1}^{N}$, $\mathbf{X} = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}]^T \in \mathbb{R}^{N \times D}$

Squared Euclidean distance between points $r$ and $s$:

$$d_{rs}^2 = \|\mathbf{x}^{(r)} - \mathbf{x}^{(s)}\|^2 = \sum_{j=1}^{D} (x_j^{(r)} - x_j^{(s)})^2$$

(1)

$$= \sum_{j=1}^{D} (x_j^{(r)})^2 - 2\sum_{j=1}^{D} x_j^{(r)} x_j^{(s)} + \sum_{j=1}^{D} (x_j^{(s)})^2$$

(2)

$$= b_{rr} + b_{ss} - 2b_{rs}$$

(3)

where

$$b_{rs} = \sum_{j=1}^{D} x_j^{(r)} x_j^{(s)}$$

(dot product of $\mathbf{x}^{(r)}$ and $\mathbf{x}^{(s)}$)

or in matrix form:

$$\mathbf{B} = \mathbf{X} \mathbf{X}^T$$

Hong Chang (ICT, CAS)
Centering of data to constrain the solution:

\[ \sum_{i=1}^{N} x_j^{(i)} = 0, \forall i = 1, \ldots, D \]

Summing up equation (3) on \( r \), \( s \) and both \( r \), \( s \), and defining

\[ T = \sum_{i=1}^{N} b_{ii} = \sum_{i} \sum_{j} (x_j^{(i)})^2 \]

we get

\[ \sum_r d_{rs}^2 = T + Nb_{ss} \]

\[ \sum_s d_{rs}^2 = Nb_{rr} + T \]

\[ \sum_r \sum_s d_{rs}^2 = 2NT \]
Derivation (3)

By defining

\[ d_{rs}^2 = \frac{1}{N} \sum_r d_{rs}^2, \quad d_{r*}^2 = \frac{1}{N} \sum_s d_{rs}^2, \quad d_{**}^2 = \frac{1}{N^2} \sum_r \sum_s d_{rs}^2 \]

and using equation (3), we get

\[ b_{rs} = \frac{1}{2} (d_{r*}^2 + d_{*s}^2 - d_{**}^2 - d_{rs}^2) \]

\( \mathbf{B} = \mathbf{X X}^T \) is positive semidefinite, so it can be expressed as its spectral decomposition:

\[ \mathbf{B} = \mathbf{C D C}^T = \mathbf{C D}^{1/2} \mathbf{D}^{1/2} \mathbf{C}^T = (\mathbf{C D}^{1/2})(\mathbf{C D}^{1/2})^T \]

where \( \mathbf{C} \) is the matrix whose columns are the eigenvectors of \( \mathbf{B} \) and \( \mathbf{D}^{1/2} \) is the diagonal matrix whose diagonal elements are the square roots of the eigenvalues.
If we ignore the eigenvector of $\mathbf{B}$ with very small eigenvalues (the eigenvalues of $\mathbf{B} = \mathbf{XX}^T$ are the same as the eigenvalues of $\mathbf{X}^T \mathbf{X}$), $\mathbf{CD}^{1/2}$ will only be a low-rank approximation of $\mathbf{X}$.

Let $\mathbf{c}_k$, $k = 1, \ldots, K$ be the eigenvectors chosen with corresponding eigenvalues $\lambda_k$, $k = 1, \ldots, K$.

New dimensions in $K$-dimensional embedding space:

$$z_k^{(i)} = \sqrt{\lambda_k} c_k^{(i)}$$

So the new coordinates of instance $i$ are given by the $i$th elements of the eigenvectors after normalization.
Unlike PCA, FA and MDS, LDA is a *supervised dimensionality reduction* method.

LDA is typically used with a *classifier* for classification problems.

Goal: the classes are *well-separated* after projecting to a low-dimensional space by utilizing the label information (output information).
Example

Optdigits after LDA
2-Class Case

- Given sample $\mathcal{X} = \{(x^{(i)}, y^{(i)})\}$, where $y^{(i)} = 1$ if $x^{(i)} \in C_1$ and $y^{(i)} = 0$ if $x^{(i)} \in C_2$.
- Find vector $\mathbf{w}$ on which the data are projected such that the examples from $C_1$ and $C_2$ are as well separated as possible.
- Projection of $\mathbf{x}$ onto $\mathbf{w}$ (dimensionality reduced from $D$ to 1):

  $$z = \mathbf{w}^T \mathbf{x}$$

- $\mathbf{m}_j \in \mathbb{R}^D$ and $m_j \in \mathbb{R}$ are sample means of $C_j$ before and after projection:

  $$m_1 = \frac{\sum_i \mathbf{w}^T \mathbf{x}^{(i)} y^{(i)}}{\sum_i y^{(i)}} = \mathbf{w}^T \mathbf{m}_1$$
  $$m_2 = \frac{\sum_i \mathbf{w}^T \mathbf{x}^{(i)} (1 - y^{(i)})}{\sum_i (1 - y^{(i)})} = \mathbf{w}^T \mathbf{m}_2$$

Hong Chang (ICT, CAS)
Projection

Hong Chang (ICT, CAS)
Linear Dimensionality Reduction
Between-Class Scatter

Between-class scatter:

\[
(m_1 - m_2)^2 = (w^T m_1 - w^T m_2)^2 = w^T (m_1 - m_2)(m_1 - m_2)^T w = w^T S_B w
\]

where \( S_B = (m_1 - m_2)(m_1 - m_2)^T \)
Within-class scatter:

\[ s_1^2 = \sum_i (w^T x^{(i)} - m_1)^2 y^{(i)} \]
\[ = \sum_i w^T (x^{(i)} - m_1)(x^{(i)} - m_1)^T w y^{(i)} \]
\[ = w^T S_1 w \]

where \( S_1 = \sum_i (x^{(i)} - m_1)(x^{(i)} - m_1)^T y^{(i)} \). Similarly, \( s_2^2 = w^T S_2 w \) with \( S_2 = \sum_i (x^{(i)} - m_2)(x^{(i)} - m_2)^T (1 - y^{(i)}) \).

So

\[ s_1^2 + s_2^2 = w^T S_W w \]

where \( S_W = S_1 + S_2 \).
Fisher’s Linear Discriminant

- Fisher’s linear discriminant refers to the vector $\mathbf{w}$ that maximizes the Fisher criterion (a.k.a. generalized Rayleigh quotient):

$$J(\mathbf{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2} = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

- Taking the derivative of $J$ w.r.t. $\mathbf{w}$ and setting it to 0, we obtain the following generalized eigenvalue problem:

$$\mathbf{S}_B \mathbf{w} = \lambda \mathbf{S}_W \mathbf{w}$$

or, if $\mathbf{S}_W$ is nonsingular, an equivalent eigenvalue problem:

$$\mathbf{S}_W^{-1} \mathbf{S}_B \mathbf{w} = \lambda \mathbf{w}$$
Fisher’s Linear Discriminant (2)

- Alternatively, for the 2-class case, we note that

\[ S_B w = (m_1 - m_2)(m_1 - m_2)^T w = c(m_1 - m_2) \]

for some constant \( c \) and hence \( S_B w \) is in the same direction of \( m_1 - m_2 \).

- So we get

\[ w = S_\omega^{-1}(m_1 - m_2) = (S_1 + S_2)^{-1}(m_1 - m_2) \]

The constant factor is irrelevant and hence is discarded.
$K > 2$ Classes

- Find the matrix $\mathbf{W} \in \mathbb{R}^{D \times K}$ such that
  \[ z = \mathbf{W}^T \mathbf{x} \in \mathbb{R}^K \]

- Within-class scatter matrix for class $C_k$:
  \[ S_k = \sum_i y_k^{(i)} (\mathbf{x}^{(i)} - \mathbf{m}_k)(\mathbf{x}^{(i)} - \mathbf{m}_k)^T \]
  where $y_k^{(i)} = 1$ if $\mathbf{x}^{(i)} \in C_k$ and 0 otherwise.

- Total within-class scatter matrix:
  \[ S_W = \sum_{k=1}^{K} S_k \]
Between-class scatter matrix:
\[ S_B = \sum_{k=1}^{K} N_k (m_k - m)(m_k - m)^T \]

where \( m \) is the overall mean and \( N_k = \sum_i y_k^{(i)} \).

The optimal solution is the matrix \( W \) that maximizes
\[ J(W) = \frac{\text{Tr}(W^T S_B W)}{\text{Tr}(W^T S_W W)} \]

which corresponds to the eigenvectors of \( S_W^{-1} S_B \) with the largest eigenvalues.

Take new dimensionality \( d \leq K - 1 \): since \( S_W \) is the sum of \( K \) rank-1 matrices and only \( K - 1 \) of them are independent, \( S_B \) has a maximum rank of \( K - 1 \).
PCA (Eigenface) maps features to a subspace that contains most energy.

FLD (Fisherface) maps features to a subspace that most separate the classes.
PCA is an unsupervised dimension reduction algorithm, while LDA is supervised. PCA is good at outlier cleaning, and LDA could learn the within-class deviation. These two methods only extract 1st and 2nd statistical moments. The combination of PCA and LDA could enhance the performance. PCA serves as the first-step processing of several kinds of face recognition technique. Techniques of dimension reduction are frequently used in face recognition.