Principal Component Analysis and Linear Discriminant Analysis

Ying Wu

Electrical Engineering and Computer Science
Northwestern University
Evanston, IL 60208

http://www.eecs.northwestern.edu/~yingwu
Decomposition and Components

- Decomposition is a great idea.

- Linear decomposition and linear basis, e.g., the Fourier transform

- The bases
  - construct the feature space
  - may be orthogonal bases, may be not
  - give the direction to find the components
  - specified vs. learnt?

- The features
  - are the “image” (or projection) of the original signal in the feature space
  - e.g., the orthogonal projection of the original signal onto the feature space
  - the projection does not have to be orthogonal

- Feature extraction
Outline

Principal Component Analysis

Linear Discriminant Analysis

Comparison between PCA and LDA
Principal Components and Subspaces

- Subspaces preserve part of the information (and energy, or uncertainty)
- Principal components
  - are orthogonal bases
  - and preserve the large portion of the information of the data
  - capture the major uncertainties (or variations) of data
- Two views
  - Deterministic: minimizing the distortion of projection of the data
  - Statistical: maximizing the uncertainty in data
  - are they the same?
  - under what condition they are the same?
View 1: Minimizing the MSE

- $x \in \mathbb{R}^n$, and assume centering $E\{x\} = 0$.
- $m$ is the dim of the subspace, $m < n$.
- orthonormal bases $W = [w_1, w_2, \ldots, w_m]$.
- where $W^T W = I$, i.e., rotation.
- orthogonal projection of $x$:
  \[
  P_x = \sum_{i=1}^{m} (w_i^T x) w_i = (W W^T) x
  \]
- it achieves the minimum mean-square error (prove it!)

\[
e_{MSE}^{\min} = E\{||x - P_x||^2\} = E\{||P^\perp x||\}
\]

PCA can be posed as: finding a subspace that minimizes the MSE:

\[
\arg\min_W J_{MSE}(W) = E\{||x - P_x||^2\}, \text{ s.t., } W^T W = I
\]
Let do it...

It is easy to see:

\[ J_{MSE}(W) = E\{x^T x\} - E\{x^T Px\} \]

So,

\( \text{minimizing } J_{MSE}(W) \rightarrow \text{maximizing } E\{x^T Px\} \)

Then we have the following constrained optimization problem

\[
\max_W E\{x^T WW^T x\} \quad s.t. \quad W^T W = I
\]

The Lagrangian is

\[
L(W, \lambda) = E\{x^T WW^T x\} + \lambda^T (I - W^T W)
\]

The set of KKT conditions gives:

\[
\frac{\partial L(W, \lambda)}{\partial w_i} = 2E\{xx^T\}w_i - 2\lambda_i w_i, \quad \forall i
\]
Let's denote by $S = E\{xx^T\}$ (note: $E\{x\} = 0$). The KKT conditions give:

\[ Sw_i = \lambda_i w_i, \quad \forall i \]

or in a more concise matrix form:

\[ SW = \lambda_i W \]

What is this?
Then, the value of minimum MSE is

\[ e_{\text{MSE}}^{\text{min}} = \sum_{i=m+1}^{n} \lambda_i \]

i.e., the sum of the eigenvalues of the orthogonal subspace to the PCA subspace.
View 2: Maximizing the Variation

Let’s look it from another perspective:

- We have a linear projection of $\mathbf{x}$ to a 1-d subspace $y = \mathbf{w}^T \mathbf{x}$
- an important note: $E\{y\} = 0$ as $E\{\mathbf{x}\} = 0$
- The first principal component of $\mathbf{x}$ is such that the variance of the projection $y$ is maximized
- of course, we need to constrain $\mathbf{w}$ to be a unit vector.
- so we have the following optimization problem

$$\max_{\mathbf{w}} J(\mathbf{w}) = E\{y^2\} = E\{(\mathbf{w}^T \mathbf{x})^2\}, \quad s.t. \quad \mathbf{w}^T \mathbf{w} = 1$$

- what is it?

$$\max_{\mathbf{w}} J(\mathbf{w}) = \mathbf{w}^T \mathbf{S} \mathbf{w}, \quad s.t. \quad \mathbf{w}^T \mathbf{w} = 1$$
The sorted eigenvalues of $\mathbf{S}$ are $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$, and eigenvectors are $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$.

It is clearly that the first PC is $y_1 = \mathbf{e}_1^T \mathbf{x}$

This can be generalized to $m$ PCs (where $m < n$) with one more constraint

$$E\{y_my_k\} = 0, \quad k < m$$

i.e., the PCs are uncorrelated with all previously found PCs

The solution is:

$$\mathbf{w}_k = \mathbf{e}_k$$

Sounds familiar?
The Two Views Converge

The two views lead to the same result!

- You should prove:

$$\text{uncorrelated components} \iff \text{orthonormal projection bases}$$

- What if we are more greedy, say needing independent components?
- Do we shall expect orthonormal bases?
- In which case, we still have orthonormal bases?
- We’ll see it in next lecture.
The Closed-Form Solution

Learning the principal components from \{x_1, \ldots, x_N\}:

1. calculating \( m = \frac{1}{N} \sum_{k=1}^{N} x_k \)
2. centering \( A = [x_1 - m, \ldots, x_N - m] \)
3. calculating \( S = \sum_{k=1}^{N} (x_k - m)(x_k - m)^T = AA^T \)
4. eigenvalue decomposition
   \[ S = U^T \Sigma U \]
5. sorting \( \lambda_i \) and \( e_i \)
6. finding the bases
   \[ W = [e_1, e_2, \ldots, e_m] \]

Note: The components for \( x \) is
\( y = W^T (x - m) \), where \( x \in \mathbb{R}^n \) and \( y \in \mathbb{R}^m \)
A First Issue

- $n$ is the dimension of input data, $N$ is the size of the training set
- In practice, $n \gg N$
  - E.g., in image-based face recognition, if the resolution of a face image is $100 \times 100$, when stacking all the pixels, we end up $n = 10,000$.
- Note that $\mathbf{S}$ is a $n \times n$ matrix
- Difficulties:
  - $\mathbf{S}$ is ill-conditioned, as in general $\text{rank}(\mathbf{S}) \ll n$
  - Eigenvalue decomposition of $\mathbf{S}$ is too demanding
- So, what should we do?
Solution I: First Trick

- $A$ is a $n \times N$ matrix, then $S = AA^T$ is $n \times n$,
- but $A^T A$ is $N \times N$
- **Trick**
  - Let’s do eigenvalue decomposition on $A^T A$
  - $A^T A e = \lambda e \implies AA^T A e = \lambda A e$
  - i.e., if $e$ is an eigenvector of $A^T A$, then $A e$ is the eigenvector of $AA^T$
  - and the corresponding eigenvalues are the same
- Don’t forget to normalize $A e$

**Note:** This trick does not fully solved the problem, as we still need to do eigenvalue decomposition on a $N \times N$ matrix, which can be fairly large in practice.
Solution II: Using SVD

- Instead of doing EVD, doing SVD (singular value decomposition) is easier
- \( A \in \mathbb{R}^{n \times N} \)
- \( A = U \Sigma V^T \)
  - \( U \in \mathbb{R}^{n \times N} \), and \( U^T U = I \)
  - \( \Sigma \in \mathbb{R}^{N \times N} \), is diagonal
  - \( V \in \mathbb{R}^{N \times N} \), and \( V^T V = VV^T = I \)
Solution III: Iterative Solution

- We can design an iterative procedure for finding $W$, i.e.,
  \[ W \leftarrow W + \Delta W \]
- looking at the View of MSE minimization, our cost function:
  \[
  \| x - \sum_{i=1}^{m} (w_i^T x)w_i \|^2 = \| x - \sum_{i=1}^{m} y_i w_i \|^2 = \| x - (WW^T)x \|^2
  \]
- we can stop updating if the KKT is met
  \[
  \Delta w_i = \gamma y_i [x - \sum_{i=1}^{m} y_i w_i]
  \]
- Its matrix form is: $\leftarrow$ subspace learning algorithm
  \[
  \Delta W = \gamma (xx^T W - WW^T xx^T W)
  \]
- Two issues:
  - The orthogonality is not reinforced
  - Slow convergence
Solution IV: PAST

To speed up the iteration, we can use recursive least squares (RLS). We can consider the following cost function

\[
J(t) = \sum_{i=1}^{t} \beta_{t-i} \|x(i) - W(t)y(i)\|^2
\]

where \( \beta \) is the forgetting factor.

\( W \) can be solved recursively by the following PAST algorithm

1. \( y(t) = W^T(t-1)x(t) \)
2. \( h(t) = P(t-1)y(t) \)
3. \( m(t) = h(t)/(\beta + y^T(t)h(t)) \)
4. \( P(t) = \frac{1}{\beta} \text{Tri}[P(t-1) - m(t)h^T(t)] \)
5. \( e(t) = x(t) - W(t-1)y(t) \)
6. \( W(t) = W(t-1) + e(t)m^T(t) \)
Outline

Principal Component Analysis

Linear Discriminant Analysis

Comparison between PCA and LDA
Face Recognition: Does PCA work well?

- The same face under different illumination conditions
- What does PCA capture?
- Is this what we really want?
From Descriptive to Discriminative

- PCA extracts features (or components) that well describe the pattern.
- Are they necessarily good for distinguishing between classes and separating patterns?
- Examples?
- We need discriminative features.
- Supervision (or labeled training data) is needed.
- The issues are:
  - How do we define the discriminant and separability between classes?
  - How many features do we need?
  - How do we maximizing the separability?
- Here, we give an example of linear discriminant analysis.
Linear Discriminant Analysis

- Finding an optimal linear projection $\mathbf{W}$
- Catches major difference between classes and discount irrelevant factors
- In the projected discriminative subspace, data are clustered
Within-class and Between-class Scatters

We have two sets of labeled data: $\mathcal{D}_1 = \{x_1, \ldots, x_{n_1}\}$ and $\mathcal{D}_2 = \{x_1, \ldots, x_{n_2}\}$. Let’s define some terms:

- The centers of two classes, $m_i = \frac{1}{n_i} \sum_{x \in \mathcal{D}_i} x$
- Data scatter by definition
  \[ S = \sum_{x \in \mathcal{D}} (x - m)(x - m)^T \]

- Within-class scatter:
  \[ S_w = S_1 + S_2 \]

- Between-class scatter:
  \[ S_b = (m_1 - m_2)(m_1 - m_2)^T \]
Fisher Liner Discriminant

Input: We have two sets of labeled data: $D_1 = \{x_1, \ldots, x_{n_1}\}$ and $D_2 = \{x_1, \ldots, x_{n_2}\}$.
Output: We want to find a 1-d linear projection $w$ that maximizes the separability between these two classes.

- Projected data: $Y_1 = w^T D_1$ and $Y_2 = w^T D_2$
- Projected class centers: $\tilde{m}_i = w^T m_i$
- Projected within-class scatter (it is a scalar in this case)
  \[ \tilde{S}_w = w^T S_w w \quad \text{prove it!} \]
- Projected between-class scatter (it is a scalar in this case)
  \[ \tilde{S}_b = w^T S_b w \quad \text{prove it!} \]
- Fisher Linear Discriminant
  \[ J(w) = \frac{\left|\tilde{m}_1 - \tilde{m}_2\right|^2}{\tilde{S}_1 + \tilde{S}_2} = \frac{\tilde{S}_b}{\tilde{S}_w} = \frac{w^T S_b w}{w^T S_w w} \]
Rayleigh Quotient

**Theorem**

\[ f(\lambda) = \|Ax - \lambda Bx\|_B \quad \text{where} \quad \|z\|_B \overset{\triangle}{=} z^T B^{-1} z \quad \text{is minimized by the Rayleigh quotient} \]

\[ \lambda = \frac{x^T Ax}{x^T B x} \]

**Proof.**

\[
\frac{\partial f(\lambda)}{\partial \lambda} = (Bx)^T (Bz) = x^T B^T B^{-1} z
\]

\[ = x^T (Ax - \lambda Bx) = x^T Ax - \lambda x^T B x \]

setting it to zero to see the result clearly.
Optimizing Fish Discriminant

Theorem

\[ J(w) = \frac{w^T S_b w}{w^T S_w w} \text{ is maximized when} \]

\[ S_b w = \lambda S_w w \]

Proof.

Let \( w^T S_w w = c \neq 0 \). We can construct the Lagrangian as

\[ L(w, \lambda) = w^T S_b w - \lambda (w^T S_w w - c) \]

Then KKT is

\[ \frac{\partial L(w, \lambda)}{\partial w} = S_b w - \lambda S_w w \]

It is clearly that

\[ S_b w^* = \lambda S_w w^* \]
A naive solution is $S_w^{-1}S_bw = \lambda w$

Then we can do EVD on $S_w^{-1}S_b$, which needs some computation.

Is there a more efficient way?

Facts:
- $S_bw$ is along the direction of $m_1 - m_2$. why?
- we don’t care about $\lambda$ the scalar factor.

So we can easily figure out the direction of $w$ by

$$w = S_w^{-1}(m_1 - m_2)$$

Note: $rank(S_b) = 1$
Multiple Discriminant Analysis

Now, we have $c$ number of classes:

- within-class scatter $S_w = \sum_{i=1}^{c} S_i$ as before
- between-class scatter is a bit different from 2-class

$$S_b \triangleq \sum_{i=1}^{c} n_i (m_i - m)(m_i - m)^T$$

- total scatter

$$S_t \triangleq \sum_{x} (x - m)(x - m)^T = S_w + S_b$$

- MDA is to find a subspace with bases $W$ that maximizes

$$J(W) = \frac{|\tilde{S}_b|}{|\tilde{S}_w|} = \frac{|W^T S_b W|}{|W^T S_w W|}$$
The Solution to MDA

- The solution is obtained by G-EVD

\[ S_b w_i = \lambda_i S_w w_i \]

where each \( w_i \) is a generalized eigenvector

- In practice, what we can do is the following
  - find the eigenvalues as the root of the characteristic polynomial
    \[ |S_b - \lambda_i S_w| = 0 \]
  - for each \( \lambda_i \), solve \( w_i \) from
    \[ (S_b - \lambda_i S_w)w_i = 0 \]

- Note: \( W \) is not unique (up to rotation and scaling)
- Note: \( \text{rank}(S_b) \leq (c - 1) \) (why?)
Outline

Principal Component Analysis

Linear Discriminant Analysis

Comparison between PCA and LDA
The Relation between PCA and LDA