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

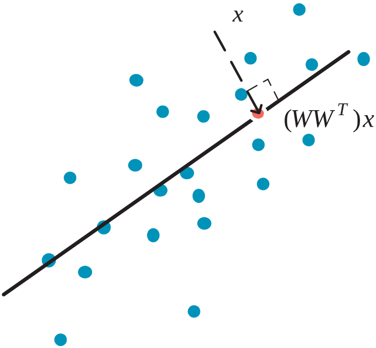
- Decomposition is a great idea.
- Linear decomposition and linear basis, e.g., the Fourier transform
- The bases
  - construct the feature space
  - may be orthogonal bases, may be not
  - give the direction to find the components
  - specified vs. learnt?
- The features
  - are the "image" (or projection) 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 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



- $\mathbf{x} \in \mathbb{R}^n$, and assume centering $E\{\mathbf{x}\} = \mathbf{0}$.
- m is the dim of the subspace, $m < n$
- orthonormal bases $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m]$
- where $\mathbf{W}^T\mathbf{W} = \mathbf{I}$, i.e., rotation
- orthogonal projection of $\mathbf{x}$:

$$\mathbf{Px} = \sum_{i=1}^{m} (\mathbf{w}_i^T \mathbf{x})\mathbf{w}_i = (\mathbf{W}\mathbf{W}^T)\mathbf{x}$$

- it achieves the minimum mean-square error (prove it!)

$$e_{MSE}^{min} = E\{||\mathbf{x} - \mathbf{Px}||^2\} = E\{||\mathbf{P}^{\perp}\mathbf{x}||\}$$

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

$$\underset{\mathbf{W}}{argmin}\, J_{MSE}(\mathbf{W}) = E\{||\mathbf{x} - \mathbf{Px}||^2\},\ s.t.,\ \mathbf{W}^T\mathbf{W} = \mathbf{I}$$

Let do it...

It is easy to see:

$$J_{MSE}(\mathbf{W}) = E\{\mathbf{x}^T\mathbf{x}\} - E\{\mathbf{x}^T\mathbf{P}\mathbf{x}\}$$

So,

$$\textit{minimizing } J_{MSE}(\mathbf{W}) \longrightarrow \textit{ maximizing } E\{\mathbf{x}^T\mathbf{P}\mathbf{x}\}$$

Then we have the following constrained optimization problem

$$\max_{W} E\{\mathbf{x}^T\mathbf{W}\mathbf{W}^T\mathbf{x}\} \; s.t. \; \mathbf{W}^T\mathbf{W} = \mathbf{I}$$

The Lagrangian is

$$L(\mathbf{W}, \lambda) = E\{\mathbf{x}^T\mathbf{W}\mathbf{W}^T\mathbf{x}\} + \lambda^T(\mathbf{I} - \mathbf{W}^T\mathbf{W})$$

The set of KKT conditions gives:

$$\frac{\partial L(\mathbf{W}, \lambda)}{\partial \mathbf{w}_i} = 2E\{\mathbf{x}\mathbf{x}^T\}\mathbf{w}_i - 2\lambda_i\mathbf{w}_i, \quad \forall i$$

What is it?

Let's denote by $\mathbf{S} = E\{\mathbf{x}\mathbf{x}^T\}$ (note: $E\{\mathbf{x}\} = 0$).
The KKT conditions give:

$$\mathbf{S}\mathbf{w}_i = \lambda_i \mathbf{w}_i, \quad \forall i$$

or in a more concise matrix form:

$$\mathbf{S}\mathbf{W} = \lambda_i \mathbf{W}$$

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

$$e_{MSE}^{min} = \sum_{i=m+1}^{n} \lambda_i$$

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 $\mathbf{y}$ is maximized
- of course, we need to constrain $\mathbf{w}$ to be a unit vector.
- so we have the following optimization problem

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

- what is it?

$$\max_{w} J(\mathbf{w}) = \mathbf{w}^T\mathbf{S}\mathbf{w}, \quad s.t.\ \mathbf{w}^T\mathbf{w} = 1$$

Maximizing the Variation (cont.)

- 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

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

  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:

$$\boxed{\textit{uncorrelated components} \iff \textit{orthonormal projection bases}}$$

- 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 $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$:

1. calculating $\mathbf{m} = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k$
2. centering $\mathbf{A} = [\mathbf{x}_1 - \mathbf{m}, \ldots, \mathbf{x}_N - \mathbf{m}]$
3. calculating $\mathbf{S} = \sum_{k=1}^{N}(\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^T = \mathbf{A}\mathbf{A}^T$
4. eigenvalue decomposition

$$\mathbf{S} = \mathbf{U}^T\Sigma\mathbf{U}$$

5. sorting λ_i and $\mathbf{e}_i$
6. finding the bases

$$\mathbf{W} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m]$$

Note: The components for $\mathbf{x}$ is

$$\mathbf{y} = \mathbf{W}^T(\mathbf{x} - \mathbf{m}), \ \textit{where } \mathbf{x} \in \mathbb{R}^n \textit{ and } \mathbf{y} \in \mathbb{R}^m$$

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 100×100, 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 $rank(\mathbf{S}) \ll n$
  - 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 $\mathbf{S} = \mathbf{A}\mathbf{A}^T$ is $n \times n$,
- but $\mathbf{A}^T\mathbf{A}$ is $N \times N$
- Trick
  - Let's do eigenvalue decomposition on $\mathbf{A}^T\mathbf{A}$
  - $\mathbf{A}^T\mathbf{A}\mathbf{e} = \lambda\mathbf{e} \longrightarrow \mathbf{A}\mathbf{A}^T\mathbf{A}\mathbf{e} = \lambda\mathbf{A}\mathbf{e}$
  - i.e., if $\mathbf{e}$ is an eigenvector of $\mathbf{A}^T\mathbf{A}$, then $\mathbf{A}\mathbf{e}$ is the eigenvector of $\mathbf{A}\mathbf{A}^T$
  - and the corresponding eigenvalues are the same
- Don't forget to normalize $\mathbf{A}\mathbf{e}$

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
- $\mathbf{A} \in \mathbb{R}^n \times \mathbb{R}^N$
- $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$
  - $\mathbf{U} \in \mathbb{R}^n \times \mathbb{R}^N$, and $\mathbf{U}^T\mathbf{U} = \mathbf{I}$
  - $\Sigma \in \mathbb{R}^N \times \mathbb{R}^N$, is diagonal
  - $\mathbf{V} \in \mathbb{R}^N \times \mathbb{R}^N$, and $\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}$

Solution III: Iterative Solution

- We can design an iterative procedure for finding $\mathbf{W}$, i.e., $\mathbf{W} \leftarrow \mathbf{W} + \Delta\mathbf{W}$
- looking at the View of MSE minimization, our cost function:

$$||\mathbf{x} - \sum_{i=1}^{m}(\mathbf{w}_i^T\mathbf{x})\mathbf{w}_i||^2 = ||\mathbf{x} - \sum_{i=1}^{m} y_i\mathbf{w}_i||^2 = ||\mathbf{x} - (\mathbf{W}\mathbf{W}^T)\mathbf{x}||^2$$

- we can stop updating if the KKT is met

$$\Delta\mathbf{w}_i = \gamma y_i[\mathbf{x} - \sum_{i=1}^{m} y_i\mathbf{w}_i]$$

- Its matrix form is: $\leftarrow$ subspace learning algorithm

$$\Delta\mathbf{W} = \gamma(\mathbf{x}\mathbf{x}^T\mathbf{W} - \mathbf{W}\mathbf{W}^T\mathbf{x}\mathbf{x}^T\mathbf{W})$$

- 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

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

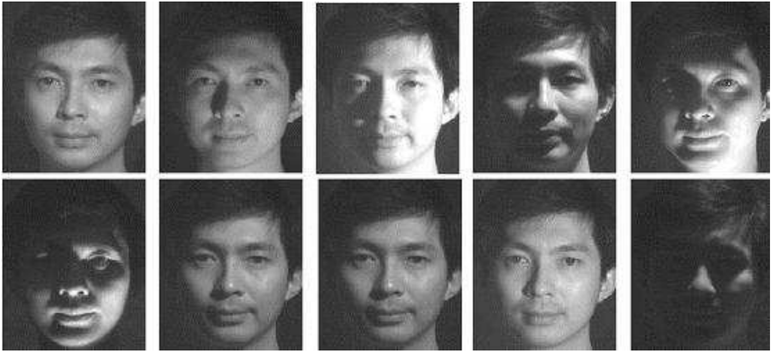
where β is the forgetting factor.

W can be solved recursively by the following PAST algorithm

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

Outline

Face Recognition: Does PCA work well?

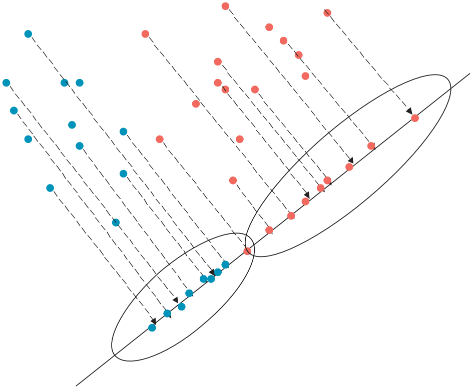


- The same face under different illumination conditions
- What does PCA capture?
- Is this what we really want?

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 **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: $\mathcal{D}_1 = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_1}\}$ and $\mathcal{D}_2 = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_2}\}$. Let's define some terms:

- The centers of two classes, $\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \mathcal{D}_i} \mathbf{x}_i$
- Data scatter by definition

$$\mathbf{S} = \sum_{x \in \mathcal{D}} (\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T$$

- **Within-class scatter**:

$$\mathbf{S}_w = \mathbf{S}_1 + \mathbf{S}_2$$

- **Between-class scatter**:

$$\mathbf{S}_b = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T$$

Fisher Liner Discriminant

Input: We have two sets of labeled data: $\mathcal{D}_1 = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_1}\}$ and $\mathcal{D}_2 = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_2}\}$.

Output: We want to find a 1-d linear projection $\mathbf{w}$ that maximizes the separability between these two classes.

- Projected data: $\mathcal{Y}_1 = \mathbf{w}^T\mathcal{D}_1$ and $\mathcal{Y}_2 = \mathbf{w}^T\mathcal{D}_2$
- Projected class centers: $\tilde{m}_i = \mathbf{w}^T\mathbf{m}_i$
- Projected within-class scatter (it is a scalar in this case)

$$\tilde{\mathbf{S}}_w = \mathbf{w}^T\mathbf{S}_w\mathbf{w} \quad \textit{prove it!}$$

- Projected between-class scatter (it is a scalar in this case)

$$\tilde{\mathbf{S}}_b = \mathbf{w}^T\mathbf{S}_b\mathbf{w} \quad \textit{prove it!}$$

- Fisher Linear Discriminant

$$J(\mathbf{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{\mathbf{S}}_1 + \tilde{\mathbf{S}}_2} = \frac{|\tilde{\mathbf{S}}_b|}{|\tilde{\mathbf{S}}_w|} = \frac{\mathbf{w}^T\mathbf{S}_b\mathbf{w}}{\mathbf{w}^T\mathbf{S}_w\mathbf{w}}$$

Rayleigh Quotient

Theorem

$f(\lambda) = ||\mathbf{Ax} - \lambda\mathbf{Bx}||_B$ where $||\mathbf{z}||_B \stackrel{\triangle}{=} \mathbf{z}^T\mathbf{B}^{-1}\mathbf{z}$ is minimized by the Rayleigh quotient

$$\lambda = \frac{\mathbf{x}^T\mathbf{Ax}}{\mathbf{x}^T\mathbf{Bx}}$$

Proof.

$$\begin{aligned} \frac{\partial f(\lambda)}{\partial \lambda} &= (\mathbf{Bx})^T(\mathbf{Bz}) = \mathbf{x}^T\mathbf{B}^T\mathbf{B}^{-1}\mathbf{z} \\ &= \mathbf{x}^T(\mathbf{Ax} - \lambda\mathbf{Bx}) = \mathbf{x}^T\mathbf{Ax} - \lambda\mathbf{x}^T\mathbf{Bx} \end{aligned}$$

setting it to zero to see the result clearly. □

Optimizing Fish Discriminant

Theorem

$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}}$ *is maximized when*

$$\mathbf{S}_b \mathbf{w} = \lambda \mathbf{S}_w \mathbf{w}$$

Proof.

Let $\mathbf{w}^T \mathbf{S}_w \mathbf{w} = c \neq 0$. We can construct the Lagrangian as

$$L(\mathbf{w}, \lambda) = \mathbf{w}^T \mathbf{S}_b \mathbf{w} - \lambda(\mathbf{w}^T \mathbf{S}_w \mathbf{w} - c)$$

Then KKT is

$$\frac{\partial L(\mathbf{w}, \lambda)}{\partial \mathbf{w}} = \mathbf{S}_b \mathbf{w} - \lambda \mathbf{S}_w \mathbf{w}$$

It is clearly that

$$\mathbf{S}_b \mathbf{w}^* = \lambda \mathbf{S}_w \mathbf{w}^*$$

□

An Efficient Solution

- A naive solution is $\mathbf{S}_w^{-1}\mathbf{S}_b\mathbf{w} = \lambda\mathbf{w}$
- Then we can do EVD on $\mathbf{S}_w^{-1}\mathbf{S}_b$, which needs some computation
- Is there a more efficient way?
- Facts:
  - $\mathbf{S}_b\mathbf{w}$ is along the direction of $\mathbf{m}_1 - \mathbf{m}_2$. why?
  - we don't care about λ the scalar factor
- So we can easily figure out the direction of $\mathbf{w}$ by

$$\mathbf{w} = \mathbf{S}_w^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$

- Note: $rank(\mathbf{S}_b) = 1$

Multiple Discriminant Analysis

Now, we have c number of classes:

- within-class scatter $\mathbf{S}_w = \sum_{i=1}^{c} \mathbf{S}_i$ as before
- between-class scatter is a bit different from 2-class

$$\mathbf{S}_b \triangleq \sum_{i=1}^{c} n_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T$$

- total scatter

$$\mathbf{S}_t \triangleq \sum_{x} (\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T = \mathbf{S}_w + \mathbf{S}_b$$

- MDA is to find a subspace with bases $\mathbf{W}$ that maximizes

$$J(\mathbf{W}) = \frac{|\tilde{\mathbf{S}}_b|}{|\tilde{\mathbf{S}}_w|} = \frac{|\mathbf{W}^T \mathbf{S}_b \mathbf{W}|}{|\mathbf{W}^T \mathbf{S}_w \mathbf{W}|}$$

The Solution to MDA

- The solution is obtained by G-EVD

$$\mathbf{S}_b \mathbf{w}_i = \lambda_i \mathbf{S}_w \mathbf{w}_i$$

  where each $\mathbf{w}_i$ is a generalized eigenvector
- In practice, what we can do is the following
  - find the eigenvalues as the root of the characteristic polynomial

$$|\mathbf{S}_b - \lambda_i \mathbf{S}_w| = 0$$

  - for each λ_i, solve $\mathbf{w}_i$ from

$$(\mathbf{S}_b - \lambda_i \mathbf{S}_w)\mathbf{w}_i = 0$$

- Note: $\mathbf{W}$ is not unique (up to rotation and scaling)
- Note: $rank(\mathbf{S}_b) \leq (c-1)$ (why?)

Outline

The Relation between PCA and LDA