Principal Component Analysis (PCA)

CE-717: Machine Learning
Sharif University of Technology
Spring 2016

Soleymani
Dimensionality Reduction:
Feature Selection vs. Feature Extraction

- **Feature selection**
  - Select a subset of a given feature set

- **Feature extraction**
  - A linear or non-linear transform on the original feature space

\[
\begin{bmatrix}
    x_1 \\
    \vdots \\
    x_d
\end{bmatrix} \rightarrow \begin{bmatrix}
    x_{i_1} \\
    \vdots \\
    x_{i_{d'}}
\end{bmatrix}
\]

Feature Selection
\((d' < d)\)

\[
\begin{bmatrix}
    x_1 \\
    \vdots \\
    x_d
\end{bmatrix} \rightarrow \begin{bmatrix}
    y_1 \\
    \vdots \\
    y_{d'}
\end{bmatrix} = f\left(\begin{bmatrix}
    x_1 \\
    \vdots \\
    x_d
\end{bmatrix}\right)
\]

Feature Extraction
Feature Extraction

- Mapping of the original data to another space
  - Criterion for feature extraction can be different based on problem settings
    - Unsupervised task: minimize the information loss (reconstruction error)
    - Supervised task: maximize the class discrimination on the projected space

- Feature extraction algorithms
  - Linear Methods
    - Unsupervised: e.g., Principal Component Analysis (PCA)
    - Supervised: e.g., Linear Discriminant Analysis (LDA)
      - Also known as Fisher’s Discriminant Analysis (FDA)
Feature Extraction

- **Unsupervised feature extraction:**

  
  \[ X = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \cdots & x_d^{(N)} \end{bmatrix} \]

  A mapping \( f : \mathbb{R}^d \rightarrow \mathbb{R}^{d'} \)
  Or only the transformed data

  \[ X' = \begin{bmatrix} x'_1^{(1)} & \cdots & x'_d^{(1)} \\ \vdots & \ddots & \vdots \\ x'_1^{(N)} & \cdots & x'_d^{(N)} \end{bmatrix} \]

- **Supervised feature extraction:**

  
  \[ X = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \cdots & x_d^{(N)} \end{bmatrix} \]

  \[ Y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix} \]

  A mapping \( f : \mathbb{R}^d \rightarrow \mathbb{R}^{d'} \)
  Or only the transformed data

  \[ X' = \begin{bmatrix} x'_1^{(1)} & \cdots & x'_d^{(1)} \\ \vdots & \ddots & \vdots \\ x'_1^{(N)} & \cdots & x'_d^{(N)} \end{bmatrix} \]
Unsupervised Feature Reduction

- **Visualization**: projection of high-dimensional data onto 2D or 3D.
- **Data compression**: efficient storage, communication, or retrieval.
- **Pre-process**: to improve accuracy by reducing features
  - As a preprocessing step to reduce dimensions for supervised learning tasks
  - Helps avoiding overfitting
- **Noise removal**
  - E.g., “noise” in the images introduced by minor lighting variations, slightly different imaging conditions, etc.
Linear Transformation

For linear transformation, we find an explicit mapping $f(x) = A^T x$ that can transform also new data vectors.

\[ A^T \in \mathbb{R}^{d' \times d} \]
\[ x \in \mathbb{R} \]
\[ x' = A^T x \]
\[ d' < d \]
Linear Transformation

- Linear transformation are simple mappings

\[ x' = A^T x \]

\[ A = \begin{bmatrix} a_{11} & \cdots & a_{1d'} \\ \vdots & \ddots & \vdots \\ a_{d1} & \cdots & a_{dd'} \end{bmatrix} \]

\[ x'_j = a_j^T x \quad j = 1, \ldots, d' \]
Linear Dimensionality Reduction

- Unsupervised
  - Principal Component Analysis (PCA) [we will discuss]
  - Independent Component Analysis (ICA) [we will discuss]
  - Singular Value Decomposition (SVD)
  - Multi Dimensional Scaling (MDS)
  - Canonical Correlation Analysis (CCA)
Principal Component Analysis (PCA)

- Also known as Karhonen-Loeve (KL) transform

- Principal Components (PCs): orthogonal vectors that are ordered by the fraction of the total information (variation) in the corresponding directions
  - Find the directions at which data approximately lie
    - When the data is projected onto first PC, the variance of the projected data is maximized

- PCA is an orthogonal projection of the data into a subspace so that the variance of the projected data is maximized.
Principal Component Analysis (PCA)

- The “best” linear subspace (i.e. providing least reconstruction error of data):
  - Find mean reduced data
  - The axes have been rotated to new (principal) axes such that:
    - Principal axis 1 has the highest variance
      ....
    - Principal axis i has the i-th highest variance.
  - The principal axes are uncorrelated
    - Covariance among each pair of the principal axes is zero.

- Goal: reducing the dimensionality of the data while preserving the variation present in the dataset as much as possible.

- PCs can be found as the “best” eigenvectors of the covariance matrix of the data points.
Principal components

- If data has a Gaussian distribution $N(\mu, \Sigma)$, the direction of the largest variance can be found by the eigenvector of $\Sigma$ that corresponds to the largest eigenvalue of $\Sigma$.
PCA: Steps

- **Input:** $N \times d$ data matrix $X$ (each row contain a $d$ dimensional data point)
  - $\mu = \frac{1}{N} \sum_{i=1}^{N} x^{(i)}$
  - $\tilde{X} \leftarrow$ Mean value of data points is subtracted from rows of $X$
  - $C = \frac{1}{N} \tilde{X}^T \tilde{X}$ (Covariance matrix)
  - Calculate eigenvalue and eigenvectors of $C$
  - Pick $d'$ eigenvectors corresponding to the largest eigenvalues and put them in the columns of $A = [v_1, \ldots, v_{d'}]$
  - $X' = \tilde{X}A$

First PC  
$d'$-th PC
Covariance Matrix

\[ \mu_x = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_d \end{bmatrix} = \begin{bmatrix} E(x_1) \\ \vdots \\ E(x_d) \end{bmatrix} \]

\[ \Sigma = E[(x - \mu_x)(x - \mu_x)^T] \]

- ML estimate of covariance matrix from data points \( \{x^{(i)}\}_{i=1}^N \):

\[ \hat{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (x^{(i)} - \hat{\mu})(x^{(i)} - \hat{\mu})^T = \frac{1}{N} (\tilde{X}^T \tilde{X}) \]

\[ \tilde{X} = \begin{bmatrix} \tilde{x}^{(1)} \\ \vdots \\ \tilde{x}^{(N)} \end{bmatrix} = \begin{bmatrix} x^{(1)} - \hat{\mu} \\ \vdots \\ x^{(N)} - \hat{\mu} \end{bmatrix} \]

\[ \hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} x^{(i)} \]

Mean-centered data

We now assume that data are mean removed and \( x \) in the later slides is indeed \( \tilde{x} \)
Correlation matrix

\[
\frac{1}{N} X^T X = \frac{1}{N} \begin{bmatrix}
    x_1^{(1)} & \cdots & x_1^{(N)} \\
    \vdots & \ddots & \vdots \\
    x_d^{(1)} & \cdots & x_d^{(N)}
\end{bmatrix}
\begin{bmatrix}
    x_1^{(1)} \\
    \vdots \\
    x_1^{(N)} \\
    \vdots \\
    x_d^{(1)} \\
    \vdots \\
    x_d^{(N)}
\end{bmatrix}
\]

\[
= \frac{1}{N} \begin{bmatrix}
    \sum_{n=1}^{N} x_1^{(n)} x_1^{(n)} & \cdots & \sum_{n=1}^{N} x_1^{(n)} x_d^{(n)} \\
    \vdots & \ddots & \vdots \\
    \sum_{n=1}^{N} x_d^{(n)} x_1^{(n)} & \cdots & \sum_{n=1}^{N} x_d^{(n)} x_d^{(n)}
\end{bmatrix}
\]
Two Interpretations

- **Maximum Variance Subspace**
  - PCA finds vectors $v$ such that projections on to the vectors capture maximum variance in the data
  - \[ \frac{1}{N} \sum_{n=1}^{N} (a^T x^{(n)})^2 = \frac{1}{N} a^T X^T X a \]

- **Minimum Reconstruction Error**
  - PCA finds vectors $v$ such that projection on to the vectors yields minimum MSE reconstruction
  - \[ \frac{1}{N} \sum_{n=1}^{N} \| x^{(n)} - (a^T x^{(n)})a \|^2 \]
Least Squares Error Interpretation

- PCs are linear least squares fits to samples, each orthogonal to the previous PCs:
  - First PC is a minimum distance fit to a vector in the original feature space
  - Second PC is a minimum distance fit to a vector in the plane perpendicular to the first PC
  - And so on
Example
Example
Least Squares Error and Maximum Variance Views Are Equivalent (1-dim Interpretation)

- Minimizing sum of square distances to the line is equivalent to maximizing the sum of squares of the projections on that line (Pythagoras).

\[ \text{red}^2 + \text{blue}^2 = \text{green}^2 \]

\[ \text{green}^2 \text{ is fixed} \text{ (shows the data vector after mean removing)} \]

\[ \Rightarrow \text{maximizing } \text{blue}^2 \text{ is equivalent to minimizing } \text{red}^2 \]
The first PC is direction of greatest variability in data.

We will show that the first PC is the eigenvector of the covariance matrix corresponding the maximum eigen value of this matrix.

If $||a|| = 1$, the projection of a d-dimensional $x$ on $a$ is $a^T x$. 

\[ \|x\| \cos \theta = \|x\| \frac{a^T x}{\|x\| \|a\|} = a^T x \]
First PC

\[
\text{argmax}_a \frac{1}{N} \sum_{n=1}^{N} \left(a^T x^{(n)}\right)^2 = \frac{1}{N} a^T X^T X a \\
\text{s.t. } a^T a = 1
\]

\[
\frac{\partial}{\partial a} \left(\frac{1}{N} a^T X^T X a + \lambda (1 - a^T a)\right) = 0 \Rightarrow \frac{1}{N} X^T X a = \lambda a
\]

- \(a\) is the eigenvector of sample covariance matrix \(\frac{1}{N} X^T X\)

- The eigenvalue \(\lambda\) denotes the amount of variance along that dimension.
  - Variance = \(\frac{1}{N} a^T X^T X a = a^T \left(\frac{1}{N} X^T X a\right) = a^T \lambda a = \lambda\)

- So, if we seek the dimension with the largest variance, it will be the eigenvector corresponding to the largest eigenvalue of the sample covariance matrix
PCA: Uncorrelated Features

\[ x' = A^T x \]

\[ R_{x'} = E[x'x'^T] = E[A^T xx^T A] = A^T E[xx^T] A = A^T R_x A \]

- If \( A = [a_1, ..., a_d] \) where \( a_1, ..., a_d \) are orthonormal eigenvectors of \( R_x \):

\[ R_{x'} = A^T R_x A = A^T (A\Lambda A^T) A = \Lambda \]

\[ \Rightarrow \forall i \neq j (i, j = 1, ..., d) \ E[x'_i x'_j] = 0 \]

then mutually uncorrelated features are obtained

- Completely uncorrelated features avoid information redundancies
PCA Derivation: Mean Square Error Approximation

- Incorporating all eigenvectors in $A = [a_1, \ldots, a_d]$:
  \[
  x' = A^T x \Rightarrow Ax' = AA^T x = x \\
  \Rightarrow x = Ax'
  \]

- If $d' = d$ then $x$ can be reconstructed exactly from $x'$
PCA Derivation: Relation between Eigenvalues and Variances

- The $j$-th largest eigenvalue of $R_x$ is the variance on the $j$-th PC:

$$\text{var}(x'_j) = \lambda_j$$

$$\text{var}(x'_j) = E[x'_j x'_j]$$

$$= E[a_j^T xx^T a_j] = a_j^T E[xx^T] a_j$$

$$= a_j^T R_x a_j = a_j^T \lambda_j a_j = \lambda_j$$

Eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots$

- The 1st PC is the eigenvector of the sample covariance matrix associated with the largest eigenvalue
- The 2nd PC is the eigenvector of the sample covariance matrix associated with the second largest eigenvalue
- And so on …
PCA Derivation: Mean Square Error Approximation

- Incorporating only $d'$ eigenvectors corresponding to the largest eigenvalues $A = [a_1, ..., a_{d'}]$ ($d' < d$)

- It minimizes MSE between $x$ and $\hat{x} = Ax'$:

$$J(A) = E[\|x - \hat{x}\|^2] = E[\|x - Ax'\|^2]$$

$$= E \left[ \left\| \sum_{j=d'+1}^{d} x'_j a_j \right\|^2 \right]$$

$$= E \left[ \sum_{j=d'+1}^{d} \sum_{k=d'+1}^{d} x'_j a_j^T a_k x'_k \right] = E \left[ \sum_{j=d'+1}^{d} x'_j^2 \right]$$

$$= \sum_{j=d'+1}^{d} E \left[ x'_j^2 \right] = \sum_{j=d'+1}^{d} \lambda_j \quad \text{Sum of the } d - d' \text{ smallest eigenvalues}$$
PCA Derivation: Mean Square Error Approximation

- In general, it can also be shown MSE is minimized compared to any other approximation of $x$ by any $d'$-dimensional orthonormal basis
  - without first assuming that the axes are eigenvectors of the correlation matrix, this result can also be obtained.

- If the data is mean-centered in advance, $R_x$ and $C_x$ (covariance matrix) will be the same.
  - However, in the correlation version when $C_x \neq R_x$ the approximation is not, in general, a good one (although it is a minimum MSE solution).
PCA on Faces: “Eigenfaces”

- ORL Database

Some Images
PCA on Faces: “Eigenfaces”

Average face

1st to 10th PCs

For eigen faces
“gray” = 0,
“white” > 0,
“black” < 0
PCA on Faces:

$x$ is a $112 \times 92 = 10304$ dimensional vector containing intensity of the pixels of this image

**Feature vector**

$$[x'_1, x'_2, \ldots, x'_{d'}]$$

$$x'_i = PC_i^T x \quad \text{The projection of } x \text{ on the i-th PC}$$

Average Face

$$= + x'_1 \times + x'_2 \times + \ldots + x'_{256} \times$$

PCA on Faces: Reconstructed Face

\[ d' = 1 \]
\[ d' = 2 \]
\[ d' = 4 \]
\[ d' = 8 \]
\[ d' = 16 \]

\[ d' = 32 \]
\[ d' = 64 \]
\[ d' = 128 \]
\[ d' = 256 \]

Original Image
Dimensionality Reduction by PCA

- In high-dimensional problems, data sometimes lies near a linear subspace (small variability around this subspace can be considered as noise)

- Only keep data projections onto principal components with large eigenvalue

- Might lose some info, but if eigenvalues are small, do not lose much
Kernel PCA

- Kernel extension of PCA

Data (approximately) lies on a lower dimensional non-linear space
PCA and LDA: Drawbacks

- **PCA drawback:** An excellent information packing transform does not necessarily lead to a good class separability.
  - The directions of the maximum variance may be useless for classification purpose

- **LDA drawback**
  - Singularity or under-sampled problem (when $N < d$)
    - Example: gene expression data, images, text documents
  - Can reduces dimension only to $d' \leq C - 1$ (unlike PCA)
PCA vs. LDA

- Although LDA often provide more suitable features for classification tasks, PCA might outperform LDA in some situations:
  - when the number of samples per class is small (overfitting problem of LDA)
  - when the number of the desired features is more than $C - 1$

- Advances in the last decade:
  - Semi-supervised feature extraction
    - E.g., PCA+LDA, Regularized LDA, Locally FDA (LFDA)
Singular Value Decomposition (SVD)

- Given a matrix $X \in \mathbb{R}^{N \times d}$, the SVD is a decomposition:

$$X = U S V^T$$

- $S$ is a diagonal matrix with the singular values $\sigma_1, \ldots, \sigma_d$ of $X$.

- Columns of $U, V$ are orthonormal matrices
Singular Value Decomposition (SVD)

- Given a matrix \( X \in \mathbb{R}^{N \times d} \), the SVD is a decomposition:
  \[
  X = USV^T
  \]

- SVD of \( X \) is related to eigen-decomposition of \( X^T X \) and \( XX^T \).
  - \( X^T X = VSU^T USV^T = VS^2 V^T \)
    - so \( V \) contains eigenvectors of \( X^T X \) and \( S^2 \) includes its eigenvalues \( (\lambda_i = \sigma_i^2) \)
  - \( XX^T = USV^T VSU^T = US^2 U^T \)
    - so \( U \) contains eigenvectors of \( XX^T \) and \( S^2 \) includes its eigenvalues \( (\lambda_i = \sigma_i^2) \)

- In fact, we can view each row of \( US \) as the coordinates of an example along the axes given by the eigenvectors.
Independent Component Analysis (ICA)

- **PCA:**
  - The transformed dimensions will be uncorrelated from each other.
  - Orthogonal linear transform.
  - Only uses second order statistics (i.e., covariance matrix).

- **ICA:**
  - The transformed dimensions will be as independent as possible.
  - Non-orthogonal linear transform.
  - High-order statistics can also be used.
Uncorrelated and Independent

- **Gaussian**
  - Independent $\iff$ Uncorrelated

- **Non-Gaussian**
  - Independent $\Rightarrow$ Uncorrelated
  - Uncorrelated $\not\Rightarrow$ Independent

Uncorrelated: $cov(X_1, X_2) = 0$
Independent: $P(X_1, X_2) = P(X_1)P(X_2)$
ICA: Cocktail party problem

- **Cocktail party problem**
  - $d$ speakers are speaking simultaneously and any microphone records only an overlapping combination of these voices.
    - Each microphone records a different combination of the speakers’ voices.
  - Using these $d$ microphone recordings, can we separate out the original $d$ speakers’ speech signals?

- **Mixing matrix $A$:**
  \[ x = As \]

- **Unmixing matrix $A^{-1}$:**
  \[ s = A^{-1}x \]

$s_j^{(i)}$: sound that speaker $j$ was uttering at time $i$.

$x_j^{(i)}$: acoustic reading recorded by microphone $j$ at time $i$. 
ICA

- Find a linear transformation \( x = As \)
- For which dimensions of \( s = [s_1, s_2, ..., s_d]^T \) are statistically independent
  \[
  p(s_1, ..., s_d) = p_1(s_1)p_2(s_2) ... p_d(s_d)
  \]

- Algorithmically, we need to identify matrix \( A \) and sources \( s \) where \( x = As \) such that the mutual information between \( s_1, s_2, ..., s_d \) is minimized:
  \[
  I(s_1, s_2, ..., s_d) = \sum_{i=1}^{d} H(s_i) - H(s_1, s_2, ..., s_d)
  \]