Feature Extraction (PCA & LDA)
Outline

- What is feature extraction?
- Feature extraction algorithms
  - Linear Methods
    - Unsupervised: Principal Component Analysis (PCA)
      - Also known as Karhonen-Loeve (KL) transform
    - Supervised: Linear Discriminant Analysis (LDA)
      - Also known as Fisher’s Discriminant Analysis (FDA)
Dimensionality Reduction: Feature Selection vs. Feature Extraction

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

- **Feature extraction** (e.g., PCA, LDA)
  - 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 onto a lower-dimensional 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

- In the previous lecture, we talked about feature selection:
  - Feature selection can be considered as a special form of feature extraction (only a subset of the original features are used).
  - Example:

\[
X' = X \times \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \begin{array}{c} \times \in \mathbb{R}^{N \times 4} \\ X' \in \mathbb{R}^{N \times 2} \end{array}
\]

Second and third features are selected
Feature Extraction

- **Unsupervised feature extraction:**
  \[
  X = \begin{bmatrix}
  x_1^{(1)} & \cdots & x_d^{(1)} \\
  \vdots & \ddots & \vdots \\
  x_1^{(N)} & \cdots & x_d^{(N)}
  \end{bmatrix}
  \xrightarrow{\text{Feature Extraction}}
  \begin{bmatrix}
  x_1^{(1)} & \cdots & x_d^{(1)} \\
  \vdots & \ddots & \vdots \\
  x_1^{(N)} & \cdots & x_d^{(N)}
  \end{bmatrix}
  \]
  A mapping \( f : \mathbb{R}^d \to \mathbb{R}^{d'} \)
  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:**
  \[
  X = \begin{bmatrix}
  x_1^{(1)} & \cdots & x_d^{(1)} \\
  \vdots & \ddots & \vdots \\
  x_1^{(N)} & \cdots & x_d^{(N)}
  \end{bmatrix}
  \]
  \[
  Y = \begin{bmatrix}
  y^{(1)} \\
  \vdots \\
  y^{(N)}
  \end{bmatrix}
  \xrightarrow{\text{Feature Extraction}}
  \begin{bmatrix}
  x_1^{(1)} & \cdots & x_d^{(1)} \\
  \vdots & \ddots & \vdots \\
  x_1^{(N)} & \cdots & x_d^{(N)}
  \end{bmatrix}
  \]
  A mapping \( f : \mathbb{R}^d \to \mathbb{R}^{d'} \)
  Or
  only the transformed data
  \[
  X' = \begin{bmatrix}
  x_1^{(1)} & \cdots & x_d^{(1)} \\
  \vdots & \ddots & \vdots \\
  x_1^{(N)} & \cdots & x_d^{(N)}
  \end{bmatrix}
  \]
For linear transformation, we find an explicit mapping $f(x) = A^T x$ that can transform also new data vectors.

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

- Linear transformation are simple mappings

\[ x' = A^T x \quad (x'_{j} = a_{j}^T x) \quad j = 1, \ldots, d \]
Linear Dimensionality Reduction

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

- Supervised
  - Linear Discriminant Analysis (LDA) [we will discuss]
Unsupervised Feature Reduction

- **Visualization**: projection of high-dimensional data onto 2D or 3D.
- **Data compression**: efficient storage, communication, or and retrieval.
- **Noise removal**: to improve accuracy by removing irrelevant features.
  - As a preprocessing step to reduce dimensions for classification tasks
Principal Component Analysis (PCA)

- The “best” subspace:
  - Centered at the sample mean
  - The axes have been rotated to new (principal) axes such that:
    - Principal axis 1 has the highest variance
    - Principal axis 2 has the next highest variance, and so on.
  - 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.
Principal Component Analysis (PCA)

- Principal Components (PCs): orthogonal vectors that are ordered by the fraction of the total information (variation) in the corresponding directions.

- PCs can be found as the “best” eigenvectors of the covariance matrix of the data points.
  - 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 $\Sigma$. 

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 \( \{x^{(i)}\}_{i=1}^N \):

\[ \hat{\Sigma} = \frac{1}{N} (\bar{X}^T \bar{X}) \]

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

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

Mean-centered data
Eigenvalues and Eigenvectors: Geometrical Interpretation

- Covariance matrix is a PSD matrix $\mathbf{C}$
  - corresponding to a hyper-ellipsoidal in an $d$-dimensional space

\[
\frac{\lambda_1}{\lambda_2} = \sqrt{\frac{\lambda_1}{\lambda_2}}
\]

\[
\mathbf{C} = [\mathbf{v}_1 \mathbf{v}_2] \begin{bmatrix}
\lambda_1 & 0 \\
0 & \lambda_2
\end{bmatrix} [\mathbf{v}_1 \mathbf{v}_2]^T
\]
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 $X$
  - $\Sigma = \frac{1}{N} \tilde{X}^T \tilde{X}$ (Covariance matrix)
  - Calculate eigenvalue and eigenvectors of $\Sigma$
  - Pick $d'$ eigenvectors corresponding to the largest eigenvalues and put them in the columns of $A = [v_1, \ldots, v_{d'}]$
  - $X' = A^T X$

First PC $d'$-th PC
Another Interpretation: Least Squares Error

- 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
Another Interpretation: Least Squares Error
Another Interpretation: Least Squares Error
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).
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 \( A = [a_1, \ldots, a_d] \) where \( a_1, \ldots, a_d \) are orthonormal eigenvectors of \( R_x \):

\[ R_{x'} = A^T R_x A = A^T (A^T \Lambda A) A = \Lambda \]
\[ \Rightarrow \forall i \neq j (i, j = 1, \ldots, d) \ E[x'_i x'_j] = 0 \]

then mutually uncorrelated features are obtained

- Completely uncorrelated features avoid information redundancies
PCA Derivation (Correlation Version): Mean Square Error Approximation

- Incorporating all eigenvectors in $A = [a_1, \ldots, a_d]$:

\[ x' = A^T x \Rightarrow Ax' = AA^T x = x \]
\[ \Rightarrow x = A^T x' \]

- $\Rightarrow$ If $d' = d$ then $x$ can be reconstructed exactly from $x'$
PCA Derivation (Correlation Version):
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 $\hat{x} = A^T x'$:

$$J(A) = E[||x - \hat{x}||^2] = E[||x - A^T x'||^2] = E \left[ \left\| \sum_{j=d'+1}^{d} x'_j a_j \right\|^2 \right]$$

$$= E \left[ \sum_{j=d'+1}^{d} \sum_{k=d'+1}^{d} x'_j a^T_j a_k x'_k \right] = E \left[ \sum_{j=d'+1}^{d} x'_j^2 \right] = \sum_{j=d'+1}^{d} E[x'_j^2]$$

$$= \sum_{j=d'+1}^{d} a^T_j E[xx^T]a_j = \sum_{j=d'+1}^{d} a^T_j R_x a_j = \sum_{j=d'+1}^{d} \lambda_j$$

Sum of the $d - d'$ smallest eigenvalues
PCA Derivation (Correlation Version): Relation between Eigenvalues and Variances

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

$$\text{var}(x'_j) = a_j^T R_x a_j = \lambda_j$$
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”

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

$x'_i = P C^T_i x$ → The projection of $x$ on the $i$-th PC

Average Face

$= +x'_1 \times +x'_2 \times + \ldots +x'_{256} \times$
PCA on Faces: Reconstructed Face

\( d' = 1 \) \hspace{1cm} \( d' = 2 \) \hspace{1cm} \( d' = 4 \) \hspace{1cm} \( d' = 8 \) \hspace{1cm} \( d' = 16 \)\n
\( d' = 32 \) \hspace{1cm} \( d' = 64 \) \hspace{1cm} \( d' = 128 \) \hspace{1cm} \( d' = 256 \) \hspace{1cm} Original Image
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.
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 are used
Uncorrelated and Independent

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

- **Non-Gaussian**
  - Independent $\implies$ Uncorrelated
  - Uncorrelated $\nimplies$ Independent

Uncorrelated: $\text{cov}(X_1, X_2) = 0$

Independent: $P(X_1, X_2) = P(X_1)P(X_2)$
PCA vs. ICA

1 class (orthogonal) 2 class (non-orthogonal) 2 class (non-orthogonal) 3 class cubic linear

PCA

ICA
Kernel PCA

- Kernel extension of PCA

Data (approximately) lies on a lower dimensional non-linear space.
Kernel PCA

- Hilbert space: $x \rightarrow \phi(x)$ (Nonlinear extension of PCA)

\[
C = \frac{1}{N} \sum_{i=1}^{N} \phi(x^{(i)})\phi(x^{(i)})^T
\]

- All eigenvectors of $C$ lie in the span of the mapped data points,

\[
Cv = \lambda v
\]

\[
v = \sum_{i=1}^{N} \phi(x^{(i)})
\]

- After some algebra, we have:

\[
K\alpha = N\lambda\alpha
\]
Linear Discriminant Analysis (LDA)

- Supervised feature extraction

- Fisher’s Linear Discriminant Analysis:
  - Dimensionality reduction
    - Finds linear combinations of features with large ratios of between-groups to within-groups scatters (as discriminant new variables)
  - Classification
    - Example: Predicts the class of an observation $x$ by the class whose mean vector is the closest to $x$ in the space of the discriminant variables
Good Projection for Classification

- What is a good criterion?
  - Separating different classes in the projected space
  - As opposed to PCA, we use also the labels of the training data
Good Projection for Classification

- What is a good criterion?
  - Separating different classes in the projected space
  - As opposed to PCA, we use also the labels of the training data
Good Projection for Classification

- What is a good criterion?
  - Separating different classes in the projected space
  - As opposed to PCA, we use also the labels of the training data
LDA Problem

- **Problem definition:**
  - \( C = 2 \) classes
  - \( \{(x^{(i)}, y^{(i)})\}_{i=1}^{N} \) training samples with \( N_1 \) samples from the first class \( (C_1) \) and \( N_2 \) samples from the second class \( (C_2) \)
  - Goal: finding the best direction \( w \) that we hope will enable accurate classification

- The projection of sample \( x \) onto a line in direction \( w \) is \( w^T x \)

- What is the measure of the separation between the projected points of different classes?
Measure of Separation in the Projected Direction

- Is the direction of the line jointing the class means is a good candidate for $\mathbf{w}$?

\[
\mu_1 = \frac{\sum_{x^{(i)} \in c_1} x^{(i)}}{N_1} \quad \mu_2 = \frac{\sum_{x^{(i)} \in c_2} x^{(i)}}{N_2}
\]
Measure of Separation in the Projected Direction

- The direction of the line jointing the class means is the solution of the following problem:
  - Maximizes the separation of the projected class means
    \[ \max_{\mathbf{w}} J(\mathbf{w}) = (\mu'_1 - \mu'_2)^2 \]
    \[ \text{s. t. } \|\mathbf{w}\| = 1 \]
  - What is the problem with the criteria considering only \( |\mu'_1 - \mu'_2| \)?
    - It does not consider the variances of the classes
LDA Criteria

- Fisher idea: maximize a function that will give
  - large separation between the projected class means
  - while also achieving a small variance within each class, thereby minimizing the class overlap.

\[
J(w) = \frac{|\mu_1' - \mu_2'|^2}{s_1'^2 + s_2'^2}
\]
LDA Criteria

- The scatters of the original data are:
  \[ s_1^2 = \sum_{x^{(i)} \in C_1} \| x^{(i)} - \mu_1 \|^2 \]
  \[ s_2^2 = \sum_{x^{(i)} \in C_2} \| x^{(i)} - \mu_2 \|^2 \]

- The scatters of projected data are:
  \[ s_1'^2 = \sum_{x^{(i)} \in C_1} \| w^T x^{(i)} - w^T \mu_1 \|^2 \]
  \[ s_2'^2 = \sum_{x^{(i)} \in C_2} \| w^T x^{(i)} - w^T \mu_2 \|^2 \]
LDA Criteria

\[ J(w) = \frac{|\mu'_1 - \mu'_2|^2}{s'_1^2 + s'_2^2} \]

\[ |\mu'_1 - \mu'_2|^2 = |w^T \mu_1 - w^T \mu_2|^2 \]

\[ = w^T (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w \]

\[ s'_1^2 = \sum_{x(i) \in C_1} \|w^T x(i) - w^T \mu_1\|^2 \]

\[ = w^T \left( \sum_{x(i) \in C_1} (x(i) - \mu_1)(x(i) - \mu_1)^T \right) w \]
LDA Criteria

\[ J(w) = \frac{w^T S_B w}{w^T S_W w} \]

**Between-class scatter matrix**

\[ S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T \]

**Within-class scatter matrix**

\[ S_W = S_1 + S_2 \]

\[ S_1 = \sum_{x^{(i)} \in C_1} (x^{(i)} - \mu_1)(x^{(i)} - \mu_1)^T \]

\[ S_2 = \sum_{x^{(i)} \in C_2} (x^{(i)} - \mu_2)(x^{(i)} - \mu_2)^T \]

scatter matrix = N x covariance matrix
LDA Derivation

\[ J(w) = \frac{w^T S_B w}{w^T S_w w} \]

\[
\frac{\partial J(w)}{\partial w} = \frac{\partial w^T S_B w}{\partial w} \times w^T S_w w - \frac{\partial w^T S_w w}{\partial w} \times w^T S_B w = \frac{(2S_B w) w^T S_w w - (2S_w w) w^T S_B w}{(w^T S_w w)^2}
\]

\[
\frac{\partial J(w)}{\partial w} = 0 \Rightarrow S_B w = \lambda S_w w
\]
LDA Derivation

$S_B w = \lambda S_W w \quad \Rightarrow \quad S_W^{-1} S_B w = \lambda w$

$S_B x$ for any vector $x$ points in the same direction as $\mu_1 - \mu_2$:

$S_B x = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T x = \alpha (\mu_1 - \mu_2)$

$w = S_W^{-1} (\mu_1 - \mu_2)$

Thus, we can solve the eigenvalue problem immediately
LDA Algorithm

- $\mu_1$ and $\mu_2 \leftarrow$ mean of samples of class 1 and 2 respectively
- $S_1$ and $S_2 \leftarrow$ scatter matrix of class 1 and 2 respectively
- $S_W = S_1 + S_2$
- $S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$

- Feature Extraction
  - $w = S_w^{-1}(\mu_1 - \mu_2)$ as the eigenvector corresponding to the largest eigenvalue of $S_w^{-1}S_b$

- Classification
  - $w = S_w^{-1}(\mu_1 - \mu_2)$
  - Using a threshold on $w^T x$, we can classify $x$
Multi-Class LDA (MDA)

- $C > 2$: the natural generalization of LDA involves $C - 1$ discriminant functions.
- The projection from a d-dimensional space to a $(C - 1)$-dimensional space (tacitly assumed that $d \geq C$).

\[
S_W = \sum_{j=1}^{C} S_j
\]

\[
S_B = \sum_{j=1}^{C} N_j (\mu_j - \mu)(\mu_j - \mu)^T
\]

\[
\mu_j = \frac{\sum_{x^{(i)} \in C_j} x^{(i)}}{N_j} \quad j = 1, ..., C
\]

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

\[
S_j = \sum_{x^{(i)} \in C_j} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T \quad j = 1, ..., C
\]
Multi-Class LDA

- $W = [w_1 \ w_2 \ \ldots \ w_{C-1}]$
- $x' = W^T x$

- Means and scatters after transform $x' = W^T x$:
  - $S'_B = W^T S_B W$
  - $S'_W = W^T S_W W$
Multi-Class LDA: Objective Function

- We seek a transformation matrix $W$ that in some sense “maximizes the ratio of the between-class scatter to the within-class scatter”.

- A simple scalar measure of scatter is the determinant of the scatter matrix.
Multi-Class LDA: Objective Function

\[ J(W) = \frac{|W^T S_B W|}{|W^T S_W W|} \rightarrow \text{determinant} \]

- The solution of the problem where \( W = [w_1 \ w_2 \ ... \ w_{C-1}] \):
  \[ S_B w_i = \lambda_i S_W w_i \]

- It is a generalized eigenvectors problem.
Multi-Class LDA: $d' \leq C - 1$

- $\text{rank}(S_B) \leq C - 1$
  - $S_B$ is the sum of $C$ matrices $(\mu_j - \mu)(\mu_j - \mu)^T$ of rank (at most) one and only $C - 1$ of these are independent,
  - $\Rightarrow$ atmost $C - 1$ nonzero eigenvalues and the desired weight vectors correspond to these nonzero eigenvalues.
Multi-Class LDA: Other Objective Functions

- There are many possible choices of criterion for multi-class LDA, e.g.:

\[
J(W) = tr(S'_w^{-1}S'_b) = tr((W^T S_w W)^{-1}(W^T S_b W))
\]

- The solution is given by solving a generalized eigenvalue problem \(S_w^{-1}S_b\)
  - Solution: eigen vectors corresponding to the largest eigen values constitute the new variables
LDA Criterion Limitation

- When $\mu_1 = \mu_2$, LDA criterion cannot lead to a proper projection ($J(w) = 0$)
  - However, discriminatory information in the scatter of the data may be helpful

- If classes are non-linearly separable they may have large overlap when projected to any line
- LDA implicitly assumes Gaussian distribution of samples of each class
Issues in LDA

- Singularity or undersampled problem (when $N < d$)
  - Example: gene expression data, images, text documents

- Can reduces dimension only to $d' \leq C - 1$ (unlike PCA)

- Approaches to avoid these problems:
  - PCA+LDA, Regularized LDA, Locally FDA (LFDA), etc.
Summary

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

- Advances in the recent decade:
  - Semi-supervised feature extraction
  - Nonlinear dimensionality reduction