Lecture 20: Dimension Reduction I
Feature Extraction
(Based on Dr. Gutierrez-Osuna’s lecture notes)
Why Dimension Reduction?

- Curse of Dimensionality

- Applications:
  - Pattern Recognition
  - Similarity Search
  - Visualization
  - Compression

- Different goals
  - Better clustering/classification results
  - Preserve distance relationship
  - Produce as few bits as possible with little loss of quality
Outline

- Curse of Dimensionality
- Feature selection vs. Feature extraction
- Signal Representation vs. Classification
- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Independent Component Analysis (ICA)
Curse of dimensionality (1)

- The **curse of dimensionality**
  - A term coined by Bellman in 1961
  - Refers to the problems associated with multivariate data analysis as the dimensionality increases
  - We will illustrate these problems with a simple example

- **Consider a 3-class pattern recognition problem**
  - A simple approach would be to
    - Divide the feature space into uniform bins
    - Compute the ratio of examples for each class at each bin and,
    - For a new example, find its bin and choose the predominant class in that bin
  - In our toy problem we decide to start with one single feature and divide the real line into 3 segments

  ![Feature Space Diagram](image)

  - After doing this, we notice that there exists too much overlap among the classes, so we decide to incorporate a second feature to try and improve separability
Curse of dimensionality (2)

- We decide to preserve the granularity of each axis, which raises the number of bins from 3 (in 1D) to $3^2=9$ (in 2D)
  - At this point we need to make a decision: do we maintain the density of examples per bin or do we keep the number of examples had for the one-dimensional case?
    - Choosing to maintain the density increases the number of examples from 9 (in 1D) to 27 (in 2D)
    - Choosing to maintain the number of examples results in a 2D scatter plot that is very sparse

- Moving to three features makes the problem worse:
  - The number of bins grows to $3^3=27$
  - For the same density of examples the number of needed examples becomes 81
  - For the same number of examples, well, the 3D scatter plot is almost empty
- Obviously, our approach to divide the sample space into equally spaced bins was quite inefficient
  - There are other approaches that are much less susceptible to the curse of dimensionality, but the problem still exists

- How do we beat the curse of dimensionality?
  - By incorporating prior knowledge
  - By providing increasing smoothness of the target function
  - By reducing the dimensionality

- In practice, the curse of dimensionality means that, for a given sample size, there is a maximum number of features above which the performance of our classifier will degrade rather than improve
  - In most cases, the additional information that is lost by discarding some features is (more than) compensated by a more accurate mapping in the lower-dimensional space
Dimensionality Reduction (1)

- Two approaches are available to perform dimensionality reduction
  - Feature extraction: creating a subset of new features by combinations of the existing features
  - Feature selection: choosing a subset of all the features (the ones more informative)

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_N \\
\end{bmatrix}
\rightarrow
\begin{bmatrix}
  x_{i1} \\
  x_{i2} \\
  \vdots \\
  x_{iM} \\
\end{bmatrix}
\quad \text{feature selection}
\]

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_N \\
\end{bmatrix}
\rightarrow
\begin{bmatrix}
  y_1 \\
  y_2 \\
  \vdots \\
  y_M \\
\end{bmatrix}
= f
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_N \\
\end{bmatrix}
\]

- The problem of feature extraction can be stated as
  - Given a feature space \( x_i \in \mathbb{R}^N \) find a mapping \( y = f(x) : \mathbb{R}^N \rightarrow \mathbb{R}^M \) with \( M < N \) such that the transformed feature vector \( y_i \in \mathbb{R}^M \) preserves (most of) the information or structure in \( \mathbb{R}^N \).
  - An optimal mapping \( y = f(x) \) will be one that results in no increase in the minimum probability of error.
    - This is, a Bayes decision rule applied to the initial space \( \mathbb{R}^N \) and to the reduced space \( \mathbb{R}^M \) yield the same classification rate.
Dimensionality Reduction (2)

- In general, the optimal mapping \( y = f(x) \) will be a non-linear function
  - However, there is no systematic way to generate non-linear transforms
    - The selection of a particular subset of transforms is problem dependent
  - For this reason, feature extraction is commonly limited to linear transforms: \( y = Wx \)
    - This is, \( y \) is a linear projection of \( x \)
    - NOTE: When the mapping is a non-linear function, the reduced space is called a manifold
The selection of the feature extraction mapping $y=f(x)$ is guided by an objective function that we seek to maximize (or minimize).

Depending on the criteria used by the objective function, feature extraction techniques are grouped into two categories:

- **Signal representation**: The goal of the feature extraction mapping is to represent the samples accurately in a lower-dimensional space.
- **Classification**: The goal of the feature extraction mapping is to enhance the class-discriminatory information in the lower-dimensional space.

Within the realm of linear feature extraction, two techniques are commonly used:

- Principal Components Analysis (PCA)
  - uses a signal representation criterion
- Linear Discriminant Analysis (LDA)
  - uses a signal classification criterion
PCA (1)

- The objective of PCA is to perform dimensionality reduction while preserving as much of the randomness (variance) in the high-dimensional space as possible.
  - Let $x$ be an $N$-dimensional random vector, represented as a linear combination of orthonormal basis vectors $[\varphi_1 \mid \varphi_2 \mid \ldots \mid \varphi_N]$ as
    \[
    x = \sum_{i=1}^{N} y_i \varphi_i \text{ where } \varphi_i \mid \varphi_j = \begin{cases} 
    0 & i = j \\
    1 & i = j
    \end{cases}
    \]
  - Suppose we choose to represent $x$ with only $M$ $(M<N)$ of the basis vectors. We can do this by replacing the components $[y_{M+1}, \ldots, y_N]^\top$ with some pre-selected constants $b_i$,
    \[
    \hat{x}(M) = \sum_{i=1}^{M} y_i \varphi_i + \sum_{i=M+1}^{N} b_i \varphi_i
    \]
  - The representation error is then
    \[
    \Delta x(M) = x - \hat{x}(M) = \sum_{i=1}^{N} y_i \varphi_i - \left( \sum_{i=1}^{M} y_i \varphi_i - \sum_{i=M+1}^{N} b_i \varphi_i \right) = \sum_{i=M+1}^{N} (y_i - b_i) \varphi_i
    \]
  - We can measure this representation error by the mean-squared magnitude of $\Delta x$
  - Our goal is to find the basis vectors $\varphi_i$ and constants $b_i$ that minimize this mean-square error
    \[
    \varepsilon^2(M) = E[\Delta x(M)]^2 = E \left[ \sum_{i=M+1}^{N} \sum_{j=M+1}^{N} (y_i - b_i)(y_j - b_j) \varphi_i \varphi_j \right] = \sum_{i=M+1}^{N} E[(y_i - b_i)^2]
    \]
PCA (2)

As we have done earlier in the course, the optimal values of \( b_i \) can be found by computing the partial derivative of the objective function and equating it to zero

\[
\frac{\partial}{\partial b_i} E[(y_i - b_i)^2] = -2(E[y_i] - b_i) = 0 \Rightarrow b_i = E[y_i]
\]

Therefore, we will replace the discarded dimensions \( y_i \)'s by their expected value (an intuitive solution).

The mean-square error can then be written as

\[
\bar{e}^2(M) = \sum_{i=M+1}^{N} E[(y_i - E[y_i])^2] = \sum_{i=M+1}^{N} E[(x\phi_i - E[x\phi_i])^T (x\phi_i - E[x\phi_i])]
\]

\[
= \sum_{i=M+1}^{N} \phi_i^T E[(x - E[x])(x - E[x])^T] \phi_i = \sum_{i=M+1}^{N} \phi_i^T \Sigma_x \phi_i
\]

where \( \Sigma_x \) is the covariance matrix of \( x \).

We seek to find the solution that minimizes this expression subject to the orthonormality constraint, which we incorporate into the expression using a set of Lagrange multipliers \( \lambda_i \)

\[
\bar{e}^2(M) = \sum_{i=M+1}^{N} \phi_i^T \Sigma_x \phi_i + \sum_{i=M+1}^{N} \lambda_i (1 - \phi_i^T \phi_i)
\]

Computing the partial derivative with respect to the basis vectors

\[
\frac{\partial}{\partial \phi_i} \bar{e}^2(M) = \frac{\partial}{\partial \phi_i} \left[ \sum_{i=M+1}^{N} \phi_i^T \Sigma_x \phi_i + \sum_{i=M+1}^{N} \lambda_i (1 - \phi_i^T \phi_i) \right] = 2(\Sigma_x \phi_i - \lambda_i \phi_i) = 0 \Rightarrow \Sigma_x \phi_i = \lambda_i \phi_i
\]

NOTE: \( \frac{d}{dx} (x^T Ax) = (A + A^T)x = 2Ax \)

So \( \phi_i \) and \( \lambda_i \) are the eigenvectors and eigenvalues of the covariance matrix \( \Sigma_x \).
PCA (3)

- We can then express the sum-square error as

\[ \bar{e}^2(M) = \sum_{i=M-1}^{N} \phi_i^T \Sigma \phi_i = \sum_{i=M-1}^{N} \phi_i^T \lambda_i \phi_i = \sum_{i=M-1}^{N} \lambda_i \]

- In order to minimize this measure, \( \lambda_i \) will have to be smallest eigenvalues
  - Therefore, to represent \( x \) with minimum sum-square error, we will choose the eigenvectors \( \phi_i \) corresponding to the largest eigenvalues \( \lambda_i \).

**PCA dimensionality reduction**

The optimal* approximation of a random vector \( x \in \mathbb{R}^N \) by a linear combination of \( M \) (\( M < N \)) independent vectors is obtained by projecting the random vector \( x \) onto the eigenvectors \( \phi_i \) corresponding to the largest eigenvalues \( \lambda_i \) of the covariance matrix \( \Sigma_x \)

*optimality is defined as the minimum of the sum-square magnitude of the approximation error
PCA (4)

- **NOTES**
  - Since PCA uses the eigenvectors of the covariance matrix $\Sigma_x$, it is able to find the independent axes of the data under the unimodal Gaussian assumption
    - For non-Gaussian or multi-modal Gaussian data, PCA simply de-correlates the axes
  - The main limitation of PCA is that it does not consider class separability since it does not take into account the class label of the feature vector
    - PCA simply performs a coordinate rotation that aligns the transformed axes with the directions of maximum variance
    - **There is no guarantee that the directions of maximum variance will contain good features for discrimination**

- **Historical remarks**
  - Principal Components Analysis is the oldest technique in multivariate analysis
  - PCA is also known as the Karhunen-Loève transform (communication theory)
  - PCA was first introduced by Pearson in 1901, and it experienced several modifications until it was generalized by Loève in 1963
PCA Example

- Compute the principal components for the following two-dimensional dataset
  - $X = (x_1, x_2) = \{(1,2), (3,3), (3,5), (5,4), (5,6), (6,5), (8,7), (9,8)\}$
  - Let's first plot the data to get an idea of which solution we should expect

- SOLUTION (by hand)
  - The (biased) covariance estimate of the data is:
    $$\Sigma_x = \begin{bmatrix} 6.25 & 4.25 \\ 4.25 & 3.5 \end{bmatrix}$$
  - The eigenvalues are the zeros of the characteristic equation
    $$\Sigma_x \mathbf{v} = \lambda \mathbf{v} \Rightarrow \left| \Sigma_x - \lambda I \right| = 0 \Rightarrow \begin{vmatrix} 6.25 - \lambda & 4.25 \\ 4.25 & 3.5 - \lambda \end{vmatrix} = 0 \Rightarrow \lambda_1 = 9.34; \ \lambda_2 = 0.41;$$
  - The eigenvectors are the solutions of the system
    $$\begin{bmatrix} 6.25 & 4.25 \\ 4.25 & 3.5 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} = \lambda_1 \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} \Rightarrow \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} = \begin{bmatrix} 0.81 \\ 0.59 \end{bmatrix}$$
    $$\begin{bmatrix} 6.25 & 4.25 \\ 4.25 & 3.5 \end{bmatrix} \begin{bmatrix} v_{21} \\ v_{22} \end{bmatrix} = \lambda_2 \begin{bmatrix} v_{21} \\ v_{22} \end{bmatrix} \Rightarrow \begin{bmatrix} v_{21} \\ v_{22} \end{bmatrix} = \begin{bmatrix} -0.59 \\ 0.81 \end{bmatrix}$$
  - HINT: To solve each system manually, first assume that one of the variables is equal to one (i.e. $v_{i1} = 1$), then find the other one and finally normalize the vector to make it unit-length
The objective of LDA is to perform dimensionality reduction while preserving as much of the class discriminatory information as possible.

- Assume we have a set of D-dimensional samples \( \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\} \), \( N_1 \) of which belong to class \( \omega_1 \), and \( N_2 \) to class \( \omega_2 \). We seek to obtain a scalar \( y \) by projecting the samples \( x \) onto a line:

\[
y = w^T x
\]

- Of all the possible lines we would like to select the one that maximizes the separability of the scalars.
  - This is illustrated for the two-dimensional case in the following figures.
In order to find a good projection vector, we need to define a measure of separation between the projections

- The mean vector of each class in $x$ and $y$ feature space is

$$\mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x \quad \text{and} \quad \tilde{\mu}_i = \frac{1}{N_i} \sum_{y \in \omega_i} y = \frac{1}{N_i} \sum_{x \in \omega_i} w^T x = w^T \mu_i$$

- We could then choose the distance between the projected means as our objective function

$$J(w) = \| \tilde{\mu}_1 - \tilde{\mu}_2 \| = \| w^T (\mu_1 - \mu_2) \|$$

- However, the distance between the projected means is not a very good measure since it does not take into account the standard deviation within the classes

---

This axis yields better class separability

This axis has a larger distance between means
The solution proposed by Fisher is to maximize a function that represents the difference between the means, normalized by a measure of the within-class scatter.

For each class we define the scatter, an equivalent of the variance, as

\[ \tilde{S}_i^2 = \sum_{y \in c_i} (y - \tilde{\mu}_i)^2 \]

where the quantity \( (\tilde{S}_1^2 + \tilde{S}_2^2) \) is called the within-class scatter of the projected examples.

The Fisher linear discriminant is defined as the linear function \( w^T x \) that maximizes the criterion function

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

Therefore, we will be looking for a projection where examples from the same class are projected very close to each other and, at the same time, the projected means are as farther apart as possible.
LDA – 2 classes (3)

- In order to find the optimum projection \( w^\ast \), we need to express \( J(w) \) as an explicit function of \( w \).
- We define a measure of the scatter in multivariate feature space \( x \), which are scatter matrices:
  \[
  S_i = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T
  \]
  \[
  S_1 + S_2 = S_W
  \]
  where \( S_W \) is called the within-class scatter matrix.
- The scatter of the projection \( y \) can then be expressed as a function of the scatter matrix in feature space \( x \):
  \[
  \tilde{S}^2 = \sum_{y \in \omega_i} (y - \tilde{\mu})^2 = \sum_{x \in \omega_i} (w^T x - w^T \mu_i)^2 = \sum_{x \in \omega_i} w^T (x - \mu_i)(x - \mu_i)^T w = w^T S_W w
  \]
  \[
  \tilde{S}^2 = w^T S_W w
  \]
- Similarly, the difference between the projected means can be expressed in terms of the means in the original feature space
  \[
  (\tilde{\mu}_1 - \tilde{\mu}_2)^2 = (w^T \mu_1 - w^T \mu_2)^2 = w^T (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w = w^T S_B w
  \]
  where \( S_B \) is called the between-class scatter. Note that, since \( S_B \) is the outer product of two vectors, its rank is at most one.
- We can finally express the Fisher criterion in terms of \( S_W \) and \( S_B \) as:
  \[
  J(w) = \frac{w^T S_B w}{w^T S_W w}
  \]
LDA – 2 classes (4)

- To find the maximum of $J(w)$ we derive and equate to zero:

$$\frac{d}{dw} J(w) = \frac{d}{dw} \left[ \frac{w^TS_Bw}{w^TS_ww} \right] = 0 \Rightarrow$$

$$\Rightarrow \left[ w^TS_ww \right] \frac{d}{dw} \left[ w^TS_Bw \right] - \left[ w^TS_Bw \right] \frac{d}{dw} \left[ w^TS_ww \right] = 0 \Rightarrow$$

$$\Rightarrow \left[ w^TS_ww \right] 2S_Bw - \left[ w^TS_Bw \right] 2S_ww = 0$$

- Dividing by $w^TS_ww$

$$\frac{w^TS_ww}{w^TS_ww} S_Bw - \frac{w^TS_Bw}{w^TS_ww} S_ww = 0 \Rightarrow$$

$$\Rightarrow S_Bw - JS_ww = 0 \Rightarrow$$

$$\Rightarrow S_w^{-1}S_Bw - Jw = 0$$

- Solving the generalized eigenvalue problem ($S_w^{-1}S_bw = Jw$) yields

$$w^* = \arg\max_w \left\{ \frac{w^TS_Bw}{w^TS_ww} \right\} = S_w^{-1}(\mu_1 - \mu_2)$$

- This is known as Fisher’s Linear Discriminant (1936), although it is not a discriminant but rather a specific choice of direction for the projection of the data down to one dimension.
LDA Example

- Compute the Linear Discriminant projection for the following two-dimensional dataset
  - \( X_1 = (x_1, x_2) = \{(4, 1), (2, 4), (2, 3), (3, 6), (4, 4)\} \)
  - \( X_2 = (x_1, x_2) = \{(9, 10), (6, 8), (9, 5), (8, 7), (10, 8)\} \)

- **SOLUTION (by hand)**
  - The class statistics are:
    - \( S_1 = \begin{bmatrix} 0.80 & -0.40 \\ -0.40 & 2.60 \end{bmatrix} \)
    - \( S_2 = \begin{bmatrix} 1.84 & -0.04 \\ -0.04 & 2.64 \end{bmatrix} \)
    - \( \mu_1 = [3.00, 3.60] \)
    - \( \mu_2 = [8.40, 7.60] \)
  - The within- and between-class scatter are:
    - \( S_B = \begin{bmatrix} 29.16 & 21.60 \\ 21.60 & 16.00 \end{bmatrix} \)
    - \( S_W = \begin{bmatrix} 2.64 & -0.44 \\ -0.44 & 5.28 \end{bmatrix} \)
  - The LDA projection is then obtained as the solution of the generalized eigenvalue problem
    \[
    S_W^{-1} S_B v = \lambda v \Rightarrow S_W^{-1} S_B - \lambda I = 0 \Rightarrow \begin{vmatrix} 11.89 - \lambda & 8.81 \\ 5.08 & 3.76 - \lambda \end{vmatrix} = 0 \Rightarrow \lambda = 15.65
    \]
    \[
    \begin{bmatrix} 11.89 & 8.81 \\ 5.08 & 3.76 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 15.65 \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \Rightarrow \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0.91 \\ 0.39 \end{bmatrix}
    \]
  - Or directly by
    \[ w^* = S_W^{-1} (\mu_1 - \mu_2) = [-0.91, -0.39]^T \]
Fisher’s LDA generalizes very gracefully for C-class problems

- Instead of one projection \( y \), we will now seek \((C-1)\) projections \([y_1, y_2, \ldots, y_{C-1}]\) by means of (C-1) projection vectors \( w_i \), which can be arranged by columns into a projection matrix \( W = [w_1, w_2, \ldots, w_{C-1}] \):

\[
y_i = w_i^T x \quad \Rightarrow \quad y = W^T x
\]

**Derivation**

- The generalization of the within-class scatter is

\[
S_w = \sum_{i=1}^{C} S_i
\]

where \( S_i = \sum_{x \in C_i} (x - \mu_i)(x - \mu_i)^T \) and \( \mu_i = \frac{1}{N_i} \sum_{x \in C_i} x \)

- The generalization for the between-class scatter is

\[
S_B = \sum_{i=1}^{C} N_i (\mu_i - \mu)(\mu_i - \mu)^T
\]

where \( \mu = \frac{1}{N} \sum_{x} x = \frac{1}{N} \sum_{x \in C_i} N_i \mu_i \)

- where \( S_T = S_B + S_W \) is called the total scatter matrix
LDA – C classes (2)

- Similarly, we define the mean vector and scatter matrices for the projected samples as
  \[
  \tilde{\mu}_i = \frac{1}{N_i} \sum_{y \in w_i} y, \quad \tilde{S}_w = \sum_{i=1}^{C} \sum_{y \in w_i} (y - \tilde{\mu}_i)(y - \tilde{\mu}_i)^T
  \]
  \[
  \tilde{\mu} = \frac{1}{N} \sum_{y} y, \quad \tilde{S}_b = \sum_{i=1}^{C} N_i (\tilde{\mu}_i - \tilde{\mu})(\tilde{\mu}_i - \tilde{\mu})^T
  \]

- From our derivation for the two-class problem, we can write
  \[
  \tilde{S}_w = W^T S_w W, \quad \tilde{S}_b = W^T S_b W
  \]

- Recall that we are looking for a projection that maximizes the ratio of between-class to within-class scatter. Since the projection is no longer a scalar (it has C-1 dimensions), we then use the determinant of the scatter matrices to obtain a scalar objective function:
  \[
  J(W) = \frac{|\tilde{S}_b|}{|\tilde{S}_w|} = \frac{|W^T S_b W|}{|W^T S_w W|}
  \]

- And we will seek the projection matrix \( W^* \) that maximizes this ratio
LDA – C classes (3)

- It can be shown that the optimal projection matrix $W^*$ is the one whose columns are the eigenvectors corresponding to the largest eigenvalues of the following generalized eigenvalue problem:

$$W^* = [w_1' \mid w_2' \mid \cdots \mid w_{c-1}'] = \text{argmax} \left( \frac{W^T S_B W}{W^T S_W W} \right) \Rightarrow (S_B - \lambda_i S_W)w_i' = 0$$

- **NOTES**
  - $S_B$ is the sum of $C$ matrices of rank one or less and the mean vectors are constrained by
    $$\frac{1}{C} \sum_{i=1}^{C} \mu_i = \mu$$
    - Therefore, $S_B$ will be of rank $(C-1)$ or less
    - This means that only $(C-1)$ of the eigenvalues $\lambda_i$ will be non-zero
  - The projections with maximum class separability information are the eigenvectors corresponding to the largest eigenvalues of $S_W^{-1}S_B$
  - LDA can be derived as the Maximum Likelihood method for the case of normal class-conditional densities with equal covariance matrices
LDA vs. PCA

- These figures show the performance of PCA and LDA on an odor recognition problem
  - Five types of coffee beans were presented to an array of chemical gas sensors
  - For each coffee type, 45 "sniffs" were performed and the response of the gas sensor array was processed in order to obtain a 60-dimensional feature vector

- Results
  - From the 3D scatter plots it is clear that LDA outperforms PCA in terms of class discrimination
  - This is one example where the discriminatory information is not aligned with the direction of maximum variance
Limitations of LDA

- LDA produces at most C-1 feature projections
  - If the classification error estimates establish that more features are needed, some other method must be employed to provide those additional features

- LDA is a parametric method since it assumes unimodal Gaussian likelihoods
  - If the distributions are significantly non-Gaussian, the LDA projections will not be able to preserve any complex structure of the data, which may be needed for classification

- LDA will fail when the discriminatory information is not in the mean but rather in the variance of the data
Independent Component Analysis (ICA)

- PCA: find subspace with largest variance
- ICA: find subspace that independent from the rest
- PCA = ICA for Gaussian Data but different significantly when not
Uncorrelated vs. Independent

- Are $x$ and $y$ uncorrelated?
- Are $x$ and $y$ independent?
Simple “Cocktail Party” Problem

Unknown mixing matrix $A$

$s_1$ → $x_1$

$s_2$ → $x_2$

Observations

$n$ sources, $m=n$ observations

$x = As$
Motivation

Two Independent Sources Mixture at two Mics

\[ x_1(t) = a_{11}s_1 + a_{12}s_2 \]
\[ x_2(t) = a_{21}s_1 + a_{22}s_2 \]

\( a_{ij} \ldots \) Depend on the distances of the microphones from the speakers
Motivation

Get the Independent Signals out of the Mixture
Definition and Applications

Problem Statement

Given inputs $x_1, x_2, \ldots, x_n$, find transformation $W$

\[
\begin{pmatrix}
y_1 \\
\vdots \\
y_m
\end{pmatrix} = W
\begin{pmatrix}
x_1 \\
x_2 \\
x_n
\end{pmatrix}
\]

such that $y_1, \ldots, y_m$ are independent of each other

Applications:

- “cocktail party problem” (Blind source separation)
  - Neurological signal separation from electroencephalograms (EEG)
  - Separation of noise from signals in mobile
  - Climate studies – separating El Nino from Volcano

- Multimedia: compression, watermarking
ICA Estimation Principle

- **Principle 1: Nonlinear decorrelation**
  - Find the matrix $W$ so that for any $i \neq j$, the components $y_i$ and $y_j$ are uncorrelated, AND the transform components $g(y_i)$ and $g(y_j)$ are uncorrelated, where $g$ and $h$ are some suitable non-linear function.

- **Principle 2: Maximum non-gaussianity**
  - Find the local maxima of non-gaussianity of a linear combination $y = \sum b_i x_i$ under the constraint that the variance of $y$ is constant. Each local maximum gives one independent component.
By Central Limit Theorem, \( y = \sum_i p_i x_i \) would normally be “more” Gaussian than individual components.

This would not be the case. \( y \) is one of the recovered independent sources \( s \) – it will be “less” Gaussian than any of the \( x \)’s.
Measures of Non-Gaussianity

- We need to have a quantitative measure of non-gaussianity for ICA Estimation.
- Kurtosis: gauss=0 (sensitive to outliers)
  \[ kurt(y) = E\{y^4\} - 3(E\{y^2\})^2 \]
- Entropy: gauss=largest
  \[ H(y) = -\int f(y) \log f(y) dy \]
- Neg-entropy: gauss = 0 (difficult to estimate)
  \[ J(y) = H(y_{gauss}) - H(y) \]
- Approximations
  \[ J(y) = \frac{1}{12} E\{y^2\}^2 + \frac{1}{48} kurt(y)^2 \]
  \[ J(y) \approx \left[ E\{G(y)\} - E\{G(v)\} \right]^2 \]
- where \( v \) is a standard gaussian random variable and:
  \[ G(y) = \frac{1}{a} \log \cosh(a.y) \]
  \[ G(y) = -\exp(-a.u^2 / 2) \]