Amirkabir University of Technology
(Tehran Polytechnic)

Machine Learning

Lecture 9.
Dimensionality Reduction

Alireza Rezvanian

Fall 2022

Last update: Dec. 04, 2022

Amirkabir University of Technology (Tehran Polytechnic)

Outline

- Data Reduction & Dimensionality Reduction
- Feature Selection
  - Filter Methods
    - Sequential forward selection (SFS) (heuristic search)
    - Sequential backward selection (SBS) (heuristic search)
  - Wrapper Methods
- Feature Extraction
  - Linear Dimensionality Reduction
    - Principal Component Analysis (PCA)
    - Singular Value Decomposition (SVD)
    - Canonical Correlation Analysis (CCA)
    - Fisher's Linear Discriminant Analysis (LDA)

Data Reduction

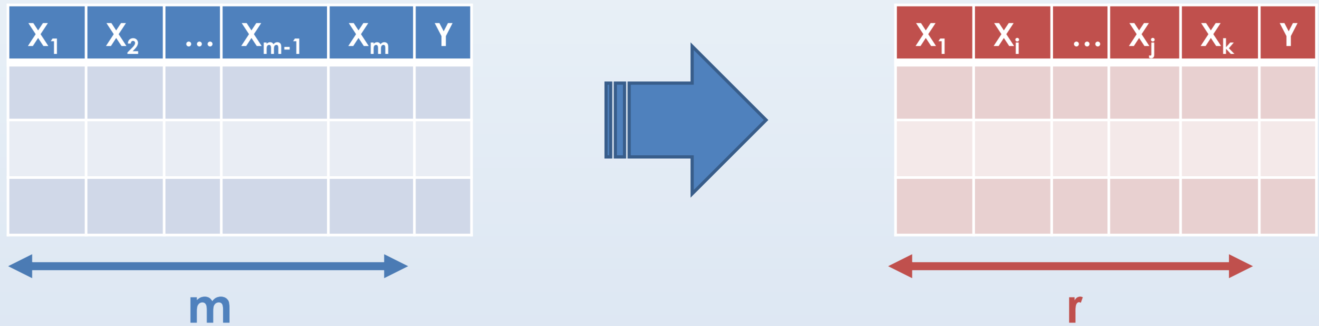
- Data reduction goal
  - Obtain a reduced representation of the data set that is much smaller in volume but yet produce the same (or almost the same) analytical results
- Data reduction methods
  - Regression
    - Data are modeled to fit a determined model (e.g. line or AR)
  - Sufficient Statistics
    - A function of the data that maintains all the statistical information of the original population
  - Histograms
    - Divide data into buckets and store average (sum) for each bucket
    - Partitioning rules: equal-width, equal-frequency, equal-variance, etc.
  - Clustering
    - Partition data set into clusters based on similarity, and store cluster representation only
    - Clustering methods will be discussed later.
  - Sampling
    - obtaining small samples to represent the whole data set D

Why Reduce Dimensionality?

- A limited yet salient feature set simplifies both pattern representation and classifier design
- Reduces time complexity: Less computation
- Reduces space complexity: Fewer parameters
- Saves the cost of observing the feature
- Simpler models are more robust on small datasets
- More interpretable; simpler explanation
- Data visualization (structure, groups, outliers, etc.) if plotted in 2 or 3 dimensions

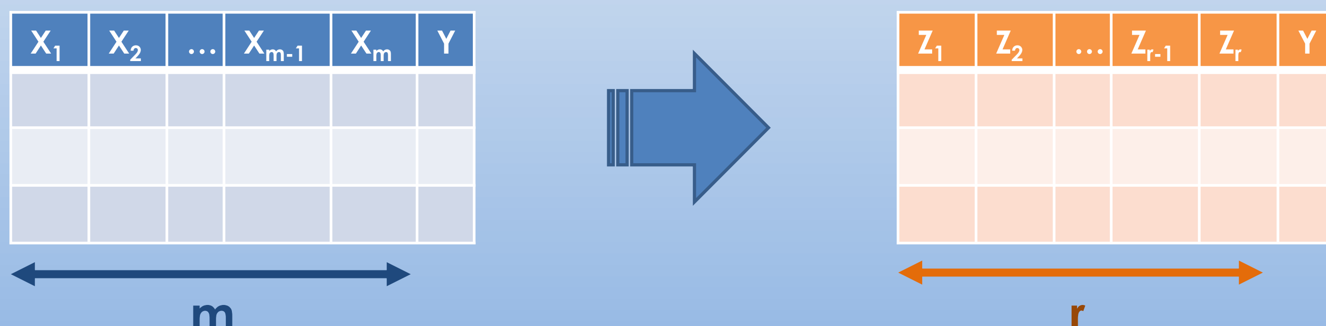
Dimensionality Reduction: Feature Selection vs. Feature Extraction

- Feature **selection**
  - Choosing a subset ($r < m$ important features) of a given feature set, ignoring the remaining $m - r$



$m >> r$

- Feature **extraction**
  - A linear or non-linear transform on the original feature space to new r<m dimentions



$m >> r$

Feature Extraction

- Unsupervised feature extraction

$$X = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \cdots & x_d^{(N)} \end{bmatrix} \Rightarrow \boxed{\text{Feature Extraction}} \Rightarrow$$

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

$$X' = \begin{bmatrix} x'^{(1)}_1 & \cdots & x'^{(1)}_{d'} \\ \vdots & \ddots & \vdots \\ x'^{(N)}_1 & \cdots & x'^{(N)}_{d'} \end{bmatrix}$$

- Supervised feature extraction

$$X = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \cdots & x_d^{(N)} \end{bmatrix} \Rightarrow \boxed{\text{Feature Extraction}} \Rightarrow$$

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

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

$$X' = \begin{bmatrix} x'^{(1)}_1 & \cdots & x'^{(1)}_{d'} \\ \vdots & \ddots & \vdots \\ x'^{(N)}_1 & \cdots & x'^{(N)}_{d'} \end{bmatrix}$$

Unsupervised Feature Reduction

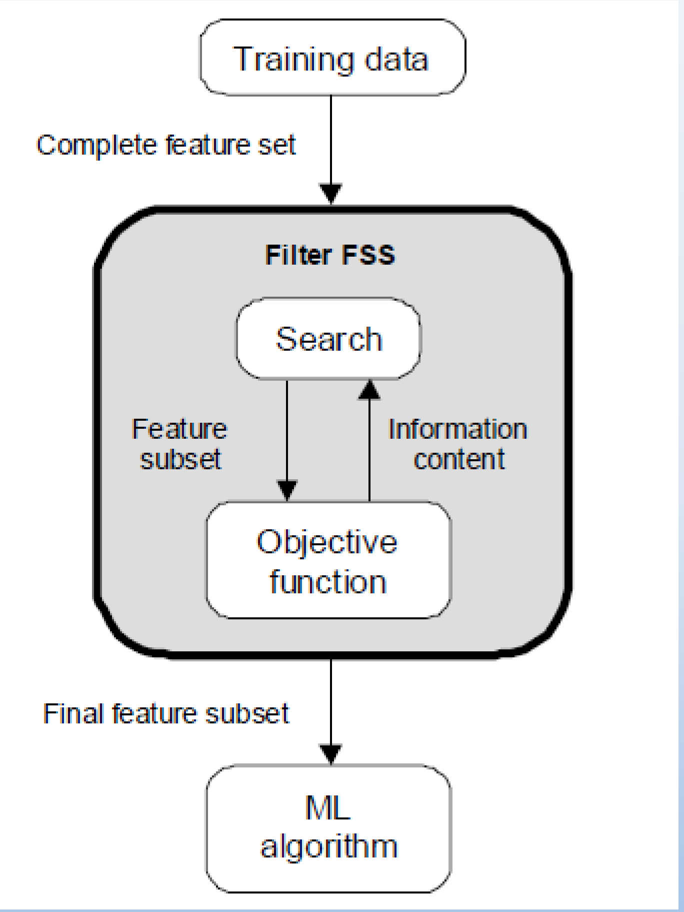
- Visualization and interpretation: projection of high-dimensional data onto 2D or 3D.
- Data compression: efficient storage, communication, or and retrieval.
- Pre-process: to improve accuracy by reducing features
  - As a preprocessing step to reduce dimensions for supervised learning tasks
  - Helps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

Feature Selection Methods

- One view
  - Univariate method
    - Considers one variable (feature) at a time
  - Multivariate method
    - Considers subsets of variables (features) together.
- Another view
  - Filter method
    - Ranks features subsets independently of the classifier.
  - Wrapper method
    - Uses a classifier to assess features subsets.
  - Embedded
    - Feature selection is part of the training procedure of a classifier (e.g. decision trees)

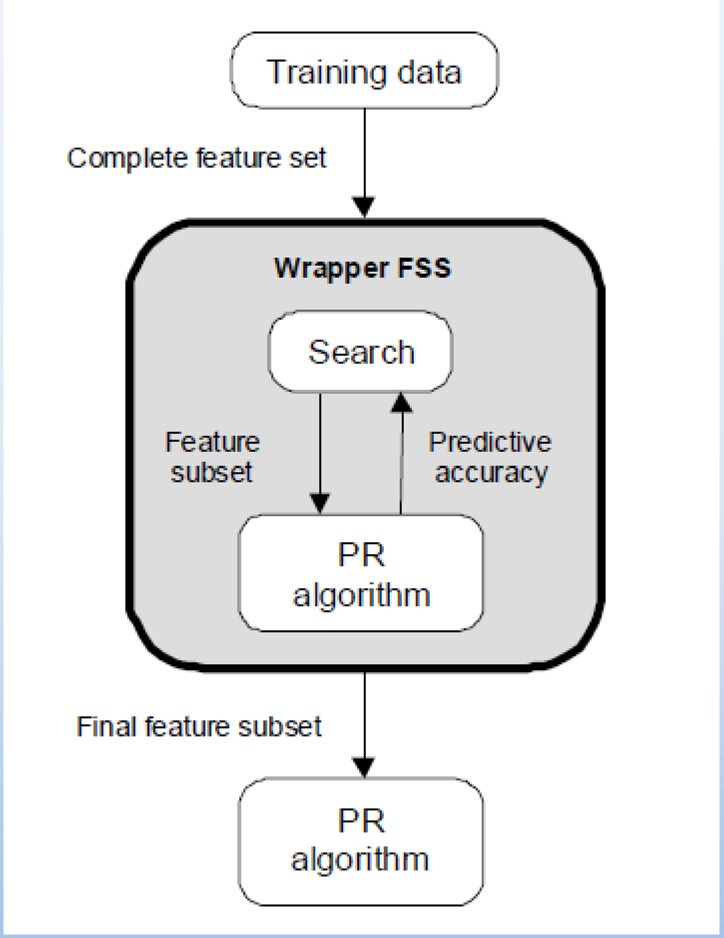
Filter Methods

- The objective function evaluates feature subsets by their information content, typically interclass distance, statistical dependence or information-theoretic measures.
- Evaluation is independent of the classification algorithm.



Wrapper Methods

- The objective function is a pattern classifier, which evaluates feature subsets by their predictive accuracy (recognition rate on test data) by statistical resampling or cross-validation.
- Evaluation uses criteria related to the classification algorithm.



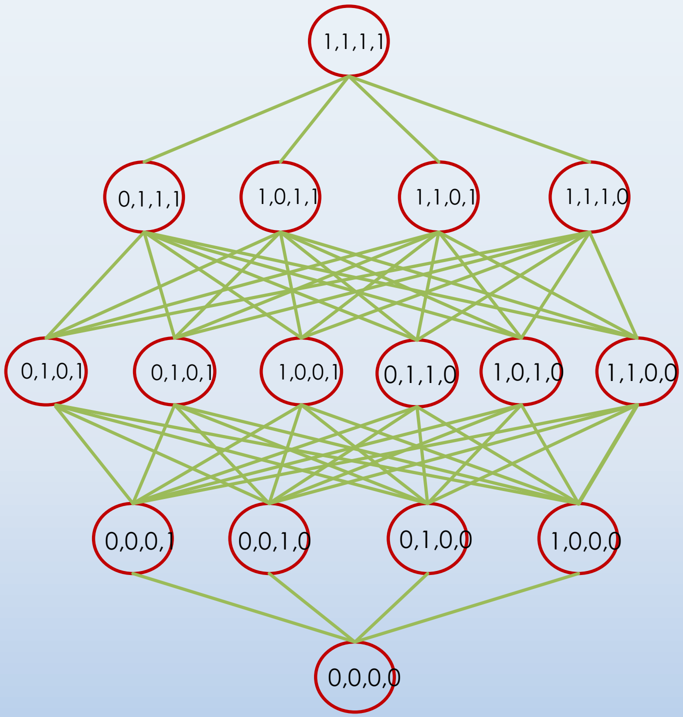
Filter vs Wrapper Approaches

- Filter Approach
  - Advantages
    - **Fast execution**: Filters generally involve a non-iterative computation on the dataset, which can execute much faster than a classifier training session
    - **Generality**: Since filters evaluate the intrinsic properties of the data, rather than their interactions with a particular classifier, their results exhibit more generality; the solution will be "good" for a large family of classifiers
  - Disadvantages
    - Tendency to select large subsets: Filter objective functions are generally monotonic
- Wrapper Approach
  - Advantages
    - **Accuracy**: wrappers generally have better recognition rates than filters since they tuned to the specific interactions between the classifier and the features.
    - **Ability to generalize**: wrappers have a mechanism to avoid over fitting, since they typically use cross-validation measures of predictive accuracy.
  - Disadvantages
    - Slow execution

Search Strategies

- Assuming N features, an exhaustive search would require:
- Examining all $\binom{N}{M}$ possible subsets of size M.
- Selecting the subset that performs the best according to the criterion function.
- The number of subsets grows combinatorically, making exhaustive search impractical.
- Iterative procedures are often used based on heuristics but they cannot guarantee the selection of the optimal subset.

Four Features – x_1, x_2, x_3, x_4

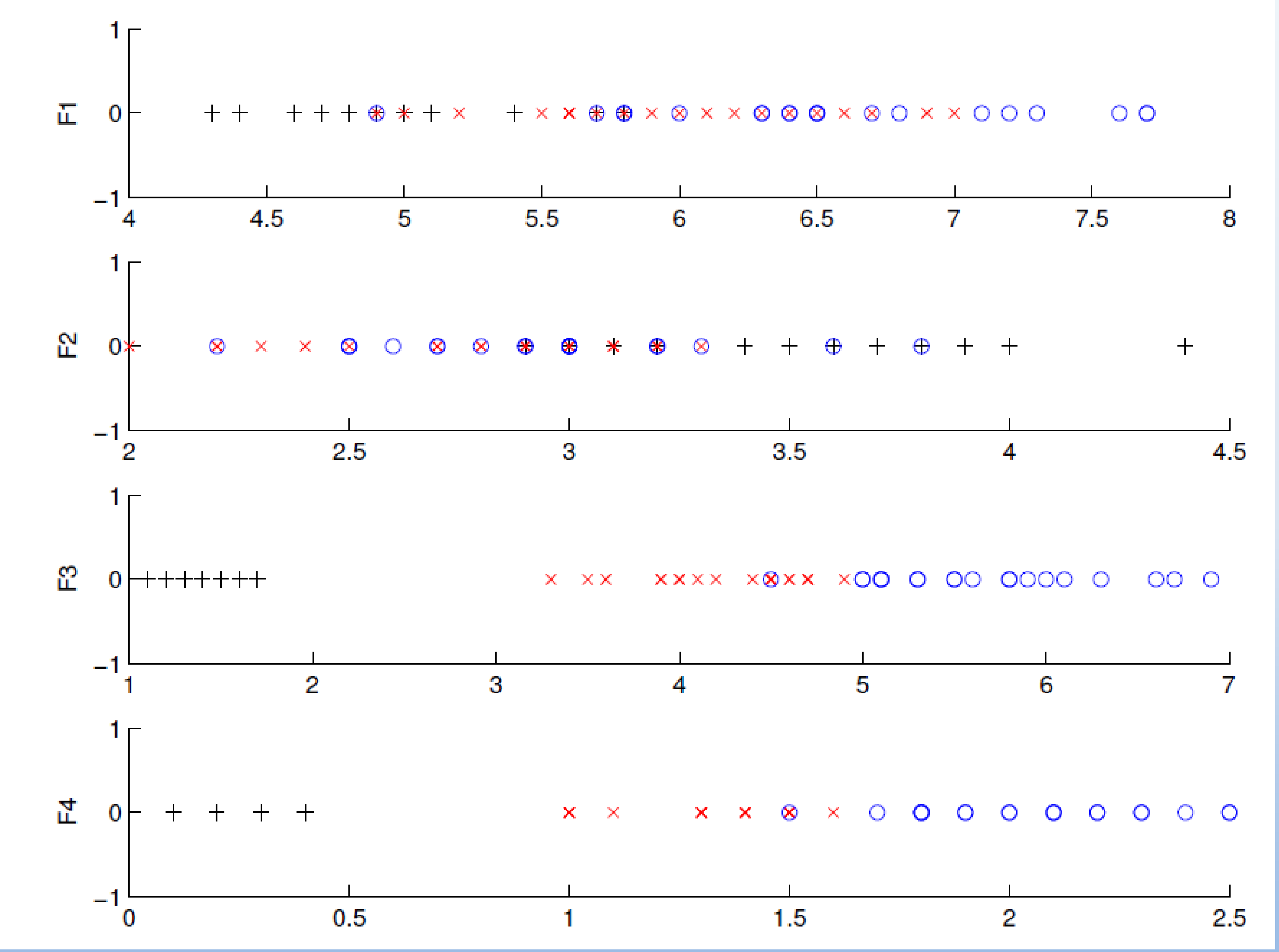


1-x_i is selected; 0-x_i is not selected

Subset Selection

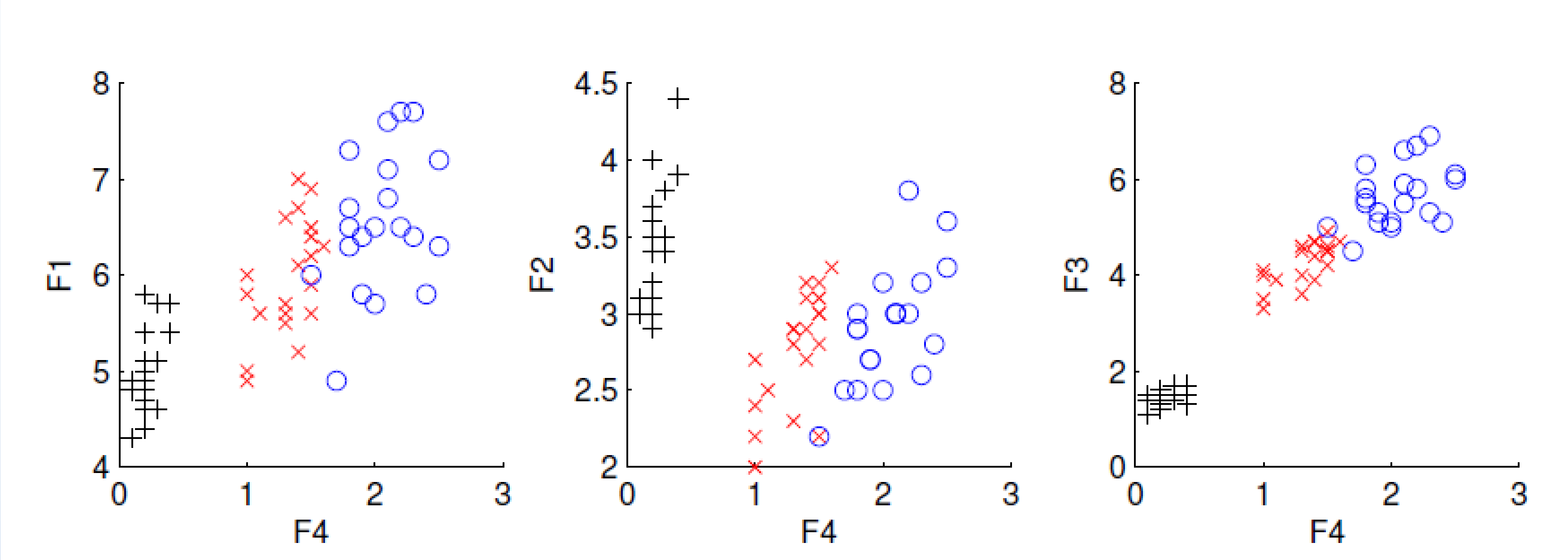
- There are 2^d subsets of d features
- Forward search: Add the best feature at each step
  - Set of features F initially Ø.
  - At each iteration, find the best new feature
    $j = \text{argmin}_i\, E\,(F \cup x_i)$
  - Add x_j to F if $E\,(F \cup x_j) < E\,(F)$
- Hill-climbing $O(d^2)$ algorithm
- Backward search: Start with all features and remove one at a time, if possible.
- Floating search (Add k, remove l)

Iris data: Single feature



Chosen

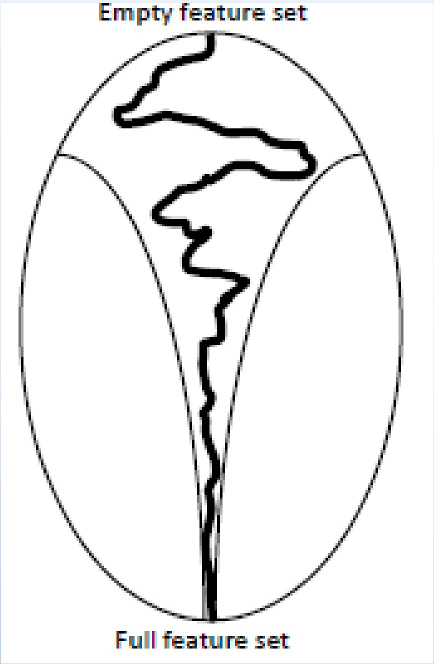
Iris data: Add one more feature to F4



Chosen

Sequential forward selection (SFS) (heuristic search)

- First, the best single feature is selected (i.e., using some criterion function).
- Then, pairs of features are formed using one of the remaining features and this best feature, and the best pair is selected.
- Next, triplets of features are formed using one of the remaining features and these two best features, and the best triplet is selected.
- This procedure continues until a predefined number of features are selected.



SFS performs best when the optimal subset is small.

Sequential forward selection (SFS) (heuristic search)

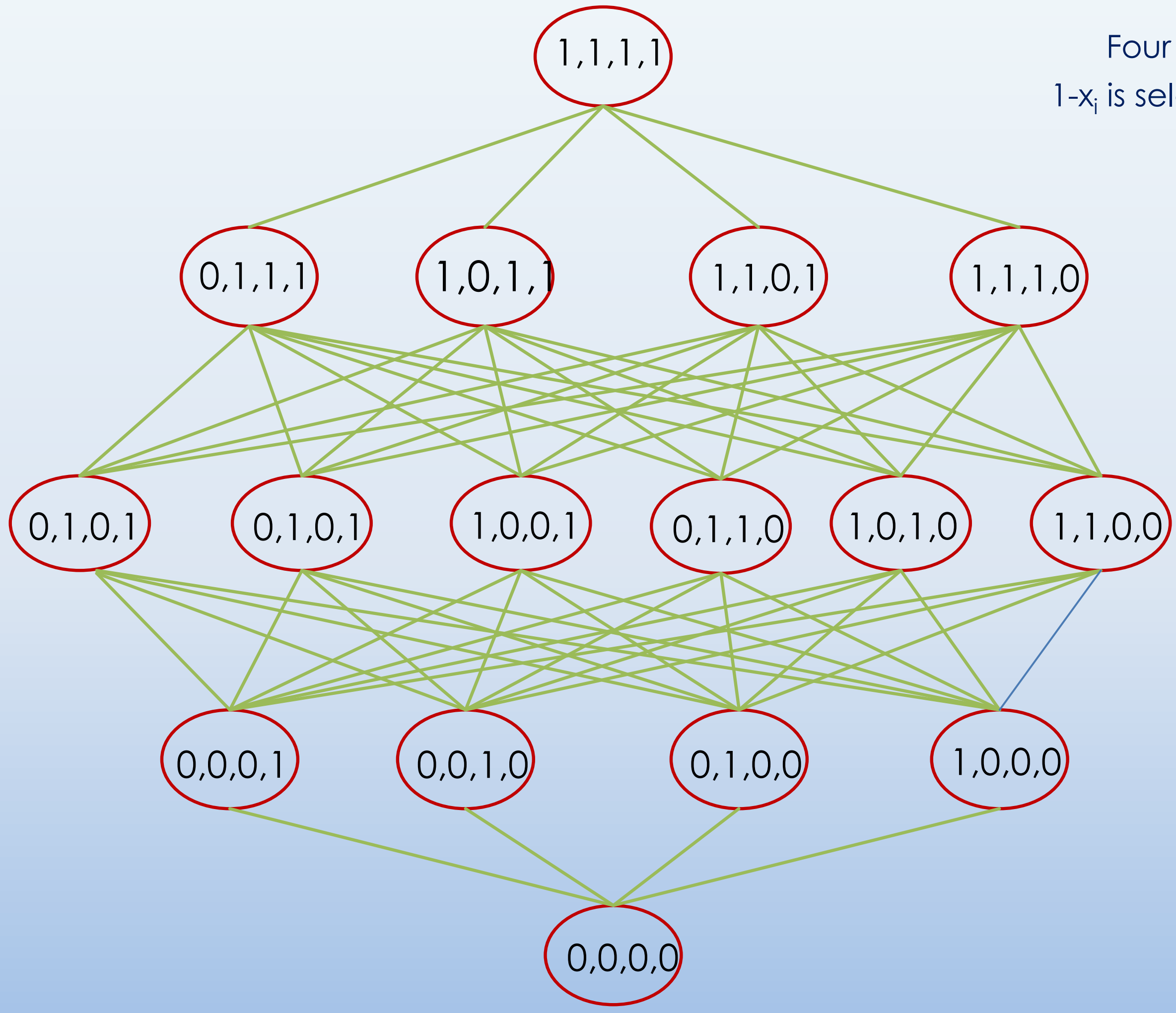
$\{x_1, x_2, x_3, x_4\}$

$\{x_2, x_3, x_1\}\{x_2, x_3, x_4\}$ $J(x_2, x_3, x_1) >= J(x_2, x_3, x_4)$

$\{x_2, x_1\}$ $\{x_2, x_3\}$ $\{x_2, x_4\}$ $J(x_2, x_3) >= J(x_2, x_i);\ i = 1,4$

$\{x_1\}$ $\{x_2\}$ $\{x_3\}$ $\{x_4\}$ $J(x_2) >= J(x_i);\ i = 1,3,4$

Illustration (SFS)



Four Features – x_1, x_2, x_3, x_4

1-x_i is selected; 0-x_i is not selected

Illustration (SFS)

Four Features – x_1, x_2, x_3, x_4

1-x_i is selected; 0-x_i is not selected

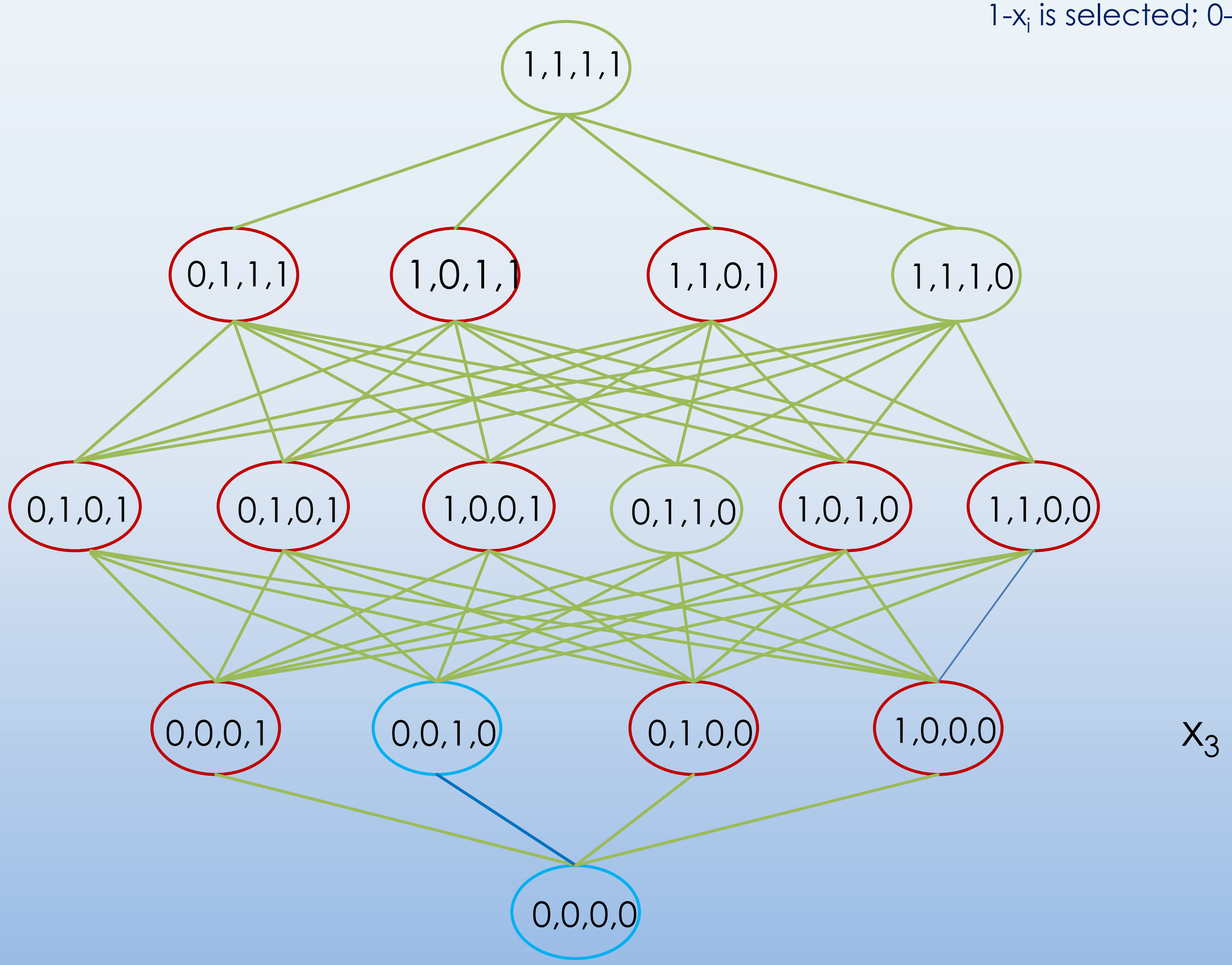


Illustration (SFS)

Four Features – x_1, x_2, x_3, x_4

1-x_i is selected; 0-x_i is not selected

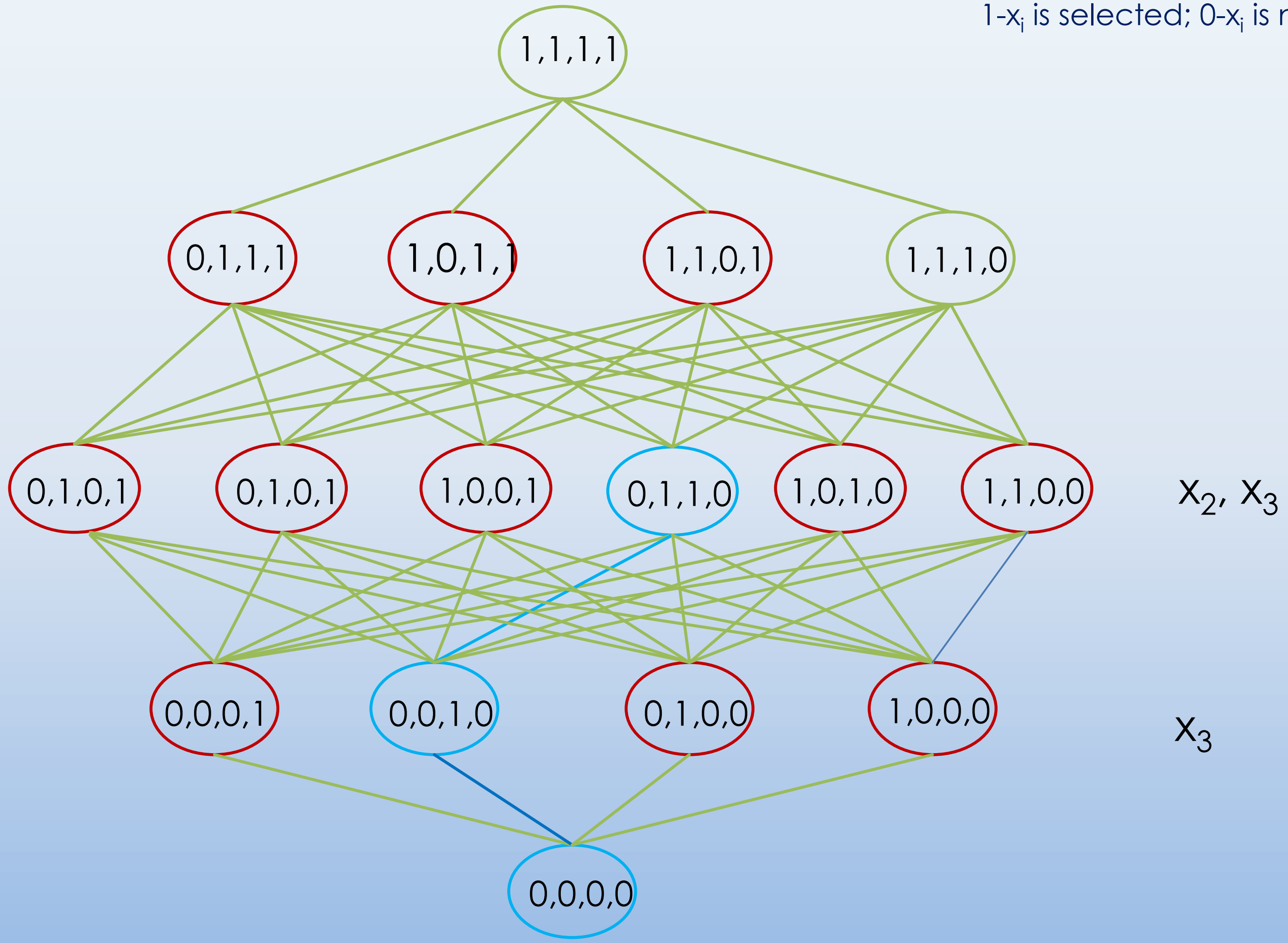


Illustration (SFS)

Four Features – x_1, x_2, x_3, x_4

1-x_i is selected; 0-x_i is not selected

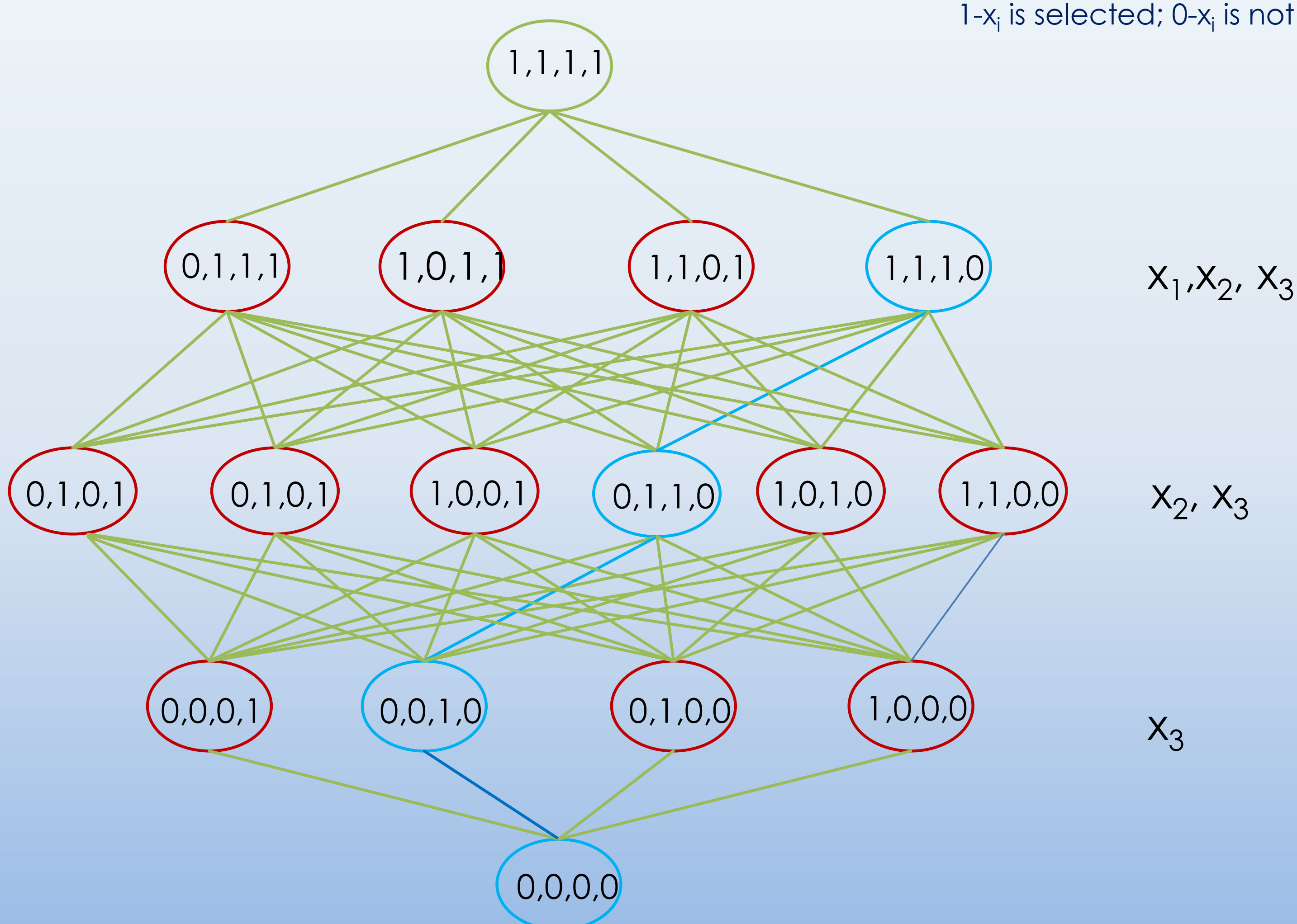
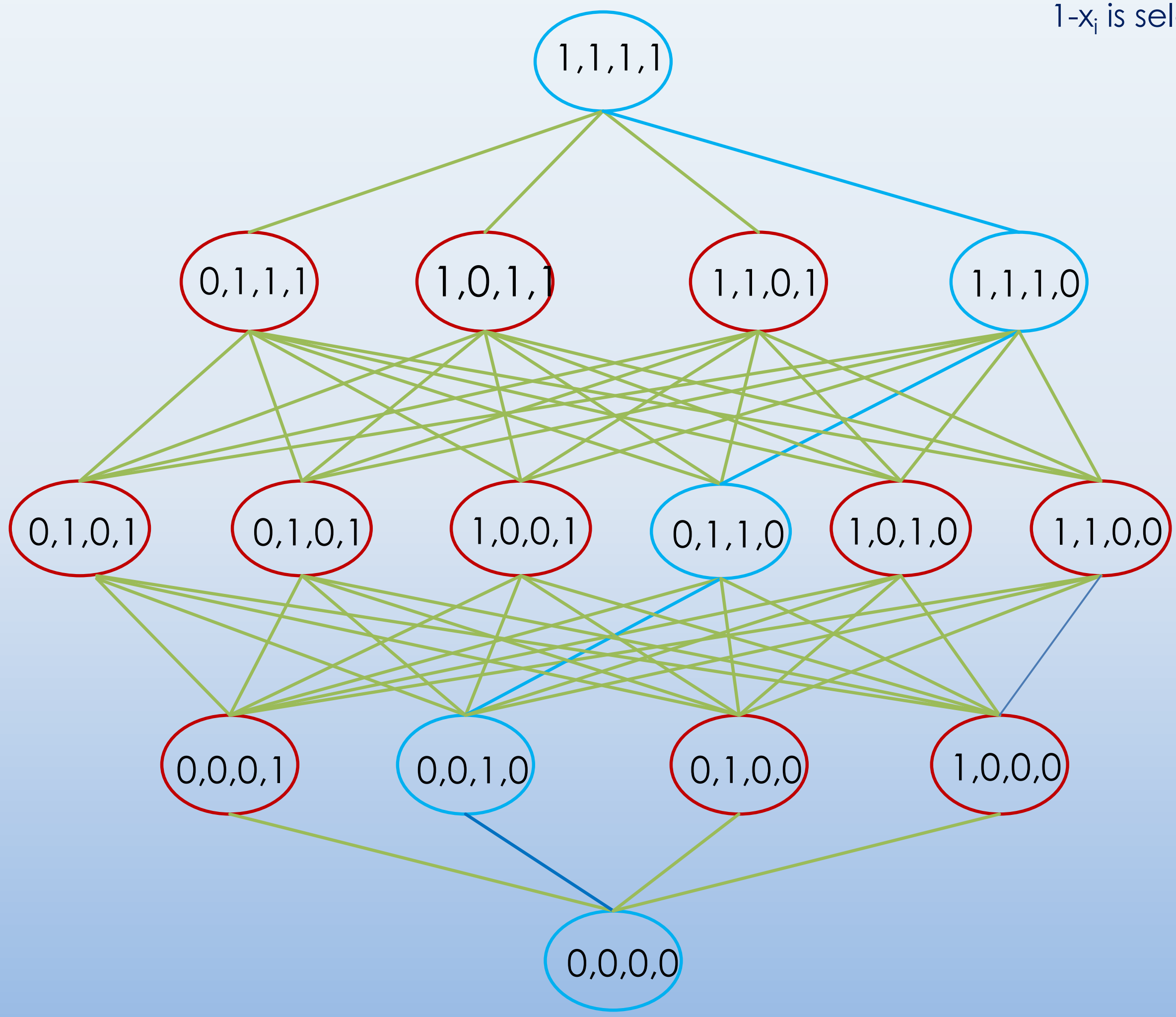


Illustration (SFS)

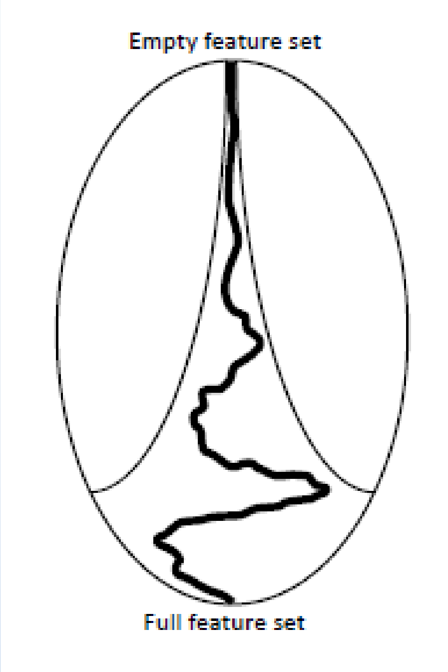
Four Features – x_1, x_2, x_3, x_4

1-x_i is selected; 0-x_i is not selected



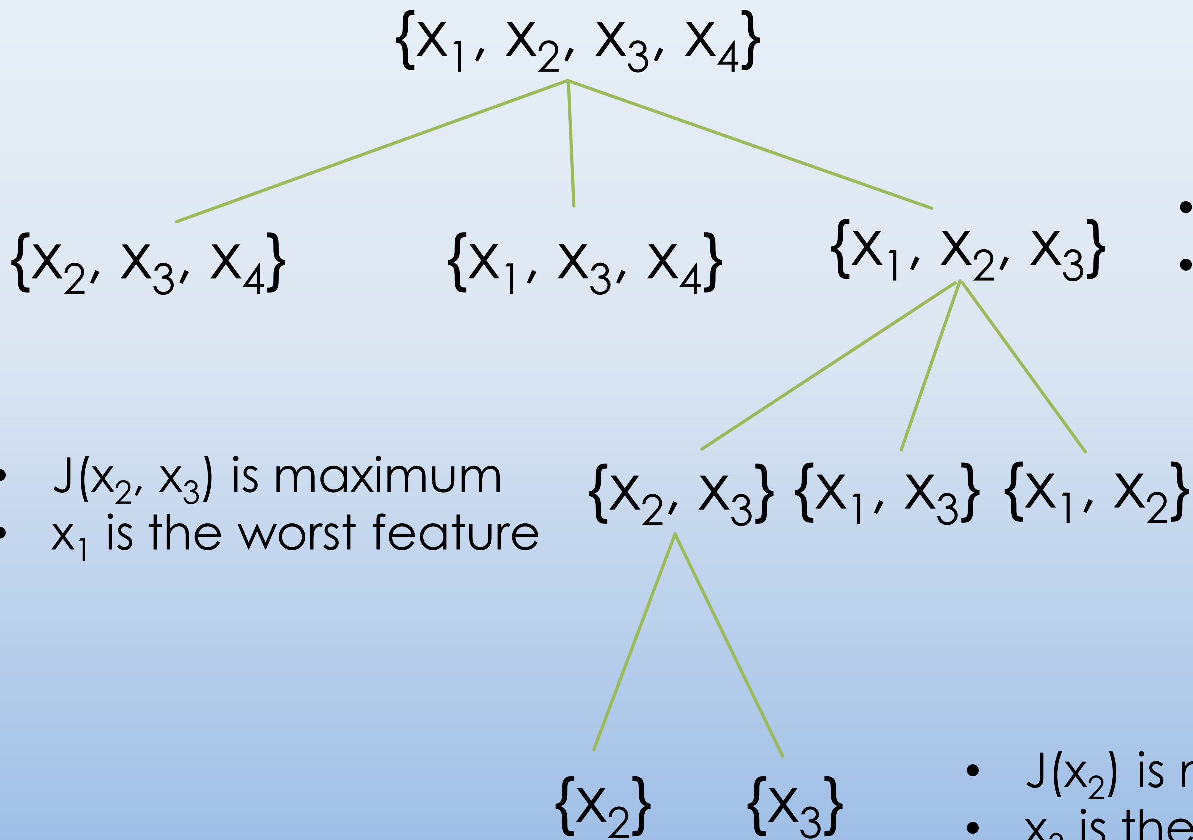
Sequential backward selection (SBS) (heuristic search)

- First, the criterion function is computed for all n features.
- Then, each feature is deleted one at a time, the criterion function is computed for all subsets with n-1 features, and the worst feature is discarded.
- Next, each feature among the remaining n-1 is deleted one at a time, and the worst feature is discarded to form a subset with n-2 features.
- This procedure continues until a predefined number of features are left.



SBS performs best when the optimal subset is large.

Sequential backward selection (SBS) (heuristic search)

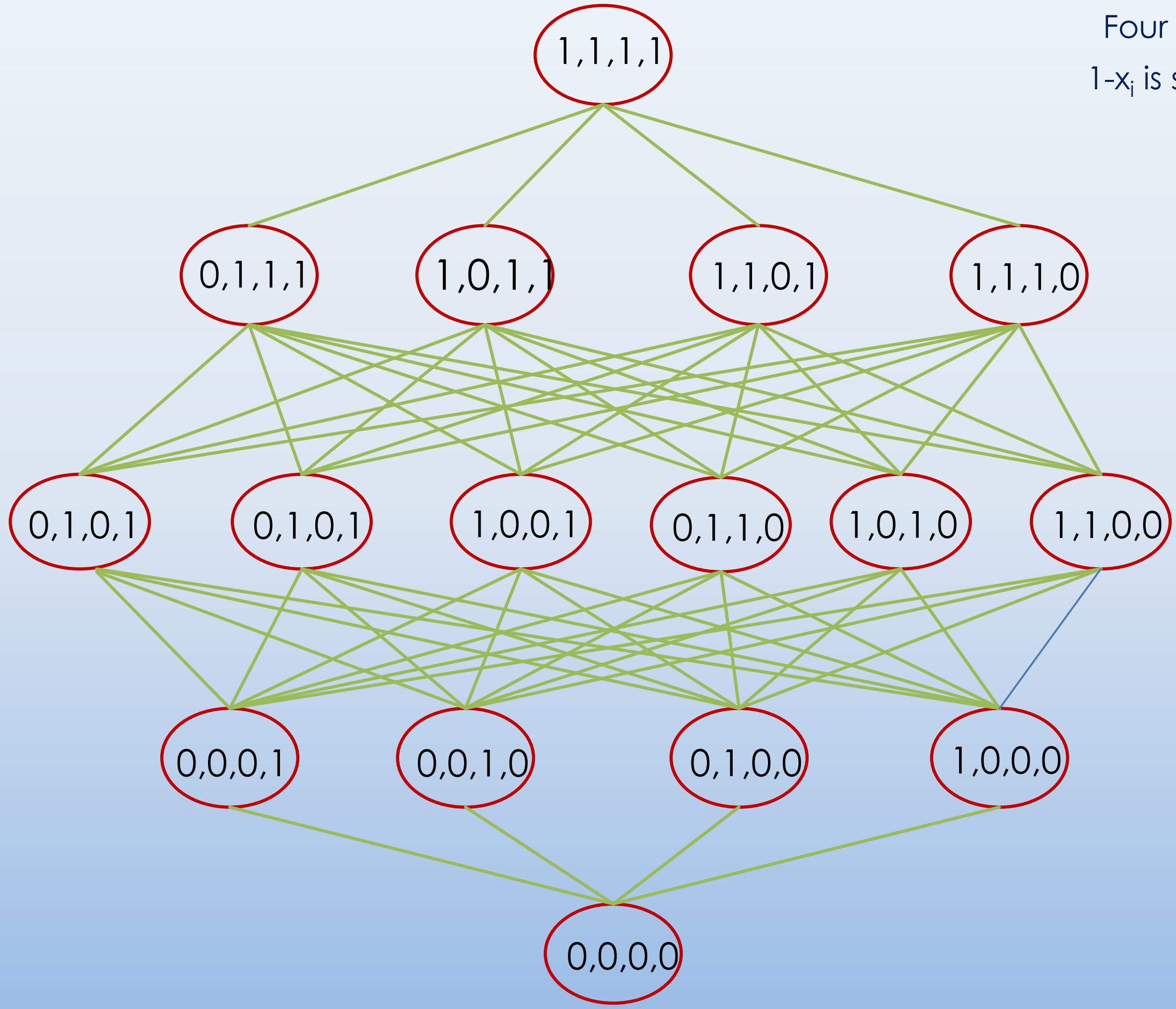


- $J(x_1, x_2, x_3)$ is maximu
- x_3 is the worst feature

- $J(x_2, x_3)$ is maximum
- x_1 is the worst feature

- $J(x_2)$ is maximum
- x_3 is the worst feature

Sequential backward selection (SBS) (heuristic search)



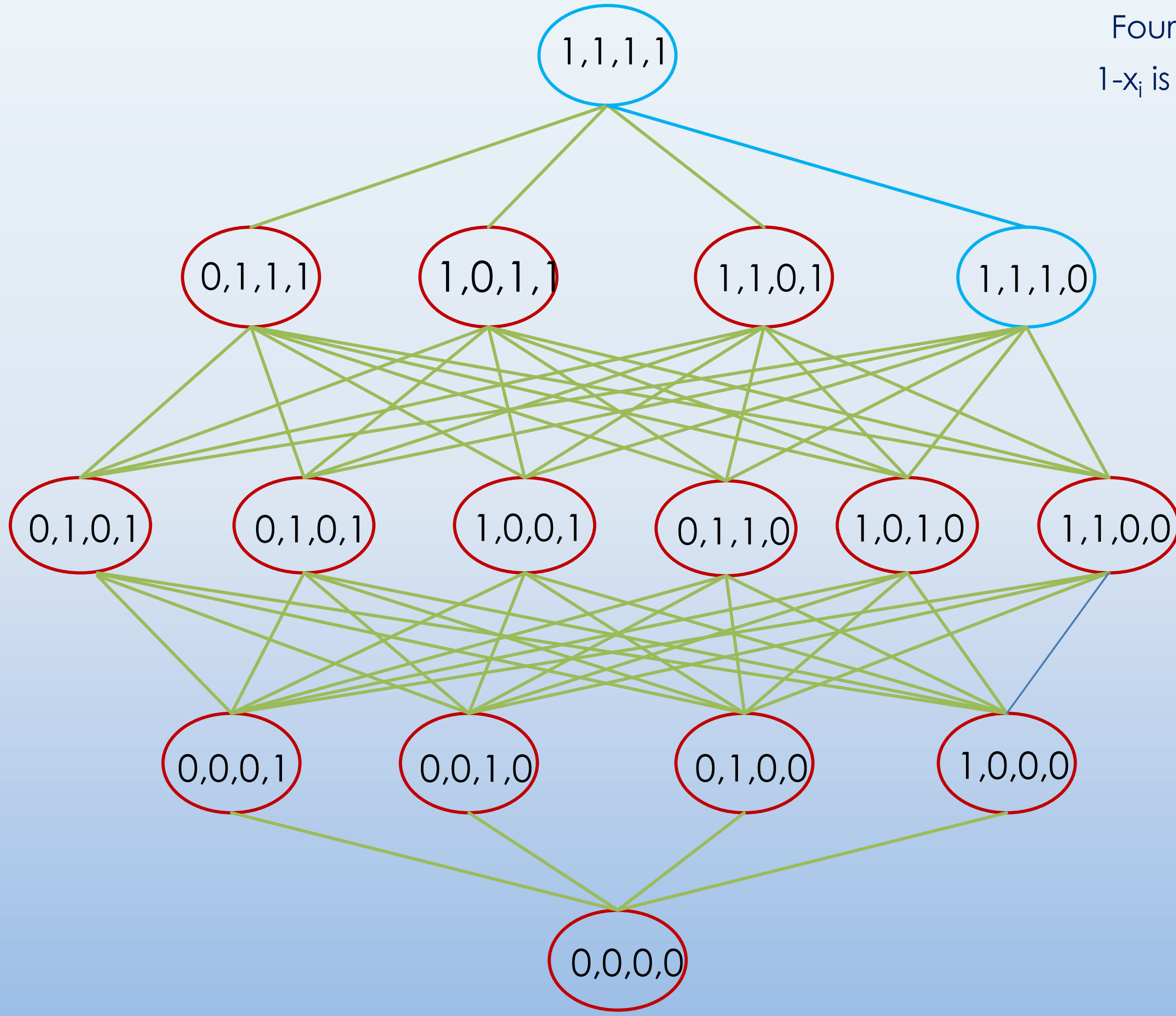
Four Features – x_1, x_2, x_3, x_4

1-x_i is selected; 0-x_i is not selected

Illustration (SFS)

Four Features – x_1, x_2, x_3, x_4

1-x_i is selected; 0-x_i is not selected

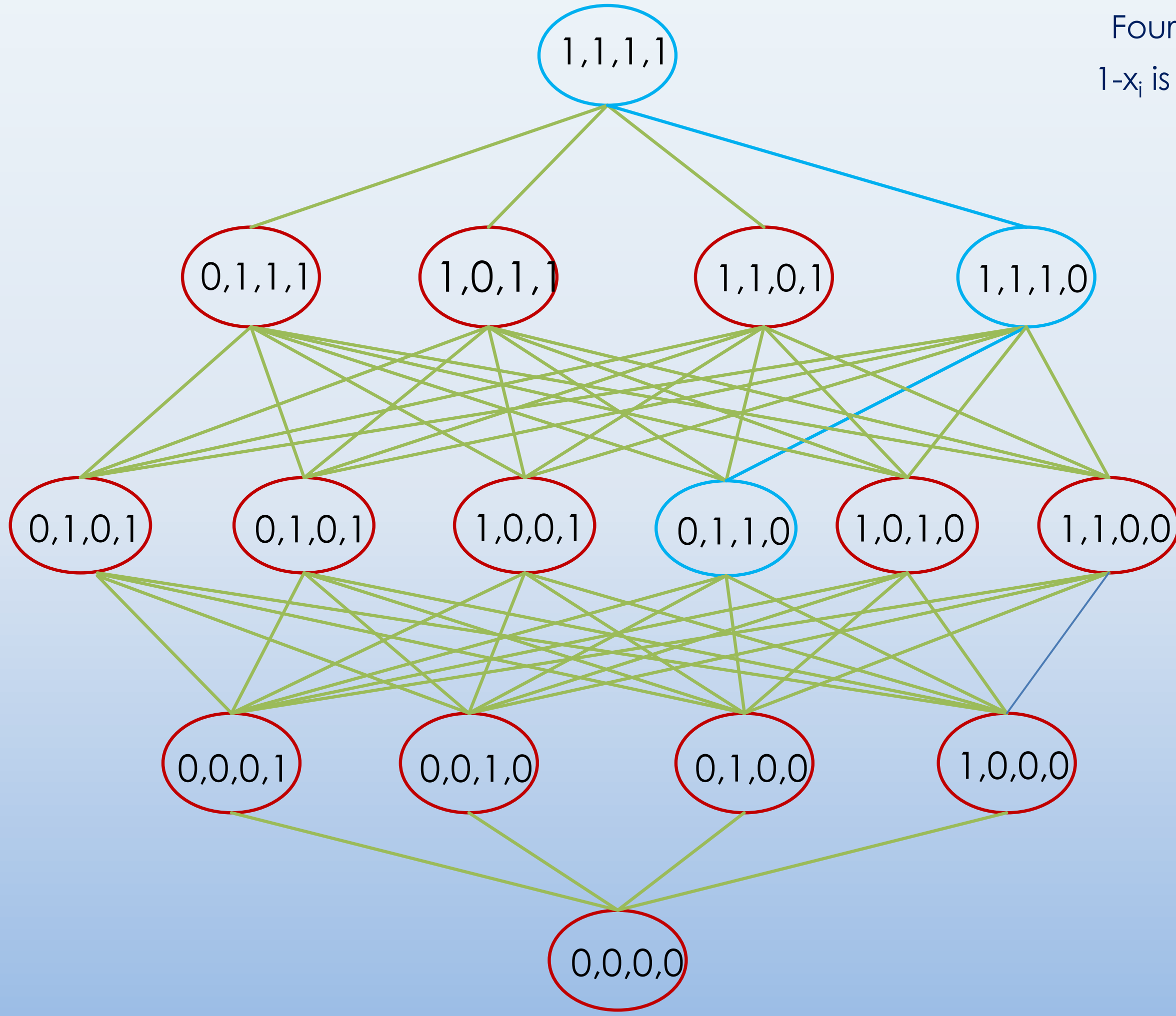


x_1, x_2, x_3

Illustration (SFS)

Four Features – x_1, x_2, x_3, x_4

1-x_i is selected; 0-x_i is not selected



x_1, x_2, x_3

x_2, x_3

Illustration (SFS)

Four Features – x_1, x_2, x_3, x_4

1-x_i is selected; 0-x_i is not selected

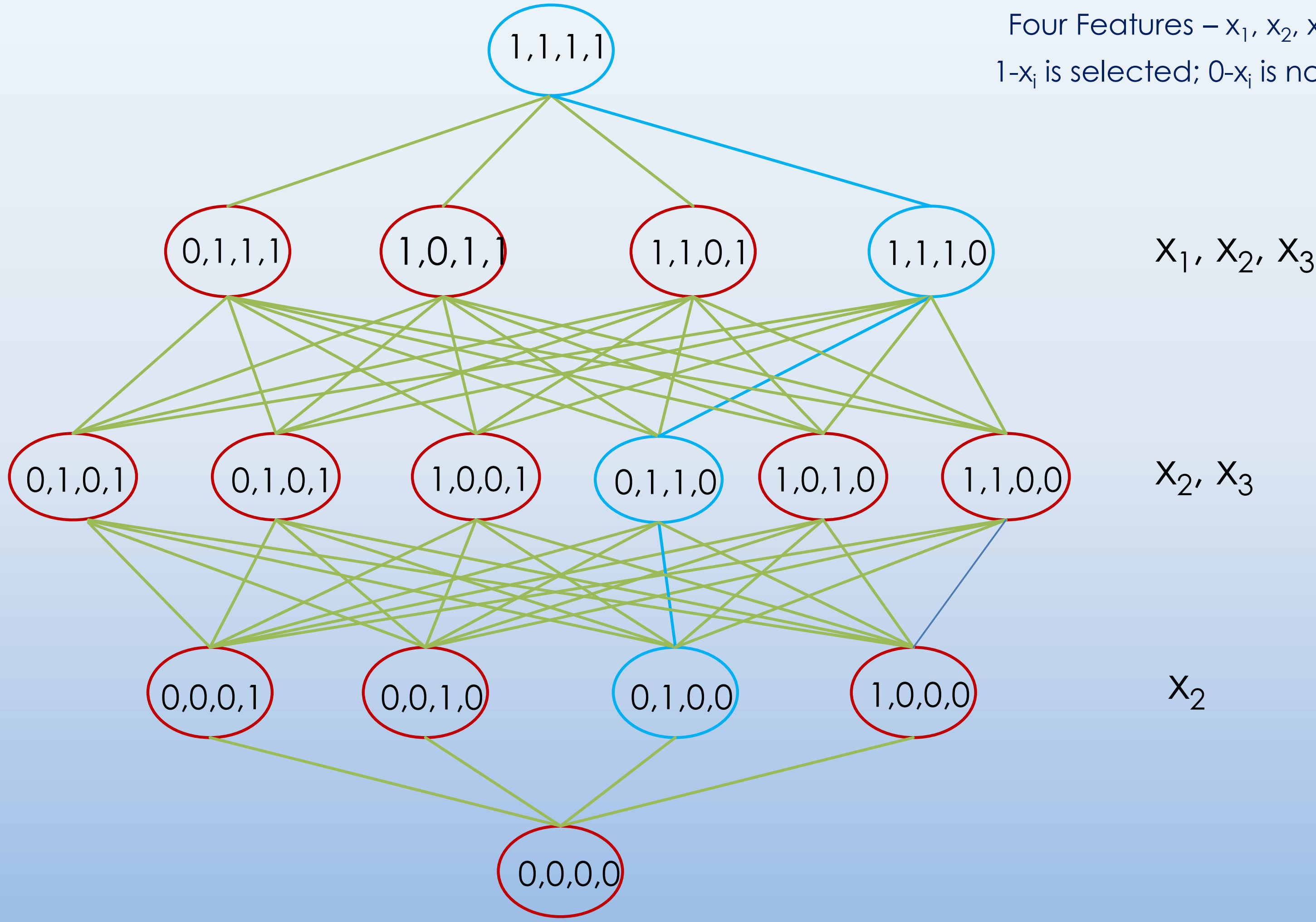
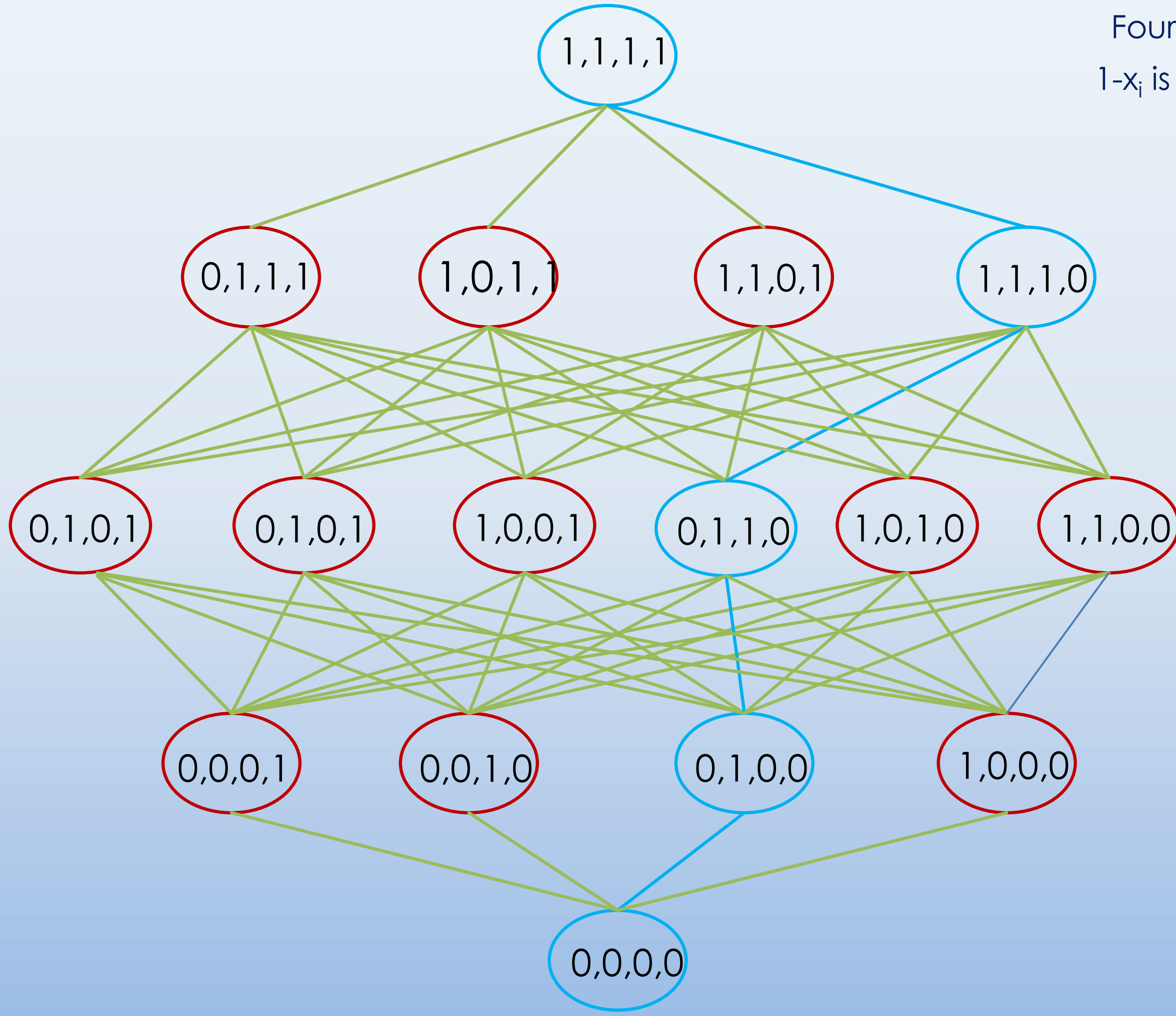


Illustration (SFS)

Four Features – x_1, x_2, x_3, x_4

1-x_i is selected; 0-x_i is not selected



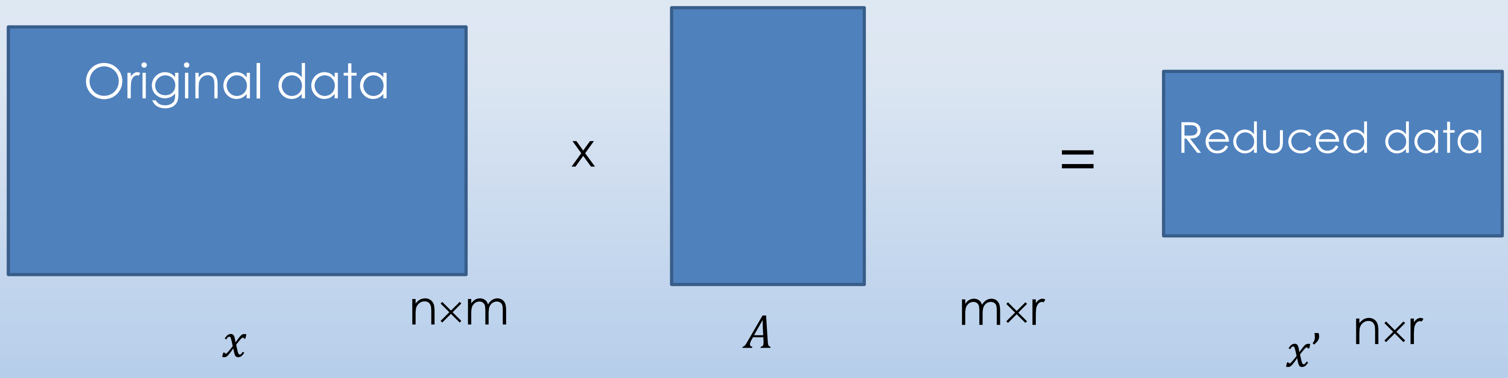
x_1, x_2, x_3

x_2, x_3

x_2

Linear Transformation

- For linear transformation, we find an explicit mapping
- $F(x) = xA$ that can transform also new data vectors.



Linear Dimensionality Reduction

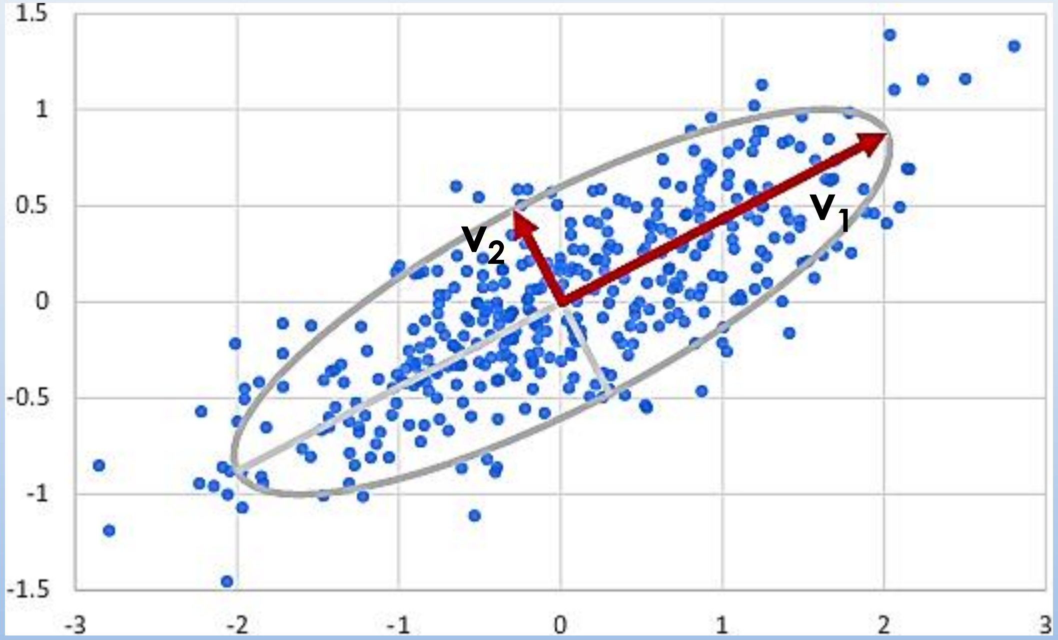
- Unsupervised
  - Principal Component Analysis (PCA)
  - Independent Component Analysis (ICA)
  - Singular Value Decomposition (SVD)
  - Multi Dimensional Scaling (MDS)
  - Canonical Correlation Analysis (CCA)
- Supervised
  - Fisher's Linear Discriminant Analysis (LDA)

Principal Component Analysis (PCA)

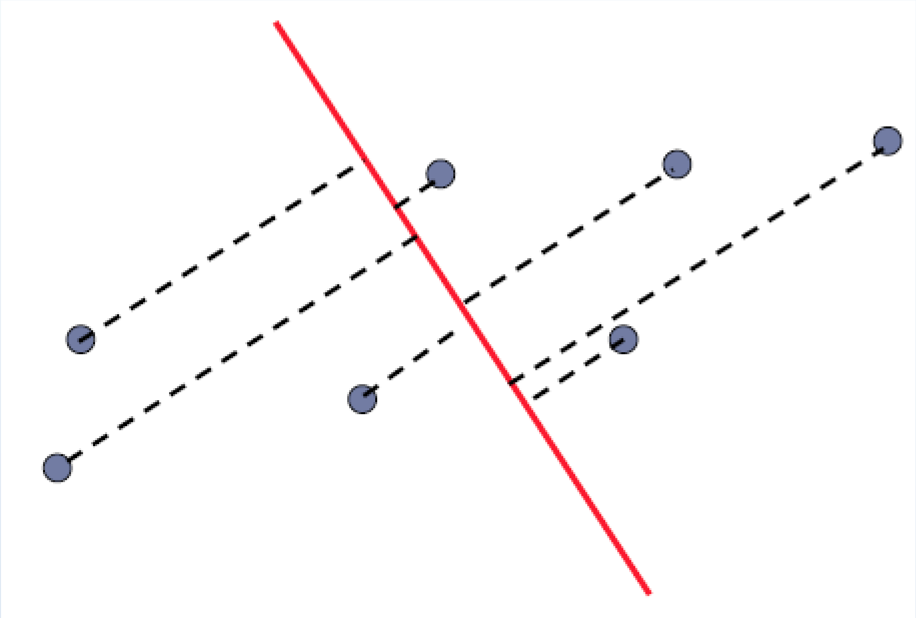
- Find a low-dimensional space such that when x is projected there, information loss is minimized.
- Also known as **Karhonen-Loeve (KL)** transform in signal processing or **Hotelling transform**
- Principal Components (PCs): orthogonal vectors that are ordered by the fraction of the total information (variation) in the corresponding directions
  - Find the directions at which data approximately lie

Principal components

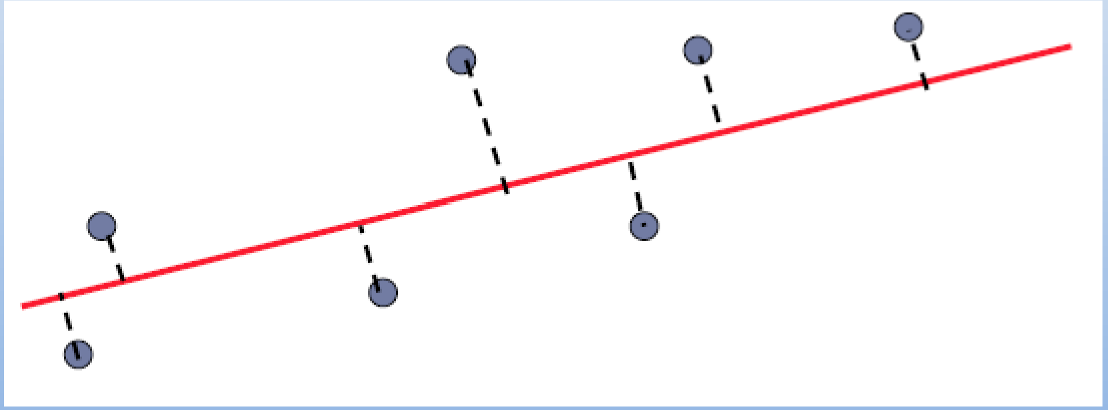
- If data has a Gaussian distribution $\boldsymbol{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the direction of the largest variance can be found by the eigenvector of $\boldsymbol{\Sigma}$ that corresponds to the largest eigenvalue of $\boldsymbol{\Sigma}$



Example: random direction vs. principal component



- Find the direction that preserves important aspect of data



Principal Component Analysis (PCA)

- **Goal**: reducing the dimensionality of the data while preserving important aspects of the data

- Two equal views: find directions for which
  - the variation presents in the dataset is as much as possible.
  - the reconstruction error is minimized.

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

Covariance Matrix

$$\boldsymbol{\mu}_x = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_d \end{bmatrix} = \begin{bmatrix} E(x_1) \\ \vdots \\ E(x_d) \end{bmatrix}$$

$$\boldsymbol{\Sigma} = E[(\boldsymbol{x} - \boldsymbol{\mu}_x)(\boldsymbol{x} - \boldsymbol{\mu}_x)^T]$$

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

$$\boldsymbol{s} = \frac{1}{N}\sum_{i=1}^{N}(x^{(i)} - \overline{x})(x^{(i)} - \overline{x})^T = \frac{1}{N}(\tilde{X}^T\tilde{X})$$

$$\tilde{\boldsymbol{X}} = \begin{bmatrix} \tilde{\boldsymbol{x}}^{(1)} \\ \vdots \\ \tilde{\boldsymbol{x}}^{(N)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}^{(1)} - \overline{\boldsymbol{x}} \\ \vdots \\ \boldsymbol{x}^{(N)} - \overline{\boldsymbol{x}} \end{bmatrix} \qquad \overline{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}^{(i)}$$

Mean-centered data

PCA

- Eigenvalues: $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots$
  - The first PC $\boldsymbol{v_1}$ is the eigenvector of the sample covariance matrix $\boldsymbol{S}$ associated with the largest eigenvalue.
  - The 2nd PC $\boldsymbol{v_2}$ is the eigenvector of the sample covariance matrix S associated with the second largest eigenvalue
  - And so on ...

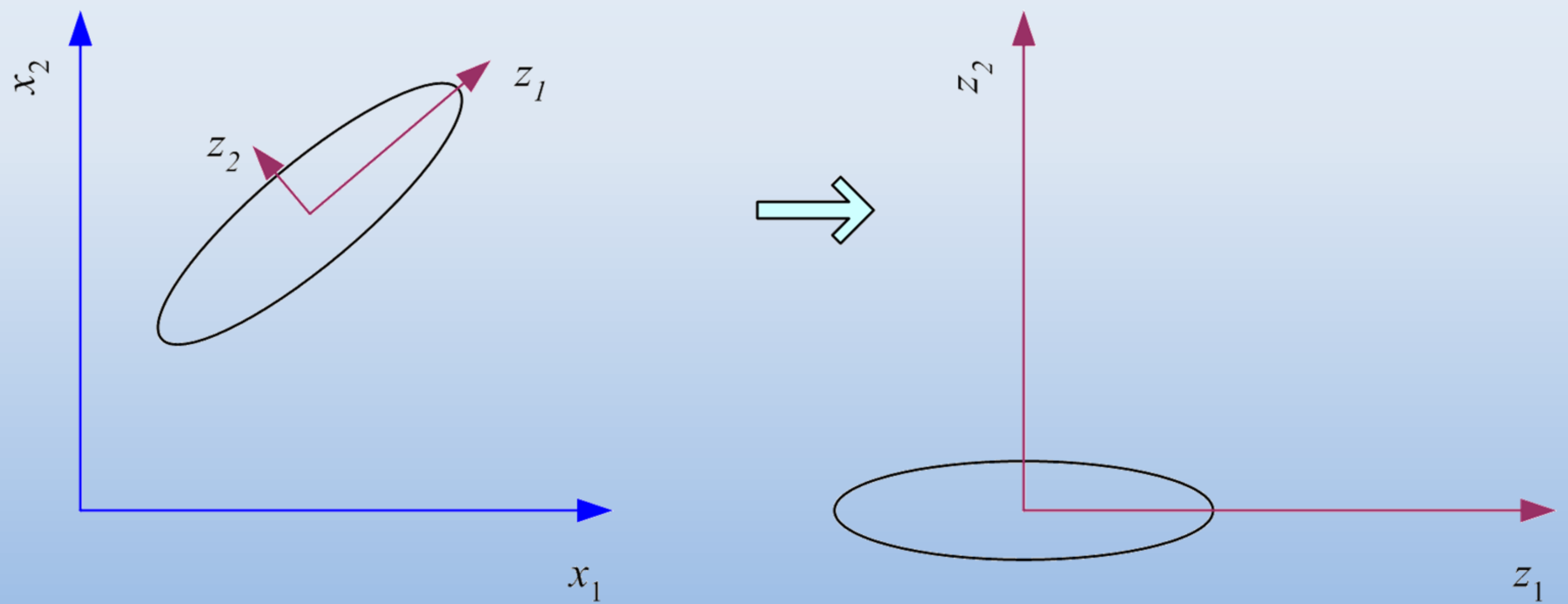
- Find eigenvectors with the top *k* eigenvalues

What PCA does

$$\mathbf{z} = \mathbf{W}^T(\mathbf{x} - \boldsymbol{m})$$

where the columns of **W** are the eigenvectors of $\sum$ and ***m*** is sample mean

Centers the data at the origin and rotates the axes



PCA: Steps

- **Input**: $n \times m$ data matrix $\boldsymbol{X}$ (each row contain a m-dimensional data point)
  - $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$
  - $\tilde{X} \leftarrow$ Mean value of data points is subtracted from rows of $\boldsymbol{X}$
  - $S = \frac{1}{N}\tilde{X}^T\tilde{X}$ (Covariance matrix)
  - Calculate eigenvalue and eigenvectors of $\boldsymbol{S}$
  - Pick $\boldsymbol{r}$ eigenvectors corresponding to the largest eigenvalues and put them in the columns of $\boldsymbol{A}$ $= [v_1, \dots, v_r]$

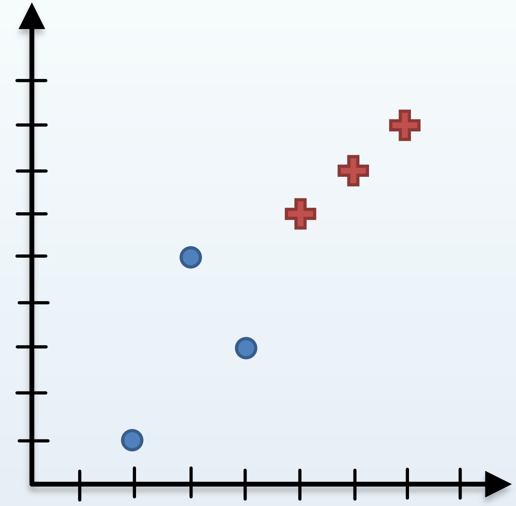
First PC

r-th PC

  - $X' = X\boldsymbol{A}$

PCA example

- Input data: $X = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 5 & 3 & 6 & 7 & 8 \end{bmatrix}$



$$\bar{X} = \begin{bmatrix} \frac{2+3+4+5+6+7}{6} \\ \frac{1+5+3+6+7+8}{6} \end{bmatrix} = \begin{bmatrix} 4.5 \\ 5 \end{bmatrix}$$

$$S = \frac{1}{N}\tilde{X}^T\tilde{X}$$

$$\tilde{X} = \begin{bmatrix} 2-4.5 & 3-4.5 & 4-4.5 & 5-4.5 & 6-4.5 & 7-4.5 \\ 1-5 & 5-5 & 3-5 & 6-5 & 7-5 & 8-5 \end{bmatrix} = \begin{bmatrix} -2.5 & -1.5 & -0.5 & 0.5 & 1.5 & 2.5 \\ -4 & 0 & -2 & 1 & 2 & 3 \end{bmatrix}$$

$$S = \begin{bmatrix} -2.5 & -1.5 & -0.5 & 0.5 & 1.5 & 2.5 \\ -4 & 0 & -2 & 1 & 2 & 3 \end{bmatrix} \times \begin{bmatrix} -2.5 & -4 \\ -1.5 & 0 \\ -0.5 & -2 \\ 0.5 & 1 \\ 1.5 & 2 \\ 2.5 & 3 \end{bmatrix} = \begin{bmatrix} \frac{17.5}{6} & \frac{22}{6} \\ \frac{22}{6} & \frac{34}{6} \end{bmatrix} = \begin{bmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{bmatrix}$$

PCA example (cont.)

$$\begin{bmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = 0 \quad , \begin{bmatrix} 2.92 - \lambda & 3.67 \\ 3.67 & 5.67 - \lambda \end{bmatrix} = 0$$

$$(2.92 - \lambda)(5.67 - \lambda) - (3.67 \times 3.67) = 0 \quad , \quad 16.56 - 2.92\lambda - 5.67\lambda + \lambda^2 = 0$$

$$\lambda^2 - 8.59\lambda + 3.09 = 0$$

$$\Delta = b^2 - 4ac = (8.59 \times 8.59) - 4 \times (1) \times (3.09) = 73.78 - 12.36 = 61.42$$

$$\lambda_1 = \frac{8.59 + \sqrt{61.42}}{2} = 8.22 \ , \ \lambda_2 = \frac{8.59 - \sqrt{61.42}}{2} = 0.37$$

$$\begin{bmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 8.22 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{cases} 2.92x_1 + 3.67x_2 = 8.22x_1 \\ 3.67x_1 + 5.67x_2 = 8.22x_2 \end{cases} \qquad \begin{cases} 3.67x_2 = 5.3x_1 \\ 3.67x_1 = 2.55x_2 \end{cases} \qquad x_1 = 0.69x_2$$

$$eigenvector = \begin{bmatrix} 0.56 & -0.82 \\ 0.82 & 0.56 \end{bmatrix}$$

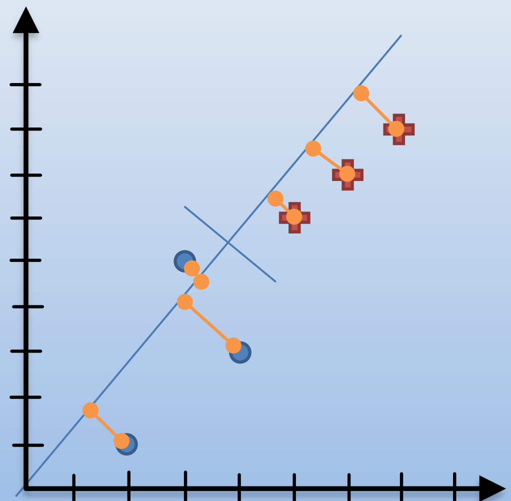
PCA example (cont.)

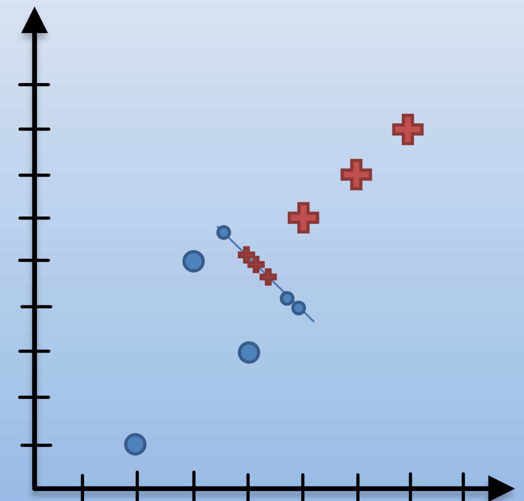
- $X' = X^T \boldsymbol{A}$

$$eigenvector = \begin{bmatrix} 0.56 & -0.82 \\ 0.82 & 0.56 \end{bmatrix}$$

$$x_1 = 0.69x_2$$

$$X = \begin{bmatrix} -2.5 & -4 \\ -1.5 & 0 \\ -0.5 & -2 \\ 0.5 & 1 \\ 1.5 & 2 \\ 2.5 & 3 \end{bmatrix} \times \begin{bmatrix} 0.56 & -0.82 \\ 0.82 & 0.56 \end{bmatrix} = \begin{bmatrix} -4.29 & -0.19 \\ -1.23 & 1.23 \\ -1.53 & -0.71 \\ 0.97 & 0.15 \\ 2.35 & -0.11 \\ 3.73 & -0.37 \end{bmatrix}$$





Dimensionality reduction by PCA

- Data may lie near a linear subspace of high-dimensional input space
- Only keep data projections onto principal components with large eigenvalues
- Proportion of Variance (PoV) explained

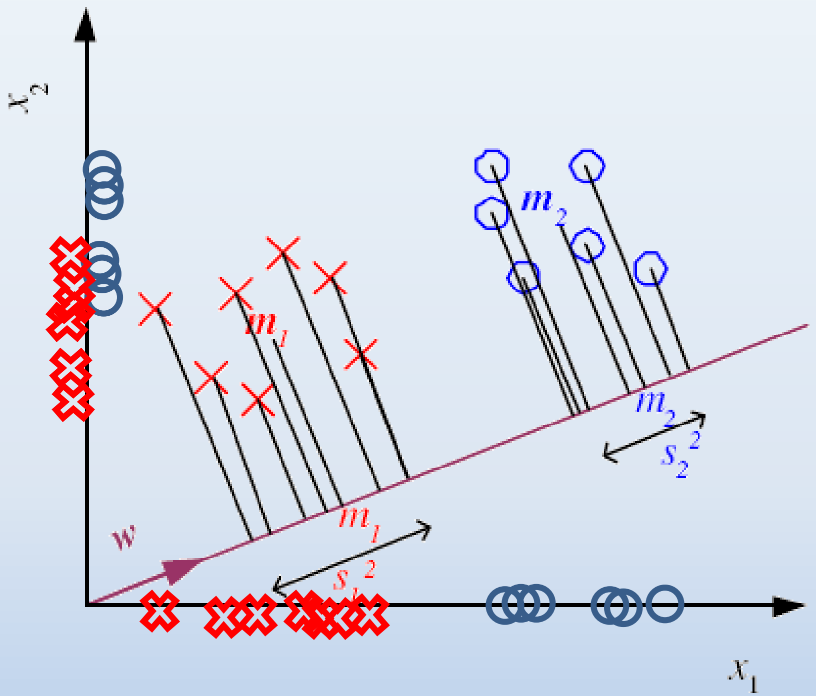
$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d}$$

when λ_i are sorted in descending order

- Typically, stop at PoV>0.9
- Scree graph plots of PoV vs k, stop at "elbow"

Linear Discriminant Analysis

- Find a low-dimensional space such that when **x** is projected, classes are well-separated.
- Find **w** that maximizes



$$J(\mathbf{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}$$

$$m_1 = \frac{\sum_t \mathbf{w}^T \mathbf{x}^t r^t}{\sum_t r^t} \qquad s_1^2 = \sum_t \left(\mathbf{w}^T \mathbf{x}^t - m_1\right)^2 r^t$$

LDA Algorithm

- Find $\mathbf{m_1}$ and $\mathbf{m_2}$ as the mean of class 1 and 2 respectively $\boldsymbol{m}_i = \frac{1}{n_i} \sum_{\boldsymbol{x} \in D_i}^{n} \boldsymbol{x}_k$
- Find $\mathbf{S_1}$ and $\mathbf{S_2}$ as scatter matrix of class 1 and 2 respectively $S_i = \sum_{\boldsymbol{x} \in D_i}^{n} (\boldsymbol{x} - \boldsymbol{m}_i)(\boldsymbol{x} - \boldsymbol{m}_i)^T$
  - $S_w = S_1 + S_2$
  - $S_B = (m_1 - m_2)(m_1 - m_2)^T$

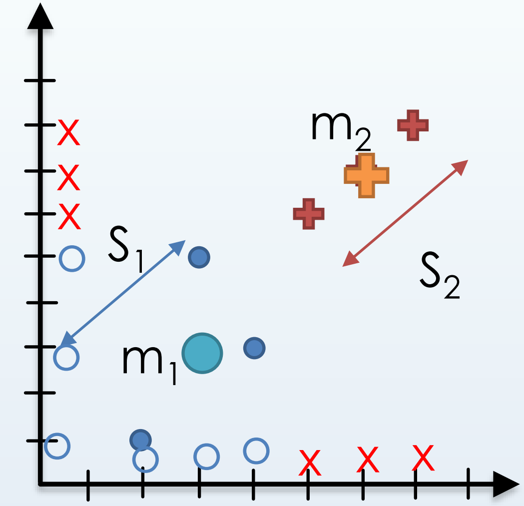
$$S_W = \sum_{i=1}^{c} S_i$$

$$S_B = \sum_{i=1}^{c} N_i (\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T$$

- Solving eigenvalues for $\boldsymbol{w} = S_w^{-1} S_b$
- Selecting the k eigenvectors corresponding to the largest eigenvalues of $S_w^{-1} S_b$
- Transforming by $\boldsymbol{Y} = \boldsymbol{X} \times \boldsymbol{w}$

LDA example

- Input data: $X = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 5 & 3 & 6 & 7 & 8 \end{bmatrix}$



$$m_1 = \begin{bmatrix} \frac{2+3+4}{3} \\ \frac{1+5+3}{3} \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix} \qquad m_2 = \begin{bmatrix} \frac{5+6+7}{3} \\ \frac{6+7+8}{3} \end{bmatrix} = \begin{bmatrix} 6 \\ 7 \end{bmatrix}$$

$$S_1 = \begin{bmatrix} 2-3 & 3-3 & 4-3 \\ 1-3 & 5-3 & 3-3 \end{bmatrix} \times \begin{bmatrix} -1 & -2 \\ 0 & 2 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 2 & 4 \end{bmatrix} \qquad S_B = \begin{bmatrix} 3-6 \\ 3-7 \end{bmatrix} \times \begin{bmatrix} 3-6 & 3-7 \end{bmatrix} = \begin{bmatrix} 9 & 12 \\ 12 & 16 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 5-3 & 6-3 & 7-3 \\ 6-3 & 7-3 & 8-3 \end{bmatrix} \times \begin{bmatrix} 2 & 3 \\ 3 & 4 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} 29 & 38 \\ 38 & 50 \end{bmatrix} \qquad S_W = \begin{bmatrix} 2 & 2 \\ 2 & 4 \end{bmatrix} + \begin{bmatrix} 29 & 38 \\ 38 & 50 \end{bmatrix} = \begin{bmatrix} 31 & 40 \\ 40 & 54 \end{bmatrix}$$

$$\boldsymbol{w} = S_W^{-1} S_b \qquad \frac{1}{74} \begin{bmatrix} 50 & -40 \\ -40 & 31 \end{bmatrix} \times \begin{bmatrix} 9 & 12 \\ 12 & 16 \end{bmatrix} = \begin{bmatrix} -0.405 & -0.540 \\ 0.162 & 0.216 \end{bmatrix}$$

LDA example (cont.)

$$\begin{bmatrix} -0.405 & -0.540 \\ 0.162 & 0.216 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = 0 \quad , \begin{bmatrix} -0.405 - \lambda & -0.540 \\ 0.162 & 0.216 - \lambda \end{bmatrix} = 0$$

$$(-0.405 - \lambda)(0.216 - \lambda) - (-0.540 \times 0.162) = 0 \quad , 0 + 0.189\lambda + \lambda^2 = 0$$

$$\lambda^2 + 0.189\lambda = 0 \; , \lambda(\lambda + 0.189) = 0$$

$$\lambda_1 = 0, \;\; \lambda_2 = -0.189$$

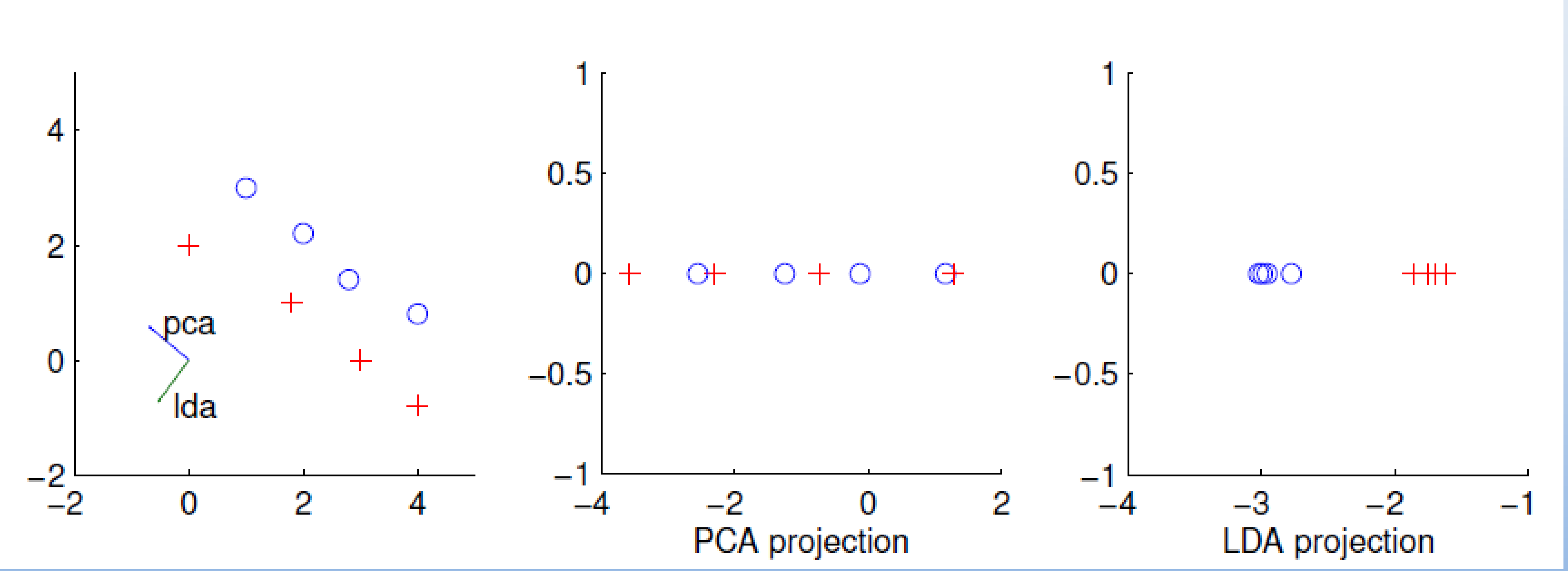
$$\begin{bmatrix} 2.92 & 3.67 \\ 3.67 & 5.67 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 8.22 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{cases} 2.92x_1 + 3.67x_2 = 8.22x_1 \\ 3.67x_1 + 5.67x_2 = 8.22x_2 \end{cases} \qquad \begin{cases} 3.67x_2 = 5.3x_1 \\ 3.67x_1 = 2.55x_2 \end{cases} \qquad x_1 = 0.69x_2$$

$$eigenvector = \begin{bmatrix} -2.5 & -1.33 \\ 1 & 1 \end{bmatrix}$$

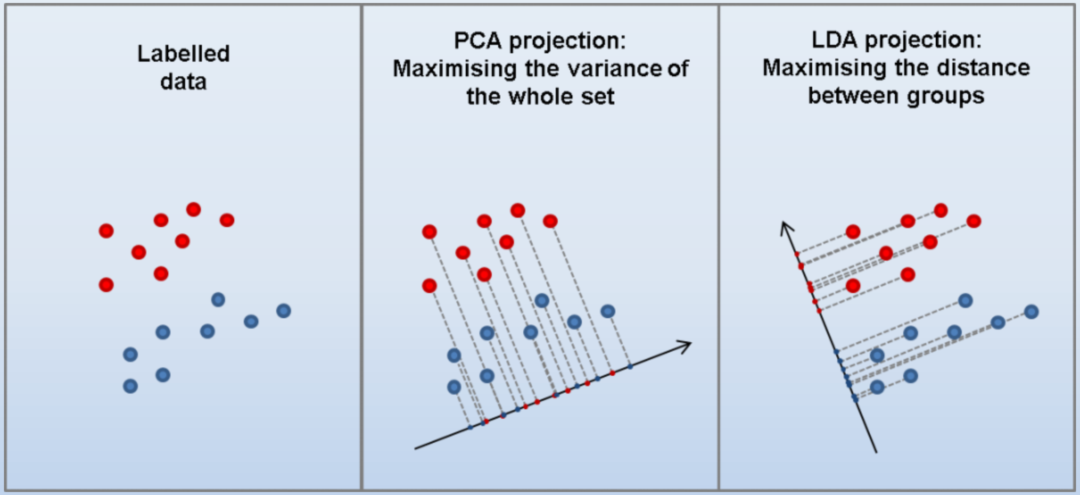
PCA vs LDA

- PCA (unsupervised)
  - Uses Total Scatter Matrix
- LDA (supervised)
  - Uses |between-class scatter matrix| / |within-class scatter matrix|



PCA and LDA: Drawbacks

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



- LDA drawback
  - Matrix Singularity problem (Det=0) or under-sampled problem (when n < m)
  - Example: gene expression data, images, text documents
  - Can reduces dimension only to r $\leq C - 1$ (unlike PCA)

Summary

- 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)
      - e.g., Principal Component Analysis (PCA)
    - Supervised task: maximize the class discrimination on the projected space
      - e.g., Linear Discriminant Analysis (LDA)
- Feature extraction algorithms
  - Linear Methods
  - Non-linear methods:
    - Supervised: MLP neural networks
    - Unsupervised: e.g., auto-encoders, kernel PCA

Reading

- E. Alpaydin, **Introduction to Machine Learning**, 4th ed., The MIT Press, 2020. (ch. 6)
- C. M. Bishop, **Pattern recognition and machine learning**, Springer, 2006. (ch. 12)