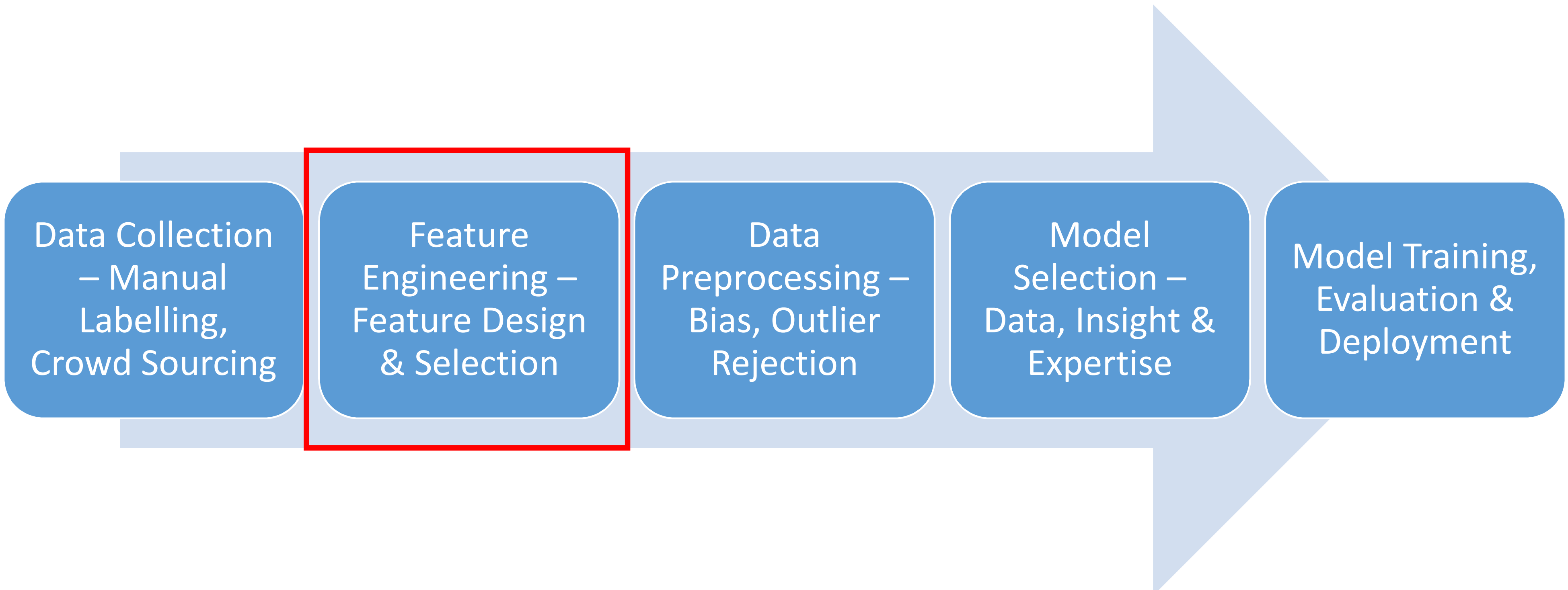
Feature Selection

Prithwijit Guha
Dept. of EEE, IIT Guwahati

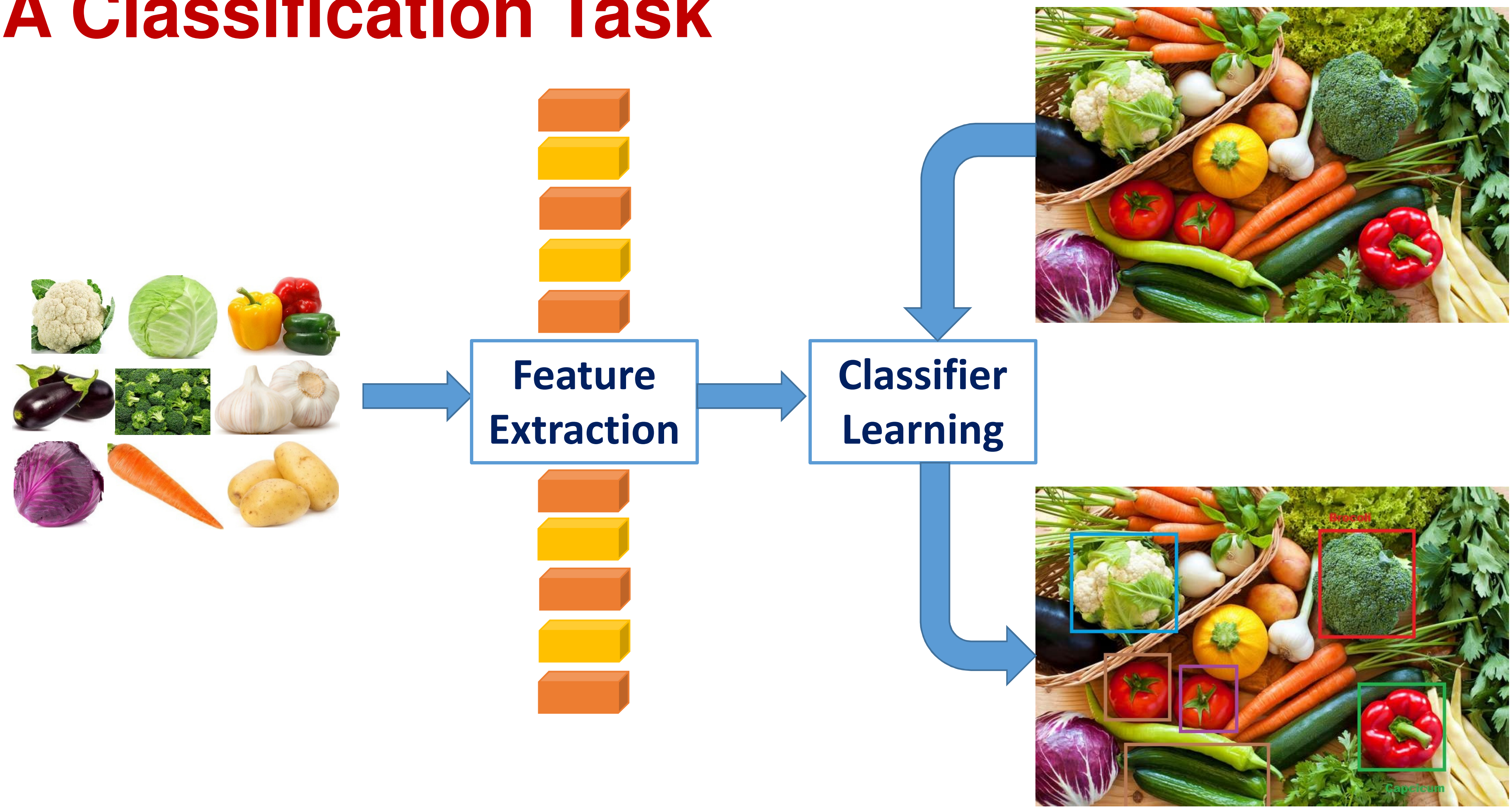
Overview

- Machine Learning Workflow
- Feature Extraction
- Separability & Classification
- The Curse of Dimensionality
- Feature Selection
- Hashing Techniques

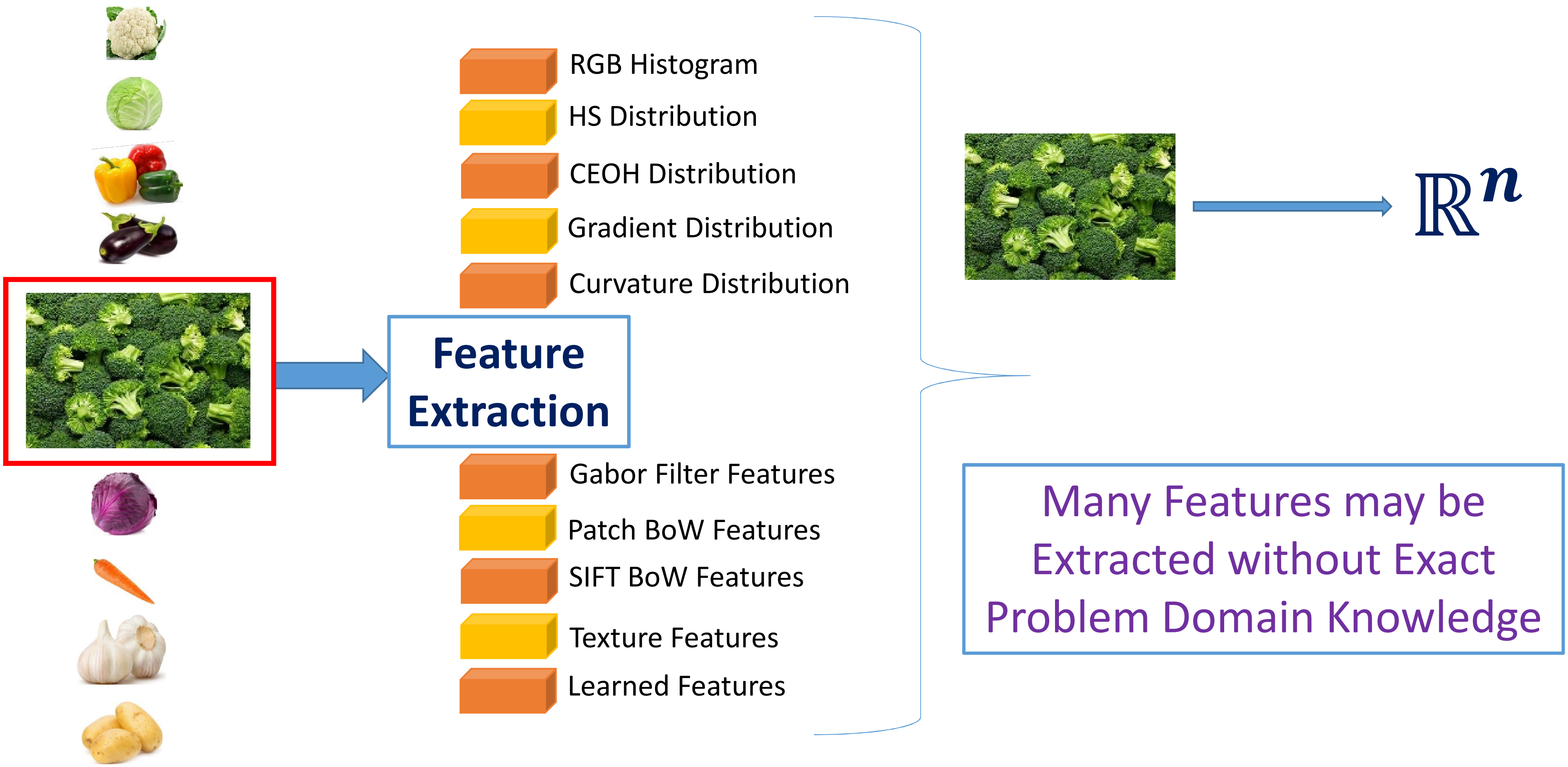
Machine Learning Task Workflow



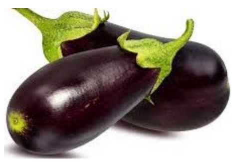
A Classification Task



From Real World to Symbol Space



Feature Extraction



Primary Features

Abstract Symbol Space

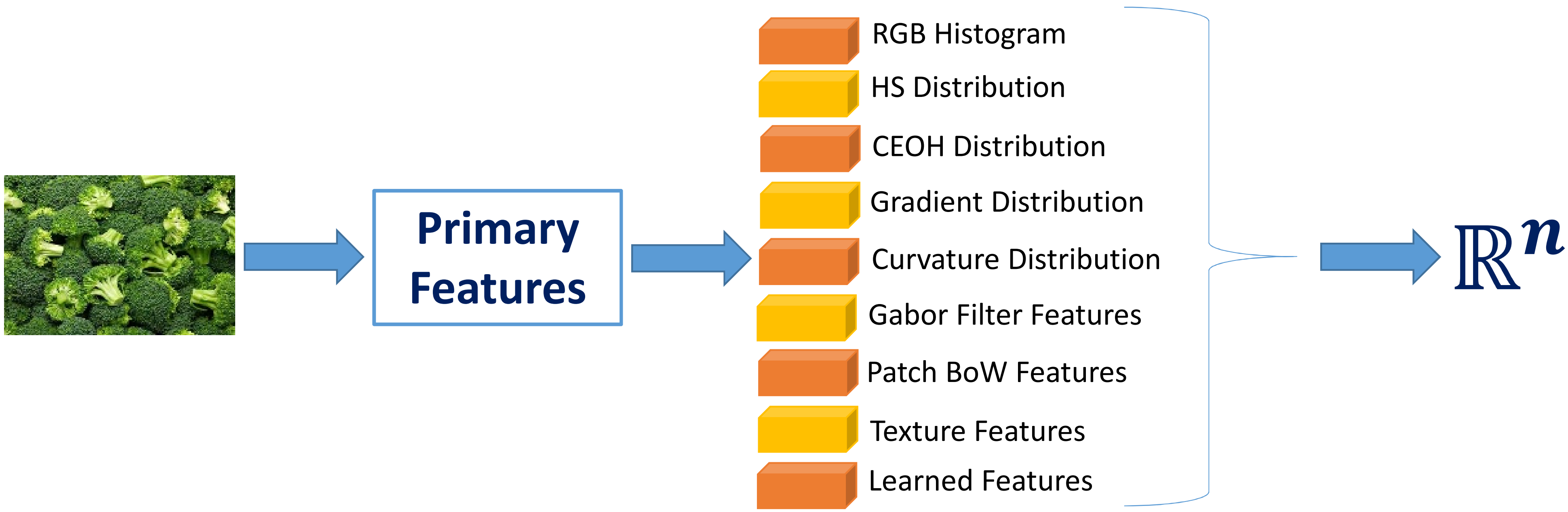
OR

Numerical Vector Space

Secondary Features

Desirably Vector Space

Primary Features: Measurement



Secondary Features Onwards: Transformation

$$\begin{pmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{pmatrix} \Rightarrow \begin{bmatrix} a_{11} & \cdots & a_{nm} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \Rightarrow \begin{pmatrix} y_1 \\ \vdots \\ y_j \\ \vdots \\ y_m \end{pmatrix}$$

$X \in \mathbb{R}^n$ $\quad$ $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ $\quad$ $Y \in \mathbb{R}^m$

Linear Transformation: $Y = \boldsymbol{A}X$

Secondary Features Onwards: Transformation

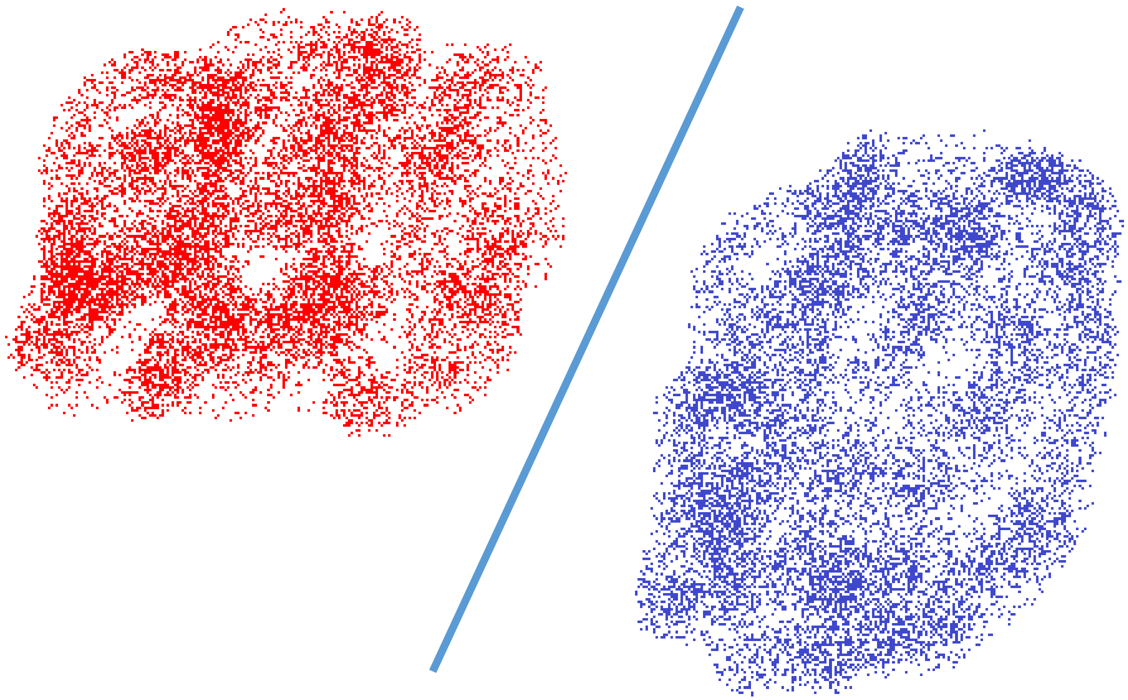
$$
\begin{pmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{pmatrix}
\Rightarrow
\begin{bmatrix} a_{11} & \cdots & a_{nm} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}
\Rightarrow
\begin{pmatrix} y_1 \\ \vdots \\ y_j \\ \vdots \\ y_m \end{pmatrix}
\Rightarrow
\begin{pmatrix} z_1 = g_1(y_1) \\ \vdots \\ z_j = g_j(y_j) \\ \vdots \\ z_m = g_m(y_m) \end{pmatrix}
$$

$X \in \mathbb{R}^n$ $\quad$ $A \in \mathbb{R}^{m \times n}$ $\quad$ $Y \in \mathbb{R}^m$ $\quad$ $Z \in \mathbb{R}^m$

Nonlinear Functions: $g_1, \dots g_j, \dots g_m$

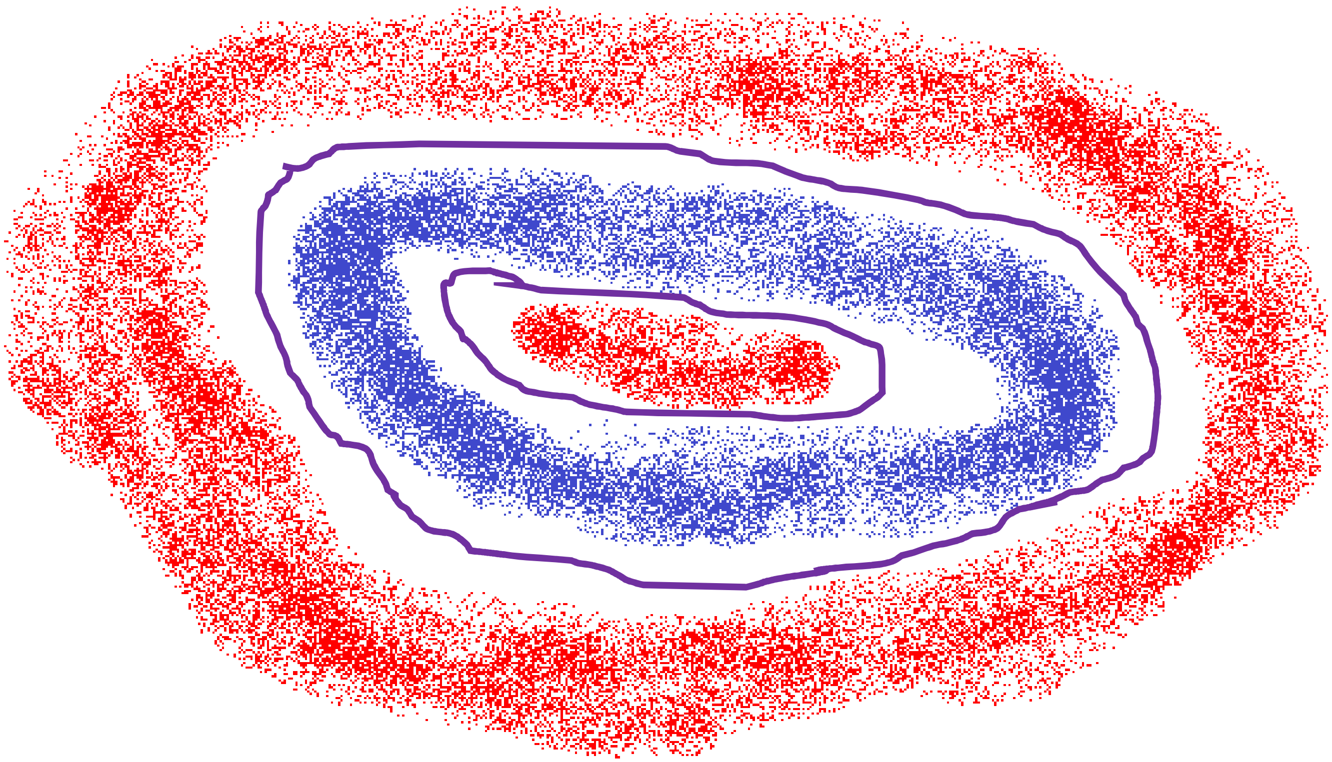
Non-Linear Transformation

The Feature Space: Linearly Separable



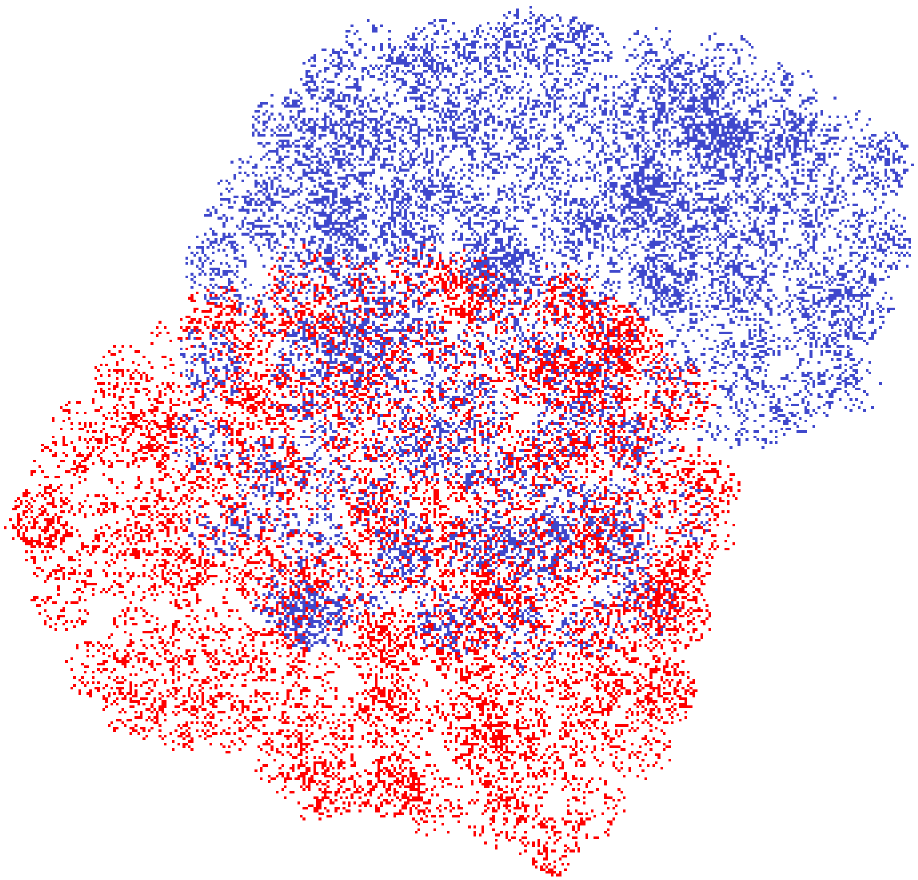
Hyperplane(s) can Separate Features Extracted from Two (Or More) Classes

The Feature Space: Not Linearly Separable



Nonlinear Hypersurfaces Separate Features Extracted from Two (Or More) Classes

The Feature Space: Nonseparable Case



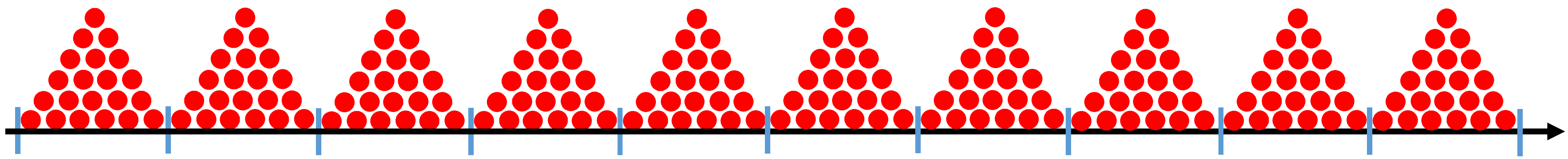
Require Further Transformations for Enhancing Separability

Cover's Theorem of Separability

Non-separable Data can be Transformed to a Higher Dimensional Space through Non-linear Transformation. The Probability of Separability Increases with the Dimension of the Destination Space.

Data is Sparse in Higher Dimensions

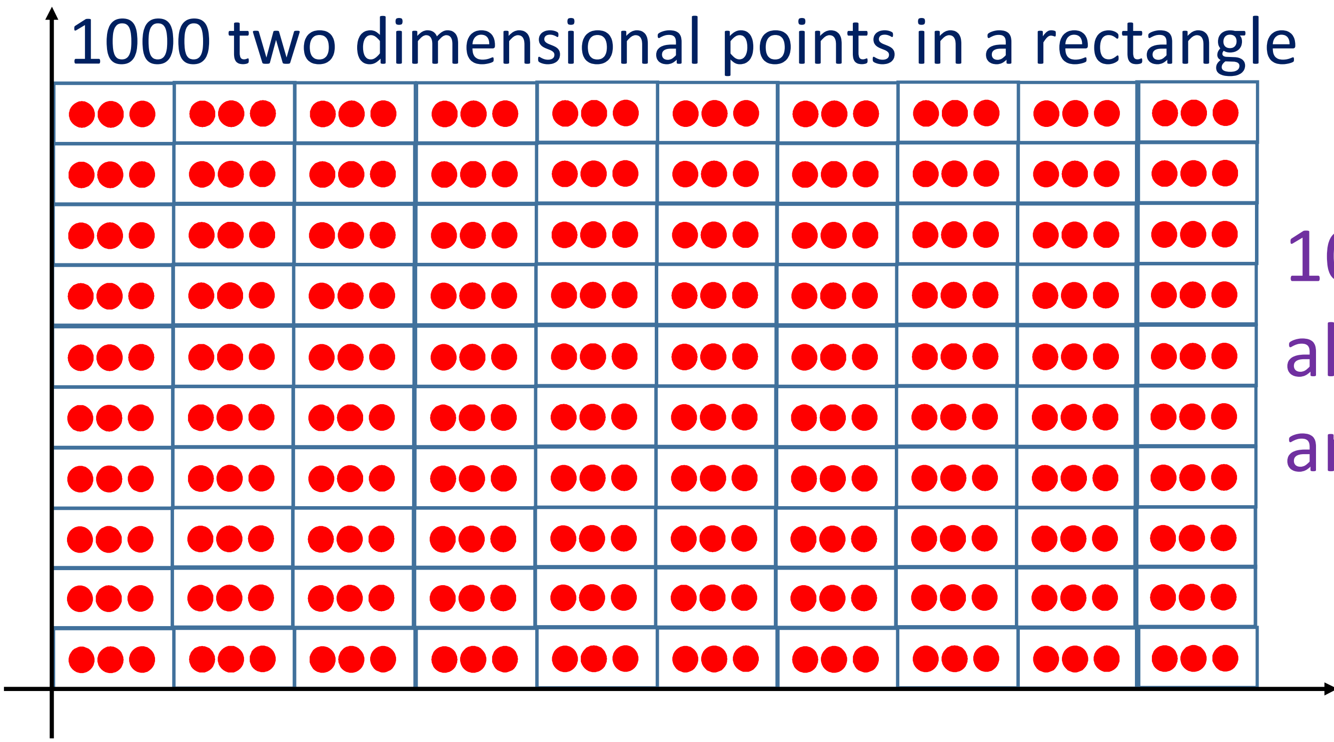
1000 one dimensional points on a line



10 Equal Sized Bins on the Line

100 / Bin: Points are Uniformly Distributed over Bins

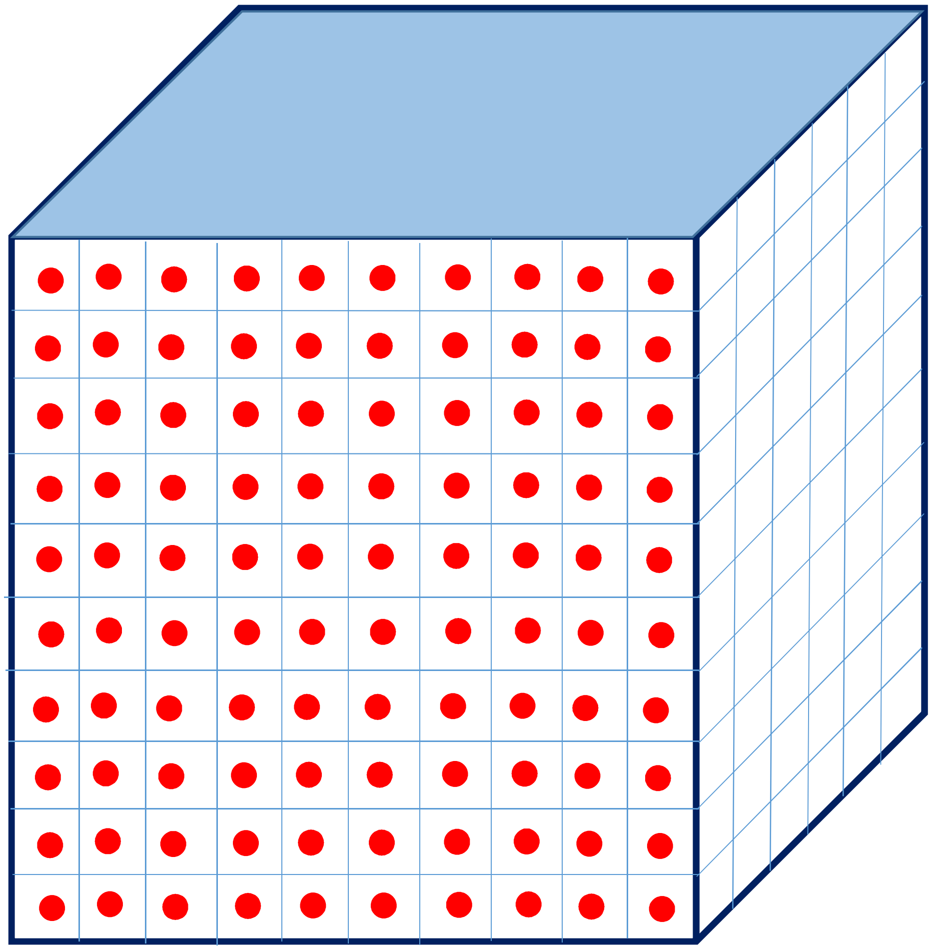
Data is Sparse in Higher Dimensions



10 Equal Sized Bins along both Horizontal and Vertical Axes

10 / Bin: Points are Uniformly Distributed over Bins

Data is Sparse in Higher Dimensions



1000 Three Dimensional points in a Cube

10 Equal Sized Bins along each Axes

1 / Bin: Points are Uniformly Distributed over Bins

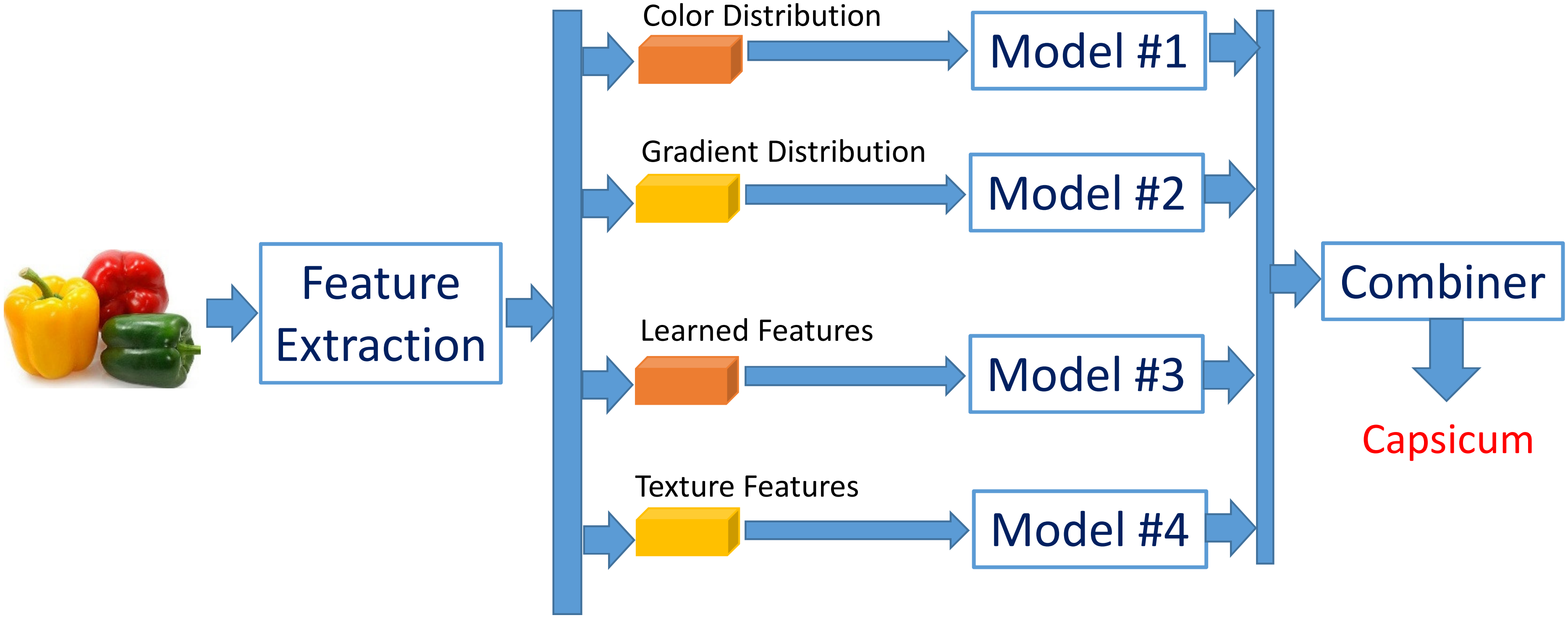
Data is Sparse in Higher Dimensions

- Consider 1000 n-dimensional (n >3) points
- We have 10 Equal Size Bins on each Axis
- There are 10^n (> 1000) bins
- Vacant bins even with 1 point/bin
- This can be generalized…

Curse of Dimensionality

- High Dimension due to Large Number of Features
- High Dimension Leads to More Storage & Computation
  - More Parameters in (Non)Linear Transformations
  - Complex Classification & Regression Models
  - More Operations (e.g. Matrix Inverse in Mahalanobis Distance)
  - Unnecessary Features often lead to Poor Performance
- Working on Feature Subspaces
  - Late Fusion in Ensemble Framework
  - Dimensionality Reduction

Late Fusion in Ensemble Framework



Dimensionality Reduction

- Feature Subset Selection
- Hashing Techniques
- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Exploratory Factor Analysis (EFA)

Feature Subset Selection

- Feature Selection vs. Extraction/Transformation(s)
- Supervised vs. Unsupervised Approaches
- Feature Subset Selection – 2^n possibilities for $x \in \mathbb{R}^n$
- Forward & Backward Feature Subset Selection
- Useful Tool for Feature Combination Analysis
- Generally used for Handcrafted Features

Feature Subset Selection: Notations

X_j: The jth Feature Subset; $j = 1, \dots m$

$\boldsymbol{S_i}$: Feature Set formed in ith Iteration

$$\boldsymbol{S_i} = \{X^{(1)} \cup X^{(2)} \dots \cup X^{(i)}\}$$

$\boldsymbol{C_i}$: Classifier Trained with Feature Set $\boldsymbol{S_i}$

$\boldsymbol{E}(\boldsymbol{S_i}; \boldsymbol{C_i})$: Evaluation Error of Classifier $\boldsymbol{C_i}$ using Feature Set $\boldsymbol{S_i}$

ϵ: A Threshold on Evaluation Error

Forward Feature Subset Selection

1. $\boldsymbol{S}_0 = \varphi; SizeOf(\cup_{i=1}^{m} X_i) = N; \boldsymbol{E}(\boldsymbol{S}_0; \boldsymbol{C}_0) = LARGE\_NUMBER$
2. **WHILE** $SizeOf(\boldsymbol{S}_{i-1}) < N$ **DO**
3. $\quad \boldsymbol{J}_i = \{k: X_k \notin \boldsymbol{S}_{i-1}\}$
4. $\quad \forall k$ *Train* $\boldsymbol{C}_{\boldsymbol{i-1}}^{(\boldsymbol{k})}$ *with* $\boldsymbol{S}_{\boldsymbol{i-1}} \cup X_{\boldsymbol{k}}$
5. $\quad j = argmin_k\, \boldsymbol{E}\left(\boldsymbol{S}_{i-1} \cup X_k; \boldsymbol{C}_{\boldsymbol{i-1}}^{(\boldsymbol{k})}\right); k \in \boldsymbol{J}_i$
6. $\quad$ **IF** $\boldsymbol{E}\left(\boldsymbol{S}_{i-1} \cup X_j; \boldsymbol{C}_{\boldsymbol{i-1}}^{(j)}\right) < \boldsymbol{E}(\boldsymbol{S}_{i-1}; \boldsymbol{C}_{i-\boldsymbol{1}}) - \epsilon$
7. $\quad$ **THEN** $\boldsymbol{S}_i = \boldsymbol{S}_{i-1} \cup X_j; \boldsymbol{C}_i = \boldsymbol{C}_{\boldsymbol{i-1}}^{(j)}$
8. $\quad$ **ELSE** break
9. **END WHILE**

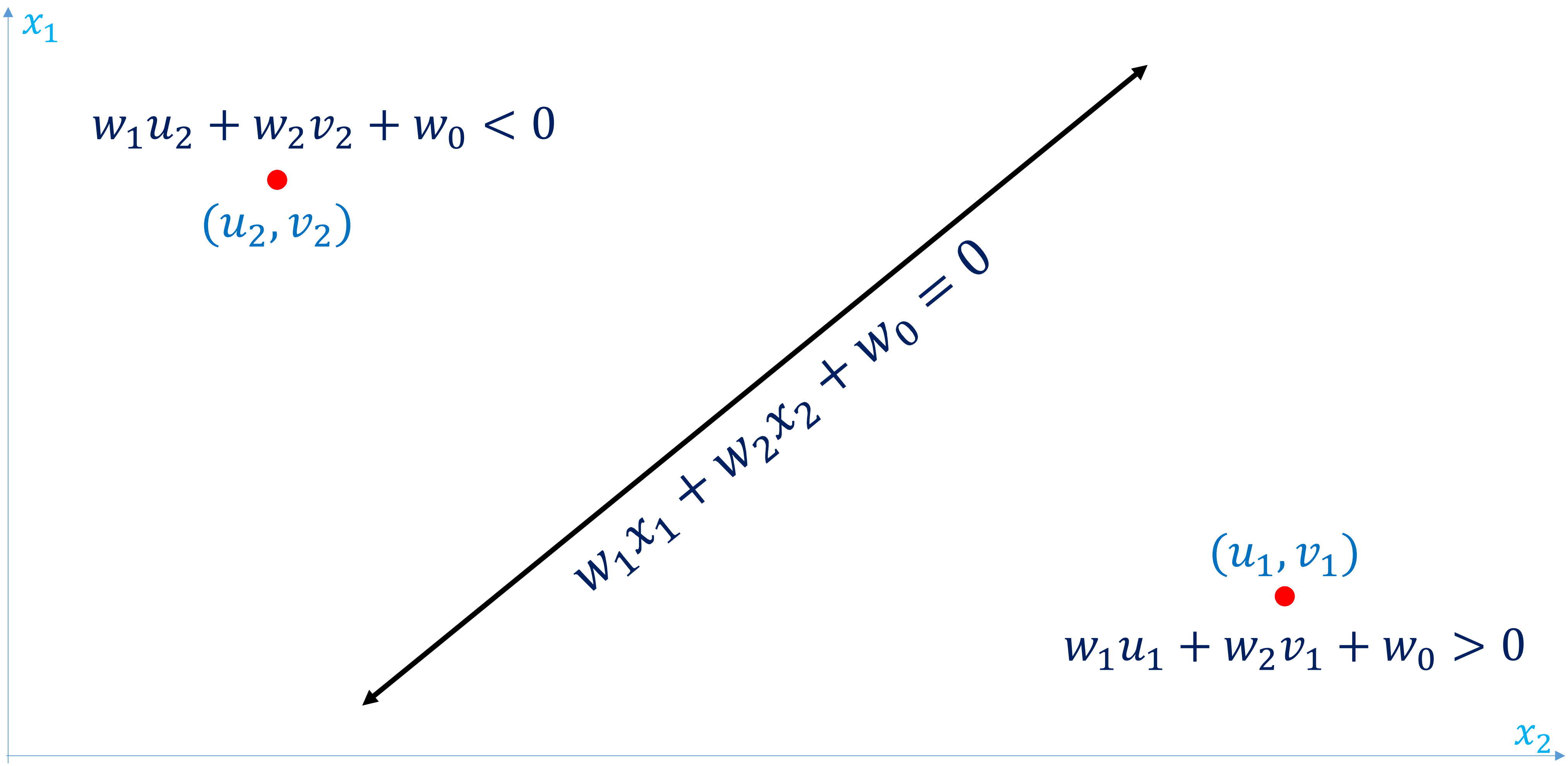
Backward Feature Subset Selection

1. $\boldsymbol{S}_0 = \cup_{i=1}^{m} X_i$
2. **WHILE** $SizeOf(\boldsymbol{S}_{i-1}) > 0$ **DO**
3. $\quad \boldsymbol{J}_i = \{k: X_k \in \boldsymbol{S}_{i-1}\}$
4. $\quad \forall k$ *Train* $\boldsymbol{C}_{\boldsymbol{i-1}}^{(\boldsymbol{k})}$ *with* $\boldsymbol{S}_{\boldsymbol{i-1}} - X_{\boldsymbol{k}}$
5. $\quad j = argmin_k\, \boldsymbol{E}\left(\boldsymbol{S}_{i-1} - X_k; \boldsymbol{C}_{\boldsymbol{i-1}}^{(\boldsymbol{k})}\right); k \in \boldsymbol{J}_i$
6. $\quad$ **IF** $\boldsymbol{E}\left(\boldsymbol{S}_{i-1} - X_j; \boldsymbol{C}_{\boldsymbol{i-1}}^{(\boldsymbol{j})}\right) < \boldsymbol{E}(S_{i-1}; \boldsymbol{C}_{i-\boldsymbol{1}}) - \epsilon$
7. $\quad$ **THEN** $\boldsymbol{S}_i = \boldsymbol{S}_{i-1} - X_j; \boldsymbol{C}_i = \boldsymbol{C}_{i-1}^{(j)}$
8. $\quad$ **ELSE** break
9. **END WHILE**

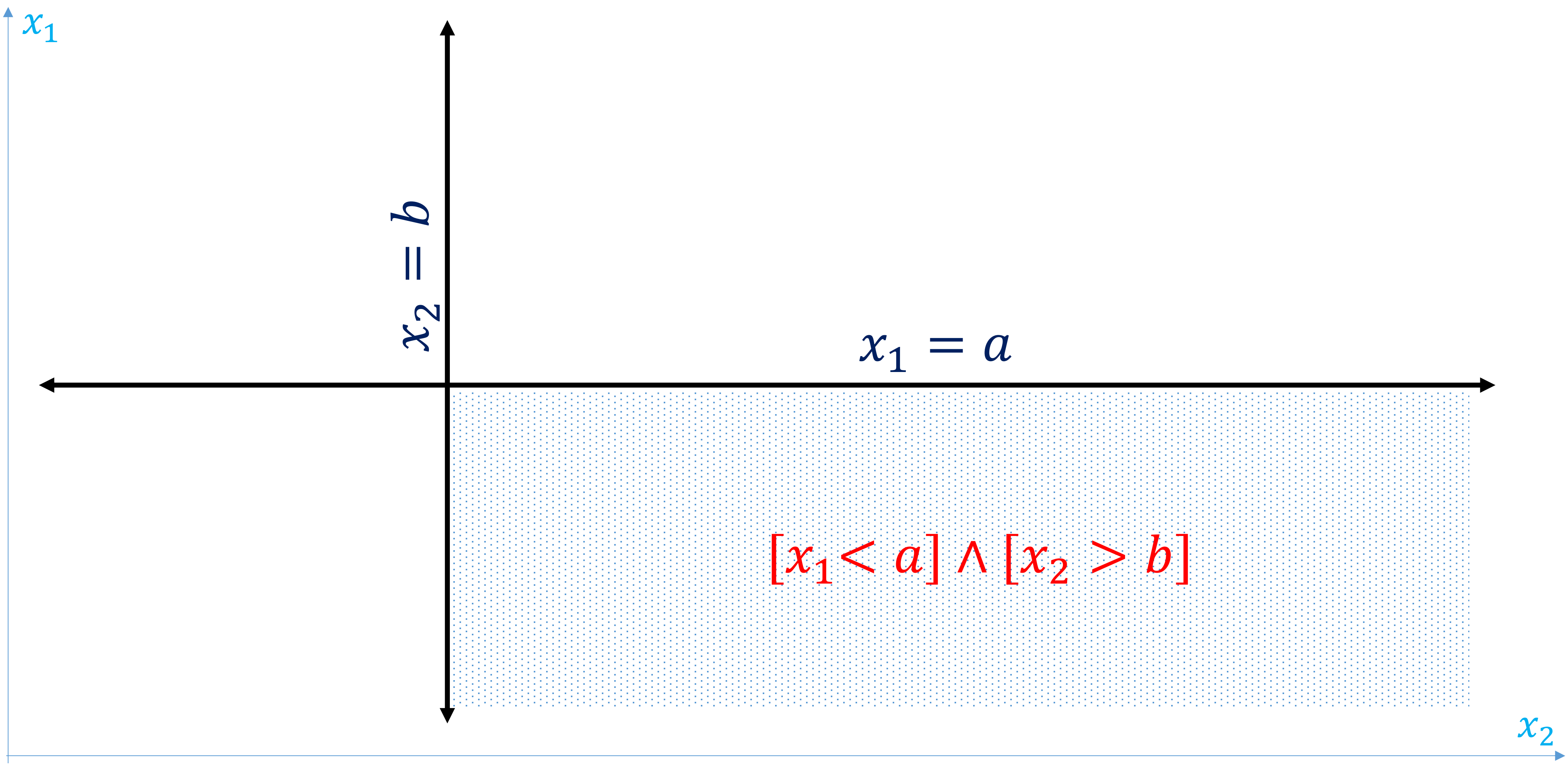
Hashing Techniques

- Dividing Feature Space into Partitions
- Assign Integer IDs (Hashes) to each Partition
- Mechanisms for Partitioning Feature Space
  - Locality Sensitive Hashing
  - Hashing with Hierarchical Structures
- Features in a Partition has the Same Hash
- Features of Similar Entities are Close in Feature Space
- Proximal Points have High Probability of Similar Hash

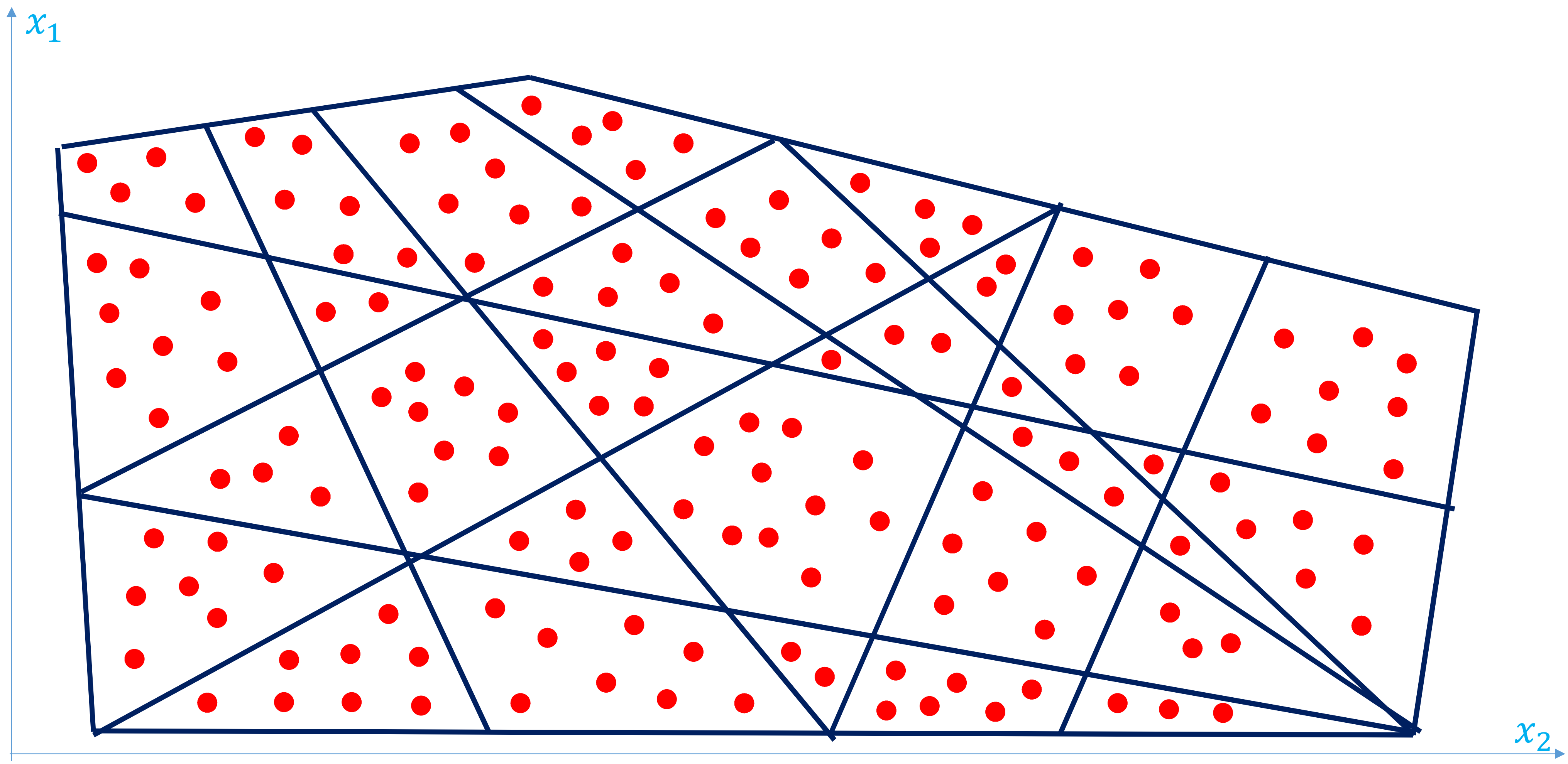
Feature Space Partitioning: Oblique



Feature Space Partitioning: Axis Aligned

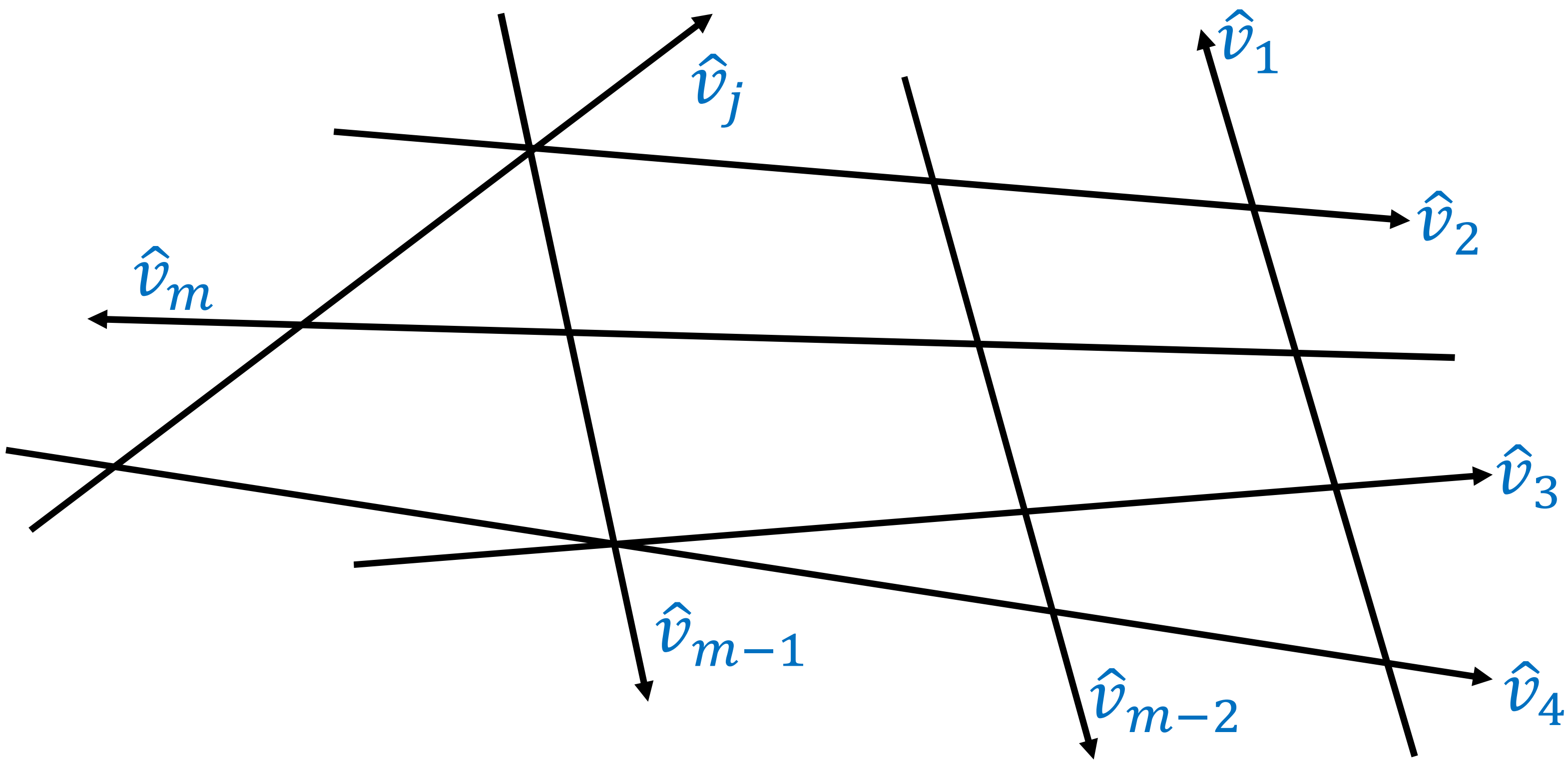


Feature Space Partitioning

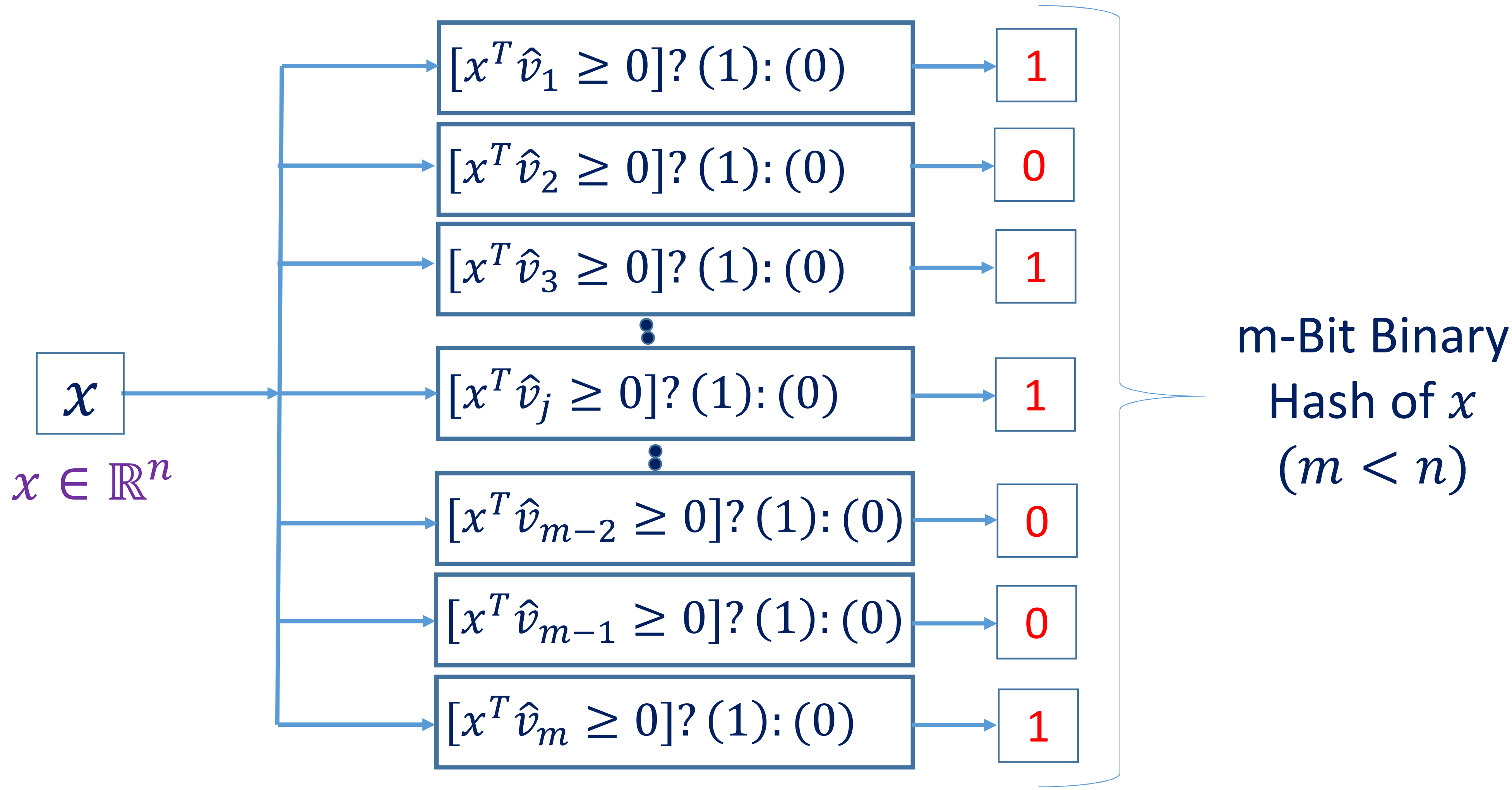


Locality Sensitive Hashing – Random Projections

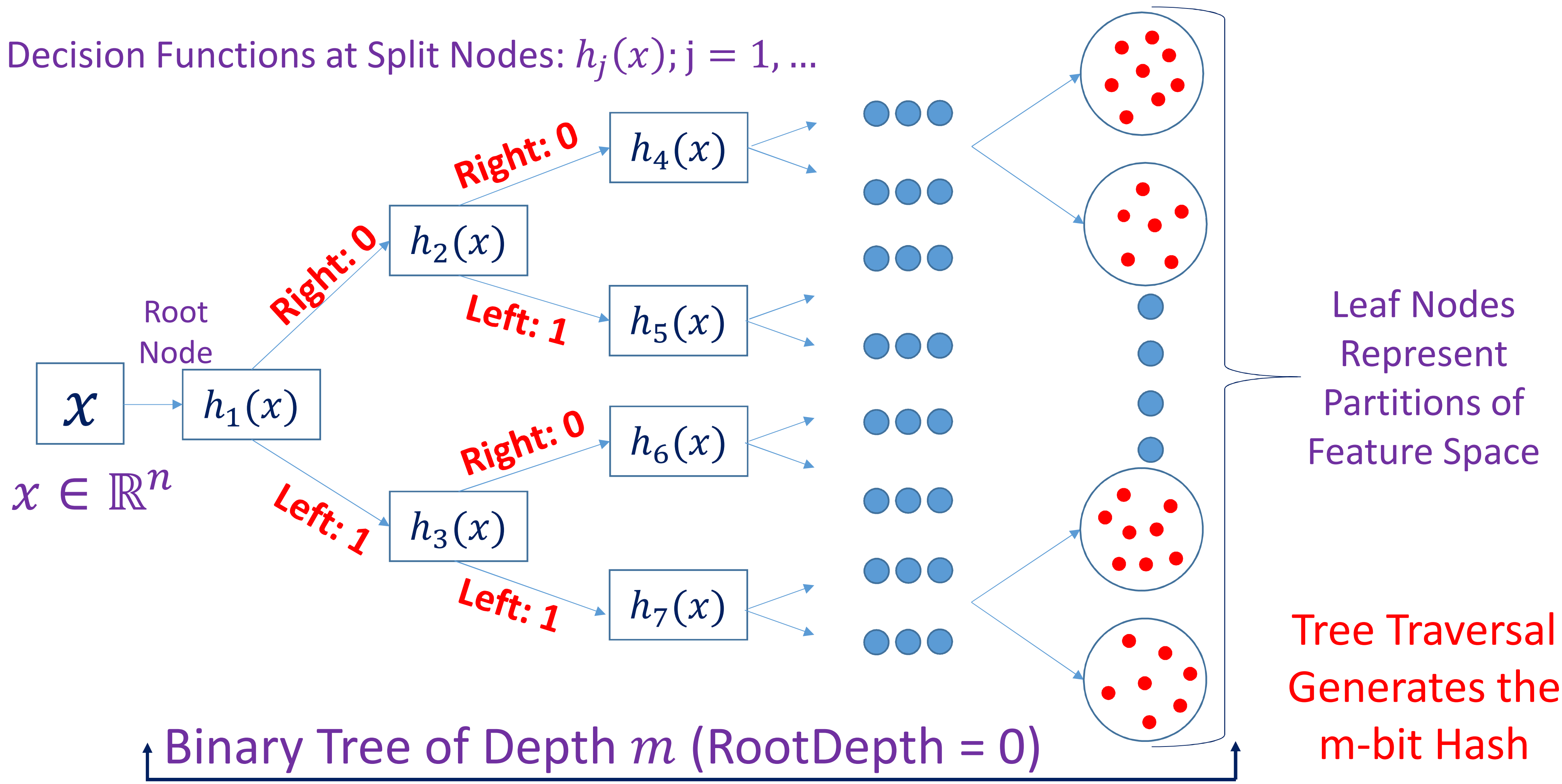
Generate Random Unit Vectors in Feature Space: $\hat{v}_1, \ldots \hat{v}_i \ldots \hat{v}_m$

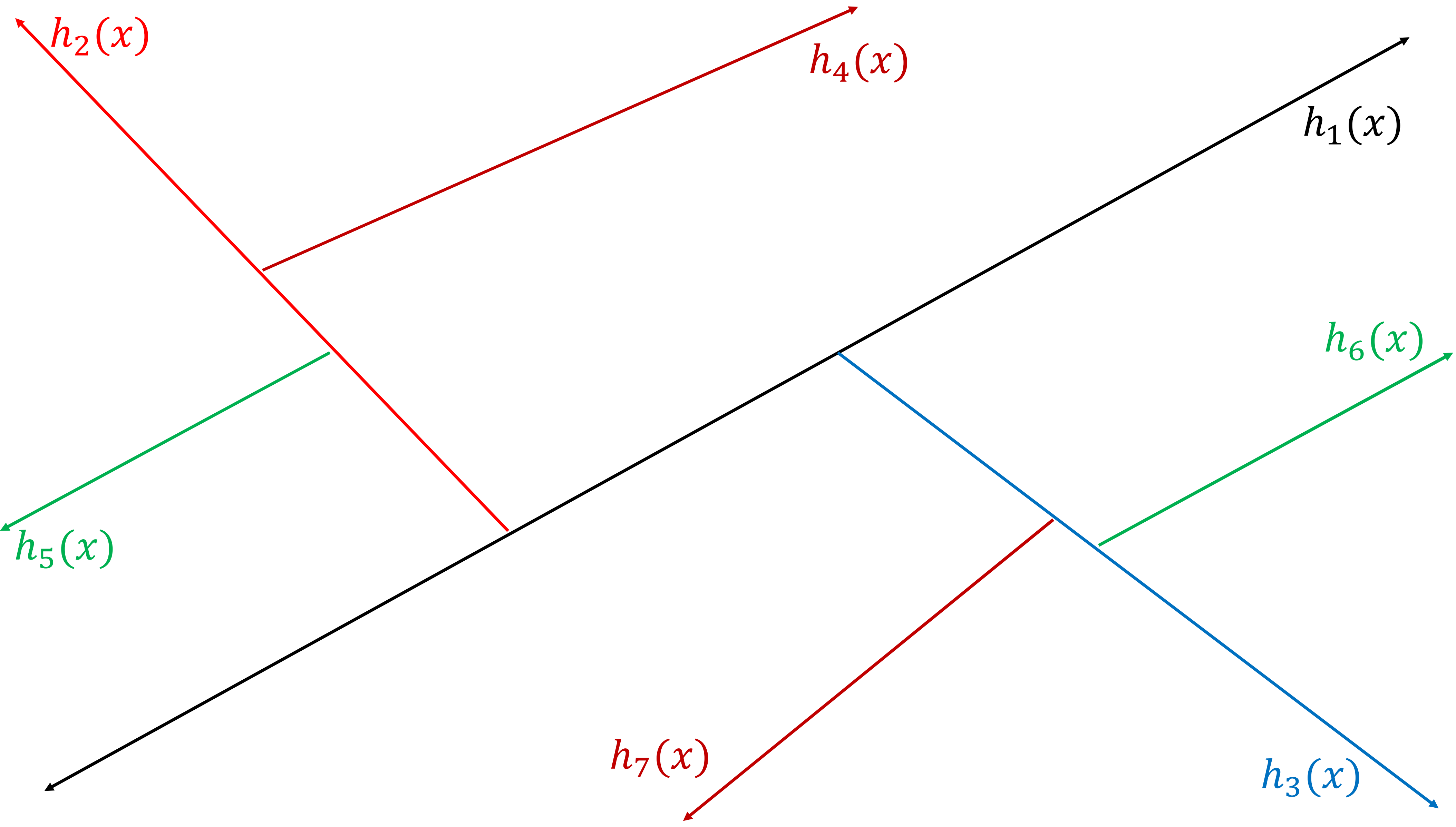


Locality Sensitive Hashing – Random Projections



Hashing with Hierarchical Structures




$h_2(x)$
$h_4(x)$
$h_1(x)$
$h_6(x)$
$h_5(x)$
$h_7(x)$
$h_3(x)$

Applications of Hashing Techniques

- Near Duplicate Detection (in Archives)
- Hierarchical Grouping
- Search in Image Databases
- Audio Similarity Identification
- Digital Audio-Video Fingerprinting

Summary

- Significance of Feature Engineering
- Feature Extraction & Transformations
- Separability in Feature Space
- Issues with Dimensionality
- Dimensionality Reduction Techniques
- Forward & Backward Feature Selection
- Binary Hashing Techniques

Thank You