Fundamentals of Image Analysis

Alexandre Xavier Falcão

Institute of Computing - UNICAMP

afalcao@ic.unicamp.br
Image analysis requires to learn models for object detection and/or delineation (sample extraction), feature extraction (descriptor learning), and classification (classifier learning).
Image analysis requires to learn models for object detection and/or delineation (sample extraction), feature extraction (descriptor learning), and classification (classifier learning).
Image analysis requires to learn models for object detection and/or delineation (sample extraction), feature extraction (descriptor learning), and classification (classifier learning).
We will start by taking object detection as an example that uses pixels as samples and requires feature extraction and classification.

Object detection evaluates candidate locations for the object(s) of interest in the image.
Agenda

- Object detection and main concepts for image analysis.
- Machine learning and basic concepts from Statistics.
- Classic pattern recognition techniques.
The important concepts for object detection are

- Multiband image, adjacency relation, and multiband kernel.
- Correlation and convolution between image and kernels (i.e., multiband image filtering).
- Fast filtering through integral images for feature extraction.
- Feature selection using systems of weak classifiers.
A multiband image $\hat{I}$ is a pair $(D_I, I)$, in which

- $D_I \in \mathbb{Z}^d$ is the image domain and

- $I$ assigns to each space element (spel) $p \in D_I$ a set (feature vector) of scalar values $I(p) = (I_1(p), I_2(p), \ldots, I_n(p))$ — i.e., a point in the feature space $\mathbb{R}^n$. 

We will focus on $d = 2$ (spel is pixel) and $n \geq 1$. We will use $I = I$ for binary and grayscale images ($n = 1$).
A multiband image $\hat{I}$ is a pair $(D_I, I)$, in which

- $D_I \in \mathbb{Z}^d$ is the image domain and
- $I$ assigns to each space element (spel) $p \in D_I$ a set (feature vector) of scalar values $I(p) = (I_1(p), I_2(p), \ldots, I_n(p))$ — i.e., a point in the feature space $\mathbb{R}^n$.

- We will focus on $d = 2$ (spel is pixel) and $n \geq 1$. 

A multiband image $\hat{I}$ is a pair $(D_I, I)$, in which

- $D_I \in \mathbb{Z}^d$ is the image domain and
- $I$ assigns to each space element (spel) $p \in D_I$ a set (feature vector) of scalar values $I(p) = (I_1(p), I_2(p), \ldots, I_n(p))$ — i.e., a point in the feature space $\mathbb{R}^n$.

- We will focus on $d = 2$ (spel is pixel) and $n \geq 1$.
- We will use $I = I$ for binary and grayscale images ($n = 1$).
A pixel $p$ in the image domain $D_I \subset \mathbb{Z}^2$, is represented by a 2D integer coordinate $p = (x_p, y_p)$.

Pixels with similar features should be mapped onto nearby positions in the feature space $\mathbb{R}^n$.

The feature space can be changed by image filtering.
An adjacency relation $A \subseteq D_I \times D_I$ may be defined in the image domain and/or feature space as a binary relation between pixels.

- $A_1: \{(p, q) \in D_I \times D_I \mid \|q - p\| \leq \alpha_1\}$,
- $A_2: \{(p, q) \in D_I \times D_I \mid \|I(q) - I(p)\| \leq \alpha_2\}$,
- $A_3: \{(p, q) \in D_I \times D_I \mid \|q - p\| \leq \alpha_1 \text{ and } \|I(q) - I(p)\| \leq \alpha_2\}$,

$\alpha_1, \alpha_2 \in \mathbb{R}^+$
An adjacency relation \( A \subseteq D_I \times D_I \) may be defined in the image domain and/or feature space as a binary relation between pixels.

\[
A_1: \{ (p, q) \in D_I \times D_I \mid \| q - p \| \leq \alpha_1 \},
\]

\[
A_2: \{ (p, q) \in D_I \times D_I \mid \| I(q) - I(p) \| \leq \alpha_2 \},
\]

\[
A_3: \{ (p, q) \in D_I \times D_I \mid \| q - p \| \leq \alpha_1 \text{ and } \| I(q) - I(p) \| \leq \alpha_2 \},
\]

\( \alpha_1, \alpha_2 \in \mathbb{R}^+ \) and \( A(p) \) is said the set of pixels \( q \) adjacent to a pixel \( p \).
Adjacency relation

An adjacency relation $\mathcal{A} \subseteq D_I \times D_I$ may be defined in the image domain and/or feature space as a binary relation between pixels.

- $\mathcal{A}_1: \{(p, q) \in D_I \times D_I | \|q - p\| \leq \alpha_1\},$
- $\mathcal{A}_2: \{(p, q) \in D_I \times D_I | \|\mathbf{l}(q) - \mathbf{l}(p)\| \leq \alpha_2\},$
- $\mathcal{A}_3: \{(p, q) \in D_I \times D_I | \|q - p\| \leq \alpha_1 \text{ and } \|\mathbf{l}(q) - \mathbf{l}(p)\| \leq \alpha_2\},$

$\alpha_1, \alpha_2 \in \mathbb{R}^+$ and $\mathcal{A}(p)$ is said the set of pixels $q$ adjacent to a pixel $p$.

For the image on the right, what is the adjacency set of $p = (2, 3)$ for $\mathcal{A}_1, \mathcal{A}_2,$ and $\mathcal{A}_3,$ when $\alpha_1 = \sqrt{5}$ and $\alpha_2 = 0$?
A multiband kernel is a pair \((\mathcal{A}, \mathcal{W})\) in which

- \(\mathcal{A}\) defines the adjacency set for any pixel \(p \in D_I\) as

\[
\mathcal{A}(p) : \{q \in \mathbb{Z}^2 \mid q - p \in \{(dx_1, dy_1), (dx_2, dy_2), \ldots, (dx_K, dy_K)\}\},
\]

\(K = |\mathcal{A}(p)|\), and

- \(\mathcal{W} = [w_1^t w_2^t \ldots w_K^t]\) is a row matrix, whose elements \(w_k^t\) can be expanded into extra columns \([w_{k1} w_{k2} \ldots w_{kn}]\), such that \(w_{ki} \in \mathbb{R}, i \in [1, n]\), is the weight assigned to a dimension of the feature space at location

\(q_k = (x_{q_k}, y_{q_k}) = (x_p, y_p) + (dx_k, dy_k), k \in [1, K].\)
Multiband kernel

Consider, for instance, a Sobel kernel \((A, W)\) to enhance vertical edges of a color image \(\hat{I} = (D_I, I)\) — i.e., \(n = 3\).

- \(A\) defines the adjacency set \(A(p)\) with pixels \(q \in \mathbb{Z}^2\), such that \(q - p \in \{(-1, -1), (0, -1), (1, -1), (-1, 0), (0, 0), (1, 0), (-1, 1), (0, 1), (1, 1)\}\), for any \(p \in D_I\), and

- \(W = [\mathbf{w}_1^t \mathbf{w}_2^t \ldots \mathbf{w}_9^t]\) is such that \(\mathbf{w}_1^t = [-1, -1, -1], \mathbf{w}_2^t = [0, 0, 0], \mathbf{w}_3^t = [1, 1, 1], \mathbf{w}_4^t = [-2, -2, -2], \mathbf{w}_5^t = [0, 0, 0], \mathbf{w}_6^t = [2, 2, 2], \mathbf{w}_7^t = [-1, -1, -1], \mathbf{w}_8^t = [0, 0, 0], \text{ and } \mathbf{w}_9^t = [1, 1, 1].\)
The internal products between \( w_k \) and \( I(q_{kj}) \), \( k \in [1, K] \) and \( j \in [1, m] \), \( m = |D_I| \), define the correlation between image and kernel. It creates the image \( \hat{J} = (D_I, J) \),

\[
J(p_j) = \sum_{k=1}^{K} w_k \cdot I(q_{kj}),
\]

\[
w_k \cdot I(q_{kj}) = \sum_{i=1}^{n} w_{ki}I_i(q_{kj}),
\]

for \( p_j \in D_I \) and \( q_{kj} \in D_I \).

This operation can be more efficiently implemented as a matrix multiplication.
Multiband image and adjacency values in a matrix

The values in the feature vector \( \mathbf{l}(q_{kj}), \ k \in [1, K], \) of each adjacent pixel \( q_{kj} \in \mathcal{A}(p_j) \) of each pixel \( p_j \in D_I, \ j \in [1, m], \ m = |D_I|, \) must be organized in a matrix \( X_{\hat{i}} \) as follows.

\[
X_{\hat{i}} = \begin{bmatrix}
\mathbf{l}(q_{11}) & \mathbf{l}(q_{12}) & \ldots & \mathbf{l}(q_{1m}) \\
\mathbf{l}(q_{21}) & \mathbf{l}(q_{22}) & \ldots & \mathbf{l}(q_{2m}) \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{l}(q_{K1}) & \mathbf{l}(q_{K2}) & \ldots & \mathbf{l}(q_{Km})
\end{bmatrix},
\]

such that \( \mathbf{l}(q_{kj}) \) can be expanded as a column matrix
\( [l_1(q_{kj}) l_2(q_{kj}) \ldots l_n(q_{kj})]^t. \)
Correlation between kernel and image

The correlation between the image \( \hat{I} = (D_I, I) \) and a kernel \( (A, W) \) is then the product

\[
X_{\hat{J}} = WX_{\hat{i}},
\]

such that \( X_{\hat{J}} \) is a row matrix whose elements \( J(p_j) \), \( j \in [1, |D_J|] \), are the resulting pixel values of the grayscale image \( \hat{J} = (D_J, J) \).
The correlation between the image $\hat{I} = (D_I, I)$ and a kernel $(\mathcal{A}, W)$ is then the product

$$X_{\hat{J}} = WX_{\hat{i}},$$

such that $X_{\hat{J}}$ is a row matrix whose elements $J(p_j)$, $j \in [1, |D_J|]$, are the resulting pixel values of the grayscale image $\hat{J} = (D_J, J)$.

By considering pixels in $D_I$ only, the correlation produces $D_J = D_I$ and adjacent pixels $q_{kj} \not\in D_I$ must not be considered in the above matrix multiplication. Indeed, for feature extraction, we often dismiss pixels with adjacent $q_{kj} \not\in D_I$, by making $J(p_j) \leftarrow 0$.  

Correlation between kernel and image
The correlation between the image $\hat{I} = (D_I, I)$ and a kernel $(A, W)$ is then the product

$$X_{\hat{j}} = WX_{\hat{i}},$$

such that $X_{\hat{j}}$ is a row matrix whose elements $J(p_j)$, $j \in [1, |D_J|]$, are the resulting pixel values of the grayscale image $\hat{J} = (D_J, J)$.

By considering pixels in $D_I$ only, the correlation produces $D_J = D_I$ and adjacent pixels $q_{kj} \not\in D_I$ must not be considered in the above matrix multiplication. Indeed, for feature extraction, we often dismiss pixels with adjacent $q_{kj} \not\in D_I$, by making $J(p_j) \leftarrow 0$.

Image filtering, however, is related to the concept of convolution.
Convolution between kernel and image

- The only difference between correlation and convolution is the reflection of the displacements in $\mathcal{A}$. Convolution uses a kernel $(\mathcal{A}', W)$ such that each displacement $(dx_k, dy_k) \in \mathcal{A}$ is mapped into a displacement $(-dx_k, -dy_k) \in \mathcal{A}'$, $k \in [1, K]$. For the Sobel kernel, $\mathcal{A}'$ would define an adjacency set $\mathcal{A}'(p)$ with $q - p \in \{(1, 1), (0, 1), (-1, 1), (1, 0), (0, 0), (-1, 0), (1, -1), (0, -1), (-1, -1)\}$ and $W = [w_t^1 \quad w_t^2 \quad \ldots \quad w_t^9]$, $w_t^1 = [-1, -1, -1]$, $w_t^2 = [0, 0, 0]$, $w_t^3 = [1, 1, 1]$, $w_t^4 = [-2, -2, -2]$, $w_t^5 = [0, 0, 0]$, $w_t^6 = [2, 2, 2]$, $w_t^7 = [-1, -1, -1]$, $w_t^8 = [0, 0, 0]$, and $w_t^9 = [1, 1, 1]$. However, it is common to define the kernel already reflected, which was the case of $(\mathcal{A}, W)$ for Sobel vertical edge enhancement.
The only difference between correlation and convolution is the reflection of the displacements in $\mathcal{A}$. Convolution uses a kernel $(\mathcal{A}', W)$ such that each displacement $(dx_k, dy_k) \in \mathcal{A}$ is mapped into a displacement $(-dx_k, -dy_k) \in \mathcal{A}'$, $k \in [1, K]$.

For the Sobel kernel, $\mathcal{A}'$ would define an adjacency set $\mathcal{A}'(p)$ with $q - p \in \{(1, 1), (0, 1), (-1, 1), (1, 0), (0, 0), (-1, 0), (1, -1), (0, -1), (-1, -1)\}$ and $W = [w^t_1 \ w^t_2 \ldots \ w^t_9]$, $w^t_1 = [-1, -1, -1]$, $w^t_2 = [0, 0, 0]$, $w^t_3 = [1, 1, 1]$, $w^t_4 = [-2, -2, -2]$, $w^t_5 = [0, 0, 0]$, $w^t_6 = [2, 2, 2]$, $w^t_7 = [-1, -1, -1]$, $w^t_8 = [0, 0, 0]$, and $w^t_9 = [1, 1, 1]$. 
The only difference between correlation and convolution is the reflection of the displacements in $\mathcal{A}$. Convolution uses a kernel $(\mathcal{A}', W)$ such that each displacement $(dx_k, dy_k) \in \mathcal{A}$ is mapped into a displacement $(-dx_k, -dy_k) \in \mathcal{A}'$, $k \in [1, K]$.

For the Sobel kernel, $\mathcal{A}'$ would define an adjacency set $\mathcal{A}'(p)$ with $q - p \in \{(1, 1), (0, 1), (-1, 1), (1, 0), (0, 0), (-1, 0), (1, -1), (0, -1), (-1, -1)\}$ and $W = [w_1^t \; w_2^t \; \ldots \; w_9^t]$, where $w_1^t = [-1, -1, -1]$, $w_2^t = [0, 0, 0]$, $w_3^t = [1, 1, 1]$, $w_4^t = [-2, -2, -2]$, $w_5^t = [0, 0, 0]$, $w_6^t = [2, 2, 2]$, $w_7^t = [-1, -1, -1]$, $w_8^t = [0, 0, 0]$, and $w_9^t = [1, 1, 1]$.

However, it is common to define the kernel already reflected, which was the case of $(\mathcal{A}, W)$ for Sobel vertical edge enhancement.
The Sobel-vertical-edge kernel can enhance the characters of a car plate and the integral image can be exploited to assign higher scores to the best candidate locations.
The pixel values of the integral image \( \hat{I}_{\text{int}} = (D_l, l_{\text{int}}) \) of an image \( \hat{I} = (D_l, I) \) are defined by

\[
l_{\text{int}}(p) = \sum_{\forall q \in A(p)} I(q)
\]

\[
A(p) : \{ q \in D_l \mid (x_q \leq x_p) \text{ and } (y_q \leq y_p) \}
\]
The integral value within any rectangular region $\mathcal{R}$, delimited by pixels $p_1$, $p_2$, $p_3$ and $p_4$, is

$$
\sum_{\forall p \in \mathcal{R}} l(p) = l_{\text{int}}(p_4) - l_{\text{int}}(p_2) - l_{\text{int}}(p_3) + l_{\text{int}}(p_1).
$$

This corresponds to the convolution between the image and an unitary kernel with adjacency defined by $\mathcal{R}$ with respect to some origin $p$. 
By defining $\mathcal{R}$ around each pixel $p \in D_I$, the integral of the edge-enhanced image can be used to define candidates for the plate location by **thresholding** (i.e., a weak classifier) and connected component analysis.
One may define kernels of different sizes and configurations based on integral images (haar-like features)— the weights are $w \geq 1$ in the white region(s) and $-w$ in the black region(s) — or the other way around.
Image filtering for feature extraction

The convolution between an image and a bank of kernels generates a multiband image \( \hat{J} = (D_J, J) \) for feature selection and classification. Viola & Jones introduced a fast scheme based on cascade of weak classifiers for face detection [5].
The convolution between an image and a bank of kernels generates a multiband image \( \hat{J} = (D_j, J) \) for feature selection and classification. Viola & Jones introduced a fast scheme based on cascade of weak classifiers for face detection \([5]\).
A kernel bank in a matrix

- When $\mathcal{A}$ is fixed for all kernels of a bank, their weights can be stored as extra rows of the matrix $W$.

- In such a case, the number of rows in $W$ is the number of kernels and so the number of bands in the resulting filtered image $\hat{J} = (D_J, J)$, for

$$X_{\hat{j}} = WX_{\hat{j}}.$$ 

Fast image filtering can then be computed by parallel algorithms for matrix multiplication.
By providing a training set with images and the corresponding annotations of the object location in each image,

- one can find the threshold that minimizes the classification error in the training set for each feature (and even at each pixel).

- Object detection on an unseen image set, named test set, can be based on the weighted combination of the decision from all classifiers.

Some post-processing, such as the analysis of the resulting components, is likely to be required.
One may rely on the knowledge about a given application to define feature extraction.
One may rely on the knowledge about a given application to define feature extraction.

However, methods based on data processing and analysis, named data-driven approaches, are more popular nowadays.
One may rely on the knowledge about a given application to define feature extraction.

However, methods based on data processing and analysis, named data-driven approaches, are more popular nowadays.

Data-driven approaches for descriptor and classifier learning may be divided into

- supervised (discriminative, wrapper),
- unsupervised (generative, filters), or
- semi-supervised (transductive).
One may rely on the knowledge about a given application to define feature extraction.

However, methods based on data processing and analysis, named data-driven approaches, are more popular nowadays.

Data-driven approaches for descriptor and classifier learning may be divided into
- supervised (discriminative, wrapper),
- unsupervised (generative, filters), or
- semi-supervised (transductive).

Supervised deep neural networks, for instance, project the descriptor and classifier at the same time [1].
Supervised and unsupervised learning

- A sample (it has a different meaning in Statistics) may be an image, pixel, superpixel, object, or subimage around the object.
Supervised and unsupervised learning

- **A sample** (it has a different meaning in Statistics) may be an image, pixel, superpixel, object, or subimage around the object.

- In **supervised learning**, we are interested in the case a classifier assigns samples to one out of \( c \) possible categories \( \{\omega_k\}_{k=1}^c \).

- In **unsupervised learning**, samples are grouped into one out of \( g \) clusters \( \{G_k\}_{k=1}^g \) (clustering) based on their proximity in the feature space. A "good" descriptor should map samples from a same category into the same group and separate the groups as much as possible in the feature space, despite the absence of category (label) information.
A sample (it has a different meaning in Statistics) may be an image, pixel, superpixel, object, or subimage around the object.

In supervised learning, we are interested in the case a classifier assigns samples to one out of \( c \) possible categories \( \{ \omega_k \}_{k=1}^c \).

In unsupervised learning, samples are grouped into one out of \( g \) clusters \( \{ G_k \}_{k=1}^g \) (clustering) based on their proximity in the feature space.
Supervised and unsupervised learning

- A **sample** (it has a different meaning in Statistics) may be an image, pixel, superpixel, object, or subimage around the object.

- In **supervised learning**, we are interested in the case a classifier assigns samples to one out of \( c \) possible categories \( \{\omega_k\}_{k=1}^c \).

- In **unsupervised learning**, samples are grouped into one out of \( g \) clusters \( \{G_k\}_{k=1}^g \) (clustering) based on their proximity in the feature space.

- A “good” descriptor should map samples from a same category into the same group and separate the groups as much as possible in the feature space, despite the absence of category (label) information.
In unsupervised learning, one can

- extract features from training samples,
- group samples into $g$ clusters, $\{G_k\}_{k=1}^{g}$, and
- apply some clustering validity measure to evaluate each candidate solution and find the best one among them.
Supervised learning

In supervised learning, one can

- take into account the labels of the samples to improve feature extraction,

- classify samples into $c$ categories, $\{\omega_k\}_{k=1}^c$, and

- apply an effectiveness measure to evaluate the candidate solution, improve the process, and find the best one among all candidates.
Training samples and feature matrix

For a given training set \( \mathcal{Z}_{tr} = \{s_j\}^m_{j=1} \), a descriptor \( D \) is a mapping \( \mathcal{Z}_{tr} \rightarrow X_{tr} \) that creates a \( n \times m \) feature matrix \( X_{tr} \).

\[
X_{tr} = \begin{bmatrix}
x_{11} & x_{12} & \ldots & x_{1m} \\
x_{21} & x_{22} & \ldots & x_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
x_{n1} & x_{n2} & \ldots & x_{nm}
\end{bmatrix}
\]

where the columns \([x_{1j}, x_{2j}, \ldots, x_{nj}]^t\) are the feature vectors \( x_j = x(s_j) = (x_1(s_j), x_2(s_j), \ldots, x_n(s_j)) \) of the samples \( s_j \in \mathcal{Z}_{tr} \).
For a given training set $\mathcal{Z}_{tr} = \{s_j\}_{j=1}^m$, a descriptor $D$ is a mapping $\mathcal{Z}_{tr} \rightarrow X_{tr}$ that creates a $n \times m$ feature matrix $X_{tr}$.

$$X_{tr} = \begin{bmatrix}
x_{11} & x_{12} & \ldots & x_{1m} \\
x_{21} & x_{22} & \ldots & x_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
x_{n1} & x_{n2} & \ldots & x_{nm}
\end{bmatrix}$$

where the columns $[x_{1j}, x_{2j}, \ldots, x_{nj}]^t$ are the feature vectors $x_j = x(s_j) = (x_1(s_j), x_2(s_j), \ldots, x_n(s_j))$ of the samples $s_j \in \mathcal{Z}_{tr}$.

Once $D$ is defined, the pair $(\mathcal{Z}_{tr}, X_{tr})$ is called the training dataset $D_{tr}$. Similarly, one can use $D: \mathcal{Z}_{ts} \rightarrow X_{ts}$ to obtain a test dataset $D_{ts} = (\mathcal{Z}_{ts}, X_{ts})$ from a test set $\mathcal{Z}_{ts}$. 
Good practice in machine learning should

- evaluate a method multiple times for **statistically independent** training and test sets,

- use the same sets for each method, and

- compare methods using a **statistical test** suitable for the experiment.
Good practice in machine learning

Good practice in machine learning should

- evaluate a method multiple times for statistically independent training and test sets,

- use the same sets for each method, and

- compare methods using a statistical test suitable for the experiment.

- The methods may be distinct classifiers based on a same descriptor, distinct descriptors using a same classification model, image segmentation algorithms, etc.
Good practice in machine learning should

- evaluate a method multiple times for statistically independent training and test sets,

- use the same sets for each method, and

- compare methods using a statistical test suitable for the experiment.

The methods may be distinct classifiers based on a same descriptor, distinct descriptors using a same classification model, image segmentation algorithms, etc.

Unfortunately, the number of labeled samples are not always enough to draw reliable conclusions.
This process must be repeated multiple times for the statistical analysis of a method or the statistical comparison among methods.
Samples are often randomly selected from a larger set $\mathcal{Z}$ to compose the training set $\mathcal{Z}_{tr}$ and the test set $\mathcal{Z}_{ts}$, such that $\mathcal{Z}_{tr} \cap \mathcal{Z}_{ts} = \emptyset$ and $\mathcal{Z}_{tr} \cup \mathcal{Z}_{ts} = \mathcal{Z}$.
Sample selection

- Samples are often **randomly selected** from a larger set $\mathcal{Z}$ to compose the training set $\mathcal{Z}_{tr}$ and the test set $\mathcal{Z}_{ts}$, such that $\mathcal{Z}_{tr} \cap \mathcal{Z}_{ts} = \emptyset$ and $\mathcal{Z}_{tr} \cup \mathcal{Z}_{ts} = \mathcal{Z}$.

- When their true labels in $\mathcal{Z}$ are known *a priori*, the selection of a same number of samples per category creates **balanced** sets.
Sample selection

- Samples are often randomly selected from a larger set $\mathcal{Z}$ to compose the training set $\mathcal{Z}_{tr}$ and the test set $\mathcal{Z}_{ts}$, such that $\mathcal{Z}_{tr} \cap \mathcal{Z}_{ts} = \emptyset$ and $\mathcal{Z}_{tr} \cup \mathcal{Z}_{ts} = \mathcal{Z}$.

- When their true labels in $\mathcal{Z}$ are known \textit{a priori}, the selection of a same number of samples per category creates \textit{balanced} sets.

- Alternatively, a same percentage of samples per category creates \textit{imbalanced} sets whenever $\mathcal{Z}$ is imbalanced.
Sample selection

- Samples are often randomly selected from a larger set $\mathcal{Z}$ to compose the training set $\mathcal{Z}_{tr}$ and the test set $\mathcal{Z}_{ts}$, such that $\mathcal{Z}_{tr} \cap \mathcal{Z}_{ts} = \emptyset$ and $\mathcal{Z}_{tr} \cup \mathcal{Z}_{ts} = \mathcal{Z}$.

- When their true labels in $\mathcal{Z}$ are known \textit{a priori}, the selection of a same number of samples per category creates \textit{balanced} sets.

- Alternatively, a same percentage of samples per category creates \textit{imbalanced} sets whenever $\mathcal{Z}$ is imbalanced.

- For descriptor learning, it is usually better to use balanced training sets whereas classifier learning must respect the characteristics of the problem as represented by $\mathcal{Z}$. 
Given that \( \mathbf{x}(s) = (x_1(s), x_2(s), \ldots, x_n(s)) \in \mathbb{R}^n \) changes with the random choice of \( s \in \mathcal{Z} \), then \( \mathbf{x} \) is said a \textit{random field} with probability density \( \rho(\mathbf{x}): \mathbf{x} \rightarrow [0, 1] \).
Given that \( \mathbf{x}(s) = (x_1(s), x_2(s), \ldots, x_n(s)) \in \mathbb{R}^n \) changes with the random choice of \( s \in \mathbb{Z} \), then \( \mathbf{x} \) is said a random field with probability density \( \rho(\mathbf{x}) : \mathbf{x} \to [0, 1] \).

Likewise, each feature \( x_i(s) \in \mathbb{R}, i \in [1, n] \), changes with the random choice of \( s \in \mathbb{Z} \), then \( x_i \) is said a random variable.
Given that \( \mathbf{x}(s) = (x_1(s), x_2(s), \ldots, x_n(s)) \in \mathbb{R}^n \) changes with the random choice of \( s \in \mathcal{Z} \), then \( \mathbf{x} \) is said a random field with probability density \( \rho(\mathbf{x}) : \mathbf{x} \rightarrow [0, 1] \). 

Likewise, each feature \( x_i(s) \in \mathbb{R}, i \in [1, n] \), changes with the random choice of \( s \in \mathcal{Z} \), then \( x_i \) is said a random variable.

Therefore, the random choices of \( s_j \in \mathcal{Z} \) generate different sequences of observations \( \mathbf{x}_j = \mathbf{x}(s_j), j = 1, 2, \ldots, |\mathcal{Z}_{tr}| = m \), for training and likewise for testing.
Whenever the descriptor (or the classifier) has parameters to be optimized, the use of a third statistically independent set $Z_{ev}$, named **evaluation set**, for optimization may reduce the chances of overfitting.
Whenever the descriptor (or the classifier) has parameters to be optimized, the use of a third statistically independent set $Z_{ev}$, named evaluation set, for optimization may reduce the chances of overfitting.

For a given descriptor, an apprentice classifier can improve performance along learning iterations as it selects training samples for user supervision. This is called active learning.
Sample selection

- Whenever the descriptor (or the classifier) has parameters to be optimized, the use of a third statistically independent set $\mathcal{Z}_{ev}$, named evaluation set, for optimization may reduce the chances of overfitting.

- For a given descriptor, an apprentice classifier can improve performance along learning iterations as it selects training samples for user supervision. This is called active learning.

- Sample selection is never perfect, but cross-validation methods are the most preferable ones.
Cross validation may be called $K$-hold-out, $K$-fold, or $N \times K$-fold [2].
Cross validation may be called $K$-hold-out, $K$-fold, or $N \times K$-fold [2].

- **$K$-hold-out**: $\mathcal{Z}$ is split $K$ times into $P\%$ of samples for $\mathcal{Z}_{tr}$ and $(100 - P)\%$ for $\mathcal{Z}_{ts}$, $0 < P < 100$, to obtain the statistics of the effectiveness measure in the test sets. The instances of $\mathcal{Z}_{tr}$ and $\mathcal{Z}_{ts}$ are not statistically independent.
Sample selection

Cross validation may be called $K$-hold-out, $K$-fold, or $N \times K$-fold [2].

- **$K$-hold-out**: $\mathcal{Z}$ is split $K$ times into $P\%$ of samples for $\mathcal{Z}_{tr}$ and $(100 - P)\%$ for $\mathcal{Z}_{ts}$, $0 < P < 100$, to obtain the statistics of the effectiveness measure in the test sets. The instances of $\mathcal{Z}_{tr}$ and $\mathcal{Z}_{ts}$ are not statistically independent.

- **$K$-fold**: $\mathcal{Z}$ is split into $K$ parts of approximately equal sizes, using each of the parts for testing and the rest for training $K$ times. The instances of $\mathcal{Z}_{ts}$ are statistically independent, but not the instances of $\mathcal{Z}_{tr}$. 
Cross validation may be called $K$-hold-out, $K$-fold, or $N \times K$-fold [2].

- **$K$-hold-out**: $\mathcal{Z}$ is split $K$ times into $P\%$ of samples for $\mathcal{Z}_{tr}$ and $(100 - P)\%$ for $\mathcal{Z}_{ts}$, $0 < P < 100$, to obtain the statistics of the effectiveness measure in the test sets. The instances of $\mathcal{Z}_{tr}$ and $\mathcal{Z}_{ts}$ are not statistically independent.

- **$K$-fold**: $\mathcal{Z}$ is split into $K$ parts of approximately equal sizes, using each of the parts for testing and the rest for training $K$ times. The instances of $\mathcal{Z}_{ts}$ are statistically independent, but not the instances of $\mathcal{Z}_{tr}$.

- **$N \times K$-fold**: $K$-fold is repeated $N$ times, usually with $K = 2$. 
Let $n_{ij}$ be the number of times test samples from category $\omega_i$ have been classified into category $\omega_j$ for $i, j \in [1, c]$ and $m_{ts}$ samples. A confusion matrix is defined as

$$
\begin{bmatrix}
\omega_1 & \omega_2 & \ldots & \omega_c \\
\omega_1 & n_{11} & n_{12} & \ldots & n_{1c} \\
\omega_2 & n_{21} & n_{22} & \ldots & n_{2c} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\omega_c & n_{c1} & n_{c2} & \ldots & n_{cc}
\end{bmatrix}
$$

The total of correct classifications is $\sum_{i=1}^{c} n_{ii}$, being $m_{ts} - \sum_{i=1}^{c} n_{ii}$ the total of misclassifications. Several effectiveness measures can be obtained from the confusion matrix (sensitivity, accuracy, specificity, precision, etc). A “good” one is the Cohen’s kappa, which is robust to imbalanced categories.
Effectiveness and confusion matrix

Let $n_{ij}$ be the number of times test samples from category $\omega_i$ have been classified into category $\omega_j$ for $i, j \in [1, c]$ and $m_{ts}$ samples. A confusion matrix is defined as

$$
\begin{array}{c|cccc}
& \omega_1 & \omega_2 & \ldots & \omega_c \\
\hline
\omega_1 & n_{11} & n_{12} & \ldots & n_{1c} \\
\omega_2 & n_{21} & n_{22} & \ldots & n_{2c} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\omega_c & n_{c1} & n_{c2} & \ldots & n_{cc} \\
\end{array}
$$

- The total of correct classifications is $\sum_{i=1}^{c} n_{ii}$, being $m_{ts} - \sum_{i=1}^{c} n_{ii}$ the total of misclassifications.
Effectiveness and confusion matrix

Let $n_{ij}$ be the number of times test samples from category $\omega_i$ have been classified into category $\omega_j$ for $i, j \in [1, c]$ and $m_{ts}$ samples. A confusion matrix is defined as

$$
\begin{bmatrix}
\omega_1 & \omega_2 & \ldots & \omega_c \\
\omega_1 & n_{11} & n_{12} & \ldots & n_{1c} \\
\omega_2 & n_{21} & n_{22} & \ldots & n_{2c} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\omega_c & n_{c1} & n_{c2} & \ldots & n_{cc}
\end{bmatrix}
$$

- The total of correct classifications is $\sum_{i=1}^{c} n_{ii}$, being $m_{ts} - \sum_{i=1}^{c} n_{ii}$ the total of misclassifications.

- Several effectiveness measures can be obtained from the confusion matrix (sensitivity, accuracy, specificity, precision, etc). A “good” one is the Cohen’s kappa, which is robust to imbalanced categories.
Cohen’s kappa measures the agreement between two raters, $A$ and $B$, such that $n_{ij}$ indicates the number of samples rater $A$ says that they are from category $\omega_i$, while rater $B$ says that they are from category $\omega_j$.

$$\kappa = \frac{P_o - P_e}{1 - P_e},$$
$$P_o = \frac{1}{m_{ts}} \sum_{i=1}^{c} n_{ii},$$
$$P_e = \frac{1}{m_{ts}^2} \sum_{i=1}^{c} N_A(i)N_B(i),$$

where $N_A(i) = \sum_{j=1}^{c} n_{ij}$ is the total of samples rater $A$ says that they are from category $\omega_i$ and $N_B(i) = \sum_{j=1}^{c} n_{ji}$ is the total of samples that rater $B$ says they are from category $\omega_i$. 
Statistical tests

- **Statistical tests** provide a formal way to decide if the results of an experiment are significant or accidental.
**Statistical tests** provide a formal way to decide if the results of an experiment are significant or accidental.

For example, one can measure the Cohen’s kappa $\kappa_i(t)$ of each execution $t = 1, 2, \ldots, T$ of each classifier $C_i$, $i \in [1, n]$, on $T$ statistically independent test sets during cross validation.
Statistical tests

- **Statistical tests** provide a formal way to decide if the results of an experiment are significant or accidental.

- For example, one can measure the Cohen’s kappa $\kappa_i(t)$ of each execution $t = 1, 2, \ldots, T$ of each classifier $C_i, i \in [1, n]$, on $T$ statistically independent test test sets during cross validation.

- A statistical test starts from a **null hypothesis**, such as all classifiers are equivalent, and verify if it can be rejected at some significance level $p$ (e.g., $p = 0.05$).
First, some measure $m_o$, that indicates differences among the classifiers, is obtained from the experiment. For example, for $n = 2$ classifiers and a $5 \times 2$-fold cross validation, one can compute the variances $s_t^2$ of the differences $\kappa_1(t) - \kappa_2(t)$ of the two folds for $t = 1, 2, \ldots, 5$ and define

$$m_o = \frac{\kappa_1(1) - \kappa_2(1)}{\sqrt{\frac{1}{5} \sum_{t=1}^{5} s_t^2}}$$
Statistical tests

- First, some measure $m_o$, that indicates differences among the classifiers, is obtained from the experiment. For example, for $n = 2$ classifiers and a $5 \times 2$-fold cross validation, one can compute the variances $s_t^2$ of the differences $\kappa_1(t) - \kappa_2(t)$ of the two folds for $t = 1, 2, \ldots, 5$ and define

$$m_o = \frac{\kappa_1(1) - \kappa_2(1)}{\sqrt{\frac{1}{5} \sum_{t=1}^{5} s_t^2}}$$

- It is shown that $m_o$ (a random variable) satisfies some probability density $\rho(m_o)$ when the null hypothesis is satisfied. For the example, a $t$-distribution of five degrees of freedom.
Statistical tests

- The areas below the curve $\rho(m_0)$ are tabulated for each value of $m_0$, representing the chances $p$ of the null hypothesis being correct.

If $m_0$ is observed above a critical value such that $p < 0.05$, for instance, we reject the null hypothesis with less than 5% chance to be wrong.

The most popular tests are student's t-test, Wilcoxon signed-rank test, analysis of variance (ANOVA), Tukey's range test, Nemenyi test, and Friedman test.
The areas below the curve $\rho(m_o)$ are tabulated for each value of $m_o$, representing the chances $p$ of the null hypothesis being correct.

If $m_o$ is observed above a critical value such that $p < 0.05$, for instance, we reject the null hypothesis with less than 5% of chance to be wrong.
The areas below the curve $\rho(m_o)$ are tabulated for each value of $m_o$, representing the chances $p$ of the null hypothesis being correct.

If $m_o$ is observed above a critical value such that $p < 0.05$, for instance, we reject the null hypothesis with less than 5% of chance to be wrong.

The most popular tests are student’s t-test, Wilcoxon signed-rank test, analysis of variance (ANOVA), Tukey’s range test, Nemenyi test, and Friedman test.
A probability density function (pdf) \( \rho \) of a random variable \( x \) (e.g., a feature) is a mapping \( \rho: x \to \mathbb{R} \), such that \( \rho(x) \geq 0 \) and \( P(x_o \leq x \leq x_f) = \int_{x_o}^{x_f} \rho(x)dx \in [0, 1] \) measures the probability of the value \( x \) be in the interval \( [x_o, x_f] \).
A probability density function (pdf) $\rho$ of a random variable $x$ (e.g., a feature) is a mapping $\rho : x \to \mathbb{R}$, such that $\rho(x) \geq 0$ and $P(x_o \leq x \leq x_f) = \int_{x_o}^{x_f} \rho(x)dx \in [0, 1]$ measures the probability of the value $x$ be in the interval $[x_o, x_f]$.

The pdf may also be called probability distribution and, for discrete random variables (e.g., pixel intensity), it may be called probability mass function.
A probability density function (pdf) $\rho$ of a random variable $x$ (e.g., a feature) is a mapping $\rho : x \rightarrow \mathbb{R}$, such that $\rho(x) \geq 0$ and $P(x_o \leq x \leq x_f) = \int_{x_o}^{x_f} \rho(x)dx \in [0, 1]$ measures the probability of the value $x$ be in the interval $[x_o, x_f]$.

The pdf may also be called probability distribution and, for discrete random variables (e.g., pixel intensity), it may be called probability mass function.

The normalized image histogram, for instance, represents the pdf of the pixel intensity. However, the color $x = (x_1, x_2, x_3)$ of the pixels is a discrete random field. In this case, the $\rho(x) \geq 0$ defines a manifold in $\mathbb{R}^4$. 
In the general case, \( x = (x_1, x_2, \ldots, x_n) \) defines a manifold in \( \mathbb{R}^{n+1} \).
Probability density function

- In the general case, \( \mathbf{x} = (x_1, x_2, \ldots, x_n) \) defines a manifold in \( \mathbb{R}^{n+1} \).

- The simplest approach to estimate the pdf starts by counting the number \( \Omega(\mathbf{x}(s)) \) of samples \( t \in \mathcal{Z}_{tr} \), whose point \( \mathbf{x}(t) \) falls within a hypercube of volume \( h^n \) (Parzen window) around \( \mathbf{x}(s) \in \mathbb{R}^n \).
In the general case, \( \mathbf{x} = (x_1, x_2, \ldots, x_n) \) defines a manifold in \( \mathbb{R}^{n+1} \).

The simplest approach to estimate the pdf starts by counting the number \( \Omega(\mathbf{x}(s)) \) of samples \( t \in \mathcal{Z}_{tr} \), whose point \( \mathbf{x}(t) \) falls within a hypercube of volume \( h^n \) (Parzen window) around \( \mathbf{x}(s) \in \mathbb{R}^n \).

Let \( \mathcal{A} \) be an adjacency relation defined by

\[
\mathcal{A}(s) : \{ t \in \mathcal{Z}_{tr} \mid |x_i(t) - x_i(s)| \leq \frac{h}{2}, i = 1, 2, \ldots, n \},
\]

and \( w(t) \) be a kernel weight defined by

\[
w(t) = \begin{cases} 
1 & \text{if } t \in \mathcal{A}(s), \\
0 & \text{otherwise}.
\end{cases}
\]
The counting $\Omega(x(s))$ is defined by

$$\Omega(x(s)) = \sum_{\forall t \in A(s)} w(t).$$

Clearly, the choice of $h$ is important and a fixed scale $k \geq 1$ can make it adaptive:

$$A_k(s) : \{ t \in Z_{tr} \mid x(t) \text{ is a } k\text{-closest observation of } x(s) \},$$

$$w(t) = \begin{cases} \exp \left[ -\frac{\|x(t) - x(s)\|^2}{2\sigma^2} \right] & \text{if } t \in A_k(s), \\ 0 & \text{otherwise}, \end{cases}$$

for $\sigma = \frac{1}{3} \max_{(s,t) \in A_k} \{\|x(t) - x(s)\|\}$ [4].
Let \( \{ u^{(1)}, u^{(2)}, \ldots, u^{(L)} \} \) be the set of distinct observations \( x(s) \), \( \forall s \in Z_{tr} \). Then, the probability density function \( \rho \) can be estimated at any point \( x(s) = u^{(j)} \in \mathbb{R}^n \), \( 1 \leq j \leq L \), as

\[
\begin{align*}
\Omega(u^{(j)}) & \leftarrow \Omega(x(s)) \\
\rho(u^{(j)}) & = \frac{\Omega(u^{(j)})}{\sum_{i=1}^{L} \Omega(u^{(i)})} \\
\rho(s) & \leftarrow \rho(x(s)) = \rho(u^{(j)})
\end{align*}
\]
For an image $\hat{I} = (D_I, I)$, for instance, one can create a pdf image by assigning to each pixel $p \in D_I$ a pdf value $\rho(p)$ as estimated in the feature space defined by $I$. 

![Image of a toy and its pdf representation]
Probability density function

For an image $\hat{I} = (D, I)$, for instance, one can create a pdf image by assigning to each pixel $p \in D$ a pdf value $\rho(p)$ as estimated in the feature space defined by $I$. 
More basic concepts in Statistics

- $\rho(x) \geq 0, \ x = (x_1, x_2, \ldots, x_n)$, is said joint pdf.

- When $\rho(x) = \rho(x_1)\rho(x_2)\ldots\rho(x_n)$, the variables are said statistically independent.
More basic concepts in Statistics

- \( \rho(x) \geq 0, \ x = (x_1, x_2, \ldots, x_n) \), is said joint pdf.

- When \( \rho(x) = \rho(x_1) \rho(x_2) \ldots \rho(x_n) \), the variables are said statistically independent.

- Let \( \mathcal{U} = \{u_1, u_2, \ldots, u_{L_i}\} \) and \( \mathcal{V} = \{v_1, v_2, \ldots, v_{L_j}\} \) be the respective sets of distinct observations \( x_i(s) \) and \( x_j(s) \), \( 1 \leq i \neq j \leq n, \ \forall s \in \mathbb{Z}_{tr} \) (i.e., observations of two random variables of \( x \)).
The Entropy $H$ of a pdf $\rho(x_i)$ measures the unpredictability of $x_i$ — i.e., less uniform is $\rho(x_i)$, lower is $H$, higher is the predictability.

$$H(\rho) = -\sum_{k=1}^{L_i} \rho(u_k) \log_2 \rho(u_k).$$
The Entropy $H$ of a pdf $\rho(x_i)$ measures the unpredictability of $x_i$ — i.e., less uniform is $\rho(x_i)$, lower is $H$, higher is the predictability.

$$H(\rho) = -\sum_{k=1}^{L_i} \rho(u_k) \log_2 \rho(u_k).$$

Given $\rho_1(x_i)$ and $\rho_2(x_i)$, obtained from two training sets (or images), the relative entropy, or Kullback-Leibler distance $D(\rho_1, \rho_2)$, measures their cross entropy.

$$D(\rho_1, \rho_2) = \sum_{k=1}^{L_i} \rho_2(u_k) \ln \frac{\rho_2(u_k)}{\rho_1(u_k)}.$$
Given $\rho_1(x_i)$ and $\rho_2(x_j)$, from possibly distinct training sets, the mutual information $I(\rho_1, \rho_2)$ is the reduction in uncertainty about one variable due to the knowledge of the other.

$$I(\rho_1, \rho_2) = H(\rho_1) - H(\rho_1 \setminus \rho_2),$$

$$I(\rho_1, \rho_2) = \sum_{k=1}^{L_i} \sum_{l=1}^{L_j} \rho(u_k, v_l) \log_2 \frac{\rho(u_k, v_l)}{\rho_1(u_k) \rho_2(v_l)}.$$

Mutual information is well used when aligning two images in the same domain (image registration). The alignment aims at maximizing the mutual information.
the mean $\mu_i = E[x_i]$ of a random variable $x_i$ can be estimated by the first moment

$$
\mu_i = \sum_{k=1}^{L_i} u_k \rho(u_k)
$$

the variance $\sigma_i = E[(x_i - \mu_i)^2]$ can be estimated by the second moment of $(x_i - \mu_i)$

$$
\sigma_i^2 = \sum_{k=1}^{L_i} [u_k - \mu_i]^2 \rho(u_k)
$$
The cross-moment $\sigma_{ij} = \sigma_{ji} = E[(x_i - \mu_i)(x_j - \mu_j)]$ (covariance) of $x_i$ and $x_j$ can be estimated as

$$
\sigma_{ij} = \sum_{k=1}^{L_i} \sum_{l=1}^{L_j} [(u_k - \mu_i)(v_l - \mu_j)] \rho(u_k, v_l)
$$
More basic concepts in Statistics

- The cross-moment $\sigma_{ij} = \sigma_{ji} = E[(x_i - \mu_i)(x_j - \mu_j)]$ (covariance) of $x_i$ and $x_j$ can be estimated as

$$\sigma_{ij} = \sum_{k=1}^{L_i} \sum_{l=1}^{L_j} [(u_k - \mu_i)(v_l - \mu_j)] \rho(u_k, v_l)$$

- The mean vector $\mu = E[x] \in \mathbb{R}^n$ and the covariance matrix $\Sigma = E[(x(s) - \mu)(x(s) - \mu)^t]$ is

$$\Sigma = \begin{bmatrix}
\sigma_1^2 & \sigma_{12} & \ldots & \sigma_{1n} \\
\sigma_{21} & \sigma_2^2 & \ldots & \sigma_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\sigma_{n1} & \sigma_{n2} & \ldots & \sigma_n^2
\end{bmatrix}.$$
The joint distribution $\rho(u_k, v_l) = \rho(u_k)\rho(v_l \setminus u_k) = \rho(v_l)\rho(u_k \setminus v_l)$.

The Cauchy-Schwarz inequality says that $\sigma_{ij} \leq \sigma_i^2 \sigma_j^2$.

The Pearson correlation coefficient is defined as $\frac{\sigma_{ij}}{\sigma_i \sigma_j}$.

The variables $x_i$ and $x_j$ are said uncorrelated when $\sigma_{ij} = 0$. 
More basic concepts in Statistics

- For \( c \) categories, \( \{ w_k \}_{k=1}^c \), the Bayes rule says that the occurrence probability of a category \( w_k \) given an observation \( x \) is

\[
P(w_k \mid x) = \frac{P(w_k) \rho(x \mid w_k)}{\rho(x)},
\]

where \( P(w_k \mid x) \) is named the posterior probability, \( P(w_k) \) is the prior probability, the conditional density function \( \rho(x \mid w_k) \) is the likelihood, and \( \rho(x) \) is the evidence.
For \( c \) categories, \( \{ w_k \}_{k=1}^c \), the Bayes rule says that the occurrence probability of a category \( w_k \) given an observation \( x \) is

\[
P(w_k \mid x) = \frac{P(w_k) \rho(x \mid w_k)}{\rho(x)} ,
\]

where \( P(w_k \mid x) \) is named the posterior probability, \( P(w_k) \) is the prior probability, the conditional density function \( \rho(x \mid w_k) \) is the likelihood, and \( \rho(x) \) is the evidence.

The evidence \( \rho(x) = \sum_{i=1}^c P(w_i) \rho(x \mid w_i) \).
For $c$ categories, $\{w_k\}_{k=1}^c$, the Bayes rule says that the occurrence probability of a category $w_k$ given an observation $x$ is

$$P(w_k \mid x) = \frac{P(w_k)\rho(x \mid w_k)}{\rho(x)},$$

where $P(w_k \mid x)$ is named the posterior probability, $P(w_k)$ is the prior probability, the conditional density function $\rho(x \mid w_k)$ is the likelihood, and $\rho(x)$ is the evidence.

The evidence $\rho(x) = \sum_{i=1}^c P(w_i)\rho(x \mid w_i)$.

The estimation of $\rho(x \mid w_k)$ can be similar to the one of $\rho(x)$, but using only adjacent samples $t \in Z_{tr}$ from category $w_k$. 
As we will see in the course, it is possible to separate the $g$ domes of the pdf manifold $\rho(x)$ into $g$ clusters.

This is the result of the method in [4].
Clustering methods

For images, using the Lab color space, it can obtain the following results.
Clustering methods

For images, using the Lab color space, it can obtain the following results.
Clustering methods

The most used method, however, assumes that the clusters are hyper-spheres — the $g$-means clustering.

$g = 2$ groups  \hspace{1cm}  g = 4$ groups \hspace{1cm}  2 categories
The $g$-means clustering algorithm finds $g$ groups \( \{G_k\}_{k=1}^g \) (clusters) by assigning each sample \( s \in \mathcal{Z}_{tr} = \{s_j\}_{j=1}^m \), \( m \gg g \), to one group, such that

\[
\sum_{k=1}^g \sum_{\forall s \in G_k} \| x(s) - \mu_k \|^2
\]

is minimized and

\[
\mu_k = \frac{1}{|G_k|} \sum_{\forall s \in G_k} x(s)
\]

is the centroid of group \( G_k \). The algorithm works as follows.
Clustering methods

Input: A training dataset \((Z_{tr}, X_{tr})\).
Output: A label map \(L: Z_{tr} \rightarrow \{k\}_{k=1}^{g}\) (i.e., \(L(s) = k \Rightarrow s \in G_k\)).

1. Select \(g\) random centroids \(\{\mu_k\}_{k=1}^{g}\) from \(\{x(s_j)\}_{j=1}^{m}\).

2. For each iteration \(t = 1, 2, \ldots, T\) do.

3. For each sample \(s \in \{s_j\}_{j=1}^{m}\) do.

4. Set \(L(s) \leftarrow \arg \min_{k=1,2,\ldots,g} \{\|x(s) - \mu_k\|^2\}\).

5. For each group \(G_k, k = 1, 2, \ldots, g\), do.

6. Update \(\mu_k \leftarrow \frac{1}{|G_k|} \sum_{\forall s \in Z_{tr} | L(s) = k} x(s)\).
Clustering methods

- The algorithm may be interrupted, when the differences between previous and current centroids are negligible, and a test sample \( s \in Z_{ts} \) is assigned to the group of its closest centroid in \( \mathbb{R}^n \).

- The representative \( x(s) \) of group \( G_k \) can also be selected as the observation closest to the others in \( G_k \).

\[
x(s) \leftarrow \arg \min_{x(s'), \forall s', t \in Z_{tr} | L(t) = L(s') = k} \| x(t) - x(s') \|^2.
\]

The observation \( x(s) \) is called medoid and the method becomes \( g \)-medoids.

- Other popular clustering approaches are mean-shift, normalized cut, Gaussian mixture models, and single-linkage.
Let $L_i(x)$ be a discriminant function that assigns a sample $s \in Z_{tr}$ with observation $x(s)$ to category $\omega_i$, by setting label $L(s) \leftarrow i$, $1 \leq i \leq c$, when

$$\omega_i = \arg\max_{j=1,2,...,c} \{L_j(x(s))\}.$$
Classification methods

Let $L_i(x)$ be a discriminant function that assigns a sample $s \in \mathcal{Z}_{tr}$ with observation $x(s)$ to category $\omega_i$, by setting label $L(s) \leftarrow i, \ 1 \leq i \leq c$, when

$$\omega_i = \arg \max_{j=1,2,...,c} \{L_j(x(s))\}.$$

A Bayesian classifier adopts $L_i(x) = P(\omega_i|x)$.

Indeed, one can use any other equivalent function, such as $L_i(x) = \log [P(\omega_i)\rho(x|\omega_i)]$. 
Let $\rho(x \mid \omega_i)$ be a Normal distribution $N(\mu_i, \Sigma_i)$. Then,

$$L_i(x) = \log[P(\omega_i)] + \log\left\{ \frac{1}{(2\pi)^{n/2} \sqrt{|\Sigma_i|}} \exp\left[ -\frac{1}{2} (x - \mu_i)^t \Sigma_i^{-1} (x - \mu_i) \right] \right\},$$

where $\mu_i$ and $\Sigma_i$ can be estimated as the mean vector and covariance matrix of the observations $x(s), s \in Z_{tr}$, whose true label $\lambda(s) = i$ (i.e., $s \in \omega_i$). The argument $(x - \mu_i)^t \Sigma_i^{-1} (x - \mu_i)$ is the squared of the Mahalanobis distance between $x(s)$ and $\mu_i$. 
A quadratic discriminant classifier (QDC) adopts

\[ L_i(x) = w_{i0} + w_i^t x + x^t W_i x, \]

where \( x, w_i \in \mathbb{R}^n \), \( W_i \) is a \( n \times n \) matrix, and \( w_{i0} \in \mathbb{R} \).

By adopting \( \rho(x|\omega_i) = N(\mu_i, \Sigma_i) \), the Bayesian classifier becomes a QDC, where \( w_{i0} = \log \left[ P(\omega_i) \right] - \frac{1}{2} \mu_i^t \Sigma_i^{-1} \mu_i - \frac{1}{2} \log (|\Sigma_i|) \), \( w_i = \mu_i^t \Sigma_i^{-1} \), and \( W_i = -\frac{1}{2} \Sigma_i^{-1} \).

Obs: Category-independent terms are eliminated.
A linear discriminant classifier (LDC) adopts

\[ L_i(x) = w_{i0} + w_i^t x. \]

By adopting \( \rho(x|\omega_i) = N(\mu_i, S_T) \), \( S_T = \frac{1}{m} \sum_{i=1}^{c} m_i \Sigma_i \), where \( m_i \) is the number of training samples from category \( \omega_i \), the Bayesian classifier becomes a LDC, where \( w_{i0} = \log [P(\omega_i)] - \frac{1}{2} \mu_i^t S_T^{-1} \mu_i \) and \( w_i^t = \mu_i^t S_T^{-1} \).

Obs: Category-independent terms are eliminated.
The \( k \)-nearest neighbor classifier, \( k \geq 1 \), adopts the \( k \)-closest adjacency relation

\[ A_k(s) : \{ t \in \mathcal{Z}_{tr} \mid \mathbf{x}(t) \text{ is a } k\text{-closest observation of } \mathbf{x}(s) \} \]

and counts the number \( k_i \) of samples \( t \) whose true label \( \lambda(t) = i \) for each \( i = 1, 2, \ldots, c \).

It then approximates \( L_i(\mathbf{x}) = P(\omega_i \mid \mathbf{x}) \approx \frac{k_i}{k} \) and assigns to \( s \) the label \( L(s) \in \{ i \}_{i=1}^c \) (i.e., it classifies \( s \) as belonging to \( \omega_i \)) when

\[ L_i(\mathbf{x}(s)) = \max_{j=1,2,\ldots,c} \{ L_j(\mathbf{x}(s)) \} \].
The reduction of the feature space $\mathbb{R}^n$ to some dimension $1 \leq k < n$ is also useful to handle the curse of high dimensionality or to understand the distribution of the observations $x(s)$, $\forall s \in Z_{tr}$, in $\mathbb{R}^n$. A linear projection is $Y_{tr} = WX_{tr}$, where $W$ is a $k \times n$ matrix and $X_{tr}$ is the $n \times m$ feature matrix of the $m$ training samples. In the reduction by principal component analysis (PCA), the rows of $W$ are the corresponding eigenvectors of the $k$ highest eigenvalues of the covariance matrix $\Sigma$ of the observations.
The reduction of the feature space $\mathbb{R}^n$ to some dimension $1 \leq k < n$ is also useful to handle the curse of high dimensionality or to understand the distribution of the observations $x(s), \forall s \in \mathcal{Z}_{tr},$ in $\mathbb{R}^n$.

The reduction can use a linear (e.g., PCA, LDA) or a non-linear (e.g., MDS, t-SNE) projection.
The reduction of the feature space $\mathbb{R}^n$ to some dimension $1 \leq k < n$ is also useful to handle the curse of high dimensionality or to understand the distribution of the observations $x(s), \forall s \in Z_{tr}$, in $\mathbb{R}^n$.

The reduction can use a linear (e.g., PCA, LDA) or a non-linear (e.g., MDS, t-SNE) projection.

A linear projection is $Y_{tr} = WX_{tr}$, where $W$ is a $k \times n$ matrix and $X_{tr}$ is the $n \times m$ feature matrix of the $m$ training samples.
The reduction of the feature space $\mathbb{R}^n$ to some dimension $1 \leq k < n$ is also useful to handle the curse of high dimensionality or to understand the distribution of the observations $\mathbf{x}(s), \forall s \in \mathcal{Z}_{tr}$, in $\mathbb{R}^n$.

The reduction can use a linear (e.g., PCA, LDA) or a non-linear (e.g., MDS,t-SNE) projection.

A linear projection is $Y_{tr} = WX_{tr}$, where $W$ is a $k \times n$ matrix and $X_{tr}$ is the $n \times m$ feature matrix of the $m$ training samples.

In the reduction by principal component analysis (PCA), the rows of $W$ are the corresponding eigenvectors of the $k$ highest eigenvalues of the covariance matrix $\Sigma$ of the observations.
Role of feature space reduction

The reduction by linear discriminant analysis (LDA) considers the true label $\lambda(s) \in \{i\}_{i=1}^c$ (i.e., $s \in \omega_i$) of the training samples $s \in Z_{tr}$ and the rows of $W$ are the corresponding eigenvectors of the $k = c - 1$ highest eigenvalues of the matrix $S_B S_W^{-1}$, where

$$S_B = \sum_{i=1}^c m_i (\mu_i - \mu)(\mu_i - \mu)^t$$

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

$$S_i = \sum_{\forall s \in Z_{tr}} (x(s) - \mu_i)(x(s) - \mu_i)^t,$$

$$\mu = \frac{1}{m} \sum_{\forall s \in Z_{tr}} x(s)$$

$$\mu_i = \frac{1}{m_i} \sum_{\forall s \in Z_{tr} | \lambda(s) = i} x(s).$$
From Rauber et al. [3].
Role of feature space reduction

From Rauber et al. [3].

Figure 2. *Madelon* dataset. (a) Training set (NH: 53.9%). (b) Test set (NH: 50.97%). (c) Training set, feature subset (NH: 83.56%). (d) Test set, feature subset (NH: 71.15%).
[1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville.  
*Deep Learning.*  

*Combining Pattern Classifiers: Methods and Algorithms.*  

Projections as visual aids for classification system design.  
*Information Visualization,* 2017.

Data clustering as an optimum-path forest problem with applications in image analysis.  

Rapid object detection using a boosted cascade of simple features.