Statistical Learning

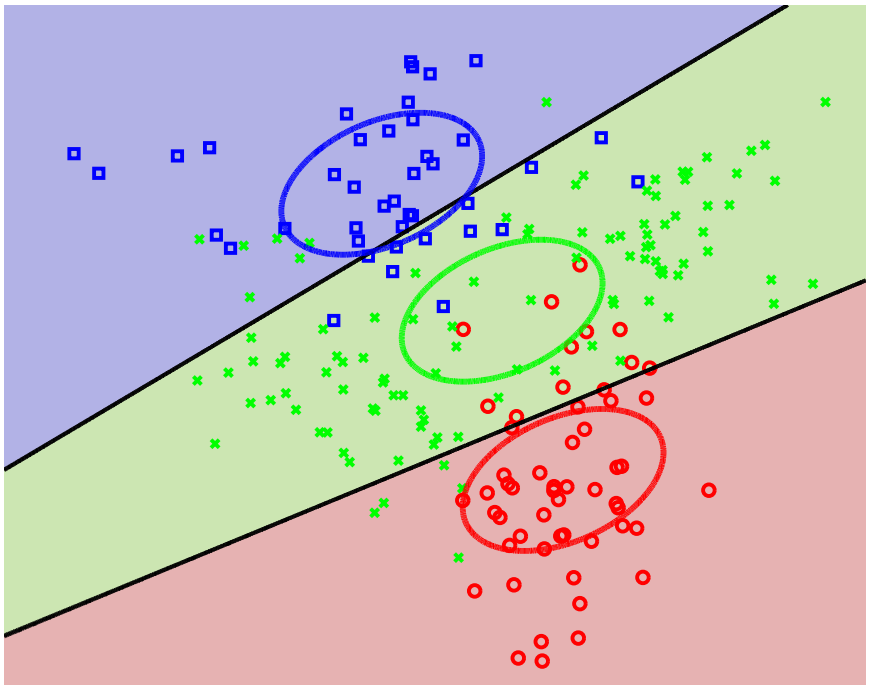
Master Spécialisé Intelligence Artificielle de Confiance (IAC)
@ Centrale Supélec en partenariat avec l'IRT SystemX
2024/2025.

Faïcel Chamroukhi

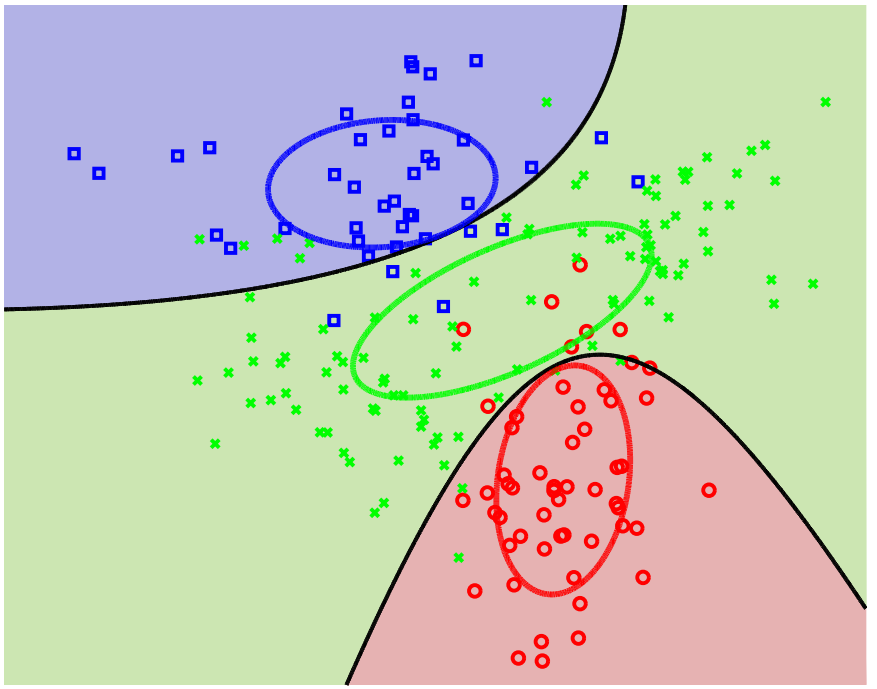
Classification

1. Supervised Learning
   - Gaussian Discriminant Analysis
   - Linear Discriminant Analysis
   - Quadratic Discriminant Analysis
   - Mixture Discriminant Analysis

Linear Discriminant Analysis (LDA)



Quadratic Discriminant Analysis (QDA)



Learning Framework for Classification

- The data are represented by a random pair $(\boldsymbol{X}, Y) \in \mathcal{X} \times \mathcal{Y}$ where $\boldsymbol{X}$ is a vector of descriptors for some variable of interest Y
- The objective is **Prediction**, i.e. to seek for a prediction function $h : \mathcal{X} \to \mathcal{Y}$ for which $\widehat{y} = h(\boldsymbol{x})$ is a good approximation of the true output y
- In a **classification** problem : typically $\boldsymbol{X} \in \mathcal{X} \subset \mathbb{R}^p$ and $Y \in \mathcal{Y} = \{0, 1\}, \{-1, +1\}$ (binary classification) or $\{1, \cdots, K\}$ (multiclass classification)

↪ We will mainly focus on parametric probabilistic models of the form

$$Y = h(X) + \epsilon, \epsilon \sim p_\theta$$

with the conditional distr. $P(Y|X, h)$ can be computed in terms of $P_\theta(Y - h(X))$. $P(X, Y|h) = P(Y|X, h)P(X|h)$

- Data : a random sample $(\boldsymbol{X}_i, Y_i)_{i=1}^n$ with observed values $\mathcal{D}_n = (\boldsymbol{x}_i, y_i)_{i=1}^n$
- **Data-Scientist's role :** given the **data**, choose a **prediction function** h from a class $\mathcal{H}$ that attempts to "minimize" the prediction error for of all possible data (**risk**) $R(h)$, under a **loss** function ℓ measuring the error of predicting Y by $h(X)$.

  ↪ minimize the **empirical risk** (data-$\mathcal{D}_n$-driven) $R_n(h)$

  ↪ Minimizing $R_n(h)$ may require an optimization **algorithm** $\mathcal{A}$
- Data-Scientist's "**Toolbox**" **:** {Data, loss, hypothesis, algorithm}

Learning Framework for Classification

- The data are represented by a random pair $(\boldsymbol{X}, Y) \in \mathcal{X} \times \mathcal{Y}$ where $\boldsymbol{X}$ is a vector of descriptors for some variable of interest Y
- The objective is **Prediction**, i.e. to seek for a prediction function $h : \mathcal{X} \to \mathcal{Y}$ for which $\widehat{y} = h(\boldsymbol{x})$ is a good approximation of the true output y
- In a **classification** problem : typically $\boldsymbol{X} \in \mathcal{X} \subset \mathbb{R}^p$ and $Y \in \mathcal{Y} = \{0, 1\}, \{-1, +1\}$ (binary classification) or $\{1, \cdots, K\}$ (multiclass classification)

↪ We will mainly focus on parametric probabilistic models of the form

$$Y = h(X) + \epsilon, \epsilon \sim p_\theta$$

with the conditional distr. $P(Y|X, h)$ can be computed in terms of $P_\theta(Y - h(X))$. $P(X, Y|h) = P(Y|X, h)P(X|h)$

- Data : a random sample $(\boldsymbol{X}_i, Y_i)_{i=1}^n$ with observed values $\mathcal{D}_n = (\boldsymbol{x}_i, y_i)_{i=1}^n$
- **Data-Scientist's role :** given the **data**, choose a **prediction function** h from a class $\mathcal{H}$ that attempts to "minimize" the prediction error for of all possible data (**risk**) $R(h)$, under a **loss** function ℓ measuring the error of predicting Y by $h(X)$.

  ↪ minimize the **empirical risk** (data-$\mathcal{D}_n$-driven) $R_n(h)$

  ↪ Minimizing $R_n(h)$ may require an optimization **algorithm** $\mathcal{A}$
- Data-Scientist's "**Toolbox**" **:** {Data, loss, hypothesis, algorithm}

Learning Framework for Classification

- The data are represented by a random pair $(\boldsymbol{X}, Y) \in \mathcal{X} \times \mathcal{Y}$ where $\boldsymbol{X}$ is a vector of descriptors for some variable of interest Y
- The objective is **Prediction**, i.e. to seek for a prediction function $h : \mathcal{X} \to \mathcal{Y}$ for which $\widehat{y} = h(\boldsymbol{x})$ is a good approximation of the true output y
- In a **classification** problem : typically $\boldsymbol{X} \in \mathcal{X} \subset \mathbb{R}^p$ and $Y \in \mathcal{Y} = \{0, 1\}, \{-1, +1\}$ (binary classification) or $\{1, \cdots, K\}$ (multiclass classification)

↪ We will mainly focus on parametric probabilistic models of the form

$$Y = h(X) + \epsilon, \epsilon \sim p_\theta$$

with the conditional distr. $P(Y|X, h)$ can be computed in terms of $P_\theta(Y - h(X))$.
$P(X, Y|h) = P(Y|X, h)P(X|h)$

- Data : a random sample $(\boldsymbol{X}_i, Y_i)_{i=1}^n$ with observed values $\mathcal{D}_n = (\boldsymbol{x}_i, y_i)_{i=1}^n$
- **Data-Scientist's role :** given the **data**, choose a **prediction function** h from a class $\mathcal{H}$ that attempts to "minimize" the prediction error for of all possible data (**risk**) $R(h)$, under a **loss** function ℓ measuring the error of predicting Y by $h(X)$.
  ↪ minimize the **empirical risk** (data-$\mathcal{D}_n$-driven) $R_n(h)$
  ↪ Minimizing $R_n(h)$ may require an optimization **algorithm** $\mathcal{A}$
- Data-Scientist's "**Toolbox**" **:** {Data, loss, hypothesis, algorithm}

Learning Framework for Classification

- The data are represented by a random pair $(\boldsymbol{X}, Y) \in \mathcal{X} \times \mathcal{Y}$ where $\boldsymbol{X}$ is a vector of descriptors for some variable of interest Y
- The objective is **Prediction**, i.e. to seek for a prediction function $h : \mathcal{X} \to \mathcal{Y}$ for which $\widehat{y} = h(\boldsymbol{x})$ is a good approximation of the true output y
- In a **classification** problem : typically $\boldsymbol{X} \in \mathcal{X} \subset \mathbb{R}^p$ and $Y \in \mathcal{Y} = \{0, 1\}, \{-1, +1\}$ (binary classification) or $\{1, \cdots, K\}$ (multiclass classification)

↪ We will mainly focus on parametric probabilistic models of the form

$$Y = h(X) + \epsilon, \epsilon \sim p_\theta$$

with the conditional distr. $P(Y|X, h)$ can be computed in terms of $P_\theta(Y - h(X))$.
$P(X, Y|h) = P(Y|X, h)P(X|h)$

- Data : a random sample $(\boldsymbol{X}_i, Y_i)_{i=1}^n$ with observed values $\mathcal{D}_n = (\boldsymbol{x}_i, y_i)_{i=1}^n$
- **Data-Scientist's role :** given the **data**, choose a **prediction function** h from a class $\mathcal{H}$ that attempts to "minimize" the prediction error for of all possible data (**risk**) $R(h)$, under a **loss** function ℓ measuring the error of predicting Y by $h(X)$.
  ↪ minimize the **empirical risk** (data-$\mathcal{D}_n$-driven) $R_n(h)$
  ↪ Minimizing $R_n(h)$ may require an optimization **algorithm** $\mathcal{A}$
- Data-Scientist's "**Toolbox**" **:** {Data, loss, hypothesis, algorithm}

Learning Framework for Classification

- The data are represented by a random pair $(\boldsymbol{X}, Y) \in \mathcal{X} \times \mathcal{Y}$ where $\boldsymbol{X}$ is a vector of descriptors for some variable of interest Y
- The objective is **Prediction**, i.e. to seek for a prediction function $h : \mathcal{X} \to \mathcal{Y}$ for which $\widehat{y} = h(\boldsymbol{x})$ is a good approximation of the true output y
- In a **classification** problem : typically $\boldsymbol{X} \in \mathcal{X} \subset \mathbb{R}^p$ and $Y \in \mathcal{Y} = \{0, 1\}, \{-1, +1\}$ (binary classification) or $\{1, \cdots, K\}$ (multiclass classification)

↪ We will mainly focus on parametric probabilistic models of the form

$$Y = h(X) + \epsilon, \epsilon \sim p_\theta$$

with the conditional distr. $P(Y|X, h)$ can be computed in terms of $P_\theta(Y - h(X))$. $P(X, Y|h) = P(Y|X, h)P(X|h)$

- Data : a random sample $(\boldsymbol{X}_i, Y_i)_{i=1}^n$ with observed values $\mathcal{D}_n = (\boldsymbol{x}_i, y_i)_{i=1}^n$
- **Data-Scientist's role :** given the **data**, choose a **prediction function** h from a class $\mathcal{H}$ that attempts to "minimize" the prediction error for of all possible data (**risk**) $R(h)$, under a **loss** function ℓ measuring the error of predicting Y by $h(X)$.

  ↪ minimize the **empirical risk** (data-$\mathcal{D}_n$-driven) $R_n(h)$

  ↪ Minimizing $R_n(h)$ may require an optimization **algorithm** $\mathcal{A}$
- Data-Scientist's "**Toolbox**" **:** {Data, loss, hypothesis, algorithm}

Learning Framework for Classification

- The data are represented by a random pair $(\boldsymbol{X}, Y) \in \mathcal{X} \times \mathcal{Y}$ where $\boldsymbol{X}$ is a vector of descriptors for some variable of interest Y
- The objective is **Prediction**, i.e. to seek for a prediction function $h : \mathcal{X} \to \mathcal{Y}$ for which $\widehat{y} = h(\boldsymbol{x})$ is a good approximation of the true output y
- In a **classification** problem : typically $\boldsymbol{X} \in \mathcal{X} \subset \mathbb{R}^p$ and $Y \in \mathcal{Y} = \{0, 1\}, \{-1, +1\}$ (binary classification) or $\{1, \cdots, K\}$ (multiclass classification)

↪ We will mainly focus on parametric probabilistic models of the form

$$Y = h(X) + \epsilon, \epsilon \sim p_\theta$$

with the conditional distr. $P(Y|X, h)$ can be computed in terms of $P_\theta(Y - h(X))$. $P(X, Y|h) = P(Y|X, h)P(X|h)$

- Data : a random sample $(\boldsymbol{X}_i, Y_i)_{i=1}^n$ with observed values $\mathcal{D}_n = (\boldsymbol{x}_i, y_i)_{i=1}^n$
- **Data-Scientist's role :** given the **data**, choose a **prediction function** h from a class $\mathcal{H}$ that attempts to "minimize" the prediction error for of all possible data (**risk**) $R(h)$, under a **loss** function ℓ measuring the error of predicting Y by $h(X)$.

  ↪ minimize the **empirical risk** (data-$\mathcal{D}_n$-driven) $R_n(h)$

  ↪ Minimizing $R_n(h)$ may require an optimization **algorithm** $\mathcal{A}$
- Data-Scientist's "**Toolbox**" **:** {Data, loss, hypothesis, algorithm}

Learning Framework for Classification

- The data are represented by a random pair $(\boldsymbol{X}, Y) \in \mathcal{X} \times \mathcal{Y}$ where $\boldsymbol{X}$ is a vector of descriptors for some variable of interest Y
- The objective is **Prediction**, i.e. to seek for a prediction function $h : \mathcal{X} \to \mathcal{Y}$ for which $\widehat{y} = h(\boldsymbol{x})$ is a good approximation of the true output y
- In a **classification** problem : typically $\boldsymbol{X} \in \mathcal{X} \subset \mathbb{R}^p$ and $Y \in \mathcal{Y} = \{0, 1\}, \{-1, +1\}$ (binary classification) or $\{1, \cdots, K\}$ (multiclass classification)

↪ We will mainly focus on parametric probabilistic models of the form

$$Y = h(X) + \epsilon, \epsilon \sim p_\theta$$

with the conditional distr. $P(Y|X, h)$ can be computed in terms of $P_\theta(Y - h(X))$. $P(X, Y|h) = P(Y|X, h)P(X|h)$

- Data : a random sample $(\boldsymbol{X}_i, Y_i)_{i=1}^n$ with observed values $\mathcal{D}_n = (\boldsymbol{x}_i, y_i)_{i=1}^n$
- **Data-Scientist's role :** given the **data**, choose a **prediction function** h from a class $\mathcal{H}$ that attempts to "minimize" the prediction error for of all possible data (**risk**) $R(h)$, under a **loss** function ℓ measuring the error of predicting Y by $h(X)$.
  ↪ minimize the **empirical risk** (data-$\mathcal{D}_n$-driven) $R_n(h)$
  ↪ Minimizing $R_n(h)$ may require an optimization **algorithm** $\mathcal{A}$
- Data-Scientist's "**Toolbox**" **:** {Data, loss, hypothesis, algorithm}

Learning Framework for Classification

- The data are represented by a random pair $(\boldsymbol{X}, Y) \in \mathcal{X} \times \mathcal{Y}$ where $\boldsymbol{X}$ is a vector of descriptors for some variable of interest Y
- The objective is **Prediction**, i.e. to seek for a prediction function $h : \mathcal{X} \to \mathcal{Y}$ for which $\widehat{y} = h(\boldsymbol{x})$ is a good approximation of the true output y
- In a **classification** problem : typically $\boldsymbol{X} \in \mathcal{X} \subset \mathbb{R}^p$ and $Y \in \mathcal{Y} = \{0, 1\}, \{-1, +1\}$ (binary classification) or $\{1, \cdots, K\}$ (multiclass classification)

↪ We will mainly focus on parametric probabilistic models of the form

$$Y = h(X) + \epsilon, \epsilon \sim p_\theta$$

with the conditional distr. $P(Y|X, h)$ can be computed in terms of $P_\theta(Y - h(X))$. $P(X, Y|h) = P(Y|X, h)P(X|h)$

- Data : a random sample $(\boldsymbol{X}_i, Y_i)_{i=1}^n$ with observed values $\mathcal{D}_n = (\boldsymbol{x}_i, y_i)_{i=1}^n$
- **Data-Scientist's role :** given the **data**, choose a **prediction function** h from a class $\mathcal{H}$ that attempts to "minimize" the prediction error for of all possible data (**risk**) $R(h)$, under a **loss** function ℓ measuring the error of predicting Y by $h(X)$.

  ↪ minimize the **empirical risk** (data-$\mathcal{D}_n$-driven) $R_n(h)$

  ↪ Minimizing $R_n(h)$ may require an optimization **algorithm** $\mathcal{A}$
- Data-Scientist's "**Toolbox**" : {Data, loss, hypothesis, algorithm}

Learning Framework for Classification

- The data are represented by a random pair $(\boldsymbol{X}, Y) \in \mathcal{X} \times \mathcal{Y}$ where $\boldsymbol{X}$ is a vector of descriptors for some variable of interest Y
- The objective is **Prediction**, i.e. to seek for a prediction function $h : \mathcal{X} \to \mathcal{Y}$ for which $\widehat{y} = h(\boldsymbol{x})$ is a good approximation of the true output y
- In a **classification** problem : typically $\boldsymbol{X} \in \mathcal{X} \subset \mathbb{R}^p$ and $Y \in \mathcal{Y} = \{0, 1\}, \{-1, +1\}$ (binary classification) or $\{1, \cdots, K\}$ (multiclass classification)

↪ We will mainly focus on parametric probabilistic models of the form

$$Y = h(X) + \epsilon, \epsilon \sim p_\theta$$

with the conditional distr. $P(Y|X, h)$ can be computed in terms of $P_\theta(Y - h(X))$. $P(X, Y|h) = P(Y|X, h)P(X|h)$

- Data : a random sample $(\boldsymbol{X}_i, Y_i)_{i=1}^n$ with observed values $\mathcal{D}_n = (\boldsymbol{x}_i, y_i)_{i=1}^n$
- **Data-Scientist's role :** given the **data**, choose a **prediction function** h from a class $\mathcal{H}$ that attempts to "minimize" the prediction error for of all possible data (**risk**) $R(h)$, under a **loss** function ℓ measuring the error of predicting Y by $h(X)$.

  ↪ minimize the **empirical risk** (data-$\mathcal{D}_n$-driven) $R_n(h)$

  ↪ Minimizing $R_n(h)$ may require an optimization **algorithm** $\mathcal{A}$

- Data-Scientist's "**Toolbox**" **:** {Data, loss, hypothesis, algorithm}

Prediction/decision function

Def. Classifier or classification rule

$$h : \mathcal{X} \to \mathcal{Y}$$
$$x \mapsto h(x)$$

is a decision/prediction function, parametric or not, linear or not, ...

Example : Linear predictors

$$h : \mathbb{R}^p \to \mathbb{R}$$
$$x \mapsto \langle x, \theta \rangle = \theta^T x$$

The **predicted** values of Y_i's for new covariates $X_i = x_i$s correspond to

$$\widehat{y}_i = h(x_i)$$

Example : Linear predictors (cont.) : $\widehat{y}_i = \langle x_i, \theta \rangle = \theta^T x_i$

Q : How good we are in prediction on a particular pair (x, y) ?

Prediction/decision function

Def. Classifier or classification rule

$$h: \mathcal{X} \to \mathcal{Y}$$
$$x \mapsto h(x)$$

is a decision/prediction function, parametric or not, linear or not, ...

Example : Linear predictors

$$h: \mathbb{R}^p \to \mathbb{R}$$
$$x \mapsto \langle x, \theta \rangle = \theta^T x$$

The **predicted** values of Y_i's for new covariates $X_i = x_i$s correspond to

$$\widehat{y}_i = h(x_i)$$

Example : Linear predictors (cont.) : $\widehat{y}_i = \langle x_i, \theta \rangle = \theta^T x_i$

Q : How good we are in prediction on a particular pair (x, y) ?

Prediction/decision function

Def. Classifier or classification rule

$$h: \mathcal{X} \to \mathcal{Y}$$
$$x \mapsto h(x)$$

is a decision/prediction function, parametric or not, linear or not, ...

Example : Linear predictors

$$h: \mathbb{R}^p \to \mathbb{R}$$
$$x \mapsto \langle x, \theta \rangle = \theta^T x$$

The **predicted** values of Y_i's for new covariates $X_i = x_i$s correspond to

$$\widehat{y}_i = h(x_i)$$

Example : Linear predictors (cont.) : $\widehat{y}_i = \langle x_i, \theta \rangle = \theta^T x_i$

Q : How good we are in prediction on a particular pair (x, y) ?

Prediction/decision function

Def. Classifier or classification rule

$$h: \mathcal{X} \to \mathcal{Y}$$
$$x \mapsto h(x)$$

is a decision/prediction function, parametric or not, linear or not, ...

Example : Linear predictors

$$h: \mathbb{R}^p \to \mathbb{R}$$
$$x \mapsto \langle x, \theta \rangle = \theta^T x$$

The **predicted** values of Y_i's for new covariates $X_i = x_i$s correspond to

$$\widehat{y}_i = h(x_i)$$

Example : Linear predictors (cont.) : $\widehat{y}_i = \langle x_i, \theta \rangle = \theta^T x_i$

Q : How good we are in prediction on a particular pair (x, y) ?

Loss

Def. Loss function

$$\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$
$$(y, h(x)) \mapsto \ell(y, h(x))$$

It measures how good we are on a particular (x, y) pair.
(We assume that the distribution of the test data is the same as for the training data).

Examples of loss functions in classification

- "0-1" loss : $\ell(y, h(x)) = \mathbb{1}_{h(x) \neq y}$
- logarithmic loss : $\ell(y, h_\theta(x)) = -\log(p_\theta(x, y))$

  Denoting $\ell(y, h(x)) = \phi(yh(x))$
- Hinge loss $\phi_{\text{hinge}}(u) = (1 - u)_+$
- Logistic loss $\phi_{\text{logistic}}(u) = \log(1 + \exp(-u))$
- Exponential loss $\phi_{\exp}(u) = \exp(-u)$

Loss

Def. Loss function

$$\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$
$$(y, h(x)) \mapsto \ell(y, h(x))$$

It measures how good we are on a particular (x, y) pair.
(We assume that the distribution of the test data is the same as for the training data).

Examples of loss functions in classification

- "0-1" loss : $\ell(y, h(x)) = \mathbb{1}_{h(x) \neq y}$
- logarithmic loss : $\ell(y, h_\theta(x)) = -\log(p_\theta(x, y))$

  Denoting $\ell(y, h(x)) = \phi(yh(x))$
- Hinge loss $\phi_{\text{hinge}}(u) = (1 - u)_+$
- Logistic loss $\phi_{\text{logistic}}(u) = \log(1 + \exp(-u))$
- Exponential loss $\phi_{\exp}(u) = \exp(-u)$

Loss

Def. Loss function

$$\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$
$$(y, h(x)) \mapsto \ell(y, h(x))$$

It measures how good we are on a particular (x, y) pair.
(We assume that the distribution of the test data is the same as for the training data).

Examples of loss functions in classification

- "0-1" loss : $\ell(y, h(x)) = \mathbb{1}_{h(x) \neq y}$
- logarithmic loss : $\ell(y, h_\theta(x)) = -\log(p_\theta(x, y))$

Denoting $\ell(y, h(x)) = \phi(yh(x))$

- Hinge loss $\phi_{\text{hinge}}(u) = (1 - u)_+$
- Logistic loss $\phi_{\text{logistic}}(u) = \log(1 + \exp(-u))$
- Exponential loss $\phi_{\exp}(u) = \exp(-u)$

Def. Loss function

$$\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$
$$(y, h(x)) \mapsto \ell(y, h(x))$$

It measures how good we are on a particular (x, y) pair.
(We assume that the distribution of the test data is the same as for the training data).

Examples of loss functions in classification

- "0-1" loss : $\ell(y, h(x)) = \mathbb{1}_{h(x) \neq y}$
- logarithmic loss : $\ell(y, h_\theta(x)) = -\log(p_\theta(x, y))$

  Denoting $\ell(y, h(x)) = \phi(yh(x))$
- Hinge loss $\phi_{\text{hinge}}(u) = (1 - u)_+$
- Logistic loss $\phi_{\text{logistic}}(u) = \log(1 + \exp(-u))$
- Exponential loss $\phi_{\exp}(u) = \exp(-u)$

Examples of loss functions in classification

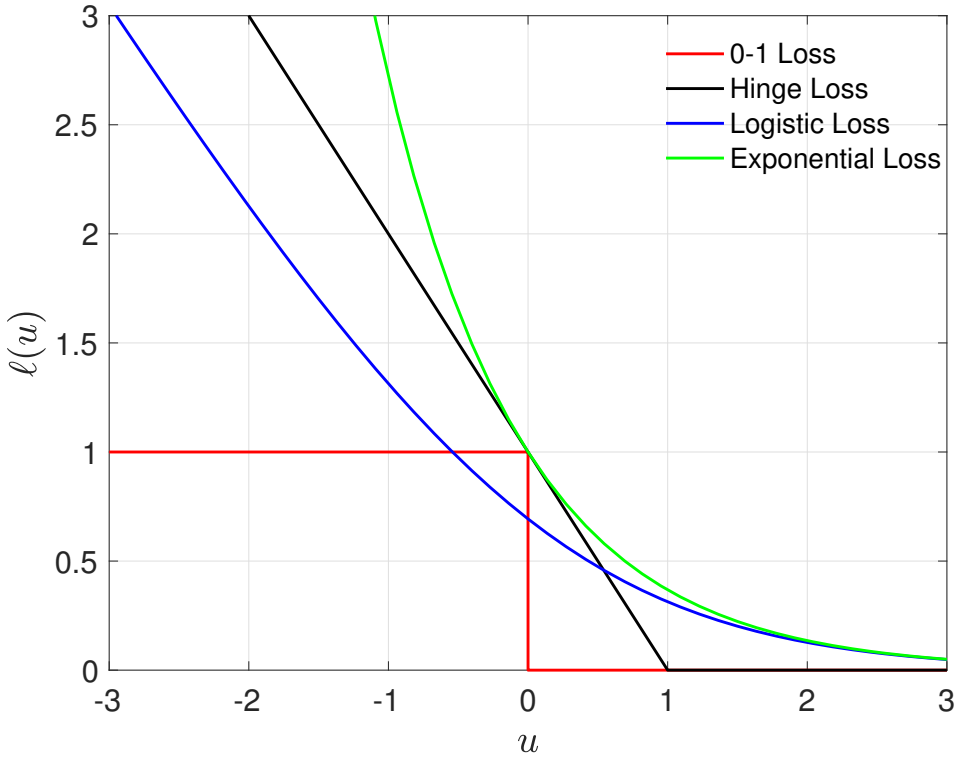


FIGURE – Some loss functions in classification : curves of $\ell(u)$ for $u = yh(x)$; $y \in \{-1, 1\}$. [plot_losses_classification.m]

For $y \in \{-1, 1\}$, with $u = yh(x)$:

- "0-1" loss : $\ell(u) = \mathbb{1}_{\text{sign}(u) \neq 1}$
- Hinge loss $\ell_{(\text{hinge}}(u) = (1 - u)_+$
- Logistic loss $\ell_{(\text{logistic}}(u) = \log(1 + \exp(-u))$
- Exponential loss $\ell_{(\text{exp}}(u) = \exp(-u)$

Risk

- **Risk** : the *Expected loss* :

$$R(h) = \mathbb{E}_P[\ell(Y, h(X))] = \int_{\mathcal{X}\times\mathcal{Y}} \ell(y, h(x)) dP(x, y)$$

↪ the error of approximating Y by model/hypothesis $h(X)$ as measured by a chosen loss function $\ell(Y, h(X))$ given the pair (X, Y) with (unknown) joint distribution P,

↪ prediction error : measures the generalization performance of the function h.

- **"0-1" Risk :** Under the "0-1"-loss $\ell(y, h(x)) = \mathbb{1}_{h(x)\neq y}$:

$$R(h) = \mathbb{E}_P[\mathbb{1}_{h(x)\neq y}] = \mathbb{P}(h(X) \neq Y). = \int_{\mathcal{X}\times\mathcal{Y}} \mathbb{1}_{h(x)\neq y} dP(x, y)$$

↪ This is the most used risk in classification

Q : what is the best function h ? or equivalently, when the risk $R(h)$ is optimal ?

Risk

- **Risk** : the *Expected loss* :

$$R(h) = \mathbb{E}_P[\ell(Y, h(X))] = \int_{\mathcal{X}\times\mathcal{Y}} \ell(y, h(x)) dP(x, y)$$

↪ the error of approximating Y by model/hypothesis $h(X)$ as measured by a chosen loss function $\ell(Y, h(X))$ given the pair (X, Y) with (unknown) joint distribution P,

↪ prediction error : measures the generalization performance of the function h.

- **"0-1" Risk :** Under the "0-1"-loss $\ell(y, h(x)) = \mathbb{1}_{h(x)\neq y}$:

$$R(h) = \mathbb{E}_P[\mathbb{1}_{h(x)\neq y}] = \mathbb{P}(h(X) \neq Y). = \int_{\mathcal{X}\times\mathcal{Y}} \mathbb{1}_{h(x)\neq y} dP(x, y)$$

↪ This is the most used risk in classification

Q : what is the best function h ? or equivalently, when the risk $R(h)$ is optimal ?

Risk

- **Risk** : the *Expected loss* :

$$R(h) = \mathbb{E}_P[\ell(Y, h(X))] = \int_{\mathcal{X}\times\mathcal{Y}} \ell(y, h(x)) dP(x, y)$$

↪ the error of approximating Y by model/hypothesis $h(X)$ as measured by a chosen loss function $\ell(Y, h(X))$ given the pair (X, Y) with (unknown) joint distribution P,

↪ prediction error : measures the generalization performance of the function h.

- **"0-1" Risk :** Under the "0-1"-loss $\ell(y, h(x)) = \mathbb{1}_{h(x)\neq y}$:

$$R(h) = \mathbb{E}_P[\mathbb{1}_{h(x)\neq y}] = \mathbb{P}(h(X) \neq Y). = \int_{\mathcal{X}\times\mathcal{Y}} \mathbb{1}_{h(x)\neq y} dP(x, y)$$

↪ This is the most used risk in classification

Q : what is the best function h ? or equivalently, when the risk $R(h)$ is optimal ?

Optimal prediction function I

Theorem (The Bayes classifier)

Under the (0-1)-loss, $\ell(Y, h(X)) = \mathbb{1}_{h(X)\neq Y}$, the classification function $h^*(x)$ minimizing the risk (the Bayes classifier)

$$R(h) = \mathbb{P}(Y \neq h(X)) = \int_{\mathcal{X}} \mathbb{P}(Y \neq h(X)|X = x) dP_X(x)$$

is given by

$$\forall x \in \mathcal{X}, \quad h^*(x) = \arg\max_{k \in \mathcal{Y}} \mathbb{P}(Y = k|X = x).$$

Def. Decision boundaries

The decision bounadry between each pair of classes k and ℓ, $(k, \ell) \in \mathcal{Y} \times \mathcal{Y}$ is defined by

$$\eta_{k,\ell}(x) = \{x : \mathbb{P}(Y = k|X = x) = \mathbb{P}(Y = \ell X = x)\}$$

Optimal prediction function II

Proof. Optimal classifier.

Given $X = x$, the conditional risk under the 0-1 loss is

$$\begin{aligned} r(h|X=x) &= \mathbb{E}_{Y|X=x}[\ell(Y,h(X))|X=x] = \mathbb{E}_{Y|X=x}[\mathbb{1}_{Y\neq h(X)}|X=x] \\ &= \mathbb{P}[Y \neq h(X)|X=x] \\ &= 1 - \mathbb{P}[Y = h(X)|X=x]. \end{aligned}$$

By noting that

$$\begin{aligned} \min_{k\in\mathcal{Y}} r(h|X=x) &= -1 + \max_{k\in\mathcal{Y}} \mathbb{P}(Y=k|X=x) \\ \arg\min_{k\in\mathcal{Y}} r(h|X=x) &= \arg\max_{k\in\mathcal{Y}} \mathbb{P}(Y=k|X=x) \end{aligned}$$

we see that $h^*(x) = \arg\max_{k\in\mathcal{Y}} \mathbb{P}(Y=k|X=x)$ achieves the minimized risk $r(h|X=x)$.

Then the risk $R(h^*) = \mathbb{E}_X[-1 + \max_{k\in\mathcal{Y}} \mathbb{P}(Y=k|X=x)]$ is Bayes. □

Goal : estimate h^*, knowing only the data sample $D_n = (X_i, Y_i)_{i=1}^n$ and loss ℓ.

Empirical Risk Minimization & MLE I

- Then *Expected loss* $R(h)$ depends on the joint distribution P of the pair (X, Y). In real situations P is in unknown, as we only have a sample $D_n = (X_i, Y_i)_{1 \leq i \leq n}$,

↪ We attempt to minimize the **Empirical Risk**

$$R_n(h) = \frac{1}{n} \sum_{i=1}^{n} l(Y_i, h(X_i))$$

to estimate h^* (within a family $\mathcal{H}$)

↪ ERM : $\widehat{h}_n = \arg\min_{h \in \mathcal{H}} R_n(h)$ is the **ERM** of h

- 0-1 Risk : Under the 0-1 loss (standard in classification) : $\ell_{0-1}(y, h(x)) = \mathbb{1}_{y \neq h(x)}$, the empirical 0-1 risk is

$$R_n(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{Y_i \neq h(X_i)}$$

ERM and MLE : maximum likelihood risks :

- MLE (density estimation framework) : In generative learning (eg. discriminant analysis), we are interested in modeling the joint distribution $P(X, Y)$ (then the conditional $P(Y|X)$ is obtained by Bayes' Theorem). We seek for an esitmator of the parameters θ of the joint distribution $p_\theta(x, y)$.

Empirical Risk Minimization & MLE II

- Consider the log-loss : $\ell(y, h_\theta(x)) = -\log(p_\theta(x, y))$. We therefore have the log-likehood risks

$$R(\theta) = -\mathbb{E}[\log p_\theta(X, Y)]$$

and

$$R_n(\theta) = -\frac{1}{n}\sum_{i=1}^{n} \log p_\theta(x_i, y_i).$$

- For an i.i.d sample $\{(x_i, y_i)_{i=1}^n\}$, the conditional log-likelihood function of θ is :

$$\log L(\theta) = \sum_{i=1}^{n} \log p_\theta(x_i, y_i)$$

Then

$$R_n(\theta) = -\frac{1}{n}\sum_{i=1}^{n} \log p_\theta(x_i, y_i) = -\frac{1}{n}\log L(\theta)$$

$\hookrightarrow$ With the log-loss, ERM coincides with MLE.

Gaussian Discriminant Analysis

Gaussian Discriminant Analysis

- Generative model

$$p(Y_i, \boldsymbol{X}_i; \boldsymbol{\theta}) = \mathbb{P}(Y_i) p(\boldsymbol{X}_i | Y_i; \boldsymbol{\theta})$$

- $\mathbb{P}(Y_i = k) = w_k$ the prior probability of class k,
- $p(\boldsymbol{X}_i = \boldsymbol{x}_i | Y_i = k; \boldsymbol{\theta}_k) = \phi_p(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ the conditional density of class k is the Gaussian p.d.f in $\mathbb{R}^p$ with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$, defined as

$$\phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = (2\pi)^{-\frac{d}{2}} |\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top {\boldsymbol{\Sigma}_k}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)\}$$

- $\boldsymbol{\theta} = (w_1, \ldots, w_K, \boldsymbol{\mu}_1^\top, \ldots, \boldsymbol{\mu}_K^\top, \text{vech}^\top(\boldsymbol{\Sigma}_1), \ldots, \text{vech}^\top(\boldsymbol{\Sigma}_K))^\top$
- **Data Generating Process** under Gaussian Discriminant Analysis : Given $\boldsymbol{\theta}$:

i) Sample a class label Y_i given the class weights $\boldsymbol{w} = \{w_1, \cdots, w_K\}$,

$$Y_i | w_1, \ldots, w_K \sim \text{Categorical}(1; w_1, \ldots, w_K).$$

ii) Sample an observation $\boldsymbol{X}_i$ from the conditional distribution $f(\cdot; \boldsymbol{\theta}_k)$:

$$\boldsymbol{X}_i | Y_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \sim \mathcal{N}_d(\cdot; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Gaussian Discriminant Analysis

- Generative model

$$p(Y_i, \boldsymbol{X}_i; \boldsymbol{\theta}) = \mathbb{P}(Y_i) p(\boldsymbol{X}_i | Y_i; \boldsymbol{\theta})$$

- $\mathbb{P}(Y_i = k) = w_k$ the prior probability of class k,
- $p(\boldsymbol{X}_i = \boldsymbol{x}_i | Y_i = k; \boldsymbol{\theta}_k) = \phi_p(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ the conditional density of class k is the Gaussian p.d.f in $\mathbb{R}^p$ with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$, defined as

$$\phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = (2\pi)^{-\frac{d}{2}} |\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top {\boldsymbol{\Sigma}_k}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)\}$$

- $\boldsymbol{\theta} = (w_1, \ldots, w_K, \boldsymbol{\mu}_1^\top, \ldots, \boldsymbol{\mu}_K^\top, \text{vech}^\top(\boldsymbol{\Sigma}_1), \ldots, \text{vech}^\top(\boldsymbol{\Sigma}_K))^\top$
- **Data Generating Process** under Gaussian Discriminant Analysis : Given $\boldsymbol{\theta}$:

i) Sample a class label Y_i given the class weights $\boldsymbol{w} = \{w_1, \cdots, w_K\}$,

$$Y_i | w_1, \ldots, w_K \sim \text{Categorical}(1; w_1, \ldots, w_K).$$

ii) Sample an observation X_i from the conditional distribution $f(\cdot; \theta_k)$:

$$\boldsymbol{X}_i | Y_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \sim \mathcal{N}_d(\cdot; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Gaussian Discriminant Analysis

- Generative model

$$p(Y_i, \boldsymbol{X}_i; \boldsymbol{\theta}) = \mathbb{P}(Y_i) p(\boldsymbol{X}_i | Y_i; \boldsymbol{\theta})$$

- $\mathbb{P}(Y_i = k) = w_k$ the prior probability of class k,
- $p(\boldsymbol{X}_i = \boldsymbol{x}_i | Y_i = k; \boldsymbol{\theta}_k) = \phi_p(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ the conditional density of class k is the Gaussian p.d.f in $\mathbb{R}^p$ with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$, defined as

$$\phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = (2\pi)^{-\frac{d}{2}} |\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top {\boldsymbol{\Sigma}_k}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)\}$$

- $\boldsymbol{\theta} = (w_1, \ldots, w_K, \boldsymbol{\mu}_1^\top, \ldots, \boldsymbol{\mu}_K^\top, \text{vech}^\top(\boldsymbol{\Sigma}_1), \ldots, \text{vech}^\top(\boldsymbol{\Sigma}_K))^\top$
- **Data Generating Process** under Gaussian Discriminant Analysis : Given θ :

i) Sample a class label Y_i given the class weights $\boldsymbol{w} = \{w_1, \cdots, w_K\}$,

$$Y_i | w_1, \ldots, w_K \sim \text{Categorical}(1; w_1, \ldots, w_K),$$

ii) Sample an observation X_i from the conditional distribution $f(\cdot; \theta_k)$:

$$\boldsymbol{X}_i | Y_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \sim \mathcal{N}_d(\cdot; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Gaussian Discriminant Analysis

- Generative model

$$p(Y_i, \boldsymbol{X}_i; \boldsymbol{\theta}) = \mathbb{P}(Y_i) p(\boldsymbol{X}_i | Y_i; \boldsymbol{\theta})$$

- $\mathbb{P}(Y_i = k) = w_k$ the prior probability of class k,
- $p(\boldsymbol{X}_i = \boldsymbol{x}_i | Y_i = k; \boldsymbol{\theta}_k) = \phi_p(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ the conditional density of class k is the Gaussian p.d.f in $\mathbb{R}^p$ with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$, defined as

$$\phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = (2\pi)^{-\frac{d}{2}} |\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top {\boldsymbol{\Sigma}_k}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)\}$$

- $\boldsymbol{\theta} = (w_1, \ldots, w_K, \boldsymbol{\mu}_1^\top, \ldots, \boldsymbol{\mu}_K^\top, \text{vech}^\top(\boldsymbol{\Sigma}_1), \ldots, \text{vech}^\top(\boldsymbol{\Sigma}_K))^\top$
- **Data Generating Process** under Gaussian Discriminant Analysis : Given θ :

i) Sample a class label Y_i given the class weights $w = \{w_1, \cdots, w_K\}$,

$$Y_i | w_1, \ldots, w_K \sim \text{Categorical}(1; w_1, \ldots, w_K).$$

ii) Sample an observation X_i from the conditional distribution $f(\cdot; \theta_k)$:

$$X_i | Y_i = k, \mu_k, \Sigma_k \sim \mathcal{N}_d(\cdot; \mu_k, \Sigma_k)$$

Gaussian Discriminant Analysis

- Generative model

$$p(Y_i, \boldsymbol{X}_i; \boldsymbol{\theta}) = \mathbb{P}(Y_i) p(\boldsymbol{X}_i | Y_i; \boldsymbol{\theta})$$

- $\mathbb{P}(Y_i = k) = w_k$ the prior probability of class k,
- $p(\boldsymbol{X}_i = \boldsymbol{x}_i | Y_i = k; \boldsymbol{\theta}_k) = \phi_p(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ the conditional density of class k is the Gaussian p.d.f in $\mathbb{R}^p$ with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$, defined as

$$\phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = (2\pi)^{-\frac{d}{2}} |\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top {\boldsymbol{\Sigma}_k}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)\}$$

- $\boldsymbol{\theta} = (w_1, \ldots, w_K, \boldsymbol{\mu}_1^\top, \ldots, \boldsymbol{\mu}_K^\top, \text{vech}^\top(\boldsymbol{\Sigma}_1), \ldots, \text{vech}^\top(\boldsymbol{\Sigma}_K))^\top$
- **Data Generating Process** under Gaussian Discriminant Analysis : Given $\boldsymbol{\theta}$:

  i) Sample a class label Y_i given the class weights $\boldsymbol{w} = \{w_1, \cdots, w_K\}$,

$$Y_i | w_1, \ldots, w_K \quad \sim \quad \text{Categorical}(1; w_1, \ldots, w_K),$$

  ii) Sample an observation $\boldsymbol{X}_i$ from the conditional distribution $f(\cdot; \boldsymbol{\theta}_k)$:

$$\boldsymbol{X}_i | Y_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \quad \sim \quad \mathcal{N}_d(\cdot; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Gaussian Discriminant Analysis

- Generative model

$$p(Y_i, \boldsymbol{X}_i; \boldsymbol{\theta}) = \mathbb{P}(Y_i) p(\boldsymbol{X}_i | Y_i; \boldsymbol{\theta})$$

- $\mathbb{P}(Y_i = k) = w_k$ the prior probability of class k,
- $p(\boldsymbol{X}_i = \boldsymbol{x}_i | Y_i = k; \boldsymbol{\theta}_k) = \phi_p(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ the conditional density of class k is the Gaussian p.d.f in $\mathbb{R}^p$ with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$, defined as

$$\phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = (2\pi)^{-\frac{d}{2}} |\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top {\boldsymbol{\Sigma}_k}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)\}$$

- $\boldsymbol{\theta} = (w_1, \ldots, w_K, \boldsymbol{\mu}_1^\top, \ldots, \boldsymbol{\mu}_K^\top, \text{vech}^\top(\boldsymbol{\Sigma}_1), \ldots, \text{vech}^\top(\boldsymbol{\Sigma}_K))^\top$
- **Data Generating Process** under Gaussian Discriminant Analysis : Given $\boldsymbol{\theta}$:
  - i) Sample a class label Y_i given the class weights $\boldsymbol{w} = \{w_1, \cdots, w_K\}$,

$$Y_i | w_1, \ldots, w_K \quad \sim \quad \text{Categorical}(1; w_1, \ldots, w_K),$$

  - ii) Sample an observation $\boldsymbol{X}_i$ from the conditional distribution $f(\cdot; \boldsymbol{\theta}_k)$:

$$\boldsymbol{X}_i | Y_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \quad \sim \quad \mathcal{N}_d(\cdot; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Gaussian Discriminant Analysis

- Generative model

$$p(Y_i, \boldsymbol{X}_i; \boldsymbol{\theta}) = \mathbb{P}(Y_i) p(\boldsymbol{X}_i | Y_i; \boldsymbol{\theta})$$

- $\mathbb{P}(Y_i = k) = w_k$ the prior probability of class k,
- $p(\boldsymbol{X}_i = \boldsymbol{x}_i | Y_i = k; \boldsymbol{\theta}_k) = \phi_p(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ the conditional density of class k is the Gaussian p.d.f in $\mathbb{R}^p$ with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$, defined as

$$\phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = (2\pi)^{-\frac{d}{2}} |\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top {\boldsymbol{\Sigma}_k}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)\}$$

- $\boldsymbol{\theta} = (w_1, \ldots, w_K, \boldsymbol{\mu}_1^\top, \ldots, \boldsymbol{\mu}_K^\top, \text{vech}^\top(\boldsymbol{\Sigma}_1), \ldots, \text{vech}^\top(\boldsymbol{\Sigma}_K))^\top$
- **Data Generating Process** under Gaussian Discriminant Analysis : Given $\boldsymbol{\theta}$:

  i) Sample a class label Y_i given the class weights $\boldsymbol{w} = \{w_1, \cdots, w_K\}$,

$$Y_i | w_1, \ldots, w_K \quad \sim \quad \text{Categorical}(1; w_1, \ldots, w_K),$$

  ii) Sample an observation $\boldsymbol{X}_i$ from the conditional distribution $f(\cdot; \boldsymbol{\theta}_k)$:

$$\boldsymbol{X}_i | Y_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \quad \sim \quad \mathcal{N}_d(\cdot; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Gaussian Discriminant Analysis

- Generative model

$$p(Y_i, \boldsymbol{X}_i; \boldsymbol{\theta}) = \mathbb{P}(Y_i) p(\boldsymbol{X}_i | Y_i; \boldsymbol{\theta})$$

- $\mathbb{P}(Y_i = k) = w_k$ the prior probability of class k,
- $p(\boldsymbol{X}_i = \boldsymbol{x}_i | Y_i = k; \boldsymbol{\theta}_k) = \phi_p(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ the conditional density of class k is the Gaussian p.d.f in $\mathbb{R}^p$ with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$, defined as

$$\phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = (2\pi)^{-\frac{d}{2}} |\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top {\boldsymbol{\Sigma}_k}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)\}$$

- $\boldsymbol{\theta} = (w_1, \ldots, w_K, \boldsymbol{\mu}_1^\top, \ldots, \boldsymbol{\mu}_K^\top, \text{vech}^\top(\boldsymbol{\Sigma}_1), \ldots, \text{vech}^\top(\boldsymbol{\Sigma}_K))^\top$
- **Data Generating Process** under Gaussian Discriminant Analysis : Given $\boldsymbol{\theta}$:

  i) Sample a class label Y_i given the class weights $\boldsymbol{w} = \{w_1, \cdots, w_K\}$,

$$Y_i | w_1, \ldots, w_K \quad \sim \quad \text{Categorical}(1; w_1, \ldots, w_K),$$

  ii) Sample an observation $\boldsymbol{X}_i$ from the conditional distribution $f(\cdot; \boldsymbol{\theta}_k)$:

$$\boldsymbol{X}_i | Y_i = k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \quad \sim \quad \mathcal{N}_d(\cdot; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Classification rule & Decision Boundaries

Def. Classification rule

The Bayes' decision rule $h(x)$ defined as

$$\widehat{y}_i = h_\theta(x) = \arg\max_{k \in \mathcal{Y}} \mathbb{P}(Y_i = k | \boldsymbol{X}_i = \boldsymbol{x}_i; \boldsymbol{\theta})$$

with

$$\begin{aligned} \mathbb{P}(Y_i = k | \boldsymbol{X}_i = \boldsymbol{x}_i; \boldsymbol{\theta}) &\propto \mathbb{P}(Y_i = k) f(\boldsymbol{X}_i = \boldsymbol{x}_i | Y_i = k; \boldsymbol{\theta}_k) \\ &\propto w_k \phi_p(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \end{aligned}$$

Def. Decision boundaries

The decision bounadry between each pair of classes k and ℓ, $(k, \ell) \in \mathcal{Y} \times \mathcal{Y}$ is defined by

$$\begin{aligned} \eta_{k,\ell}(\boldsymbol{x}) &= \{\boldsymbol{x} : \mathbb{P}(Y = k | \boldsymbol{X} = \boldsymbol{x}; \boldsymbol{\theta}) = \mathbb{P}(Y = \ell | \boldsymbol{X} = \boldsymbol{x}; \boldsymbol{\theta})\} \\ &= \{\boldsymbol{x} : w_k \phi_p(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = w_\ell \phi_p(\boldsymbol{x}_i; \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)\} \end{aligned}$$

Linear Discriminant Analysis

Linear Discriminant Analysis (**LDA**) arises when we assume that all the classes have a common covariance matrix $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}\ \forall k = 1, \ldots, K$.

Consider GDA with $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}\ \forall k = 1, \ldots, K$, then the decision boundary between two classes k and ℓ is

$$
\begin{aligned}
\eta_{k,\ell}(\boldsymbol{x}; \boldsymbol{\theta}) &= \{\boldsymbol{x} : \mathbb{P}(Y = k|\boldsymbol{x}; \boldsymbol{\theta}) = \mathbb{P}(Y = k|\boldsymbol{x}; \boldsymbol{\theta})\} \\
&= \left\{\boldsymbol{x} : \log \frac{\mathbb{P}(Y = k|\boldsymbol{x}; \boldsymbol{\theta})}{\mathbb{P}(Y = \ell|\boldsymbol{x}; \boldsymbol{\theta})} = 0\right\} \\
&= \left\{\boldsymbol{x} : \log \frac{w_k}{w_\ell} + \log \frac{\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma})}{\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma})} = 0\right\} \\
&= \left\{\boldsymbol{x} : \underbrace{\log \frac{w_k}{w_\ell} - \frac{1}{2}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_\ell)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_\ell)}_{\alpha} + \underbrace{(\boldsymbol{\mu}_k - \boldsymbol{\mu}_\ell)^T \boldsymbol{\Sigma}^{-1}\boldsymbol{x}}_{\boldsymbol{\beta}^T \boldsymbol{x}} = 0\right\} \\
&= \left\{\boldsymbol{x} : \alpha + \boldsymbol{\beta}^T \boldsymbol{x} = 0\right\},
\end{aligned}
$$

$\hookrightarrow$ the classes are separated by hyperplane in the input space.

Linear Discriminant Analysis

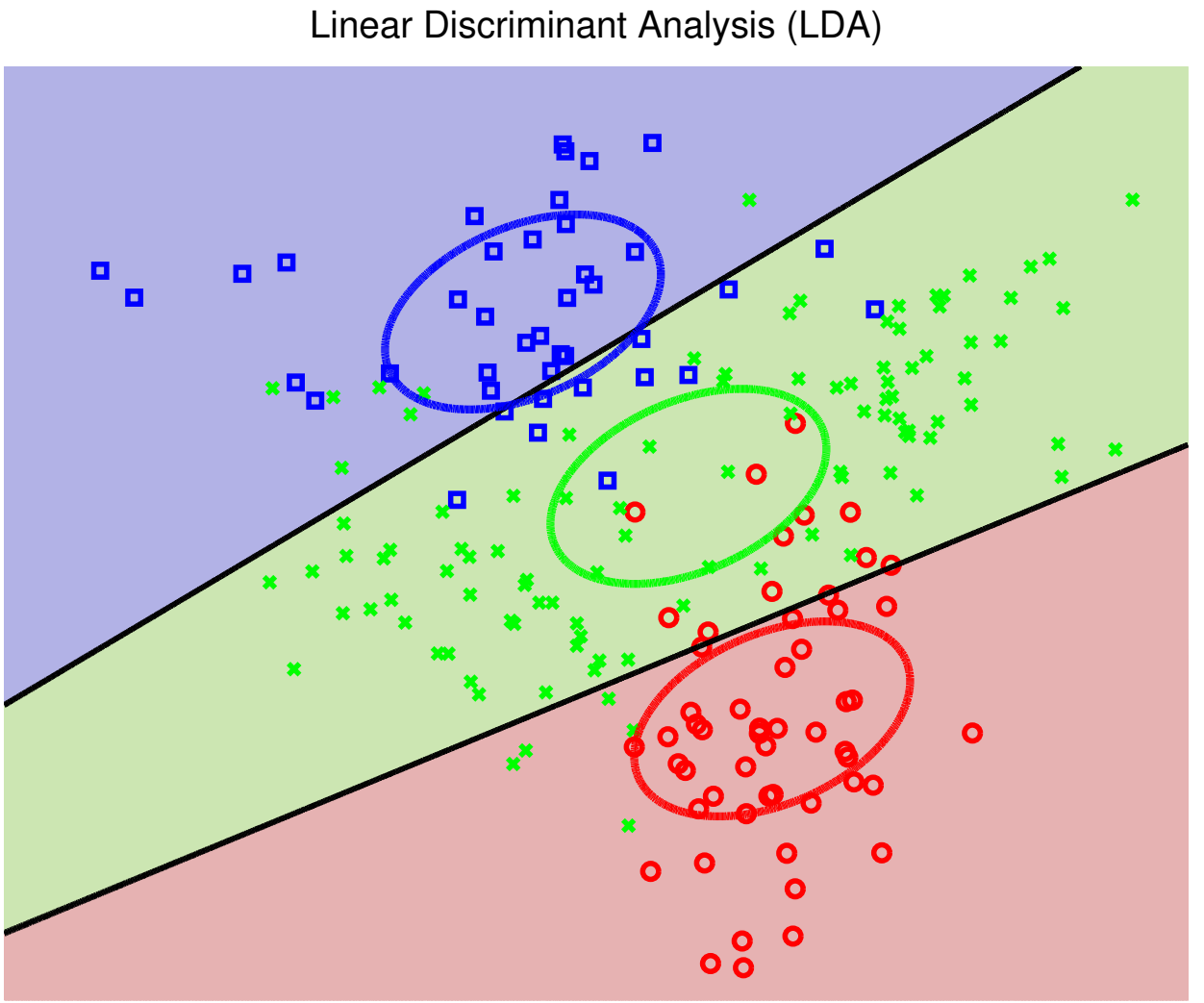
Linear Discriminant Analysis (**LDA**) arises when we assume that all the classes have a common covariance matrix $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}\ \forall k = 1, \ldots, K$.

Consider GDA with $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}\ \forall k = 1, \ldots, K$, then the decision boundary between two classes k and ℓ is

$$
\begin{aligned}
\eta_{k,\ell}(\boldsymbol{x}; \boldsymbol{\theta}) &= \{\boldsymbol{x} : \mathbb{P}(Y = k|\boldsymbol{x}; \boldsymbol{\theta}) = \mathbb{P}(Y = k|\boldsymbol{x}; \boldsymbol{\theta})\} \\
&= \left\{\boldsymbol{x} : \log \frac{\mathbb{P}(Y = k|\boldsymbol{x}; \boldsymbol{\theta})}{\mathbb{P}(Y = \ell|\boldsymbol{x}; \boldsymbol{\theta})} = 0\right\} \\
&= \left\{\boldsymbol{x} : \log \frac{w_k}{w_\ell} + \log \frac{\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma})}{\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma})} = 0\right\} \\
&= \left\{\boldsymbol{x} : \underbrace{\log \frac{w_k}{w_\ell} - \frac{1}{2}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_\ell)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_\ell)}_{\alpha} + \underbrace{(\boldsymbol{\mu}_k - \boldsymbol{\mu}_\ell)^T \boldsymbol{\Sigma}^{-1}\boldsymbol{x}}_{\boldsymbol{\beta}^T\boldsymbol{x}} = 0\right\} \\
&= \left\{\boldsymbol{x} : \alpha + \boldsymbol{\beta}^T \boldsymbol{x} = 0\right\},
\end{aligned}
$$

$\hookrightarrow$ the classes are separated by hyperplane in the input space.

Linear Discriminant Analysis



Logistic/Softmax Regression vs LDA

Binary classification

Consider LDA with $\mathcal{Y} = \{0, 1\}$, then the posterior is logistic. $\forall k \in \mathcal{Y}$, we have

$$\mathbb{P}(Y_i = 1|\boldsymbol{x}_i; \boldsymbol{\theta}) = \frac{\exp\{\alpha + \boldsymbol{\beta}^T \boldsymbol{x}_i\}}{1 + \exp\{\alpha + \boldsymbol{\beta}^T \boldsymbol{x}_i\}}$$

with $\alpha = \log \frac{w_1}{w_0} - \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0)$ and $\boldsymbol{\beta} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$

Multiclass classification

Consider LDA with $\mathcal{Y} = \{1, \ldots, K\}$, then the posterior is softmax. $\forall k \in \mathcal{Y}$, we have :

$$\mathbb{P}(Y_i = k|\boldsymbol{x}_i; \boldsymbol{\theta}) = \frac{\exp\{\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\}}{1 + \sum_{l=1}^{K-1} \exp\{\alpha_l + \boldsymbol{\beta}_l^T \boldsymbol{x}_i\}}$$

with $\alpha_k = \log \frac{w_k}{w_K} - \frac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_K)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_K)$ and $\beta_k = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_K)$

$\hookrightarrow$ Logistic/Softmax Regression and LDA are almost the same

$\hookrightarrow$ They lead to linear decision boundaries

Logistic/Softmax Regression vs LDA

Binary classification

Consider LDA with $\mathcal{Y} = \{0, 1\}$, then the posterior is logistic. $\forall k \in \mathcal{Y}$, we have

$$\mathbb{P}(Y_i = 1|\boldsymbol{x}_i; \boldsymbol{\theta}) = \frac{\exp\{\alpha + \boldsymbol{\beta}^T \boldsymbol{x}_i\}}{1 + \exp\{\alpha + \boldsymbol{\beta}^T \boldsymbol{x}_i\}}$$

with $\alpha = \log \frac{w_1}{w_0} - \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0)$ and $\boldsymbol{\beta} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$

Multiclass classification

Consider LDA with $\mathcal{Y} = \{1, \ldots, K\}$, then the posterior is softmax. $\forall k \in \mathcal{Y}$, we have :

$$\mathbb{P}(Y_i = k|\boldsymbol{x}_i; \boldsymbol{\theta}) = \frac{\exp\{\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\}}{1 + \sum_{l=1}^{K-1} \exp\{\alpha_l + \boldsymbol{\beta}_l^T \boldsymbol{x}_i\}}$$

with $\alpha_k = \log \frac{w_k}{w_K} - \frac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_K)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_K)$ and $\boldsymbol{\beta}_k = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_K)$

$\hookrightarrow$ Logistic/Softmax Regression and LDA are almost the same

$\hookrightarrow$ They lead to linear decision boundaries

Logistic/Softmax Regression vs LDA

Binary classification

Consider LDA with $\mathcal{Y} = \{0, 1\}$, then the posterior is logistic. $\forall k \in \mathcal{Y}$, we have

$$\mathbb{P}(Y_i = 1|\boldsymbol{x}_i; \boldsymbol{\theta}) = \frac{\exp\{\alpha + \boldsymbol{\beta}^T \boldsymbol{x}_i\}}{1 + \exp\{\alpha + \boldsymbol{\beta}^T \boldsymbol{x}_i\}}$$

with $\alpha = \log \frac{w_1}{w_0} - \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0)$ and $\boldsymbol{\beta} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$

Multiclass classification

Consider LDA with $\mathcal{Y} = \{1, \ldots, K\}$, then the posterior is softmax. $\forall k \in \mathcal{Y}$, we have :

$$\mathbb{P}(Y_i = k|\boldsymbol{x}_i; \boldsymbol{\theta}) = \frac{\exp\{\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\}}{1 + \sum_{l=1}^{K-1} \exp\{\alpha_l + \boldsymbol{\beta}_l^T \boldsymbol{x}_i\}}$$

with $\alpha_k = \log \frac{w_k}{w_K} - \frac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_K)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_K)$ and $\boldsymbol{\beta}_k = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_K)$

$\hookrightarrow$ Logistic/Softmax Regression and LDA are almost the same

$\hookrightarrow$ They lead to linear decision boundaries

Proof. Case of binary classification.

Consider LDA with $\mathcal{Y} = \{0, 1\}$, then the posterior is logistic.

$$
\begin{aligned}
\mathbb{P}(Y_i = 1|\boldsymbol{x}_i; \boldsymbol{\theta}) &= \frac{\mathbb{P}(Y_i = 1) f(\boldsymbol{X}_i = \boldsymbol{x}_i|Y_i = 1; \boldsymbol{\theta})}{\mathbb{P}(Y_i = 0) f(\boldsymbol{X}_i = \boldsymbol{x}_i|Y_i = 0; \boldsymbol{\theta}) + \mathbb{P}(Y_i = 1) f(\boldsymbol{X}_i = \boldsymbol{x}_i|Y_i = 1; \boldsymbol{\theta})} \\
&= \frac{w_1 \phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma})}{w_0 \phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_0, \boldsymbol{\Sigma}) + w_1 \phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma})} \\
&= \frac{w_1 \phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}) / w_0 \phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_0, \boldsymbol{\Sigma})}{1 + w_1 \phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}) / w_0 \phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_0, \boldsymbol{\Sigma})} \\
&= \frac{\exp\{\log \frac{w_1}{w_0} - \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x}_i\}}{1 + \exp\{\log \frac{w_1}{w_0} - \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x}_i\}} \\
&= \frac{\exp\{\alpha + \boldsymbol{\beta}^T \boldsymbol{x}_i\}}{1 + \exp\{\alpha + \boldsymbol{\beta}^T \boldsymbol{x}_i\}}
\end{aligned}
$$

with $\alpha = \log \frac{w_1}{w_0} - \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_0)$ and $\boldsymbol{\beta} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$ □

Proof. Case of multiclass classification.

Consider LDA with $\mathcal{Y} = \{1, \ldots, K\}$, then the posterior is softmax : $\forall k \in \mathcal{Y}$, we have

$$\mathbb{P}(Y_i = k|\boldsymbol{x}_i; \boldsymbol{\theta}) = \frac{\mathbb{P}(Y_i = k) f(\boldsymbol{X}_i = \boldsymbol{x}_i|Y_i = k; \boldsymbol{\theta})}{\sum_{l=1}^{K} \mathbb{P}(Y_i = l) f(\boldsymbol{X}_i = \boldsymbol{x}_i|Y_i = l; \boldsymbol{\theta})} = \frac{w_k \phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma})}{\sum_{l=1}^{K} w_l \phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_l, \boldsymbol{\Sigma})}$$

$$= \frac{w_k \phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}) / w_K \phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_K, \boldsymbol{\Sigma})}{1 + \sum_{l=1}^{K-1} w_l \phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}) / w_K \phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_K, \boldsymbol{\Sigma})}$$

$$= \frac{\exp\{\log \frac{w_k}{w_K} - \frac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_K)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_K) + (\boldsymbol{\mu}_k - \boldsymbol{\mu}_K)^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x}_i\}}{1 + \sum_{l=1}^{K-1} \exp\{\log \frac{w_k}{w_K} - \frac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_K)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_K) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x}_i\}}$$

$$= \frac{\exp\{\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\}}{1 + \sum_{l=1}^{K-1} \exp\{\alpha_l + \boldsymbol{\beta}_l^T \boldsymbol{x}_i\}}$$

with $\alpha_k = \log \frac{w_k}{w_K} - \frac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_K)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_K)$ and $\boldsymbol{\beta}_k = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_K)$ □

Quadratic Discriminant Analysis

Quadratic Discriminant Analysis

Quadratic Discriminant Analysis (QDA) corresponds to allowing a different covariance matrix for each class. The QDA decision boundaries are quadratic functions in $\boldsymbol{x}$:

Proof.

Consider GDA with different $\{\boldsymbol{\Sigma}_k\}_{k=1}^K$, then the decision boundary between two classes k and ℓ is

$$\begin{aligned}
\eta_{k,\ell}(\boldsymbol{x};\boldsymbol{\theta}) &= \left\{\boldsymbol{x} : \log \frac{\mathbb{P}(Y=k|\boldsymbol{x};\boldsymbol{\theta})}{\mathbb{P}(Y=\ell|\boldsymbol{x};\boldsymbol{\theta})} = 0\right\} \\
&= \left\{\boldsymbol{x} : \log \frac{w_k}{w_\ell} + \log \frac{\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_\ell,\boldsymbol{\Sigma}_\ell)} = 0\right\} \\
&\quad \left\{\boldsymbol{x} : \log \frac{w_k}{w_\ell} - \frac{1}{2}\log \frac{|\boldsymbol{\Sigma}_k|}{|\boldsymbol{\Sigma}_\ell|}\right. \\
&\quad \left. -\frac{1}{2}\left[(\boldsymbol{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k) - (\boldsymbol{x}-\boldsymbol{\mu}_\ell)^T\boldsymbol{\Sigma}_\ell^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_\ell)\right] = 0\right\}.
\end{aligned}$$

$\hookrightarrow$ We then get quadratic discriminant functions in the input space. □

Quadratic Discriminant Analysis

Quadratic Discriminant Analysis (QDA) corresponds to allowing a different covariance matrix for each class. The QDA decision boundaries are quadratic functions in $\boldsymbol{x}$:

Proof.

Consider GDA with different $\{\boldsymbol{\Sigma}_k\}_{k=1}^K$, then the decision boundary between two classes k and ℓ is

$$\begin{aligned}
\eta_{k,\ell}(\boldsymbol{x};\boldsymbol{\theta}) &= \left\{\boldsymbol{x} : \log \frac{\mathbb{P}(Y=k|\boldsymbol{x};\boldsymbol{\theta})}{\mathbb{P}(Y=\ell|\boldsymbol{x};\boldsymbol{\theta})} = 0\right\} \\
&= \left\{\boldsymbol{x} : \log \frac{w_k}{w_\ell} + \log \frac{\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_\ell,\boldsymbol{\Sigma}_\ell)} = 0\right\} \\
&\quad \left\{\boldsymbol{x} : \log \frac{w_k}{w_\ell} - \frac{1}{2}\log \frac{|\boldsymbol{\Sigma}_k|}{|\boldsymbol{\Sigma}_\ell|}\right. \\
&\quad \left. -\frac{1}{2}\left[(\boldsymbol{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k) - (\boldsymbol{x}-\boldsymbol{\mu}_\ell)^T\boldsymbol{\Sigma}_\ell^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_\ell)\right] = 0\right\}.
\end{aligned}$$

$\hookrightarrow$ We then get quadratic discriminant functions in the input space. □

Maximum-Likelihood Estimation

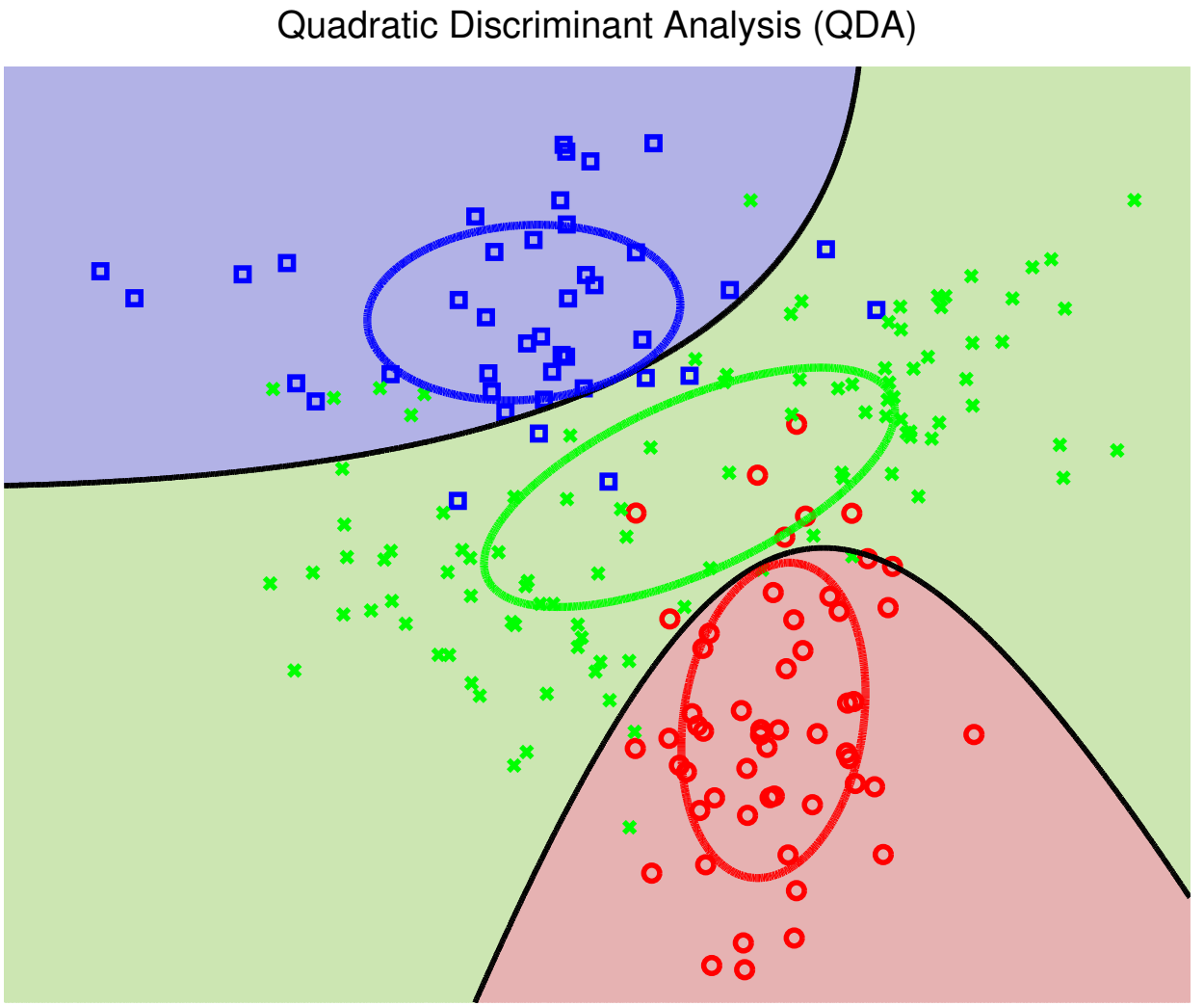
- Maximize the joint log-likelihood function : $\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}\in\Theta} L(\boldsymbol{\theta})$ with $L(\boldsymbol{\theta}) = \log\prod_{i=1}^{n} p(Y_i = y_i, \boldsymbol{X}_i = \boldsymbol{x}_i; \boldsymbol{\theta}) = \log\prod_{i=1}^{n} [\mathbb{P}(Y_i = y_i) f(\boldsymbol{X}_i = \boldsymbol{x}_i | Y_i = y_i; \boldsymbol{\theta})]$.
- Let $y_{ik} = \mathbb{1}_{y_i \neq k}$ the binary indicator variable. Then we have

$$
\begin{aligned}
L(\boldsymbol{\theta}) &= \log \prod_{i=1}^{n}\prod_{k=1}^{K} [\mathbb{P}(Y_i = k) f(\boldsymbol{X}_i = \boldsymbol{x}_i | Y_i = k; \boldsymbol{\theta}_k)]^{y_{ik}} \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik} \log w_k + \sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik} \log \phi_d(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik} \log w_k \\
&+ \sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik} \left[-\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top {\boldsymbol{\Sigma}_k}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k) \right]
\end{aligned}
$$

- A concave function in $\boldsymbol{\theta}$ $\hookrightarrow$ Global maximization is guaranteed

$\hookrightarrow$ A closed-form solution

Quadratic Discriminant Analysis



Maximum-Likelihood Estimation I

Let us denote by

$$L(w_k) = \sum_{i=1}^{n} y_{ik} \log w_k,$$

$$L(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = -\frac{1}{2} \sum_{i=1}^{n} y_{ik} \left[\log |\boldsymbol{\Sigma}_k| + (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top {\boldsymbol{\Sigma}_k}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k) \right],$$

Then

$$L(\boldsymbol{\theta}) = \sum_{k=1}^{K} \left[L(w_k) + L(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right] + constant,$$

- $\widehat{w}_k = \arg\max_{w_k} L(w_k)$ subject to $\sum_{l=1}^{K} w_l = 1$
- $\widehat{\boldsymbol{\mu}}_k = \arg\max_{\boldsymbol{\mu}_k} L(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
- $\widehat{\boldsymbol{\Sigma}}_k = \arg\max_{\boldsymbol{\Sigma}_k} L(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

Maximum-Likelihood Estimation II

- To perform this constrained maximization, we introduce the Lagrange multiplier λ ; the resulting unconstrained maximization consists of maximizing the Lagrangian function for $k \in \{1, \ldots, K\}$

$$L_\lambda(w_k) = \sum_{i=1}^{n} y_{ik} \log \pi_k + \lambda \left(1 - \sum_{l=1}^{K} \pi_l\right).$$

Taking the derivative of $L_\lambda(w_k)$ w.r.t w_k we obtain : $\frac{\partial L_\lambda(w_k)}{\partial w_k} = \frac{\sum_{i=1}^{n} y_{ik}}{w_k} - \lambda.$

Then, setting these derivative to zero yields :

$$\frac{\sum_{i=1}^{n} y_{ik}}{w_k} = \lambda.$$

By multiplying each hand side of (24) by π_k and summing over k we get $\sum_{k=1}^{K} \frac{w_k \times \sum_{i=1}^{n} y_{ik}}{w_k} = \sum_{k=1}^{K} \lambda \times \pi_k$ which implies that $\lambda = n$.

Finally, from (24) we get the updating formula for the weights w_k's, that is

$$\widehat{w}_k = \frac{\sum_{i=1}^{n} y_{ik}}{n} = \frac{n_k}{n} = \frac{\#\text{Class}k}{n}, \quad \forall k \in \{1, \ldots, K\}.$$

Estimation the Gaussian parameters I

- Maximizing w.r.t the means $\boldsymbol{\mu}_k$ $\forall k \in \{1, \ldots, K\}$ the function

$$\begin{aligned} L(\boldsymbol{\mu}_k) &= -\frac{1}{2}\sum_{i=1}^{n} y_{ik}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top {\boldsymbol{\Sigma}_k}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k) \\ &= -\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik}\left[\boldsymbol{x}_i^\top \boldsymbol{\Sigma}_k^{-1}\boldsymbol{x}_i - 2\boldsymbol{x}_i^\top \boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k + \boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\right] \end{aligned}$$

Taking the derivative w.r.t $\boldsymbol{\mu}_k$ yields :

$$\frac{\partial L(\boldsymbol{\mu}_k)}{\partial \boldsymbol{\mu}_k} = -\frac{1}{2}\sum_{i=1}^{n} y_{ik}\left[-2\boldsymbol{\Sigma}_k^{-1}\boldsymbol{x}_i + 2\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\right].$$

Then, by setting these derivative to zero we get the MLE for the mean $\boldsymbol{\mu}_k$:

$$\widehat{\boldsymbol{\mu}}_k = \frac{\sum_{i=1}^{n} y_{ik}\boldsymbol{x}_i}{\sum_{i=1}^{n} y_{ik}} \quad \forall k \in \{1, \ldots, K\}.$$

Estimation the Gaussian parameters I

- Maximizing w.r.t the covariance matrix $\boldsymbol{\Sigma}_k$ for $k = 1, \ldots, K$ the function

$$L(\boldsymbol{\Sigma}_k) = -\frac{1}{2}\sum_{i=1}^{n} y_{ik}\left[-\log|\boldsymbol{\Sigma}_k^{-1}| + (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k{}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)\right]$$

where we used the fact that $\log|\mathbf{A}^{-1}| = -\log|\mathbf{A}|$

- Taking the derivative w.r.t the precision matrix $\boldsymbol{\Sigma}_k^{-1}$ (technically easier) :

$$\frac{\partial L(\boldsymbol{\Sigma}_k)}{\partial \boldsymbol{\Sigma}_k^{-1}} = -\frac{1}{2}\sum_{i=1}^{n} y_{ik}\left[-\boldsymbol{\Sigma}_k + (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top\right],$$

we used the properties : $\dfrac{\partial}{\partial \mathbf{A}}\log|\mathbf{A}| = \mathbf{A}^{-T}$

$$\mathbf{u}^T\mathbf{A}\mathbf{u} = \mathrm{trace}(\mathbf{u}^T\mathbf{A}\mathbf{u}) = \mathrm{trace}(\mathbf{u}\mathbf{u}^T\mathbf{A})$$

$$\frac{\partial \mathrm{trace}(\mathbf{B}\mathbf{A})}{\partial \mathbf{A}} = \mathbf{B}^T$$

Setting these derivative to zero we get $\widehat{\boldsymbol{\Sigma}}_k = \frac{1}{n_k}\sum_{i=1}^{n} y_{ik}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top$.
Since the mean is unknown, then we replace it by its MLE $\widehat{\boldsymbol{\mu}}_k$. We then get

$$\widehat{\boldsymbol{\Sigma}}_k = \frac{1}{n_k}\sum_{i=1}^{n} y_{ik}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)^\top.$$

Estimation the Gaussian parameters II

Bias correction for $\widehat{\boldsymbol{\Sigma}}_k$: We note that unlike for the proportions w_k and the mean vectos $\boldsymbol{\mu}_k$, the estimator of the covriance matrix is biased. Indeed

$$\begin{aligned}
\mathbb{E}\widehat{\boldsymbol{\Sigma}}_k &= \frac{1}{n_k}\sum_{i=1}^{n}\mathbb{E}[y_i^k(\boldsymbol{X}_i-\widehat{\boldsymbol{\mu}}_k)(\boldsymbol{X}_i-\widehat{\boldsymbol{\mu}}_k)^\top] \\
&= \frac{1}{n_k}\sum_{i=1}^{n}\mathbb{E}[y_i^k\boldsymbol{X}_i\boldsymbol{X}_i^\top - y_i^k\widehat{\boldsymbol{\mu}}_k\boldsymbol{X}_i^\top - y_i^k\boldsymbol{X}_i\widehat{\boldsymbol{\mu}}_k^\top + y_i^k\widehat{\boldsymbol{\mu}}_k\widehat{\boldsymbol{\mu}}_k^\top] \\
&= \frac{1}{n_k}\sum_{i=1}^{n}\mathbb{E}\left[y_i^k\boldsymbol{X}_i\boldsymbol{X}_i^\top - \frac{2}{n_k}y_i^k\sum_{j=1}^{n}y_j^k\boldsymbol{X}_j\boldsymbol{X}_i^\top + \frac{y_{ik}}{n_k^2}\sum_{j=1}^{n}y_j^k\boldsymbol{X}_j\sum_{l=1}^{n}y_j^l\boldsymbol{X}_l^\top\right]
\end{aligned}$$

Estimation the Gaussian parameters III

$$
\begin{aligned}
\mathbb{E}[y_i^k \boldsymbol{X}_i \boldsymbol{X}_i^\top] &= \boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top \\
\mathbb{E}\big[y_i^k \sum_{j=1}^{n} y_j^k \boldsymbol{X}_j \boldsymbol{X}_i^\top\big] &= \mathbb{E}[\boldsymbol{X}_i \boldsymbol{X}_i^\top] + (n_k - 1)\boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top \\
&= \boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top + (n_k - 1)\boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top = \boldsymbol{\Sigma}_k + n_k \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top \\
\mathbb{E}\left[y_i^k \sum_{j=1}^{n} y_j^k \boldsymbol{X}_j \sum_{l=1}^{n} y_l^k \boldsymbol{X}_l^\top\right] &= \mathbb{E}\left[y_i^k \sum_{j=1}^{n} \sum_{l=1}^{n} y_j^k y_l^k \boldsymbol{X}_j \boldsymbol{X}_l^\top\right] \\
&= n_k \mathbb{E}[y_i^k \boldsymbol{X}_i \boldsymbol{X}_i^\top] + n_k(n_k - 1)\boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top \\
&= n_k(\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top) + n_k(n_k - 1)\boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top \\
&= n_k \boldsymbol{\Sigma}_k + n_k^2 \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top
\end{aligned}
$$

Then

$$
\mathbb{E}[\widehat{\boldsymbol{\Sigma}}_k] = \boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top - \frac{2}{n_k}\boldsymbol{\Sigma}_k - 2\boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top + \frac{1}{n_k}\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top = \boldsymbol{\Sigma}_k - \frac{1}{n_k}\boldsymbol{\Sigma}_k = \frac{n_k - 1}{n_k}\boldsymbol{\Sigma}_k
$$

Estimation the Gaussian covriance in LDA I

- Consider the problem of maximizing w.r.t the covariance matrix $\boldsymbol{\Sigma}$ the function

$$L(\boldsymbol{\Sigma}) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik}\left[-\log|\boldsymbol{\Sigma}^{-1}| + (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)\right]$$

- Taking the derivative of this function w.r.t the precision matrix $\boldsymbol{\Sigma}^{-1}$, we obtain :

$$\frac{\partial L(\boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}^{-1}} = -\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik}\left[-\boldsymbol{\Sigma} + (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top\right].$$

Then, by setting these derivatives to zero we get the updating formula for the covariance matrix $\boldsymbol{\Sigma}$, that is

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top.$$

Since the mean is unknown, then we replace it by its MLE $\widehat{\boldsymbol{\mu}}_k$. We then get

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)^\top.$$

- **Bias correction for $\widehat{\boldsymbol{\Sigma}}$** : We note that unlike for the proportions w_k and the mean vectos $\boldsymbol{\mu}_k$, the estimator of the covriance matrix is biased. Indeed, we can show (similarly as for the MLE of $\boldsymbol{\Sigma}_k$) that

$$\mathbb{E}[\widehat{\boldsymbol{\Sigma}}] = \frac{n-K}{n}\boldsymbol{\Sigma}$$

We then take

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n-K}\sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)^\top.$$

as an unbiased estimator of $\boldsymbol{\Sigma}$

$$\mathbb{E}\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} y_i^k \mathbb{E}\left[\boldsymbol{X}_i\boldsymbol{X}_i^\top - 2\frac{1}{n_k}\sum_{j=1}^{n} y_j^k \boldsymbol{X}_j\boldsymbol{X}_i^\top + \frac{1}{n_k}\sum_{j=1}^{n} y_j^k \boldsymbol{X}_j \frac{1}{n_k}\sum_{l=1}^{n} y_j^l \boldsymbol{X}_l^\top\right]$$

$$\begin{aligned}
\mathbb{E}[\boldsymbol{X}_i\boldsymbol{X}_i^\top] &= \boldsymbol{\Sigma} + \boldsymbol{\mu}_k\boldsymbol{\mu}_k^\top \\
\mathbb{E}[\sum_{j=1}^{n} y_j^k \boldsymbol{X}_j\boldsymbol{X}_i^\top] &= \mathbb{E}[\boldsymbol{X}_i\boldsymbol{X}_i^\top] + (n_k - 1)\boldsymbol{\mu}_k\boldsymbol{\mu}_k^\top \\
&= \boldsymbol{\Sigma} + \boldsymbol{\mu}_k\boldsymbol{\mu}_k^\top + (n_k - 1)\boldsymbol{\mu}_k\boldsymbol{\mu}_k^\top = \boldsymbol{\Sigma} + n_k\boldsymbol{\mu}_k\boldsymbol{\mu}_k^\top \\
\mathbb{E}[\sum_{j=1}^{n} y_j^k \boldsymbol{X}_j \sum_{l=1}^{n} y_l^k \boldsymbol{X}_l^\top] &= \mathbb{E}[\sum_{j=1}^{n}\sum_{l=1}^{n} y_j^k y_l^k \boldsymbol{X}_j\boldsymbol{X}_l^\top] = n_k\mathbb{E}[\boldsymbol{X}_1\boldsymbol{X}_1^\top] + n_k(n_k - 1)\boldsymbol{\mu}_k\boldsymbol{\mu}_k^\top \\
&= n_k(\boldsymbol{\Sigma} + \boldsymbol{\mu}_k\boldsymbol{\mu}_k^\top + (n_k - 1)\boldsymbol{\mu}_k\boldsymbol{\mu}_k^\top) = n_k\boldsymbol{\Sigma} + n_k^2\boldsymbol{\mu}_k\boldsymbol{\mu}_k^\top
\end{aligned}$$

Then

$$
\begin{aligned}
\mathbb{E}[\widehat{\boldsymbol{\Sigma}}] &= \frac{1}{n}[n\boldsymbol{\Sigma} + n\boldsymbol{\mu}_k\boldsymbol{\mu}_k^\top - 2\sum_{i,k} y_i^k \frac{1}{n_k}\boldsymbol{\Sigma} - 2\sum_{i,k} y_i^k \boldsymbol{\mu}_k\boldsymbol{\mu}_k^\top + \sum_{i,k} y_i^k \frac{1}{n_k}\boldsymbol{\Sigma}_k + \sum_{i,k} y_i^k \boldsymbol{\mu}_k\boldsymbol{\mu}_k^\top] \\
&= \frac{1}{n}[n\boldsymbol{\Sigma} + n\boldsymbol{\mu}_k\boldsymbol{\mu}_k^\top - 2\sum_{i,k} y_i^k \frac{1}{n_k}\boldsymbol{\Sigma} - 2n\boldsymbol{\mu}_k\boldsymbol{\mu}_k^\top + \sum_{i,k} y_i^k \frac{1}{n_k}\boldsymbol{\Sigma}_k + n\boldsymbol{\mu}_k\boldsymbol{\mu}_k^\top] \\
&= \frac{1}{n}[n\boldsymbol{\Sigma} - \sum_{i,k} y_i^k \frac{1}{n_k}\boldsymbol{\Sigma}] \\
&= \frac{1}{n}[n\boldsymbol{\Sigma} - \boldsymbol{\Sigma}(\frac{n_1}{n_1} + \frac{n_2}{n_2} + \ldots + \frac{n_K}{n_K})] \\
&= \frac{1}{n}[n\boldsymbol{\Sigma} - K\boldsymbol{\Sigma}] \\
&= \frac{(n-K)}{n}\boldsymbol{\Sigma}
\end{aligned}
$$

Algorithm 1 Pseudo Code `Train_LDA`.

Inputs : n sample $(\boldsymbol{x}_i, y_i)_{i=1}^n$ arranged as $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$ and $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$

$\widehat{\boldsymbol{\Sigma}} = 0$ % `d-by-d matrix of zeros`

for $k = 1, \ldots, K$ **do**

$$\widehat{w}_k = \frac{\sum_{i=1}^n y_{ik}}{n}$$

$$\widehat{\boldsymbol{\mu}}_k = \frac{\sum_{i=1}^n y_{ik} \boldsymbol{x}_i}{\sum_{i=1}^n y_{ik}}$$

$$\widehat{\boldsymbol{\Sigma}} = \widehat{\boldsymbol{\Sigma}} + \frac{1}{n-K} \sum_{i=1}^n y_{ik} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)^\top$$

end

Result: $\widehat{\boldsymbol{\theta}} = \{\widehat{w}_k, \widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}\}$ the MLE of $\boldsymbol{\theta}$

Algorithm 2 Pseudo Code `Predict_LDA`.

Inputs : Test sample $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$ and parameters $\{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}\}$

for $k = 1, \ldots, K$ **do**

$$w_k \mathbf{P}_k = w_k \phi_d(\mathbf{X}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma})$$

end

$\widehat{\boldsymbol{y}} = \arg\max_k w_k \mathbf{P}_k$ % `Predicted labels using Bayes rule`

Result: $\widehat{\boldsymbol{y}}$ the predicted class labels

Algorithm 3 Pseudo Code `Train_QDA`.

Inputs : n sample $(\boldsymbol{x}_i, y_i)_{i=1}^n$ arranged as $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$ and $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$

for $k = 1, \ldots, K$ **do**

$$\widehat{w}_k = \frac{\sum_{i=1}^n y_{ik}}{n}$$

$$\widehat{\boldsymbol{\mu}}_k = \frac{\sum_{i=1}^n y_{ik} \boldsymbol{x}_i}{\sum_{i=1}^n y_{ik}}$$

$$\widehat{\boldsymbol{\Sigma}}_k = \frac{1}{\sum_{i=1}^n y_{ik}} \sum_{i=1}^n y_{ik} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)^\top$$

end

Result: $\widehat{\boldsymbol{\theta}} = \{\widehat{w}_k, \widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}_k\}$ the MLE of $\boldsymbol{\theta}$

Algorithm 4 Pseudo Code `Predict_QDA`.

Inputs : Test sample $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$ and parameters $\{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$

for $k = 1, \ldots, K$ **do**

$$w_k \mathbf{P}_k = w_k \phi_d(\mathbf{X}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

end

$\widehat{\boldsymbol{y}} = \arg\max_k w_k \mathbf{P}_k$ % `Predicted labels using Bayes rule`

Result: $\widehat{\boldsymbol{y}}$ the predicted class labels

Tasks :

- Implement (from the scratch) each of the following functions and apply them to the given data :
  - `train_LDA` and `predict_LDA`
  - `train_QDA` and `predict_QDA`

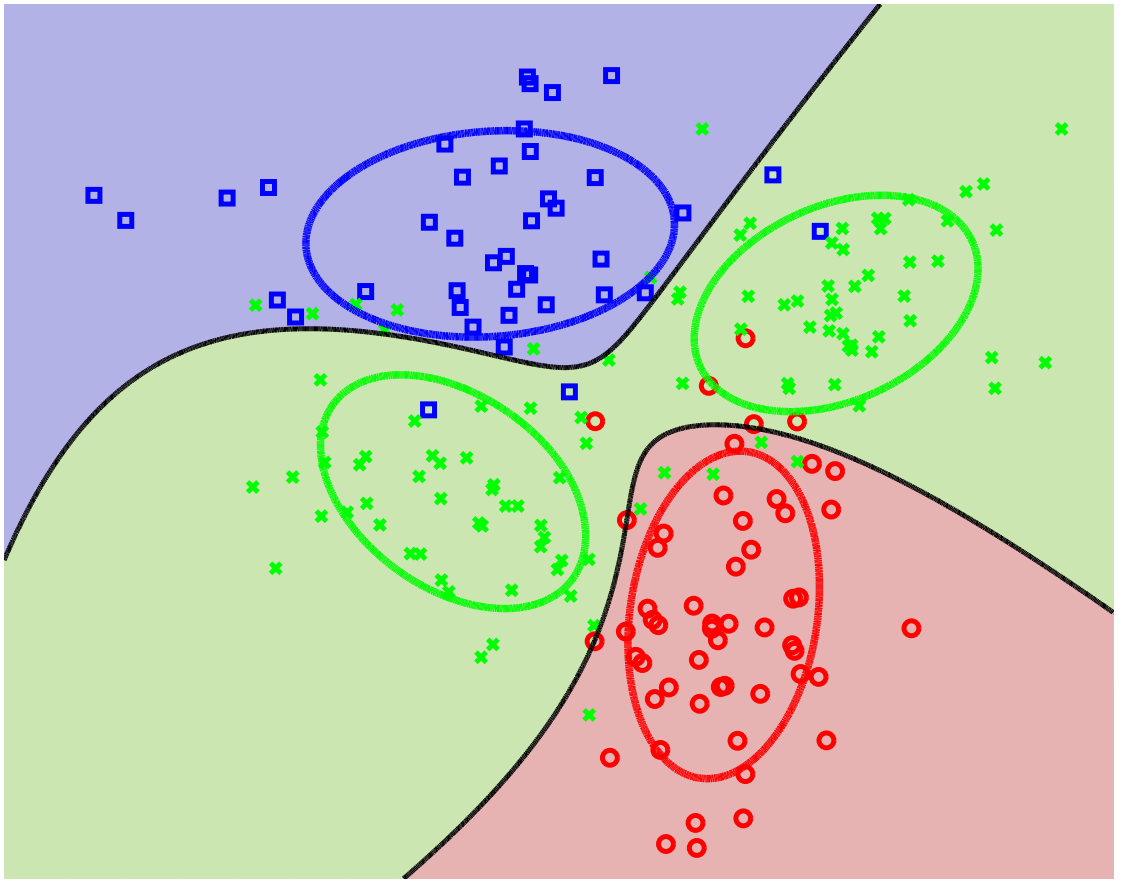
  Datasets :
  - Training data Xtrain.txt and ytrain.txt
  - Testing data : Xtest.txt
- Plot the results by highlighting the classification and the generative model for each class
- compare your results to those you could obtain by using standard packages, for example :

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.discriminant_analysis import
QuadraticDiscriminantAnalysis
```

Mixture Discriminant Analysis

- LDA and QDA model each class conditional density as a Gaussian.
- This may be limited for modeling non-homogeneous classes where some classes are composed of different sub-groups.

Mixture Discriminant Analysis (MDA)



Mixture Discriminant Analysis (MDA)

- Mixture Discriminant Analysis (MDA) models each class conditional density as Gaussian mixture density, rather than a single Gaussian
- with MDA, we can therefore capture many specific properties of real data such as multimodality, heterogeneity, heteroskedasticity, etc.
- Model : $p(\boldsymbol{X} = \boldsymbol{x}, Y = k; \boldsymbol{\theta}) = \mathbb{P}(Y = k)p(\boldsymbol{x}|Y = k; \boldsymbol{\theta}_k) = w_k p(\boldsymbol{x}|Y = k; \boldsymbol{\theta}_k)$ with each class k has an M_k-component Gaussian mixture density :

$$p(\boldsymbol{x}|Y = k; \boldsymbol{\theta}_k) = \sum_{l=1}^{M_k} \alpha_{kl} \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{kl}, \boldsymbol{\Sigma}_{kl})$$

- The α_{kl}'s are the non-negative mixing proportions that sum to 1 $\sum_{l=1}^{M_k} \alpha_{kl} = 1 \; \forall k$.
- $\boldsymbol{\theta}_k = \{\alpha_{kl}, \boldsymbol{\mu}_{kl}, \boldsymbol{\Sigma}_{kl}\}_{l=1}^{M_k}$ is the parameter vector of class k
- we can allow a different covariance matrix for each mixture component as well as a common covariance matrix

Training MDA I

We can show that (Will be detailed later)

$$\widehat{\pi}_k = \frac{\sum_{i=1}^{n} y_{ik}}{n} = \frac{\#\text{Class}k}{n}, \quad \forall k \in \{1, \ldots, K\}.$$

The EM algorithm for each class k

$\forall l \in \{1, \ldots, M_k\}$

$$\tau_l(\boldsymbol{x}_i; \boldsymbol{\theta}_k)^{(t)} = \mathbb{P}(Z_i = l | Y_i = k; \boldsymbol{\theta}^{(t)}) = \frac{\alpha_{kl} \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_{kl}^{(t)}, \boldsymbol{\Sigma}_{kl}^{(t)})}{\sum_{k\ell=1}^{K} \alpha_\ell \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_{k\ell}^{(t)}, \boldsymbol{\Sigma}_{k\ell}^{(t)})} \text{ for } i = 1, \ldots, n$$

$$\begin{aligned}
\alpha_{kl}^{\text{new}} &= \sum_{i=1}^{n} y_{ik} \tau_l(\boldsymbol{x}_i; \boldsymbol{\theta}_k)^{(t)} \Big/ \sum_{i=1}^{n} y_{ik} \\
\boldsymbol{\mu}_{kl}^{\text{new}} &= \sum_{i=1}^{n} y_{ik} \tau_l(\boldsymbol{x}_i; \boldsymbol{\theta}_k)^{(t)} \boldsymbol{x}_i \Big/ \sum_{i=1}^{n} \tau_l(\boldsymbol{x}_i; \boldsymbol{\theta}_k)^{(t)} y_{ik} \\
\boldsymbol{\Sigma}_{kl}^{\text{new}} &= \sum_{i=1}^{n} y_{ik} \tau_l(\boldsymbol{x}_i; \boldsymbol{\theta}_k)^{(t)} (\boldsymbol{x}_i - \boldsymbol{\mu}_{kl}^{\text{new}})(\boldsymbol{x}_i - \boldsymbol{\mu}_{kl}^{\text{new}})^\top \Big/ \sum_{i=1}^{n} \tau_l(\boldsymbol{x}_i; \boldsymbol{\theta}_k)^{(t)} y_{ik}.
\end{aligned}$$