

Decision-theoretic Machine Learning

Krzysztof Dembczyński and Wojciech Kotłowski

Intelligent Decision Support Systems Laboratory (IDSS)
Poznań University of Technology, Poland



Poznań University of Technology, Summer 2018

Agenda

- 1 **Introduction to Machine Learning**
- 2 Binary Classification
- 3 Bipartite Ranking
- 4 Multi-Label Classification

Outline

- 1 Statistical decision theory for supervised learning
- 2 Learning paradigms and principles
- 3 Examples of learning algorithms
- 4 Summary

Outline

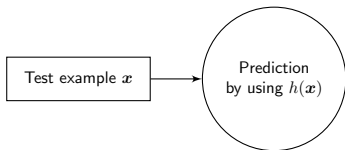
- 1 Statistical decision theory for supervised learning
- 2 Learning paradigms and principles
- 3 Examples of learning algorithms
- 4 Summary

Supervised learning

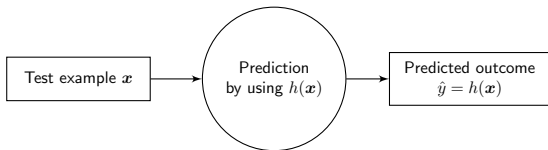
Supervised learning

Test example x

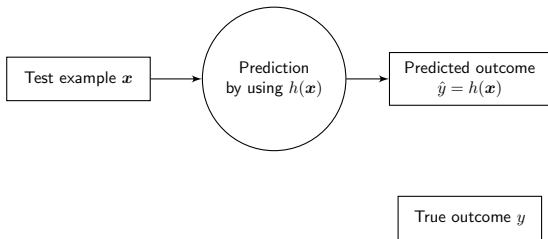
Supervised learning



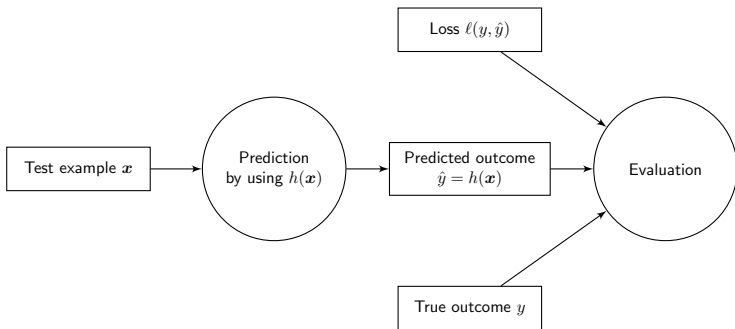
Supervised learning



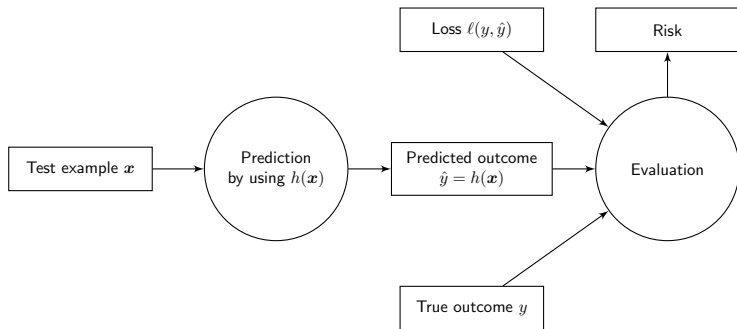
Supervised learning



Supervised learning



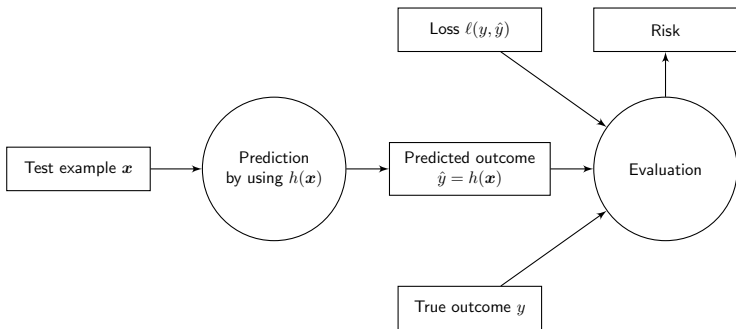
Supervised learning



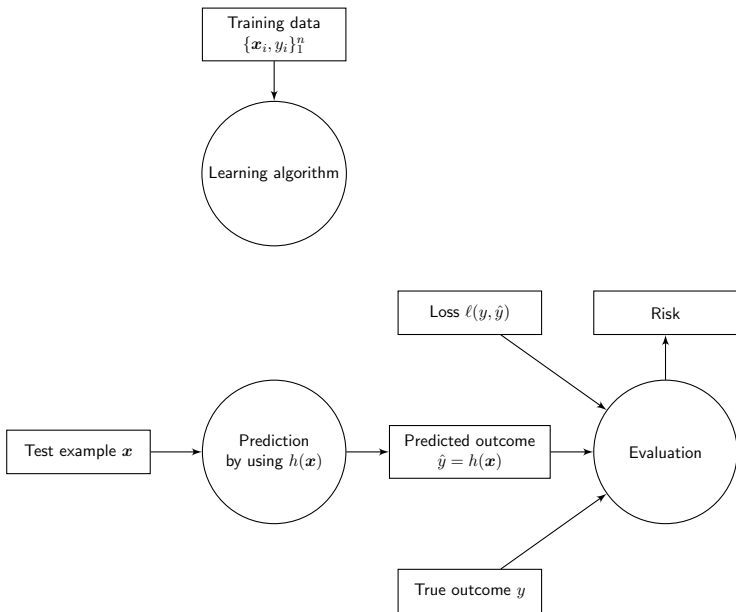
Supervised learning

Training data

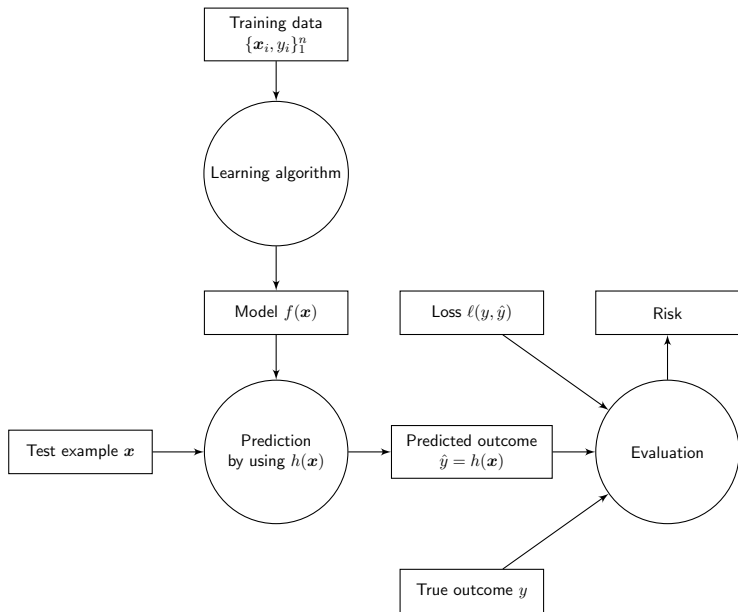
$$\{\mathbf{x}_i, y_i\}_1^n$$



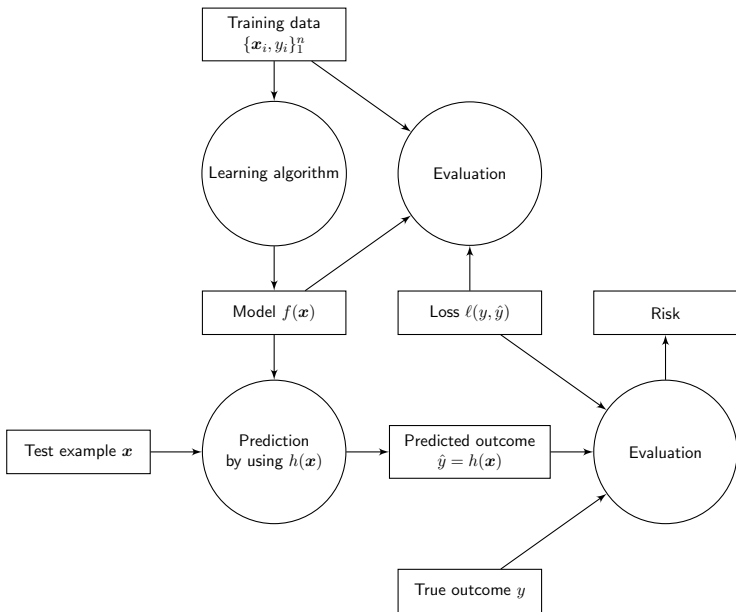
Supervised learning



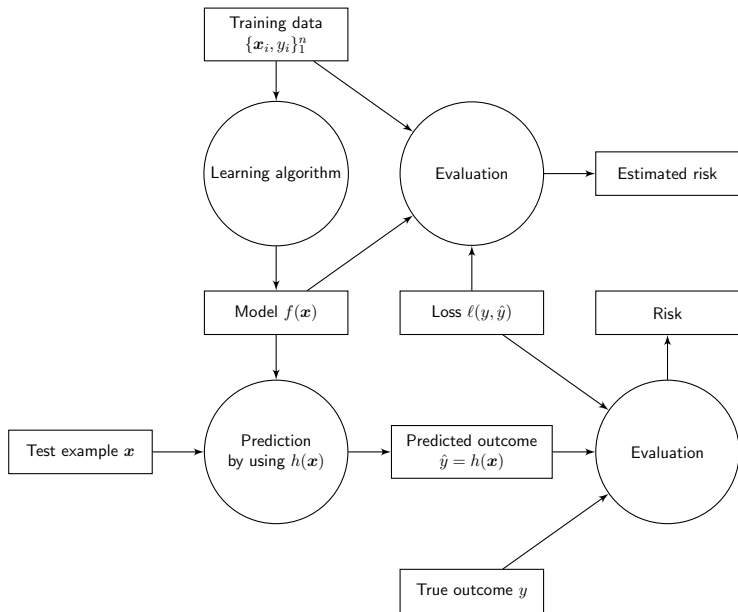
Supervised learning



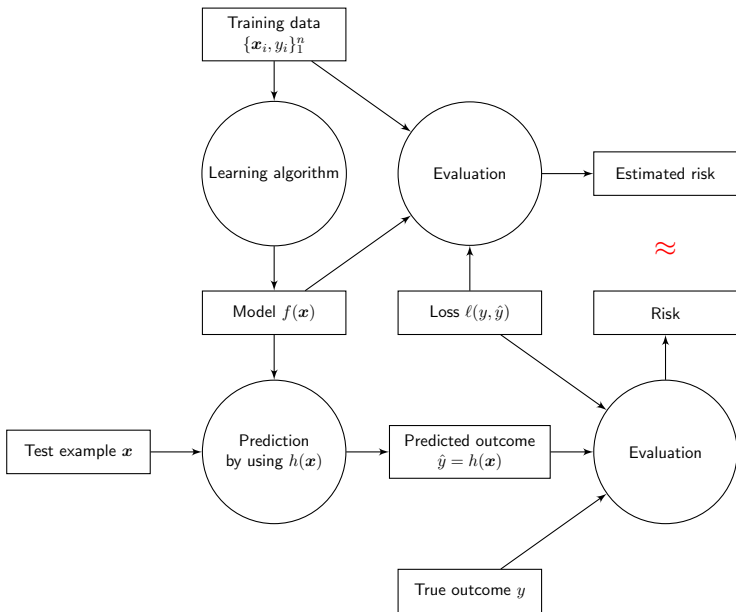
Supervised learning



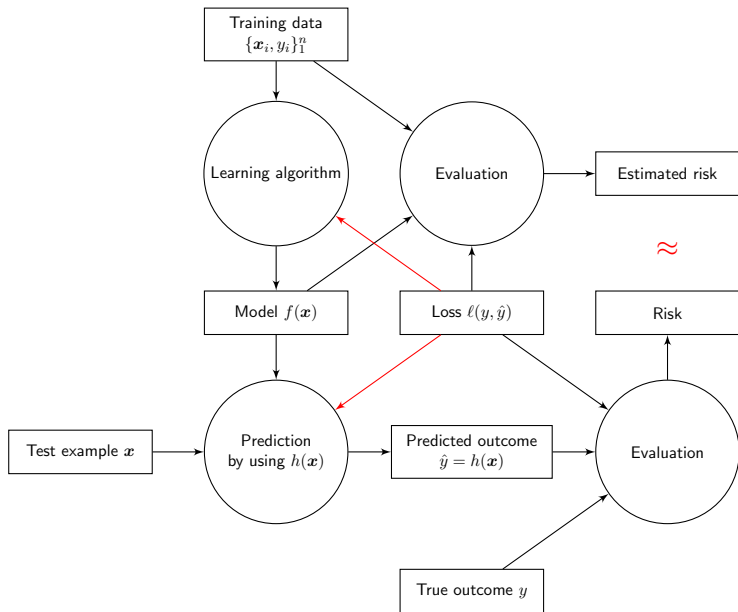
Supervised learning



Supervised learning



Supervised learning



Statistical learning framework

Statistical learning framework

- **Input** $\mathbf{x} \in \mathcal{X}$ drawn from a distribution $P(\mathbf{x})$.
 - ▶ usually a feature vector, $\mathcal{X} \subseteq \mathbb{R}^d$.

Statistical learning framework

- **Input** $\mathbf{x} \in \mathcal{X}$ drawn from a distribution $P(\mathbf{x})$.
 - ▶ usually a feature vector, $\mathcal{X} \subseteq \mathbb{R}^d$.
- **Outcome** $y \in \mathcal{Y}$ drawn from a distribution $P(y | \mathbf{x})$.
 - ▶ target of our prediction: class label, real value, label vector, etc.,
 - ▶ alternative view: **example** (\mathbf{x}, y) drawn from $P(\mathbf{x}, y)$.

Statistical learning framework

- **Input** $\mathbf{x} \in \mathcal{X}$ drawn from a distribution $P(\mathbf{x})$.
 - ▶ usually a feature vector, $\mathcal{X} \subseteq \mathbb{R}^d$.
- **Outcome** $y \in \mathcal{Y}$ drawn from a distribution $P(y | \mathbf{x})$.
 - ▶ target of our prediction: class label, real value, label vector, etc.,
 - ▶ alternative view: **example** (\mathbf{x}, y) drawn from $P(\mathbf{x}, y)$.
- **Prediction** $\hat{y} = h(\mathbf{x})$ by means of **prediction function** $h: \mathcal{X} \rightarrow \mathcal{Y}$.
 - ▶ h returns prediction $\hat{y} = h(\mathbf{x})$ for every input \mathbf{x} .

Statistical learning framework

- **Input** $\mathbf{x} \in \mathcal{X}$ drawn from a distribution $P(\mathbf{x})$.
 - ▶ usually a feature vector, $\mathcal{X} \subseteq \mathbb{R}^d$.
- **Outcome** $y \in \mathcal{Y}$ drawn from a distribution $P(y | \mathbf{x})$.
 - ▶ target of our prediction: class label, real value, label vector, etc.,
 - ▶ alternative view: **example** (\mathbf{x}, y) drawn from $P(\mathbf{x}, y)$.
- **Prediction** $\hat{y} = h(\mathbf{x})$ by means of **prediction function** $h: \mathcal{X} \rightarrow \mathcal{Y}$.
 - ▶ h returns prediction $\hat{y} = h(\mathbf{x})$ for every input \mathbf{x} .
- **Loss** of our prediction: $\ell(y, \hat{y})$.
 - ▶ $\ell: \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$ is a problem-specific **loss function**.

Statistical learning framework

- **Input** $\mathbf{x} \in \mathcal{X}$ drawn from a distribution $P(\mathbf{x})$.
 - ▶ usually a feature vector, $\mathcal{X} \subseteq \mathbb{R}^d$.
- **Outcome** $y \in \mathcal{Y}$ drawn from a distribution $P(y | \mathbf{x})$.
 - ▶ target of our prediction: class label, real value, label vector, etc.,
 - ▶ alternative view: **example** (\mathbf{x}, y) drawn from $P(\mathbf{x}, y)$.
- **Prediction** $\hat{y} = h(\mathbf{x})$ by means of **prediction function** $h: \mathcal{X} \rightarrow \mathcal{Y}$.
 - ▶ h returns prediction $\hat{y} = h(\mathbf{x})$ for every input \mathbf{x} .
- **Loss** of our prediction: $\ell(y, \hat{y})$.
 - ▶ $\ell: \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$ is a problem-specific **loss function**.
- **Goal**: find a prediction function with small loss.

Risk

- **Goal:** minimize the **expected** loss over all examples (**risk**):

$$L_{\ell}(h) = \mathbb{E}_{(\mathbf{x}, y) \sim P} [\ell(y, h(\mathbf{x}))].$$

Risk

- **Goal:** minimize the **expected** loss over all examples (**risk**):

$$L_{\ell}(h) = \mathbb{E}_{(\mathbf{x}, y) \sim P} [\ell(y, h(\mathbf{x}))].$$

- The **optimal** prediction function over all possible functions:

$$h^* = \arg \min_h L(h),$$

sometimes referred to as the **Bayes prediction function**.

Risk

- **Goal:** minimize the **expected** loss over all examples (**risk**):

$$L_{\ell}(h) = \mathbb{E}_{(\mathbf{x}, y) \sim P} [\ell(y, h(\mathbf{x}))].$$

- The **optimal** prediction function over all possible functions:

$$h^* = \arg \min_h L(h),$$

sometimes referred to as the **Bayes prediction function**.

- The smallest achievable risk (**Bayes risk**):

$$L_{\ell}^* = L_{\ell}(h^*).$$

Decomposition of risk

$$L_\ell(h) = \mathbb{E}_{(\mathbf{x}, y)} [\ell(y, h(\mathbf{x}))]$$

Decomposition of risk

$$\begin{aligned}L_{\ell}(h) &= \mathbb{E}_{(\mathbf{x},y)} [\ell(y, h(\mathbf{x}))] \\ &= \int_{\mathcal{X} \times \mathcal{Y}} \ell(y, h(\mathbf{x})) P(\mathbf{x}, y) d\mathbf{x} dy\end{aligned}$$

Decomposition of risk

$$\begin{aligned}L_{\ell}(h) &= \mathbb{E}_{(\mathbf{x}, y)} [\ell(y, h(\mathbf{x}))] \\&= \int_{\mathcal{X} \times \mathcal{Y}} \ell(y, h(\mathbf{x})) P(\mathbf{x}, y) d\mathbf{x} dy \\&= \int_{\mathcal{X}} \left(\int_{\mathcal{Y}} \ell(y, h(\mathbf{x})) P(y | \mathbf{x}) dy \right) P(\mathbf{x}) d\mathbf{x}\end{aligned}$$

Decomposition of risk

$$\begin{aligned}L_{\ell}(h) &= \mathbb{E}_{(\mathbf{x}, y)} [\ell(y, h(\mathbf{x}))] \\&= \int_{\mathcal{X} \times \mathcal{Y}} \ell(y, h(\mathbf{x})) P(\mathbf{x}, y) d\mathbf{x} dy \\&= \int_{\mathcal{X}} \left(\int_{\mathcal{Y}} \ell(y, h(\mathbf{x})) P(y | \mathbf{x}) dy \right) P(\mathbf{x}) d\mathbf{x} \\&= \mathbb{E}_{\mathbf{x}} [L_{\ell}(h | \mathbf{x})].\end{aligned}$$

Decomposition of risk

$$\begin{aligned}L_{\ell}(h) &= \mathbb{E}_{(\mathbf{x}, y)} [\ell(y, h(\mathbf{x}))] \\&= \int_{\mathcal{X} \times \mathcal{Y}} \ell(y, h(\mathbf{x})) P(\mathbf{x}, y) d\mathbf{x} dy \\&= \int_{\mathcal{X}} \left(\int_{\mathcal{Y}} \ell(y, h(\mathbf{x})) P(y | \mathbf{x}) dy \right) P(\mathbf{x}) d\mathbf{x} \\&= \mathbb{E}_{\mathbf{x}} [L_{\ell}(h | \mathbf{x})].\end{aligned}$$

- $L_{\ell}(h | \mathbf{x})$ is the **conditional risk** of $\hat{y} = h(\mathbf{x})$ at \mathbf{x} .

Decomposition of risk

$$\begin{aligned}L_{\ell}(h) &= \mathbb{E}_{(\mathbf{x}, y)} [\ell(y, h(\mathbf{x}))] \\&= \int_{\mathcal{X} \times \mathcal{Y}} \ell(y, h(\mathbf{x})) P(\mathbf{x}, y) d\mathbf{x} dy \\&= \int_{\mathcal{X}} \left(\int_{\mathcal{Y}} \ell(y, h(\mathbf{x})) P(y | \mathbf{x}) dy \right) P(\mathbf{x}) d\mathbf{x} \\&= \mathbb{E}_{\mathbf{x}} [L_{\ell}(h | \mathbf{x})].\end{aligned}$$

- $L_{\ell}(h | \mathbf{x})$ is the **conditional risk** of $\hat{y} = h(\mathbf{x})$ at \mathbf{x} .
- Bayes prediction **minimizes the conditional risk** for every \mathbf{x} :

$$h^*(\mathbf{x}) = \arg \min_h L_{\ell}(h | \mathbf{x}).$$

Making optimal decisions

Example

- Pack of cards: 7 diamonds (red), 5 hearts (red), 5 spades (black), 3 clubs (black).

Making optimal decisions

Example

- Pack of cards: 7 diamonds (red), 5 hearts (red), 5 spades (black), 3 clubs (black).
- Decision = bet (four choices).

Making optimal decisions

Example

- Pack of cards: 7 diamonds (red), 5 hearts (red), 5 spades (black), 3 clubs (black).
- Decision = bet (four choices).
- If you win you get 100\$, if you loose you must give 50\$.

Making optimal decisions

Example

- Pack of cards: 7 diamonds (red), 5 hearts (red), 5 spades (black), 3 clubs (black).
- Decision = bet (four choices).
- If you win you get 100\$, if you loose you must give 50\$.
- What is the loss and optimal decision?

Making optimal decisions

Example

- Pack of cards: 7 diamonds (red), 5 hearts (red), 5 spades (black), 3 clubs (black).
- Decision = bet (four choices).
- If you win you get 100\$, if you loose you must give 50\$.
- What is the loss and optimal decision?
- Suppose we know the card is black. What is the optimal decision now?

Making optimal decisions

Example

- Pack of cards: 7 diamonds (red), 5 hearts (red), 5 spades (black), 3 clubs (black).
- Decision = bet (four choices).
- If you win you get 100\$, if you loose you must give 50\$.
- What is the loss and optimal decision?
- Suppose we know the card is black. What is the optimal decision now?
- What are the input variables?

Making optimal decisions

Example

- Pack of cards: 7 diamonds (red), 5 hearts (red), 5 spades (black), 3 clubs (black).

Making optimal decisions

Example

- Pack of cards: 7 diamonds (red), 5 hearts (red), 5 spades (black), 3 clubs (black).
- Bet the color:

Making optimal decisions

Example

- Pack of cards: 7 diamonds (red), 5 hearts (red), 5 spades (black), 3 clubs (black).
- Bet the color:
 - ▶ if the true color is red and you are correct you win 50, otherwise you lose 100,

Making optimal decisions

Example

- Pack of cards: 7 diamonds (red), 5 hearts (red), 5 spades (black), 3 clubs (black).
- Bet the color:
 - ▶ if the true color is red and you are correct you win 50, otherwise you loose 100,
 - ▶ if the true color is black and you are correct you win 200, otherwise you loose 100.

Making optimal decisions

Example

- Pack of cards: 7 diamonds (red), 5 hearts (red), 5 spades (black), 3 clubs (black).
- Bet the color:
 - ▶ if the true color is red and you are correct you win 50, otherwise you loose 100,
 - ▶ if the true color is black and you are correct you win 200, otherwise you loose 100.
- What is the loss and optimal decision now?

Regression

- Prediction of a **real-valued** outcome $y \in \mathbb{R}$.
- Find a prediction function $h(\mathbf{x})$ that accurately predicts value of y .
- The most common loss function used is **squared error loss**:

$$\ell_{se}(y, \hat{y}) = (y - \hat{y})^2,$$

where $\hat{y} = h(\mathbf{x})$.

Regression

- The conditional risk for squared error loss is :

Regression

- The conditional risk for squared error loss is :

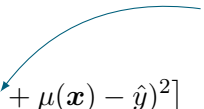
$$L_{se}(h | \mathbf{x}) = \mathbb{E}_{y|\mathbf{x}} [(y - \hat{y})^2]$$

Regression

- The conditional risk for squared error loss is :

$$L_{se}(h | \mathbf{x}) = \mathbb{E}_{y|\mathbf{x}} [(y - \hat{y})^2]$$

$$= \mathbb{E}_{y|\mathbf{x}} [(y - \mu(\mathbf{x}) + \mu(\mathbf{x}) - \hat{y})^2]$$

$$\mu(\mathbf{x}) = \mathbb{E}_{y|\mathbf{x}}[y]$$


Regression

- The conditional risk for squared error loss is :

$$\begin{aligned}L_{se}(h | \mathbf{x}) &= \mathbb{E}_{y|\mathbf{x}} [(y - \hat{y})^2] \\&= \mathbb{E}_{y|\mathbf{x}} [(y - \mu(\mathbf{x}) + \mu(\mathbf{x}) - \hat{y})^2] \\&= \mathbb{E}_{y|\mathbf{x}} \left[(y - \mu(\mathbf{x}))^2 + \underbrace{2(y - \mu(\mathbf{x}))(\mu(\mathbf{x}) - \hat{y})}_{=0 \text{ under expectation}} + (\mu(\mathbf{x}) - \hat{y})^2 \right]\end{aligned}$$

$\mu(\mathbf{x}) = \mathbb{E}_{y|\mathbf{x}}[y]$

Regression

- The conditional risk for squared error loss is :

$$\begin{aligned}L_{se}(h | \mathbf{x}) &= \mathbb{E}_{y|\mathbf{x}} [(y - \hat{y})^2] \\&= \mathbb{E}_{y|\mathbf{x}} [(y - \mu(\mathbf{x}) + \mu(\mathbf{x}) - \hat{y})^2] \\&= \mathbb{E}_{y|\mathbf{x}} \left[(y - \mu(\mathbf{x}))^2 + \underbrace{2(y - \mu(\mathbf{x}))(\mu(\mathbf{x}) - \hat{y})}_{=0 \text{ under expectation}} + (\mu(\mathbf{x}) - \hat{y})^2 \right] \\&= \underbrace{\mathbb{E}_{y|\mathbf{x}} [(y - \mu(\mathbf{x}))^2]}_{\text{independent of } \hat{y}} + (\mu(\mathbf{x}) - \hat{y})^2.\end{aligned}$$

$\mu(\mathbf{x}) = \mathbb{E}_{y|\mathbf{x}}[y]$

Regression

- The conditional risk for squared error loss is :

$$\begin{aligned}L_{se}(h | \mathbf{x}) &= \mathbb{E}_{y|\mathbf{x}} [(y - \hat{y})^2] \\&= \mathbb{E}_{y|\mathbf{x}} [(y - \mu(\mathbf{x}) + \mu(\mathbf{x}) - \hat{y})^2] \\&= \mathbb{E}_{y|\mathbf{x}} \left[(y - \mu(\mathbf{x}))^2 + \underbrace{2(y - \mu(\mathbf{x}))(\mu(\mathbf{x}) - \hat{y})}_{=0 \text{ under expectation}} + (\mu(\mathbf{x}) - \hat{y})^2 \right] \\&= \underbrace{\mathbb{E}_{y|\mathbf{x}} [(y - \mu(\mathbf{x}))^2]}_{\text{independent of } \hat{y}} + (\mu(\mathbf{x}) - \hat{y})^2.\end{aligned}$$

$\mu(\mathbf{x}) = \mathbb{E}_{y|\mathbf{x}}[y]$

- Hence, $h^*(\mathbf{x}) = \mu(\mathbf{x})$, the **conditional expectation** of y at \mathbf{x} , and:

$$L_{se}(h^* | \mathbf{x}) = \mathbb{E}_{y|\mathbf{x}} [(y - \mu(\mathbf{x}))^2] = \text{Var}(y|\mathbf{x}).$$

Regression

- Another loss commonly used in regression is the **absolute error**:

$$\ell_{ae}(y, \hat{y}) = |y - \hat{y}|.$$

- The Bayes classifier for the absolute-error loss is:

$$h^*(\mathbf{x}) = \arg \min_h L_{ae}(h | \mathbf{x}) =$$

Regression

- Another loss commonly used in regression is the **absolute error**:

$$\ell_{ae}(y, \hat{y}) = |y - \hat{y}|.$$

- The Bayes classifier for the absolute-error loss is:

$$h^*(\mathbf{x}) = \arg \min_h L_{ae}(h | \mathbf{x}) = \text{median}(y|\mathbf{x}),$$

i.e., **median** of the conditional distribution of y given \mathbf{x} .

Regression

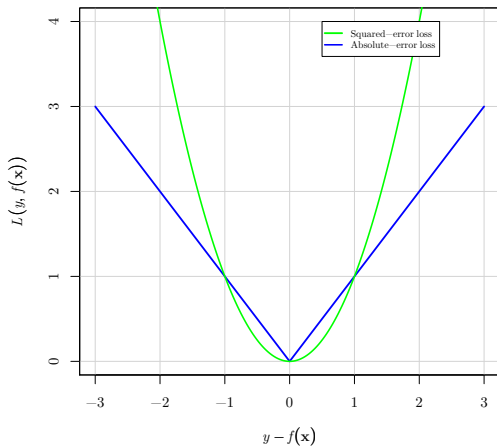


Figure: Loss functions for regression task

Binary Classification

- Prediction of a **binary** outcome $y \in \{-1, 1\}$ (alternatively $y \in \{0, 1\}$).
- Find a prediction function $h(\mathbf{x})$ that accurately predicts value of y .
- The most common loss function used is **0/1 loss**:

$$\ell_{0/1}(y, \hat{y}) = \begin{cases} 0, & \text{if } y = \hat{y}, \\ 1, & \text{otherwise.} \end{cases}$$

Binary Classification

- Define $\eta(\mathbf{x}) = P(y = 1|\mathbf{x})$.

Binary Classification

- Define $\eta(\mathbf{x}) = P(y = 1|\mathbf{x})$.
- The conditional 0/1 risk at \mathbf{x} is:

Binary Classification

- Define $\eta(\mathbf{x}) = P(y = 1|\mathbf{x})$.
- The conditional 0/1 risk at \mathbf{x} is:

$$L_{0/1}(h|\mathbf{x}) = \eta(\mathbf{x})\mathbb{I}[h(\mathbf{x}) = -1] + (1 - \eta(\mathbf{x}))\mathbb{I}[h(\mathbf{x}) = 1].$$

Binary Classification

- Define $\eta(\mathbf{x}) = P(y = 1|\mathbf{x})$.
- The conditional 0/1 risk at \mathbf{x} is:

$$L_{0/1}(h|\mathbf{x}) = \eta(\mathbf{x})\mathbb{I}[h(\mathbf{x}) = -1] + (1 - \eta(\mathbf{x}))\mathbb{I}[h(\mathbf{x}) = 1].$$

- The **Bayes classifier**:

Binary Classification

- Define $\eta(\mathbf{x}) = P(y = 1|\mathbf{x})$.
- The conditional 0/1 risk at \mathbf{x} is:

$$L_{0/1}(h|\mathbf{x}) = \eta(\mathbf{x})\mathbb{I}[h(\mathbf{x}) = -1] + (1 - \eta(\mathbf{x}))\mathbb{I}[h(\mathbf{x}) = 1].$$

- The **Bayes classifier**:

$$h^*(\mathbf{x}) = \begin{cases} 1 & \text{if } \eta(\mathbf{x}) > 1 - \eta(\mathbf{x}) \\ -1 & \text{if } \eta(\mathbf{x}) < 1 - \eta(\mathbf{x}) \end{cases} = \text{sgn}(\eta(\mathbf{x}) - 1/2),$$

and the **Bayes conditional risk**:

Binary Classification

- Define $\eta(\mathbf{x}) = P(y = 1|\mathbf{x})$.
- The conditional 0/1 risk at \mathbf{x} is:

$$L_{0/1}(h|\mathbf{x}) = \eta(\mathbf{x})\mathbb{I}[h(\mathbf{x}) = -1] + (1 - \eta(\mathbf{x}))\mathbb{I}[h(\mathbf{x}) = 1].$$

- The **Bayes classifier**:

$$h^*(\mathbf{x}) = \begin{cases} 1 & \text{if } \eta(\mathbf{x}) > 1 - \eta(\mathbf{x}) \\ -1 & \text{if } \eta(\mathbf{x}) < 1 - \eta(\mathbf{x}) \end{cases} = \text{sgn}(\eta(\mathbf{x}) - 1/2),$$

and the **Bayes conditional risk**:

$$L_{\ell}(h^* | \mathbf{x}) = \min\{\eta(\mathbf{x}), 1 - \eta(\mathbf{x})\}.$$

Multi-class classification

- **Domain** of outcome variable y is a set of labels $\mathcal{Y} = \{1, \dots, K\}$.
- **Goal**: find a prediction function $h(\mathbf{x})$ that for any object \mathbf{x} predicts accurately the actual value of y .
- **Loss function**: the most common is 0/1 loss:

$$\ell_{0/1}(y, \hat{y}) = \begin{cases} 0, & \text{if } y = \hat{y}, \\ 1, & \text{otherwise.} \end{cases}$$

Multi-class classification

- The conditional risk of the 0/1 loss is:

$$L_{0/1}(h | \mathbf{x}) = \mathbb{E}_{y|\mathbf{x}} \ell_{0/1}(y, h(\mathbf{x}))$$

Multi-class classification

- The conditional risk of the 0/1 loss is:

$$\begin{aligned}L_{0/1}(h | \mathbf{x}) &= \mathbb{E}_{y|\mathbf{x}} \ell_{0/1}(y, h(\mathbf{x})) \\ &= \sum_{k \in \mathcal{Y}} P(y = k | \mathbf{x}) \ell_{0/1}(k, h(\mathbf{x}))\end{aligned}$$

Multi-class classification

- The conditional risk of the 0/1 loss is:

$$\begin{aligned}L_{0/1}(h | \mathbf{x}) &= \mathbb{E}_{y|\mathbf{x}} \ell_{0/1}(y, h(\mathbf{x})) \\ &= \sum_{k \in \mathcal{Y}} P(y = k | \mathbf{x}) \ell_{0/1}(k, h(\mathbf{x}))\end{aligned}$$

- Therefore, the Bayes classifier is given by:

$$h^*(\mathbf{x}) = \arg \min_h L_{0/1}(h | \mathbf{x})$$

Multi-class classification

- The conditional risk of the 0/1 loss is:

$$\begin{aligned}L_{0/1}(h | \mathbf{x}) &= \mathbb{E}_{y|\mathbf{x}} \ell_{0/1}(y, h(\mathbf{x})) \\ &= \sum_{k \in \mathcal{Y}} P(y = k | \mathbf{x}) \ell_{0/1}(k, h(\mathbf{x}))\end{aligned}$$

- Therefore, the Bayes classifier is given by:

$$\begin{aligned}h^*(\mathbf{x}) &= \arg \min_h L_{0/1}(h | \mathbf{x}) \\ &= \arg \max_k P(y = k | \mathbf{x}),\end{aligned}$$

the class with **the largest conditional probability** $P(y|\mathbf{x})$.

Multi-class classification

- The conditional risk of the 0/1 loss is:

$$\begin{aligned}L_{0/1}(h | \mathbf{x}) &= \mathbb{E}_{y|\mathbf{x}} \ell_{0/1}(y, h(\mathbf{x})) \\ &= \sum_{k \in \mathcal{Y}} P(y = k | \mathbf{x}) \ell_{0/1}(k, h(\mathbf{x}))\end{aligned}$$

- Therefore, the Bayes classifier is given by:

$$\begin{aligned}h^*(\mathbf{x}) &= \arg \min_h L_{0/1}(h | \mathbf{x}) \\ &= \arg \max_k P(y = k | \mathbf{x}),\end{aligned}$$

the class with **the largest conditional probability** $P(y|\mathbf{x})$.

- The Bayes conditional risk:

Multi-class classification

- The conditional risk of the 0/1 loss is:

$$\begin{aligned}L_{0/1}(h | \mathbf{x}) &= \mathbb{E}_{y|\mathbf{x}} \ell_{0/1}(y, h(\mathbf{x})) \\ &= \sum_{k \in \mathcal{Y}} P(y = k | \mathbf{x}) \ell_{0/1}(k, h(\mathbf{x}))\end{aligned}$$

- Therefore, the Bayes classifier is given by:

$$\begin{aligned}h^*(\mathbf{x}) &= \arg \min_h L_{0/1}(h | \mathbf{x}) \\ &= \arg \max_k P(y = k | \mathbf{x}),\end{aligned}$$

the class with **the largest conditional probability** $P(y|\mathbf{x})$.

- The Bayes conditional risk:

$$L_\ell(h^* | \mathbf{x}) = \min\{1 - P(y = k | \mathbf{x}) : k \in \mathcal{Y}\}.$$

Deterministic learning framework

- **Input** $x \in \mathcal{X}$ drawn from a distribution $P(x)$.
- **Outcome** $y \in \mathcal{Y}$.
- **Unknown target function** $h^*: \mathcal{X} \rightarrow \mathcal{Y}$, such that $y = h^*(x)$.
- **Goal**: discover h^* by observing examples of (x, y) .

Deterministic learning framework

- **Input** $\mathbf{x} \in \mathcal{X}$ drawn from a distribution $P(\mathbf{x})$.
- **Outcome** $y \in \mathcal{Y}$.
- **Unknown target function** $h^*: \mathcal{X} \rightarrow \mathcal{Y}$, such that $y = h^*(\mathbf{x})$.
- **Goal**: discover h^* by observing examples of (\mathbf{x}, y) .

- This is a **special case** of the statistical framework:
 - ▶ What is $P(y|\mathbf{x})$?

 - ▶ Bayes prediction function?

 - ▶ Risk of h^* ? (assuming $\ell(y, \hat{y}) = 0$ whenever $y = \hat{y}$)

Deterministic learning framework

- **Input** $\mathbf{x} \in \mathcal{X}$ drawn from a distribution $P(\mathbf{x})$.
- **Outcome** $y \in \mathcal{Y}$.
- **Unknown target function** $h^*: \mathcal{X} \rightarrow \mathcal{Y}$, such that $y = h^*(\mathbf{x})$.
- **Goal**: discover h^* by observing examples of (\mathbf{x}, y) .

- This is a **special case** of the statistical framework:
 - ▶ What is $P(y|\mathbf{x})$?
 - $P(y|\mathbf{x})$ is a **degenerate** distribution for every \mathbf{x} .
 - ▶ Bayes prediction function?

 - ▶ Risk of h^* ? (assuming $\ell(y, \hat{y}) = 0$ whenever $y = \hat{y}$)

Deterministic learning framework

- **Input** $\mathbf{x} \in \mathcal{X}$ drawn from a distribution $P(\mathbf{x})$.
- **Outcome** $y \in \mathcal{Y}$.
- **Unknown target function** $h^*: \mathcal{X} \rightarrow \mathcal{Y}$, such that $y = h^*(\mathbf{x})$.
- **Goal**: discover h^* by observing examples of (\mathbf{x}, y) .

- This is a **special case** of the statistical framework:
 - ▶ What is $P(y|\mathbf{x})$?
 - $P(y|\mathbf{x})$ is a **degenerate** distribution for every \mathbf{x} .
 - ▶ Bayes prediction function?
 - h^*
 - ▶ Risk of h^* ? (assuming $\ell(y, \hat{y}) = 0$ whenever $y = \hat{y}$)

Deterministic learning framework

- **Input** $\mathbf{x} \in \mathcal{X}$ drawn from a distribution $P(\mathbf{x})$.
- **Outcome** $y \in \mathcal{Y}$.
- **Unknown target function** $h^*: \mathcal{X} \rightarrow \mathcal{Y}$, such that $y = h^*(\mathbf{x})$.
- **Goal**: discover h^* by observing examples of (\mathbf{x}, y) .

- This is a **special case** of the statistical framework:
 - ▶ What is $P(y|\mathbf{x})$?
 - $P(y|\mathbf{x})$ is a **degenerate** distribution for every \mathbf{x} .
 - ▶ Bayes prediction function?
 - h^*
 - ▶ Risk of h^* ? (assuming $\ell(y, \hat{y}) = 0$ whenever $y = \hat{y}$)
 - h^* has **zero risk**.

Deterministic learning framework

- **Input** $\mathbf{x} \in \mathcal{X}$ drawn from a distribution $P(\mathbf{x})$.
- **Outcome** $y \in \mathcal{Y}$.
- **Unknown target function** $h^*: \mathcal{X} \rightarrow \mathcal{Y}$, such that $y = h^*(\mathbf{x})$.
- **Goal**: discover h^* by observing examples of (\mathbf{x}, y) .

- This is a **special case** of the statistical framework:
 - ▶ What is $P(y|\mathbf{x})$?
 - $P(y|\mathbf{x})$ is a **degenerate** distribution for every \mathbf{x} .
 - ▶ Bayes prediction function?
 - h^*
 - ▶ Risk of h^* ? (assuming $\ell(y, \hat{y}) = 0$ whenever $y = \hat{y}$)
 - h^* has **zero risk**.
 - ▶ Unrealistic scenario in real life.

Outline

- 1 Statistical decision theory for supervised learning
- 2 Learning paradigms and principles**
- 3 Examples of learning algorithms
- 4 Summary

Learning

- Distribution $P(\mathbf{x}, y)$ is unknown **unknown**.
- Therefore, Bayes classifier h^* is also **unknown**.
- Instead, we have access to n independent and identically distributed (i.i.d) **training examples (sample)**:

$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}.$$

- **Learning**: use training data to find a good **approximation** of h^* .

Spam filtering

- Problem: Predict whether a given email is spam or not.
- An object to be classified: an email.
- There are two possible responses (classes): spam, not spam.

From: mr jove markson <mjove_marks03@live.fr> ☆
Subject: **[! SPAM] ***SPAM*** I AM LOOKING FOR GOLD DUST BUYER.**
Reply to: mjove_markson3@hotmail.fr ☆
To: undisclosed recipients: ; ☆

I AM LOOKING FOR GOLD DUST BUYER,

Dearest Buyer,

MY NAME IS MR JOVE MARKSON.

I am contacting you for a contract on GOLDDUST,And GOLD BARS, There are bulk of gold dust for sell to interested buyers,each kilo is 3 allthe 9 localmining communities, to sale there gold dust and bars.

If you are interested, you can visit our company and mines; you can see the quantity available and go to refinery to inspect the quality k gold dust to your destination.

1. Gold Dust
 2. 22 Carat plus and Purity 92%
 3. 30,500 USD for one Kg. Bush price
 4. 2500 kilos available.
 5. 650 kgs Reserve for shipment now.
- Origin: Cote D'Ivoire.
Commodity: Aurum Utallum

1. Form: Gold Bar,
2. Purity: 96.4 % like minimum value 96.6% like maximum value.
3. Price :31,500 USD for one kg.

Spam filtering

Example

- Representation of an email through (meaningful) features:

Spam filtering

Example

- Representation of an email through (meaningful) features:
 - ▶ length of subject
 - ▶ length of email body,
 - ▶ use of colors,
 - ▶ domain,
 - ▶ words in subject,
 - ▶ words in body.

length of subject	length of body	use of colors	domain	gold	price	USD	...	machine learning	spam?	
7	240	1	live.fr	1	1	1	...	0	0	1
2	150	0	poznan.pl	0	0	0	...	1	1	0
2	250	0	tibco.com	0	1	1	...	1	1	0
4	120	1	r-project.org	0	1	0	...	0	0	?

Learning

- Four types of datasets:
 - ▶ **training** data: historical emails,
 - ▶ **validation** data: a subset of historical emails used for tuning learning algorithms
 - ▶ **test** data: a subset of historical emails used for estimating the risk,
 - ▶ **new incoming** data to be classified: new incoming emails.

Different learning paradigms

Different learning paradigms

- **Generative learning**

Different learning paradigms

- **Generative learning**

- ▶ Follow a data generating process
- ▶ Learn a model of the joint distribution $P(\mathbf{x}, y)$ and then use the Bayes theorem to obtain $P(y | \mathbf{x})$.
- ▶ Make the final prediction by computing the optimal decision based on $P(y | \mathbf{x})$ with respect to a given $\ell(y, \hat{y})$.

Different learning paradigms

- **Generative learning**

- ▶ Follow a data generating process
- ▶ Learn a model of the joint distribution $P(\mathbf{x}, y)$ and then use the Bayes theorem to obtain $P(y | \mathbf{x})$.
- ▶ Make the final prediction by computing the optimal decision based on $P(y | \mathbf{x})$ with respect to a given $\ell(y, \hat{y})$.

- **Discriminative learning**

Different learning paradigms

- **Generative learning**

- ▶ Follow a data generating process
- ▶ Learn a model of the joint distribution $P(\mathbf{x}, y)$ and then use the Bayes theorem to obtain $P(y | \mathbf{x})$.
- ▶ Make the final prediction by computing the optimal decision based on $P(y | \mathbf{x})$ with respect to a given $\ell(y, \hat{y})$.

- **Discriminative learning**

- ▶ Approximate $h^*(\mathbf{x})$ which is a direct map from \mathbf{x} to y or
- ▶ Model the conditional probability $P(y | \mathbf{x})$ directly, and
- ▶ Make the final prediction by computing the optimal decision based on $P(y | \mathbf{x})$ with respect to a given $\ell(y, \hat{y})$.

Different learning paradigms

- **Generative learning**

- ▶ Follow a data generating process
- ▶ Learn a model of the joint distribution $P(\mathbf{x}, y)$ and then use the Bayes theorem to obtain $P(y | \mathbf{x})$.
- ▶ Make the final prediction by computing the optimal decision based on $P(y | \mathbf{x})$ with respect to a given $\ell(y, \hat{y})$.

- **Discriminative learning**

- ▶ Approximate $h^*(\mathbf{x})$ which is a direct map from \mathbf{x} to y or
- ▶ Model the conditional probability $P(y | \mathbf{x})$ directly, and
- ▶ Make the final prediction by computing the optimal decision based on $P(y | \mathbf{x})$ with respect to a given $\ell(y, \hat{y})$.

- **Two phases** of the learning models: learning and prediction (inference).

Different learning paradigms

- Various principles on how to learn:
 - ▶ Empirical risk minimization,
 - ▶ Maximum likelihood principle,
 - ▶ Bayes approach,
 - ▶ Minimum description length,
 - ▶ ...

Empirical Risk Minimization (ERM)

- Choose a prediction function \hat{h} which minimizes the loss on the training data within some **restricted** class of functions \mathcal{H} .

$$\hat{h} = \arg \min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(y_i, h(\mathbf{x}_i)).$$

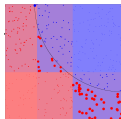
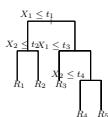
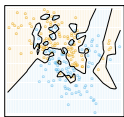
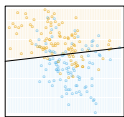
- The average loss on the training data is called **empirical risk** $\hat{L}_\ell(h)$.

Empirical Risk Minimization (ERM)

- Choose a prediction function \hat{h} which minimizes the loss on the training data within some **restricted** class of functions \mathcal{H} .

$$\hat{h} = \arg \min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(y_i, h(\mathbf{x}_i)).$$

- The average loss on the training data is called **empirical risk** $\hat{L}_\ell(h)$.
- \mathcal{H} can be: linear functions, polynomials, trees of a given depth, rules, linear combinations of trees, etc.¹



¹ T. Hastie, R. Tibshirani, and J.H. Friedman. *Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, second edition, 2009

Outline

- 1 Statistical decision theory for supervised learning
- 2 Learning paradigms and principles
- 3 Examples of learning algorithms**
- 4 Summary

Selected learning methods

- Almost no-learning methods: histogram-based classifier, nearest neighbors
- Generative methods: naive Bayes
- Linear methods and their generalizations: linear regression, logistic regression, perceptron, support vector machines, AdaBoost.

Almost no-learning methods

- Based on empirical distribution and direct application of the Bayes rule to a local estimate of $P(y | \mathbf{x})$.

Almost no-learning methods

- Based on empirical distribution and direct application of the Bayes rule to a local estimate of $P(y | \mathbf{x})$.
- The simplest approach estimates conditional probabilities $P(y|\mathbf{x})$ for any \mathbf{x} from training data:
 - ▶ Based on *group-bys* and simple counting.
 - ▶ Needs a lot of data to get reasonable estimates!!!
 - ▶ Data should be discrete/nominal or we need to discretize numerical data before.

Learning

Example

gold	price	spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1 | \text{gold} = 1 \wedge \text{price} = 1) = 0.75$$

$$P(y = 0 | \text{gold} = 1 \wedge \text{price} = 1) = 0.25$$

$$P(y = 1 | \text{gold} = 0 \wedge \text{price} = 0) = 0.33$$

$$P(y = 0 | \text{gold} = 0 \wedge \text{price} = 0) = 0.66$$

$$P(y = 1 | \text{gold} = 0 \wedge \text{price} = 1) = 0.5$$

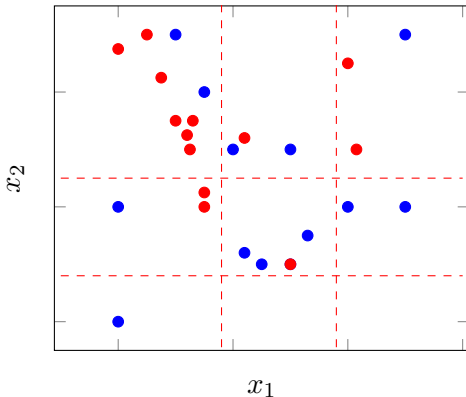
$$P(y = 0 | \text{gold} = 0 \wedge \text{price} = 1) = 0.5$$

$$P(y = 1 | \text{gold} = 1 \wedge \text{price} = 0) = ?$$

$$P(y = 0 | \text{gold} = 1 \wedge \text{price} = 0) = ?$$

Histogram-based methods

- Build a multidimensional grid and estimate the conditional probability in each element of the grid,
- Plug the estimates to the Bayes classifier for a given $\ell(y, \hat{y})$ to obtain prediction.



Histogram-based methods

- The predictive performance depends on the grid resolution, dimensionality of data and the size of training data.

Histogram-based methods

- The predictive performance depends on the grid resolution, dimensionality of data and the size of training data.
- With some tricks can be efficiently implemented.

Histogram-based methods

- The predictive performance depends on the grid resolution, dimensionality of data and the size of training data.
- With some tricks can be efficiently implemented.
- Piecewise-constant prediction for a given region.

Histogram-based methods

- The predictive performance depends on the grid resolution, dimensionality of data and the size of training data.
- With some tricks can be efficiently implemented.
- Piecewise-constant prediction for a given region.
- Computation of the estimates in the region: well-know statistical problem, maximum likelihood estimates, regularization.

Histogram-based methods

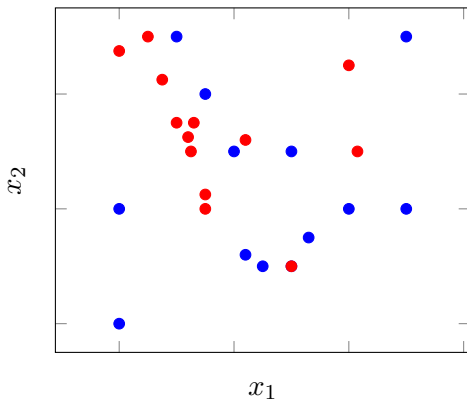
- The predictive performance depends on the grid resolution, dimensionality of data and the size of training data.
- With some tricks can be efficiently implemented.
- Piecewise-constant prediction for a given region.
- Computation of the estimates in the region: well-know statistical problem, maximum likelihood estimates, regularization.
- The grid can be given as a domain knowledge, simple discretization, or random splits.

Histogram-based methods

- The predictive performance depends on the grid resolution, dimensionality of data and the size of training data.
- With some tricks can be efficiently implemented.
- Piecewise-constant prediction for a given region.
- Computation of the estimates in the region: well-know statistical problem, maximum likelihood estimates, regularization.
- The grid can be given as a domain knowledge, simple discretization, or random splits.
- One can use more intelligent methods to obtain a grid, for example, supervised discretization or supervised recursive splitting like in decision trees.

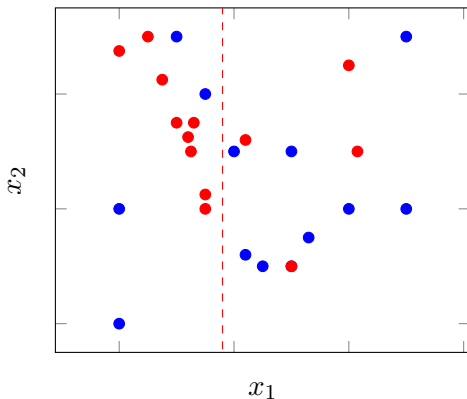
Decision trees

- Recursively make a partition of the feature space (in a smart way),
- Compute the optimal decision in each region.



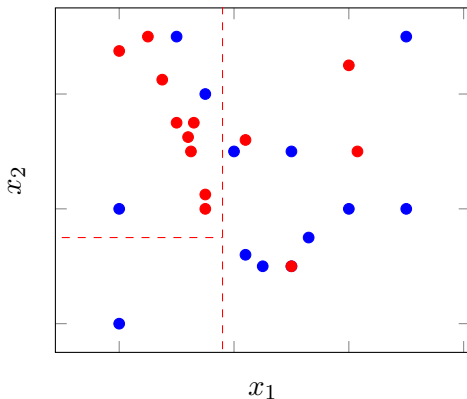
Decision trees

- Recursively make a partition of the feature space (in a smart way),
- Compute the optimal decision in each region.



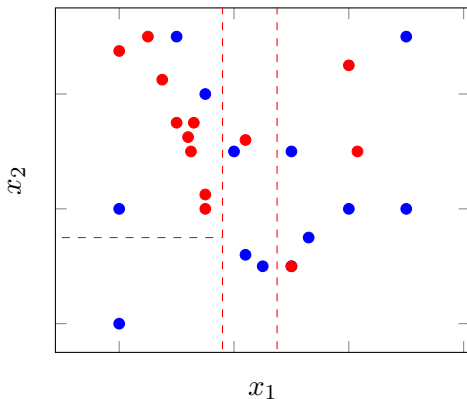
Decision trees

- Recursively make a partition of the feature space (in a smart way),
- Compute the optimal decision in each region.



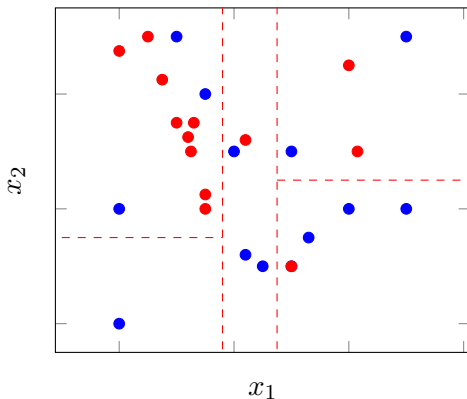
Decision trees

- Recursively make a partition of the feature space (in a smart way),
- Compute the optimal decision in each region.

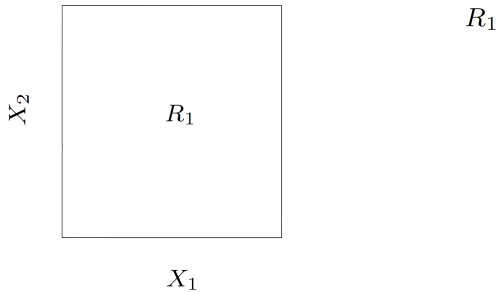


Decision trees

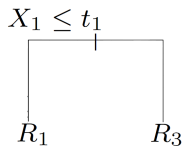
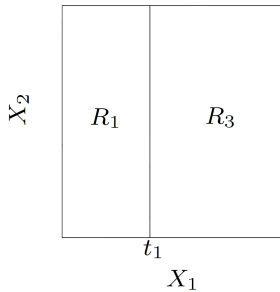
- Recursively make a partition of the feature space (in a smart way),
- Compute the optimal decision in each region.



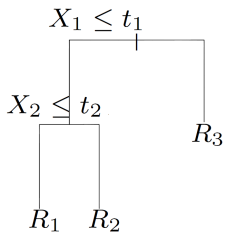
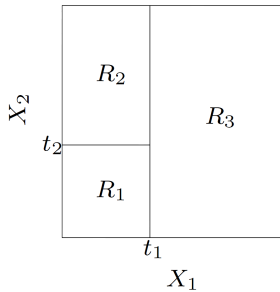
Decision trees



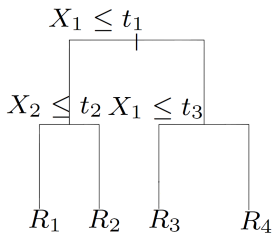
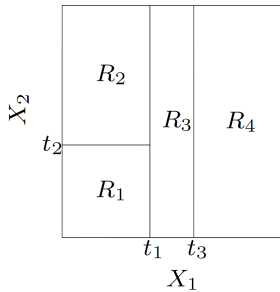
Decision trees



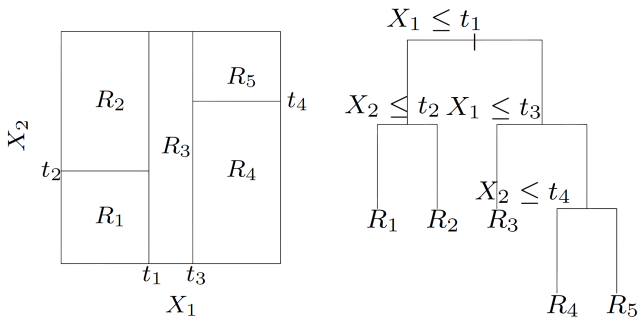
Decision trees



Decision trees



Decision trees



Decision trees

- The learning method seeks an optimal tree shape (e.g. feature space partition) by minimizing the empirical risk (usually expressed in terms of a surrogate loss).

Decision trees

- The learning method seeks an optimal tree shape (e.g. feature space partition) by minimizing the empirical risk (usually expressed in terms of a surrogate loss).
- Greedy methods used for constructing a tree.

Decision trees

- The learning method seeks an optimal tree shape (e.g. feature space partition) by minimizing the empirical risk (usually expressed in terms of a surrogate loss).
- Greedy methods used for constructing a tree.
- The resulting model can be easily interpreted.

Decision trees

- The learning method seeks an optimal tree shape (e.g. feature space partition) by minimizing the empirical risk (usually expressed in terms of a surrogate loss).
- Greedy methods used for constructing a tree.
- The resulting model can be easily interpreted.
- The most influential splits are close to the root (like in the 20-question game).

Decision trees

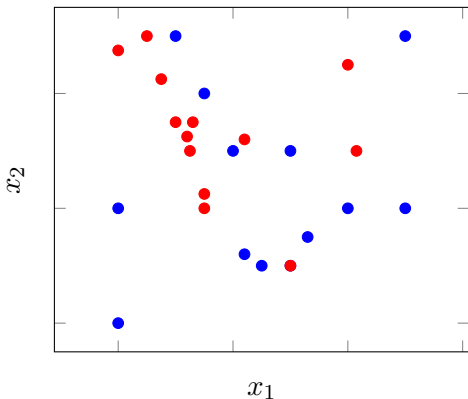
- The learning method seeks an optimal tree shape (e.g. feature space partition) by minimizing the empirical risk (usually expressed in terms of a surrogate loss).
- Greedy methods used for constructing a tree.
- The resulting model can be easily interpreted.
- The most influential splits are close to the root (like in the 20-question game).
- Learning and prediction is very efficient.

Decision trees

- The learning method seeks an optimal tree shape (e.g. feature space partition) by minimizing the empirical risk (usually expressed in terms of a surrogate loss).
- Greedy methods used for constructing a tree.
- The resulting model can be easily interpreted.
- The most influential splits are close to the root (like in the 20-question game).
- Learning and prediction is very efficient.
- Estimation of the decision in each leaf – the same problem like in histogram-based methods.

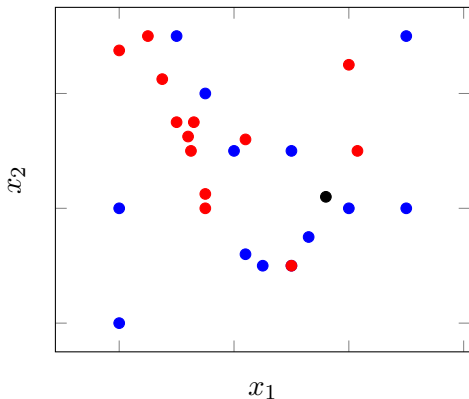
Nearest neighbor methods

- Find k -nearest neighbors of the test example according to a given metric,
- Estimate the Bayes classifier based on the neighborhood.



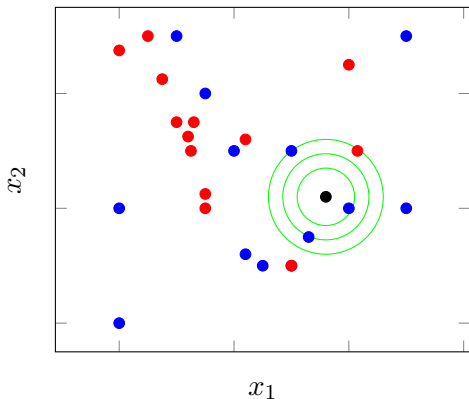
Nearest neighbor methods

- Find k -nearest neighbors of the test example according to a given metric,
- Estimate the Bayes classifier based on the neighborhood.



Nearest neighbor methods

- Find k -nearest neighbors of the test example according to a given metric,
- Estimate the Bayes classifier based on the neighborhood.



Nearest neighbor methods

- Prediction for a test example is computed based on nearest training examples – there is no learning.

Nearest neighbor methods

- Prediction for a test example is computed based on nearest training examples – there is no learning.
- The same principles for computing the prediction as in histogram-based and tree classifiers.

Nearest neighbor methods

- Prediction for a test example is computed based on nearest training examples – there is no learning.
- The same principles for computing the prediction as in histogram-based and tree classifiers.
- Training set can be used for tuning k and finding a metric.

Nearest neighbor methods

- Prediction for a test example is computed based on nearest training examples – there is no learning.
- The same principles for computing the prediction as in histogram-based and tree classifiers.
- Training set can be used for tuning k and finding a metric.
- Specialized data structures for efficient search of nearest neighbors.

Nearest neighbor methods

- Prediction for a test example is computed based on nearest training examples – there is no learning.
- The same principles for computing the prediction as in histogram-based and tree classifiers.
- Training set can be used for tuning k and finding a metric.
- Specialized data structures for efficient search of nearest neighbors.
- Reduction of training data: prototypes, feature selection, dimensionality reduction by PCA or similar methods.

Nearest neighbor methods

- Prediction for a test example is computed based on nearest training examples – there is no learning.
- The same principles for computing the prediction as in histogram-based and tree classifiers.
- Training set can be used for tuning k and finding a metric.
- Specialized data structures for efficient search of nearest neighbors.
- Reduction of training data: prototypes, feature selection, dimensionality reduction by PCA or similar methods.
- Approximate nearest neighbors.

Naive Bayes

- Generative methods rely on the Bayes theorem:

$$P(y = k|\mathbf{x}) = \frac{P(\mathbf{x}|y = k)P(y = k)}{P(\mathbf{x})}$$

where $P(\mathbf{x}|y = k)$ is the density function $f_k(\mathbf{x})$ (for example, multivariate Gaussian distribution), and $P(\mathbf{x})$ is given by:

Naive Bayes

- Generative methods rely on the Bayes theorem:

$$P(y = k|\mathbf{x}) = \frac{P(\mathbf{x}|y = k)P(y = k)}{P(\mathbf{x})}$$

where $P(\mathbf{x}|y = k)$ is the density function $f_k(\mathbf{x})$ (for example, multivariate Gaussian distribution), and $P(\mathbf{x})$ is given by:

$$P(\mathbf{x}) = \sum_j P(\mathbf{x}|y = j)P(y = j)$$

from the law of total probability.

Learning

- The main algorithms:
 - ▶ Linear and quadratic discriminant analysis that use Gaussian densities,
 - ▶ General nonparametric density estimates for each class density,
 - ▶ Naive Bayes model that assumes that each of the class densities are products of marginal densities, i.e., the features are conditionally independent in each class.

Naive Bayes

- The naive Bayes model assumes that given a class $y = k$, the features $\mathbf{x} = (x_1, x_2, \dots, x_m)$ are independent:

$$P(\mathbf{x}|y) =$$

Naive Bayes

- The naive Bayes model assumes that given a class $y = k$, the features $\mathbf{x} = (x_1, x_2, \dots, x_m)$ are independent:

$$P(\mathbf{x}|y) = \prod_{j=1}^m P(x_j|y).$$

Naive Bayes

- The naive Bayes model assumes that given a class $y = k$, the features $\mathbf{x} = (x_1, x_2, \dots, x_m)$ are independent:

$$P(\mathbf{x}|y) = \prod_{j=1}^m P(x_j|y).$$

- The model takes the following form:

$$P(y = k|\mathbf{x}) =$$

Naive Bayes

- The naive Bayes model assumes that given a class $y = k$, the features $\mathbf{x} = (x_1, x_2, \dots, x_m)$ are independent:

$$P(\mathbf{x}|y) = \prod_{j=1}^m P(x_j|y).$$

- The model takes the following form:

$$P(y = k|\mathbf{x}) = \frac{P(y = k) \prod_{j=1}^m P(x_j|y = k)}{\sum_{k'} P(y = k') \prod_{j=1}^m P(x_j|y = k')}$$

Naive Bayes

- The naive Bayes model assumes that given a class $y = k$, the features $\mathbf{x} = (x_1, x_2, \dots, x_m)$ are independent:

$$P(\mathbf{x}|y) = \prod_{j=1}^m P(x_j|y).$$

- The model takes the following form:

$$P(y = k|\mathbf{x}) = \frac{P(y = k) \prod_{j=1}^m P(x_j|y = k)}{\sum_{k'} P(y = k') \prod_{j=1}^m P(x_j|y = k')}$$

- The individual class-conditional marginal densities f_{jk} can each be estimated separately using univariate Gaussian distributions:

$$N(\mathbb{E}(x_j|y = k), \text{Var}(x_j|y = k))$$

- If a component x_j of \mathbf{x} is discrete, then an appropriate histogram estimate can be used.

Naive Bayes

Example

gold price		spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1) =$$

Naive Bayes

Example

gold price		spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1) = 0.5$$

$$P(y = 0) =$$

Naive Bayes

Example

gold price		spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1) = 0.5$$

$$P(y = 0) = 0.5$$

$$P(\text{gold} = 1 | Y = 1) =$$

Naive Bayes

Example

gold price		spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1) = 0.5$$

$$P(y = 0) = 0.5$$

$$P(\text{gold} = 1|Y = 1) = 0.5$$

$$P(\text{gold} = 0|Y = 1) =$$

Naive Bayes

Example

gold price		spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1) = 0.5$$

$$P(y = 0) = 0.5$$

$$P(\text{gold} = 1|Y = 1) = 0.5$$

$$P(\text{gold} = 0|Y = 1) = 0.5$$

$$P(\text{gold} = 1|Y = 0) =$$

Naive Bayes

Example

gold price		spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1) = 0.5$$

$$P(y = 0) = 0.5$$

$$P(\text{gold} = 1|Y = 1) = 0.5$$

$$P(\text{gold} = 0|Y = 1) = 0.5$$

$$P(\text{gold} = 1|Y = 0) = 0.17$$

$$P(\text{gold} = 0|Y = 0) =$$

Naive Bayes

Example

gold price		spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1) = 0.5$$

$$P(y = 0) = 0.5$$

$$P(\text{gold} = 1|Y = 1) = 0.5$$

$$P(\text{gold} = 0|Y = 1) = 0.5$$

$$P(\text{gold} = 1|Y = 0) = 0.17$$

$$P(\text{gold} = 0|Y = 0) = 0.83$$

$$P(\text{price} = 1|Y = 1) =$$

Naive Bayes

Example

gold price		spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1) = 0.5$$

$$P(y = 0) = 0.5$$

$$P(\text{gold} = 1|Y = 1) = 0.5$$

$$P(\text{gold} = 0|Y = 1) = 0.5$$

$$P(\text{gold} = 1|Y = 0) = 0.17$$

$$P(\text{gold} = 0|Y = 0) = 0.83$$

$$P(\text{price} = 1|Y = 1) = 0.66$$

$$P(\text{price} = 0|Y = 1) =$$

Naive Bayes

Example

gold price		spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1) = 0.5$$

$$P(y = 0) = 0.5$$

$$P(\text{gold} = 1|Y = 1) = 0.5$$

$$P(\text{gold} = 0|Y = 1) = 0.5$$

$$P(\text{gold} = 1|Y = 0) = 0.17$$

$$P(\text{gold} = 0|Y = 0) = 0.83$$

$$P(\text{price} = 1|Y = 1) = 0.66$$

$$P(\text{price} = 0|Y = 1) = 0.33$$

$$P(\text{price} = 1|Y = 0) =$$

Naive Bayes

Example

gold price		spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1) = 0.5$$

$$P(y = 0) = 0.5$$

$$P(\text{gold} = 1|Y = 1) = 0.5$$

$$P(\text{gold} = 0|Y = 1) = 0.5$$

$$P(\text{gold} = 1|Y = 0) = 0.17$$

$$P(\text{gold} = 0|Y = 0) = 0.83$$

$$P(\text{price} = 1|Y = 1) = 0.66$$

$$P(\text{price} = 0|Y = 1) = 0.33$$

$$P(\text{price} = 1|Y = 0) = 0.33$$

$$P(\text{price} = 0|Y = 0) =$$

Naive Bayes

Example

gold price		spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1) = 0.5$$

$$P(y = 0) = 0.5$$

$$P(\text{gold} = 1|Y = 1) = 0.5$$

$$P(\text{gold} = 0|Y = 1) = 0.5$$

$$P(\text{gold} = 1|Y = 0) = 0.17$$

$$P(\text{gold} = 0|Y = 0) = 0.83$$

$$P(\text{price} = 1|Y = 1) = 0.66$$

$$P(\text{price} = 0|Y = 1) = 0.33$$

$$P(\text{price} = 1|Y = 0) = 0.33$$

$$P(\text{price} = 0|Y = 0) = 0.66$$

We can, for example, compute:

$$P(y = 1|\text{gold} = 1 \wedge \text{price} = 0) =$$

Naive Bayes

Example

gold price		spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1) = 0.5$$

$$P(y = 0) = 0.5$$

$$P(\text{gold} = 1|Y = 1) = 0.5$$

$$P(\text{gold} = 0|Y = 1) = 0.5$$

$$P(\text{gold} = 1|Y = 0) = 0.17$$

$$P(\text{gold} = 0|Y = 0) = 0.83$$

$$P(\text{price} = 1|Y = 1) = 0.66$$

$$P(\text{price} = 0|Y = 1) = 0.33$$

$$P(\text{price} = 1|Y = 0) = 0.33$$

$$P(\text{price} = 0|Y = 0) = 0.66$$

We can, for example, compute:

$$P(y = 1|\text{gold} = 1 \wedge \text{price} = 0) = \frac{0.5 \times 0.33 \times 0.5}{0.1386} = \frac{0.825}{0.1386} = 0.595$$

$$P(y = 0|\text{gold} = 1 \wedge \text{price} = 0) =$$

Naive Bayes

Example

gold price		spam?
1	1	1
1	1	1
1	1	1
1	1	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	1

$$P(y = 1) = 0.5$$

$$P(y = 0) = 0.5$$

$$P(\text{gold} = 1|Y = 1) = 0.5$$

$$P(\text{gold} = 0|Y = 1) = 0.5$$

$$P(\text{gold} = 1|Y = 0) = 0.17$$

$$P(\text{gold} = 0|Y = 0) = 0.83$$

$$P(\text{price} = 1|Y = 1) = 0.66$$

$$P(\text{price} = 0|Y = 1) = 0.33$$

$$P(\text{price} = 1|Y = 0) = 0.33$$

$$P(\text{price} = 0|Y = 0) = 0.66$$

We can, for example, compute:

$$P(y = 1|\text{gold} = 1 \wedge \text{price} = 0) = \frac{0.5 \times 0.33 \times 0.5}{0.1386} = \frac{0.825}{0.1386} = 0.595$$

$$P(y = 0|\text{gold} = 1 \wedge \text{price} = 0) = 1 - 0.595 = 0.405$$

Naive Bayes

Naive Bayes

- If the independence assumption is not valid, then the model can provide very bad predictions.

Naive Bayes

- If the independence assumption is not valid, then the model can provide very bad predictions.
- In many applications, however, this assumption seems to be at least partially satisfied, for example, in text classification.

Naive Bayes

- If the independence assumption is not valid, then the model can provide very bad predictions.
- In many applications, however, this assumption seems to be at least partially satisfied, for example, in text classification.
- Training is very efficient: one pass over training data to collect all necessary statistics.

Naive Bayes

- If the independence assumption is not valid, then the model can provide very bad predictions.
- In many applications, however, this assumption seems to be at least partially satisfied, for example, in text classification.
- Training is very efficient: one pass over training data to collect all necessary statistics.
- Prediction is linear in number of features.

Naive Bayes

- If the independence assumption is not valid, then the model can provide very bad predictions.
- In many applications, however, this assumption seems to be at least partially satisfied, for example, in text classification.
- Training is very efficient: one pass over training data to collect all necessary statistics.
- Prediction is linear in number of features.
- Some tricks to improve quality of computed statistics: Laplace correction and similar.

Naive Bayes

- If the independence assumption is not valid, then the model can provide very bad predictions.
- In many applications, however, this assumption seems to be at least partially satisfied, for example, in text classification.
- Training is very efficient: one pass over training data to collect all necessary statistics.
- Prediction is linear in number of features.
- Some tricks to improve quality of computed statistics: Laplace correction and similar.

Question

Is Naive Bayes a linear classifier? **Prove** under which conditions it is true.

Linear models

Linear models

- Consider a linear model of the form:

$$f(\mathbf{x}) = w_0 + \sum_{j=1}^m w_j x_j,$$

where $\mathbf{w} = (w_0, w_1, \dots, w_m)$ are the parameters of the model and $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is a feature vector describing an example.

Linear models

- Consider a linear model of the form:

$$f(\mathbf{x}) = w_0 + \sum_{j=1}^m w_j x_j,$$

where $\mathbf{w} = (w_0, w_1, \dots, w_m)$ are the parameters of the model and $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is a feature vector describing an example.

- It is often convenient to use vector notation:

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x},$$

where $\mathbf{x} = (1, x_1, x_2, \dots, x_n)$ has an additional 1 in the first position.

Linear models

- Linear models constitute a very general class of models:
 - ▶ Basic transformations and expansion of original features,
 - ▶ Kernel trick (SVM),
 - ▶ Linear combination of weak classifiers (AdaBoost),
 - ▶ The output layer of neural networks.

Fitting linear models

- We fit parameters \boldsymbol{w} of a linear model using training data

$$\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_n, y_n)\}$$

where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \dots, x_{im})$ is a feature vector of the i -th training example.

Fitting linear models

- We fit parameters \boldsymbol{w} of a linear model using training data

$$\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_n, y_n)\}$$

where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \dots, x_{im})$ is a feature vector of the i -th training example.

- We use loss function $\ell(y, f(\boldsymbol{x}))$ to guide the learning process.

Fitting linear models

- We fit parameters \mathbf{w} of a linear model using training data

$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$$

where $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{im})$ is a feature vector of the i -th training example.

- We use loss function $\ell(y, f(\mathbf{x}))$ to guide the learning process.
- Since direct optimization of $\ell(y, f(\mathbf{x}))$ can be hard (e.g., 0/1 loss is neither convex nor differentiable), we use the so-called surrogate loss functions ℓ_s :

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} \sum_{i=1}^n \ell_s(y_i, \mathbf{w}\mathbf{x}_i)$$

Outline

- 1 Statistical decision theory for supervised learning
- 2 Learning paradigms and principles
- 3 Examples of learning algorithms
- 4 Summary**

Empirical risk minimization

- A wide spectrum of learning algorithms can be given in a general form of surrogate loss minimization:

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^n \ell_s(y_i, f(x))$$

Empirical risk minimization

- A wide spectrum of learning algorithms can be given in a general form of surrogate loss minimization:

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^n \ell_s(y_i, f(x))$$

- The differences between algorithms: form of the surrogate loss, model class, optimization procedure.

Empirical risk minimization

- A wide spectrum of learning algorithms can be given in a general form of surrogate loss minimization:

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^n \ell_s(y_i, f(x))$$

- The differences between algorithms: form of the surrogate loss, model class, optimization procedure.
- This general form allows to compare and analyze learning algorithms.

Problems to be discussed

- Surrogate losses and learning algorithms for linear models.

Problems to be discussed

- Surrogate losses and learning algorithms for linear models.
- Is learning possible?

Problems to be discussed

- Surrogate losses and learning algorithms for linear models.
- Is learning possible?
- Can learning converge to an optimal classifier?

Problems to be discussed

- Surrogate losses and learning algorithms for linear models.
- Is learning possible?
- Can learning converge to an optimal classifier?
- How to solve complex problems such as ranking or multi-label classification?

Summary

- Statistical decision theory for supervised learning.

Summary

- Statistical decision theory for supervised learning.
- Two phases: learning and prediction.

Summary

- Statistical decision theory for supervised learning.
- Two phases: learning and prediction.
- A wide spectrum of learning methods:

Summary

- Statistical decision theory for supervised learning.
- Two phases: learning and prediction.
- A wide spectrum of learning methods:
 - ▶ Histogram-based classifiers,

Summary

- Statistical decision theory for supervised learning.
- Two phases: learning and prediction.
- A wide spectrum of learning methods:
 - ▶ Histogram-based classifiers,
 - ▶ Decision trees,

Summary

- Statistical decision theory for supervised learning.
- Two phases: learning and prediction.
- A wide spectrum of learning methods:
 - ▶ Histogram-based classifiers,
 - ▶ Decision trees,
 - ▶ Nearest neighbors,

Summary

- Statistical decision theory for supervised learning.
- Two phases: learning and prediction.
- A wide spectrum of learning methods:
 - ▶ Histogram-based classifiers,
 - ▶ Decision trees,
 - ▶ Nearest neighbors,
 - ▶ Naive Bayes,

Summary

- Statistical decision theory for supervised learning.
- Two phases: learning and prediction.
- A wide spectrum of learning methods:
 - ▶ Histogram-based classifiers,
 - ▶ Decision trees,
 - ▶ Nearest neighbors,
 - ▶ Naive Bayes,
 - ▶ Linear models.