Classification: Part 2

Statistical Analysis and Document Mining

Spring 2019

Vasilii Feofanov

Université Grenoble Alpes

vasilii.feofanov@univ-grenoble-alpes.fr

Outline

1 Naive Bayes Classification
1.1 Concept
1.2 Particular Cases

2 Classification in Practice
2.1 Performance Estimation
2.2 Script

Classification Task

- *Input space:* $\mathcal{X} \subseteq \mathbb{R}^d$;
- Output space: $\mathcal{Y} = \{-1, +1\}$ (binary classification),
  $\mathcal{Y} = \{1, \ldots, K\}$ (multi-class classification);
- *Assumption:* all $(\mathbf{X}, Y) \in \mathcal{X} \times \mathcal{Y}$ are **i.i.d.** from $\mathcal{D}$ with respect to a fixed unknown probability distribution $P(\mathbf{X}, Y)$;
- *Sample Data:* we observe $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$;

Classification Task

- *Input space:* $\mathcal{X} \subseteq \mathbb{R}^d$;
- Output space: $\mathcal{Y} = \{-1, +1\}$ (binary classification),
  $\mathcal{Y} = \{1, \ldots, K\}$ (multi-class classification);
- *Assumption:* all $(\mathbf{X}, Y) \in \mathcal{X} \times \mathcal{Y}$ are **i.i.d.** from $\mathcal{D}$ with respect to a fixed unknown probability distribution $P(\mathbf{X}, Y)$;
- *Sample Data:* we observe $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$;
- *Loss Function:*

$$\ell^{0/1}(h(\mathbf{x}), y) = \mathbb{I}(h(\mathbf{x}) \neq y) = \begin{cases} 1, & \text{if } h(\mathbf{x}) \neq y; \\ 0, & \text{if } h(\mathbf{x}) = y. \end{cases}$$

- *Target*: minimise the misclassification error:

$$P(h(\mathbf{X}) \neq Y) = \sum_{c \in \{1, \ldots, K\}} P(Y = c) P(h(\mathbf{X}) \neq c | Y = c).$$

$$\underbrace{P(Y|\mathbf{X})}_{\text{}}^{\text{Posterior}} = \frac{\overbrace{P(\mathbf{X}|Y)}^{\text{Likelihood}}\overbrace{P(Y)}^{\text{Class Prior}}}{\underbrace{P(\mathbf{X})}_{\text{Evidence}}}$$

The Bayes classifier predicts a class with the highest posterior probability:

$$h_B(\mathbf{x}) := \operatorname*{argmax}_{y\in\mathcal{Y}} P(Y = y|\mathbf{X} = \mathbf{x}).$$

This is equivalent to:

$$\begin{aligned} h_B(\mathbf{x}) &\propto \operatorname*{argmax}_{y\in\mathcal{Y}} P(\mathbf{X} = \mathbf{x}|Y = y)P(Y = y) \\ &\propto \operatorname*{argmax}_{y\in\mathcal{Y}} \log P(\mathbf{X} = \mathbf{x}|Y = y) + \log P(Y = y). \end{aligned}$$

Naive Bayes Classifier

- *Problem:* When d is large, parametric estimation of $P(\mathbf{X}|Y)$ requires a large number of samples.

Naive Bayes Classifier

- *Problem:* When d is large, parametric estimation of $P(\mathbf{X}|Y)$ requires a large number of samples.
- *Idea:* We assume *naively* that features are conditionally *independent* given the class.

Naive Bayes Classifier

- *Problem:* When d is large, parametric estimation of $P(\mathbf{X}|Y)$ requires a large number of samples.
- *Idea:* We assume *naively* that features are conditionally *independent* given the class.
- Then, denoting $\mathbf{X} = (X_1, \ldots, X_d)$, $\mathbf{x} = (x_1, \ldots, x_d)$, we obtain that:

$$P(\mathbf{X} = \mathbf{x}|Y = c) = P(X_1 = x_1|Y = c) \cdots P(X_d = x_d|Y = c).$$

- *Problem:* When d is large, parametric estimation of $P(\mathbf{X}|Y)$ requires a large number of samples.
- *Idea:* We assume *naively* that features are conditionally *independent* given the class.
- Then, denoting $\mathbf{X} = (X_1, \ldots, X_d)$, $\mathbf{x} = (x_1, \ldots, x_d)$, we obtain that:

$$P(\mathbf{X} = \mathbf{x}|Y = c) = P(X_1 = x_1|Y = c) \cdots P(X_d = x_d|Y = c).$$

- Thus, the *naive Bayes classifier* is defined in the following way:

$$h_B(\mathbf{x}) := \operatorname*{argmax}_{c \in \mathcal{Y}} P(Y = c) \prod_{j=1}^{d} P(X_j = x_j|Y = c)$$

$$\propto \operatorname*{argmax}_{c \in \mathcal{Y}} \log P(Y = c) + \sum_{j=1}^{d} \log P(X_j = x_j|Y = c).$$

Gaussian Naive Bayes Classifier

- We assume that the j-th feature of observations from the class $c \in \mathcal{Y}$ is normally distributed:

$$[X_j | Y = c] \sim \mathcal{N}(\mu_{j,c}, \sigma_{j,c}).$$

- We assume that the j-th feature of observations from the class $c \in \mathcal{Y}$ is normally distributed:

$$[X_j | Y = c] \sim \mathcal{N}(\mu_{j,c}, \sigma_{j,c}).$$

- Then, the distribution of the class $c \in \mathcal{Y}$ is defined as:

$$P(\mathbf{X} = \mathbf{x} | Y = c) = \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi}\sigma_{j,c}} e^{-\frac{(x_j - \mu_{j,c})^2}{2\sigma_{j,c}^2}}.$$

- We assume that the j-th feature of observations from the class $c \in \mathcal{Y}$ is normally distributed:

$$[X_j | Y = c] \sim \mathcal{N}(\mu_{j,c}, \sigma_{j,c}).$$

- Then, the distribution of the class $c \in \mathcal{Y}$ is defined as:

$$P(\mathbf{X} = \mathbf{x} | Y = c) = \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi}\sigma_{j,c}} e^{-\frac{(x_j - \mu_{j,c})^2}{2\sigma_{j,c}^2}}.$$

- Finally, the Gaussian naive Bayes classifier is defined as:

$$h_B(\mathbf{x}) := \underset{c \in \mathcal{Y}}{\operatorname{argmax}} \left[\ln P(Y = c) - \sum_{j=1}^{d} \ln \sigma_{j,c} - \sum_{j=1}^{d} \frac{(x_j - \mu_{j,c})^2}{2\sigma_{j,c}^2} \right].$$

Bernoulli Naive Bayes Classifier

Bernoulli Naive Bayes Classifier

- Each feature X_j is a binary variable ($X_j \in \{0, 1\}$).

Bernoulli Naive Bayes Classifier

- Each feature X_j is a binary variable ($X_j \in \{0, 1\}$).
- Suppose that $X_j|Y = c$ is distributed according to *Bernoulli* distribution so that the probability of success and failure are defined respectively as:

$$P(X_j = 1|Y = c) = p_{j,c};$$
$$P(X_j = 0|Y = c) = 1 - p_{j,c}.$$

Bernoulli Naive Bayes Classifier

- Each feature X_j is a binary variable ($X_j \in \{0,1\}$).
- Suppose that $X_j|Y=c$ is distributed according to *Bernoulli* distribution so that the probability of success and failure are defined respectively as:

$$P(X_j = 1|Y = c) = p_{j,c};$$
$$P(X_j = 0|Y = c) = 1 - p_{j,c}.$$

- Then, the Bernoulli naive Bayes classifier is defined as:

$$h_B(\mathbf{x}) := \operatorname*{argmax}_{c\in\mathcal{Y}} P(Y=c)\prod_{j=1}^{d} p_{j,c}^{x_j}(1-p_{j,c})^{1-x_j}$$

$$\propto \operatorname*{argmax}_{c\in\mathcal{Y}} \log P(Y=c) + \sum_{j=1}^{d} x_j \log p_{j,c} + \sum_{j=1}^{d}(1-x_j)\log(1-p_{j,c}).$$

Multinomial Distribution

Multinomial Distribution

- Let $T_1, \dots, T_m$ be i.i.d. random variables that represent trials with the output $\in \{1, \dots, d\}$ and are distributed as follows:

$$P(T_i = j) = p_j, \quad {\scriptstyle i\in\{1,\dots,m\},\ j\in\{1,\dots,d\},\ \sum_{j=1}^{d} p_j=1.}$$

- Let $T_1, \dots, T_m$ be i.i.d. random variables that represent trials with the output $\in \{1, \dots, d\}$ and are distributed as follows:

$$P(T_i = j) = p_j, \quad {\scriptstyle i \in \{1,\dots,m\},\ j \in \{1,\dots,d\},\ \sum_{j=1}^{d} p_j = 1.}$$

- Random variable X_j is a number of trials with the outcome j:

$$X_j = \sum_{i=1}^{m} \mathbb{I}(T_i = j), \quad {\scriptstyle j \in \{1,\dots,d\},\ \sum_{j=1}^{d} X_j = m.}$$

Multinomial Distribution

- Let $T_1, \ldots, T_m$ be i.i.d. random variables that represent trials with the output $\in \{1, \ldots, d\}$ and are distributed as follows:

$$P(T_i = j) = p_j, \quad {\scriptstyle i \in \{1,\ldots,m\},\ j \in \{1,\ldots,d\},\ \sum_{j=1}^{d} p_j = 1.}$$

- Random variable X_j is a number of trials with the outcome j:

$$X_j = \sum_{i=1}^{m} \mathbb{I}(T_i = j), \quad {\scriptstyle j \in \{1,\ldots,d\},\ \sum_{j=1}^{d} X_j = m.}$$

- Then, the *multinomial* distribution $P(X_1, \ldots, X_d)$ is defined as:

$$P(X_1 = x_1, \ldots, X_d = x_d) = \frac{m!}{x_1! \cdots x_d!} p_1^{x_1} \cdots p_d^{x_d}.$$

- Suppose $\mathbf{X}|Y = c$ is distributed according to the multinomial distribution with parameters $\mathbf{p}_c = (p_{1,c}, \ldots, p_{d,c})$. Denoting $\mathbf{X} = (X_1, \ldots, X_d)$, $\mathbf{x} = (x_1, \ldots, x_d)$, we obtain:

$$P(\mathbf{X} = \mathbf{x}|Y = c) \propto \prod_{j=1}^{d} p_{j,c}^{x_j}.$$

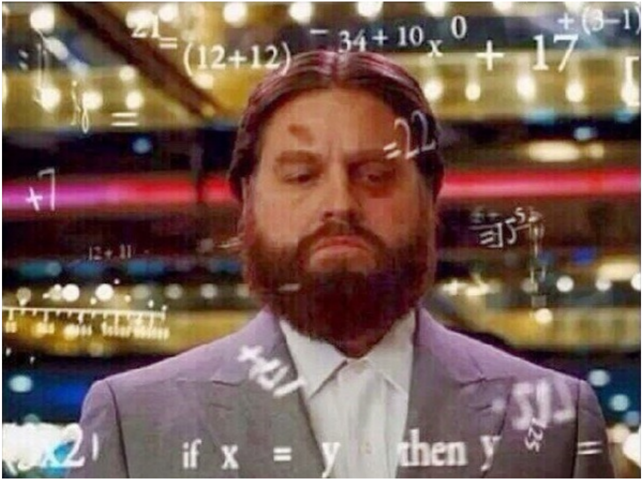
- Suppose $\mathbf{X}|Y = c$ is distributed according to the multinomial distribution with parameters $\mathbf{p}_c = (p_{1,c}, \ldots, p_{d,c})$. Denoting $\mathbf{X} = (X_1, \ldots, X_d)$, $\mathbf{x} = (x_1, \ldots, x_d)$, we obtain:

$$P(\mathbf{X} = \mathbf{x}|Y = c) \propto \prod_{j=1}^{d} p_{j,c}^{x_j}.$$

- Then, we classify by the following rule:

$$h_B(\mathbf{x}) := \operatorname*{argmax}_{c \in \mathcal{Y}} P(Y = c) \prod_{j=1}^{d} p_{j,c}^{x_j}$$

$$\propto \operatorname*{argmax}_{c \in \mathcal{Y}} P(Y = c) \sum_{j=1}^{d} x_j \log p_{j,c}.$$


(12+12)
17
if x = y then y

Next Lecture

Outline

1 Naive Bayes Classification

1.1 Concept

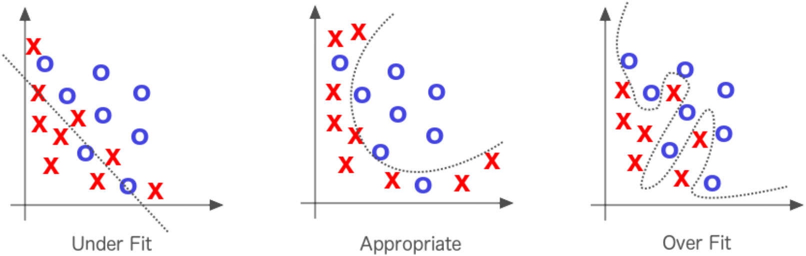
1.2 Particular Cases

2 Classification in Practice

2.1 Performance Estimation

2.2 Script

Performance of an Algorithm



Estimation of the Error Value

Question: How to estimate the error value (e.g. $P(h(\mathbf{X}) \neq Y)$) in practice when the sample data $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$ is available only?

- Error on the training set?

Estimation of the Error Value

Question: How to estimate the error value (e.g. $P(h(\mathbf{X}) \neq Y)$) in practice when the sample data $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$ is available only?

- Error on the training set? $\Rightarrow$ Overfitting.
- Train/test split?

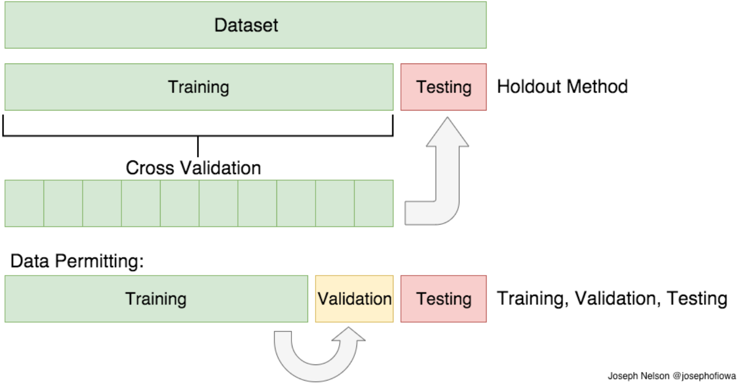
Question: How to estimate the error value (e.g. $P(h(\mathbf{X}) \neq Y)$) in practice when the sample data $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$ is available only?

- Error on the training set? $\Rightarrow$ Overfitting.
- Train/test split? $\Rightarrow$ We should have enough data.
- Cross-validation?

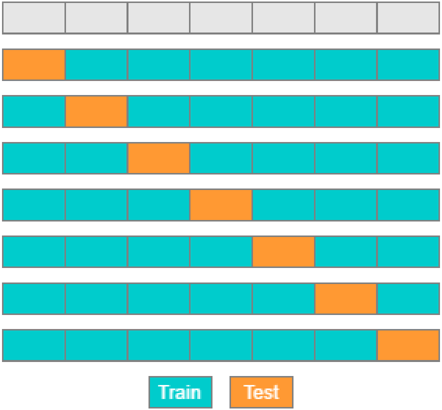
Estimation of the Error Value

Question: How to estimate the error value (e.g. $P(h(\mathbf{X}) \neq Y)$) in practice when the sample data $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$ is available only?

- Error on the training set? $\Rightarrow$ Overfitting.
- Train/test split? $\Rightarrow$ We should have enough data.
- Cross-validation? What is an impact of the number of folds?


Dataset
Training
Testing
Holdout Method
Cross Validation
Data Permitting:
Training
Validation
Testing
Training, Validation, Testing
Joseph Nelson @josephofiowa

Leave-One-Out



n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

Please follow the link:

https://chamilo.grenoble-inp.fr/courses/ENSIMAG4MMSADM/document/DemoR/script_classif_part_I.html

MNIST Features