Master Program in *Data Science and Business Informatics*

Statistics for Data Science

Lesson 23 - Statistical decision theory

Salvatore Ruggieri

Department of Computer Science
University of Pisa, Italy
salvatore.ruggieri@unipi.it

MAP

- Question: which hypothesis is the most probable given the observed data?
  - Maximum Likelihood Estimation (MLE) is a frequentist method:

$$\theta_{MLE} = arg\max_{\theta} P(X_1 = x_1, \ldots, X_n = x_n|\theta) = arg\max_{\theta} \prod_{i=1}^{n} f_{\theta}(x_i)$$

  - Maximum a Posteriori (MAP) is a Bayesian method (requires prior distribution):

$$\theta_{MAP} = arg\max_{\theta} P(\theta|X_1 = x_1, \ldots, X_n = x_n) = arg\max_{\theta} P(X_1 = x_1, \ldots, X_n = x_n|\theta)P(\theta)$$

    - since by the Bayes theorem

$$P(\theta|X_1 = x_1, \ldots, X_n = x_n) = \frac{P(X_1 = x_1, \ldots, X_n = x_n|\theta)P(\theta)}{P(X_1 = x_1, \ldots, X_n = x_n)}$$

  - MAP = MLE if prior is uniform

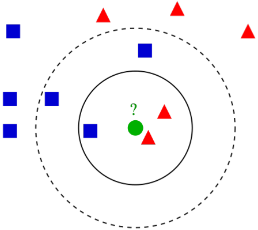
Classification/concept learning

- $X = (W, C)$ where W are predictive features and C class, with $support(C) = \{0, 1, \ldots, n_C - 1\}$
- $x_1, \ldots, x_n$ are observations (training set), with $x_i = (w_i, c_i)$ for $i = 1, \ldots, n$
- $\theta \in \Theta$ with Θ hypothesis space (parameters of ML model) with f_θ joint density of W, C
- Question: which hypothesis (parameters) is the most probable given the observed data?
  - $\theta_{MLE} = arg\ \max_\theta \ell(\theta) = arg\ \min_\theta -\ell(\theta) = arg\ \min_\theta \sum_{i=1}^{n} - \log f_\theta(x_i)$
  - $f_\theta(x_i) = f_\theta(w_i, c_i) = f_\theta(c_i|w_i) f_\theta(w_i)$
  - $\theta_{MLE} = arg\ \min_\theta \sum_{i=1}^{n} - \log f_\theta(c_i|w_i) - \sum_{i=1}^{n} \log f_\theta(w_i)$
  - Assuming $\theta \perp\!\!\!\perp W$, we have $f_{\theta_1}(w_i) = f_{\theta_2}(w_i)$, and then:

$$\theta_{MLE} = arg\ \min_\theta \sum_{i=1}^{n} - \log f_\theta(c_i|w_i)$$

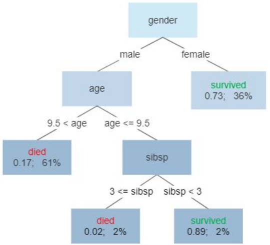
  - How to compute θ_{MLE}? Closed form, brute force enumeration of $\theta \in \Theta$, heuristic search, ...
- $f_\theta(c|w) = P(C = c|W = w, \theta)$ is called a **probabilistic classifier** learned/trained from $x_1, \ldots, x_n$

Probabilistic classifiers: examples



- Logistic regression
- k-Nearest Neighbors (k-NN)
- Decision trees
- Neural networks
- Naive Bayes $P(C = c_0 | W = w) = P(C = c_0) \prod_i P(W_i = w_i | C = c_0) / P(W = w)$ assuming $P(W = w | C = c_0) = \prod_i P(W_i = w_i | C = c_0)$
- Ensembles
- Gradient boosting
- ...
- More classifiers at the Data Mining course

Survival of passengers on the Titanic



See R script

MLE and KL divergence/Cross-Entropy

$$\theta_{MLE} = arg \min_{\theta} \sum_{i=1}^{n} - \log f_{\theta}(c_i|w_i)$$

- Assume data is generated from $f_{\theta_{TRUE}}$, i.e., $(W, C) \sim f_{\theta_{TRUE}}$
- We compute:

$$\theta_{MLE} = arg \min_{\theta} \sum_{i=1}^{n} (-\log f_{\theta}(c_i|w_i) + \log f_{\theta_{TRUE}}(c_i|w_i)) = arg \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \log \frac{f_{\theta_{TRUE}}(c_i|w_i)}{f_{\theta}(c_i|w_i)}$$

$$\xrightarrow{n \to \infty}_{LLN} arg \min_{\theta} E_{(W,C) \sim f_{\theta_{TRUE}}} [log \frac{f_{\theta_{TRUE}}(C|W)}{f_{\theta}(C|W)}] = arg \min_{\theta} D(\theta_{TRUE} \parallel \theta) = arg \min_{\theta} H(\theta_{TRUE}; \theta)$$

- Asymptotically: ML maximization = KL divergence minimization = Cross-entropy minimization

Classification/concept prediction

- Question: which is the most probable prediction for given w and θ?
- Classification/concept prediction problem
  - **Problem**: given $\theta \in \Theta$ and $W = w$, what is the most probable $C = c$? i.e.:

$$arg \max_c P(C = c, W = w|\theta)$$

which is equivalent, assuming $\theta \perp\!\!\!\perp W$, to:

$$arg \max_c P(C = c|W = w, \theta) = arg \max_c f_\theta(c|w)$$

- **Bayes decision rule** $y^*_\theta(w) = arg \max_c f_\theta(c|w)$ *[or simply, y^*]*

Theorem (Bayes decision rule is optimal)

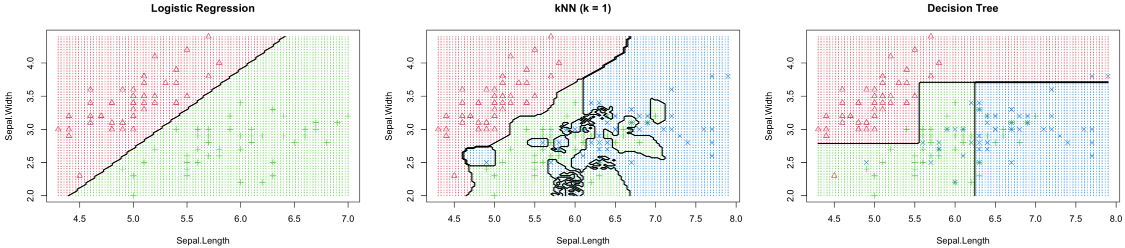
Fix $\theta \in \Theta$. For any decision rule $y^+_\theta : \mathbb{R}^{|W|} \to \{0, \ldots, n_C - 1\}$:

$$P(y^*_\theta(W) \neq C) \leq P(y^+_\theta(W) \neq C)$$

Proof. $P(y^*_\theta(W) = C) = E[\mathbb{1}_{y^*_\theta(W)=C}] = E[E_C[\mathbb{1}_{y^*_\theta(W)=C}|W = w]] \geq$

$$\geq E[E_C[\mathbb{1}_{y^+_\theta(W)=C}|W = w]] = E[\mathbb{1}_{y^+_\theta(W)=C}] = P(y^+_\theta(W) = C)$$

Decision boundary



- A decision boundary for a decision rule $y_\theta^+()$ is the region $w \in \mathbb{R}^{|W|}$ such that $y_\theta^+(w)$ could admit as possible answers two or more classes
- For y_θ^*, it is the region $w \in \mathbb{R}^{|W|}$ such that $\max_c f_\theta(c|w)$ is not unique.
- For y_θ^* and $n_C = 2$, it is the region $w \in \mathbb{R}^{|W|}$ such that $f_\theta(1|w) = 0.5$.

See R script

Bayes optimal predictions

- Question: which is the most probable prediction given w?
- Possible answer: $arg\max_c P(C = c|W = w, \theta_{MAP}) = y^*_{\theta_{MAP}}(w)$
- No, we can do better
  - Let $\Theta = \{\theta_1, \theta_2, \theta_3\}$ and
    - $P(\theta_1|X_1 = x_1, \ldots, X_n = x_n) = 0.4$
    - $P(\theta_2|X_1 = x_1, \ldots, X_n = x_n) = P(\theta_3|X_1 = x_1, \ldots, X_n = x_n) = 0.3$
  - Hence $\theta_{MAP} = \theta_1$
  - Assume $f_{\theta_1}(1|w) = 1$ and $f_{\theta_2}(0|w) = f_{\theta_3}(0|w) = 1$
  - Hence, prediction 0 has the largest probability, whilst prediction of θ_{MAP} is 1
- **Problem**: given $W = w$, what is the most probable $C = c$? i.e.:

$$arg\max_c P(C = c|W = w, X_1 = x_1, \ldots, X_n = x_n)$$

Bayes optimal prediction

$$arg\max_c \sum_{\theta \in \Theta} f_\theta(c|w) P(\theta|X_1 = x_1, \ldots, X_n = x_n)$$

Probabilistic classifier

- Probabilistic classifier: $f_\theta(c|w) \in [0,1]$ with $\sum_c f_\theta(c|w) = 1$:
  - learned from $x_1, \dots, x_n$
  - predicted probabilities $(p_0, \dots, p_{n_C-1})$ with $p_i = f_\theta(i|w)$
  - most probable class $y^*_\theta = arg\max_c f_\theta(c|w)$
  - confidence (of most probable class) $p^*_\theta = \max_c f_\theta(c|w)$
- Unnormalized classifier: $uc_\theta(c|w) \in \mathbb{R}$
  - unnormalized values $(v_0, \dots, v_{n_C-1})$ with $v_i = uc_\theta(i|w)$
  - normalization:

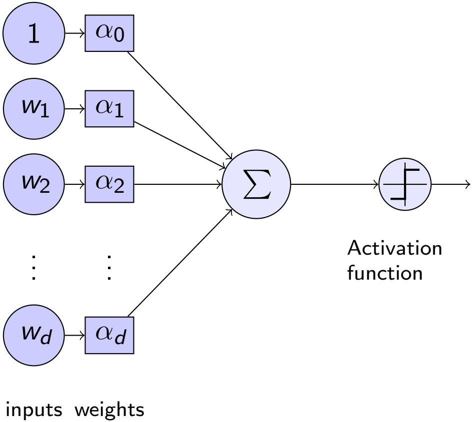
$$softmax((v_0, \dots, v_{n_C-1})) = (\frac{e^{v_0}}{\sum_i e^{v_i}}, \dots, \frac{e^{v_{n_C-1}}}{\sum_i e^{v_i}})$$

  - binary classes $(v_0 = 0, v_1)$:

$$softmax((0, v_1)) = (1 - z, z) \qquad \text{where } z = sigmoid(v_1) = inv.logit(v_1) = \frac{1}{1 + e^{-v_1}}$$

  - $softmax(\mathbf{v} + c) = softmax(\mathbf{v})$
  - $\frac{d}{d\mathbf{v}} softmax(\mathbf{v}) = softmax(\mathbf{v})(1 - softmax(\mathbf{v}))$

Example: Perceptron with sigmoid activation



$$\theta = (\alpha_0, \alpha_1, \dots, \alpha_d)$$

$$w = (w_1, \dots, w_d)$$

$$z = sigmoid(\theta \cdot (1, w)^T) = sigmoid(\alpha_0 + \sum_{i=1}^{d} \alpha_i \cdot w_i)$$

$$y^*_\theta = arg\max\,(1 - z, z)$$

- Difference with logistic regression?
  - Weights calculated differently (MLE vs gradient descent)
  - Perceptron is parametric to **activation functions**
  - Perceptron with sigmoid activation = Logistic regression

Binary classification/concept learning

- $X = (W, C)$ where W are predictive features and C class, with $support(C) = \{0, 1\}$
- $x_1, \ldots, x_n$ are observations (training set), with $x_i = (w_i, c_i)$
- **Definition.** Score function: $s_\theta(w) = f_\theta(1|w) = P(C = 1|W = w, \theta)$
  - predicted probabilities $(1 - s_\theta(w), s_\theta(w))$
  - confidence (of most probable class): $\max\{1 - s_\theta(w), s_\theta(w)\}$
  - $f_\theta(x_i) = f_\theta((w_i, c_i)) = f_\theta(c_i|w_i)f_\theta(w_i) = s_\theta(w_i)^{c_i}(1 - s_\theta(w_i))^{(1-c_i)}P(w_i)$
- MLE estimation

$$\theta_{MLE} = arg\,\min_\theta \sum_{i=1}^{n} -\log f_\theta(c_i|w_i) = arg\,\min_\theta \frac{1}{n}\sum_{i=1}^{n} -c_i \log s_\theta(w_i) - (1 - c_i)\log(1 - s_\theta(w_i))$$

- Cross-entropy loss or log-loss:

$$\ell_\theta(c, w) = \begin{cases} -\log s_\theta(w) & \text{if} c = 1 \\ -\log(1 - s_\theta(w)) & \text{if} c = 0 \end{cases}$$

- MLE maximization = Log-loss minimization

$$\theta_{MLE} = arg\,\min_\theta \frac{1}{n}\sum_{i=1}^{n} \ell_\theta(c_i, w_i)$$

MLE and ERM for classification/concept learning

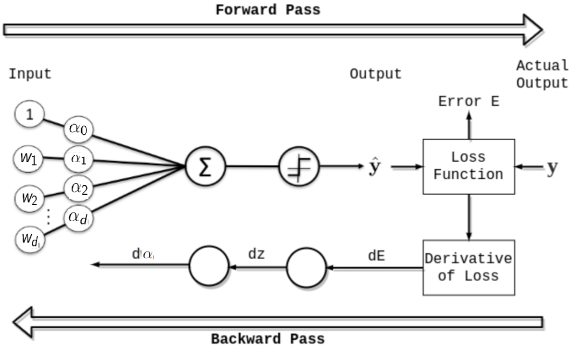
Empirical risk minimization

Let $\ell_\theta : \{0, \dots, n_C - 1\} \times \mathbb{R}^{|W|} \to \mathbb{R}_{\geq 0}$ be a loss function.

$$\theta_{ERM} = arg \min_\theta \frac{1}{n} \sum_{i=1}^{n} \ell_\theta(c_i, w_i)$$

- MLE is ERM with Log-loss $\ell_\theta(c, w) = -\log f_\theta(c|w) = \log \frac{1}{P(c|w,\theta)}$
- 0-1 loss $\ell_\theta(c, w) = \mathbb{1}_{y_\theta^+(w) \neq c}$ where $y_\theta^+(w) \in \{0, \dots, n_C - 1\}$ is a decision rule
  - not convex, not differentiable, optimization problem is NP-hard
- L_p error loss for binary classifiers $\ell_\theta(c, w) = |s_\theta(w) - c|^p$
  - absolute error loss or L_1: $|s_\theta(w) - c|$
  - squared error loss or L_2 or Brier score: $(s_\theta(w) - c)^2$

Loss functions and classifiers



- Gradient of loss function determines updates of weights $\alpha_0, \ldots, \alpha_d$ in the direction of improving the loss (Backpropagation)
- Similar idea in ensamble of decision trees, where each one improves on the error of the previous one (Gradient boosting trees)

Loss functions and margin

- Binary classes $C = \{-1, 1\}$, unnormalized scores $s_\theta(w) \in \mathbb{R}$
  - Bayes decision rule becomes: $y^*_\theta = sgn(s_\theta(w))$
- Margin defined as

$$m = c \cdot s_\theta(w)$$

  - Margin > 0 if prediction is correct (i..e, $s_\theta(w) \geq 0$ and $c = 1$, or if $s_\theta(w) < 0$ and $c = -1$)
  - Loss minimization equivalent to margin maximization
- **Margin-based loss**: Loss function $\ell_\theta(c, w)$ that can be written as $\phi(m)$:
  - 0-1 loss: $\phi(m) = \mathbb{1}_{m \leq 0}$
  - Logistic log-loss: $\phi(m) = \log_2(1 + e^{-m})$
  - L_2 loss: $\phi(m) = (1 - m)^2$
  - SVM/Hinge loss: $\phi(m) = \max\{0, 1 - m\}$
  - AdaBoost/Exponential loss: $\phi(m) = e^{-m}$
- Methods for margin maximization exists for a convex margin-based loss
  - that also provide bounds on 0-1 loss
  - that encode regularizations in the margin-based loss

See R script

MSE and the bias-variance trade-off

- Squared error loss $\theta_{ERM} = arg\min_\theta MSE$, where the Mean Squared Error is:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(s_\theta(w_i) - c_i)^2$$

  - Why named MSE? Because $MSE \xrightarrow{n\to\infty}_{LLN} E_{(W,C)\sim f_{\theta_{TRUE}}}[(s_\theta(W) - C)^2]$
  - MSE approximates the Mean Squared-Error over the population
  - Notice: in MSE for estimators C was a constant (parameter)
- Assumes that $C = D + \epsilon$, where $E[\epsilon] = 0$
  - Observed class labels c_i include some noise w.r.t. true labels, i.e., $c_i = d_i + \epsilon_i$
- Decomposition of MSE:

$$E[MSE] = Var(s_\theta(W)) + E[s_\theta(W) - C]^2 + Var(\epsilon)$$

  - $Var(\epsilon)$ irreducible error
  - $E[s_\theta(W) - C]^2$ is $Bias^2$. Minimized by interpolating training data, but with high variance.
  - $Var(s_\theta(W))$ variance of the scores. Minimized by a constant score, but with high bias.

Loss functions and risk

Consider the squared error loss. For $n \to \infty$:

$$\theta_{ERM} = arg\min_{\theta} \frac{1}{n}\sum_{i=1}^{n}(s_\theta(w) - c_i)^2 \to arg\min_{\theta} E_{(W,C)\sim f_{\theta_{TRUE}}}[(s_\theta(W) - C)^2]$$

Risk (or Expected Prediction Error EPE)

The risk w.r.t. a loss function ℓ_θ is $R(\theta_{TRUE}, \theta) = E_{(W,C)\sim f_{\theta_{TRUE}}}[\ell_\theta(C, W)]$.

Definition. A loss function is a **proper scoring rule** if:

$$\theta_{TRUE} = arg\min_{\theta} R(\theta_{TRUE}, \theta)$$

- For log-loss, $R(\theta_{TRUE}, \theta) = D(\theta_{TRUE} \parallel \theta) \geq 0$ and $D(\theta_{TRUE} \parallel \theta) = 0$ iff $\theta = \theta_{TRUE}$
- Log-loss and L_2 are proper scoring rules, whilst L_1 is not
  - Hence, for squared error loss, for $n \to \infty$, $\theta_{ERM} \to \theta_{TRUE}$

Bayes optimal classifier for 0-1 loss

- Risk of 0-1 loss, binary case. Let $\eta(w) = P_{\theta_{TRUE}}(C = 1|W = w)$, and y^+_θ a decision rule.

$$\begin{aligned}
E_{(W,C)\sim f_{\theta_{TRUE}}}[\mathbb{1}_{y^+_\theta(W)\neq C}] &= E_W[E_C[\mathbb{1}_{y^+_\theta(W)\neq C}|W]] \\
&= E_W[P(C = 1|W) \cdot \mathbb{1}_{y^+_\theta(W)\neq 1} + P(C = 0|W) \cdot \mathbb{1}_{y^+_\theta(W)\neq 0}] \\
&= E_W[\eta(W) \cdot \mathbb{1}_{y^+_\theta(W)=0} + (1 - \eta(W)) \cdot \mathbb{1}_{y^+_\theta(W)=1}] \\
&\geq E_W[\min\{\eta(W), 1 - \eta(W)\}] \\
&= E_W[\eta(W) \cdot \mathbb{1}_{y^*_{\theta_{TRUE}}(W)=0} + (1 - \eta(W)) \cdot \mathbb{1}_{y^*_{\theta_{TRUE}}(W)=1}] \\
&= E_{(W,C)\sim f_{\theta_{TRUE}}}[\mathbb{1}_{y^*_{\theta_{TRUE}}(W)\neq C}] \qquad \textbf{Bayes error rate}
\end{aligned}$$

where $y^*_{\theta_{TRUE}}$ is the **Bayes optimal classifier** (or **Bayes rule**):

$$y^*_{\theta_{TRUE}}(w) = \begin{cases} 1 & \text{if } \eta(w) \geq 1/2 \\ 0 & \text{if } \eta(w) < 1/2 \end{cases}$$

- Hence, $\arg\min_{y^+_\theta} E_{(W,C)\sim f_{\theta_{TRUE}}}[\mathbb{1}_{y^+_\theta(W)\neq C}] = y^*_{\theta_{TRUE}}$
- Optimal decision boundary: $\eta(w) = 1/2$

See R script

Bayes optimal classifier

$$\eta(w) = P_{\theta_{TRUE}}(C = 1|W = w)$$

- $\eta()$ is unknown! (unless we are controlling data generation)
- Naive Bayes $P(C = c_0|W = w) = P(C = c_0)\prod_i P(W_i = w_i|C = c_0)/P(W = w)$ assuming $P(W = w|C = c_0) = \prod_i P(W_i = w_i|C = c_0)$
  - Naive Bayes estimates $\eta(w)$ from empirical distribution of $x_1, \ldots, x_n$
  - and assuming independence of features
- 1-NN asymptotically converges ($|\theta| \to \infty$) to risk: [**Cover and Hart (1967)**]

$$r \leq E_{(W,C)\sim f_{\theta_{TRUE}}}[\mathbb{1}_{y_\theta^{1\text{-}NN}(W)\neq C}] \leq 2r(1-r) \leq 2r$$

  where r is the Bayes error rate.
- Bayes optimal classifier is optimal also for squared loss [**Prove it**]
  - Square loss is convex and differentiable (better use for optimization)

Maximum and Bayes risks

Risk

The risk w.r.t. a loss function ℓ is $R(\theta_{TRUE}, \theta) = E_{(W,C)\sim f_{\theta_{TRUE}}}[\ell_\theta(C, W)]$.

Definition. The maximum risk is

$$\bar{R}(\theta) = \sup_{\theta_{TRUE}} R(\theta_{TRUE}, \theta)$$

A classifier $f_{\theta'}$ such that $\bar{R}(\theta') = \inf_\theta \bar{R}(\theta)$ is called a **minimax rule**.

Definition. Let $f(\theta_{TRUE})$ be a prior for θ_{TRUE}. The Bayes risk is

$$r(\theta) = \int R(\theta_{TRUE}, \theta) f(\theta_{TRUE}) d\theta_{TRUE}$$

A classifier $f_{\theta'}$ such that $r(\theta') = \inf_\theta r(\theta)$ is called a **Bayes rule**.

Classifiers in R: Libraries

The Caret package (Classification And REgression Training)
rpart (Recursive PARTitioning for classification, regression and survival trees)
randomForest (Breiman and Cutler's Random Forests for classification and regression)
lightgbm (LIGHT Gradient Boosting Machine)
fastai (Fast and accurate neural networks training)
kernlab (KERNel-Based Machine Learning LAB)
rminer (Data mining classification and regression methods)

...

See R script