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
Question: which hypothesis is the most probable given the observed data?

- Maximum Likelihood Estimation (MLE) is a frequentist method:
  \[
  \theta_{\text{MLE}} = \arg \max_{\theta} P(X_1 = x_1, \ldots, X_n = x_n|\theta) = \arg \max_{\theta} \prod_{i=1}^{n} f(\theta, x_i)
  \]

- Maximum a Posteriori (MAP) is a Bayesian method (requires prior distribution of \(\theta\)):
  \[
  \theta_{\text{MAP}} = \arg \max_{\theta} P(\theta|X_1 = x_1, \ldots, X_n = x_n) = \arg \max_{\theta} P(X_1 = x_1, \ldots, X_n = x_n|\theta)P(\theta)
  \]
  since by the Bayes theorem
  \[
  P(\theta|X_1 = x_1, \ldots, X_n = x_n) = \frac{P(X_1 = x_1, \ldots, X_n = x_n|\theta)P(\theta)}{P(X_1 = x_1, \ldots, X_n = x_n)}
  \]
  and \(P(X_1 = x_1, \ldots, X_n = x_n)\) does not depend on \(\theta\).

- MAP = MLE if prior is uniform
The classification/concept learning problem

- \( X = (W, C) \) where \( W \) are predictive features and \( C \) class, with \( \text{support}(C) = \{0, 1, \ldots, n_C - 1\} \)
- \( x_1, \ldots, x_n \) are observations (training set), with \( x_i = (w_i, c_i) \) for \( i = 1, \ldots, n \)
- \( \theta \in \Theta \) with \( \Theta \) hypothesis space (parameters of ML model) with \( f_\theta \) joint density of \( W, C \)

**Classification/concept learning**: which hypothesis 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 \( \theta_{MLE} \)? 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

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:

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

\[ \xrightarrow{n \to \infty \text{ LLN}} \arg \min_{\theta} E_{(W, C) \sim f_{\theta_{TRUE}}} \left[ \log \frac{f_{\theta_{TRUE}}(C|W)}{f_{\theta}(C|W)} \right] = \arg \min_{\theta} D_{KL}(\theta_{TRUE} \parallel \theta) = \arg \min_{\theta} H(\theta_{TRUE}; \theta) \]

[See Lesson 11 for \( D_{KL}(\parallel \) \) and \( H(;) \), and the bottom of the R Script in Lesson 19]

- Asymptotically: ML maximization = KL divergence minimization = Cross-entropy minimization
The classification/concept prediction problem

**Question:** which is the most probable class value given \( w \) and \( \theta \)?

**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:

\[
\arg\max_c P(C = c|W = w, \theta) \cdot P(W = w|\theta) = \arg\max_c f_\theta(c|w)
\]

**Bayes decision rule** \( y_\theta^*(w) = \arg\max_c f_\theta(c|w) \quad \text{[or simply, } y^*]\)

---

**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.**

\[
P(y_\theta^*(W) = C) = E[\mathbbm{1}_{y_\theta^*(W) = C}] = E[E_C[\mathbbm{1}_{y_\theta^*(W) = C}|W = w]] \geq
\]

\[
\geq E[E_C[\mathbbm{1}_{y_\theta^+(W) = C}|W = w]] = E[\mathbbm{1}_{y_\theta^+(W) = C}] = P(y_\theta^+(W) = C)
\]
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 $y^*_{\theta}$, it is the region $w \in \mathbb{R}^{|W|}$ such that $\arg\max_c f_{\theta}(c|w)$ is not unique.

For $y^*_{\theta}$ 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 class value given $w$ only (i.e., without fixing the parameters)?

- Possible answer: the prediction of the most probable model, i.e., $\arg \max_c P(C = c|W = w, \theta_{MAP})$
- 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, class 0 has the largest probability (over the hypothesis space), whilst $\theta_{MAP}$ predicts 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

$$\arg \max_c \sum_{\theta \in \Theta} f_{\theta}(c|w)P(\theta|X_1 = x_1, \ldots, X_n = x_n)$$
No-Free-Lunch theorem

- A learner $\mathcal{A}$ is a computable function that maps a training set $x_1, \ldots, x_n$ into a decision rule $y_\theta()$

**Question:** Is there a learner $\mathcal{A}$ that always maps a training set into a decision rule with zero error?

### No-Free-Lunch theorem (Wolpert, 1996)

Consider binary classification, i.e., $n_C = 2$, and a finite domain $\text{dom}(W) < \infty$. For any learner $\mathcal{A}$, there exists a distribution $F$ with $(W, C) \sim F$ such that:

- for at least $1/7$ of the training sets $x_1, \ldots, x_n$ (realizations of $F^n$) with $n < \frac{\text{dom}(W)}{2}$, the decision rule $y_\theta^+$ in output by $\mathcal{A}$ has an error of at least $1/8$, i.e.:

  $$P_F(y_\theta^+(W) \neq C) \geq 1/8$$

- and there exists an error-free decision rule $y_\theta^*$ s.t. $P_F(y_\theta^*(W) \neq C) = 0$.

See here for an accessible proof

- A universal learner does no exist! No learner can succeed on all learning tasks: every learner has tasks on which it fails whereas other learners succeed.

- The learnt $y_\theta^+$ is likely to have a large error for $F$, whereas there exists another learner that will output a decision rule $y_\theta^*$ with no error.
Probabilistic classifiers

- Probabilistic classifier: \( f_\theta(c|w) \in [0, 1] \) with \( \sum_c f_\theta(c|w) = 1 \):
  - learned from \( x_1, \ldots, x_n \)
  - predicted probabilities \( (p_0, \ldots, p_{n_c-1}) \) 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 \( (v_0, \ldots, v_{n_c-1}) \) with \( v_i = uc_\theta(i|w) \)
  - normalization using softmax:
    \[
    \text{softmax}((v_0, \ldots, v_{n_c-1})) = \left( \frac{e^{v_0}}{\sum_i e^{v_i}}, \ldots, \frac{e^{v_{n_c-1}}}{\sum_i e^{v_i}} \right)
    \]
  - binary classes \( (v_0 = 0, v_1) \):
    \[
    \text{softmax}((0, v_1)) = (1 - z, z) \quad \text{where} \quad z = \text{sigmoid}(v_1) = \text{inv.logit}(v_1) = \frac{1}{1 + e^{-v_1}}
    \]
  - \( \text{softmax}(\mathbf{v} + c) = \text{softmax}(\mathbf{v}) \)
  - \( \frac{d}{dv} \text{softmax}(\mathbf{v}) = \text{softmax}(\mathbf{v})(1 - \text{softmax}(\mathbf{v})) \)
**Example: Perceptron with sigmoid activation**

\[ \theta = (\alpha_0, \alpha_1, \ldots, \alpha_d) \]
\[ w = (w_1, \ldots, w_d) \]
\[ z = \text{sigmoid}(\theta \cdot (1, w)^T) = \text{sigmoid}(\alpha_0 + \sum_{i=1}^{d} \alpha_i \cdot w_i) \]

\[ y_\theta^* = \text{arg max} \ (1 - z, z) \]

- 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 \( \text{support}(C) = \{0, 1\} \)
- \( x_1, \ldots, x_n \) 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): \( \max\{1 - s_\theta(w), s_\theta(w)\} \)
  - \( f_\theta(c_i|w_i) = s_\theta(w_i)^{c_i}(1 - s_\theta(w_i))^{(1-c_i)} \)
- **MLE estimation**
  \[
  \theta_{\text{MLE}} = \arg\min_\theta \sum_{i=1}^{n} -\log f_\theta(c_i|w_i) = \arg\min_\theta \frac{1}{n} \sum_{i=1}^{n} -c_i \log s_\theta(w_i) - (1 - c_i) \log (1 - s_\theta(w_i))
  \]
- **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**
  \[
  \theta_{\text{MLE}} = \arg\min_\theta \frac{1}{n} \sum_{i=1}^{n} \ell_\theta(c_i, w_i)
  \]
MLE and ERM for classification/concept learning

**Empirical risk minimization**

Let $\ell_\theta : \{0, \ldots, n_C - 1\} \times \mathbb{R}^{|W|} \to \mathbb{R}_{\geq 0}$ 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 $\ell_\theta(c, w) = \mathbb{I}_{y_\theta^+(w) \neq c}$ where $y_\theta^+(w) \in \{0, \ldots, n_C - 1\}$ 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: $(s_\theta(w) - c)^2$
• Gradient of loss function determines updates of weights $\alpha_0, \ldots, \alpha_d$ in the direction of improving the loss (Backpropagation)

• Similar idea in ensemble of decision trees, where each one improves on the error of the previous one (Gradient boosting trees)
MSE and the bias-variance trade-off

• Squared error loss \( \theta_{ERM} = \arg \min_\theta \text{MSE} \), where the Mean Squared Error is:

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

▶ Why named \( \text{MSE} \)? Because \( \text{MSE} \xrightarrow{n\to\infty} \text{LLN} \ E(W,C) \sim f_{\theta, \text{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)

[See Lesson 18]

• 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(W,C) \sim f_{\theta, \text{TRUE}} \[(s_\theta(W) - C)^2\] = \text{Var}(s_\theta(W)) + E[s_\theta(W) - C]^2 + \text{Var}(\epsilon)
\]

▶ \( \text{Var}(\epsilon) \) irreducible error (would require better curated class values in the training set)
▶ \( E[s_\theta(W) - C]^2 \) is \( \text{Bias}^2 \). Minimized by interpolating training data, but with high variance.
▶ \( \text{Var}(s_\theta(W)) \) variance of the scores. Minimized by a constant score, but with high bias.

See R script
Loss functions and risk

Squared error loss minimization on training set generalizes to the population:

\[
\theta_{ERM} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} (s_\theta(w) - c_i)^2 \xrightarrow{n \to \infty} \text{LLN} \arg\min_{\theta} \mathbb{E}_{(W, C) \sim f_{\theta_{\text{TRUE}}}} [(s_\theta(W) - C)^2]
\]

Risk (or Expected Prediction Error EPE)

The risk w.r.t. a loss function \( \ell_\theta \) is

\[
R(\theta_{\text{TRUE}}, \theta) = \mathbb{E}_{(W, C) \sim f_{\theta_{\text{TRUE}}}} [\ell_\theta(C, W)].
\]

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

\[
\theta_{\text{TRUE}} = \arg\min_{\theta} R(\theta_{\text{TRUE}}, \theta)
\]

- For log-loss, \( R(\theta_{\text{TRUE}}, \theta) = D_{KL}(\theta_{\text{TRUE}} \parallel \theta) \geq 0 \) and \( D_{KL}(\theta_{\text{TRUE}} \parallel \theta) = 0 \) iff \( \theta = \theta_{\text{TRUE}} \)
- Log-loss, squared error (\( L_2 \)) and 0-1 loss are proper scoring rules, whilst \( L_1 \) is not
  - For proper scoring rules, \( \theta_{ERM} \xrightarrow{n \to \infty} \theta_{\text{TRUE}} \) – recall we assume such \( (W, C) \sim f_{\theta_{\text{TRUE}}} \) exists
  - Still, 0-1 loss is discontinuous and can be harmful!
Question: what is the decision rule with the smallest 0-1 risk? i.e., \[ \arg \min_{y_\theta^+} E(W, C \sim f_{\theta \text{TRUE}}) [\mathbb{1}_{y_\theta^+(W) \neq C}] \]

Binary class Bayes optimal classifier (or Bayes rule):

\[
y_{\theta \text{TRUE}}^*(w) = \begin{cases} 
1 & \text{if } \eta(w) \geq 1/2 \\
0 & \text{if } \eta(w) < 1/2 
\end{cases}
\]

where \( \eta(w) = P_{\theta \text{TRUE}}(C = 1|W = w) \).

\[
E(W, C \sim f_{\theta \text{TRUE}}) [\mathbb{1}_{y_\theta^+(W) \neq C}] = E_W[E_C[\mathbb{1}_{y_\theta^+(W) \neq C}|W]] \\
= E_W[P(C = 1|W) \cdot \mathbb{1}_{y_\theta^+(W) \neq 1} + P(C = 0|W) \cdot \mathbb{1}_{y_\theta^+(W) = 0}] \\
= E_W[\eta(W) \cdot \mathbb{1}_{y_\theta^+(W) = 0} + (1 - \eta(W)) \cdot \mathbb{1}_{y_\theta^+(W) = 1}] \\
\geq E_W[\min \{\eta(W), 1 - \eta(W)\}] \\
= E_W[\eta(W) \cdot \mathbb{1}_{y_{\theta \text{TRUE}}^*(W) = 0} + (1 - \eta(W)) \cdot \mathbb{1}_{y_{\theta \text{TRUE}}^*(W) = 1}] \\
= E(W, C \sim f_{\theta \text{TRUE}}) [\mathbb{1}_{y_{\theta \text{TRUE}}^*(W) \neq C}] \quad \text{Bayes error rate}
\]

See R script
Bayes optimal classifier

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

- \( \eta() \) is unknown! (unless we are controlling data generation)
- **Plug-in rule:** use \( \hat{\eta}(w) = f_{\theta}(C = 1|W = w) \) as an estimate of \( \eta(w) \)
- 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: \[ \begin{align*}
  r &\leq E_{(W,C)\sim f_{\theta_{TRUE}}}[\mathbb{I}_{y^{1-\text{NN}}(W)\neq C}] \\
  &\leq 2r(1-r) \leq 2r
\end{align*} \]
  where \( r \) is the Bayes error rate.
- Bayes optimal classifier is optimal also for squared loss \[ \text{Prove it} \]
  - Squared loss is convex and differentiable (good for optimization solving)
Maximum and Bayes risks

Risk (or Expected Prediction Error EPE)

The risk w.r.t. a loss function $\ell_\theta$ is

$$R(\theta_{\text{TRUE}}, \theta) = E_{(W, C) \sim f_{\theta_{\text{TRUE}}}}[\ell_\theta(C, W)].$$

Risk is defined w.r.t. a specific $\theta_{\text{TRUE}}$. What is the maximum risk at the variation of $\theta_{\text{TRUE}}$?

**Definition.** The maximum risk is

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

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

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

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

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

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

- Margin for $(w, c)$ 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: $\phi(m) = \mathbb{I}_{m \leq 0}$
  - 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
The Caret package (Classification And REgression Training)
npart (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)

...
Reject option in binary classification

\[ \eta(w) = P_{\theta_{\text{TRUE}}}(C = 1|W = w) \]

Bayes optimal classifier (or Bayes rule):

\[ y_{\theta_{\text{TRUE}}}(w) = \begin{cases} 
1 & \text{if } \eta(w) \geq \frac{1}{2} \\
0 & \text{if } \eta(w) < \frac{1}{2}
\end{cases} \]

- If \( \eta(w) \approx \frac{1}{2} \), we might just as well toss a coin to make a decision
- This motivates the introduction of a reject option for classifiers
  - reject, or abstain, expressing doubt or uncertainty in decisions
  - relevant in practice (e.g., to understand the cases where a classifier performs poorly),
  - relevant ethically for socially sensitive decision tasks (e.g., credit scoring, disease prediction, CV screening, etc.)
Reject option in binary classification

\[ \eta(w) = P_{\theta_{\text{TRUE}}}(C = 1|W = w) \]

**Bayes optimal classifier (with reject option):**

\[ y_{\theta_{\text{TRUE}}}^{*,d}(w) = \begin{cases} 
1 & \text{if } \eta(w) > 1 - d \\
0 & \text{if } \eta(w) < d \\
\text{abstain} & \text{otherwise, i.e., } d \leq \min \{\eta(w), 1 - \eta(w)\} 
\end{cases} \]

where \( d \in [0, \frac{1}{2}] \) is the reject cost.

- If \( y_{\theta_{\text{TRUE}}}^{*,d}(w) \neq \text{abstain} \) [\( d \) upper bound on misclassification error]

\[ d > \min \{\eta(w), 1 - \eta(w)\} = P_{\theta_{\text{TRUE}}}(y_{\theta}^*(w) \neq C) \] [error of Bayes optimal]

**Theorem (Chow 1970).**

\[ \arg \min_{y_{\theta}^{\text{+}}} \mathbb{E}_{(W,C) \sim f_{\theta_{\text{TRUE}}}}[d \mathbb{1}_{y_{\theta}^{\text{+}}(W) = \text{abstain}} + \mathbb{1}_{y_{\theta}^{\text{+}}(W) \neq C \cdot y_{\theta}^{\text{+}}(W) \neq \text{abstain}}] = y_{\theta_{\text{TRUE}}}^{*,d} \]
Selective binary classification

A **selective binary classifier** (score) is a pair \((s_\theta, g_\theta)\), where \(s_\theta()\) is a classifier (score) and \(g_\theta : \mathbb{R}^{|W|} \rightarrow \{0, 1\}\) is a **selection function**, which determines when to accept/abstain from using \(s_\theta\):

\[
(s_\theta, g_\theta)(w) = \begin{cases} 
    s_\theta(w) & \text{if } g_\theta(w) = 1 \\
    \text{abstain} & \text{otherwise}
\end{cases}
\]

**Support and Risk**

The **coverage** of a selective classifier is \(\phi(g_\theta) = E_{(W,C) \sim f_{\theta\text{TRUE}}}[g_\theta(W)]\), i.e., the expected probability of the accepted region.

The risk w.r.t. a loss function \(\ell_\theta\) is \(R(s_\theta, g_\theta) = E_{(W,C) \sim f_{\theta\text{TRUE}}}[\ell_\theta(C, W) g_\theta(W)] / \phi(g_\theta)\).

- **Empirical coverage and empirical selective risk:**

  \[
  \hat{\phi}(g_\theta) = \frac{\sum_{i=1}^n g_\theta(w_i)}{n} \quad \hat{r}(s_\theta, g_\theta) = \frac{1}{n} \sum_{i=1}^n \frac{\ell_\theta(c_i, w_i) g_\theta(w_i)}{\hat{\phi}(g_\theta)}
  \]

- **Selective classification problem:** minimize risk while guaranteeing a minimum support \(c\)

  \[
  \arg \min_{\theta} R(s_\theta, g_\theta) \quad \text{s.t.} \quad \phi(g_\theta) \geq c
  \]
Soft selective binary classification

A soft selective binary classifier:

\[(s_\theta, g_\theta)(w) = \begin{cases} s_\theta(w) & \text{if } k_\theta(w) \geq \tau \\ \text{abstain} & \text{otherwise} \end{cases}\]

- \(k_\theta(w)\) is called the confidence function
  - A good confidence function should rank instances based on descending loss, i.e., if \(k(w) \leq k(w')\) then \(E[\ell_\theta(C, w)] \geq E[\ell_\theta(C, w')]\).
- Confidence of the classifier (see slide 10) and \(\tau \in [0.5, 1]\):
  \[k_\theta(w) = \max\{s_\theta(w), 1 - s_\theta(w)\}\]
- The inherent trade-off between risk and coverage is summarized by the risk-coverage curve