Bayes decisions and Model evaluation

Sandro Cumani

sandro.cumani@polito.it

Politecnico di Torino

Model evaluation for classification

Assess how good our model is on a held–out evaluation (test) set

We start considering binary classification problems

A possible solution is to compute the accuracy of the model, or, equivalently, the error rate, defined as

$$accuracy = \frac{\#\ of\ correctly\ classified\ samples}{\#\ of\ samples}$$

$$error\ rate = \frac{\#\ of\ incorrectly\ classified\ samples}{\#\ of\ samples} = 1 - accuracy$$

Model evaluation for classification

Accuracy can be misleading if the classes are not balanced

Let's consider, for example, rain prediction in arid climates. Over one year, the model makes the following predictions:

	Rain	Clear
Prediction: Rain	15 days	30 days
Prediction: Clear	20 days	300 days

$$accuracy = \frac{300 + 15}{365} \approx 86\%$$

Model evaluation for classification

Accuracy can be misleading if the classes are not balanced

Let's consider, for example, rain prediction in arid climates. Over one year, the model makes the following predictions:

	Rain	Clear
Prediction: Rain	15 days	30 days
Prediction: Clear	20 days	300 days

$$accuracy = \frac{300 + 15}{365} \approx 86\%$$

86% looks like a good accuracy ... But a model that always predicts Clear would achieve an accuracy of $\approx 90\%$!

Model evaluation for classification

Let's analyze the outcomes table

	Rain	Clear
Prediction: Rain	15 days	30 days
Prediction: Clear	20 days	300 days

This table is also called confusion matrix. In general:

	Class $\mathcal{H}_F$	Class $\mathcal{H}_T$
Prediction $\mathcal{H}_F$	True Negative	False Negative
Prediction $\mathcal{H}_T$	False Positive	True Positive

Model evaluation for classification

We can compute different accuracy measures

- False negative rate FNR (false rejection / miss rate): $\frac{FN}{FN+TP}$
- False positive rate FPR (false acceptance): $\frac{FP}{FP+TN}$
- True positive rate TPR (recall, sensitivity): $\frac{TP}{FN+TP} = 1 - \text{FNR}$
- True negative rate TNR (specificity): $\frac{TN}{FP+TN} = 1 - \text{FPR}$
- ...

We can also compute a *weighted* accuracy

$$acc = \alpha FPR + (1 - \alpha) FNR$$

The weight α measures how important are different kind of errors (we shall see this in more detail later)

Model evaluation for classification

Different kind of errors may have different impact on applications

Systems providing only hard decisions do not allow for trade-offs between different error types

Rather than labels, often classifiers output scores

- Generative models: log–likelihood ratios

$$s = \log \frac{f(x|\mathcal{H}_T)}{f(x|\mathcal{H}_F)}$$

- Discriminative models: posterior log–likelihood ratios

$$s = \log \frac{P(\mathcal{H}_T|x)}{P(\mathcal{H}_F|x)}$$

- Non–probabilistic models: score (e.g. SVM)

$$s = \boldsymbol{w}^T \boldsymbol{x}$$

Model evaluation for classification

A higher score means we should favor class $\mathcal{H}_T$

Class assignment is performed by comparing scores to a threshold t

$$s \geq t \longrightarrow \mathcal{H}_T$$

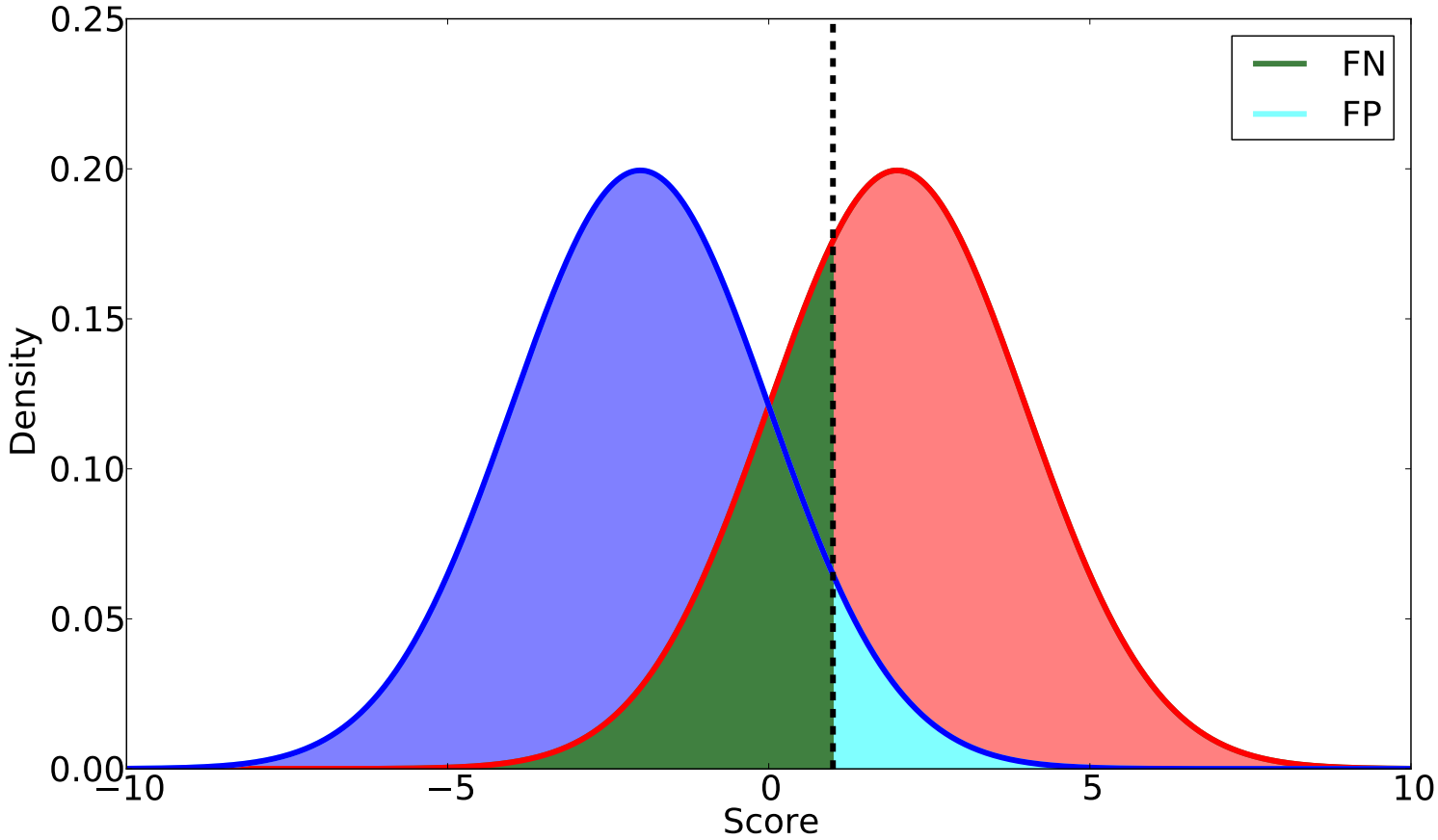
$$s < t \longrightarrow \mathcal{H}_F$$

Different thresholds correspond to different error rates

Thresholds are related to class priors and error costs

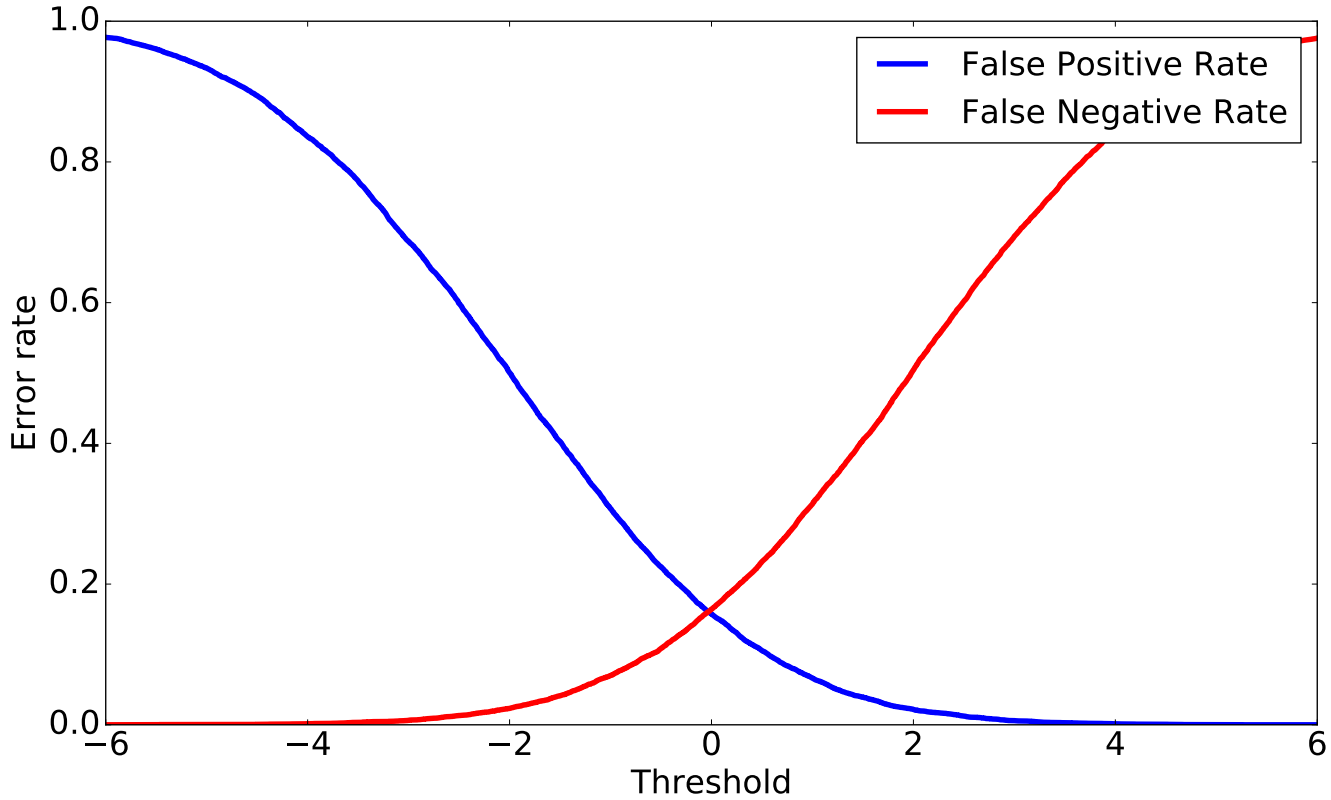
Model evaluation for classification

Score thresholding:



Model evaluation for classification

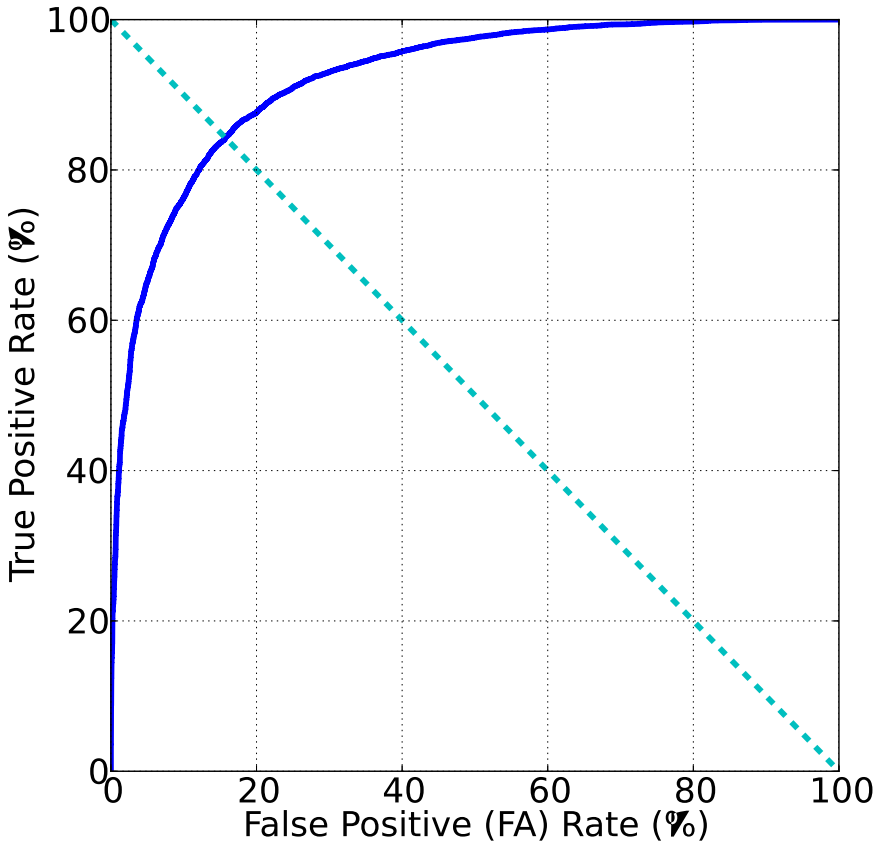
We can visualize the performance of the classifier for different thresholds by plotting the error rates as a function of the threshold



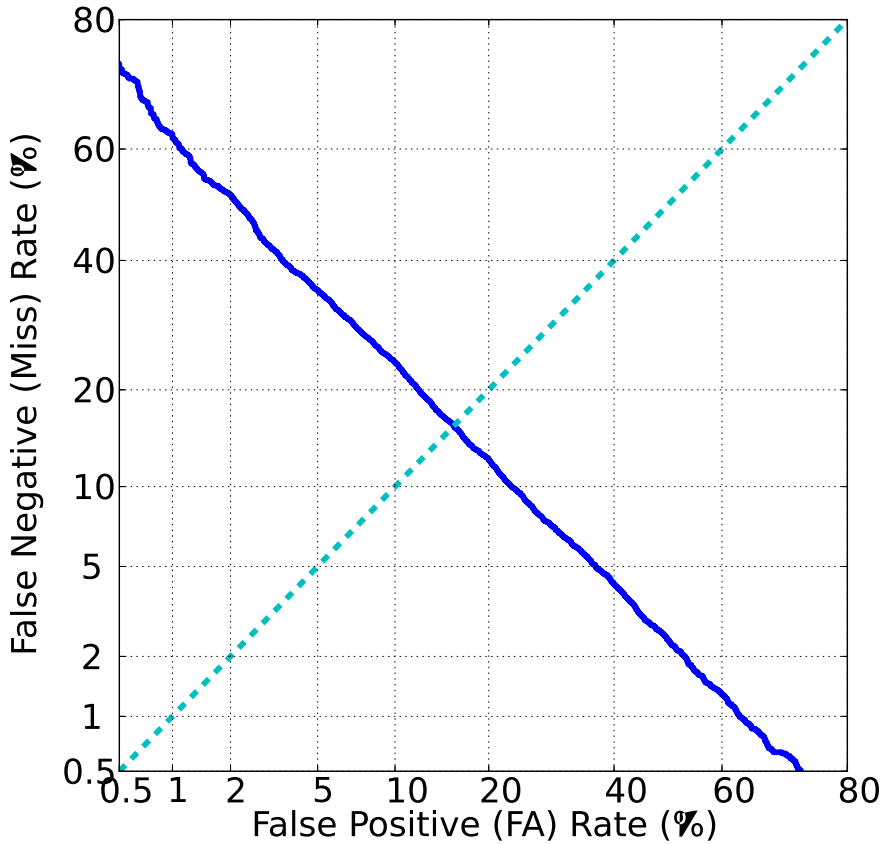
Equal Error Rate (EER): The error rate for which $FPR = FNR$

Model evaluation for classification

- Receiver Operating Characteristic (ROC) curve



- Detection Error Trade-off (DET) curve



Bayes decisions

The goal of the classifier is to allow us to choose an action a to perform among a set of actions $\mathcal{A}$

Example: accepting vs rejecting a sample

Example: assign label a to the sample

We can associate to each action a cost $\mathcal{C}(a|k)$ that we have to pay when we choose action a and the sample belongs to class k

In the following we consider the set of actions corresponding to labeling a sample with label a

Bayes decisions

$\mathcal{C}(a|k)$ represents the cost of labeling the sample as belonging to class a when it actually belongs to class k

We do not know k, however we have a classifier $\mathcal{R}$ that allows us computing class posterior probabilities $P(C = k|x, \mathcal{R})$ for sample x

We can thus compute the expected cost of action a when the posterior probability for each class is $P(C = k|x, \mathcal{R})$

$$\mathcal{C}_{x,\mathcal{R}}(a) = \mathbb{E}[\mathcal{C}(a|k)|x, \mathcal{R}] = \sum_{k=1}^{K} \mathcal{C}(a|k)P(C = k|x, \mathcal{R})$$

It measures the cost that we expect to pay given our knowledge of the class distribution $P(C = k|x, \mathcal{R})$

Bayes decisions

The Bayes decision consists in choosing the action $a^*(x, \mathcal{R})$ that minimizes the expected cost: $a^*(x, \mathcal{R}) = \arg\min_a \mathcal{C}_{x,\mathcal{R}}(a)$

It represents the action that will result in the lower expected cost, according to the recognizer beliefs

Different recognizers may have different posterior beliefs, and thus provide different decisions

Bayes decisions

For example, let's consider we have a 3-class problem, with cost matrix and priors given by

$$\boldsymbol{C} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix}, \quad \boldsymbol{\pi} = \begin{bmatrix} 0.3 \\ 0.4 \\ 0.3 \end{bmatrix}$$

For a test sample $\boldsymbol{x}_t$, we have computed the posterior class probabilities (using the prior $\boldsymbol{\pi}$)

$$\boldsymbol{q}_t = \begin{bmatrix} P(C=1|\boldsymbol{x}_t, \mathcal{R}) \\ P(C=2|\boldsymbol{x}_t, \mathcal{R}) \\ P(C=3|\boldsymbol{x}_t, \mathcal{R}) \end{bmatrix} = \begin{bmatrix} 0.40 \\ 0.25 \\ 0.35 \end{bmatrix}$$

Bayes decisions

The expected cost of actions "Predict a" are

$$\mathcal{C}_{\boldsymbol{x}_t,\mathcal{R}}(1) = 0 \times 0.40 + 1 \times 0.25 + 2 \times 0.35 = 0.95$$
$$\mathcal{C}_{\boldsymbol{x}_t,\mathcal{R}}(2) = 1 \times 0.40 + 0 \times 0.25 + 1 \times 0.35 = 0.75$$
$$\mathcal{C}_{\boldsymbol{x}_t,\mathcal{R}}(3) = 2 \times 0.40 + 1 \times 0.25 + 0 \times 0.35 = 1.05$$

or, in matrix form:

$$\begin{bmatrix} \mathcal{C}_{\boldsymbol{x}_t,\mathcal{R}}(1) \\ \mathcal{C}_{\boldsymbol{x}_t,\mathcal{R}}(2) \\ \mathcal{C}_{\boldsymbol{x}_t,\mathcal{R}}(3) \end{bmatrix} = \boldsymbol{C}\boldsymbol{q}_t$$

The optimal decision would therefore to assign label 2, even though it has the lowest posterior probability, since the expected cost due to mis-calssifications would be lower.

Bayes decisions

Let's now consider again a binary problem

We have four costs:

	Class $\mathcal{H}_F$	Class $\mathcal{H}_T$
Prediction $\mathcal{H}_F$	$\mathcal{C}(\mathcal{H}_F\|\mathcal{H}_F)$	$\mathcal{C}(\mathcal{H}_F\|\mathcal{H}_T)$
Prediction $\mathcal{H}_T$	$\mathcal{C}(\mathcal{H}_T\|\mathcal{H}_F)$	$\mathcal{C}(\mathcal{H}_T\|\mathcal{H}_T)$

Without loss of generality we assume

$$\mathcal{C}(\mathcal{H}_T|\mathcal{H}_T) = 0\ , \quad \mathcal{C}(\mathcal{H}_F|\mathcal{H}_F) = 0$$

i.e. correct decisions have no cost.

We also assume $\mathcal{C}(\mathcal{H}_F|\mathcal{H}_T) \geq 0$ and $\mathcal{C}(\mathcal{H}_T|\mathcal{H}_F) \geq 0$

Bayes decisions

The costs reflect the costs of the two different kind of errors:

	Class $\mathcal{H}_F$	Class $\mathcal{H}_T$
Prediction $\mathcal{H}_F$	0	$\mathcal{C}(\mathcal{H}_F \vert \mathcal{H}_T) = C_{fn}$
Prediction $\mathcal{H}_T$	$\mathcal{C}(\mathcal{H}_T \vert \mathcal{H}_F) = C_{fp}$	0

C_{fn} is the cost of false negative errors, C_{fp} is the cost of false positive errors

Bayes decisions

The expected Bayes cost for action $\mathcal{H}_T$ (i.e. for predicting $\mathcal{H}_T$) is

$$\mathcal{C}_{x,\mathcal{R}}(\mathcal{H}_T) = C_{fp}P(\mathcal{H}_F|x,\mathcal{R}) + 0 \cdot P(\mathcal{H}_T|x,\mathcal{R}) = C_{fp}P(\mathcal{H}_F|x,\mathcal{R})$$

whereas the cost for action $\mathcal{H}_F$ (i.e. for predicting $\mathcal{H}_F$) is

$$\mathcal{C}_{x,\mathcal{R}}(\mathcal{H}_F) = C_{fn}P(\mathcal{H}_T|x,\mathcal{R}) + 0 \cdot P(\mathcal{H}_F|x,\mathcal{R}) = C_{fn}P(\mathcal{H}_T|x,\mathcal{R})$$

The optimal decision is the labeling that has lowest cost

Bayes decisions

For binary problems, the optimal decision can be expressed as

$$a^*(x,\mathcal{R}) = \begin{cases} \mathcal{H}_T & \text{if } C_{fp}P(\mathcal{H}_F|x,\mathcal{R}) < C_{fn}P(\mathcal{H}_T|x,\mathcal{R}) \\ \mathcal{H}_F & \text{if } C_{fp}P(\mathcal{H}_F|x,\mathcal{R}) > C_{fn}P(\mathcal{H}_T|x,\mathcal{R}) \end{cases}$$

and we can choose any action when the two costs are equal.

Alternatively, we can express the optimal decision (up to tie-breaking) as

$$a^*(x,\mathcal{R}) = \begin{cases} \mathcal{H}_T & \text{if } r(x) > 0 \\ \mathcal{H}_F & \text{if } r(x) < 0 \end{cases}$$

where

$$r(x) = \log \frac{C_{fn}P(\mathcal{H}_T|x,\mathcal{R})}{C_{fp}P(\mathcal{H}_F|x,\mathcal{R})}$$

Bayes decisions

If $\mathcal{R}$ is a generative model for x, then we can express r in terms of costs, prior probabilities and likelihoods as

$$r(x) = \log \frac{\pi_T C_{fn}}{(1-\pi_T)C_{fp}} \cdot \frac{f_{X|\mathcal{H},\mathcal{R}}(x|\mathcal{H}_T)}{f_{X|\mathcal{H},\mathcal{R}}(x|\mathcal{H}_F)}$$

where $\pi_T = P(\mathcal{H} = \mathcal{H}_T)$ is the prior probability for class $\mathcal{H}_T$.

The decision rule thus becomes

$$r(x) \lessgtr 0 \iff \log \frac{f_{X|\mathcal{H},\mathcal{R}}(x|\mathcal{H}_T)}{f_{X|\mathcal{H},\mathcal{R}}(x|\mathcal{H}_F)} \lessgtr -\log \frac{\pi_T C_{fn}}{(1-\pi_T)C_{fp}}$$

Bayes decisions

The triplet (π_T, C_{fn}, C_{fp}) represents the working point of an application for a binary classification task.

We can show that the triplet is actually redundant, in the sense that we can build equivalent applications $(\pi'_T, C'_{fn}, C'_{fp})$ which have the same decision rule as the original application, but different costs and priors.

For example, the application $(\tilde{\pi}, 1, 1)$ with

$$\tilde{\pi} = \frac{\pi_T C_{fn}}{\pi_T C_{fn} + (1 - \pi_T) C_{fp}}$$

is equivalent to the application (C_{fn}, C_{fp}, π_T). Indeed, we have

$$\frac{\tilde{\pi}}{1 - \tilde{\pi}} = \frac{\frac{\pi_T C_{fn}}{\pi_T C_{fn} + (1 - \pi_T) C_{fp}}}{1 - \frac{\pi_T C_{fn}}{\pi_T C_{fn} + (1 - \pi_T) C_{fp}}} = \frac{\pi_T C_{fn}}{(1 - \pi_T) C_{fp}}$$

Bayes decisions

We can interpret $\tilde{\pi}$ as an effective prior: if the class prior for $\mathcal{H}_T$ was $\tilde{\pi}$ and we assumed uniform costs, we would obtain the same decisions as for our original application

Similarly, we can devise an equivalent application where the effective prior is uniform $\tilde{\pi} = \frac{1}{2}$, and the application prior π_T absorbed in "effective" classification costs (we won't prove it here)

Bayes decisions

We have, up to now, considered how to perform decisions for a sample x

We now return to the problem of evaluating the goodness of our decisions

We assume that recognizer $\mathcal{R}$ takes decisions $a(x, \mathcal{R})$ for sample x with correct class label c

The cost of each decision is thus

$$\mathcal{C}(a(x, \mathcal{R})|c)$$

If actions correspond to class labeling, it's the cost of predicting label $a(x, \mathcal{R})$ when the correct class is c

Bayes decisions

We can then compute the expected cost (Bayes risk) of decisions made by our classifier for the evaluation population

$$\mathcal{B} = \mathbb{E}_{X,C|\mathcal{E}}\left[\mathcal{C}(a(x,\mathcal{R})|c)\right]$$

$\mathcal{E}$ denotes the evaluation population, assumed to be distributed according to

$$X, C|\mathcal{E}$$

$\mathcal{E}$ represents the evaluation population, or, alternatively, an evaluator who has complete knowledge on the data

Note that, even with complete knowledge of the evaluation population, we may have $0 < P(C = k|X, \mathcal{E}) < 1$ for different classes (e.g., when samples with the same feature values may have different labels, as in the gender detection example)

Bayes decisions

We can express the Bayes risk as

$$\begin{aligned}\mathcal{B} &= \mathbb{E}_{X,C|\mathcal{E}}\left[\mathcal{C}(a(x,\mathcal{R})|c)\right] \\ &= \int f_{X|\mathcal{E}}(x)\sum_{c=1}^{K}\mathcal{C}(a(x,\mathcal{R})|c)P(C|X=x,\mathcal{E})dx \\ &= \int f_{X|\mathcal{E}}(x)\mathbb{E}_{C|X,\mathcal{E}}\left[\mathcal{C}(a(x,\mathcal{R})|c)\right]dx\end{aligned}$$

We can observe that, as long as $C|X,\mathcal{R} \sim C|X,\mathcal{E}$, then the risk is minimized by minimum-Bayes-cost decisions, since

$$\begin{aligned}\mathbb{E}_{C|X,\mathcal{E}}\left[\mathcal{C}(a(x,\mathcal{R})|c)\right] &= \mathbb{E}_{C|X,\mathcal{R}}\left[\mathcal{C}(a(x,\mathcal{R})|c)\right] \\ &= \mathcal{C}_{x,\mathcal{R}}(a(x,\mathcal{R})) \\ &\geq \mathcal{C}_{x,\mathcal{R}}(a^*(x,\mathcal{R}))\end{aligned}$$

The Bayes risk, in this case, represents the best possible cost we would pay for classifying test data according to $C|X,\mathcal{E} \sim C|X,\mathcal{R}$

Bayes decisions

In practice, however, the posterior distribution of $C|X, \mathcal{R}$ will not correspond to the distribution of $C|X, \mathcal{E}$

- $\mathcal{R}$ is only a model of the data
- Distribution mismatches may exist between training and evaluation data
- The model may not be producing good approximations of the data distributions
- ...

We usually do not know f_X either, and we are interested in evaluating the performance of a classifier that may have different posterior beliefs for the test samples

Ideally, these decisions will be Bayes optimal decisions for the classifier, according to the recognizer posterior distribution $C|X, \mathcal{R}$

Empirical Bayes Risk

We can nevertheless define the Bayes risk $\mathcal{B}$ for decisions made by $\mathcal{R}$ over evaluation data sampled from $X, C|\mathcal{E}$:

$$\mathcal{B} = \mathbb{E}_{X,C|\mathcal{E}}\left[\mathcal{C}(a(x,\mathcal{R})|c)\right] = \sum_{c=1}^{K} \pi_c \int \mathcal{C}(a(x,\mathcal{R})|c) f_{X|C,\mathcal{E}}(x|c)dx$$

The distribution $X|C,\mathcal{E}$ is the conditional distribution of the evaluation population

Again, $\mathcal{E}$ represents the evaluation population, or, alternatively, an evaluator who has complete knowledge on the data

Note again that the distribution reflects the knowledge of the evaluator $\mathcal{E}$, not the knowledge of the recognizer $\mathcal{R}$

The evaluator $\mathcal{E}$ is measuring how good are the decisions made by the recognizer $\mathcal{R}$ for his own task (data sampled from $X|C,\mathcal{E}$)

Empirical Bayes Risk

Unfortunately, we don't have access to $f_{X|C,\mathcal{E}}(x|c)$

However, if we have at our disposal a set of labeled evaluation samples $(x_1, c_1) \ldots (x_N, c_N)$, then we can approximate the expectations by averaging the cost over the samples

Indeed, if samples x_i are generated by $X|C, \mathcal{E}$, as the number of samples per class becomes large, it's possible to show that

$$\int \mathcal{C}(a(x, \mathcal{R})|c) f_{X|C,\mathcal{E}}(x|c) dx \approx \frac{1}{N_c} \sum_{i|c_i=c} \mathcal{C}(a(x_i, \mathcal{R})|c)$$

i.e. the integral can be approximated by the average cost computed over samples of each class

Empirical Bayes Risk

We can finally define the empirical Bayes risk as

$$\mathcal{B}_{emp} = \sum_{c=1}^{K} \frac{\pi_c}{N_c} \sum_{i|c_i=c} \mathcal{C}(a(x_i, \mathcal{R})|c)$$

The risk measures the costs of our decisions over the evaluation samples.

We can use $\mathcal{B}_{emp}$ to compare recognizers.

A recognizer that has lower cost will provide more accurate answers

Empirical Bayes Risk

The empirical Bayes risk can be computed from the confusion matrix and the matrix of costs

For example, let's consider again the 3-class problem, with cost matrix and priors given by

$$\boldsymbol{C} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix}, \quad \boldsymbol{\pi} = \begin{bmatrix} 0.3 \\ 0.4 \\ 0.3 \end{bmatrix}$$

For all test samples, we computed the Bayes decisions. We can then compute the confusion matrix, Let's assume that we obtain

$$\boldsymbol{M} = \begin{bmatrix} 205 & 111 & 56 \\ 145 & 199 & 121 \\ 50 & 92 & 225 \end{bmatrix}$$

Empirical Bayes Risk

We can compute, for each class, the term[1]

$$\frac{\pi_c}{N_c} \sum_{i|c_i=c} \mathcal{C}(a^*(x_i, \mathcal{R})|c)$$

For samples that belong to class 1, we have

$$\pi_1 = 0.3 \ , \quad N_1 = 205 + 145 + 50 = 400 \ .$$

For samples that are correctly classified (205) the cost is 0; for samples that are classified as class 2 (145) the cost is 1; for samples that are classified as class 3 (50) the cost is 2. Thus

$$\frac{\pi_1}{N_1} \sum_{i|c_i=1} \mathcal{C}(a^*(x_i, \mathcal{R})|c) = \frac{0.3}{400} (0 \times 205 + 1 \times 145 + 2 \times 50) = 0.18375$$

[1] We use $a^*(x_i, \mathcal{R})$ rather than $a(x_i, \mathcal{R})$ since these are optimal Bayes decisions from the classifier point of view

Empirical Bayes Risk

Similarly,

$$\frac{\pi_2}{N_2}\sum_{i|c_i=2}\mathcal{C}(a^*(x_i,\mathcal{R})|c) = \frac{0.4}{402}\left(1\times 111 + 0\times 199 + 1\times 92\right) \approx 0.20199$$

$$\frac{\pi_3}{N_3}\sum_{i|c_i=3}\mathcal{C}(a^*(x_i,\mathcal{R})|c) = \frac{0.3}{402}\left(2\times 56 + 1\times 121 + 0\times 225\right) \approx 0.17388$$

The empirical Bayes risk is

$$\mathcal{B}_{emp} \approx 0.18375 + 0.20199 + 0.17388 = 0.55962$$

Empirical Bayes Risk

Let's now consider again the binary problem

	Class $\mathcal{H}_F$	Class $\mathcal{H}_T$
Prediction $\mathcal{H}_F$	0	$C(\mathcal{H}_F\|\mathcal{H}_T) = C_{fn}$
Prediction $\mathcal{H}_T$	$C(\mathcal{H}_T\|\mathcal{H}_F) = C_{fp}$	0

We have seen that we can compute predicted labels by comparing the log-likelihood ratio to a threshold that depends on (π_T, C_{fn}, C_{fp})

Empirical Bayes Risk

Let c_i^* be the predicted label for sample x_i, whose label is c_i. The empirical Bayes risk is

$$\begin{aligned}
\mathcal{B}_{emp} &= \frac{\pi_T}{N_T} \sum_{i|c_i=\mathcal{H}_T} \mathcal{C}(c_i^*|\mathcal{H}_T) + \frac{1-\pi_T}{N_F} \sum_{i|c_i=\mathcal{H}_F} \mathcal{C}(c_i^*|\mathcal{H}_F) \\
&= \pi_T \frac{\sum_{i|c_i=\mathcal{H}_T} C_{fn} \mathbb{I}[c_i^* = \mathcal{H}_F]}{N_T} + (1-\pi_T) \frac{\sum_{i|c_i=\mathcal{H}_F} C_{fp} \mathbb{I}[c_i^* = \mathcal{H}_T]}{N_F} \\
&= \pi_T \frac{\sum_{i|c_i=\mathcal{H}_T, c_i^*=\mathcal{H}_F} C_{fn}}{N_T} + (1-\pi_T) \frac{\sum_{i|c_i=\mathcal{H}_F, c_i^*=\mathcal{H}_T} C_{fp}}{N_F} \\
&= \pi_T C_{fn} \cdot FNR + (1-\pi_T) C_{fp} \cdot FPR \\
&= \pi_T C_{fn} P_{fn} + (1-\pi_T) C_{fp} P_{fp}
\end{aligned}$$

where $P_{fn} = FNR$ is the false negative rate (false rejection rate) and $P_{fp} = FPR$ is the false positive rate (false acceptance rate)

$\mathcal{B}_{emp}$ is also called (un-normalized) Detection Cost Function (DCF)

Model evaluation for classification

Detection Cost Functions:

- Define the costs of different kind of errors (C_{fn}, C_{fp})
- Define the class prior probability (π_T, $\pi_F = 1 - \pi_T$)
- Evaluate by computing empirical Bayes risk

$$DCF_u(C_{fn}, C_{fp}, \pi_T) = \pi_T C_{fn} P_{fn} + (1 - \pi_T) C_{fp} P_{fp}$$

- P_{fn} and P_{fp} are the false negative and false positive rates, and depend on the selected threshold t

Model evaluation for classification

C_{fn}, C_{fp} and π_T depend only on the application

A dummy system that always accepts a test segment ($c_t = \mathcal{H}_T$):

$$P_{fp} = 1, P_{fn} = 0 \implies DCF_u = (1 - \pi_T)C_{fp}$$

A dummy system that always rejects a test segment ($c_t = \mathcal{H}_F$):

$$P_{fp} = 0, P_{fn} = 1 \implies DCF_u = \pi_T C_{fn}$$

Normalized DCF: we compare the system DCF w.r.t. the best dummy system

$$DCF(\pi_T, C_{fn}, C_{fp}) = \frac{DCF_u(\pi_T, C_{fn}, C_{fp})}{\min(\pi_T C_{fn}, (1 - \pi_T)C_{fp})}$$

Note that the best dummy system corresponds to optimal Bayes decisions based on prior information alone (i.e. $llr(x) = 0$ for all samples)

Model evaluation for classification

Normalized DCF is invariant to scaling

We can thus re-scale the un-normalized DCF by $\frac{1}{\pi_T C_{fn}+(1-\pi_T)C_{fp}}$

Let $\tilde{\pi} = \frac{\pi_T C_{fn}}{\pi_T C_{fn}+(1-\pi_T)C_{fp}}$, so that $1-\tilde{\pi} = \frac{(1-\pi_T)C_{fp}}{\pi_T C_{fn}+(1-\pi_T)C_{fp}}$

The un-normalized DCF becomes

$$DCF_u(\tilde{\pi}) = \tilde{\pi}P_{fn} + (1-\tilde{\pi})P_{fp}$$

whereas the corresponding normalized DCF has the same value

In terms of normalized DCF, the applications (π_T, C_{fp}, C_{fn}) and $(\tilde{\pi}, 1, 1)$ are again equivalent

Model evaluation for classification

We can observe that the error rate we defined at the beginning as

$$e = \frac{\#\ of\ incorrectly\ classified\ samples}{\#\ of\ samples} = 1 - accuracy$$

corresponds to

$$e = \frac{N_T P_{fn} + N_F P_{fp}}{N} = \frac{N_T}{N} P_{fn} + \frac{N_F}{N} P_{fp}$$

i.e., up to a scaling factor, to the DCF of an application $(\frac{N_T}{N}, 1, 1)$, where $\frac{N_T}{N}$ is the empirical prior of the evaluation set (not necessarily the same as the application prior)

The weighted error rate

$$e = \frac{1}{2}(P_{fn} + P_{fp})$$

corresponds to the application $(\frac{1}{2}, 1, 1)$

Model evaluation for classification

For systems producing well-calibrated log–likelihood ratios

$$s = \log \frac{f_{X|C}(x|\mathcal{H}_T)}{f_{X|C}(x|\mathcal{H}_F)}$$

the optimal threshold (optimal Bayes decision) is given by

$$t = -\log \frac{\tilde{\pi}}{1-\tilde{\pi}}$$

Indeed, since $\tilde{\pi}$ is the effective prior probability of $\mathcal{H}_T$, the posterior log–likelihood ratio is

$$\log \frac{P(\mathcal{H}_T|x)}{P(\mathcal{H}_F|x)} = \log \frac{f_{X|C}(x|\mathcal{H}_T)}{f_{X|C}(x|\mathcal{H}_F)} + \log \frac{\tilde{\pi}}{1-\tilde{\pi}}$$

Model evaluation for classification

LLRs allow disentangling the classifier from the application

In general, systems often do not produce well–calibrated LLRs

- Non-probabilistic scores (e.g. SVM)
- Mis-match between train and test populations
- Non-accurate model assumptions

In these cases, we say that scores are mis-calibrated

The theoretical threshold $-\log \frac{\tilde{\pi}}{1-\tilde{\pi}}$ is not optimal anymore

Model evaluation for classification

For a given application, we can measure the additional cost due to the use of mis-calibrated scores

We can define the minimum cost DCF_{min} corresponding to the use of the optimal threshold for the evaluation set

We consider varying the threshold t to obtain all possible combinations of P_{fn} and P_{fp} for the evaluation set

We select the threshold corresponding to the lowest DCF

Model evaluation for classification

The corresponding value DCF_{min} is the cost we would pay if we knew before-hand the optimal threshold for the evaluation

We can think of this value as a measure of the quality of the classifier

We can also compute the actual DCF obtained using the threshold corresponding to the effective prior $\tilde{\pi}$

The difference between the actual and minimum DCF represents the loss due to score mis-calibration

Model evaluation for classification

We can also compare different systems over different applications through Bayes error plots

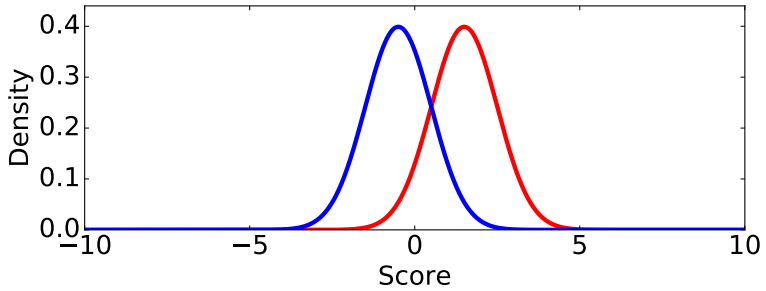
These plots can be used to report actual and / or minimum DCF for different applications

A binary application is parametrized by a single value $\tilde{\pi}$

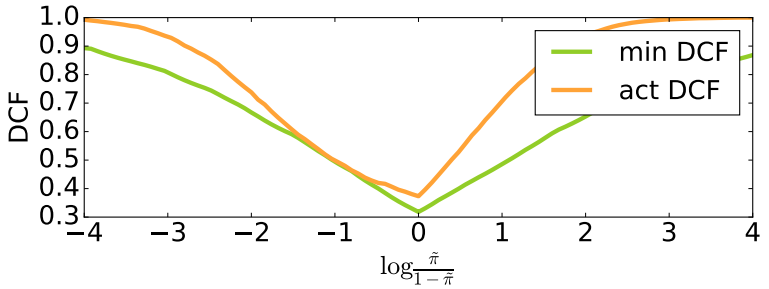
We can thus plot the DCF as a function of prior log-odds $\log \frac{\tilde{\pi}}{1-\tilde{\pi}}$, i.e. the negative of the Bayes optimal threshold.

Model evaluation for classification

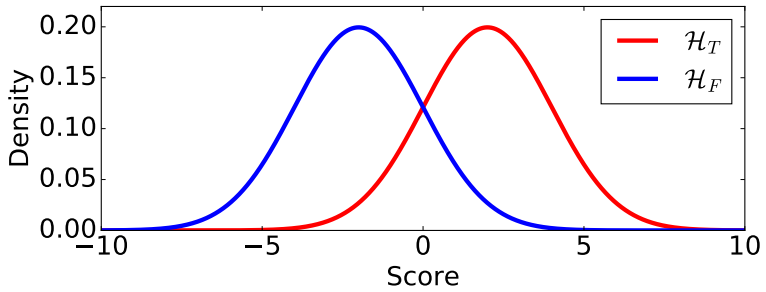
Non calibrated scores



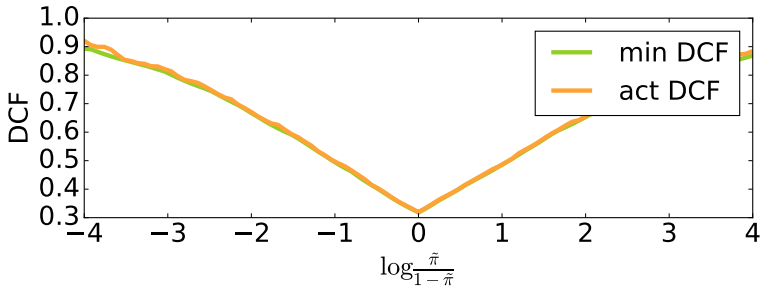
Bayes error plot



Calibrated scores



Bayes error plot



Score calibration

To reduce mis-calibration we can adopt different calibration strategies

We can use a validation set to find a (close-to) optimal threshold for a given application

More general approaches look for functions that transform the classifier scores s into approximately well-calibrated LLRs, in a way that is as much as possible independent from the target application

Score calibration approaches:

- Isotonic regression
- Prior-weighted logistic regression
- Generative score models (e.g. Gaussian score models)

We want to compute a transformation function f that maps the classifiers scores s to well-calibrated scores $s_{cal} = f(s)$

Score calibration

Isotonic regression

- Non-linear, monotonic transformation that provides optimal calibration for the data it's trained on
- Piecewise non-linear, may require some sort of interpolation for unseen scores
- Does not allow extrapolating outside of training scores range
- Expensive to evaluate when the calibration training set is large

Score calibration

Score models

- Approximation to the isotonic regression transformation
- Require assumptions on the calibration transformation (e.g. linear mapping) or on the distribution of class scores
- Models estimated over a training set, also allow for extrapolation outside of training score ranges
- Typically, fast to evaluate
- May provide good fit only for a small range of operating points

Example: Prior-weighted logistic regression

Score calibration

Calibration: prior-weighted logistic regression

We consider the non-calibrated scores as features

We assume a linear mapping from non-calibrated scores to calibrated scores

$$f(s) = \alpha s + \gamma$$

Since $f(s)$ should produce well-calibrated scores, $f(s)$ can be interpreted as the log-likelihood ratio for the two class hypotheses

$$f(s) = \log \frac{f_{S|C}(s|\mathcal{H}_T)}{f_{S|C}(s|\mathcal{H}_F)} = \alpha s + \gamma$$

The class posterior probabilities for prior $\tilde{\pi}$ correspond to

$$\log \frac{P(C = \mathcal{H}_T|s)}{P(C = \mathcal{H}_F|s)} = \alpha s + \gamma + \log \frac{\tilde{\pi}}{1 - \tilde{\pi}} = \alpha s + \beta$$

Score calibration

We can employ the prior-weighted Logistic Regression model with $\pi_T = \tilde{\pi}$ to learn the model parameters α, β over our training scores (we will refer to the set of scores as calibration set in the following)

To recover the calibrated score $f(s)$ we will need to compute

$$f(s) = \alpha s + \gamma = \alpha s + \beta - \log \frac{\tilde{\pi}}{1 - \tilde{\pi}}$$

Note that we have to specify a prior $\tilde{\pi}$, and we are still effectively optimizing the calibration for a specific application $\tilde{\pi}$. However, as we will see, the model will often provide good calibration for a wider range of different applications

Alternatively, we can modify the prior-weighted logistic regression objective to compute directly α and γ:

$$R(\alpha, \gamma) = \sum_i w_i \log \left(1 + e^{-z_i\left(\alpha s + \gamma + \log \frac{\tilde{\pi}}{1-\tilde{\pi}}\right)}\right), \quad w_i = \begin{cases} \tilde{\pi}/n_T & \text{if } z_i = +1 \\ (1-\tilde{\pi})/n_F & \text{if } z_i = -1 \end{cases}$$

Score calibration

In all cases, we need a calibration set to estimate the transformation

Typically, two scenarios:

- Mis-calibration due to non-probabilistic scores, or to overfitting or underfitting models: the calibration set can be taken from the training set material
- Mis-calibration due to mismatched between training and evaluation population: the calibration set should mimic the evaluation population
  - Requires collecting data similar to the evaluation population
  - The amount of required matching data, however, is usually small compared to the amount required to train a complete classifier
  - Some models can work with unlabeled samples (unsupervised training)

Score calibration

For the projects we are in the first use-case

The calibration set must be taken from the validation set (the evaluation set cannot be used to estimate any part of the model)

With a single-split approach we require 3 splits:

- Model training
- Calibration set (samples that are scored with the trained classifier and are used to compute calibration parameters)
- Actual validation set (samples that are scored with the trained model and whose scores are calibrated with the calibration model, used to evaluate the performance of the complete system)

Score calibration

With K-fold a possible approach consists in a 2-step procedure

First step: apply K-fold to train the classifier:

- Train K classifiers, each without fold k (model $\mathcal{R}_k$)
- Score each fold k with model $\mathcal{R}_k$
- Train a classifier $\mathcal{R}_{\mathcal{F}}$ over the whole training set (we will need this later)
- Pool the scores of each fold to obtain a score set, that we will use as calibration set

Score calibration

Second step: apply K-fold over the calibration scores, using a different randomized shuffle (alternatively, you may apply a single-split protocol for the validation data, but the split should be the same for all models to be compared)

- Train K calibration models $\mathcal{C}_k$
- Train a calibration model over all pooled scores $\mathcal{C}_{\mathcal{F}}$ (we will need this later)
- Calibrate the scores of each fold k with model $\mathcal{C}_k$
- Pool the calibrated scores and evaluate the model performance over the pooled scores (to choose the best configuration)

Chose the classification system according to evaluation data

Classify the evaluation scores: apply classification model $\mathcal{R}_{\mathcal{F}}$ for the chosen system followed by the corresponding calibration model $\mathcal{C}_{\mathcal{F}}$

Model evaluation for classification

Evaluation of multiclass tasks is more complex

As we have seen, we can build confusion matrices

We can also compute the empirical Bayes risk for multiclass problems

We can compute a normalized detection cost, obtained by scaling the Bayes risk by the cost of the best dummy system — in this case, we have K dummy systems, each of them predicting a different class k regardless of the sample

For the multiclass problem, it's more difficult to separate the costs due to mis-calibration from those due to poor discriminant capabilities of the classifier

Also in this case, calibration models allow reducing mis-calibration costs (e.g. multiclass logistic regression)