Bayes decisions and Model evaluation

Sandro Cumani

sandro.cumani@polito.it

Politecnico di Torino

Bayes decisions and model evaluation

Up to now we have assigned labels to samples based on the maximum-a-posteriori class probabilities

This approach, however, does not consider the fact that different labeling errors may have a different impact

For example, if we are building a diagnostic system, we may incorrectly label a healthy patient as potentially ill or vice-versa.

In the former case, the error may lead to additional, potentially expensive exams, however, in the second case, the error may lead to severe or permanent injuries of the patient

Maximum-a-posterior class assignment would not necessarily lead to the best decision in a similar context

Model evaluation for classification

To improve on our class assignment approach we start by revisiting how we measure the performance of our system

We have already seen two simple metrics, *accuracy* and *error rate*, defined as

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

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

We could therefore try to devise a decision making approach that would optimize the accuracy metric under suitable assumptions

Model evaluation for classification

However, although in some scenarios accuracy can be a good metric, it's affected by several issues that may result in the metric being less useful for the evaluation of a classifier, or the comparison of different classifiers:

- it does not account for the cost of the different kinds of errors
- it's dependent on the empirical class prior of the different classes, which does not necessarily reflect the application prior
- it's *unnormalized*, i.e., it does not, by itself, allow judging whether a classifier is indeed "useful" (e.g. "better than chance", or, more precisely, better than decisions taken *without* the classifier)

Model evaluation for classification

Ideally we would like our system to provide optimal performance on the application data. However, we cannot measure a priori the performance of a system on the application data, since typically we don't have neither the data nor the corresponding labels

We can, on the other hand, employ an evaluation set to compute the performance of different models on the evaluation set itself

The evaluation metrics thus:

- measure *exactly* our performance on the given dataset
- can be used to obtain an *estimate* of our performance on application data that behaves similarly to our evaluation set

For the latter point, we shall see that it's important that our metrics do not depend on the evaluation set empirical prior, when this may differ from the application prior

Model evaluation for classification

To understand the shortcomings of accuracy, and to devise a more suitable metric, we start defining the *confusion matrix*, which provides the complete summary of the mis-classifications of a classifier *on a given dataset*.

The matrix collects, for each combination of class label C_i and predicted label C_j, the number of samples of C_i predicted as C_j:

	Class C_1	Class C_2	...	Class C_K
Prediction C_1	# samples of C_1 predicted as C_1	# samples of C_2 predicted as C_1		# samples of C_K predicted as C_1
Prediction C_2	# samples of C_1 predicted as C_2	# samples of C_2 predicted as C_2		# samples of C_K predicted as C_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Prediction C_K	# samples of C_1 predicted as C_K	# samples of C_2 predicted as C_K		# samples of C_K predicted as C_K

The elements on the diagonal are correctly classified, while the elements outside of the diagonal are incorrectly classified

Model evaluation for classification

For a binary problem:

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

True Negatives (TN) and True Positives (TP) are the *correctly* classified samples

False Negatives and False Positives (FP) are the *incorrectly* classified samples

Accuracy and error rate correspond to

$$acc = \frac{TN + TP}{TN + FP + TP + FN}, \quad err = \frac{FP + FN}{TN + FP + TP + FN}$$

Model evaluation for classification

To better understand the limitations of accuracy, 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

To better understand the limitations of accuracy, 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 now consider a diagnostic task, where we need to classify a medical exam as "positive" or "negative"

To assess the effectiveness of a classifier, we employ a dataset that contains both actual positive and negative samples (sample label)

To have a good coverage of both cases, we collect the results for few hundreds of patients of each kind. Let's assume we have 1000 positive and 1000 negative samples, and that two classifiers $\mathcal{R}_1$ and $\mathcal{R}_2$ have the following confusion matrices for the evaluation set:

	Pos.	Neg.
Pred. Pos.	940	20
Pred. Neg.	60	980

	Pos.	Neg.
Pred. Pos.	980	40
Pred. Neg.	20	960

The second recognizer has an overall larger accuracy, 97% vs. 96%

Model evaluation for classification

However, the accuracy and error rate measure the number of errors for the dataset, ignoring the fact that our real application may have a different positive patient prior than the evaluation set empirical prior (i.e., the ratio of positive patients in the field may differ from the ratio of positive patients in the evaluation set)

To better understand the contribution to the error rate of the different kinds of error and of the empirical prior we start defining the per-class error rates (and corresponding correct classification rates)

Model evaluation for classification

Per-class error rates (and derived measures):

- False negative rate FNR (false rejection / miss rate): $P_{fn} = \frac{FN}{FN+TP}$ (error rate for the positive class)
- False positive rate FPR (false acceptance): $P_{fp} = \frac{FP}{FP+TN}$ (error rate for the negative class)
- 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 notice that these metrics all depend on the sample of a single class, i.e., they are unaffected by the proportion of samples of each class

Model evaluation for classification

The evaluation set *emprical prior* corresponds to the fraction of positive samples

$$\pi_{\mathcal{E}}^{emp} = \frac{TP + FN}{TN + FP + TP + FN}$$

The error rate can be represented as

$$\begin{aligned} err &= \frac{\text{\# of errors}}{\text{\# of samples}} = \frac{FP + FN}{TN + FP + TP + FN} \\ &= \frac{FP}{TN + FP + TP + FN} + \frac{FN}{TN + FP + TP + FN} \\ &= \frac{FP}{TN + FP} \cdot \frac{TN + FP}{TN + FP + TP + FN} + \frac{FN}{TP + FN} \cdot \frac{TP + FN}{TN + FP + TP + FN} \\ &= P_{fp}(1 - \pi_{\mathcal{E}}^{emp}) + P_{fn}\pi_{\mathcal{E}}^{emp} \end{aligned}$$

i.e., it's a weighted sum of the error rates *of each class*, where the weights are the *empirical class priors of the dataset*

Model evaluation for classification

Let's now imagine that the classifier is employed in a scenario where the prior probability of a positive result is smaller, e.g. 1%.

If our evaluation set was to reflect the application prior, it would need to contain 99 negative samples for each positive one.

Assuming we want at least 1000 samples for each class, we would need to collect 99 000 samples for the negative class.

On the other hand, if we assume that the class error rates would be the approximately the same as those we measured in our original, balanced ($\pi_{emp} = 0.5$) evaluation set, we could estimate the error rate that we would have for the application-prior-balanced ($\pi = 0.01$) dataset, without actually collecting the large amount of negative samples:

$$err_{app} = P_{fp}(1 - \pi) + P_{fn}\pi$$

Model evaluation for classification

For the two considered systems, we would have

$$P_{fp}(\mathcal{R}_1) = \frac{20}{20+980} = 0.02\ , \quad P_{fp}(\mathcal{R}_2) = \frac{40}{40+960} = 0.04$$
$$P_{fn}(\mathcal{R}_1) = \frac{60}{60+940} = 0.06\ , \quad P_{fn}(\mathcal{R}_2) = \frac{20}{20+980} = 0.02$$

The empirical prior is $\pi_{\mathcal{E}}^{emp} = \frac{1000}{2000} = 0.5$, and corresponds to the original error rates

$$err(\mathcal{R}_1) = 0.04\ , \quad err(\mathcal{R}_2) = 0.03$$

For an application with prior $\pi = 0.01$ and the same per-class error rates, however, we would have

$$err_{app}(\mathcal{R}_1) = 0.0204\ , \quad err_{app}(\mathcal{R}_2) = 0.0398$$

i.e., in this case the expected number of errors of $\mathcal{R}_2$ would be larger than that of $\mathcal{R}_1$

Model evaluation for classification

Finally, the error rate does not account for the *cost* of the different errors

For the diagnostic task, we have seen that, if the prior probability of being positive is low, than the first recognizer results in a lower total number of errors. However, it will incorrectly label as negative more positive patients than the second system.

We can imagine that incorrectly labeling as negative a positive patient may be more dangerous than the opposite, i.e., it may have a larger *cost*[1]

[1] The concept of cost is not restricted to a monetary cost, but is a generalization that should quantify the effects of an incorrect classification, and in particular reflect the *relative* effects of different error types. Typically, costs will change from application to application.

Model evaluation for classification

We shall, in the following, introduce a metric that overcomes all these issues, allowing us to summarize the performance of a classifier with a single number that:

- depends on the application, rather than on the empirical, prior
- accounts for the different costs of classification errors
- allows comparing different classifiers
- is normalized, i.e., allows assessing whether a classifier is indeed extracting useful information at all from a sample

Bayes decisions

Let's go back to the core definition of a classification problem

Given a task, the goal of a classifier is to allow us to choose a suitable action (or decision) a to perform among a set of possible 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

Although we have not yet defined how to obtain optimal decisions, we assume that a classifier is able to produce a decision (labeling in our case) for each sample according to some rule

We denote with $a(x, \mathcal{R})$ the decision made by $\mathcal{R}$ for sample x, whose correct class label c

The cost of such decision is

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

If decisions 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 express the expected cost (*Bayes risk*) of decisions made by our classifier for the target (application) population, i.e., the cost we expect to pay for using our system on the application population:

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

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

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

$\mathcal{E}$ can also be interpreted as an *evaluator* who has complete knowledge of the application data

Bayes decisions

We can express the Bayes risk as

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

Unfortunately, we do not have knowledge of the distribution $X, C|\mathcal{E}$.

If we assume we know the application prior probabilities

$$\pi_c = P(C = c|\mathcal{E})$$

we can express the Bayes risk *for the target application* as

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

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

Empirical Bayes Risk

Since we don't have access to $f_{X|C,\mathcal{E}}(x|c)$, we cannot compute the Bayes risk

However, if we have at our disposal a set of labeled evaluation samples $(x_1, c_1) \dots (x_N, c_N)$, 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, we have that

$$\begin{aligned}\mathbb{E}_{X|C,\mathcal{E}}\left[\mathcal{C}(a(x,\mathcal{R})|c)|c\right] &= \int \mathcal{C}(a(x,\mathcal{R})|c) f_{X|C,\mathcal{E}}(x|c)dx \\ &\approx \frac{1}{N_c}\sum_{i|c_i=c} \mathcal{C}(a(x_i,\mathcal{R})|c)\end{aligned}$$

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

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 for a *target application* over the *evaluation samples*

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

Lower costs correspond to better performance

If we set $\pi_c = \pi_c^{emp}$ then the empirical Bayes risk corresponds to the total (mis-)classification costs for the evaluation samples (similar to error rates, but accounting for costs)

Empirical Bayes Risk

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

For example, let's consider a 3-class problem, with cost matrix and priors given by[2]

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

For all test samples, we computed the recognizer decisions, which correspond, in this example, to the confusion matrix:

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

[2]Without loss of generality, we can assume that correct assignments have zero cost

Empirical Bayes Risk

We can compute, for each class, the term

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

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

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

Empirical Bayes Risk

Similarly,

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

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

The empirical Bayes risk is

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

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

Empirical Bayes Risk

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

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

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

Model evaluation for classification

Detection Cost Function / empirical Bayes risk:

- 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 (defined in the following slides) based on *prior information alone*, i.e., using the prior probability in place of the recognizer posterior probability for any given sample

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 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 normalized costs 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)

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 $(\frac{1}{2}, 1, 1)$

Final remarks

Evaluation of multiclass tasks is more complex, and we cannot represent any application with a single parameter (effective prior)

However, we can compute the empirical Bayes risk

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

Model evaluation and Bayes decisions

Before we introduce Bayes optimal decision, let's again consider a two-class problem

We have seen that, in many cases, binary classifiers output a single score that "measures" the strength of the $\mathcal{H}_T$ class hypothesis:

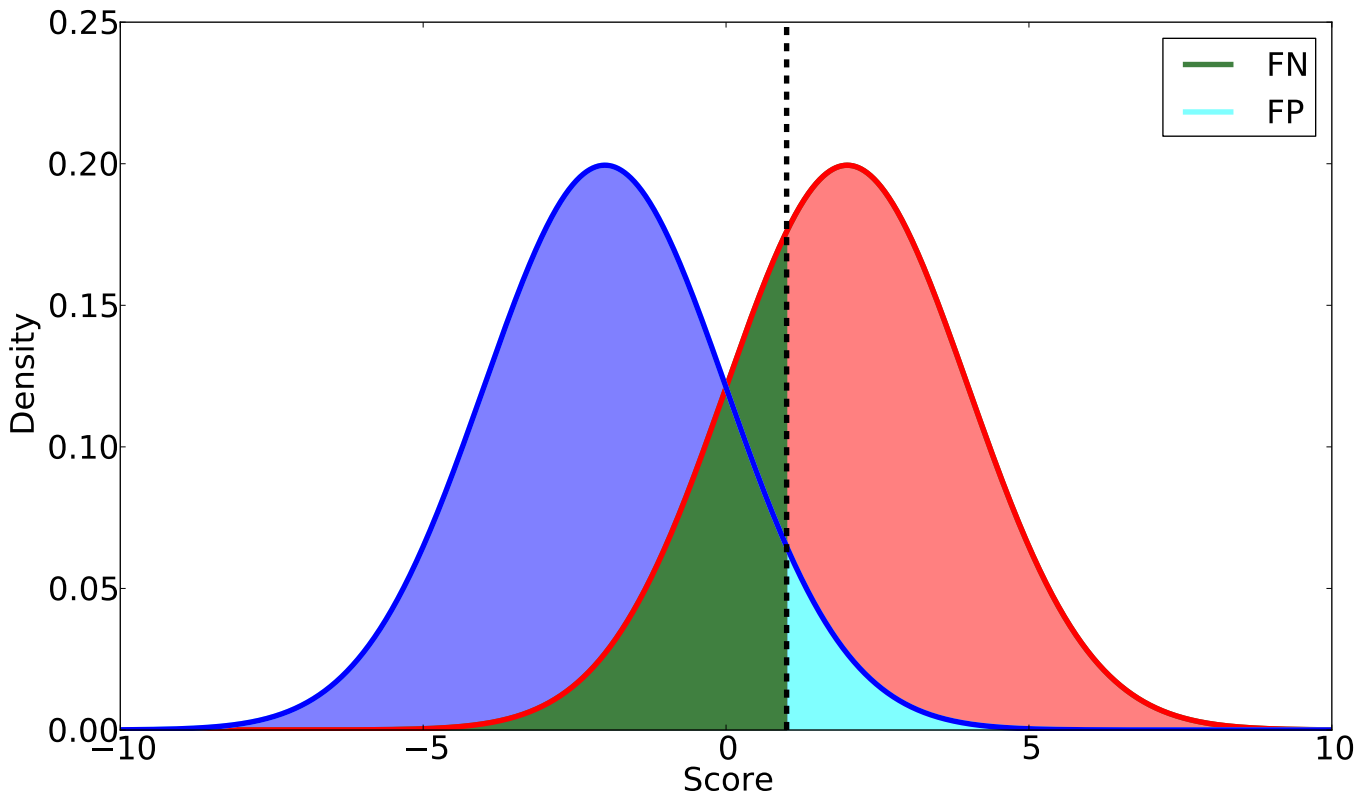
- Generative models: log–likelihood ratios $s = \log \frac{f(x|\mathcal{H}_T)}{f(x|\mathcal{H}_F)}$
- Discriminative models: posterior log-probability ratios $s = \log \frac{P(\mathcal{H}_T|x)}{P(\mathcal{H}_F|x)}$
- Non–probabilistic models: score (e.g. SVM) $s = \boldsymbol{w}^T \boldsymbol{x}$

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

Model evaluation and Bayes decisions

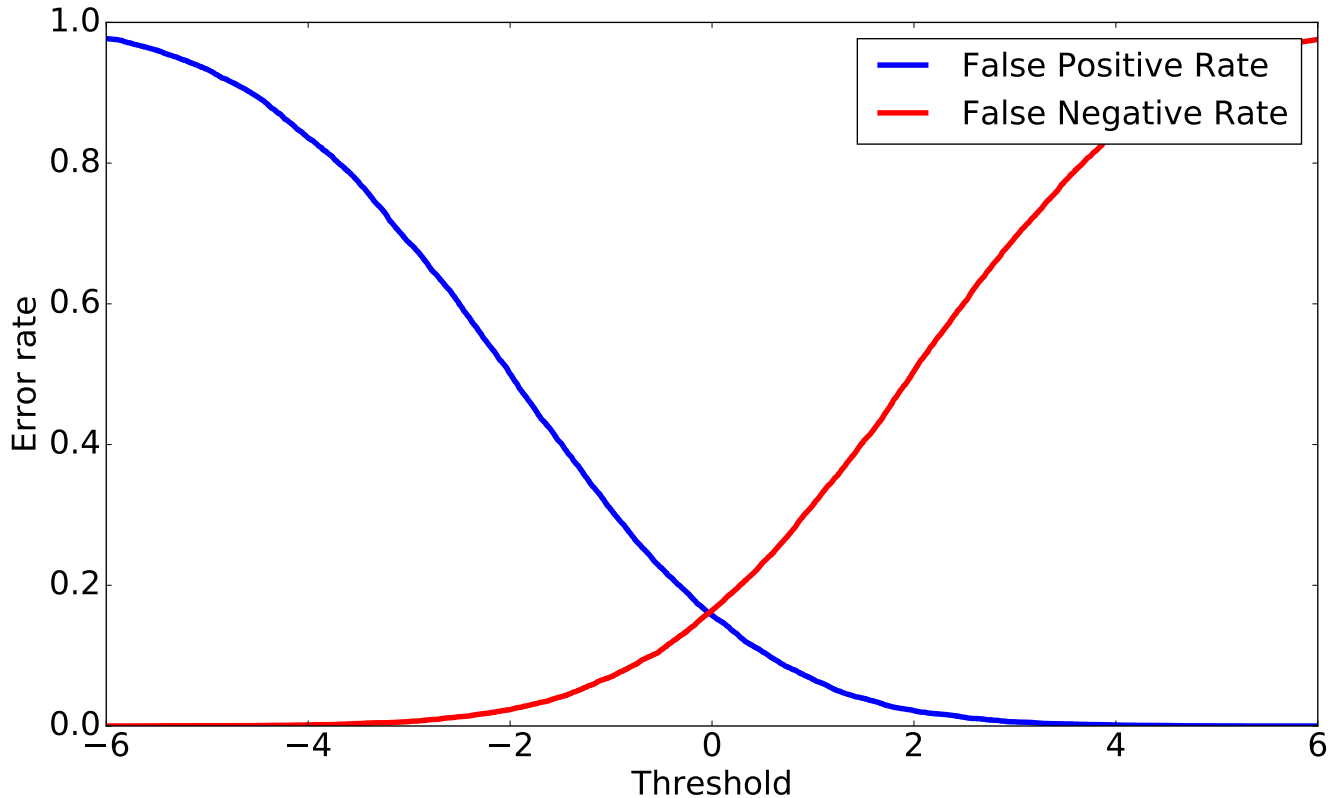
We have seen that decisions can be cast as comparing the score with a threshold

We can observe that different thresholds correspond to different error rates for the two classes:



Model evaluation and Bayes decisions

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

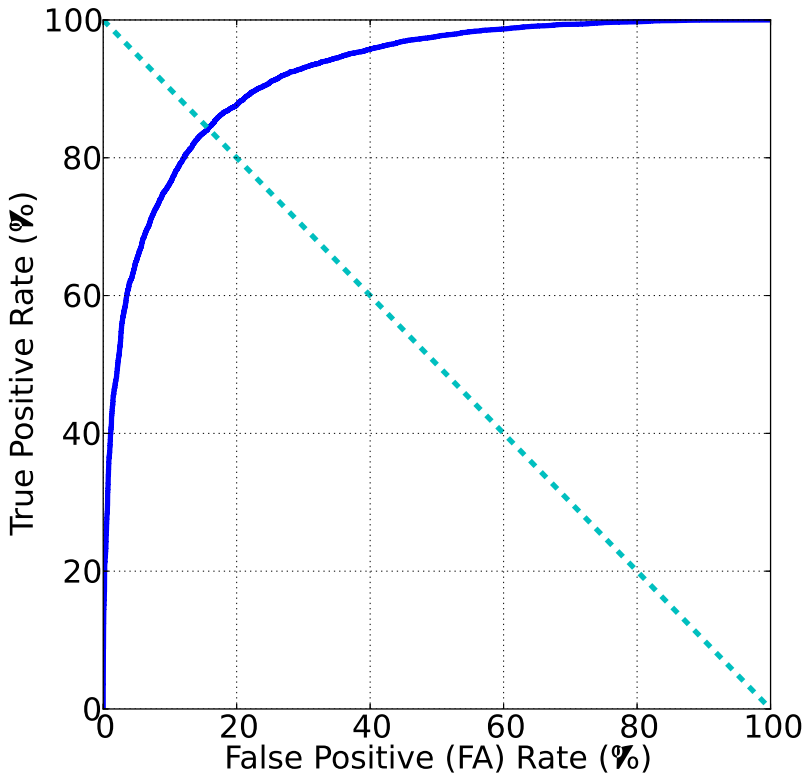


An additional metric we can define is the Equal Error Rate (EER), corresponding to the threshold for which $FPR = FNR$ (it's related to the empirical Bayes risk, but we won't analyze the details here)

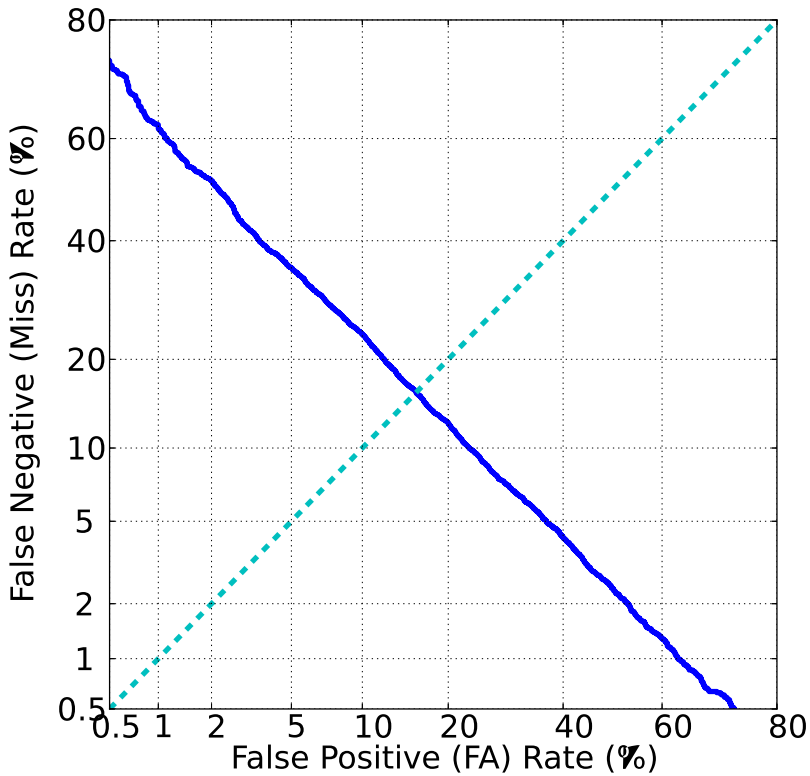
Model evaluation and Bayes decision

We can also visualize directly the trade-offs of the different kinds of error as we change the threshold

- Receiver Operating Characteristic (ROC) curve



- Detection Error Trade-off (DET) curve



Bayes decisions

Since the error rates depend on the selected threshold, we would like to optimize the threshold selection for a given application. More in general, we would like to devise a good way to transform the classifier scores (binary or multiclass) in effective decisions

For the moment, we consider that our classifier is probabilistic, i.e., it is able to provide class posterior probabilities $P(C = k|x, \mathcal{R})$ for a given application[3] for sample x

We can thus compute the expected cost of action a according to the posterior probabilities $P(C = k|x, \mathcal{R})$

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

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

[3]Note the *subjective* nature of probabilities: the posterior probabilities reflect the knowledge of $\mathcal{R}$, a recognizer $\mathcal{R}'$ may assign different probabilities to the classes

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-classifications would be lower.

Bayes decisions

Optimal Bayes decisions are optimal *from the point of view of the recognizer*

If the evaluator and the recognizer agree (i.e., they report the same class posterior probabilities for any sample), the Bayes decisions minimize the Bayes risk

Indeed, we can express the Bayes risk as

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

Bayes decisions

If $C|X,\mathcal{R} \sim C|X,\mathcal{E}$, then the risk becomes[4]

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

since, for any x, $a^*(x,\mathcal{R})$ is the action that minimizes the expected cost

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

[4]Note that this result does not depend on $f_{X|\mathcal{E}}$

Bayes decisions

We can interpret optimality in two ways:

- optimal Bayes decisions of a recognizer $\mathcal{R}$ minimize the Bayes risk, *as evaluated by the recognizer itself*
- optimal Bayes decisions of a system that has complete data knowledge $\mathcal{R} = \mathcal{E}$ minimize the Bayes risk of the evaluator $\mathcal{E}$

In the latter case the Bayes risk represents the best possible cost we would pay for classifying test data (but, of course, since we don't have full data knowledge we cannot compute the risk in this case)

Note that, even with complete knowledge, 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), thus even in this case the Bayes risk may be non-zero

Bayes decisions

Let's now consider again a binary problem with cost matrix

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

where 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, as for the Bayes empirical risk, 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, we can represent a binary application in terms of the effective prior

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

as application $(\tilde{\pi}, 1, 1)$. The application $(\tilde{\pi}, 1, 1)$ is indeed equivalent to the application (C_{fn}, C_{fp}, π_T), since

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

Bayes decisions and model evaluation

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) becomes, in terms of effective prior:

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

Bayes decisions and model evaluation

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

Bayes decisions and model evaluation

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 a given 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

Bayes decisions and model evaluation

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

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

Bayes decisions and model evaluation

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

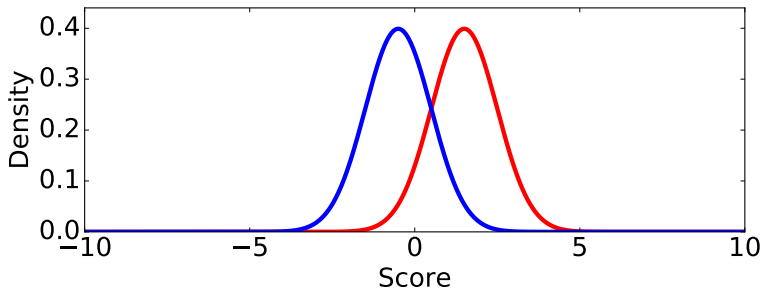
These plots can be used to report actual and / or minimum DCF for different applications

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

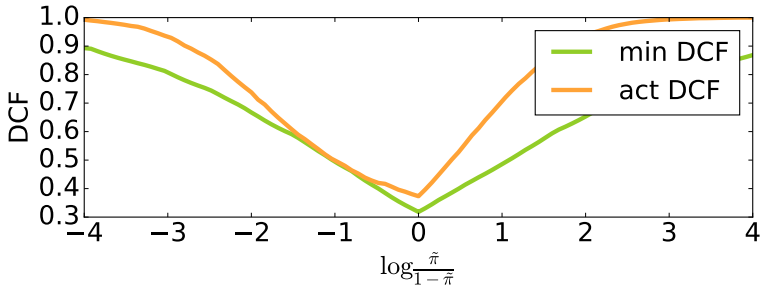
We can thus plot the DCF as a function of prior log-odds $\log \frac{\tilde{\pi}}{1-\tilde{\pi}}$, i.e. the negative of the Bayes optimal threshold.

Bayes decisions and model evaluation

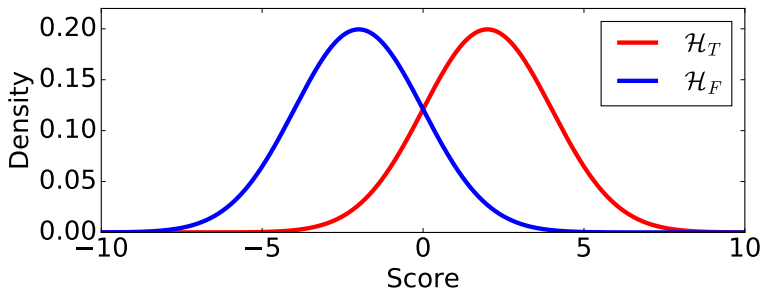
Non calibrated scores



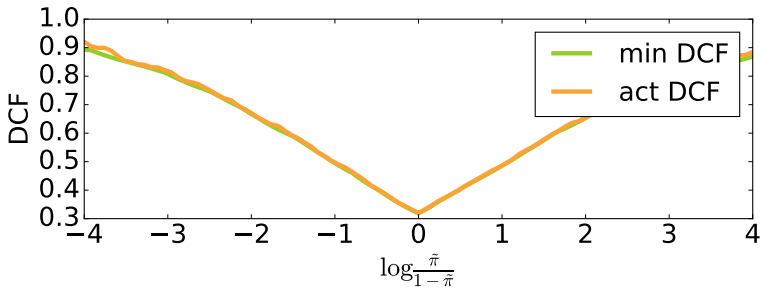
Bayes error plot



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 (a variant of Platt scaling)
- 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