The Expected Performance Curve: a New Assessment Measure for Person Authentication

Samy Bengio *Johnny Mariéthoz*

IDIAP
CP 592, rue du Simplon4
1920 Martigny, Switzerland
{bengio,marietho}@idiap.ch

Abstract

ROC and DET curves are often used in the field of person authentication to assess the quality of a model or even to compare several models. We argue in this paper that this measure can be misleading as it compares performance measures that cannot be reached simultaneously by all systems. We propose instead new curves, called Expected Performance Curves (EPC). These curves enable the comparison between several systems according to a criterion, decided by the application, which is used to set thresholds according to a separate validation set. A free sofware is available to compute these curves. A real case study is used throughout the paper to illustrate it. Finally, note that while this study was done on an authentication problem, it also applies to most 2-class classification tasks.

1. Introduction

The general field of person authentication comprises several well-established research domains such as verification of voice, face, signature, fingerprints, etc [1]. In all these cases, researchers tend to use the same performance measures to estimate and compare their models. Two broad classes of performance measures appear in the literature: *a priori* measures, where the performance is computed on a set of data which was never seen by the model, reflecting realistic expected performances, and *a posteriori* measures, where the test data was used to set some parameters (such as thresholds), reflecting optimistically biased expected performances. An other very popular method to present the performance is through the use of curves showing the performance on the test set for various thresholds. The most well known of these curves is the famous *Receiver Operating Characteristic* (ROC).

The main purpose of this paper is to argue that such curves can not be used to either compare two or more models, nor obtain a realistic estimate of the performance of a given model.

In Section 2, we review the various performance measures used in the field of person authentication. Then in Section 3, we explain, using a real case study, why some of these measures can be misleading. In Section 4, we propose instead a family of curves that really reflects the expected performance of a given model, hence enabling a fair comparison between models. Finally, in Section 5, we show how these curves can be applied to related domains which can also be casted into the framework of 2-class classification problems. Finally, Section 6 concludes the paper.

2. Current Performance Measures in Verification Tasks

A verification system has to deal with two kinds of events: either the person claiming a given identity is the one who he claims to be (in which case, he is called a *client*), or he is not (in which case, he is called an *impostor*). Moreover, the system may generally take two decisions: either *accept* the *client* or *reject* him and decide he is an *impostor*.

Thus, the system may make two types of errors: a *false acceptance* (FA), when the system accepts an *impostor*, and a *false rejection* (FR), when the system rejects a *client*.

In order to be independent on the specific dataset distribution, the performance of the system is often measured in terms of these two different errors, as follows:

$$\text{FAR} = \frac{\text{number of FAs}}{\text{number of impostor accesses}}\ , \tag{1}$$

$$\text{FRR} = \frac{\text{number of FRs}}{\text{number of client accesses}}\ . \tag{2}$$

A unique measure often used combines these two ratios into the so-called *detection cost function* (DCF) [2] as follows:

$$\text{DCF} = \left\{ \begin{array}{l} \text{Cost(FR)} \cdot P(\text{client}) \cdot \text{FRR}+ \\ \text{Cost(FA)} \cdot P(\text{impostor}) \cdot \text{FAR} \end{array} \right. \tag{3}$$

where $P(\text{client})$ is the prior probability that a client will use the system, $P(\text{impostor})$ is the prior probability that an impostor will use the system, Cost(FR) is the cost of a false rejection, and Cost(FA) is the cost of a false acceptance.

A particular case of the DCF is known as the *half total error rate* (HTER) where the costs are equal to 1 and the probabilities are 0.5 each:

$$\mathrm{HTER} = \frac{\mathrm{FAR} + \mathrm{FRR}}{2} . \tag{4}$$

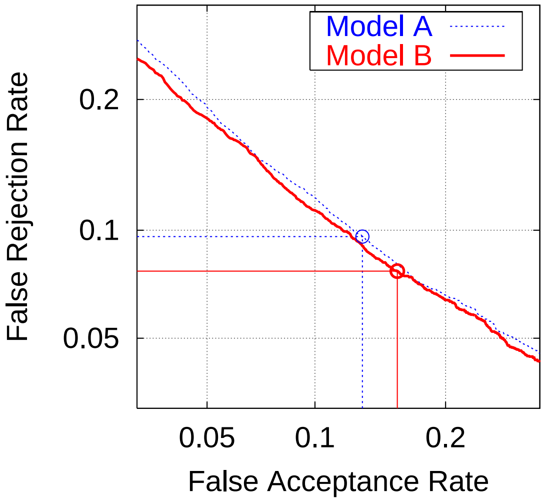

Figure 1: DET curves for models A and B: the lower the better.

Note however that in most cases, the system can be tuned using a *decision threshold* in order to obtain a compromise between either a small FAR or a small FRR. There is thus a trade-off which depends on the application: it might sometimes be more important to have a system with a very small FAR, while in other situations it might be more important to have a system with a small FRR. In order to see the performance of a system with respect to this trade-off, we usually plot the so-called *Receiver Operating Characteristic* (ROC) curve, which represents the FRR as a function of the FAR [3] (hence, the curve which is nearer the $(0, 0)$ coordinate is the best ROC curve). Other researchers have also proposed the DET curve [4], which is a non-linear transformation of the ROC curve in order to make results easier to compare. The non-linearity is in fact a normal deviate, coming from the hypothesis that the scores of client accesses and impostor accesses follow a Gaussian distribution. If this hypothesis is true, the DET curve should be a line. Figure 1 shows an example of two typical DET curves.

On either the ROC or the DET curve, each point of the curve corresponds to a particular decision threshold that should be determined specifically for a given application. A typical threshold chosen to compare models is the one that minimizes the HTER (4) or its generalized version, the DCF (3). Another typical threshold chosen is the one that reaches the *Equal Error Rate* (EER) where FAR=FRR on a separate validation set.

Note however that many researchers publish results with a threshold chosen to reach the EER on the test set, which is not realistic as the test set is not supposed to be used to estimate any parameter of a model. In fact, these results will be systematically optimistically biased, so they should be regarded with caution.

Other researchers simply publish the whole ROC or DET curve on the test set, letting the user select his own threshold. The object of this paper is to show that this is not a good practice either.

To make things clear, we will call a result *a priori* when it has been computed using a threshold chosen on a separate validation set, and *a posteriori* when the threshold was chosen on the test set. Hence, given two DET curves, only *a posteriori* performances can be compared.

3. The Problem with ROC and DET Curves

As explained in Section 2, ROC and DET curves show the performance of the system on the test set for different thresholds (also called operating points). However, in a real-life application, one would normally have to select the threshold before looking at the test set. This is why measures such as DCF (3) or HTER (4) computed using a threshold chosen on a separate dataset are more realistic. However, these measures reflect only one possible operating point, which might be misleading in some cases.

Criterion	Method	FAR	FRR	HTER
HTER min (validation)	Model A	0.114	0.108	**0.111**
	Model B	0.139	0.086	0.112
EER (validation)	Model A	0.131	0.096	**0.114**
	Model B	0.158	0.078	0.118
EER (test)	Model A	0.110	0.110	0.110
	Model B	0.107	0.107	**0.107**

Table 1: Performance comparison between models A and B using three different criteria: minimum HTER and EER on a separate validation set, and EER on the test set.

We would like here to present a real case study[1] comparing 2 speaker verification models (hereafter called *model A* and *model B*) on the NIST'2000 benchmark database, where the respective DET curves and HTER performances yield incompatible results, showing that one of the measures (or both) does not fully represent the expected performance of the system. Figures 1 and 2 compare the *a posteriori* DET/ROC curves of the two models, while Table 1 compares the performances of the

[1] While this is not important for this paper, people interested in knowing more about this case study are referred to [5].

two models on three different operating points: one that minimizes the HTER on a separate validation set, one such that FAR = FRR on the validation set, and one such that FAR = FRR on the test set itself.

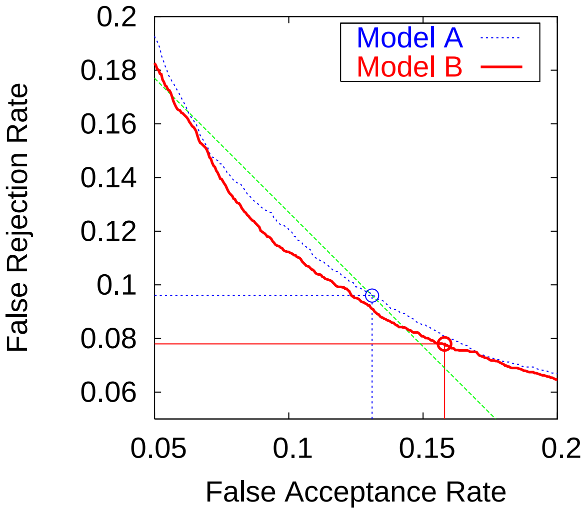

Figure 2: ROC curves for models A and B: the lower the better. The straight line represents a constant HTER line passing through the selected solution for model A. We can see that it also passes *under* the selected solution for model B.

Looking at the DET and ROC curves of Figures 1 and 2, we see that model B's performance is always below (better than) model A's performance, letting think that for any threshold, model B should always be better. However, looking at Table 1, we see that for the two operating points computed *a priori* (on a separate validation set), model A is indeed better that model B, while on the operating point computed *a posteriori* (on the test set), model B is better than model A. Moreover, results obtained with either the *a priori* EER criterion or the *a posteriori* EER criterion are both statistically significant[2] with a confidence level of 95%, although showing opposite behaviors!

In order to explain why the DET and ROC curves misled us, consider the two circles on Figure 1. They represent the performance of the model when the threshold was selected using the same criterion (EER) on a separate validation set. The selected thresholds are quite different from each other and from the test data EERs, thus the circles are far from each other. The naive approach would have compared two points coming from the same line crossing the origin. Indeed, it might happen, and it is the case here for many points, that the HTER of a given point of model A becomes less than the HTER of another point of model B. Another way to see the problem is looking at Figure 2. The additional straight line represents a constant HTER: all points along this line have an HTER similar to the solution obtained by model A. We can see that this lines passes *under* the solution proposed by model B, hence is in fact better!

4. The Expected Performance Curve

In a real application setting, one has in general a criterion to optimize which reflects the relative costs of each type of error (FAs and FRs). Hence we would like to propose a method that presents the expected performance of a model with respect to this criterion. In this Section, we propose three such curves, each reflecting a particular way to express this criterion. We shall call these curves Expected Performance Curves (EPC).

4.1. EPC Curves for Three Criteria

As a general framework for EPC curves, we would like to show performance obtained on a test set (for instance the HTER) with respect to performance expected when the threshold is set on a separate validation set. This threshold could be chosen in several ways. Note that all the curves that are presented in this Section have been computed using the freely available EPC software[3].

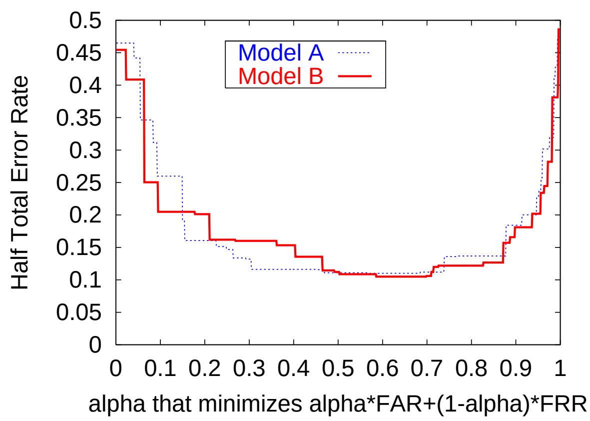

Figure 3: DCF Expected Performance Curves for models A and B.

The first solution is to select the threshold in order to minimize the DCF criterion (3) on a separate validation set. Algorithm 1 presents the general method to obtain such a curve, where α aggregates both the relative costs and prior distributions of clients and impostors. For our case study presented in Section 3, the result can be seen in Figure 3, where *alpha* represents α. For instance, we

[2] with a standard proportion test on the corresponding classification error, assuming a binomial distribution for the decisions, and using a normal approximation since there was 63573 test accesses.

[3] EPC is available at `http://www.Torch.ch/extras/epc` as a package of the Torch machine learning library.

Algorithm 1 Method to generate the DCF Expected Performance Curve.

Let $valid$ be the validation set
Let $test$ be the test set
Let FAR$(\theta, valid)$ be the FAR obtained on the validation set for threshold θ
for values $\alpha \in [0, 1]$ **do**
  $\theta^\star = \arg\min_\theta \begin{pmatrix} \alpha \cdot \text{FAR}(\theta, valid) + \\ (1-\alpha) \cdot \text{FRR}(\theta, valid) \end{pmatrix}$
  compute FAR$(\theta^\star, test)$, FRR$(\theta^\star, test)$ and HTER$(\theta^\star, test)$
  plot HTER$(\theta^\star, test)$ with respect to α
end for

can see that if one selects the threshold such that it minimizes the HTER on a separate validation set (which corresponds to the performances obtained when *alpha* $= 0.5$ on this Figure), the obtained test HTER of model A is slightly better than the one of model B (as confirmed in Table 1), while if the threshold is chosen to minimize, say, (0.8 FAR + 0.2 FRR) on a separate validation set, then model B is better than model A. More generally, this Figure shows that neither of the two models is better for a wide range of *alpha* values.

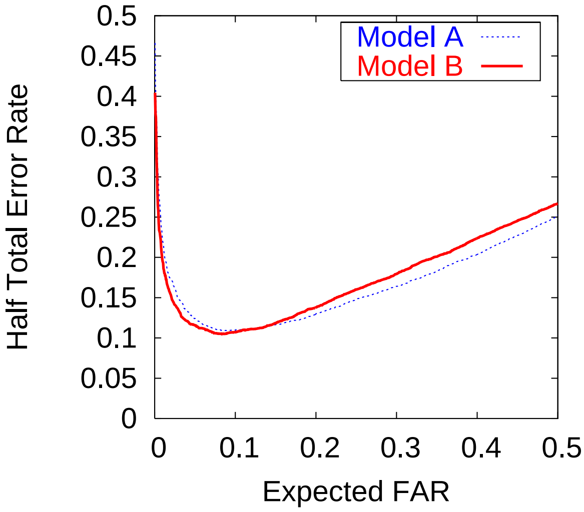

Figure 4: FAR Expected Performance Curves for models A and B.

On the other hand, if the criterion is to control the expected FAR (this is often the case for some banking applications), then we should look at Figure 4, which compares the model for several values of the expected FAR (using again a separate validation set to select the corresponding thresholds). Using this graph, it is clear that model B is always better than model A for small values of expected FAR. Figure 5 shows the same graph when the criterion is to control the expected FRR instead of

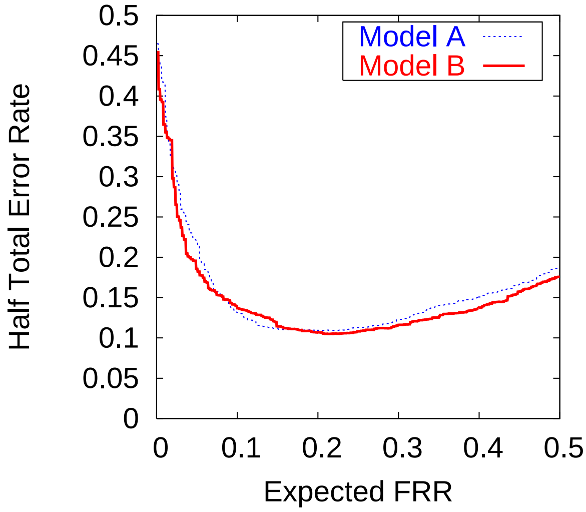

Figure 5: FRR Expected Performance Curves for models A and B.

FAR. Here, depending on the expected FRR, there is no clear winner between models A and B. In order to generate Figures 4 and 5, algorithm 1 needs only to modify the evaluation of $\theta^\star$ as follows: For FAR EPC curves,

$$\theta^\star = \arg\min_\theta |\alpha - \text{FAR}(\theta, valid)| \tag{5}$$

while for FRR EPC curves,

$$\theta^\star = \arg\min_\theta |\alpha - \text{FAR}(\theta, valid)| \tag{6}$$

and α now represents a target value of the expected FAR/FRR respectively.

4.2. More Analysis of EPC Curves

In order to understand a little bit more the behavior of each model, we can also compare the expected FAR (computed on a separate validation set) with the obtained FAR (on the test set). Figure 6 shows this curve for models A and B. We see that model A is nearer the correct answer (which is represented by the line $y = x$), while model B always underestimate the actual FAR. The same graph comparing expected and obtained FRR can be seen in Figure 7. Here, clearly, both models have largely overestimated the FRR. In fact, this bad estimation has a significant impact on the choice of the threshold, which then impact on the obtained results, hence explaining why the original DET cannot be used to compare models: the DET does not take into account the error made during the threshold estimation.

4.3. Discussion on the Validation Set

All the EPC curves rely on the availability of a separate validation set which is used to compute the various

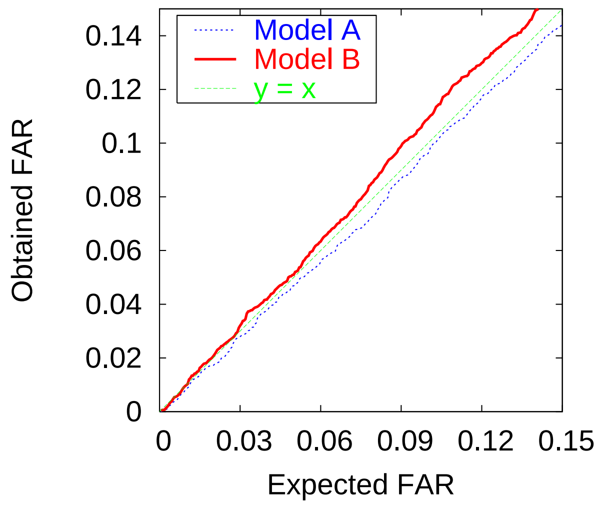

Figure 6: Obtained FAR with respect to expected FAR for models A and B.

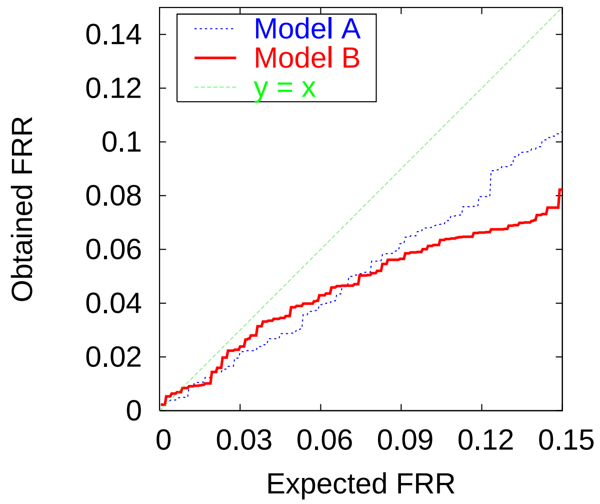

Figure 7: Obtained FRR with respect to expected FRR for models A and B.

thresholds that are then applied on the test set. Unfortunately, such validation set is often not readily available. We discuss in this section several options to encompass this problem.

One alternative to the separate validation set is to use the data that was available to tune the other parameters of the model. This dataset is often called the *training set* or the *development set*. The cleanest solution is to separate this training set into two parts, use the first one as usual to tune the various models and use the second part as the validation set (hence, to compute the various thresholds). Note that this separation must be done carefully, making sure that the accesses are divided client-wise (hence all information from a given client should either be in the new training set or in the validation set, but not in both).

When the training set is too small to be divided into two parts, one can also rely on a cross-validation strategy on the training set, such as the leave-one-client-out cross-validation technique.

Another option could be to directly use the training set as a validation set. Given that we are only tuning one parameter (the threshold), this should not really affect the overall performance, since we are not using the test set.

One could also be tempted to perform some kind of cross-validation on the test set itself. While this looks like a reasonable solution, it is unfortunately not. The reason is that doing so, we miss one important reason for bad threshold estimation: the mismatch between training and test data. If such a mismatch exists, one would get much better *apparent* performance with this technique than any other since it would not be affected by the mismatch (because the threshold would be set by using some part of the test set).

Note that all the results presented in this paper used a real separate validation set. We also performed some experiments using a cross-validation technique directly on the test set, which indeed ended up in obtaining optimistically biased results (all expected performances were almost the same as real obtained performances), as expected from the discussion of the previous paragraph.

5. Application to Other Tasks

It is interesting to note that several other application domains use ROC curves (or derivatives of them) to present their results and hence could benefit from this study.

For instance, in the field of *information retrieval*, the practical problem of *text categorization* can be stated as follows [6]: categorize a new document in one or many of a given set of categories. In this domain, results are often presented in terms of a ROC curve where the axes are slightly different from those used in authentication. The axes are defined as *precision* and *recall*, where *precision* is the number of true acceptances (TA) divided by the sum of TA and FA, while *recall* is the number of TA divided by the sum of TA and FR. Moreover, results in this research community are most often reported as a combination of these two terms, such as the *break-even point* (BEP), which, similarly to the EER, is the point such that *precision* equals *recall*, which can only be computed *a posteriori* on a given test set. Another way results are presented is through the so-called *eleven-point average precision*, which estimates the area under the ROC curve through the average of 11 estimated values of the curve. Thus in both cases results use *a posteriori* information, and are hence expected to be unreliable for the same reason explained in Section 3.

Yet another type of ROC curves, often found in the medical research domain (see for instance [7]), shows the *sensitivity* with respect to 1 minus the *specificity*, where *sensitivity* is defined as the TA ratio (TAR) and the *specificity* is defined as the true rejection ratio (TRR). Hence, we argue that here again, comparing two models according to this type of curve can be misleading.

6. Conclusion

In this paper, we have proposed the use of the Expected Performance Curve (EPC) to assess the quality of a person authentication system. The current measures either show the unbiased performance on only one point (such as the HTER or the DCF) or show a biased performance on a wide range of settings (such as the DET or the ROC). The proposed EPC enables to show, for a wide range of settings, the unbiased expected performance of a given system. More precisely, one can decide a given criterion (a small expected FAR, a parameterized DCF, etc) according to some real-life application, and compute the expected performance of several systems under these conditions, which enable a more realistic comparison between models as well as a better analysis of their respective expected performance. Note that a free software is available to compute these curves (`http://www.torch.ch/extras/epc`).

7. Acknowledgments

The authors would like to thank Mikaela Keller for fruitful discussions.This research has been carried out in the framework of the Swiss NCCR project (IM)2.

8. References

[1] P. Verlinde, G. Chollet, and M. Acheroy, "Multimodal identity verification using expert fusion," *Information Fusion*, vol. 1, pp. 17–33, 2000.

[2] A. Martin and M. Przybocki, "The NIST 1999 speaker recognition evaluation - an overview," *Digital Signal Processing*, vol. 10, pp. 1–18, 2000.

[3] H. L. Van Trees, *Detection, Estimation and Modulation Theory, vol. 1*, Wiley, New York, 1968.

[4] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," in *Proceedings of Eurospeech'97, Rhodes, Greece*, 1997, pp. 1895–1898.

[5] J. Mariéthoz and S. Bengio, "An alternative to silence removal for text-independent speaker verification," Technical Report IDIAP-RR 03-51, IDIAP, Martigny, Switzerland, 2003.

[6] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[7] Zweig and Campbell, "ROC plots: a fundamental evaluation tool in clinical medicine," *Clinical Chemistry*, vol. 39, no. 4, pp. 561–577, 1993.