The BOSARIS Toolkit: Theory, Algorithms and Code for Surviving the New DCF

Niko Brümmer and Edward de Villiers
AGNITIO Research, South Africa

December 2011

Abstract

The change of two orders of magnitude in the *new DCF* of SRE'10, relative to the old DCF evaluation criterion, posed a difficult challenge for participants and evaluator alike. Initially, participants were at a loss as to how to calibrate their systems, while the evaluator underestimated the required number of evaluation trials. After the fact, it is now obvious that both calibration and evaluation require very large sets of trials. This poses the challenges of (i) how to decide what number of trials is enough, and (ii) how to process such large data sets with reasonable memory and CPU requirements.

After SRE'10, at the BOSARIS Workshop, we built solutions to these problems into the freely available BOSARIS Toolkit. This paper explains the principles and algorithms behind this toolkit. The main contributions of the toolkit are:

1. The *Normalized Bayes Error-Rate Plot*, which analyses likelihood-ratio calibration over a wide range of DCF operating points. These plots also help in judging the adequacy of the sizes of calibration and evaluation databases.
2. Efficient algorithms to compute DCF and minDCF for large score files, over the range of operating points required by these plots.
3. A new score file format, which facilitates working with very large trial lists.
4. A faster logistic regression optimizer for fusion and calibration.
5. A principled way to define equal error rate, which is of practical interest when the absolute error count is small.

1 Introduction

The BOSARIS Toolkit provides MATLAB code for calibrating, fusing and evaluating scores from (automatic) binary classifiers. It was developed to provide solutions for automatic speaker recognition, but we envision that much of the code will have wider applicability for other biometric and/or forensics problems, where the calibration of likelihood-ratios is of interest. This document serves as a *user guide*, to explain theory and algorithms and is complementary to the user manual.

The theory behind the toolkit is based on the Ph.D. dissertation [1], which can be consulted for further details. The core implementation (code) was written by the authors of this document, as part of the ABC: AGNITIO, BUT, CRIM submission for the 2010 NIST Speaker Recognition Evaluation (SRE'10) [2]. After the evaluation, at the BOSARIS Workshop,[1] we collaborated with a wider group of researchers to make these algorithms available in toolkit form.[2]

This document is organized in three sections: **Theory** is the bulk of the document, which explains what the toolkit does and why. **Algorithms** explains how the toolkit does it. **Code** gives a high-level summary of the implementation.

2 Theory

This section provides the theoretical framework which is necessary for a good understanding of the BOSARIS Toolkit. For the typical speaker recognition expert, part of this material should be very familiar, while other parts may be new. All readers should nevertheless review the familiar parts, where the terminology for discussing the new material will be established. This section is organized as follows:

- Subsection 2.1 discusses the problem of running out of errors and the way this is addressed in the toolkit.
- Subsection 2.2 reviews Bayes decision theory, while 2.3 reviews NIST's DCF criterion for evaluating goodness of decisions.
- Subsection 2.4 introduces the idea that we can evaluate system outputs in the form of *likelihood-ratios*, rather than decisions. The key is to let the evaluator make the decisions, at the theoretically optimal Bayes threshold. Subsection 2.5 develops this idea into practical evaluation criteria.
- Subsection 2.6 discusses perhaps unfamiliar relationships between the familiar evaluation tools, ROC/DET, EER and minDCF.
- Subsection 2.7 discusses solutions for fusing and calibrating scores.

2.1 Sampling effects

All of the evaluation methods used in this toolkit explicitly or implicitly depend on estimating various error-rates by counting occurrences of those errors in a supervised evaluation database. The error-rates depend not only on the accuracy of the system under evaluation, but also on the *operating point.* We explain operating points in more detail later. What is important here is that no matter what the accuracy of the system under evaluation, or no matter what the size of the evaluation database, there will be operating points where the error-rates become so small that no more errors are observed. More generally, there will be operating points where the numbers of observed errors become so small that the error-rate estimates become unreliable.

There are various frequentist (confidence interval) or Bayesian (credible interval) methods to theoretically quantify the accuracy of such estimates—see

[1] See `http://speech.fit.vutbr.cz/workshops/bosaris2010`

[2] Available at: `http://sites.google.com/site/bosaristoolkit/`

for example [3] and references therein. The results of any such analysis will depend on various modelling assumptions.

For the speaker recognition problem, one such analysis, *Doddington's Rule of 30* [4], is rendered tractable via the assumption of independent Bernoulli trials.[3] This rule suggests one needs at least 30 errors to get a probably approximately correct error-rate estimate. In practice, we have found this rule to work well. We get sensible results in both training and test, if we ensure that there are at least 30 misses and at least 30 false-alarms at the operating point of interest.

2.1.1 Toolkit solution

In the BOSARIS Toolkit, we address the problem by flagging on our plots (DET curves as well as normalized Bayes error-rate curves) the points at which the various error-rates drop below 30. It is up to the user of the toolkit to understand that regions on the plot beyond these flags must be treated with caution.

2.1.2 In SRE'10

In SRE'10, at the 'new DCF' operating point of interest, there was a scarcity of false-alarms, which we addressed by manufacturing many more non-target trials. (This was possible because the number of possible non-target trials grows quadratically with the number of speakers in the available data.)

We used the rule of thumb that:

- If we want to use a database for calibration/fusion, that database has to be sufficiently large so that the calibrated/fused system makes at least 30 *training* errors of both types, at all operating points of interest.

- If we want to use an independent database for testing/evaluation, the same holds. That database has to be sufficiently large so that the system makes at least 30 *test* errors of both types, at all operating points of interest.

2.2 Bayes decision theory

The toolkit is focused on the canonical *speaker detection* problem, where independent decisions must be made for independent trials, based on the output *scores* of an automatic speaker recognition system. In most of this section, we consider the case of making decisions by using the output scores of a single system. We defer fusion of multiple systems to subsection 2.7.

The input to the toolkit is in the form of *scores* calculated by the automatic system. We assume that for every trial, the system has calculated a scalar detection score and that a decision has to be made based on this score. The recipe for doing so is given by Bayes decision theory [5].

In the canonical detection problem, there are two alternative hypotheses, called *target* and *non-target*, exactly one of which must be true for every trial. By convention, larger (more positive) scores favour the target hypothesis and smaller (more negative) scores favour the non-target hypothesis.

For every trial, an *accept/reject* decision is required. We define the *outcome* of a trial as the pair (hypothesis, decision), so that there are four possible

[3] Are different scores of the same speaker independent? Are miss and false-alarm rates independent?

outcomes. Two of these are considered to be *errors*: miss $=$ (target, reject) and false-alarm $=$ (non-target, accept). The other two outcomes are the correct outcomes.

The *consequence* of an outcome is expressed as a *cost function*, which maps outcomes to positive real numbers. Without loss of generality (see [1, section 3.4] and [5]), we restrict attention to cost functions which assign zero cost to correct outcomes. This leaves two costs to be specified: C_{miss}, the cost of a miss; and C_{fa}, the cost of a false-alarm.

When given a score, say s, the Bayes decision chooses the option, accept or reject, that *minimizes the risk.* That is, we choose to accept if

$$P(\text{target}|s,\pi)C_{\text{miss}} \geq P(\text{non-target}|s,\pi)C_{\text{fa}} \tag{1}$$

and to reject otherwise. The two risks being compared are products of costs and posterior probabilities. The posteriors are conditioned not only on s, but also on some independent *prior information*, which we represent as:

$$\pi = P(\text{target}) = 1 - P(\text{non-target}) \tag{2}$$

We refer to π as the *target prior*, or simply as the *prior.* By using Bayes' rule and taking logs, we can rewrite the decision rule as follows:

$$\text{accept, if } \ell(s) \geq \eta, \text{or reject otherwise.} \tag{3}$$

where we have defined the *log-likelihood-ratio*:

$$\ell(s) = \log \frac{P(s|\text{target})}{P(s|\text{non-target})} \tag{4}$$

the *Bayes decision threshold*:

$$\eta = \log \frac{C_{\text{fa}}}{C_{\text{miss}}} - \operatorname{logit} \pi \tag{5}$$

and the *prior log odds*:[4]

$$\operatorname{logit} \pi = \log \frac{\pi}{1-\pi} \tag{6}$$

We refer to the function $\ell : \mathbb{R} \mapsto \mathbb{R}$ as the *calibration mapping.* It maps the score, s, to the log-likelihood-ratio, $\ell(s)$. Since log-likelihood-ratios follow the same convention as the scores (larger values favour the target hypothesis), they are also scores. We shall therefore also refer to them as *calibrated scores.* On the other hand, scores are generally not calibrated and cannot do the work of log-likelihood-ratios: when scores are thresholded at the Bayes decision threshold, they usually do not make good decisions.

The toolkit is concerned with: (i) evaluating the potential ability of the scores, s, to make Bayes decisions, even if the calibration mapping, ℓ, is not available; (ii) creating such mappings, by training on a supervised calibration database; and (iii) evaluating the ability of the calibrated log-likelihood-ratios $\ell(s)$ to make Bayes decisions.

[4] The invertible function $\operatorname{logit}(p) = \log \frac{p}{1-p}$ maps probabilities in $[0,1]$ to log odds in $[-\infty,\infty]$.

2.3 DCF: criterion for goodness of hard decisions

The Bayes decision paradigm leads naturally to a recipe for evaluating the goodness of detection decisions made on a database of supervised trials. In the Speaker Recognition Evaluations (SREs) of 1997 to the present (2010), NIST has required systems under evaluation to submit a hard accept/reject decision, as well as a score, for each trial. The primary evaluation criterion, called DCF (detection cost function), evaluated the goodness of the hard decisions, while secondary criteria (minDCF and DET-curves) evaluated the goodness of the scores.[5]

In what follows, we shall always assume that hard decisions, if made by the evaluee, are made by thresholding all scores against a single fixed system-dependent threshold, set by each evaluee. If the evaluee believes the scores to be well-calibrated log-likelihood-ratios, then (s)he may use the Bayes decision threshold η. Otherwise, the threshold may be tuned by the evaluee to minimize DCF on a supervised calibration database.

The errors that result from the hard decisions on the supervised evaluation database are summarized as the empirical error-rates: P_{miss}, the ratio of misses to target trials; and P_{fa}, the ratio of false-alarms to non-target trials. The primary evaluation criterion is defined as:

$$\text{DCF} = \pi C_{\text{miss}} P_{\text{miss}} + (1-\pi) C_{\text{fa}} P_{\text{fa}} \tag{7}$$

It is important to realize that π is a *synthetic* parameter, which models the target prior in the domain of application. It does not necessarily reflect the proportion of targets in the evaluation database.

The DCF parametrization, $\pi, C_{\text{miss}}, C_{\text{fa}}$, can loosely be referred to as the DCF *operating point*. The DCF recipe requires the operating point to be *fixed and known* to the evaluee. Below we show how to relax this requirement.

2.4 Bayes Risk: criterion for goodness of log-likelihood-ratios

A small modification [6] to the DCF evaluation recipe makes it applicable to calibrated log-likelihood-ratios, rather than hard decisions: The evaluee submits log-likelihood-ratios (rather than decisions) and *the evaluator makes the decisions*. The requirement for well-calibratedness is enforced by the fact that the evaluator applies the above-defined *Bayes decision threshold*, η.

The error-rates now depend on the evaluator's threshold and we indicate this by the notation $P_{\text{miss}}(\eta)$ and $P_{\text{fa}}(\eta)$. Since the submitted log-likelihood-ratios are also scores, it should be clear that if the evaluator were to sweep η from $-\infty$ to ∞, then $P_{\text{miss}}(\eta), P_{\text{fa}}(\eta)$ would map out the familiar ROC/DET curve.

Let $\mathcal{L} = \ell_1, \ell_2, \cdots, \ell_t, \cdots$ be the log-likelihood-ratios computed by the system under evaluation for every trial, t, in the whole supervised evaluation database, so that:

$$P_{\text{miss}}(\eta) = \frac{1}{|\mathcal{T}|} \sum_{t\in\mathcal{T}} I(\ell_t < \eta), \qquad P_{\text{fa}}(\eta) = \frac{1}{|\mathcal{N}|} \sum_{t\in\mathcal{N}} I(\ell_t \geq \eta) \tag{8}$$

[5] DCF as defined here is often referred to as *actual* DCF, to distinguish it from minDCF, which will be defined later.

where I is the indicator function and $\mathcal{T}$ and $\mathcal{N}$ are the sets of indices belonging to target and non-target trials.

The resulting evaluation criterion, the *empirical Bayes risk*, is given by:

$$\begin{aligned} \mathcal{R}(\mathcal{L}|\pi, C_{\text{miss}}, C_{\text{fa}}) &= \pi C_{\text{miss}} P_{\text{miss}}(\eta) + (1-\pi) C_{\text{fa}} P_{\text{fa}}(\eta) \\ \text{where } \eta &= \log \frac{C_{\text{fa}}}{C_{\text{miss}}} - \operatorname{logit} \pi \end{aligned} \tag{9}$$

If the evaluator always applies a fixed, known DCF parametrization, $\pi, C_{\text{miss}}, C_{\text{fa}}$, then nothing essential has changed. For a 'calibrated' log-likelihood-ratio, the evaluee could just submit $\ell_t = s_t - \gamma + \eta$, where s_t is his original uncalibrated score and $s_t \geq \gamma$ is his original decision rule. In this case $\mathcal{R}$ would be numerically equal to DCF.

But, if the evaluator sweeps η over a range of values, then everything changes. Now mere shifting will not adequately calibrate the scores. Now scaling as well as finer details of the calibration mapping also matter. (After taking care of a few more details below, we will demonstrate this experimentally.)

The empirical Bayes risk as evaluation criterion for log-likelihood-ratios is discussed in detail in [7, 8, 1]. It can be interpreted as:

- A *proper scoring rule*, which encourages both good discrimination (i.e. a good DET-curve) as well as good probabilistic calibration (in the sense of [9]). See for example [10], Chapter 13, the section entitled 'The honest weatherman', for an insightful explanation.

- *Generalized cross-entropy* [11] between the evaluator's perfect empirical posterior given by the labels and the posterior $P(\text{target}|s, \pi)$ of the evaluee. This information-theoretical analysis provides useful inequalities to understand the essential properties of this evaluation criterion [1, Chapter 2].

2.4.1 The default system

Define the *default system*, which always outputs log-likelihood-ratio of zero, so that $\mathcal{L}_0 = 0, 0, \cdots$ for every trial. Notice that the posterior of the default system is the same as the prior: $P(\text{target}|\ell_t = 0, \pi) = \pi$. Making Bayes decisions with the default system is the same as making decisions with the prior alone.

It is easy to show [1, Chapter 2] that if the likelihood-ratios of a system, $\mathcal{L}$, are sufficiently well calibrated, then $\mathcal{R}(\mathcal{L}|\pi, C_{\text{miss}}, C_{\text{fa}}) \leq \mathcal{R}(\mathcal{L}_0|\pi, C_{\text{miss}}, C_{\text{fa}})$, for every operating point $\pi, C_{\text{miss}}, C_{\text{fa}}$. A system that fails this test at some operating point can be said to be *badly calibrated* at that operating point. At such operating points, on average, better Bayes decisions are obtained by *not* using the system.

2.4.2 Simplifying risk to error-rate

As shown above, a system that outputs well-calibrated likelihood-ratios can be expected to make useful (better than default) Bayes decisions at every operating point. It therefore seems reasonable to expect of an evaluation procedure to test calibration over a wide range of such operating points. The problem is that the Bayes risk, as we have defined it, is parametrized by three independent parameters, $\pi, C_{\text{miss}}, C_{\text{fa}}$. How can we design our evaluation recipe to take account of all operating points in this *three-dimensional* space?

This problem is solved by realizing that all these operating points can be represented by an equivalent one-dimensional range of operating points, which is much easier to cover with an evaluation recipe. We show how this is done.

Define the *effective prior* as:

$$\tilde{\pi} = \frac{\pi C_{\text{miss}}}{\pi C_{\text{miss}} + (1-\pi) C_{\text{fa}}} \tag{10}$$

and now parametrize the Bayes risk with $\tilde{\pi}$ and $\tilde{C}_{\text{miss}} = \tilde{C}_{\text{fa}} = 1$. This reparametrization leaves the Bayes decision threshold, η, *unchanged*:

$$\eta = -\operatorname{logit}\tilde{\pi} = \log\frac{C_{\text{fa}}}{C_{\text{miss}}} - \operatorname{logit}\pi \tag{11}$$

and the evaluation criterion, $\mathcal{R}$ is merely *scaled*:

$$\mathcal{R}(\mathcal{L}|\tilde{\pi}, 1, 1) = \frac{1}{\pi C_{\text{miss}} + (1-\pi) C_{\text{fa}}} \mathcal{R}(\mathcal{L}|\pi, C_{\text{miss}}, C_{\text{fa}}) \tag{12}$$

where the scaling factor is positive and is not a function of $\mathcal{L}$ or of the error-rates. This means that if we are comparing the *relative* benefits of two systems, say $\mathcal{L}_1$ and $\mathcal{L}_2$, then:

$$\mathcal{R}(\mathcal{L}_1|\tilde{\pi},1,1) \leq \mathcal{R}(\mathcal{L}_2|\tilde{\pi},1,1) \quad\Leftrightarrow\quad \mathcal{R}(\mathcal{L}_1|\pi, C_{\text{miss}}, C_{\text{fa}}) \leq \mathcal{R}(\mathcal{L}_2|\pi, C_{\text{miss}}, C_{\text{fa}})$$

from which we conclude that the two criteria are *equivalent* for evaluation purposes.[6]

2.5 Empirical Bayes error-rate: a practical evaluation recipe

We now define our final evaluation criterion for evaluating the goodness of log-likelihood-ratios. The *empirical Bayes error-rate* is $\mathcal{E}(\mathcal{L}|\tilde{\pi}) = \mathcal{R}(\mathcal{L}|\tilde{\pi},1,1)$, so that:

$$\mathcal{E}(\mathcal{L}|\tilde{\pi}) = \tilde{\pi} P_{\text{miss}}(-\operatorname{logit}\tilde{\pi}) + (1-\tilde{\pi}) P_{\text{fa}}(-\operatorname{logit}\tilde{\pi}) \tag{13}$$

This criterion is parametrized by the *single, scalar* parameter, $\tilde{\pi}$, or equivalently by the Bayes decision threshold, $-\operatorname{logit}\tilde{\pi}$. Again, we refer to this parameter as the *operating point.*

The *old operating point* defined by NIST for the SREs between 1997 and 2008 was at $\tilde{\pi} \approx 0.092$, while the *new operating point* of 2010 was at $\tilde{\pi} = 0.001$.

In this toolkit, we are interested in evaluation that spans operating points. By having confined the operating point to one dimension, this becomes do-able. By sweeping over the threshold, this criterion exercises the decision-making ability of log-likelihood-ratios in a similar way that the ROC/DET-curve exercises the potential decision-making ability of uncalibrated scores. In subsections below, we shall discuss two ways of sweeping the operating point: one is an integral, the other a plot.

[6] This equivalence still holds if we allow more general cost functions, which can have negative costs (i.e. rewards) for correct decisions. In this case, the relationship between the criteria is affine, rather than linear. [5]

2.5.1 The default system: reference for bad calibration

We provide two references which can be compared to $\mathcal{E}(\mathcal{L}|\tilde{\pi})$ to judge calibration of $\mathcal{L}$. The first, discussed here, is the upper boundary where calibration fails. The other (the familiar minDCF), discussed in the next subsection, is an ideal lower bound, where calibration is optimal.

The default system, $\mathcal{L}_0$, provides the reference error-rate:

$$\mathcal{E}(\mathcal{L}_0|\tilde{\pi}) = \min(\tilde{\pi}, 1 - \tilde{\pi}) \tag{14}$$

As mentioned above, a system $\mathcal{L}$, for which $\mathcal{E}(\mathcal{L}|\tilde{\pi}) > \mathcal{E}(\mathcal{L}_0|\tilde{\pi})$, is said to be badly calibrated at the operating point $\tilde{\pi}$, because then it would be better not to use the system.

2.5.2 minDCF: reference for ideal calibration

NIST's minDCF is obtained by allowing the *evaluator, who has access to the true class labels*, to choose an optimal threshold at every operating point:

$$\text{minDCF}(\mathcal{L}|\pi, C_{\text{miss}}, C_{\text{fa}}) = \min_{-\infty \leq \gamma \leq \infty} \pi C_{\text{miss}} P_{\text{miss}}(\gamma) + (1-\pi) C_{\text{fa}} P_{\text{fa}}(\gamma) \tag{15}$$

Here we are interested in the specialization of minDCF, where the costs are unity. In analogy with $\mathcal{E}$, we denote it $\mathcal{E}_{\text{min}}$:

$$\mathcal{E}_{\text{min}}(\mathcal{L}|\tilde{\pi}) = \text{minDCF}(\mathcal{L}|\tilde{\pi}, 1, 1) \tag{16}$$

Note:

$$\mathcal{E}(\mathcal{L}|\tilde{\pi}) \geq \mathcal{E}_{\text{min}}(\mathcal{L}|\tilde{\pi}) \leq \mathcal{E}(\mathcal{L}_0|\tilde{\pi}) \tag{17}$$

Like minDCF, $\mathcal{E}_{\text{min}}$ is a secondary evaluation criterion, which fulfils two functions:

- It provides an ideal reference value for judging calibration. If $\mathcal{E}$ and $\mathcal{E}_{\text{min}}$ are close, then the system can be said to be very well calibrated.
- In the earlier stages of the development of a speaker recognition algorithm, one is typically not interested in calibration, but just in the *potential* to make good decisions at some operating point. $\mathcal{E}_{\text{min}}$ provides a calibration-insensitive criterion, which can be evaluated over a range of different operating points.

2.5.3 Cllr: scalar summary of goodness of log-likelihood-ratios

The BOSARIS Toolkit provides two ways to sweep the operating point: one *integrates* out the operating point to give a scalar, summary criterion; and the other *plots* the error-rate as a function of the operating point. We discuss the integral here and the plot in the next subsection.

We can define the calibration-sensitive, scalar summary criterion of the goodness of log-likelihood-ratios, known as C_{llr}, by integrating out the operating point [7]:

$$\begin{aligned} C_{\text{llr}}(\mathcal{L}) &= k \int_{-\infty}^{\infty} \mathcal{E}(\mathcal{L}|\operatorname{logit}^{-1} x)\, dx \\ &= \frac{0.5}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \log_2(1 + e^{-\ell_t}) + \frac{0.5}{|\mathcal{N}|} \sum_{t \in \mathcal{N}} \log_2(1 + e^{\ell_t}) \end{aligned} \tag{18}$$

where $k > 0$ is an unimportant scale factor and $\operatorname{logit}^{-1} x = (1 + e^{-x})^{-1}$ is the inverse[7] of the logit function.

This criterion is further discussed in [1, 7, 8, 12]. It can be interpreted as a strictly proper scoring rule, empirical cross-entropy, negative log-likelihood and as optimization objective for logistic regression.

2.5.4 Normalized Bayes-error-rate plots

To plot $\mathcal{E}(\mathcal{L}|\tilde{\pi})$ as a function of the operating point, it is helpful to transform both the horizontal and vertical axes.

Using $\tilde{\pi} \in [0, 1]$ as the horizontal axis would compress interesting parts of the graph against the sides of the interval. We therefore use $\operatorname{logit} \tilde{\pi}$ on the horizontal axis instead. This axis now becomes infinite in both directions and we plot only a suitable interval, near the origin, $\operatorname{logit} 0.5 = 0$. Plotting an interval that is too wide is meaningless anyway, because in those regions the prior becomes so close to 0 or 1 that either the miss or the false-alarm counts drop to zero.

The vertical axis is non-linearly amplified by normalizing with $\mathcal{E}(\mathcal{L}_0|\tilde{\pi}) = \min(\tilde{\pi}, 1 - \tilde{\pi})$. If this were not done, low error-rates would compress all the interesting action against the bottom of the plot.

The *normalized Bayes-error-rate plot* can be described as a plot of (x, y) such that:

$$y = \frac{\mathcal{E}(\mathcal{L}|\operatorname{logit}^{-1} x)}{\mathcal{E}(\mathcal{L}_0|\operatorname{logit}^{-1} x)} \tag{19}$$

Figure 1 gives an example, using synthetic Gaussian scores to compare the true log-likelihood-ratio against some deliberately miscalibrated versions. This plot demonstrates:

- The deliberately miscalibrated 'systems' have worse error-rates than the (green) 'true LR' system, almost everywhere.

- The only region where the green system does worse than the miscalibrated dashed magenta is due to small sample effects. This is to the left of the red triangle, where the number of false-alarms becomes very low. The red and green triangles indicate the points were false-alarms and misses become scarce (less than 30) and therefore indicate the boundaries were small-sample effects may become a problem for meaningful evaluation. The safe region is between the two triangles. The error-rates that determine the horizontal positions of these triangles are obtained from the dashed black curve, where the evaluator has optimized the thresholds.

- The dashed black curve is $\mathcal{E}_{\min}(\mathcal{L}|\tilde{\pi})$. Between the triangles, it coincides closely to the theoretically optimal 'true LR' green curve. In real cases, we are *not* given a true probability model that generated the data, so that $\mathcal{E}_{\min}$ forms a useful practical reference for judging calibration.

- The solid black line at $y = 1$ represents the default performance of $\mathcal{E}(\mathcal{L}_0|\tilde{\pi})$. In places, the miscalibrated systems do worse than this reference. The only one which does not is the underoptimistic $0.5 \times \log \text{LR}$.

[7] $\operatorname{logit}^{-1}$ is also known as the *logistic sigmoid*.

(The reason why the dashed and solid black lines meet just to the right of +2 for this dataset is that the Gaussian log-likelihood-ratio as a function of the score is a parabola, with a minimum just below −2. The system never outputs log-likelihood-ratios with smaller values, so that in the far right of the plot, all decisions are identical to those made by the default system (i.e. accept).)

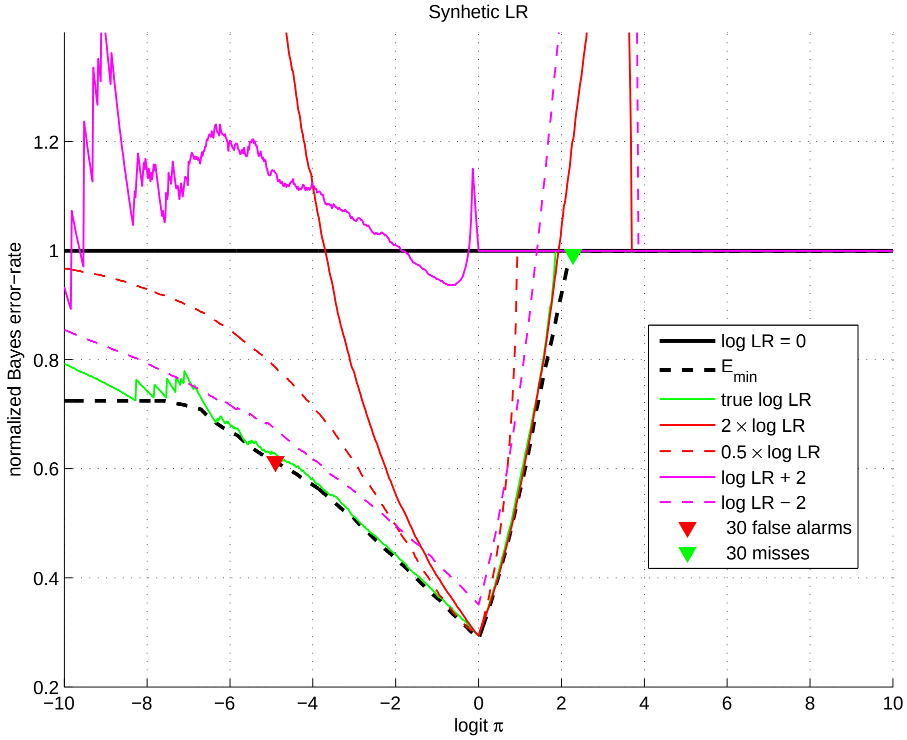

Figure 1: Normalized Bayes error-rate plot for a synthetic system with Gaussian scores: targets $\sim \mathcal{N}(\mu = 3, \sigma = 2)$ and non-targets $\sim \mathcal{N}(0, 1)$. The true likelihood-ratio is compared against deliberate additive and multiplicative miscalibrations.

Figures 2 and 3 show further examples of normalized Bayes error-rate plots, but now for real speaker recognition scores of systems submitted to SRE'10.

The plots show curves for tests on two databases: *dev* is the database used to train the calibration (SRE2008 eval database in this case) and *eval* is the evaluation database (SRE2010). The Bayes error-rate for the dev database is shown in dashed red and that for the eval database in solid red. The minimum Bayes error-rate (thick red) is only shown for the eval database. The toolkit can also plot the contributions of the misses and false alarms to both the minimum Bayes error-rate and actual Bayes error-rate. In the example plots, only the contributions to the actual Bayes error-rate are shown (misses in blue, false alarms in green). The new (SRE'10) operating point is shown on the plots by the vertical dashed magenta line at −6.91.

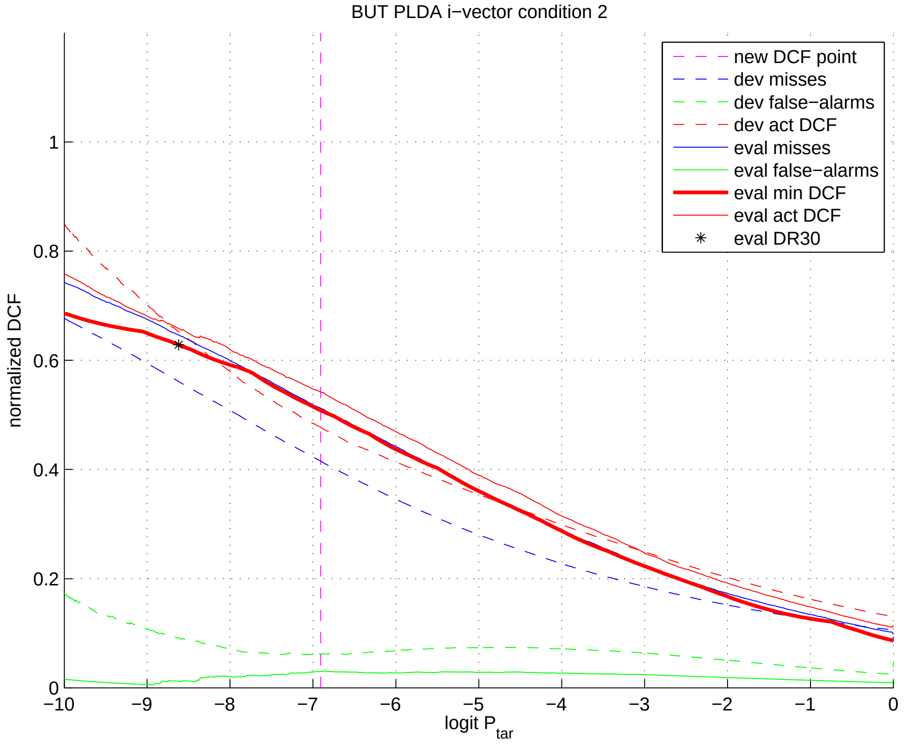

Figure 2: Normalized Bayes error-rate plot for an SRE 2010 speaker detector with *good* calibration. Here *eval* denotes the evaluation database and *dev* the development database. $P_{\text{tar}} = \tilde{\pi}$, while act DCF and min DCF refer to $\mathcal{E}$ and $\mathcal{E}_{\text{min}}$. P_{miss} and normalized P_{fa} are shown separately. DR30 refers to the point to the left of which there are fewer than 30 false-alarms. The vertical magenta dashed line represents the *new operating point* at $\tilde{\pi} = 0.001$.

In the region of interest, $x < 0$, which we plot in these figures, the vertical axis (normalized error-rate) is:

$$\begin{aligned} y &= \frac{\mathcal{E}(\mathcal{L}|\tilde{\pi})}{\min(\tilde{\pi}, 1-\tilde{\pi})} && (20)\\ &= \frac{\tilde{\pi} P_{\text{miss}}(\eta) + (1-\tilde{\pi}) P_{\text{fa}}(\eta)}{\tilde{\pi}} && (21)\\ &= P_{\text{miss}}(\eta) + \exp(-\operatorname{logit}\tilde{\pi}) P_{\text{fa}}(\eta) && (22)\\ &= P_{\text{miss}}(-\operatorname{logit}^{-1} x) + \exp(-x) P_{\text{fa}}(-\operatorname{logit}^{-1} x) && (23) \end{aligned}$$

The exponential amplification of false-alarms induced by this normalization explains the shape of the curves for regions of bad calibration. Some form of amplifying normalization is needed to make the effects of calibration visible in regions of low error-rate. This normalization is the main difference between these curves and APE-curves [7]. The normalized Bayes error-rate plot is able

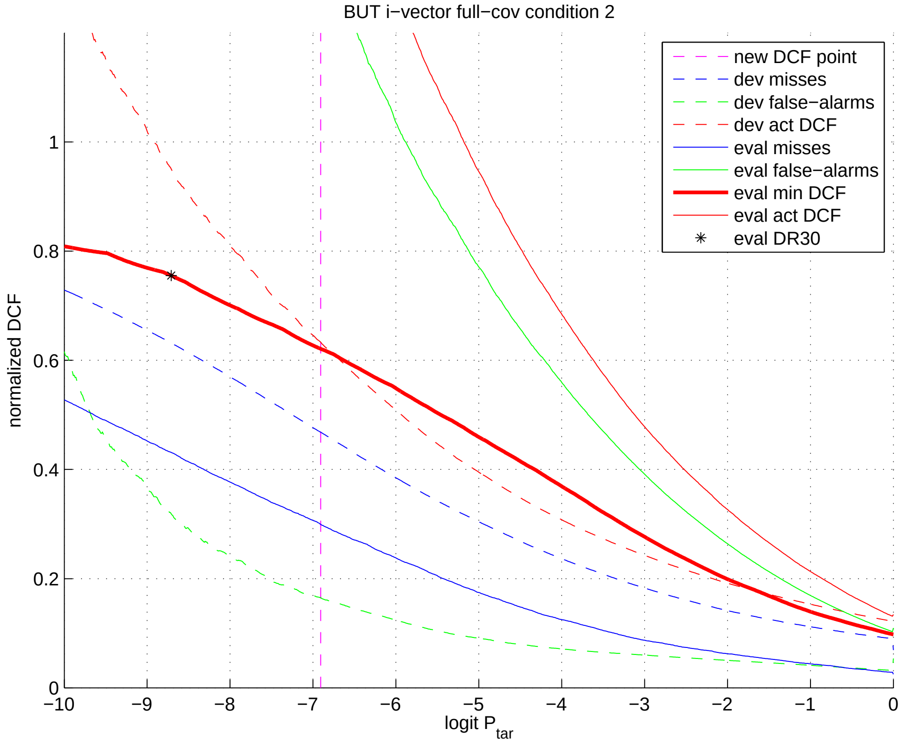

Figure 3: Normalized Bayes error-rate plot for an SRE 2010 speaker detector with *bad* calibration. See caption of figure 2 for details.

to display a wider range of operating points than the APE-curve.

The points in the plot marked with asterisks (we used triangles in the first plot), labelled DR30 refer to Doddington's Rule of 30 [4]. This rule suggests you need at least 30 false-alarms and at least 30 misses for meaningful evaluation. The toolkit can plot both the DR30 point for the misses (to the right of which the absolute number of misses drops below 30) and the one for the false alarms (to the left of which the absolute number of false-alarms drops below 30). These points are on the $\mathcal{E}_{\text{min}}$ curve, because we use the false-alarm count and miss count that result from the evaluator's optimized threshold.

2.6 ROC/DET and related criteria for goodness of scores

This subsection deals with ROC/DET curves and associated summaries such as EER and minDCF, all of which can be applied for calibration-insensitive evaluation of the goodness of uncalibrated scores. This is useful for the earlier stages of algorithm development, when calibration is not of immediate interest.

We assume the reader is familiar with the ROC (receiver operating characteristic) [13]. In this section we concentrate on perhaps unfamiliar relationships that exist between the ROC, minDCF and EER. In summary: the ROC spans operating points by plotting error-rates as a function of the threshold; minDCF samples the ROC at a fixed operating point; EER summarizes the span of operating points by *maximizing* over minDCF as a function of the operating point. The *ROC convex hull* is central to this analysis and also provides the key to efficient minDCF and EER calculation.

In our discussion below, we use the term ROC, but (unless otherwise noted) everything applies also to DET-curves [14]. For ROC, we assume the speaker-recognition convention where $x = P_{\text{fa}}$ is on the horizontal axis and $y = P_{\text{miss}}$ on the vertical axis.[8] The DET-curve differs from the ROC by axis warping:[9] $x = \text{probit}(P_{\text{fa}})$ and $y = \text{probit}(P_{\text{miss}})$.

There are some aspects of the ubiquitous ROC/DET that seem to be misunderstood by many of its users. Here we highlight the following:

- The ROC is an *optimistic* view of the decision-making ability of scores, because calibration is not tested. If Bayes risk is minimized (i.e. minDCF) at a particular operating point 'on the ROC curve', then the calibration problem remains of how to choose a threshold that will place the actual performance at this operating point. This actual performance is usually worse (and cannot be better) than minDCF.
- The empirical ROC is not a continuous curve. It is a collection of discrete points in $(P_{\text{fa}}, P_{\text{miss}})$ space, where every point corresponds to a decision threshold between adjacent scores. If the points are connected with line segments[10] then those segments are either vertical or horizontal, corresponding to target and non-target scores. We shall refer to this plot as the *steppy ROC.*
- minDCF operating points do not live exactly on the steppy ROC. They live on the *ROCCH curve*: the lower left boundary of the convex hull around the discrete points of the ROC.
- Although the EER is fixed at $P_{\text{miss}} = P_{\text{fa}}$, it nevertheless forms a summary of the whole curve: it is a tight upper bound of the decision making ability over all operating points. Using EER as optimization objective is a good idea, because forcing the tight upper bound down, forces the whole curve down. This can be generalized to any other point on the ROCCH curve by fixing the ratio $\frac{P_{\text{miss}}}{P_{\text{fa}}}$.

We elaborate on the last two points below.

2.6.1 The ROCCH is where minDCF lives

Let there be n points, $[p_{\text{fa}}(i), p_{\text{miss}}(i)]$ in the empirical ROC. A point in $\mathbb{R}^2$ is in the *convex hull* of the ROC, if and only if it is a two-dimensional interpolation between all of the ROC points. That is, a point

$$[x, y] = \sum_{i=1}^{n} \alpha_i [p_{\text{fa}}(i), p_{\text{miss}}(i)] \tag{24}$$

[8] In other fields, the vertical axis is $1 - P_{\text{miss}}$.

[9] The probit function maps $[0, 1]$ to $[-\infty, \infty]$ in a very similar way to the logit function: $\text{probit}(p) = \sqrt{2}\,\text{erf}^{-1}(2p - 1)$.

[10] assuming no two scores coincide

is in the convex hull if and only if all $\alpha_i \geq 0$ and $\sum_{i=1}^{n} \alpha_i = 1$.

We already know that minDCF can be expressed either as a continuous minimization over the threshold (γ), or as a discrete minimization over the ROC points. But it can also be expressed [15, 1] as a continuous minimization over the convex hull, or as a discrete minimization over the set of vertices, $\mathcal{V}_{ch}$, of the convex hull:

$$\begin{aligned}
\text{minDCF}(\pi, C_\text{miss}, C_\text{fa}) &= \min_{\gamma} \pi C_\text{miss} P_\text{miss}(\gamma) + (1-\pi) C_\text{fa} P_\text{fa}(\gamma) \\
&= \min_{i=1}^{n} \pi C_\text{miss} p_\text{miss}(i) + (1-\pi) C_\text{fa} p_\text{fa}(i) \\
&= \min_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i \big(\pi C_\text{miss} p_\text{miss}(i) + (1-\pi) C_\text{fa} p_\text{fa}(i) \big) \\
&= \min_{i \in \mathcal{V}_{ch}} \pi C_\text{miss} p_\text{miss}(i) + (1-\pi) C_\text{fa} p_\text{fa}(i)
\end{aligned} \tag{25}$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]$ is subject to the above-mentioned convexity constraint. This means that although parts of the convex hull seem more optimistic than the steppy ROC, these parts do not give lower minDCF, no matter what the operating point.

The DCF minima live on the lower left boundary of the convex hull, which forms a continuous, piecewise linear, convex curve between the points $(0, 1)$ and $(1, 0)$. We shall refer to this curve as the *ROCCH curve*.

The BOSARIS Toolkit provides the functionality to compute the ROCCH curve, as well as the associated DET-curve obtained by applying the non-linear (probit) mapping to the axes.[11] Figure 4 shows two examples. For further examples, see [1, Chapter 7], or [16], or try to plot some of your own, using the toolkit.

The ROCCH vertex set, $\mathcal{V}_{ch}$, is typically much, much smaller than the empirical ROC. Since the convex hull can be computed efficiently (see the PAV algorithm below), and since it is valid for all operating points, this is the key to efficient minDCF computations for large score sets, over a large range of operating points.

2.6.2 EER as upper bound

The EER (equal-error-rate) is usually defined as the ‘point on the ROC’, where $P_\text{miss} = P_\text{fa}$. For the empirical ROC, in general, no point exactly satisfies this equality, but it can be satisfied by interpolation. If we choose to interpolate between *all* points in the ROC, we again find ourselves on the ROCCH curve. We denote the point on the ROCCH curve where $P_\text{miss} = P_\text{fa}$ as the ROCCH-EER. We propose to use the ROCCH-EER as a well-defined, practical version of the EER and this functionality is provided as such by the toolkit.

The ROCCH-EER has the following interesting property [1]:

$$\begin{aligned}
\text{ROCCH-EER} &= \max_{\tilde{\pi}} \min_{-\infty \leq \gamma \leq \infty} \tilde{\pi} P_\text{miss}(\gamma) + (1-\tilde{\pi}) P_\text{fa}(\gamma) \\
&= \max_{\tilde{\pi}} \text{minDCF}(\tilde{\pi}, 1, 1)
\end{aligned} \tag{26}$$

Figure 4 demonstrates this.

[11] The convexity does not hold when these curves are translated to DET space.

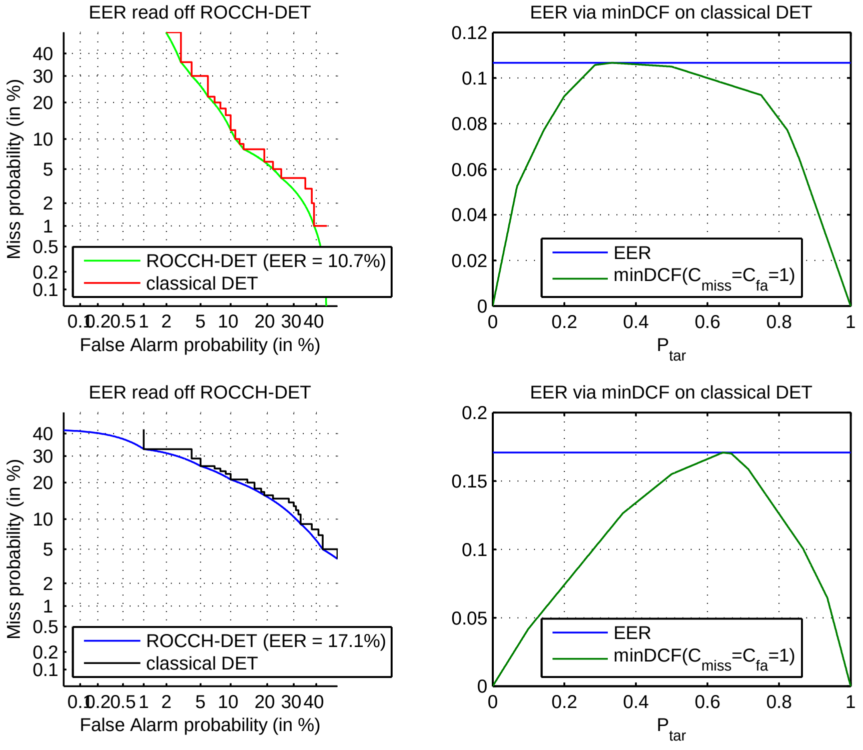

Figure 4: Two examples of ROCCH-DET vs classical steppy DET. The equality of ROCCH-EER and max minDCF is demonstrated. (Here $\tilde{\pi} = P_{\text{tar}}$).

ROCCH-EER is obtained by maximizing w.r.t. the operating point, while minimizing w.r.t. the threshold. The minimization confines us to the ROCCH curve, while the maximization finds the most *pessimistic* operating point on this curve. The ROCCH-EER therefore forms a tight upper bound on the Bayes error-rate that can be obtained with perfect calibration. By pushing down on the EER, we are pushing down the whole curve.

Another way to see this is the fact that minDCF$(\tilde{\pi}, 1, 1)$ is a *concave* function (see figure 4). If we push down at the maximum of this curve (by trying to build a system that gets better EER) it cannot form a dent in the curve that violates concavity. If anything moves, the whole curve has to go down in such a way as to respect concavity.

This does *not* guarantee that if we reduce ROCCH-EER, we will have reduced minDCF at *all* operating points. Even if the value of the maximum is reduced, its position, $\tilde{\pi}$, can move in such a way that error-rates can increase somewhere far from the maximum. This lateral movement is roughly analogous to tilting of the DET-curve. If, however, we want to target a specific region of operating points of interest, we can generalize this idea. This is shown in the next subsection.

2.6.3 UER: Unequal-error-rate

We can generalize ROCCH-EER by considering a point, $[\check{P}_{\text{fa}}, \check{P}_{\text{miss}}]$, on the ROCCH curve where $\check{P}_{\text{fa}} = r\check{P}_{\text{miss}}$. For any $r \neq 1$, this is an *unequal-error-rate.*

Such points also have the interpretation that they form tight upper bounds on minDCF. To see this, choose any costs such that: $C_{\text{miss}} = rC_{\text{fa}}$. We can show [1] that there exists a point, $[\check{P}_{\text{fa}}(r), \check{P}_{\text{miss}}(r)]$, on the ROCCH curve, such that:

$$C_{\text{miss}}\check{P}_{\text{miss}}(r) = C_{\text{fa}}\check{P}_{\text{fa}}(r) = \max_{\pi} \text{minDCF}(\pi, C_{\text{miss}}, C_{\text{fa}}) \tag{27}$$

The point on the curve depends just on the ratio r. By varying r between zero and infinity, we can map out the whole ROCCH curve.[12] If we arbitrarily set $C_{\text{fa}} = 1$ and $C_{\text{miss}} = r$, we can define the unequal-error-rate as:

$$\text{UER}(r) = \check{P}_{\text{fa}}(r) = r\check{P}_{\text{miss}}(r) = \max_{\pi} \text{minDCF}(\pi, r, 1) \tag{28}$$

Again, this value forms a tight upper bound of a concave function of π, so that using UER as optimization objective pushes down the whole curve. If we choose $r = \tilde{\pi}$, then we will be targeting operating points in the vicinity of $\tilde{\pi}$.

In summary, the whole ROC/DET curve has this 'stiffness' property induced by the concavity, so that trying to optimize some point on the curve will tend to also improve the decision-making ability of the curve over a larger region.

2.6.4 PRBEP

Finally, we mention another variant on this idea, where we re-weight the error-rates to represent absolute error counts. By choosing $C_{\text{miss}} = T$, the number of target trials; and $C_{\text{fa}} = N$, the number of non-target trials, the toolkit provides the functionality to compute the *precision-recall-break-even-point*:

$$\text{PRBEP} = N \times \text{UER}\big(\frac{T}{N}\big) = T\check{P}_{\text{miss}} = N\check{P}_{\text{fa}} = \max_{\pi} \text{minDCF}(\pi, T, N) \tag{29}$$

which represents the point on the ROCCH curve where the *absolute* number of misses and false alarms are equal.[13]

If the error-rates of the recognizer are low relative to the number of available evaluation trials, then this forms a sensible evaluation objective, which balances the two error-counts, keeping them both from becoming too small for as long as possible.

Here we prefer to present the result as an absolute number of errors, rather than as an error-rate, so that if the number of errors becomes small, the user is effectively warned that this is happening.

PRBEP cannot be used for meaningful comparisons across databases of different sizes. It is meant for comparison of different systems on the same database.

[12] Interestingly, if we exchange max and min, the error-rates that satisfy $\min_{\gamma} \max_{\pi} \text{DCF}(\gamma|\pi, r, 1)$, map out the steppy ROC as we vary r.

[13] Since the ROCCH curve is an interpolation, this will in general not be a whole number.

2.7 Fusion and Calibration

The toolkit provides two solutions for calibration, which is the task of finding a mapping ℓ, that maps scores to log-likelihood-ratios. In both cases, the mapping is 'trained' on a supervised calibration database. One solution is non-parametric, based on isotonic regression. The other is parametric, based on logistic regression. The logistic regression solution generalizes also to a fusion recipe.[14]

The non-parametric calibration finds a solution that is (on the *training* data) simultaneously optimal for *any* sensible objective function[15] for measuring the goodness of calibration [1, Appendix C]. In practice however, we have found that the parametric solution usually performs better on independent test data.

2.7.1 PAV: Non-parametric calibration

The convention that the larger the score, the more it favours the target hypothesis, suggests that the calibration mapping, ℓ, should be monotonically rising (isotonic) [17]. Since we have a finite number of training scores, each of which must be mapped to a log-likelihood-ratio, this can be done in a non-parametric way. We can independently choose the value for each point, subject only to the monotonicity constraint. This problem is known as *isotonic regression* and an efficient implementation is given by the PAV (pool adjacent violators) algorithm, which we discuss in the next section.

Attractive features of this solution are:

- On training data, as mentioned above, it is optimal, no matter how you measure optimality.
- It corresponds exactly to $\mathcal{E}_{\text{min}}$ (minDCF): If a data set is optimized with PAV, and then evaluated on the same data set with $\mathcal{E}$ (DCF), then DCF = minDCF.
- It also corresponds exactly to using the *slope* of the ROCCH curve as calibrated likelihood-ratio [18].
- The type of the score distribution is unimportant. In fact, the procedure is invariant to any monotonic warping of the scores. In contrast, the parametric logistic regression calibration solution below works best for approximately normal score distributions.

2.7.2 Logistic regression: parametric fusion and calibration

The toolkit provides a logistic regression solution, which can:

- train a calibration mapping, $\ell(s)$, for a single system;
- train combination weights to fuse multiple subsystems into a single subsystem which outputs well-calibrated log-likelihood-ratios; and
- also incorporate certain kinds of side-information, or quality measures.

[14] It is shown in [15], that isotonic regression can also be used for fusion, but this is not yet implemented in the toolkit.

[15] This is, any strict, or non-strict proper scoring rule, or Bayes risk criterion.

All of this functionality is provided by optimization of the parameters of the following mapping:

$$\ell_t = a + \sum_{i=1}^{N} b_i s_{it} + \mathbf{q}_t' \mathbf{W} \mathbf{r}_t \tag{30}$$

where ℓ_t is the fused and calibrated output log-likelihood-ratio for trial t; N is the number of subsystems to be fused (if $N = 1$, then the result is just calibration); s_{it} is the score of subsystem i for trial t; $\mathbf{q}_t$ and $\mathbf{r}_t$ are optional 'quality vectors', derived from the two sides (enrol, verify) of trial t. The parameters to be optimized are the scalar offset a, the scalar combination weights b_i and a symmetric matrix $\mathbf{W}$, which effectively combines the two quality vectors into a quality score for the trial.

The parameters are optimized with logistic regression, which minimizes an objective function, which is very similar to the above-defined C_{llr}. This objective function is the evaluation criterion for a *supervised calibration database*, which must be provided by the user. Since the objective function is calibration sensitive, optimizing it causes the fused output to be well calibrated. See [19], or [1, Chapter 8] for more details.

3 Algorithms

This section describes the key algorithms that help the toolkit to efficiently process very large sets of scores.

3.1 Efficient DCF and minDCF

This subsection describes efficient algorithms for computing DCF and minDCF. With more traditional implementations, computation of $\mathcal{E}$ (DCF) and $\mathcal{E}_{\text{min}}$ (minDCF), over the range required by a normalized Bayes error-rate plot, may take several minutes for large trial lists (a few million scores). By comparison, the implementation in the BOSARIS Toolkit takes a few seconds to execute.

3.1.1 DCF

To efficiently compute $\mathcal{E}$, pool all the scores, $\mathcal{L} = \ell_1, \ell_2, \ldots$, with all the different thresholds, $-\operatorname{logit} \tilde{\pi}_i$, at which $\mathcal{E}(\mathcal{L}|\tilde{\pi}_i)$ is to be evaluated. Sort them all together, in increasing order, keeping track of where the thresholds end up. The miss and false-alarm rates at threshold i are given by

$$P_{\text{miss}}(i) = \{t_i - (D - i + 1)\}/T \tag{31}$$

$$P_{\text{fa}}(i) = \{N - (n_i - (D - i + 1))\}/N \tag{32}$$

where t_i is the position of the ith threshold in the sorted list (after deleting non-target scores), n_i is the position of the ith threshold in the sorted list (after deleting target scores), D is the number of thresholds, T is the number of target scores and N is the number of non-target scores. Equation 13 then gives $\mathcal{E}$.

3.1.2 minDCF

To efficiently compute $\mathcal{E}_{\min}$, compute the vertices of the ROCCH curve, using the PAV algorithm (see section 3.2). There are typically very few of these vertices and as shown in section 2.6.1, the original large ROC can be replaced with these vertices, without changing the value of minDCF. Then use the last line of (25).

3.2 The PAV Algorithm

The PAV (pool adjacent violators) algorithm is central to the efficient implementation of many of the toolkit functions. We use it to efficiently compute the vertices[16] of the ROCCH curve [18]. Once we have these vertices, we can compute minDCF, EER, UER, PRBEP and the non-parametric calibration mapping (see the relevant subsections in the theory section).

The PAV algorithm solves the problem of assigning a likelihood-ratio to each score in some supervised database of target and non-target scores. The likelihood-ratios are adjusted non-parametrically and independently, subject only to the monotonicity constraint that if the scores are sorted, then the likelihood-ratios must also be sorted. The PAV solution turns out to be simultaneously optimal for any proper scoring rule and therefore for any Bayes risk criterion, with any cost function and any prior [1, Appendix C].

The PAV algorithm complexity is linear in the number of scores and the preceding sort has complexity of order $T\log(T)$. In our implementation, sorting and applying PAV takes a few seconds for a few million scores.

3.3 New logistic regression optimizer

The BOSARIS Toolkit uses a general-purpose, unconstrained convex optimization algorithm to train the logistic regression fusion and calibration solutions. It uses a quasi-Newton method, which is faster, generally better behaved and converges to a better solution than the conjugate gradient optimizer which was used in its predecessor, the FoCal Toolkit.[17]

The new optimizer uses the *trust region Newton conjugate gradient algorithm* for large-scale unconstrained minimization [20, 21].

4 Code

This section gives a high-level overview of some of the salient features of the implementation of the algorithms. More detail is available in the user manual which is distributed with the toolkit.

The current implementation is written in MATLAB, with an object-oriented API (application programmer's interface). The objects are not an essential part of the code,[18] they are just a way to organize the API. If this type of interface turns out to be a hindrance rather than a help to users, it would be possible to replace this API.

[16] The vertices of the whole convex hull are the same as the vertices (cusps) of the piece-wise linear ROCCH curve.

[17] Available at: `http://sites.google.com/site/nikobrummer/focal`

[18] MATLAB object oriented code does not scale well to large problems.

The main feature of the code that remains to be highlighted in this last section is the efficient, binary, platform-independent score file format. The efficiency of the format relies on the assumption that trial lists can be represented as dense matrices, where the row and column indices are the two sides (enrol, verify) of a trial. We assume that each enrolment or each verification side is to be matched against many—or even all—others. (Such dense score matrices were necessary for ensuring an adequate number of non-target trials and therefore an adequate number of false-alarms at the new operating point, $\tilde{\pi} = 0.001$, of SRE'10.)

We use a platform-independent HDF5 binary score format to encourage interoperability with other tools. Text files would also give interoperability, but are much larger and much slower to process.

4.1 Data

The code in the toolkit is primarily concerned with storing and manipulating the following data types:

indexes list model and test segment names and indicate which pairs of model and test segment are in the trial list described by the index.

keys are similar to indexes, but also give the answers i.e. which trials are target trials and which are non-target trials.

scores store scores for a list of trials (specified by an index or a key). In addition to the actual scores, a score object contains all the information that an index describes.

quality measures can be seen as scores for a model or test segment (instead of for a trial). These can be fused with ordinary scores (see section 2.7).

Indexes can be used:

- for aligning scores from different systems before fusing them
- for selecting parts of score objects of interest (e.g. those for male trials)
- by external code that produces scores. This code can load an index file which indicates which segment pairs to produce scores for.

Two score objects can be merged to make a new score object provided that they don't provide scores for the same trial. Parts of score objects can be selected (to produce a new score object) either by using an index or by using lists of models or segments to discard.

4.2 Plots

The toolkit can produce two types of plots:

DET plots (see section 2.6) either from points on the ROC or from the ROCCH curve.

Normalized Bayes error-rate plots (see section 2.5.4). Both minimum and actual Bayes error-rate curves can be plotted, as well as curves showing the contributions of the misses and false alarms, respectively, to those curves. A vertical line indicating the operating point can be placed on the plot.

DR30 points (see section 2.1) for misses and false alarms can be placed on both of types of plots.

4.3 Calibration

The high level wrapper functions for calibration have two variants: those that train the calibration transformation on a single set [19] of scores and then apply that transformation to the same set, and those that train the transformation on one set of scores (*dev*) and apply it to another set (*eval*). A second partitioning of the functions can be made according to whether the transformation is affine or whether it uses the PAV algorithm (see section 3.2).

4.4 Fusion

The main functions for doing fusion can again be divided (as for calibration) according to whether there is a set of unsupervised *eval* scores in addition to the *dev* scores or not. There are separate wrapper functions for doing fusion when quality measures are to be used.

4.5 Other functions

There are functions for calculating EER, minimum DCF, actual DCF, PRBEP and the effective prior.

4.6 File format

With approximately eight million trials in our development list for SRE'10, loading and saving score files in text format became unfeasible. We therefore created a binary file format which both reduced the size of the file on disk and made loading and saving faster. For example, one of our *tel-tel* development files is about 60 times larger on disk in text format than in binary format and the binary file loads about 160 times faster than the text file.

The binary score files contain two lists and two matrices. The lists contain the model and test segment names. One matrix contains the scores as real numbers and the other matrix is a logical matrix of the same size which indicates which scores correspond to valid trials. The dimensions of the matrices are the number of models by the number of test segments and the score at position (i, j) is for a trial between the ith model in the model list and the jth test segment in the test segment list.

The toolkit provides both Matlab *.mat* and HDF5 versions of the binary format, as well as functions for converting between binary and text formats.

[19] By *set*, we mean *multiset*, because the collection should retain duplicate values.

5 Acknowledgements

We would like to thank our SRE'10 collaborators, BUT and CRIM, as well as the other participants of the BOSARIS Workshop for their contributions, especially Lukáš Burget, Oldřich Plchot and Nicolas Scheffer.

References

[1] Niko Brümmer, *Measuring, Refining and Calibrating Speaker and Language Information Extracted from Speech*, Ph.D. thesis, University of Stellenbosch, Stellenbosch, South Africa, Dec. 2010.

[2] The National Institute of Standards and Technology, "The NIST year 2010 speaker recognition evaluation plan," `http://www.itl.nist.gov/iad/mig//tests/sre/2010/NIST_SRE10_evalplan.r6.pdf`, Apr. 2010.

[3] Lawrence D. Brown, T. Tony Cai, and Anirban DasGupta, "Interval estimation for a binomial proportion," *Statistical Science*, vol. 16, no. 2, pp. 101–133, 2001.

[4] George R. Doddington, "Speaker recognition evaluation methodology: a review and perspective," in *Proceedings of RLA2C Workshop: Speaker Recognition and its Commercial and Forensic Applications*, Avignon, France, Apr. 1998, pp. 60–66.

[5] Morris H. DeGroot, *Optimal Statistical Decisions*, McGraw-Hill, 1970.

[6] Niko Brümmer, "Application-independent evaluation of speaker detection," in *Proceedings of the Odyssey Speaker and Language Recognition Workshop*, Toledo, Spain, June 2004, pp. 33–40.

[7] Niko Brümmer and Johan A. du Preez, "Application-independent evaluation of speaker detection," *Computer Speech and Language*, vol. 20, no. 2–3, pp. 230–275, 2006.

[8] David A. van Leeuwen and Niko Brümmer, "An introduction to application-independent evaluation of speaker recognition systems," in *Speaker Classification I: Fundamentals, Features, and Methods*, Christian Müller, Ed., Lecture Notes in Computer Science / Lecture Notes in Artificial Intelligence, pp. 330–353. Springer, first edition, 2007.

[9] Morris H. DeGroot and Stephen E. Fienberg, "The comparison and evaluation of forecasters," *The Statistician*, vol. 32, pp. 14–22, 1983.

[10] Edwin T. Jaynes, *Probability Theory: The Logic of Science*, Cambridge University Press, 2003.

[11] A. Philip Dawid, "Coherent measures of discrepancy, uncertainty and dependence, with applications to Bayesian predictive experimental design," Technical Report 139, Department of Statistical Science, University College London, Aug. 1998, Online: `http://www.ucl.ac.uk/Stats/research/reports/abs94.html#139`.

[12] Daniel Ramos-Castro, *Forensic Evaluation of the Evidence Using Automatic Speaker Recognition Systems*, Ph.D. thesis, Universidad Autónoma de Madrid, Madrid, Spain, Nov. 2007.

[13] Tom Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, June 2006.

[14] Alvin Martin, George Doddington, Terri Kamm, Mark Ordowski, and Mark Przybocki, "The DET curve in assessment of detection task performance," in *Proceedings of the 5th European Conference on Speech Communication and Technology, EUROSPEECH*, Rhodes, Greece, Sept. 1997, pp. 1895–1898.

[15] Foster J. Provost and Tom Fawcett, "Robust classification for imprecise environments," *Machine Learning*, vol. 42, no. 3, pp. 203–231, Mar. 2001.

[16] Guido Aversano, Niko Brümmer, and Mauro Falcone, "EVALITA 2009 speaker identity verification "application" track organizer's report," in *Proceedings of EVALITA*, Reggio Emilia, Italy, Dec. 2009, Online: `http://evalita.fbk.eu/proceedings.html`.

[17] Bianca Zadrozny and Charles Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *Proceedings of the Eighth International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, 2002, pp. 694–699.

[18] Tom Fawcett and Alexandru Niculescu-Mizil, "PAV and the ROC convex hull," *Machine Learning*, vol. 68, no. 1, pp. 97–106, July 2007.

[19] Niko Brümmer, Lukáš Burget, Jan "Honza" Černocký, Ondřej Glembek, František Grézl, Martin Karafiát, David A. van Leeuwen, Pavel Matějka, Petr Schwarz, and Albert Strasheim, "Fusion of heterogenous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2072–2084, Sept. 2007.

[20] Jorge Nocedal and Stephen J. Wright, *Numerical Optimization*, Springer, second edition, 2006.

[21] Chih-Jen Lin, Ruby C. Weng, and S. Sathiya Keerthi, "Trust region Newton method for large-scale logistic regression," *Journal of Machine Learning Research*, Sept. 2008.