GENERATIVE MODELLING FOR UNSUPERVISED SCORE CALIBRATION

*Niko Brümmer**

AGNITIO Research
Somerset West, South Africa

Daniel Garcia-Romero

HLTCOE, Johns Hopkins University
Baltimore, MD, USA

ABSTRACT

Score calibration enables automatic speaker recognizers to make cost-effective accept / reject decisions. Traditional calibration requires supervised data, which is an expensive resource. We propose a 2-component GMM for unsupervised calibration and demonstrate good performance relative to a supervised baseline on NIST SRE'10 and SRE'12. A Bayesian analysis demonstrates that the uncertainty associated with the unsupervised calibration parameter estimates is surprisingly small.

Index Terms— calibration, unsupervised learning, Laplace approximation, automatic speaker recognition

1. INTRODUCTION

Automatic speaker recognizers map trials to scores. A *trial* has two parts: some speech of a known speaker, and some of an unknown speaker. When the same speaker is present in both parts, we have a *target* trial. When the speakers differ, we have a *non-target* trial. The *score* is a real number, which is expected to be larger (more positive) for target trials and smaller (more negative) for non-target trials. When a speaker recognizer is deployed in a new environment, which may differ from previously seen environments w.r.t. factors like language, demographics, vocal effort, noise level, microphone, transmission channel, duration, etc., the behaviour of the scores may change. Although the scores can still be expected to discriminate between targets and non-targets in the new environment, score distributions could change between environments.

If scores are to be used to make hard decisions, then we need to *calibrate* the scores for the appropriate environment. The ideal calibration of a score, s, would be of the form:

$$s \to \log \frac{P(s|\text{target, environment})}{P(s|\text{non-target, environment})}$$

but of course, we are not given these score distributions. Our only resource would be some data collected from the new environment. In special cases (usually involving considerable expense), this data can be supervised, such that each trial is labelled as target or non-target. In a more realistic scenario however, all or most of this data would be *unsupervised*.

To date, most works on calibration have made use of supervised data. In this paper, we explore the problem of calibration where our only resource is a large database of completely unsupervised scores.

2. CALIBRATION MODEL

In the supervised setting, score calibration can be viewed as a straight-forward, 2-class pattern recognition problem, for which both generative and discriminative solutions exist [1, 2, 3, 4, 5]. For the unsupervised case, we found the generative approach more convenient. Here we introduce the score model for supervised calibration, followed by a generalization to the unsupervised case.

2.1. Supervised calibration model

For the supervised case, we adopt the simple generative model of [4]. Denoting targets as H_1 and non-targets as H_2, we model a score $s \in \mathbb{R}$, with class-conditional Gaussian distributions:

$$P(s|H_i, \mathcal{C}) = \mathcal{N}(s|\mu_i, \sigma^2) \tag{1}$$

where μ_1, μ_2 are class-conditional means and σ^2 is the common within-class variance. We collectively refer to $\mathcal{C} = (\mu_1, \mu_2, \sigma^2)$ as the *calibration parameters*. This model gives an affine calibration transformation, from score to log-likelihood-ratio, of the form:

$$\log R(s|\mathcal{C}) = \log \frac{\mathcal{N}(s|\mu_1, \sigma^2)}{\mathcal{N}(s|\mu_2, \sigma^2)} = \frac{d'}{\sigma} s + \frac{\mu_2^2 - \mu_1^2}{2\sigma^2} \tag{2}$$

where d', the separation between targets and non-targets [6],

$$d' = \frac{\mu_1 - \mu_2}{\sigma}, \tag{3}$$

represents *accuracy*, since the theoretical equal-error-rate is EER $= \Phi(-\frac{d'}{2})$.[1] At complete overlap, $d' = 0$ and EER $= 0.5$. As d' increases, the EER decreases.

We refer to $R(s|\mathcal{C})$ as the *plug-in LR*, because it can be calculated only if $\mathcal{C}$ is given. In reality, these parameters are not given, so they must be estimated before being plugged into (2). When labelled calibration training scores are given, maximum-likelihood parameter estimates are straight-forward—see [4], where this plug-in recipe is shown to give similar accuracy to logistic regression calibration.

2.2. Unsupervised calibration model

In the unsupervised case, we are given a collection of T training scores, denoted $\mathcal{S} = s_1, \ldots, s_T$, but we are *not* given the corresponding class labels. Denoting these labels as $\mathcal{L} = \ell_1, \ldots, \ell_T$, we treat them as hidden variables and our calibration model generalizes to a 2-component Gaussian mixture model (GMM), for which we need an additional mixture-weight parameter, π_1. Letting $\pi_2 = 1 - \pi_1$, the *GMM likelihood* is:

$$P(\mathcal{S}|\mathcal{M}) = \prod_{t=1}^{T} \sum_{i=1}^{2} \pi_i \mathcal{N}(s_t|\mu_i, \sigma^2) \tag{4}$$

*The experiments were done while attending the CLSP 2013 Speaker Recognition Workshop at Johns Hopkins University.

[1] Φ is the *normal cumulative density*, given in terms of the *error-function* as $\Phi(x) = (1 + \text{erf}(x/\sqrt{2}))/2$.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

where $\mathcal{M} = (\mathcal{C}, \pi_1) = (\mu_1, \mu_2, \sigma^2, \pi_1)$ are the GMM parameters.

Now consider a new *test score*, s', generated by the same model, and with associated hidden class label $\ell' \in \{H_1, H_2\}$. The conditional dependency structure of all the parameters and variables can be summarized in *graphical model* [7] notation as:

$$\pi_1 \rightarrow \mathcal{L} \rightarrow \mathcal{S} \leftarrow \mathcal{C} \rightarrow s' \leftarrow \ell' \leftarrow \pi_1' \tag{5}$$

where we have introduced an independent prior, π_1', for ℓ'. The importance of this diagram cannot be overstressed. It is used repeatedly below, to be able to remove irrelevant conditioning terms.[2]

Our end-goal will be to infer the value of ℓ', when $\mathcal{S}$ and s' are observed and π_1' is given, but where $\pi_1, \mathcal{L}, \mathcal{C}$ are unknown nuisance variables. The result could be given as the posterior $P(\ell'|\mathcal{S}, s', \pi_1')$, or equivalently[3], as the *predictive likelihood-ratio*:

$$R(s'|\mathcal{S}) = \frac{P(s'|\ell' = H_1, \mathcal{S})}{P(s'|\ell' = H_2, \mathcal{S})} \tag{6}$$

Our end-goal is the *calibration* of s' via the mapping:

$$s' \rightarrow \log R(s'|\mathcal{S})$$

3. INFERENCE

Here we discuss computational strategies to compute $R(s'|\mathcal{S})$. The computation involves summing over all of the hidden labels, $\mathcal{L}$, and integrating out the parameters, $\mathcal{M}$. Unfortunately this cannot be done in closed form. If conjugate priors are used, the parameters can be integrated out in closed form, but this makes the labels dependent, so that summing them out requires an intractable sum over 2^T terms. Conversely, if you start with the labels, they can be summed out in closed form, using (4), but then the parameters cannot be integrated out in closed form. For this work, we shall follow the latter route, because the parameter space is just 4-dimensional, allowing approximate integration in this space.

The numerator and denominator of (6) are obtained by marginalizing w.r.t. $\mathcal{C}$ and simplifying by (5):

$$R(s'|\mathcal{S}) = \frac{\big\langle P(s'|\mathcal{C}, H_1)\big\rangle_{\mathcal{S}}}{\big\langle P(s'|\mathcal{C}, H_2)\big\rangle_{\mathcal{S}}} \tag{7}$$

where $\langle\rangle_{\mathcal{S}}$ denotes expectation w.r.t. the parameter posterior $P(\mathcal{C}|\mathcal{S})$. We show below how to derive $P(\mathcal{C}|\mathcal{S})$ from the *Laplace approximation* for $P(\mathcal{M}|\mathcal{S})$.

Although we shall use (7) in practice, we develop an interesting alternative form below that helps to theoretically illuminate the relationship between the plug-in and predictive LRs.

3.1. Plug-in vs predictive LR

The predictive likelihoods can be expanded by the product rule as:

$$P(s'|\mathcal{S}, H_i) = P(s'|\mathcal{C}, H_i) \times \frac{P(\mathcal{C}|\mathcal{S})}{P(\mathcal{C}|\mathcal{S}, s', H_i)} \tag{8}$$

[2]Observation at a node with convergent arrows induces dependency between variables linked via this node; when not observed, such nodes block dependency. Conversely, nodes with divergent or aligned arrows induce dependency when not observed; and block dependency when observed. A node is 'observed' if it appears to the right of the | in probability notation.

[3]To see this, use (5) to find: $\frac{P(H_1|\mathcal{S}, s', \pi_1')}{P(H_2|\mathcal{S}, s', \pi_1')} = \frac{\pi_1'}{1-\pi_1'} \times R(s'|\mathcal{S})$.

where a ratio of unsupervised to semi-supervised posteriors modulates the plug-in likelihood. The numerator is conditioned only on the unsupervised scores, while the denominator is conditioned on one additional supervised score, with assumed label $\ell' = H_i$. Although (8) holds for any value of $\mathcal{C}$ with non-zero posteriors, we find a more convenient form by taking logarithms and the expectation w.r.t. $P(\mathcal{C}|\mathcal{S})$ on both sides:

$$\log P(s'|\mathcal{S}, H_i) = \big\langle \log P(s'|\mathcal{C}, H_i)\big\rangle_{\mathcal{S}} + D_i(s') \tag{9}$$

where D_i denotes KL-divergence from unsupervised to semi-supervised posterior:

$$D_i(s') = \big\langle \log \frac{P(\mathcal{C}|\mathcal{S})}{P(\mathcal{C}|\mathcal{S}, s', H_i)}\big\rangle_{\mathcal{S}} \tag{10}$$

Using (9) in (6) gives the predictive log-LR as:

$$\log R(s'|\mathcal{S}) = \big\langle \log R(s'|\mathcal{C})\big\rangle_{\mathcal{S}} + D_1(s') - D_2(s') \tag{11}$$

Notice that:

$$\big\langle \log R(s'|\mathcal{C})\big\rangle_{\mathcal{S}} = s'\big\langle \frac{d'}{\sigma}\big\rangle_{\mathcal{S}} + \big\langle \frac{\mu_2^2 - \mu_1^2}{2\sigma^2}\big\rangle_{\mathcal{S}} \tag{12}$$

which remains affine in s', just like (2). Moreover, if $P(\mathcal{C}|\mathcal{S})$ has a *sharp, dominant*[4] *peak*, then $\big\langle \log R(s'|\mathcal{C})\big\rangle_{\mathcal{S}} \approx \log R(s'|\hat{\mathcal{C}})$, where $\hat{\mathcal{C}}$ is the mode of the dominant peak. Finally, if there are many scores in $\mathcal{S}$, then a single additional score s' that is similar to the scores in $\mathcal{S}$, will result in small $D_i(s')$, so that $\log R(s'|\mathcal{S}) \approx \log R(s'|\hat{\mathcal{C}})$. Only if $\mathcal{S}$ has very few scores, or s' is very far away, will the $D_i(s')$ cause significant non-linearity in $\log R(s'|\mathcal{S})$.

We already know that we have a large collection of unsupervised scores, but it remains to be demonstrated that $P(\mathcal{C}|\mathcal{S})$ has a dominant peak, which we shall do below, via an experimental exploration of the likelihood. We shall also quantify the sharpness of that peak by using the *Laplace approximation*.

3.2. Laplace approximation

The Laplace approximation (LA) is ideally suited to approximating sharply peaked, low-dimensional posteriors [7, 8]. We have only 4 parameters and the likelihood is sharply peaked because we have lots of data. The only pitfall is that label swapping causes two identical peaks in the likelihood. We kill the unwanted peak by assigning a prior of the form $P(\mathcal{M}) \propto u(\mu_1 - \mu_2)$, where u is the unit step function. We do not need to specify the prior in any more detail, because any reasonable prior that we might want to assign here would be effectively constant relative to the sharply peaked likelihood.

Following the standard LA recipe to approximate the posterior $P(\mathcal{M}|\mathcal{S})$, we define:

$$f(\mathcal{M}) = \log P(\mathcal{S}, \mathcal{M}) = \log[P(\mathcal{S}|\mathcal{M})P(\mathcal{M})] \tag{13}$$

which is computable by (4). Notice $P(\mathcal{M}|\mathcal{S}) \propto e^{f(\mathcal{M})}$. Let $\hat{\mathcal{M}}$ be the dominant mode of f and form a 2nd-order Taylor-series approximation here. The gradient at the mode is zero, but we need the Hessian (2nd derivative matrix), denoted Λ. This forms a multivariate Gaussian, approximate posterior:

$$\tilde{P}(\mathcal{M}|\mathcal{S}) = \mathcal{N}(\mathcal{M}|\hat{\mathcal{M}}, -\Lambda^{-1}) \tag{14}$$

[4]By *dominant*, we mean that the peak contains almost all of the probability mass in $P(\mathcal{C}|\mathcal{S})$.

Since (7) calls for $P(\mathcal{C}|\mathcal{S})$, we still have to marginalize[5] (14) w.r.t. π_1. With the Gaussian approximation this is easy. Let $\mathbf{C} = -\Lambda^{-1}$ denote the covariance of $\tilde{P}(\mathcal{M}|\mathcal{S}) = \mathcal{N}(\mathcal{M}|\hat{\mathcal{M}}, \mathbf{C})$, then the corresponding marginal $\tilde{P}(\mathcal{C}|\mathcal{S})$ is also multivariate Gaussian, where the elements in $\hat{\mathcal{M}}$ and $\mathbf{C}$ corresponding to π_1 have been removed [7]. In summary, we get the 4-dimensional mode and Hessian, invert the Hessian and then discard one dimension.

3.2.1. Model parametrization

The LA is *not* invariant to parametrization [8]. Moreover, it is obvious that the true posterior for the parameters π_1 and σ^2 cannot be Gaussian. But, for a sharply peaked posterior, the parametrization is not that important and the behaviour far from the maximum is almost irrelevant. As long as $f(\mathcal{M})$ is smooth enough so that a 2nd-order approximation is accurate close to the maximum, the posterior peak will be approximately Gaussian. If the likelihood magnitudes are large, then by the time the 2nd-order approximation becomes inaccurate, this inaccuracy becomes irrelevant because of the effect of the exponentiation. The reader is encouraged to consult Wikipedia,[6] where this is graphically illustrated. Our experimental results below are reported for the parametrization $[\mu_1, \mu_2, \log(\sigma^2), \log(\pi_1)]$.

4. EXPERIMENTS

We used two score corpora, *DAC* and *ABC*:

The Domain Adaptation Challenge (DAC)[7] has telephone speech from the LDC's Switchboard and Mixer databases as well as NIST SRE'10 [9]. It has three parts:

Recognizer: Switchboard was used to train the hyperparameters of an i-vector PLDA speaker recognizer [10], which was then used to produce the scores below.

Calibration: About 7 million trials (single enrollment), from pre-SRE'10 Mixer corpora, provided the unsupervised scores, $\mathcal{S}$, with target proportion about 4% and (empirical) EER $= 2.38\%$.

Evaluation: About 400 000 trials, composed of pairs of segments from SRE'10, provided the *test* scores, s', with EER $= 5.54\%$.

The ABC corpus used a different speaker recognizer, applied to a different data set, drawn from the AGNITIO-BUT-CRIM submission to NIST SRE'12 [11, 12]. The mixture of conditions was more diverse than for DAC, having telephone and microphone speech, variable number of enrollment segments, full and truncated test segments and varied noise levels.

Recognizer: Switchboard, Fisher and pre-SRE'12 Mixer was used to train an i-vector PLDA system.

Calibration: About 42 million scores, $\mathcal{S}$, from pre-SRE'12 Mixer. The target proportion is about 0.07%, and EER $= 2.38\%$.

Evaluation: About 9 million test scores, s', from SRE'12, with EER $= 3.25\%$.

4.1. Exploration of likelihood

The success of the whole venture depends critically on the behaviour of the GMM likelihood, $P(\mathcal{S}|\mathcal{M})$, given by (4). If we are a-priori very uncertain about the proportion of targets in the unsupervised data, and also about the accuracy of the recognizer, it is not at all obvious whether there is enough information in the likelihood[8] to be able to infer calibration parameters with a useful level of accuracy. Moreover, since our inference tools (plug-in and LA) both rely on finding likelihood optima, it is important to know whether the likelihood is plagued by local optima.[9]

To learn how the likelihood behaves, we did an exhaustive experimental exploration of the parameter space, $(\mu_1, \mu_2, \sigma^2, \pi_1)$. To facilitate visual representation, we used a 2-dimensional model representation, namely $(d', \log \frac{\pi_1}{1-\pi_1})$, which we plotted against log-likelihood, where the remaining two degrees of freedom were optimized.[10] Recall from section 2.1 that d' represents *accuracy*, while π_1 represents *target proportion*. If the likelihood has a single dominant peak for these two critical parameters, then there is hope that the calibration exercise will work.

We made such plots for the calibration parts of DAC and ABC. The results are similar. In figure 1 we show the latter, which we believe is more challenging, because of the smaller target proportion. The log-likelihood is smooth as a function of d', but is bi-modal as a function of π_1—a warning that initialization for the EM algorithm is important. Although the modes look flat, and similar, in the log-likelihood plot, the normalized[11] likelihood plot of figure 2 reveals that there is just a *single, sharp, dominant peak*, the location of which is given in table 2.

4.2. Analysis of sharpness

To approximate $P(\mathcal{M}|\mathcal{S})$, we use the LA recipe of section 3.2. The mode is found with the EM-algorithm. Complex-step differentiation [14] and the Pearlmutter trick [15] are used for the Hessian. We find the *error-bars* (posterior standard deviations [13, 8]) for the parameters to be suprisingly small:

parametrization	μ_1	μ_2	$\log(\sigma^2)$	$\log(\pi_i)$
ABC error-bars	0.0226	0.0004	0.0002	0.0071
DAC error-bars	0.1183	0.0192	0.0006	0.0022

4.3. Calibration experiments

The sharpness of the parameter posterior shows there is no practical difference between the plug-in and predictive likelihood-ratios. We therefore proceed to report our final experimental results for maximum-likelihood plug-in calibration. We estimated the parameters on the calibration parts of DAC (7 million scores, 4% targets) and ABC (42 million scores, 0.07% targets). The performance of these calibrations was tested on the independent evaluation parts of those corpora, in terms of normalized Bayes error-rate, also known as normalized DCF [16]:

$$\text{normDCF}(\pi_1') = \frac{\pi_1' P_{\text{miss}}(\pi_1') + (1-\pi_1)' P_{\text{fa}}(\pi_1')}{\min(\pi_1', 1-\pi_1')} \qquad (15)$$

[5] Recall $\mathcal{M} = (\mathcal{C}, \pi_1)$. By (5), π_1 and $\mathcal{C}$ are dependent in $P(\mathcal{C}, \pi_1|\mathcal{S})$, so we cannot just ignore π_1.

[6] `en.wikipedia.org/wiki/Laplace_approximation`

[7] `www.clsp.jhu.edu/workshops/archive/ws13-summer-workshop/groups/spk-13/`.

[8] The likelihood function, $P(\mathcal{S}|\mathcal{M})$, represents *all* of the information that our chosen model can extract from $\mathcal{S}$.

[9] This is true even for more sophisticated Bayesian tools, like variational Bayes and Gibbs sampling [7].

[10] Integrating them out using LA would also be feasible, but we found this unnecessary—when d' and π_1 are fixed, there remains very little uncertainty about the scale and location. The constrained optimization was done with a bespoke EM algorithm, where the M-step had to make use of numerical optimization.

[11] Subtract the maximum over the graph and then exponentiate [13].

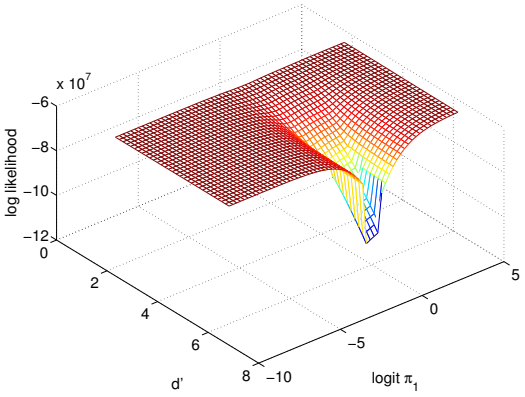

Fig. 1. $\log P(\mathcal{S}|\mathcal{M})$ for ABC

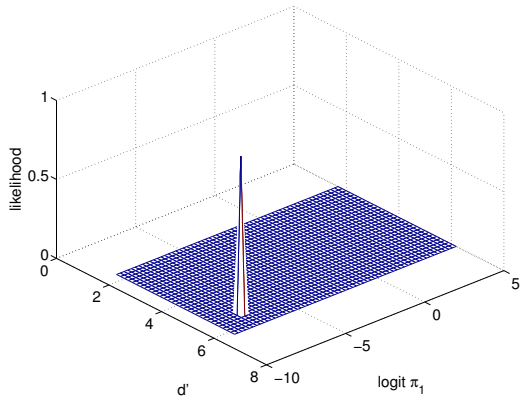

Fig. 2. normalized $P(\mathcal{S}|\mathcal{M})$ for ABC

where $P_{\text{miss}}(\pi_1')$ and $P_{\text{fa}}(\pi_1')$ are the empirical miss and false-alarm error-rates obtained when using the log-likelihood-ratios to make Bayes decisions at the theoretical threshold, $-\text{logit}(\pi_1')$. The denominator is the Bayes error-rate for the default decision that always accepts, or always rejects, depending only on π_1'. Smaller values of normDCF are better, while a value of smaller than one shows the recognizer is doing better than the default decision.

Table 1 reports normDCF for 4 different values of π_1'. For both databases, we compare the supervised recipe of [4] against the proposed unsupervised recipe. The supervised method used the *same* data as the unsupervised recipe, except that the labels were supplied. We also report *minDCF*, which uses an empirically optimized threshold at each operating point, where the optimization makes use of the evaluation labels. Finally, for DAC, we also report results for a supervised calibration trained on the *mismatched*[12], Switchboard data. The high error-rates for this case emphasizes the need for calibration on matched data.

Surprisingly, for the DAC database, the unsupervised method does mostly better than the supervised one. This may be because of errors[13] in the labels supplied to the supervised method.

In an effort to test whether our method holds up for very low target proportions, we noticed that we could go as far as removing *all* trials labelled as targets, so that $\mathcal{S}$ contained only trials labelled as non-targets. These entries in the table are marked as *unsupervised**. The fact that calibration still works in these cases can perhaps also be attributed to labelling errors.

For additional insight, tables 2 and 3 compare the estimates of model and calibration parameters for the supervised and unsupervised cases.

π_1'	0.001	0.01	0.1	0.5
ABC supervised	0.32	0.22	0.13	0.08
ABC unsupervised	0.33	0.24	0.16	0.11
ABC unsupervised*	0.32	0.23	0.15	0.10
ABC minDCF	0.31	0.21	0.12	0.06
DAC mismatched	0.73	0.54	0.35	0.21
DAC supervised	0.63	0.44	0.28	0.13
DAC unsupervised	0.65	0.43	0.25	0.11
DAC unsupervised*	0.65	0.43	0.24	0.12
DAC minDCF	0.63	0.42	0.24	0.11

Table 1. Calibration performance in terms of normDCF.

	μ_1	μ_2	σ	d'	π_1
ABC super	8.2	-5.9	2.9	4.9	6.6e-4
ABC unsup	9.9	-5.9	2.9	5.5	5.6e-4
ABC unsup*	9.6	-5.9	2.9	5.4	5.1e-5
DAC super	34.0	-169.3	48.4	4.2	3.9E-2
DAC unsup	45.9	-168.7	48.8	4.4	3.4E-2
DAC unsup*	72.3	-169.3	48.0	5.0	1.4E-5

Table 2. GMM parameter estimates

	scale	offset
ABC super	1.7	-2.0
ABC unsup	1.9	-3.8
ABC unsup*	1.9	-3.5
DAC super	0.087	5.9
DAC unsup	0.090	5.5
DAC unsup*	0.105	5.1

Table 3. Calibration parameters

5. CONCLUSION

The outcome of this work held two surprises for us. The first is that unsupervised calibration works at all. The second is that the missing labels contribute surprisingly little uncertainty to the parameter estimates.

For future work on different data, we caution against blind application of the plug-in recipe. We feel that some Bayesian analysis similar to ours should also be done to illuminate the interaction between model and data.

[12]Switchboard is at least a decade older than Mixer, during which time telephony changed dramatically [17].

[13]Some Mixer subjects registered multiple times, with different PINs, thereby causing some target trials to be falsely labelled as non-targets.

6. REFERENCES

[1] Daniel Ramos-Castro, Joaquin Gonzalez-Rodriguez, Christophe Champod, Julian Fierrez-Aguilar, and Javier Ortega-Garcia, "Between-source modelling for likelihood ratio computation in forensic biometric recognition," in *Proceedings of the 5th international conference on Audio- and Video-Based Biometric Person Authentication*, Berlin, Heidelberg, 2005, AVBPA'05, pp. 1080–1089, Springer-Verlag.

[2] 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.

[3] Niko Brümmer, Lukáš Burget, Jan "Honza" Lukáš, 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.

[4] David van Leeuwen and Niko Brümmer, "The distribution of calibrated likelihood ratios," in *Interspeech*, 2013.

[5] Niko Brümmer and George Doddington, "Likelihood-ratio calibration using prior-weighted proper scoring rules," in *Interspeech*, 2013.

[6] D.M. Green and J.A. Swets, *Signal Detection Theory and Psychophysics*, New York: Wiley, 1966.

[7] Christopher M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer, 2007.

[8] David J. C. MacKay, *Information Theory, Inference, and Learning Algorithms)*, Cambridge University Press, 2003.

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

[10] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Interspeech*, Florence, Italy, August 2011, pp. 249–252.

[11] AGNITIO, BUT, and CRIM, "ABC SRE12 presentation," in *NIST SRE 2012 Workshop, Orlando*, 2012.

[12] The National Institute of Standards and Technology, "The NIST year 2012 speaker recognition evaluation plan," `www.nist.gov/itl/iad/mig/upload/NIST_SRE12_evalplan-v17-r1.pdf`, 2012.

[13] D. S. Sivia, *Data Analysis: A Bayesian Tutorial*, Oxford University Press, 1996.

[14] Joaquim R. R. A. Martins, Peter Sturdza, and Juan J. Alonso, "The complexstep derivative approximation," *ACM Transactions on Mathematical Software*, p. 262, 2003.

[15] Barak A. Pearlmutter, "Fast exact multiplication by the Hessian," *Neural Computation*, vol. 6, pp. 147–160, 1994.

[16] Niko Brümmer and Edward de Villiers, "The BOSARIS Toolkit: Theory, algorithms and code for surviving the new DCF," in *NIST SRE'11 Analysis Workshop, Atlanta*, 2011.

[17] Christopher Cieri, Linda Corson, David Graff, and Kevin Walker, "Resources for new research directions in speaker recognition: The Mixer 3, 4 and 5 corpora," in *Interspeech*, 2007.