A comparison of linear and non-linear calibrations for speaker recognition

Niko Brümmer, Albert Swart* and David van Leeuwen+*

*AGNITIO Research, South Africa
+Radboud University, Nijmegen, The Netherlands

Abstract

In recent work on both generative and discriminative score to log-likelihood-ratio calibration, it was shown that linear transforms give good accuracy only for a limited range of operating points. Moreover, these methods required tailoring of the calibration training objective functions in order to target the desired region of best accuracy. Here, we generalize the linear recipes to non-linear ones. We experiment with a non-linear, non-parametric, discriminative PAV solution, as well as parametric, generative, maximum-likelihood solutions that use Gaussian, Student's T and normal-inverse-Gaussian score distributions. Experiments on NIST SRE'12 scores suggest that the non-linear methods provide wider ranges of optimal accuracy and can be trained without having to resort to objective function tailoring.

1. Introduction

In our recent work on score calibration for speaker recognition, we employed *linear* score-to-log-likelihood-ratio transforms, the parameters of which were trained via generative [1, 2], or discriminative [3] methods. In both cases, we noticed that a linear transform could not calibrate well everywhere over a wide range of operating points. This meant we had to *choose* in which operating region we wanted calibration to work best, by tailoring the training objective function. In both generative and discriminative cases, this was achieved (essentially) by artificially weighting the importance of target and non-target trials in the training data. In [1], we used a weighted maximum-likelihood criterion, while in [3], a variety of different calibration-sensitive discriminative objective functions were explored.

While these strategies both resulted in good calibration in the targeted operating region, it also gave poorer calibration in other operating regions. In this paper, we explore the possibility of using more general, non-linear calibration transforms, with the hope that (i) they can give good calibration over a wider range of operating points and (ii) they can be trained without having to resort to specially tailored objective functions.

In what follows, we recapitulate our generative and discriminative linear calibration strategies, and then introduce several non-linear strategies. All of these are compared experimentally on scores from SRE'12.

2. Calibration

We briefly summarize the calibration problem and some of its solutions.

We consider a speaker recognizer that, when given speech input, outputs a raw score. The score should help to decide which of two hypotheses, known as *target* and *non-target* is true. The speech input has two parts, the enrollment speech and the test speech. The target hypothesis says the test speech is of the same speaker as the enrollment. The non-target hypothesis says the speakers are different.

In order to be able to use the recognizer to make cost-effective decisions, we can calibrate the recognizer scores, to give us *log-likelihood-ratios* (LLRs) [4]. Calibration transforms a score, s, as:

$$s \to \log \frac{P(s|H_1, B)}{P(s|H_2, B)} \tag{1}$$

where the likelihoods are conditioned on H_1, the target hypothesis, or H_2, the non-target hypothesis. The likelihoods are further conditioned on some background information, B, which may include generative or discriminative score modelling assumptions, model parameters, or data. If B includes model parameters rather than data, the calibration method is known as a *plug-in* method. If instead, B contains data rather than parameters, the method is known as *fully Bayesian*. In this paper, we shall work with plug-in methods, which perform well in situations where a large amount of training data is available, which is the case here. See [2] for an analysis of the relationship between plug-in and fully Bayesian solutions and [5] for an example where fully Bayesian calibration outperforms plug-in calibration when very little training is available.

2.1. Generative calibration

For training our parametric generative models, we shall use two flavours of maximum-likelihood (ML) criterion. Let $\mathcal{T}$ and $\mathcal{N}$ respectively denote the sets of target and non-target scores available for training. The *plain* ML criterion is:

$$\sum_{s\in\mathcal{T}} \log P(s|H_1, \lambda) + \sum_{s\in\mathcal{N}} \log P(s|H_2, \lambda) \tag{2}$$

which is to be maximized w.r.t. λ, the calibration parameters. We use λ to jointly denote the parameters of both target and non-target score distributions. In some cases, the parameters for the two distributions will be independent, so that $\lambda = (\lambda_1, \lambda_2)$, but in other cases, some of the parameters may be shared between the two distributions.

The *weighted* ML criterion is:

$$\frac{\alpha}{T} \sum_{s\in\mathcal{T}} \log P(s|H_1, \lambda) + \frac{1-\alpha}{N} \sum_{s\in\mathcal{N}} \log P(s|H_2, \lambda) \tag{3}$$

where T is number of targets, N is number of non-targets, and $0 < \alpha < 1$ is a user-supplied relative weighting for targets vs non-targets. Notice that if $\alpha = \frac{T}{T+N}$, then the two criteria are equivalent. In [1], we did linear calibration, which required weighted ML, which has the disadvantage that α has to be chosen appropriately. In this paper, we experiment with more general calibration models that can be trained with plain ML.

2.2. Discriminative calibration

For *parametric* discriminative calibration, we choose a calibration function, denoted $\ell = f(s, \lambda)$, which maps scores, s, to LLRs, ℓ. The parameters are trained by minimizing, w.r.t. λ, a criterion of the form:

$$\frac{\alpha}{T}\sum_{s\in\mathcal{T}} C\left(f(s,\lambda), H_1\right) + \frac{1-\alpha}{N}\sum_{s\in\mathcal{N}} C\left(f(s,\lambda), H_2\right) \quad (4)$$

where $C(\ell, H_i)$ is a special cost function known as a *proper scoring rule* [3]. Here α fulfils a similar function as in the weighted ML criterion. For the results reported here, we choose a linear calibration transform, so that $f(s, \lambda) = As + B$, where $\lambda = (A, B)$, while for C we use the logarithmic proper scoring rule, which is equivalent to logistic regression—see [3, 6] for more details.

For *non-parametric* discriminative calibration, the calibration function $\ell = f(s)$ does not depend on a small number of parameters. Instead, it is allowed to vary freely within the class of all *monotonic rising* functions from $\mathbb{R}$ to $\mathbb{R}$. It turns out that this class of functions is general enough that it does not matter any more which proper scoring rule is used, or what the value of α is. A calibration function can be found which simultaneously minimizes the discriminative criterion for all proper scoring rules and all values of α, see [7, 8]. Moreover, an efficient algorithm, known as *pool-adjacent-violators* (PAV),[1] exists to find the calibration function [9].

3. Evaluating goodness of calibration

Our experimental setup is the same as in [2]. We performed all our calibration experiments on scores from a single speaker recognizer (an i-vector PLDA system), which was part of the ABC submission [10] to the NIST SRE'12 speaker recognition evaluation [11].

We trained our calibration parameters on a large development set, having multiple microphone and telephone speech segments of male speakers from SRE'04, '05, '06, '08 and '10. This gave about 42 million scores, of which 0.07% were targets, for calibration training.

We tested the goodness of these calibrators on male speakers from the NIST SRE'12 extended trial set [11], where we pooled all 5 common evaluation conditions, giving about 9 million trials, of which 0.1% were targets.

For evaluation of goodness of calibration, we shall use *normalized Bayes error-rate plots* [12, 7]. In these plots, we compute the error-rate that results when Bayes decisions are made by thresholding the to-be-evaluated LLR scores at the minimum-expected cost Bayes threshold. The x-axis represents the operating point in the form of *prior log odds*, and the y-axis represents normalized Bayes-error rate:

$$y = \frac{pP_{\text{miss}}(\theta) + (1-p)P_{\text{fa}}(\theta)}{\min(p, 1-p)} \quad (5)$$

where $p = \frac{1}{1+\exp(-x)}$ is a *synthetic target prior*, while $P_{\text{miss}}(\theta)$ and $P_{\text{fa}}(\theta)$ are miss and false-alarm rates obtained when thresholding LLRs at the theoretically optimal *Bayes threshold*, $\theta = -x$. The normalization factor, $\min(p, 1-p)$ is the Bayes error-rate of a default recognizer that makes Bayes decisions based on the prior, p, alone. The y-axis can also be interpreted as the well-known *detection cost function* (DCF) metric of the NIST Speaker Recognition Evaluation (SRE) series, provided the cost coefficients are set to unity. In addition to the *actual* Bayes error-rate, y, every plot will also display the *minimum* error-rate, y', which can be obtained at the empirically optimal decision threshold at every operating point:

$$y' = \frac{\min_\theta pP_{\text{miss}}(\theta) + (1-p)P_{\text{fa}}(\theta)}{\min(p, 1-p)} \quad (6)$$

Again, y' corresponds to min-DCF of the NIST SRE series. In all plots, y' will be displayed as a dashed black line.

To keep error-rates meaningful, we respect *Doddinton's Rule of 30* [13], by choosing the range of the x-axis so that there are always at least 30 errors of each kind (misses and false-alarms), at the empirically optimal threshold—see the discussions in [12, 7].

4. Linear calibrations

We summarize and compare our previous linear calibration solutions of both generative and discriminative flavours. Both need prior-weighting to target the desired operating region.

4.1. Generative: Gaussians with shared variance

We repeat the calibration method of [1]. We let $\lambda = (\mu_1, \mu_2, v)$ and we assign Gaussian score distributions of the form:

$$P(s|H_i, \lambda) = \mathcal{N}(s|\mu_i, v) \quad (7)$$

where the μ_i are hypothesis-conditional means, but where the variance, v, is shared. This gives a linear calibration transform. To train λ, we use the weighted ML criterion (3). The maximizing parameters have simple, closed-form solutions—see [1] for details.

Figure 1 shows the calibration performance for the cases $\alpha = \frac{T}{T+N} = 0.0007$, $\alpha = \frac{1}{2}$ and $\alpha = 0.92$. It is clear that good performance can be obtained locally by adjusting α. Note the agreement between α (the training parameter) and $p = \frac{1}{1+e^{-x}}$ (the evaluation parameter). If $\alpha \ll 1$, then performance is good for $p \ll 1$, (on the negative x-axis). Conversely, if $\alpha \approx 1$, then performance is good for $p \approx 1$ (on the positive x-axis). Unfortunately, tuning for good performance in one place, causes higher error-rates elsewhere.

4.2. Discriminative: logistic regression

We repeat the linear logistic regression calibration of [3], trained with the weighted criterion (4). In figure 2 we show performance for weightings $\alpha = \frac{T}{T+N} = 0.0007$, $\alpha = \frac{1}{2}$ and $\alpha = 0.92$. The same conclusions hold as in the generative case, except that performance near $x = 0$ is better and is less sensitive to α on the positive x-axis.

The green plot in figure 5 shows the linear logistic regression calibration transform for $\alpha = 1/1000$.

5. Non-linear calibrations

Now we introduce the new work in this paper, namely several non-linear calibration strategies, none of which need objective function tailoring.

[1] PAV is also known as isotonic regression.

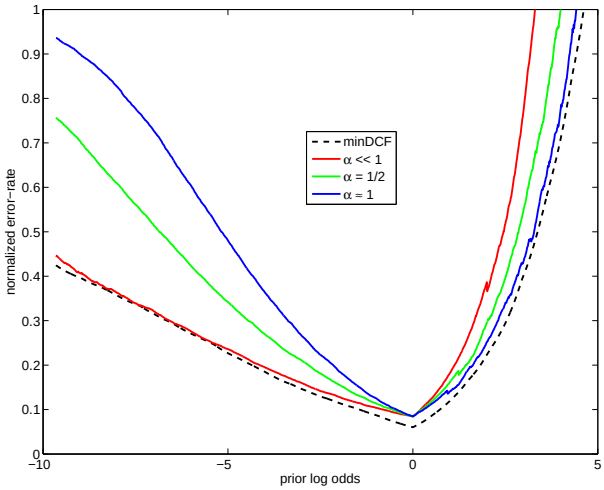

Figure 1: Accuracy of common-variance Gaussian calibration, using various values of ML weighting, α.

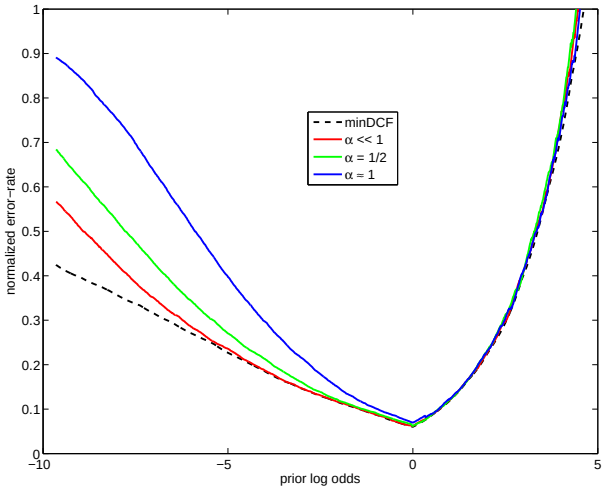

Figure 2: Accuracy of weighted logistic regression, with various values of weighting, α.

5.1. Non-linear, discriminative: PAV

The result of applying PAV calibration[2] is shown in figure 3. There is only one solution, because there are no weighting parameters to tune. Very good calibration is obtained everywhere, except in the extreme left. The PAV calibration is optimal on the training data (for which indeed DCF equals min-DCF everywhere), but on the independent evaluation data (shown) calibration can be sub-optimal. We attribute the problem in the left to overtraining in this region, where there may not be enough false-alarms in the training data, relative to the rich choice of monotonic calibration functions.

The red plot in figure 5 shows the PAV calibration transform.

[2]A MATLAB implementation is available in the BOSARIS Toolkit at `sites.google.com/site/bosaristoolkit`.

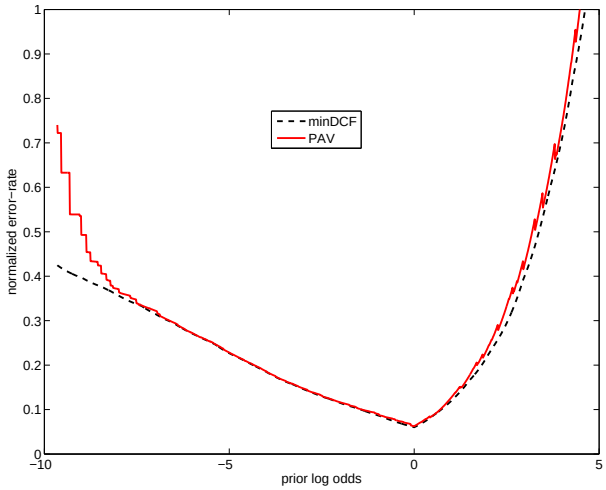

Figure 3: Accuracy of non-parametric, discriminative PAV calibration.

5.2. Non-linear, generative: Gaussian distributions

We obtain a non-linear (quadratic) calibration function by allowing separate variances for targets and non-targets, so that $\lambda = (\mu_1, v_1, \mu_2, v_2)$ and:

$$P(s|H_i, \lambda) = \mathcal{N}(s|\mu_i, v_i) \tag{8}$$

We now use the plain ML criterion (2). The ML parameters have independent, closed-form solutions. The target parameters are:

$$\mu_1 = \frac{1}{T}\sum_{s\in\mathcal{T}} s, \qquad v_1 = \frac{1}{T}\sum_{s\in\mathcal{T}} (s-\mu_1)^2 \tag{9}$$

and similar for non-targets. The accuracy is shown as the red plot in figure 4. The blue plot in figure 5 shows the (quadratic) Gaussian calibration transform.

5.3. Non-linear, generative: T-distributions

We generalize the Gaussian solution by adopting Student's T (or just T) distributions [14]. Whereas the Gaussian distribution has two parameters (location and scale), the T distribution has three: location, scale and *degrees of freedom* (d.o.f.). If the d.o.f. is large, the T distribution approaches the Gaussian. For small d.o.f., the distribution has heavy tails.

Using independent distributions for targets and non-targets, the total number of parameters for this calibration model is 6 (3 each). Closed-form solutions for the ML parameters do not exist. One way to obtain an ML solution involves designing an EM-algorithm, based on a hidden scale variable associated with every score—see [15] for a similar EM-algorithm. However, we found our EM-algorithm was slow and prone to get stuck in saddle points, or other sub-optimal areas of small gradient. Instead, we found that direct, quasi-Newton numerical optimization, in our case BFGS [16], was reliable and much faster.

Since the basic BFGS algorithm that we used is an unconstrained optimizer[3] and the scale and d.o.f. parameters are constrained to be positive, we needed to reparametrize those parameters via some suitable transform from $\mathbb{R}$ to the positive reals. We tested squaring and exponentiation. The former worked well, the latter not at all.

[3]Modified versions of BFGS exist that can handle constraints.

The accuracy is shown in the green plot in figure 4. The magenta plot in figure 5 shows the T-distribution calibration transform.

5.4. Non-linear, generative: NIG-distributions

Finally, we generalize the Gaussian even futher, to a four-parameter family known as the *normal-inverse-Gaussian* (NIG) distribution [17]. The four parameters encode location, scale, skewness and tail weight. Using independent NIG parameters for targets and non-targets, the total number of parameters is 8.

Although the ML solution may be sought via an EM-algorithm [17], we did not try it, preferring as before, direct optimization. In this case, however, we found that BFGS got stuck in saddle points. BFGS does not even know when it is in a saddle point, because it makes use of a positive-definite approximation to the Hessian.[4]

Next, we tried the more powerful *trust-region-Newton* algorithm [16, 18], which uses the true Hessian of the objective function. We computed the Hessian using the Pearlmutter trick [19] and complex-step algorithmic differentiation [20]. The first problem was that the Hessian computation took too long to perform over 42 million scores, because the NIG density requires the evaluation of Bessel functions [17]. This was solved by using (for non-targets) small (1%) randomly selected samples of the scores for the Hessian computation, but still using all the data for function value and gradient [21].

The second problem was that this algorithm still got stuck in saddle points. Simply trying to escape saddle points along the gradient did not help. What worked was to escape along the direction of the most negative curvature.[5]

The accuracy of the NIG solution is shown as the blue plot in figure 4. Of the three generative non-linear calibration models, the NIG variant performs best on this data. We would however hesitate to recommend it before testing it on several other data sets. A disadvantage of the NIG solution is that it requires working with Bessel functions, which can be tricky and slow.

The black plot in figure 5 shows the NIG calibration transform. Figure 6 shows the NIG probability densities (green) compared to histograms of the scores.

5.5. Discussion

Comparing Gaussian, T and NIG accuracies in figure 4, the main differences are at operating points on the extreme left. It is perhaps surprising that the T-distribution (3 parameters), does worse than the Gaussian (2 parameters) and the NIG (4 parameters). One would expect that the more flexible T should be able to model the data more closely than the Gaussian, while being more immune to overtraining than the NIG.

We speculate that this behaviour can be explained as follows. The Gaussian does not have the ability to accurately model the tails of the distributions and effectively ignores the tails. The T distribution has more flexible tail modelling capability, but being symmetric, it has to treat left and right tails the same. Effectively it will be compensating for skewness by making the tails thicker and this causes the observed inaccuracy. The NIG has an additional skewness parameter, so it does not have to use tail thickness to model skewness and can therefore model both tails more accurately.

It should also be mentioned that this data suffers from mild dataset shift [22] (changes in score distributions between training and evaluation) and this complicates explanations of the observed accuracy.

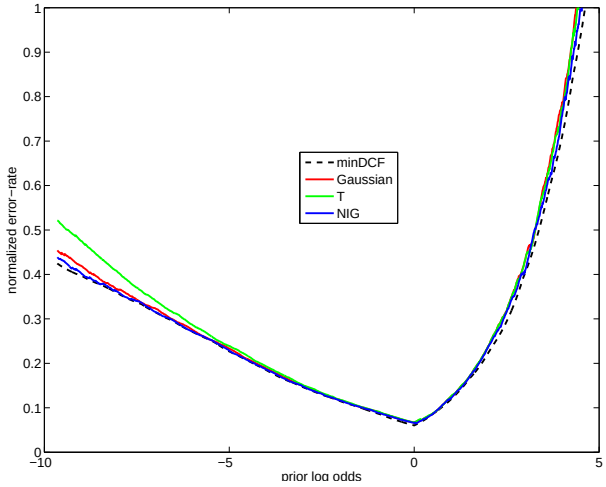

Figure 4: Accuracy of generative solutions: Gaussian, T and NIG distributions.

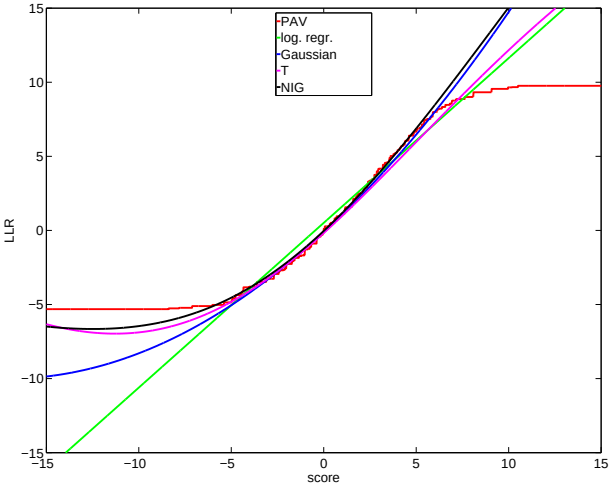

Figure 5: Comparison of score-to-LLR calibration transform functions.

6. Conclusion

We have shown that linear score-to-LLR calibration transformations struggle to give optimal accuracy over a wide range of operating points. If they are used, their training objective functions must be tailored to the desired operating region.

More flexible, non-linear calibrations can remain accurate over a wider range of operating points, while being trained with standard criteria that do not need to be tuned.

The danger remains, as always, that more flexible recognizers can be more easily overtrained. In future work, we would like to investigate fully Bayesian methods as a safeguard against

[4] In numerical optimization, it is customary to *minimize* the objective function. In this case we minimize the negative likelihood. Minima have positive-definite Hessians (matrix of 2nd-order partial derivatives), but saddle points have Hessians with eigenvalues of mixed sign.

[5] This is along the eigenvector corresponding to the most negative eigenvalue of the Hessian.

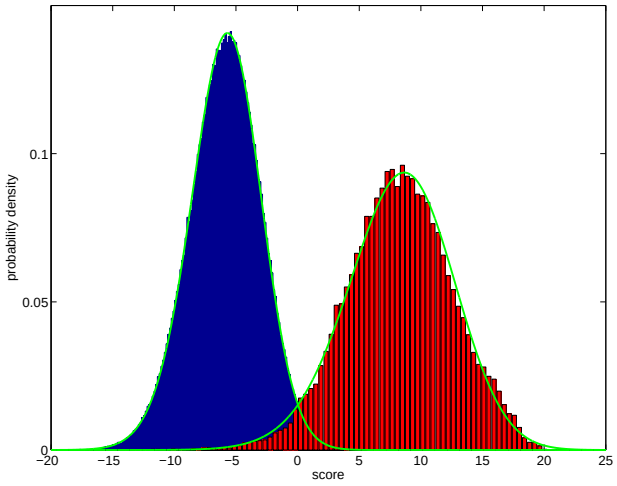

Figure 6: The maximum-likelihoood NIG solution, compared to normalized histograms of target and non-target scores.

overtraining—see [5], our first work in this direction, where we give a Bayesian solution for Gaussian score models.

We would also like to explore the richer calibration models introduced here, for the problem of unsupervised calibration [2].

7. Acknowledgements

We thank the anonymous reviewers for spotting the missing logarithms in (2) and (3) and for asking a few interesting questions, some of which we were able to answer in the final version of this paper.

8. References

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

[2] Niko Brümmer and Daniel Garcia-Romero, "Generative modelling for unsupervised score calibration," in *ICASSP*, 2014.

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

[4] 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.

[5] Niko Brümmer and Albert Swart, "Bayesian calibration for forensic evidence reporting," in *Interspeech 2014*, submitted.

[6] 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.

[7] Niko Brümmer, *Measuring, refining and calibrating speaker and language information extracted from speech*, Ph.D. thesis, Stellenbosch University, 2010.

[8] Niko Brümmer and Johan du Preez, "The PAV algorithm optimizes binary proper scoring rules," `arxiv.org/abs/1304.2331`, 2013.

[9] 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.

[10] AGNITIO, BUT, and CRIM, "ABC SRE'12 presentation," in *NIST SRE 2012 Workshop, Orlando*, 2012.

[11] 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.

[12] 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.

[13] 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.

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

[15] Patrick Kenny, "Bayesian speaker verication with heavy-tailed priors," in *Odysey 2010: The speaker and language recognition workshop*, Brno, Czech Republic, 2010, pp. 249–252.

[16] Jorge Nocedal and Stephen J. Wright, *Numerical Optimization*, Springer, 2nd edition, 2006.

[17] Dimitris Karlis, "An EM type algorithm for maximum likelihood estimation of the normalinverse Gaussian distribution," *Statistics & Probability Letters*, vol. 57, pp. 43–52, 2002.

[18] A.R. Conn, N.I.M. Gould, and P.L. Toint, *Trust-region Methods*, MPS-SIAM series on optimization. Society for Industrial and Applied Mathematics, 2000.

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

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

[21] Richard H. Byrd, Gillian M. Chin, Will Neveitt, and Jorge Nocedal, "On the use of stochastic hessian information in optimization methods for machine learning," *SIAM Journal on Optimization*, vol. 21, no. 3, pp. 977–995, 2011.

[22] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence, *Dataset Shift in Machine Learning*, MIT Press, 2009.