Automatic Regularization of Cross-entropy Cost for Speaker Recognition Fusion

Ville Hautamäki, Kong Aik Lee, David van Leeuwen, Rahim Saeidi, Anthony Larcher, Tomi Kinnunen, Taufiq Hasan, Seyed Omid Sadjadi, Gang Liu, Hynek Boril, et al.

Automatic Regularization of Cross-entropy Cost for Speaker Recognition Fusion

Ville Hautamäki[1], *Kong Aik Lee*[2], *David van Leeuwen*[3], *Rahim Saeidi*[3], *Anthony Larcher* [2], *Tomi Kinnunen*[1], *Taufiq Hasan*[4], *Seyed Omid Sadjadi*[4], *Gang Liu*[4], *Hynek Boril*[4], *John H. L. Hansen*[4], *Benoit Fauve*[5]

[1]University of Eastern Finland, Finland, [2]Institute for Infocomm Research, Singapore, [3]Radboud University Nijmegen, The Netherlands,[4] University of Texas at Dallas,USA, [5]ValidSoft Ltd, London, UK

Abstract

In this paper we study automatic regularization techniques for the fusion of automatic speaker recognition systems. Parameter regularization could dramatically reduce the fusion training time. In addition, there will not be any need for splitting the development set into different folds for cross- validation. We utilize majorization-minimization approach to automatic ridge regression learning and design a similar way to learn LASSO regularization parameter automatically. By experiments we show improvement in using automatic regularization.

1. Introduction

Score level fusion of multiple sub-systems has been the most effective way of obtaining very high accuracy in speaker verification. This trend is observed in the NIST SRE evaluations for many years now. These systems utilize multiple classifiers in order to produce the final score on whether the person speaking in the test segment is the hypothesized speaker [1]. The goal is to select a small set of base classifiers that are believed to be complementary in order to improve the discriminiation ability of the whole ensemble. To improve the generalization ability of the ensemble, regularization is a commonly used approach [2].

A regularizer imposes an additional constraint on the fusion weight optimization problem, where a regularization parameter defines the maximum norm of fusion weights [3]. Then unconstrained optimization task is turned into a constraint optimization task, where role of the constraint is to avoid fusion training to overfit on the training data. In the Lagrangian formulation, the role of the constraint is played by the well known Langrange multiplier, λ. The meaning of the λ is that the larger it is, the smaller search space is placed on the optimizer. Constraint is origin centric. So regularization is commonly called *shrinkage*, as it tends to shrink the weight coefficients towards zero.

The functional form of the regularizer also plays a role in overfit avoidance, as it defines the shape of the constraint region. The most common is the *Ridge* regularizer, which is just quadratic function of the norm (l_2), thus constraint is an origin centric hypersphere [4]. Bigger λ will then shrink all classifier weights fairly equally. *Least absolute shrinkage* (LASSO) [5] is a regularizer, where l_1 norm leads to a origin centric diamod as a constraint. LASSO is so called sparsity promoting regularizer, because if the corner of the diamond is optimum, then some of the weights will turn out to be be exactly zero.

Regularized fusion could be carried out in a two-stage process. First, a range of possible λ values are selected, then the weights are estimated for all these selected values. In this regard, a fusion training set is split into two parts for weight optimization and cross-validation estimation, respectively. If the training set is large, as was the case in NIST SRE 2012 [6], this requires a large computational effort [7].

From the Bayesian perspective, we see that regularized fusion training is actually the *maximum a posteriori* (MAP) estimation of the fusion parameters consisting of the weight vector $\mathbf{w}$ and bias b [7]. The regularizer is then the prior and the λ is the hyper-parameter of the prior distribution. An improvement to the frequentist interpretation is observed by using an "integrating out" approach to estimate the regularization parameters [7], for instance, via variational Bayes. There, the goal is to estimate the full posterior, while integrating out everything else [8]. Hyperparameters, are iteratively re-estimated, with no need for cross-validation. In [9], it was noticed that in speaker verification fusion, variational Bayes provides stable results over different ensemble sizes, but the fused score is not well calibrated as the synthetic prior cannot be easily included into the optimization cost, in contrast to cross-entropy where it is an additive term.

In case of using sparsity promoting priors, such as l_1, few Bayesian approaches have been reported in the literature, where the λ is integrated out. In [7], hierarchial Bayesian model was derived and non-informative and improper Jeffreys prior was placed in the last stage. An *expectation maximization* (EM) algorithm was specifically derived for this purpose. In [10], Jeffreys prior was placed directly to the parameter λ, resulting in a very tractable integral, but the resulting solution is less flexible than [7]. The integration by [10] results in an estimate that λ is the number of non-zero weights. In the case of [10], the hyperparameter is not estimated from the observed data.

In this work we apply the method for Bayesian ridge regression [11] to regularized classifier fusion training. In that method λ is integrated out, but still, the selected method, the majorization minimization lends to a possibility to use as any tool that finds regularized optimum of the cross-entropy cost. In addition, the method gives an estimate of the λ based on the training data. In this work we find the closed form solution to l_1 regularized optimization cost, where λ has been integrated out. We also attempt to optimize the weights using the majorization minimization scheme. Additional benefit of such a scheme is

This work was partly supported by Academy of Finland (projects 253000, 253120 and 132129) and by European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement no. 238803.

that not only is computational time reduced[1], but no training set splitting is needed and all data can be used for training.

2. Regularized cross-entropy

Given a development set of trials $\mathcal{D} = \{(\mathbf{s}_n, y_n), n = 1, 2, \ldots, N_{\text{dev}}\}$ containing N_{dev} score vectors from L base classifiers, we are interested to find a linear model $\mathbf{w}^t\mathbf{s} + b$ that minimizes errors on the development set and generalizes well to an unseen corpus. Here, $\mathbf{w} = (w_1, w_2, \ldots, w_L)^t$ are the weights to be applied to the L base classifier scores, b is the bias term added to the fused scores, and the class label y is defined to take values 0 and $+1$ for non-target and target trial, respectively.

For any class-conditional densities that follow exponential family and share the dispersion parameter [12], the logistic regression model can be written as:

$$p(y = 1|\mathbf{s}) = (1 + \exp\{-(\mathbf{w}^t\mathbf{s} + b)\})^{-1} = \sigma(\mathbf{w}^t\mathbf{s} + b).$$

In addition, the bias term b can be absorbed into $\mathbf{w}$ via a standard trick, by adding a default system that produces score 1 for each trial. Then, the likelihood of the logistic regression model can be written as:

$$p(\mathcal{D}|\mathbf{w}) = \prod_{n=1}^{N_{\text{dev}}} \left\{\sigma(\mathbf{w}^t\mathbf{s}_n)^{y_n}\sigma(-\mathbf{w}^t\mathbf{s}_n)^{1-y_n}\right\}. \quad (1)$$

Taking the negative logarithm of (1), we obtain the cost function [8]:

$$-\sum_{n=1}^{N_{\text{dev}}} \left\{y_n \ln\sigma(\mathbf{w}^t\mathbf{s}_n) + (1 - y_n)\ln\sigma(-\mathbf{w}^t\mathbf{s}_n)\right\}. \quad (2)$$

In a speaker detection task, it is common to train the fusion to be sensitive to a predefined cost function. Taking into account the Bayes risk optimal decision, the cost parameters $(C_{\text{miss}}, C_{\text{FA}}, P_{\text{tar}})$ can be summarized as an effective prior, P_{eff} [13] such that:

$$P_{\text{eff}} = \frac{P_{\text{tar}}C_{\text{miss}}}{P_{\text{tar}}C_{\text{miss}} + (1 - P_{\text{tar}})C_{\text{fa}}} \quad (3)$$

Finally, due to the fact that the posterior probability of the target trials is different from the synthetic P_{tar}, the cost function in (2) was further modified in [14] as follows:

$$\begin{aligned} E(\mathbf{w}, \mathcal{D}) &= \frac{P_{\text{eff}}}{N_t}\sum_{i=1}^{N_t}\log\left(1 + e^{-\mathbf{w}^t\mathbf{s}_i - \text{logit}\,P_{\text{eff}}}\right) \\ &+ \frac{1 - P_{\text{eff}}}{N_f}\sum_{j=1}^{N_f}\log\left(1 + e^{\mathbf{w}^t\mathbf{s}_j + \text{logit}\,P_{\text{eff}}}\right), \quad (4) \end{aligned}$$

where the summation is over the N_t target score vectors $\mathbf{s}_i$, and the N_f non-target score vectors $\mathbf{s}_j$, respectively.

Regularization can be added to the cost (4) in order to improve the generalization ability of the classifier. In the case of p-norm regularizer we will obtain

$$\min_{\mathbf{w}} E(\mathbf{w}, \mathcal{D}) \quad \text{s.t.} \quad \|\mathbf{w}\|_p^q \le t. \quad (5)$$

Then, the method of Lagrange multipliers will give us

$$\min_{\mathbf{w}} \left\{E(\mathbf{w}, \mathcal{D}) + \lambda\|\mathbf{w}\|_p^q\right\}. \quad (6)$$

In most typical settings, p and q are both set to 2 for the standard Ridge regression, and set to 1 for the LASSO [5].

[1] In practice, the optimizer is needed to run only few times, versus computing it for each λ candidate.

3. Integrating out the hyperparameters

The solution to (6) can be interpreted as an MAP estimate of [11]:

$$\mathbf{w}_{\text{MAP}} = \arg\min_{\mathbf{w}}\left\{E(\mathbf{w}, \mathcal{D}) - \log p(\mathbf{w}|\lambda)\right\} \quad (7)$$

We can simplify (7) by marginalizing λ out from the prior, i.e., $p(\mathbf{w}) = \int_\lambda p(\mathbf{w}|\lambda)p(\lambda)d\lambda$. Depending on the choice of regularizer, we now need to define the $p(\lambda)$ and compute the integral.

In [11], for the case of Ridge regression ($p = 2$), prior is defined as an isotropic Gaussian $p(\mathbf{w}|\lambda) \propto \exp\{-\frac{1}{2}\lambda\|\mathbf{w}\|_2^2\}$ and $\lambda \sim \text{Ga}(\alpha, \beta)$, a Gamma distribution. Then, solving marginalization and removing the terms independent of $\mathbf{w}$ leads to [11]:

$$\log p(\mathbf{w}) = -\left(\frac{L}{2} + \alpha\right)\log\left(\frac{1}{2}\|\mathbf{w}\|_2^2 + \beta\right) \quad (8)$$

3.1. Sparsity promoting priors

In the case of $p = 1$, a Laplacian distribution $p(\mathbf{w}|\lambda) = \frac{\lambda}{2}\exp\{-\lambda\|\mathbf{w}\|_1\}$ is an appropriate representation of LASSO regularization in the prior form. In [7], non-informative, also improper at the same time, Jeffreys hyperprior was placed on $\lambda \sim \frac{1}{\lambda}$ Then a simple EM algorithm was derived that optimizes the model parameters in the hierarchial Bayes framework. In this work, we assume $\lambda \sim \text{Ga}(\alpha, \beta)$. Similar approach was taken in the context of variational Bayes approach by [15].

We follow the example of [11] in deriving $\log p(\mathbf{w})$ in the case of Laplacian prior. We start by assuming that

$$\lambda \sim p(\lambda|\alpha, \beta) = \text{Ga}(\alpha, \beta) = \beta^\alpha\frac{1}{\Gamma(\alpha)}\lambda^{\alpha-1}\exp\{-\lambda\beta\} \quad (9)$$

and we can then write

$$\begin{aligned} p(\mathbf{w}) &= \int p(\mathbf{w}|\lambda)p(\lambda|\alpha, \beta)d\lambda \\ &= \int\frac{\lambda}{2}\exp\{-\lambda\|\mathbf{w}\|_1\}\beta^\alpha\frac{1}{\Gamma(\alpha)}\lambda^{\alpha-1}\exp\{-\lambda\beta\}d\lambda \\ &= \beta^\alpha\frac{1}{2\Gamma(\alpha)}\int\lambda^\alpha\exp\{-\lambda(\|\mathbf{w}\|_1 + \beta)\}d\lambda \\ &= \frac{\beta^\alpha\Gamma(\alpha + 1)}{2\Gamma(\alpha)}(\|\mathbf{w}\|_1 + \beta)^{-(1+\alpha)}\int I(\lambda)d\lambda, \quad (10) \end{aligned}$$

where

$$I(\lambda) = \frac{\lambda^\alpha(\|\mathbf{w}\|_1 + \beta)^{\alpha+1}}{\Gamma(\alpha + 1)}\exp\{-\lambda(\|\mathbf{w}\|_1 + \beta)\}, \quad (11)$$

which happens to be a Gamma distribution with parameters $\text{Ga}(\alpha+1, \|\mathbf{w}\|_1+\beta)$, so the last integral in (10) is 1. Finally, by taking the logarithm of the result and removing additive terms where $\mathbf{w}$ does not appear, we obtain:

$$\log p(\mathbf{w}) = -(1 + \alpha)\log(\|\mathbf{w}\|_1 + \beta). \quad (12)$$

3.2. Majorization-minimization

Using the results in (12) and (7) we arrive at the cost function to be minimized:

$$f(\mathbf{w}) = E(\mathbf{w}, \mathcal{D}) + \left(\frac{L}{2} + \alpha\right)\log\left(\frac{1}{2}\|\mathbf{w}\|^2 + \beta\right), \quad (13)$$

which is differentiable, but unfortunately not convex. The solution proposed in [11] is to use the *majorization-minimization* (MM) [16] algorithmic framework to solve the optimization task in (13).

Majorization-minimization is based on the idea that we solve the optimization problem iteratively, where in each iteration we form a surrogate cost $g(\mathbf{w}, \mathbf{w}')$, which is an upper bound of the original cost function $f(\mathbf{w}) \leq g(\mathbf{w}, \mathbf{w}')$. Minimum of $g(\mathbf{w}, \mathbf{w}')$ is found w.r.t. $\mathbf{w}$, and the new bound is constructed. Algorithm converges when minimum, whether global or local, is reached. For convergence, we require that equality is reached only when $f(\mathbf{w}') \leq g(\mathbf{w}', \mathbf{w}')$. Convergence is then assured by the so called *descent property* [16]:

$$f(\mathbf{w}') = g(\mathbf{w}', \mathbf{w}') \leq g(\mathbf{w}'', \mathbf{w}') \leq f(\mathbf{w}'') \qquad (14)$$

The reader might now notice that EM framework is actually a special case of MM optimization framework.

In [11], a surrogate function $g(\mathbf{w}, \mathbf{w}')$ is constructed in the following way. For notational convenience, we denote the terms inside the log in (8) as $h(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + \beta$. Then, taking the first order Taylor approximation around $\mathbf{w}'$ of the term, we notice that

$$\log(h(\mathbf{w})) \leq \log(h(\mathbf{w})') + \frac{h(\mathbf{w})}{h(\mathbf{w}')} - 1, \qquad (15)$$

for any $\mathbf{w}'$ and equality is obtained if and only if $\mathbf{w} = \mathbf{w}'$.

Using the result in (15), $g(\mathbf{w}, \mathbf{w}')$ can be written as [11]:

$$g(\mathbf{w}, \mathbf{w}') = E(\mathbf{w}, \mathcal{D}) + \left(\frac{L/2 + \alpha}{1/2\|\mathbf{w}'\|^2 + \beta}\right)\left(\frac{1}{2}\|\mathbf{w}\|^2\right), \qquad (16)$$

where terms independent of $\mathbf{w}$ have been removed. Obtained algorithm is presented in Algorithm 1. We notice first of all that, there is no need to manually (or by cross-validation) set the λ-parameter. But what is even more interesting is that we read an estimate of the λ directly from the $g(\mathbf{w}, \mathbf{w}')$. Using the same strategy as in (15) for a LASSO regularizer, we can do away with the $\log$ operation in $f(\mathbf{w})$.

$$f(\mathbf{w}) = -\log p(\mathcal{D}|\mathbf{w}) + (1 + \alpha)\log\left(\|\mathbf{w}\|_1 + \beta\right). \qquad (18)$$

Which will lead to $\lambda = \frac{1+\alpha}{\|\mathbf{w}^{(t)}\|_1 + \beta}$ as a new λ estimate in t^{th} iteration. However, the approach taken here requires that the bound is exact if and only if $\mathbf{w} = \mathbf{w}^{(t)}$, and this unfortunately is not the case with the l_1 norm. It will lead to non-convergent algorithm, however, in preliminary experiments we found out that after a few iterations, differences between consecutive λ's is not very large.

Algorithm 1 Majorization-minimization for fusion training

1: $t = 0$;
2: **repeat**
3:
$$\lambda^q(t) = \begin{cases} 1 & \text{if } t = 0 \\ \frac{L/2+\alpha}{\frac{1}{2}\|\mathbf{w}^{(t)}\|^2 + \beta} & \text{otherwise.} \end{cases} \qquad (17)$$
4: $\mathbf{w}^{(t+1)} = \arg\min_{\mathbf{w}} \left[E(\mathbf{w} + \frac{1}{2}\lambda^{(t)}\|\mathbf{w}\|^2\right]$
5: **until** convergence

4. Experiments

Various solutions were put in place as part of the I4U[2] submission consisting of 17 base classifiers of which 15 were used in these experiments These include the GMM-SVM [17], JFA [18], and the most advanced i-vector based classifier [19]. Briefly, the ensemble consists of 1 GMM-UBM, 3 GMM-SVMs, 1 JFA, and 13 i-vector based classifiers. In essence, all of these classifiers are based on the use of Gaussian mixture model (GMM) in a GMM-UBM framework. In other words, these classifiers require and have their own UBM for extracting the Baum-Welch statistics. To this end, UBMs are trained mainly using data drawn from SRE04, though SRE05, SRE06, Switchboard and Fisher were also used. The channel compensation components (NAP for GMM-SVM classifiers, channel factors in JFA, LDA and PLDA for i-vector) were trained using both the telephone and microphone data drawn from SRE04-SRE10. Some i-vector based classifiers use cosine scoring, while most classifiers use PLDA, such as back-ends of CRSS systems [20].

To enable the researchers in the I4U coalition to optimize speaker recognition systems performance on a condition similar to SRE'12 evaluation plan, the development sets were generated[3]. Two sets called development (DEV) and evaluation (EVAL) are made to assess the absolute performance as well as calibration performance of the systems. A disjoint set of speakers from SRE'06 are held out from our DEV and EVAL training sets and only included in test phase to simulate the *unknown non-target trials*. Number of segments, speakers and trials for each set are given in Table 1. For the SRE12 fusion experiments, we train the fusion device on DEV portion of the I4U and apply it as is to *common conditions* (CC) 1 and 3.

We experimented with the maximum likelihood optimized logistic regression (cross-entropy cost) [4], Ridge regularized cross-entropy, and LASSO regularized cross-entropy. In the case of LASSO regularization, non-differentiability of the prior is handled by the optimizer package we utilized. More details on the optimizer can be found in [2]. Regularized variants are the implementation of Section 3.2, where λ is first integrated out from the optimization cost leading to non-convex cost function. This cost function is then optimized by a local search scheme, namely majorization-minimization. The MM estimates gives as a side product the estimate of the λ. In all experiments $p_{\text{known}} = 0$ and we report only the *equal error rate* (EER) and $\min C_{\text{primary}}$. In both regularized variants, parameters of Gamma distribution were set to $\text{Ga}(0, 1)$. We also show best individual base classifier, based on the performance on the dev set, namely GMM-SVM utilizing an anti model [21].

In Table 2, we notice that majorization-minimization optimized Ridge and l_1 improves the $\min C_{\text{primary}}$over non-regularized approach and best individual classifier. However, for EER, regularized approaches do not bring improvement over non-regularized and even in the case of Female subset EER is worse than best individual. Estimated λ on Ridge and l_1 were similar for both Female and Male subsets, indicat-

[2]This is a collaboration of the following institutes: (1) Institute for Infocomm Research, Singapore, (2) University of Eastern Finland, Finland, (3) Radboud University Nijmegen (RUN), The Netherlands, (4) CRSS, University of Texas at Dallas, USA (4) ValidSoft Ltd, London, UK, (5) LIA, University of Avignon, France, (6) Idiap Research Institute, Switzerland, (7) Swansea University, UK, (8) University of New South Wales, Australia

[3]The lists are available via `http://lands.let.ru.nl/~saeidi/I4U.tgz`

[4]Similar to software package FoCal, but a different implementation

Table 1: Number of speakers, speech segments and trials in the development sets.

	Number of speakers				Number of segments				Number of trials			
	DEV		EVAL		DEV		EVAL		DEV		EVAL	
	Train	Test	Train	Test	Train	Test	Train	Test	True	False	True	False
Males	680	868	763	804	16941	19866	29961	21837	14589	13494291	15483	16646148
Females	1039	1243	1155	1102	24693	25980	43119	28548	19863	26973357	20763	32952177

Table 2: Fusion performance on the I4U evalset

Method	EER (%)	$\min C_{\text{primary}}$	Estimated λ
Female			
Best ind.	0.54	0.0667	n/a
No Regul.	**0.33**	0.0524	n/a
Ridge	0.63	0.0449	4.0749
l_1	0.63	0.0448	0.2963
Male			
Best ind.	0.68	0.0656	n/a
No Regul.	**0.40**	0.0483	n/a
Ridge	0.49	**0.0399**	4.0835
l_1	0.49	**0.0398**	0.2759

Table 3: Results on SRE12 evaluation corpus, common condition 1 (CC1) subset.

Method	EER (%)	$\min C_{\text{primary}}$
Female		
Best ind.	5.74	0.36179
No regul.	**2.67**	**0.27674**
Ridge	3.75	0.30752
l_1	3.74	0.30775
Male		
Best ind.	5.45	0.34738
No regul.	3.48	**0.20204**
Ridge	**3.38**	0.22670
l_1	3.39	0.22670

ing close similarity in the trainingset score distributions. The majorization-minimization algorithm converged for Ridge regularizer in three iterations and for l_1 in two iterations.

In Tables 3 and 4 we see equal error rate and $\min C_{\text{primary}}$ for SRE12 evaluation corpus subset CC1 and CC2. We notice that the ridge and l_1 give practically identical results for these sets. In terms of both EER and $\min C_{\text{primary}}$ we note that regularization is better only in one case out of eight, in other cases regularization does not help. This is an indication that optimizing fusion weights on I4U DEV did not lead to an overfit, which would make it necessary to regularize in order to avoid it.

5. Conclusions

In this paper, we have compared different regularization techniques for fusion of speaker verification systems when optimizing a cross-entropy function. We proposed to remove the need of setting a hyper-parameter and to apply cross-validation by integrating out the regularization parameter. Several regularization constraint have been compared. For the case of Ridge regularized logistic regression, we applied an existing marginalization technique in complement of majorization-minimization

Table 4: Results on SRE12 evaluation corpus, common condition 3 (CC3) subset.

Method	EER (%)	$\min C_{\text{primary}}$
Female		
Best ind.	3.52	**0.15913**
No regul.	**2.78**	0.22766
Ridge	2.92	0.21396
l_1	2.92	0.21396
Male		
Best ind.	4.52	0.13511
No regul.	**3.87**	**0.07187**
Ridge	4.06	0.07604
l_1	4.06	0.07604

algorithm to learn the regularization parameters. We also derived a similar method for the case of sparsity promoting prior focusing on the l_1 regularization as sparse fusion. As a future work, we plan to investigate an other bounds in the LASSO case, so that majorization-minimization algorithm converges in all cases.

6. References

[1] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: from features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, January 2010.

[2] V. Hautamäki, T. Kinnunen, F. Sedlák, K. A. Lee, B. Ma, and H. Li, "Sparse classifier fusion for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, no. 8, pp. 1622–1631, August 2013.

[3] V. Hautamäki, K. A. Lee, T. Kinnunen, B. Ma, and H. Li, "Regularized logistic regression fusion for speaker verification," in *Interspeech 2011*, Florence, Italy, August 2011, pp. 2745–2748.

[4] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning – Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2008.

[5] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[6] "The NIST year 2012 speaker recognition evaluation plan," 2012. [Online]. Available: http://www.nist.gov/itl/iad/mig/sre12.cfm

[7] M. A. T. Figueiredo, "Adaptive sparseness for supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1150–1159, September 2003.

[8] C. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer Science+Business Media, LLC, 2006.

[9] V. Hautamäki, K. A. Lee, A. Larcher, T. Kinnunen, B. Ma, and H. Li, "Variational Bayes logistic regression as regularized fusion for nist sre 2010," in *Speaker Odyssey 2012*, Singapore, June 2012.

[10] G. Cawley, N. L. Talbot, and M. Girolami, "Sparse multinomial logistic regression via Bayesian l1 regularization," in *NIPS 19*, 2007, pp. 209–216.

[11] C. Foo, C. B. Do, and A. Y. Ng, "A majorization-minimization algorithm for (multiple) hyperparamtere learning," in *ICML 2009*, 2009.

[12] M. I. Jordan, "Why the logistic function? a tutorial discussion on probabilities and neural networks," Massachusetts Institute of Technology, Cambridge, MA, Tech. Rep., August 1995.

[13] N. Brümmer and J. Preez, "Application-independent evaluation of speaker detection," *Computer Speech and Language*, vol. 20, pp. 230–275, April-July 2006.

[14] N. Brümmer, L. Burget, J. Černocký, O. Glembek, F. Grézl, M. Karafiát, D. Leeuwen, P. Matějka, P. Schwartz, and A. Strasheim, "Fusion of heterogeneous 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, September 2007.

[15] A. Kabán, "On Bayesian classification with Laplacian priors," *Pattern Recognition Letters*, vol. 28, pp. 1271–1282, 2007.

[16] K. Lange, D. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions," *Journal of Computational and Graphical Statistics*, vol. 9, no. 1, pp. 1–20, 2000.

[17] W. Campbell, D. Sturim, and D. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, May 2006.

[18] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of inter-speaker variability in speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 5, pp. 980–988, July 2008.

[19] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[20] T. Hasan, S. O. Sadjadi, G. Liu, N. Shokouhi, H. Boril, and J. H. Hansen, "CRSS systems for 2012 NIST speaker recognition evaluation," in *Proc. Int. conference on acoustics, speech, and signal processing (ICASSP 2013)*, 2013, accepted.

[21] H. Sun, K. A. Lee, and B. Ma, "Anthi-Model KL-SVM-NAP System for NIST SRE 2012 Evaluation," in *Proc. Int. conference on acoustics, speech, and signal processing (ICASSP 2013)*, 2013.