Odyssey 2014:
The Speaker and Language Recognition Workshop
16-19 June 2014, Joensuu, Finland

Multiclass Discriminative Training of i-vector Language Recognition

Alan McCree

Human Language Technology Center of Excellence
Johns Hopkins University, Baltimore, MD, USA
alan.mccree@jhu.edu

Abstract

The current state-of-the-art for acoustic language recognition is an i-vector classifier followed by a discriminatively-trained multiclass back-end. This paper presents a unified approach, where a Gaussian i-vector classifier is trained using Maximum Mutual Information (MMI) to directly optimize the multiclass calibration criterion, so that no separate back-end is needed. The system is extended to the open set task by training an additional Gaussian model. Results on the NIST LRE11 standard evaluation task confirm that high performance is maintained with this new single-stage approach.

1. Introduction

In recent years, the i-vector approach [1] has provided the best performance in NIST evaluations of both speaker and language recognition. In this method, built on techniques originally developed for subspace modeling of Gaussian Mixture Models (GMMs) using Joint Factor Analysis, the GMM for each speech cut is assumed to differ from a Universal Background Model (UBM) by only a low-dimensional offset of the mean supervector. The Maximum a Posteriori (MAP) estimate of this offset, called an i-vector, is generated for each cut and treated as an input feature for classification.

For i-vector language recognition, a range of classifier approaches have been successful, including a Gaussian model, Support Vector Machines (SVMs), Logistic Regression, and cosine scoring [2, 3, 4]. However, all these approaches require an additional multiclass back-end classifier which provides significant performance improvement as well as producing calibrated probability outputs. In addition, these methods only produce scores for the known set of training languages, i.e. the closed set, and rely on the back-end to introduce the capability of out-of-set (OOS) rejection. This paper describes a new approach to discriminative training of Gaussian i-vector classifiers, such that both state-of-the-art closed set performance and OOS scoring can be attained without the need for a separate back-end.

The paper is organized as follows. First, Section 2 presents the Gaussian model that serves as the basis for this work. Section 3 then describes discriminative training of Gaussian model parameters using MMI. Experimental results on the NIST LRE11 task are presented in Section 4, followed by concluding remarks in Section 5.

2. Additive Gaussian Noise Model

In the additive Gaussian noise model [5], an observed i-vector $\mathbf{z}_n$ from a given speech cut is assumed to have been generated by a language model $\mathbf{m}_i$ corrupted by a channel noise $\mathbf{c}_n$:

$$\mathbf{z}_n = \mathbf{m}_i + \mathbf{c}_n, \tag{1}$$

where both the language and channel are drawn from Gaussian distributions:

$$\mathbf{m}_i \sim \mathcal{N}(\mathbf{m}_0, \mathbf{\Sigma}_m); \mathbf{c}_n \sim \mathcal{N}(0, \mathbf{\Sigma}_c) \tag{2}$$

These assumptions are also the foundation for the two-covariance model [6] and Probabilistic Linear Discriminant Analysis (PLDA) [7]. Solutions for model training and scoring under various additional assumptions are given in the following sections.

2.1. Known Model

If we assume that the true language models are known, then the likelihood of each language for the test i-vector is given by

$$\mathbf{z}_n | L_i \sim \mathcal{N}(\mathbf{m}_i, \mathbf{\Sigma}_c). \tag{3}$$

Each language L_i is represented by a Gaussian model, all of which share a common covariance [3, 4].

2.1.1. Scoring

For the closed set case, the posterior probability for each language can be computed using Bayes' rule:

$$P(L_i|\mathbf{z}_n) = \frac{p(\mathbf{z}_n|L_i)P(L_i)}{\sum_j p(\mathbf{z}_n|L_j)P(L_j)} \tag{4}$$

Equivalently, the detection likelihood ratio for each class can be computed by

$$LR(L_i|\mathbf{z}_n) = \frac{p(\mathbf{z}_n|L_i)}{\sum_{j\neq i} p(\mathbf{z}_n|L_j)P(L_j|not\ i)} \tag{5}$$

For the open set case, there is the additional possibility that the test language is not in the training set (OOS). In this case, the test cut represents a random language in a random channel, and since both are independent and Gaussian, the OOS likelihood is also Gaussian:

$$\mathbf{z}_n|L_{oos} \sim \mathcal{N}(\mathbf{m}_0, \mathbf{\Sigma}_m + \mathbf{\Sigma}_c). \quad (6)$$

Posteriors are then given by:

$$P(L_i|\mathbf{z}_n) = \frac{p(\mathbf{z}_n|L_i)P(L_i)}{\sum_j p(\mathbf{z}_n|L_j)P(L_j) + p(\mathbf{z}_n|L_{oos})P(L_{oos})} \quad (7)$$

For the pure OOS situation, the likelihood ratio is easily written as:

$$LR(L_i|\mathbf{z}_n) = \frac{p(\mathbf{z}_n|L_i)}{p(\mathbf{z}_n|L_{oos})}. \quad (8)$$

2.1.2. Training

In practice, point estimates are used for the language models. Maximum likelihood estimates are given by the training data means $\bar{\mathbf{z}}_i$. Alternatively, MAP estimates adapt from the prior distribution of language models [8], resulting in:

$$\mathbf{m}_i = \mathbf{\Sigma}_m \left(\mathbf{\Sigma}_m + \frac{\mathbf{\Sigma}_c}{N}\right)^{-1} \bar{\mathbf{z}}_i + \frac{\mathbf{\Sigma}_c}{N}\left(\mathbf{\Sigma}_m + \frac{\mathbf{\Sigma}_c}{N}\right)^{-1} \mathbf{m}_0 \quad (9)$$

2.2. Unknown Model: Bayesian Method

More generally, Bayesian methods can be used to estimate posterior distributions of the language models [8]. For the additive Gaussian model, the posterior model mean is given by the MAP estimate above, and the posterior covariance is

$$\mathbf{\Sigma}_i = \mathbf{\Sigma}_m \left(\mathbf{\Sigma}_m + \frac{\mathbf{\Sigma}_c}{N}\right)^{-1} \frac{\mathbf{\Sigma}_c}{N}. \quad (10)$$

Scoring is accomplished using the predictive distribution, which again is Gaussian:

$$\mathbf{z}_n|L_j \sim \mathcal{N}(\mathbf{m}_i, \mathbf{\Sigma}_c + \mathbf{\Sigma}_i). \quad (11)$$

These equations are the ones used for Bayesian Speaker Comparison in [9]. Although they provide the same answer as the solutions for the 2-covariance model or full-rank PLDA, they differ significantly in form. The other derivations are based on directly computing the likelihood ratio between the same vs. different hypotheses, and never explicitly compute model parameter distributions. This makes it more difficult to see the similarity between point estimate and Bayesian approaches. In particular, it is well known that so long as the prior distribution does not preclude the true solution, then as the amount of training data becomes large the Bayesian posterior distribution approaches the ML solution, and scoring via Eq. 3 and Eq. 11 become equivalent.

A heuristic variation of this scoring method has been successful for speaker recognition with multiple enrollment cuts. This approach, referred to here as 1-cut Bayesian scoring, simply pretends there was only one training cut for each model, i.e. replaces N in Eq. 9 and 10 with 1.

2.3. Hyperparameter Estimation

This model has three hyperparameters: $\mathbf{m}_0$, $\mathbf{\Sigma}_c$, and $\mathbf{\Sigma}_m$. Under the known model assumption, they can be estimated using within-class and across-class covariance matrices:

$$\mathbf{\Sigma}_c = \sum_i P_i \left\{ \frac{1}{N_i} \sum_{\mathbf{z}_n \in D_i} (\mathbf{z}_n - \mathbf{m}_i)(\mathbf{z}_n - \mathbf{m}_i)^T \right\} \quad (12)$$

$$\mathbf{m}_0 = \sum_i P_i \mathbf{m}_i \quad (13)$$

$$\mathbf{\Sigma}_m = \sum_i P_i (\mathbf{m}_i - \mathbf{m}_0)(\mathbf{m}_i - \mathbf{m}_0)^T \quad (14)$$

where $\{\mathbf{m}_i\}$ are the ML class means from training data $\{D_i\}$, and $\{P_i\}$ are the class prior probabilities, typically either the training data proportions or a uniform prior.

In a Bayesian model formulation, maximum likelihood estimation of the hyperparameters can be done with an Expectation-Maximization (EM) algorithm [7]. Again, as the amount of training data per class becomes large, these two hyperparameter estimation techniques should give the same answer. Based on preliminary experiments, this work uses covariance matrix estimation with uniform class priors.

2.4. Dimension Reduction and Diagonalization

Although i-vectors are already low-dimensional as compared to the original GMM supervectors (600 vs. about 115,000), further dimension reduction is often applied. Mathematically this comes from the assumption that $\mathbf{\Sigma}_m$ is reduced rank (note that reducing the rank of $\mathbf{\Sigma}_c$ would result in poorly-defined likelihoods). The EM algorithm for PLDA can naturally handle this constraint while maximizing the likelihood. Alternatively, Linear Discriminant Analysis (LDA) reduces dimensions using a discriminative criterion which concentrates the scatter of the training set across rather than within classes [8].

To reduce computation, this work uses diagonal covariance matrices, based on the fact that two symmetric matrices can be simultaneously diagonalized with a linear transformation [10]. This process is given by:

1. perform eigendecomposition $\mathbf{\Sigma}_c = E_1 \Lambda_1 E_1^T$
2. transform $\mathbf{\Sigma}_m$ with $\mathbf{\Sigma}'_m = \Lambda_1^{-\frac{1}{2}} E_1^T \mathbf{\Sigma}_m E_1 \Lambda_1^{-\frac{1}{2}}$

3. perform eigendecomposition $\mathbf{\Sigma}'_m = E_2 \Lambda_2 E_2^T$

4. (optional) keep only principal components

5. final transform: $\mathbf{z}'_n = E_2^T \Lambda_1^{-\frac{1}{2}} E_1^T \mathbf{z}_n$

In this transformed space, $\mathbf{\Sigma}_c = I$ and $\mathbf{\Sigma}_m = \Lambda_2$. Since the error criterion for LDA is to maximize $tr(\mathbf{\Sigma}_c^{-1}\mathbf{\Sigma}_m)$, keeping only the eigenvectors corresponding to the largest eigenvalues in step 4 finds the same subspace as traditional LDA, although the linear transformation of this diagonalized LDA is not identical. In general the LDA criterion only specifies the optimal subspace, not the coordinates within it.

2.5. Model Summary

This section has reviewed multiple forms of model training and scoring equations for the additive Gaussian noise model. For the known model, closed set case, the equations result in a Gaussian classifier with a single shared channel covariance, as used successfully in [3, 4]. To extend to OOS test cases, a second covariance matrix is required to model the language space. For the more general Bayesian case commonly used in speaker recognition, the classifier is still Gaussian but boundaries between classes are no longer linear, since the covariances of the predictive distributions differ based on the number of training samples per class.

3. Discriminative Training

Most previous work on discriminative training of i-vector language recognition has focused on the one-vs-rest training paradigm, using either SVMs or logistic regression [2, 3], relying on a multiclass back-end to further improve performance and calibration for the closed set detection or identification task. However, there has been work using multiclass discriminative training [4, 11]. This work expands on the MMI approach briefly mentioned in [4]; we will contrast it with the multiclass logistic regression of [11].

The MMI algorithm is commonly used for updating GMM/HMM parameters in automatic speech recognition [12] and also language recognition. We have also used it in our multiclass language recognition back-end [4], where we used an MMI-trained Gaussian classifier to replace the commonly used sequence of ML Gaussian followed by multiclass logistic regression [3].

The MMI error criterion is equivalent to that used in logistic regression, the multiclass cross-entropy between the answer key and the posterior probabilities from the system:

$$-\sum_i \sum_{\mathbf{z}_n \in D_i} \log P(L_i|\mathbf{z}_n) \tag{15}$$

The extended Baum-Welch update equations for mean and diagonal covariance are given by:

$$\mathbf{m}_i = \frac{\mathbf{S}_i^1 + C_0 \mathbf{m}_i}{S_i^0 + C_0} \tag{16}$$

$$\mathbf{\Sigma}_i = \frac{\mathbf{S}_i^2 - 2\mathbf{m}_i \mathbf{S}_i^1 + \mathbf{m}_i^2 S_i^0 + C_0 \mathbf{\Sigma}_i}{S_i^0 + C_0} \tag{17}$$

using the statistics:

$$S_i^0 = \sum_{\mathbf{z}_n \in D_i} 1 - \sum_n P(L_i|\mathbf{z}_n) \tag{18}$$

$$\mathbf{S}_i^1 = \sum_{\mathbf{z}_n \in D_i} \mathbf{z}_n - \sum_n P(L_i|\mathbf{z}_n)\mathbf{z}_n \tag{19}$$

$$\mathbf{S}_i^2 = \sum_{\mathbf{z}_n \in D_i} \mathbf{z}_n^2 - \sum_n P(L_i|\mathbf{z}_n)\mathbf{z}_n^2 \tag{20}$$

where $C_0 = 2$ is a stepsize related constant.

In the i-vector system, we use MMI training to update the ML parameters of the additive Gaussian model. In principle the Bayesian modeling approaches could be used as input to MMI refinement, but these are better viewed as complimentary approaches for small or large training data conditions, respectively. Using multiclass closed set training, the algorithm updates the language model means $\{\mathbf{m}_i\}$ and shared diagonal covariance matrix $\mathbf{\Sigma}_c$. We have experimented with four variations of this approach: update the means only, update means and covariance, update means and scaling factor of shared covariance, and finally a two-stage process where the scaling factor is updated first and then the means. Note that the traditional literature on MMI focuses on improving discrimination performance, while back-end work strives for calibration. The goal here is to accomplish both at the same time.

In addition to closed-set MMI, the OOS model can also be refined. A straightforward way to accomplish this is to view the OOS model as a separate Gaussian with its own mean and covariance, initialized by Eq. 6. Regrouping the closed training set into a round-robin of target/non-target trials allows the discriminative update of the OOS model without disturbing the already optimized closed-set performance. A practical advantage of this approach is that it does not require separate OOS training data, which can be difficult to find for language recognition.

When compared to the multiclass logistic regression of [11], MMI has many similarities. For the closed set case, the shared-covariance Gaussian is a linear classifier, and since both approaches use the same error criterion they should give the same solution. However, overtraining is a problem for both methods, and MMI has the following desirable optimization properties:

- the diagonal covariance constraint gives fewer free parameters

- the ML parameters provide a useful regularization
- initializing with ML parameters and limiting to 10 iterations provides a particularly simple regularization

In addition, the MMI output retains the Gaussian parameter form so that scoring looks the same for generative or discriminative versions. Finally, extension to OOS makes the classifier nonlinear, which cannot be trained with logistic regression.

4. Experimental Results

To compare the performance of these various techniques for language recognition, experiments were performed on the NIST LRE11 30 second test corpus [13].

4.1. Corpora

LRE11 is the latest in a series of formal language detection evaluations performed by NIST since 1994. This task used 24 languages that were specifically chosen to maximize confusions. Evaluation audio cuts were drawn from both conversational telephone speech recordings collected specifically for NIST and segments drawn from narrowband segments identified within foreign language broadcast sources, such as the Voice of America (VOA). These experiments use training and development sets built by MIT LL from previous LREs as well as external data [4]. More specifically, audio for training and development testing was obtained from

- Telephone data from previous LREs (1996, 2003, 2005, 2007, 2009): CallFriend, CallHome, Mixer, OHSU, and OGI-22 collections.
- Narrowband segments from VOA broadcasts.
- NIST 2011 development data (Telephone and narrowband broadcast segments).
- Narrowband segments from Radio Free Asia, Radio Free Europe, and GALE broadcasts.
- Arabic corpora from LDC and Appen.

For back-end training, a development set was created using 30 second segments from previous LREs and segments extracted from longer files.

4.2. Metrics

In previous LREs, the metrics used were C_{avg} and minimum C_{avg}, as defined in [14]. These represent a prior-balanced Bayes decision cost using a target prior of 0.5. LRE11 introduced a new metric, average pair detection cost (APD), which measured only the ability of the system to make pairwise decisions between language classes. Results presented in [4] for both of these metrics

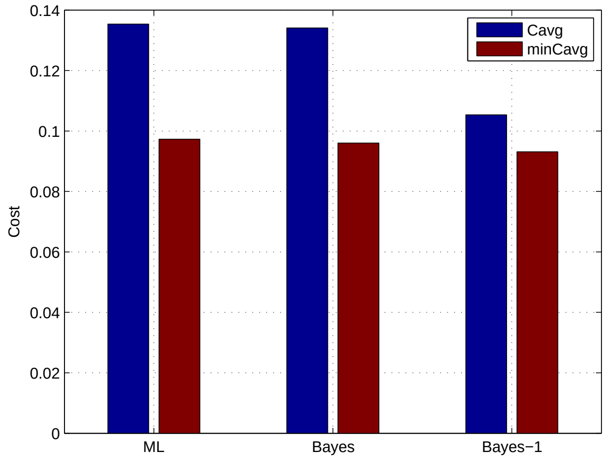

Figure 1: C_{avg} and minimum C_{avg} for closed set scoring without discriminative training. Systems are *ML*: ML Gaussian, *Bayes*: Bayesian scoring, and *Bayes-1*: 1-cut Bayesian scoring.

on LRE11 showed that they are highly related. APD is less general, since it is only measures pairwise decisions and does not require the ability to distinguish a language from all other languages. Therefore, the results reported below use C_{avg}.

There is some controversy within the language recognition community on the meaning of calibration and detection in a closed set language task. This stems from the fact that Bayes' rule is used, and calibration affects the interaction between the terms in the denominator sum of Eq. 5. Therefore, in contrast to the typical open-set scoring in speaker recognition evaluations, here calibration changes not just actual but also minimum cost. For discussion of results in this paper, we make the following statements:

- Actual detection cost C_{avg} is a reasonable measure for a calibrated system.
- A well-calibrated system will have actual C_{avg} very close to minimum C_{avg}.
- Calibration will improve actual C_{avg} and may also improve minimum C_{avg}.

4.3. Feature Processing and i-vector Extraction

The acoustic front-end for this work is similar to that in [4]. Feature processing uses 24 Mel Frequency Cepstral Coefficients (MFCC) from 0-4 kHz, windows of 20 ms length with 10 ms shift, vocal tract length normalization (VTLN) trained with four iterations of speaker adaptive training, RASTA filtering, conversion to MFCCs, shifted delta cepstra (SDC) coefficients with 7-1-3-7

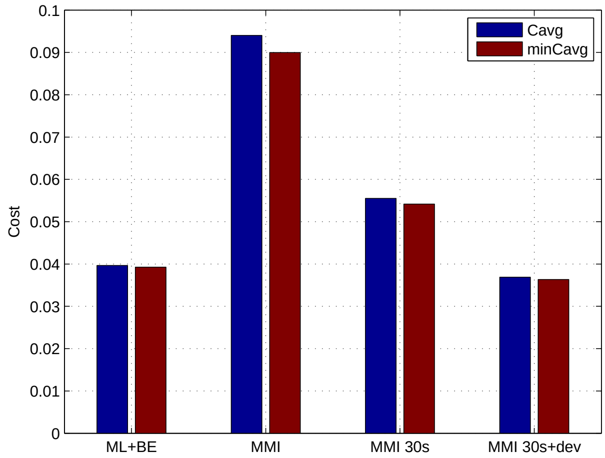

Figure 2: C_{avg} and minimum C_{avg} for closed set scoring with discriminative training. Systems are *ML+BE*: ML with back-end, *MMI*: MMI trained, *MMI 30s*: MMI trained with training files truncated at 30s, and *MMI 30s+dev*: MMI trained using 30s train + development set.

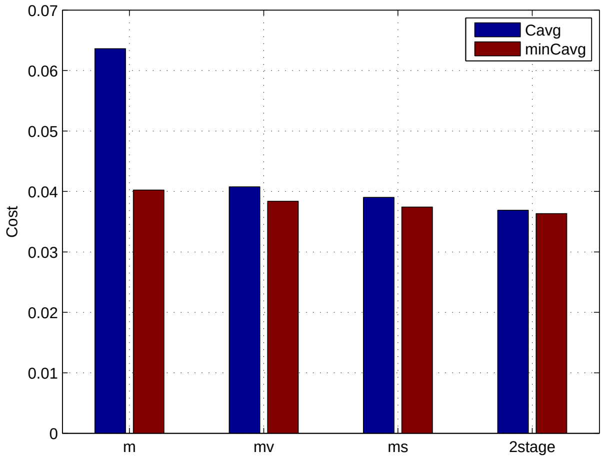

Figure 3: C_{avg} and minimum C_{avg} for different versions of MMI training. Systems are *m*: mean only, *mv*: mean and variance, *ms*: mean and covariance scaling, and *2stage*: scaling first then mean.

configuration, static cepstra appended to produce a 56-dimensional feature vector, gating with a GMM-based speech activity detector, and feature vector mean and variance normalization with a 3 second sliding window. The resulting feature sequence is then aligned to a 2048 mixture GMM trained on the entire training set, and a 600-dimensional i-vector is estimated using an i-vector extractor trained on the same set. Finally, whitening and length-normalization [15] are applied, followed by diagonalized LDA dimension reduction to 23 dimensions.

4.4. Results

Fig. 1 shows the actual and minimum detection costs for the LRE11 30 second test set using closed set scoring with Bayes' rule. The baseline maximum likelihood Gaussian system minimum cost of almost 0.1 is quite poor without a back-end, and the actual cost is even higher due to a lack of calibration. As expected, the Bayesian version of this system (equivalent to the two-covariance model or PLDA) gives almost exactly the same performance, as the amount of training data per class is enough to justify ML modeling. The MAP system (not shown) also performs the same. Finally, the 1-cut version of Bayesian scoring, while mathematically incorrect, does provide some improvement in actual cost. This is because it (accidentally) gives a larger covariance in the predictive distribution, which the MMI algorithm also finds since it improves the classification performance. Perhaps this is also the reason for its success in speaker recognition. Regardless, after the use of discriminative training, either in a separate back-end or for the Gaussian parameters directly, the advantage of this heuristic approach disappears, so the remainder of this work uses the ML system.

Fig. 2 shows the performance with discriminative training. After the multiclass discriminatively-trained back-end, both minimum and actual cost of the ML system are greatly improved to less than 0.04, competitive with the best acoustic system result presented in [4]. That i-vector system got significant performance gains with improved features using a combination of VTLN and feature-domain channel compensation (FNAP); this result implies that of the two it is VTLN that is most important. Discriminative training of the Gaussian system provides better performance than ML without a back-end, as shown in the second result of Fig. 2, but is not nearly as good as the state-of-the-art. However, using better training data can in fact close this gap. The next presented system improves by simply truncated all training set audio files to 30 seconds to match the properties of the test set. Finally, combining both 30 sec training and development sets together for a discriminatively-trained system provides the best performance of all for both minimum and actual cost without any use of a separate back-end classifier.

Fig. 3 presents the performance of different versions of MMI training. Updating the mean only provides very good minimum cost, but the actual cost is quite high implying poor calibration. MMI training of the covariance as well provides much better calibration. Training the covariance scale factor instead of the entire diagonal matrix gives a little better results, and the best performance is obtained with the two-stage approach of scaling fol-

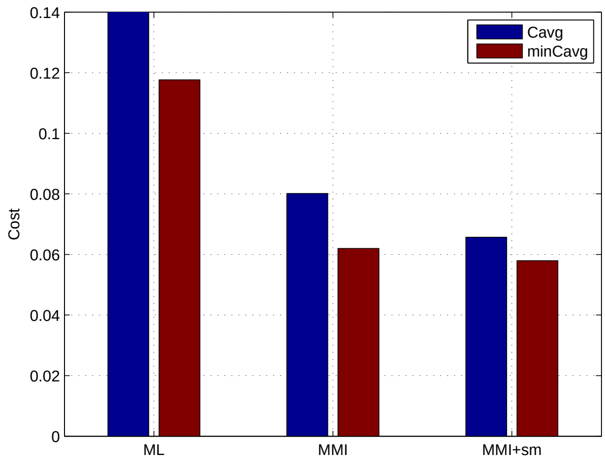

Figure 4: C_{avg} and minimum C_{avg} for open set scoring. Systems are *ML*: ML Gaussian baseline, *MMI*: MMI trained (closed set), *MMI+sm*: additional MMI training of OOS scale factor and mean. Note that C_{avg} for ML is off this scale (0.27).

lowed by mean updates. Not included in the plot are untied covariance updates where each class is allowed a different covariance; this system performs even worse than the mean-only version.

Fig. 4 shows the same results using open-set scoring, where Bayes' rule is not used and the non-target hypothesis comes only from the single global Gaussian representing OOS. In this case, the ML system provides poor performance. The closed-set trained MMI system already works better for this task but is still not well-calibrated as evidenced by the significant gap between minimum and actual cost. Further open-set discriminative training of the OOS model improves the actual cost significantly. Even though the LRE11 corpus does not support a true open set test, this does show the capability of training an OOS model using only in-set data.

5. Conclusion

This paper has presented a unified approach to i-vector language recognition where a Gaussian classifier is trained using MMI to directly optimize multiclass calibration and no separate back-end is needed. Results on the NIST LRE11 standard evaluation task confirm that high performance and calibration are maintained with this new single-stage approach. In addition, the system is extended to the open set task using the additive Gaussian noise model, and this is also discriminatively trained to improve performance. While the LRE11 paradigm does not allow true testing of OOS, results do show this gives significant improvement when not using closed-set information in the scoring process.

6. Acknowledgements

The author gratefully acknowledges interesting discussions related to this work with Daniel Garcia-Romero, Doug Reynolds, Pedro Torres-Carrasquillo, and Niko Brümmer.

7. References

[1] N. Dehak, P. Kenny, R. Dehak, P. Ouellet, and P. Dumouchel, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, pp. 788–798, May 2011.

[2] N. Dehak, P. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via ivectors and dimensionality reduction," in *Proc. Interspeech*, 2011, pp. 857–860.

[3] D. Martinez, O. Plchot, L. Burget, O. Glembek, and P. Matejka, "Language recognition in ivectors space," in *Proc. Interspeech*, 2011, pp. 861–864.

[4] E. Singer, P. Torres-Carrasquillo, D. Reynolds, A. McCree, F. Richardson, N. Dehak, and D. Sturim, "The MITLL NIST LRE 2011 language recognition system," *Proc. Odyssey*, pp. 209–215, 2012.

[5] A. McCree, D. Sturim, and D. Reynolds, "A new perspective on GMM subspace compensation based on PPCA and Wiener filtering," in *Proc. Interspeech*, 2011, pp. 145–148.

[6] Niko Brümmer and Edward De Villiers, "The speaker partitioning problem," in *Proc. Odyssey*, 2010.

[7] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. ICCV*, 2007, pp. 1–8.

[8] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley, 2001.

[9] B. J. Borgstrom and A. McCree, "Discriminatively trained Bayesian speaker comparison of i-vectors," in *Proc. ICASSP*, 2013.

[10] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, 1990.

[11] N. Brümmer, S. Cumani, O. Glembek, M. Karafiat, P. Matejka, J. Pesan, O. Pichot, M. Soufifar, E. De Villiers, and J. Cernocky, "Description and analysis of the Brno276 system for LRE2011," in *Proc. Odyssey*, 2012.

[12] Daniel Povey, "Discriminative training for large vocabulary speech recognition," *Ph.D. thesis, Cambridge University*, 2004.

[13] "The NIST year 2011 language recognition evaluation plan," `http://www.nist.gov/itl/iad/mig/upload/LRE11_EvalPlan_releasev1.pdf`, 2011.

[14] "The NIST year 2009 language recognition evaluation plan," `http://www.itl.nist.gov/iad/mig/tests/lang/2009/LRE09_EvalPlan_v6.pdf`, 2009.

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