Discriminative Acoustic Language Recognition via Channel-Compensated GMM Statistics

Niko Brümmer[1], *Albert Strasheim*[1],
Valiantsina Hubeika[2], *Pavel Matějka*[2], *Lukáš Burget*[2] *and Ondřej Glembek*[2]

[1]AGNITIO, South Africa, {nbrummer|astrasheim}@agnitio.es
[2]Speech@FIT, Brno University of Technology, Czech Republic

Abstract

We propose a novel design for acoustic feature-based automatic spoken language recognizers. Our design is inspired by recent advances in text-independent speaker recognition, where intra-class variability is modeled by factor analysis in Gaussian mixture model (GMM) space. We use approximations to GMM-likelihoods which allow variable-length data sequences to be represented as statistics of fixed size. Our experiments on NIST LRE'07 show that variability-compensation of these statistics can reduce error-rates by a factor of three. Finally, we show that further improvements are possible with discriminative logistic regression training.

Index Terms: acoustic language recognition, intersession variability compensation, discriminative training

1. Introduction

Spoken language recognition is the problem of automatically recognizing the language spoken in a given speech segment. In recent literature, e.g. [1], language recognition methods have been classified as either *phonotactic* or *acoustic*. Phonotactic language recognizers make explicit use of phoneme recognizers, similar to those used in LVCSR (large vocabulary continuous speech recognition). In contrast, acoustic language recognizers directly model short-term frequency analyses, such as MFCC or PLP features [2]. An advantage of the acoustic method is that it avoids the complexities and resources needed to train and run phoneme recognizers. In this paper we explore novel ways of constructing acoustic language recognizers. These recognizers form part of our 'BUT-AGNITIO' submission to the 2009 NIST Language Recognition Evaluation[1].

We build our approach on the existing recipe of transforming sequences of MFCC features to sequences of *shifted delta cepstra* (SDC) and then modeling the SDC sequences with language-dependent GMMs (Gaussian mixture models) [3]. Several variants on this basic technique have been published, some involving discriminative training and/or channel-compensation, see e.g. [1, 2, 4, 5, 6] and references therein.

The novel aspect of our proposal is that we apply a technique which has recently led to progress in GMM-based text-independent automatic speaker recognition. This technique replaces all computationally expensive evaluations of GMM log-likelihoods by evaluation of a simplified lower-bound. This simplification allows replacement of variable-length input feature sequences with *sufficient statistics* of fixed size [7, 8, 9]. We use these statistics for all further computation, in both training and test. The advantages include (i) efficient implementation of speaker-recognition-style channel compensation and (ii) the possibility to perform discriminative training of language models with multiclass logistic regression.

[1]http://www.itl.nist.gov/iad/mig/tests/lre/2009

Below we discuss feature extraction and computation of sufficient statistics. Next we present the factor-analysis model, which leads to the channel compensation procedure. Finally, we show how to use these ideas to create a number of different language recognition systems, which we then exercise on the NIST 2007 Language Recognition Evaluation [10].

2. Acoustic Features and UBM

This section is a brief summary of acoustic feature extraction and UBM training. For more detail, see our previous work [4, 11]. The inputs to the language recognizer are segments of recorded telephone speech of varying duration. Every speech segment is mapped to a variable-length sequence of feature vectors as follows: After discarding silent portions, every 10ms speech-frame is mapped to a 56-dimensional feature vector. The feature vector is the concatenation of an SDC-7-1-3-7 vector and 7 MFCC coefficients (including C0). We shall refer to the feature vector of frame i of segment s as $\vec{\phi}_{si}$.

A 2048-component, language-independent, maximum-likelihood GMM was trained with the EM-algorithm on the pooled acoustic feature vectors of all development data of all available languages. We follow speaker recognition terminology and refer to this language-independent GMM as the *universal background model*, or UBM [12]. We shall refer respectively to the mean-vector and (diagonal) precision matrix of Gaussian component k of the UBM as $\vec{\mu}_k$ and $\mathbf{P}_k$.

3. Sufficient statistics

This section describes how variable-length input sequences of feature vectors are mapped to sufficient statistics of fixed size. This mapping is parametrized by the UBM. All input sequences, for both training and test purposes, are mapped to sufficient statistics and all further processing is based only on the statistics, rather than the original feature sequences. Let $P_{ksi} = P(k|\vec{\phi}_{si})$ denote the posterior probability of UBM component k, given feature vector $\vec{\phi}_{si}$, computed with the standard recipe for GMM observations, assuming frame-independence. For segment s, with frames indexed $i = 1, 2, \ldots, N_s$, we define the zero- and first-order statistics respectively as:

$$n_{sk} = \sum_{i=1}^{N_s} P_{ksi} \tag{1}$$

$$\mathbf{f}_{sk} = \sum_{i=1}^{N_s} P_{ksi} \mathbf{P}_k^{\frac{1}{2}} (\vec{\phi}_{si} - \vec{\mu}_k) \tag{2}$$

where $k = 1, \cdots, 2048$. For later convenience, we stack the first-order vectors for all components into a single *supervector*, denoted $\mathbf{f}_s = [\mathbf{f}'_{s1} \cdots \mathbf{f}'_{s2048}]'$. (A supervector is just a very large vector, in this case of size 114688).

In contrast to the original recipe in [7], we do not use second-order statistics, because they cancel when forming GMM log likelihood ratios. We also center and reduce our statistics relative to the UBM, so that we can henceforth regard the UBM as having zero mean and unity precision for all components. This simplifies the formulas below for working with the statistics, because after this transformation we do not need to refer to the UBM parameters again.

4. Modeling language and channel

In both speaker and language recognition, the term *channel variability* has become a synecdoche, which refers to more general within-class variability. In speaker recognition a more accurate term is *intersession variability*. In language recognition, it is understood to mean variability between speech segments of a given language, when different things are said in that language, by different speakers, in different ways, over different channels and generally under different circumstances on different occasions.

Following [7], we defer the responsibility of understanding these complex phenomena to a statistical model. Although conceptually simple, the model is powerful because it has a few million parameters. The model has a two-level hierarchy: First, we assume there is a different Gaussian mixture model (GMM) that generates every observed speech segment. Second, we assume a meta-model that generates the GMM for every segment. These GMMs have segment-and-language-dependent component means, but fixed component weights and precisions, chosen to be equal to the UBM weights and precisions. In other words, we parametrize *segment GMMs* by their means only. Specifically, we use a factor-analysis model for the kth component mean of the GMM for segment s:

$$\mathbf{m}_{sk} = \mathbf{t}_{\ell(s)k} + \mathbf{U}_k \mathbf{x}_s \tag{3}$$

where $\ell(s)$ denotes the language of segment s; the $\mathbf{t}_{\ell k}$ are language *location* vectors; $\mathbf{x}_s$ is a vector of C segment-dependent *'channel' factors*; and $\mathbf{U}_k$ is a 56-by-C *factor loading matrix*. The channel factors are assumed to be drawn independently from the standard normal distribution.

As in the case of the first-order statistics, we stack component-dependent vectors into supervectors $\mathbf{m}_s$ and $\mathbf{t}_\ell$ and we stack the component-dependent $\mathbf{U}_k$ matrices into a single tall matrix $\mathbf{U}$, so that (3) can be expressed more compactly as:

$$\mathbf{m}_s = \mathbf{t}_{\ell(s)} + \mathbf{U}\mathbf{x}_s \tag{4}$$

We refer to $\mathbf{U}$ as the *channel matrix*. Finally, if there are K different languages, then we let $\mathbf{T} = [\mathbf{t}_1 \mathbf{t}_2 \cdots \mathbf{t}_K]$. Our meta-model for language-and-segment-dependent GMMs is now parametrized by $(\mathbf{T}, \mathbf{U})$, where $\mathbf{T}$ represents the locations of languages in GMM space and $\mathbf{U}$ represents within-language variability. The training of the language recognizer is therefore the problem of using development data to assign values to $(\mathbf{T}, \mathbf{U})$.

4.1. Estimating the channel matrix

We train the channel matrix $\mathbf{U}$ with maximum likelihood, by using the EM-algorithm of [7]. We use all speech segments for all of the languages that we have available in our development data. We start by assigning tentative values to the language location vectors $\mathbf{t}_{\ell k}$, by using a single iteration of relevance-MAP adaptation from the UBM in the manner of [12]. This adaptation can be expressed succinctly in terms of our statistics as:

$$\mathbf{t}_{\ell k} = \frac{\sum_s \mathbf{f}_{sk}}{r + \sum_s n_{sk}} \tag{5}$$

where the sums are over all segments s belonging to language ℓ. In our experiments we used a relevance factor of $r = 2$. With the language locations held fixed, the EM-algorithm iteratively re-estimates $\mathbf{x}_s$ for every segment s and then $\mathbf{U}_k$ for every component k, over all of the data. The EM-algorithm maximizes a lower-bound to the log-likelihood of the model (3), over all of the training data. As noted above, the lower bound allows us to represent all of this data by their sufficient statistics. This means that the EM-algorithm needs to iterate only over all segments, rather than over all frames of all segments. We tried different sizes for $\mathbf{U}$ and found $C = 50$ to be a good choice.

4.2. Channel compensation

Given the channel matrix $\mathbf{U}$ and the statistics $\mathbf{f}_{sk}, n_{sk}$ for a segment s, we can perform a language-independent maximum-a-posteriori (MAP) point-estimate of the channel factors $\mathbf{x}_s$, relative to the UBM. This estimate is computed as:

$$\hat{\mathbf{x}}_s = \left(\mathbf{I} + \sum_k n_{sk} \mathbf{U}'_k \mathbf{U}_k \right)^{-1} \mathbf{U}'\mathbf{f}_s \tag{6}$$

Next, the effect of the channel factors can be approximately removed from the first-order statistic thus:

$$\tilde{\mathbf{f}}_{sk} = \mathbf{f}_{sk} - n_{sk} \mathbf{U}_k \hat{\mathbf{x}}_s \tag{7}$$

We refer to $\tilde{\mathbf{f}}_{sk}$ (or the stacked supervector $\tilde{\mathbf{f}}_s$) as the *compensated first-order statistic*. In our experiments reported below, we try both uncompensated and compensated statistics for every system variant. In all cases we find the compensation to dramatically improve accuracy.

5. System descriptions

Here we describe three variants of language recognition systems, all based on sufficient statistics. All systems use the same estimate of $\mathbf{U}$ as explained above, but they differ in the way $\mathbf{T}$ is estimated:

5.1. Baseline

Our baseline system used no channel compensation and no discriminative training. We used uncompensated segment statistics to make relevance-MAP estimates of the language locations and uncompensated statistics to score new test segments. Specifically, we used (5) for the language locations, and as explained in section 4, we pack the K location supervectors for each of the K languages into the columns of a matrix denoted $\mathbf{T}$.

We score new test segments with a fast approximate linear scoring technique for GMMs, that we have recently proposed for speaker recognition [9]. This scoring technique is also based on sufficient statistics, but further simplifies the scoring in [7] by omitting non-linear terms. Given the first-order statistic $\mathbf{f}_s$ for a test segment s and the language location matrix $\mathbf{T}$ (with K columns for each of K languages), we generate a vector of K language-dependent scores thus:

$$\vec{\lambda}_s = \mathbf{T}'\mathbf{f}_s \tag{8}$$

5.1.1. Channel-compensated baseline

Next, we add channel compensation to the baseline, but still without discriminative training. This system is the same as above, except that we use channel-compensated first-order statistics everywhere. For the language locations, we reuse (5), but now with compensation:

$$\tilde{\mathbf{t}}_{\ell k} = \frac{\sum_s \tilde{\mathbf{f}}_{sk}}{r + \sum_s n_{sk}} \tag{9}$$

Again we pack the location supervectors into the columns of a matrix denoted $\tilde{\mathbf{T}}$ and we score thus:

$$\vec{\lambda}_s = \tilde{\mathbf{T}}'\tilde{\mathbf{f}}_s \tag{10}$$

5.2. Discriminative training of locations

Next, we note that the linear scoring formula (10) is a clear invitation to try discriminative training of the language locations via *multiclass logistic regression* (MCLR), where the class log-likelihoods are formed analogously to $\vec{\lambda}_s$, by multiplying a discriminatively optimized coefficient matrix with the data vector. For a general introduction to logistic regression, see e.g. [13], and for our previous application of logistic regression to language recognition see [14]. The complication in the present case is that the logistic regression is of challenging scale—the number of training examples is of order 10^4 and the input supervector size is of order 10^5. We solved this by implementing an algorithm for large-scale unconstrained convex optimization, known as *trust-region Newton conjugate gradient*, and which uses both first- and second-order partial derivatives of the objective function [15, 16]. For this system our scores are expressed as:

$$\vec{\lambda}_s = \breve{\mathbf{T}}'\tilde{\mathbf{f}}_s \tag{11}$$

where the matrix $\breve{\mathbf{T}}$, the columns of which can be interpreted as discriminatively trained language locations, is optimized via logistic regression.

5.3. Discriminative recognition via segment GMMs

There is a variant of the above logistic regression formula, where the input for every segment to the discriminative recognizer is a *segment-GMM estimate*, rather than a first-order statistic. Using relevance-MAP adaptation and compensated statistics, the mean estimate of component k of the GMM for segment s is:

$$\tilde{\mathbf{g}}_{sk} = \frac{\tilde{\mathbf{f}}_{sk}}{r + n_{sk}} \tag{12}$$

The stacked GMM supervectors $\tilde{\mathbf{g}}_s$ can then be used as train/test inputs for logistic regression. In this case we denote the scores:

$$\vec{\lambda}_s = \mathbf{W}'\tilde{\mathbf{g}}_s \tag{13}$$

where $\mathbf{W}$ is optimized via logistic regression. We note that (except for the particulars of channel compensation) this method is very similar to previous work where the training is performed by linear SVMs (support vector machines) rather than logistic regression, see e.g. [1, 2].

5.4. Score normalization

In some of the above systems, we found *frame-count normalization* to be helpful. That is, we formed normalized scores as:

$$\vec{\lambda}'_s = \frac{\vec{\lambda}_s}{\sum_{k=1}^{2048} n_{ks}} \tag{14}$$

In the case of logistic regression systems, we pre-normalized the first-order statistics, so that the discriminative training took account of the normalization. In our experimental results below, we shall indicate which systems were thus normalized.

6. Evaluation criteria

The question of how to best judge the accuracy of language recognizers has been answered in the literature in different ways. A straight-forward solution is multiclass misclassification error-rate. However, this solution is lacking in two respects: (i) It does not account for variation in the costs and priors associated with application of the recognizer, and (ii) it does not allow for analysis of performance in terms of discrimination and calibration. Presumably in response to these needs, several authors reporting on the series of NIST Language Recognition Evaluations have adopted the solution of *pooling language detection scores* over multiple targets and then analyzing the pooled scores with tools borrowed from speaker recognition, such as EER and DET-curves. Unfortunately this practice is in our opinion theoretically unfounded—and indeed our experience shows that it can give misleading results. Briefly, the problem is that the speaker recognition tools allow for a *single* score threshold to decide between two competing hypotheses, but language recognition (even when formulated as a one-against-the-rest detection problem) remains a multiclass problem which cannot be analyzed in terms of a single threshold. For further discussion see [17, 18].

The solution which we use to report on our experiments below is based on [17]. It is designed with two purposes in mind: (i) To facilitate comparison with others, we want it to be as close as possible to the language detection error-rate, C_{avg}, as used in the NIST Language Recognition Evaluations [10, 19]. (ii) However, since we are busy with basic recognizer development and we want to judge the discrimination rather than the calibration of our algorithms, we prefer not to use the calibration-sensitive C_{avg} as is. Our solution is to discount the effect of calibration by letting the *evaluator* calibrate every system. That is, the evaluator optimizes calibration on the evaluation data[2] and then reports the value of C_{avg} obtained with this calibration. We denote this measure by C^*_{avg}.

The evaluator's calibration transformation involves only scaling and translation of the score-vector, so that it does not alter the ability of the scores to discriminate between classes. In particular, the calibration transformation is *invertible*, so it does not alter the information content of the scores.

In summary C^*_{avg} measures discrimination, not calibration. It is therefore similar in spirit to the EER (equal-error-rate) and

[2]Our MATLAB code to perform this optimization is freely available at http://niko.brummer.googlepages.com/focalmulticlass.

'minimum detection cost function' of speaker recognition, but it avoids the above-mentioned problems of score pooling.

7. Experimental results

We evaluated the three system variants, as described in section 5, on the 14 languages of the closed-set language detection task of the NIST 2007 Language Recognition Evaluation (LRE'07), with input segments of nominal duration 30 seconds [10]. Our development data, which was used to train all system parameters was the same data we used in preparation for LRE'07 and does not overlap with the LRE'07 evaluation data, see [4]. We used C^*_{avg} as evaluation criterion, as explained above.

The results are shown in Table 1: Column 1 refers to the sub-section in the text describing the system. In column 2, MAP denotes *relevance-MAP adaptation* and MCLR denotes *multiclass logistic regression*. Column 3 refers to the kind of supervector that was used as input to each system: $\mathbf{f}$ denotes *first-order statistic* and $\mathbf{g}$ denotes *segment-GMM supervector*. Column 4 refers to frame-count normalization. The last two columns give C^*_{avg}, expressed as a % error-rate: Column 5 shows results *without* any channel compensation, while column 6 is *with* compensation.

For all three systems, channel compensation gives dramatic reduction in error-rate. A further improvement is achieved by the last variant of logistic regression training. It is however interesting to note that good results may be achieved by a moderately simple system design without any discriminative training.

Table 1: *Results on LRE'07, closed-set, 30s.*

section	train	input	norm	C^*_{avg} raw	C^*_{avg} comp
5.1	MAP	$\mathbf{f}$	no	11.32	1.74
5.2	MCLR	$\mathbf{f}$	yes	9.5	3.09
5.3	MCLR	$\mathbf{g}$	yes	4.56	**1.55**

8. Conclusion

We have demonstrated by experiments on NIST LRE'07, that the GMM factor-analysis modeling of [7] for speaker recognition, as implemented by using sufficient statistics, can also be used to build accurate acoustic language recognizers.

9. Acknowledgements

The authors would like to thank our colleagues in the *Robust Speaker Recognition* team at the JHU 2008 Summer Workshop, who have contributed to bringing the techniques in this paper to fruition.

BUT researchers were partly supported by European projects MOBIO (FP7-214324) and AMIDA (FP6-033812), by Grant Agency of Czech Republic project No. 102/08/0707, by Czech Ministry of Education project No. MSM0021630528 and by Czech Ministry of Interior project No. VD20072010B16.

10. References

[1] W.M. Campbell, "A covariance kernel for SVM language recognition", in Proc. ICASSP 2008. pp.:4141-4144.

[2] D. van Leeuwen and N. Brümmer, "Building language detectors using small amounts of training data", in Proc. Odyssey 2008: The Speaker and Language Recognition Workshop, Stellenbosch, January 2008.

[3] P.A. Torres-Carrasquillo, E. Singer, M.A. Kohler, R.J. Greene, D.A. Reynolds, and J.R. Deller Jr., Approaches to language identification using Gaussian mixture models and shifted delta cepstral features, in Proc. ICSLP 2002, pp.8992.

[4] V. Hubeika, L. Burget, P. Matejka, P. Schwarz, "Discriminative Training and Channel Compensation for Acoustic Language Recognition", in Proc. Interspeech 2008, Brisbane, AU, ISCA, 2008, p. 4, ISSN 1990-9772.

[5] F. Castaldo, D. Colibro, F. Dalmasso, P. Laface, C. Vair, "Compensation of Nuisance Factors for Speaker and Language Recognition",IEEE Trans. AUDIO, SPEECH, AND LANGUAGE PROCESSING Vol.15-7, 2007, pp.1969-1978 .

[6] F. Castaldo, F. Dalmasso, P. Laface, D. Colibro, C. Vair, "Politecnico di Torino System for the 2007 NIST Language Recognition Evaluation", in Interspeech 2008, pp.227-230.

[7] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition", IEEE Transactions on Audio, Speech and Language Processing 15 (4), pp. 1435-1447, May 2007.

[8] P. Kenny, N. Dehak, P. Ouellet, V. Gupta, and P. Dumouchel, "Development of the Primary CRIM System for the NIST 2008 Speaker Recognition Evaluation", in Proc. Interspeech 2008, Brisbane, Australia, Sept 2008.

[9] O. Glembek, L. Burget, N. Dehak, N. Brummer and P. Kenny, "Comparison of Scoring Methods used in Speaker Recognition with Joint Factor Analysis" in Proc. ICASSP 2009, Taipei, Taiwan, April 2009.

[10] "The 2007 NIST Language Recognition Evaluation Plan (LRE07)", Online, http://www.itl.nist.gov/iad/mig/tests/lre/2007.

[11] P. Matějka, L. Burget, P. Schwarz, and J. Černocký, "Brno University of Technology system for NIST 2005 Language recognition evaluation," in *IEEE Odyssey: The Speaker and Language Recognition Workshop*, San Juan, Puerto Rico, June 2006, pp. 57–64.

[12] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1–3, pp. 19–41, 2000.

[13] C.M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007.

[14] D. van Leeuwen and N. Brümmer, "Channel-dependent GMM and Multi-class Logistic Regression", in Proc. IEEE Odyssey 2006: The Speaker and Language Recognition Workshop, San Juan, June 2006.

[15] J. Nocedal, and S.J. Wright, *Numerical Optimization*. Springer, 2006.

[16] C.J. Lin, R.C. Weng, and S.S. Keerthi, "Trust Region Newton Method for Logistic Regression". J. Mach. Learn. Res, 9:627-650, 2008.

[17] N. Brümmer and D. van Leeuwen, "On calibration of language recognition scores", in Proc. IEEE Odyssey 2006: The Speaker and Language Recognition Workshop, San Juan, June 2006.

[18] D. van Leeuwen and K.P. Truong, "An open-set detection evaluation methodology applied to language and emotion recognition". in Proc. Interspeech, pages 338341, Antwerp, August 2007.

[19] "The 2009 NIST Language Recognition Evaluation Plan (LRE09)", Online, http://www.itl.nist.gov/iad/mig/tests/lre/2009.