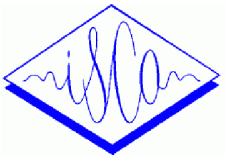
ISCA Archive
http://www.isca-speech.org/archive



Machine Learning in Speech and Language Processing (MLSLP)
Portland, OR, USA
September 14, 2012

# ITERATIVE CLASSIFICATION OF REGIONAL BRITISH ACCENTS IN I-VECTOR SPACE

*Andrea DeMarco, Stephen J. Cox*

School of Computing Sciences, University of East Anglia, Norwich, England
a.de-marco@uea.ac.uk, s.j.cox@uea.ac.uk

## ABSTRACT

Joint-Factor Analysis (JFA) and I-vectors have been shown to be effective for speaker verification and language identification. Channel factor adaptation has also been used for language and accent identification. In this paper, we show how these techniques can be used successfully in the task of accent classification, and we achieve good accuracy on a 14 accent problem using a novel iterative classification framework based on an iterative linear/quadratic classifier. These results compare favourably with recent results obtained using other non-fused acoustic techniques.

***Index Terms***— Ivector, accent classification, discriminant analysis, confidence measure

## 1. INTRODUCTION

Recently, it has been demonstrated that the use of *I-vectors* has increased the accuracy of speaker verification and language identification tasks [1, 2, 3, 4]. In this paper, we extend the use of these techniques to the problem of accent classification. After conversion of our utterances to I-vectors, we use an iterative classifier based on either linear discriminant (LDA) classification or quadratic discriminant analysis (QDA) to successively eliminate unlikely classes, and achieve results that are comparable to recently published results.

Our model of voice production output $O$ can be roughly described as the combination of three factors - the utterance 'plan', $P$, the speaker characteristics $S$ and the speaker's accent $A$, as shown in Equation 1.

$$O = P \rightarrow S \rightarrow A \tag{1}$$

To a large extent, accents are acoustic substitutions in the production of equivalent utterance plans. Differences in accent are realised at a phonetic, rather than at a phonological level [5]. Therefore, if we hypothetically assume that a given speaker could perfectly reproduce the speaking style of $n$ different accents (with no change in speaker), the same utterance plan $P$ spoken by the same speaker $S$ in $n$ accents would result in the outputs $O_1 \dots O_n$. The acoustic differences in these different outputs would pinpoint the effect that different accents have on the different productions. In this paper, we utilize and evaluate the I-Vector approach (followed by channel compensation) to provide speaker and channel normalisation, i.e. to suppress the effects of $S$ in our model. We ignore the term $P$ to construct a text-independent, unsupervised classification system for accents, our term $A$. Our goal is to evaluate whether the I-Vector subspace is comparable to other methods for accent classification, and to introduce an iterative method of classification.

The classification of accents of British English has been studied in depth by Hanani et. al. [6]. They produced several systems with different acoustic classifiers: a GMM-UBM system, a GMM-SVM system, GMM-ngram systems, and a fusion of these methods. In the GMM-UBM approach, an accent-independent Universal Background Model (UBM) is constructed from the training data from each of the classes. MAP adaptation of means and weights is then performed on the UBM using class-specific data, to generate accent-dependent Gaussian Mixture Models (GMMs), one for each accent. These GMMs can then be used to evaluate the likelihood of test utterances as belonging to a particular accent. The GMM with the highest likelihood gives the accent classification.

In the GMM-SVM method, speech data from individual speakers in the training set is used to estimate parameters of a GMM, by MAP adaptation of the UBM, based only on speaker-specific data. The adapted GMM mean vectors are concatenated into supervectors, and supervectors of each accent class are used to train Support Vector Machines (SVMs). One SVM per accent is trained to classify supervectors of one accent class (positive examples) against supervectors of all other accent classes (negative examples).

In the GMM-ngram method, the UBM is used as an acoustic tokenizer to generate GMM sequence indexes from sequences of feature vectors. These sequences are used to train unigram and bigram language models for each accent using SVMs. Essentially, this is an unsupervised language model. Crucial to the classification methods described, was inter-channel and inter-speaker variability compensation applied to the feature vectors. In their work, Hanani et. al. [6] apply the method outlined in Vair et. al. [7]. The idea of speaker and channel variability compensation is to normalize differences between speakers of the same accent class, as well as differences in the channel. The compensation method by Vair et. al. [7] employs compensation in the feature domain rather than the model domain. For this reason, it allows the construction of any classifier built on top of compensated features. The fusion of all the above methods together yielded an accuracy of 74% on 30 second cuts.

## 2. CORPUS DESCRIPTION

The Accents of the British Isles (ABI) corpus [8] was used for this investigation. This corpus comprises fourteen different accent groups, with ten speakers per gender per accent. Speakers were divided into three sets. Each set includes both male and female speakers, and speakers were balanced equally across all sets. In total, there are two sets of 98 speakers, and one set of 84 speakers. In each experiment, the training set consists of two of these groups, whilst the testing set consists of the remaining group. No speaker in any one group is present in any other group. This ensures that each test is speaker-independent. Groups are transposed three times, so that all speakers are eventually tested. Training data is gathered from all utterances available in the training set, whilst testing is performed on the basis of three long passages of 30-45 seconds each for each speaker. Results over this corpus are gathered by pooling the three test trials together.

## 3. SYSTEM DESCRIPTION

### 3.1. Feature extraction

Feature extraction stage is performed in a number of stages:

- Perform voice activity detection on the speech utterance based on the algorithm of Sohn et. al. [9]. Only segments with voice activity are retained.
- Extract 13-dimensional MFCC vectors on the speech utterance, with a window of 30ms and a frame rate of 15ms.
- Convert each MFCC vector into a 49-dimensional shifted delta cepstral (SDC) vector using a 7-1-3-7 SDC parameterization [10].
- Warp original MFCC feature vectors to a standard normal distribution with a 3 second time window to minimize effects of channel mismatch [11].
- Concatenate the warped MFCC feature vectors with their respective SDC vectors, to form a final set of 62-dimensional feature vectors.

### 3.2. Universal background model

We construct a universal background model (UBM) by first obtaining a VQ codebook via the Linde-Buzo-Gray (LBG) algorithm [12]. The codebook splitting criteria we used was to double the number of centroids at every LBG iteration, and then re-estimate the centroid means via traditional k-means algorithm until the desired number of centroids is reached. Once the cluster centroids (the VQ codes) are estimated, the covariances and weights of each cluster are estimated. This initial estimation is then passed on to a GMM trainer to perform five iterations of Expectation-Maximization, which outputs the final UBM.

### 3.3. Accent total variability

Total variability and I-vector methods were introduced first in the area of speaker verification [3, 1]. This method followed from the success of joint factor analysis (JFA). In speaker recognition, factor analysis is used to construct a low-dimensional space, called the total variability space. This space contains both speaker and channel variability, which is modelled in separate spaces in JFA. Intersession compensation can then be applied in a low-dimensional space. In the task of accent classification, there is one important difference: the total variability space $T$ is estimated differently compared to eigenvoice space estimation in speaker verification. In eigenvoice estimation, all utterances of a particular speaker are considered to belong to the same person. In total variability modelling for accent classification, however, we consider every utterance of a particular accent as having been produced by a different accent class.

The premise of a representation of the data in total variability space is that the UBM (trained on data across multiple accents) can be adapted to a given utterance, creating an utterance-dependent GMM. The eigenvoice adaptation technique assumes that the matrix $T$ contains speaker and channel variability information. The utterance GMM supervector (concatenation of mean vectors of adapted GMM) is obtained as shown in Equation 2. In this equation, $m$ is the UBM supervector, $T$ is the total variability space matrix, and $w$ is the I-vector, which is a random vector with a normal distribution $\mathcal{N}(0, I)$.

$$M = m + Tw \tag{2}$$

The I-vector $w$ is obtained for any given utterance. The method is outlined in full in [1].

### 3.4. Dimensionality reduction via LDA

Following the work done in language identification, we perform dimensionality reduction on I-vectors based on Linear Discriminant Analysis (LDA). The idea of LDA is to project I-vectors into a new subspace of reduced dimensionality that aims to maximize the ratio of between-class variance to the within-class variance, thus optimizing linear separability. Being a linear projection, it is defined everywhere in the ambient space. The projection matrix $A$ obtained through LDA on the training set is therefore applicable to the I-vectors in the testing set. The classes in our case are the different accents. LDA reduces dimensionality to one fewer than the number of classes. Therefore, for 14 accents, LDA will reduce dimensionality to 13.

### 3.5. Classification method

The low dimensional I-vectors themselves can be used for further processing and arbitrary classifiers. The compactness of having each utterance represented by a single I-vector provides for very efficient classification. We propose a novel classification framework based around two generative classification methods: one based on Fisher's linear discriminant analysis (LDA), and another based on quadratic discriminant analysis (QDA) [13]. Classification of a test utterance proceeds as follows:

1. Obtain initial LDA classifier $L^*$ and QDA classifier $Q^*$ using all accent classes. $L \leftarrow L^*$ , $Q \leftarrow Q^*$
2. Classify test utterance using $L$ and $Q$ and rank classes in order of likelihood. Identify lowest ranking class and remove it from training data.

**if** a single class remains in the training data **then**
  Classify utterance (see below)
**else**
  Re-train LDA/QDA classifiers using reduced training data
  $L \leftarrow L^*$ , $Q \leftarrow Q^*$
  Goto 2 with new classifiers $L$ and $Q$
**end if**

Traditional LDA/QDA classification would produce a result after the first scoring, by selecting the class with the highest likelihood. However, by applying the above iterative algorithm, we remove at an early stage classes that are likely to be incorrect, and hence strengthen the accumulation of evidence for classes that appear to be good contenders for the correct class. Because each iteration of the algorithm removes a class, the vector dimensionality reduces by one on each iteration, so that the algorithm is very fast.

The motivation for this iterative approach is that traditional acoustic methods have not performed as well in accent identification as in language identification, probably because of a larger acoustic overlap between classes in accent identification. The proposed algorithm attempts to iteratively sharpen the separation between classes by removing the weakest candidates at each iteration: these classes contribute mainly noise to the classification process. The rank of each class and the order in which classes are removed is recorded for the test utterance. Two possible ways in which this information could be used are:

- Classification method 1: the last class to be eliminated is the classification result, or
- Classification method 2: the class that had the best (top rank) likelihood for most iterations is the classification result

**Table 1**. Classification accuracy for all classifier types under all tested conditions, grouped by the number of factors used in I-Vector extraction.

| UBM Size | # of Factors | LDA Method 1 | QDA Method 1 | LDA Method 2 | QDA Method 2 | LDA Non-Iterative | QDA Non-Iterative | Linear SVM | RBF SVM |
|---|---|---|---|---|---|---|---|---|---|
| 256 | 50 | 40% | 39% | 48% | 48% | 46% | 40% | 44% | 46% |
| 512 | 50 | 45% | 38% | 51% | 50% | 51% | 43% | 46% | 47% |
| 1024 | 50 | 44% | 41% | 51% | 48% | 49% | 43% | 48% | 46% |
| 256 | 100 | 51% | 50% | 59% | 57% | 58% | 48% | 54% | 56% |
| 512 | 100 | 48% | 47% | 58% | 60% | 57% | 51% | 56% | 56% |
| 1024 | 100 | 49% | 48% | 58% | 57% | 59% | 58% | 57% | 58% |
| 256 | 200 | 49% | 48% | 62% | 63% | 64% | 60% | 60% | 61% |
| 512 | 200 | 53% | 46% | 64% | 63% | 63% | 60% | 61% | 62% |
| 1024 | 200 | 46% | 37% | 58% | 58% | 62% | 59% | 59% | 57% |
| 256 | 300 | 41% | 42% | **68%** | 66% | 65% | 66% | 63% | 62% |
| 512 | 300 | 37% | 43% | 63% | 63% | 64% | 63% | 64% | 61% |
| 1024 | 300 | 41% | 43% | 61% | 64% | 63% | 62% | 64% | 64% |

**Table 2**. This table shows three examples (rows two, three and four) of the iterative classification procedure working. The target class is given in the first column, and columns 3 through 16 show the ranked position of the target class (upper number) and the identity of the class removed (lower three letter identifier) at each iteration. A discussion of these examples is given in section 4.

| Scenario | Iteration | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Target = ULS | Rank of target | 4 | 3 | 3 | 2 | 4 | 3 | 3 | 1 | 3 | 1 | 1 | 1 | 1 | 1 |
| | Class removed | shl | ean | lan | crn | ilo | eyk | lvp | ncl | nwa | sse | gla | brm | roi | uls |
| Target = ULS | Rank of target | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | - |
| | Class removed | shl | ilo | nwa | ean | eyk | lan | crn | lvp | ncl | gla | brm | sse | uls | roi |
| Target = CRN | Rank of target | 6 | 6 | 6 | 5 | 5 | 5 | 4 | 5 | 4 | 4 | 3 | 3 | - | - |
| | Class removed | ean | shl | uls | gla | roi | lan | eyk | brm | ncl | lvp | sse | crn | nwa | ilo |

## 4. RESULTS

Results are based on two testing conditions: one is the number of components in the UBM, the other is the number of factors used for I-Vector extraction. Three UBM sizes are tested: 256, 512 and 1024. Four factor sizes are tested: 50, 100, 200 and 300. For each combination of these testing conditions, we report classification accuracy for the iterative algorithm introduced in this paper, as well as results for non-iterative counterparts of LDA/QDA, linear SVM and radial basis function (RBF) kernel SVM techniques (as used in e.g. [4, 14]). The results are summarized in Table 1. They show that iterative classification method 2 is consistently better than classification method 1. Also, the general trend for both LDA/QDA based classification is that classification performance improves mainly with an increase in the number of factors in the I-vector extraction system. The order of the GMM to construct the UBM was not so influential. LDA and QDA seem to give very similar performance.

Table 3 shows the individual classifier performance for each individual accent for the best iterative LDA classifier. It is interesting to note that the classifier seems to perform badly mostly on southern accents, whilst the better performing accents are mostly in northern/central areas of the British Isles. Ferragne & Pellegrino [15] made extensive studies on the ABI-1 corpus, and pointed out that the Inner London accent shows extreme heterogeneity, and should not really be considered as a single accent, which may account for the poor performance of the classifier on this accent. There is some evidence from the pattern of confusions that accents from regions that are geographically close tend to be confused: for instance, the Newcastle accent was mainly confused with Liverpool and Lancashire, and standard Southern English was confused mostly with Birmingham and Northern Wales. Cornwall, the worst performing accent, was confused across the board with most other accents except for East Anglia and Ulster. Analysis of I-vectors in low-dimensionality revealed that poorly performing accents seemed to be ones that overlap each other or a large set of other accents in this space, making them hard to have discriminative I-vector based acoustic characteristics from just acoustic information.

**Table 3**. This table shows accent classification performance on the best classifier developed in this paper.

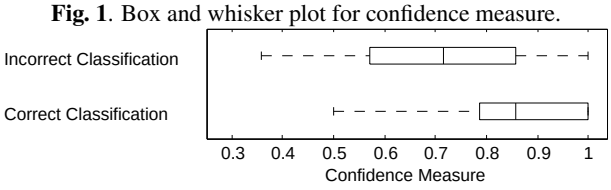
| Accent | Accuracy |
|---|---|
| Scottish Highlands (shl) | 95.24% |
| Glasgow (gla) | 94.44% |
| East Yorkshire (eyk) | 91.67% |
| East Anglia (ean) | 88.89% |
| Liverpool (lvp) | 88.89% |
| Republic Of Ireland (roi) | 83.33% |
| Ulster (uls) | 77.78% |
| North Wales (nwa) | 71.43% |
| Birmingham (brm) | 61.11% |
| Lancashire (lan) | 52.38% |
| Newcastle (ncl) | 38.89% |
| Inner London (ilo) | 38.10% |
| Standard Southern English (sse) | 33.33% |
| Cornwall (crn) | 22.22% |

Three examples of the classification processes are shown in Table 2. The first example (row two) shows a case where the target rank gradually climbs to one as incorrect classes are removed by the LDA/QDA classifier during the iterative procedure. The second example (row three) is a case where the final classification would be

incorrect under the first classification technique, but is correct using the second. The third example (row four) is a case where classification is incorrect under both classification methods. However, we can see that the iterative procedure has still managed to increase the target class rank during the iterations.

Table 4 compares the classification accuracies obtained in this work (methods 6–11) with those found in the work by Hanani et. al. [6] (methods 1–5) for 30 second test utterances. The best classification method proposed in this paper performs better (three cases) or on a par (one case) with all the unfused methods developed in [6], but not as well as a fusion-based classifier. We posit that including our method in the fusion process can achieve even better results for a fused classification system.

**Fig. 1**. Box and whisker plot for confidence measure.




We can construct a simple confidence measure using output from the iterative classification algorithm which should be useful in any techniques that integrate decisions from different classifiers, as was done in [6]. For a given test utterance, let $N_C$ be the number of times the *correct* class was top-ranked at each iteration, which is a maximum of 14 (the number of classes), and a minimum of zero. Figure 1 shows the distribution of a confidence measure, $CM = N_C/14$, for correctly classified utterances and incorrectly classified utterances. Although there is some overlap between the two distributions, it is clear that higher values of $CM$ are correlated with correct classification, and this encourages us that $CM$ will be useful in fusion with other classifiers.

**Table 4**. Comparison of classification accuracy of the results obtained in Hanani et. al. [6] (methods 1–5) with the results obtained in this work (methods 6–11) for 30 second test utterances.

| Method | Classifier Type | Accuracy |
|---|---|---|
| 1 | GMM-UBM(4096) | 56% |
| 2 | GMM-SVM (4096) | 68% |
| 3 | GMM-uni-gram | 60% |
| 4 | GMM-bi-gram | 52% |
| 5 | Acoustic fused #1 to #4 | 74% |
| **6** | **Iterative LDA via I-Vectors** | **68%** |
| 7 | Iterative QDA via I-Vectors | 66% |
| 8 | Non-Iterative LDA via I-Vectors | 65% |
| 9 | Non-Iterative QDA via I-Vectors | 66% |
| 10 | Linear SVM | 64% |
| 11 | RBF Kernel SVM | 64% |

## 5. CONCLUSIONS AND FUTURE WORK

The ABI corpus consists of fourteen accents of British English spoken by native speakers, which is a considerably more difficult classification task than, e.g. foreign accent identification. We have introduced a new, iterative, discriminative algorithm, and have shown that performance using this algorithm is better than using a standard SVM technique, and comparable to the GMM-SVM technique presented in [6]. The algorithm is low footprint and fast in both training and testing. Future work will include investigating fusion of the output from the iterative algorithm with other classifier outputs to improve overall accent classification and evaluating performance on shorter test utterances. We shall also investigate unsupervised phonotactic ngram models over specific prosodic contexts.

## 6. REFERENCES


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

[2] Najim Dehak, Pedro A. Torres-Carrasquillo, Douglas A. Reynolds, and Réda Dehak, "Language recognition via i-vectors and dimensionality reduction," in *INTERSPEECH*, 2011, pp. 857–860.

[3] Najim Dehak, Réda Dehak, Patrick Kenny, Niko Brümmer, Pierre Ouellet, and Pierre Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *INTERSPEECH*, 2009, pp. 1559–1562.

[4] David Martínez, Oldrich Plchot, Lukás Burget, Ondrej Glembek, and Pavel Matejka, "Language recognition in ivectors space," in *INTERSPEECH*, 2011, pp. 861–864.

[5] J.C. Wells, *Accents of English: An Introduction*, Accents of English. Cambridge University Press, 1982.

[6] Abualsoud Hanani, Martin J. Russell, and Michael J. Carey, "Human and computer recognition of regional accents and ethnic groups from British English speech," *Computer Speech and Language*, 2012.

[7] C. Vair, D. Colibro, F. Castaldo, E. Dalmasso, and P. Laface, "Channel factors compensation in model and feature domain for speaker recognition," in *IEEE Odyssey*, June 2006, pp. 1 –6.

[8] S.M. D'Arcy, J.M. Russell, S.R. Browning, and M.J. Tomlinson, "The Accents of the British Isles (ABI) Corpus," in *Modelisations pour l'Identification des Langues. MIDL Paris*, 2005, pp. 115–119.

[9] Jongseo Sohn, Student Member, Nam Soo Kim, and Wonyong Sung, "A statistical model-based voice activity detection," *IEEE Signal Process. Lett*, vol. 6, pp. 1–3, 1999.

[10] Pedro A. Torres-Carrasquillo, Elliot Singer, Mary A. Kohler, and J. R. Deller, "Approaches to language identification using gaussian mixture models and shifted delta cepstral features," in *Proc. ICSLP 2002*, 2002, pp. 89–92.

[11] Jason Pelecanos and Sridha Sridharan, "Feature Warping for Robust Speaker Verification," in *IEEE Odyssey*, 2001, pp. 213–218.

[12] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *Communications, IEEE Transactions on*, vol. 28, no. 1, pp. 84 – 95, Jan 1980.

[13] Brian D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, Jan. 1996.

[14] Carol Pedersen and Joachim Diederich, "Accent in speech samples: Support vector machines for classification and rule extraction," in *Rule Extraction from Support Vector Machines*, vol. 80 of *Studies in Computational Intelligence*, pp. 205–226. Springer, 2008.

[15] Emmanuel Ferragne and François Pellegrino, "Speaker classification ii," chapter Automatic Dialect Identification: A Study of British English, pp. 243–257. Springer-Verlag, Berlin, Heidelberg, 2007.