Artif Intell Rev (2007) 27:295–307
DOI 10.1007/s10462-008-9071-8

An evaluation of one-class classification techniques for speaker verification

Anthony Brew · Marco Grimaldi · Pádraig Cunningham

Published online: 13 September 2008

Abstract Speaker verification is a challenging problem in speaker recognition where the objective is to determine whether a segment of speech in fact comes from a specific individual. In supervised machine learning terms this is a challenging problem as, while examples belonging to the target class are easy to gather, the set of counter-examples is completely open. This makes it difficult to cast this as a supervised classification problem as it is difficult to construct a *representative* set of counter examples. So we cast this as a one-class classification problem and evaluate a variety of state-of-the-art one-class classification techniques on a benchmark speech recognition dataset. We construct this as a two-level classification process whereby, at the lower level, speech segments of 20 ms in length are classified and then a decision on an complete speech sample is made by aggregating these component classifications. We show that of the one-class classification techniques we evaluate, Gaussian Mixture Models shows the best performance on this task.

Keywords One-class classifiers · Speaker verification · Gaussian mixture models

1 Introduction

In speaker recognition research two separate problem categories are identified; speaker identification and speaker verification (Reynolds 1995). In machine learning, speaker identification is an n-class supervised learning problem where the query sample is matched to one of n classes in the training data. Speaker verification might be considered a binary classification problem in that the objective is to determine whether or not the query is from the individual

A. Brew (✉) · M. Grimaldi · P. Cunningham
Department of Computer Science and Informatics, University College Dublin, Dublin, Ireland
e-mail: anthony.brew@ucd.ie

M. Grimaldi
e-mail: marco.grimaldi@ucd.ie

P. Cunningham
e-mail: padraig.cunningham@ucd.ie

whose identity is claimed for the query. Given that binary classification is normally easier than multi-class classification, speaker verification would appear to be an easier problem to solve than speaker identification. However, real-world examples of the speaker verification problem, as arising for instance in security applications, are very challenging because of their *open* nature. If the utterances of an individual are the examples of the class to be recognised then the *non-class* examples cover everything else. For this reason it is worth analysing the merit of casting this as a one-class classification problem rather than a binary classification problem.

One-class classifiers (OCCs) have emerged as a set of techniques for situations where labelled data exists for only one of the classes in a two-class problem. For instance, in industrial inspection tasks, abundant data may only exist describing the process operating correctly. Training data describing the myriad of ways the system might operate incorrectly are difficult or impossible to gather. The philosophy behind the OCC approach is to develop a classifier that *characterises* the target class, and thus can distinguish it from all counter-examples.

A related problem arises where negative examples exist, but their distribution cannot be characterised. For example, it is easy to provide characteristic examples of the writings of Shakespeare but impossible to provide examples of the counter-class (material *not* by Shakespeare). While such problems are also appropriate for the OCC approach, the motivation is slightly different—counter examples are in fact available but it is difficult to construct a set of counter examples with good coverage of the universe of possible counter examples.

In speaker verification the problem is generally cast as a binary problem, unfortunately the impostor class is impossible to accurately model. Nevertheless non-class examples can have a role in OCCs whereby they are used for threshold setting. While we recognise that such a use of non-class data may dramatically improve performance in this paper we are concerned with preparing a base-line analysis where OCCs are trained solely on target data. The evaluation presented here is carried out on the CHAINS corpus introduced by Cummins et al (2006).

The paper proceeds with an overview of the speaker recognition research area in Sect. 2 and a brief review of the relevant OCC techniques in Sect. 3. The main results of the evaluation are presented in Sect. 4 and the paper concludes with a summary and some suggestions for future work in Sect. 5.

2 Speaker verification

Speaker recognition systems aim to extract, characterise and recognise the information enclosed in the speech signal conveying the identity of a speaker. The general area of speaker recognition includes two fundamental tasks: *speaker identification* and *speaker verification* (Bimbot et al. 2004; Reynolds 2002). Speaker identification is the task of assigning an unknown voice to one of the speakers known by the system: it is assumed that the voice must come from a fixed set of speakers. Thus, the system must solve a *n*-class classification problem and the task is often referred to as *closed-set* identification.

On the other hand, speaker verification refers to the case of *open-set* identification: it is generally assumed that the unknown voice may come from an impostor, not all the speakers accessing the system are known. In this case, the standard approach is based on a likelihood ratio test to distinguish between the two hypotheses: the test speech comes from the claimed speaker or from an impostor. Furthermore, depending on the specific application, speaker verification systems can work in a text-dependent or text-independent setup. In text-dependent

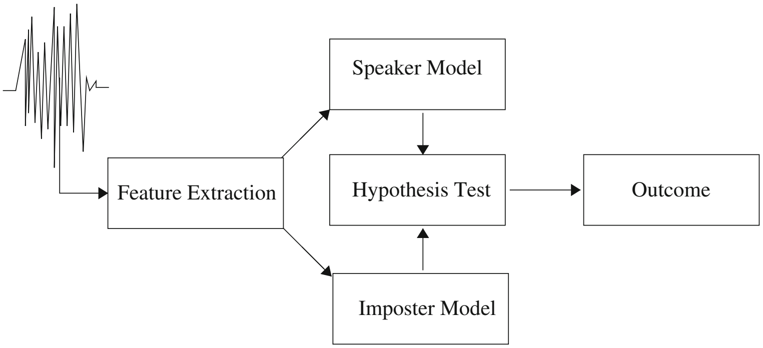

Fig. 1 The components of a speaker verification system

applications the verification system has prior knowledge of the text to be spoken (e.g. a pass-phrase). In a text-independent application, no prior knowledge of the text to be spoken is provided to the system.

Generally, speaker verification systems are composed of three main components as shown in Fig. 1: a front-end responsible for signal processing and feature extraction, a model for each speaker allowed to access the system and a model for impostor detection. In the next two sections we introduce each individual component of the verification system.

2.1 Front-end processing and feature extraction

As a first step, the front-end module of the verification system generally performs speech activity detection to remove the non-speech portion of the signal. Next, features embodying information on the speaker identity are extracted from the speech signal. Finally, the front-end implements some form of channel compensation in order to remove those spectral characteristics that are dependent on the acquisition channel (e.g. microphone) and do not reflect the speaker identity.

In most speaker verification and identification systems, some form of spectral-based parametrisation is used to encode the speech in machine readable form. Typically short-term analysis (about 20 ms) is used to compute a sequence of magnitude spectra. Most commonly, the spectra obtained are then converted into cepstral coefficients and the frequency scale warped into the Mel scale (Bimbot et al. 2004).

In this work, conventional Mel Frequency Cepstral Coefficients (MFCCs) feature vectors are employed for speech parametrisation. Twenty-five MFCCs are used for speech parametrisation, extracted using a Hamming window of about 20 ms. The zeroth cepstral coefficients (the DC level of the log-spectral energies) are not used in the feature vector.

2.2 Speaker modelling

The feature vectors extracted from the training speech material are used to create a set of speaker models, to verify if the test speech sample belongs to one of the speakers in the pool. The modelling of a speaker may be implemented according to various approaches, i.e. nearest neighbour, neural networks, Hidden Markov Models (HMMs) and Support Vector Machines (SVMs). Generally, the selection of the model adopted is largely dependent on the type of speech used, the expected performance and the computational and storage cost (Reynolds

2002). From published results (e.g. Reynolds 1995), HMMs based systems generally produce the best performance and in the case of text-independent applications single state HMMs—also known as Gaussian Mixture Models (GMMs)—are the most commonly used. Neural networks have been largely tested in this context also, however some of their shortcomings (such as the fact that the optimal structure has to be selected by trial-and-error procedures) have been judged crucial in the area of speaker verification (Bimbot et al. 2004; Reynolds 2002). SVMs, on the other hand, have been the subject of recent studies (Bimbot et al. 2004) aimed at adapting this extremely powerful technique to the problem of speaker verification.

2.3 Impostor modelling

Two main approaches are used to obtain the impostor model used in the likelihood ratio test implemented in speaker verification systems. The first approach—known as likelihood sets, cohort or background speakers (Bimbot et al. 2004; Reynolds 2002)—uses a set of other speakers to cover the space of alternative hypotheses. The impostor score is usually computed as a function (e.g. max, average) of the match scores obtained from the alternative models. It is generally recognised that this approach requires a speaker-specific background speaker set to obtain the best performance (Bimbot et al. 2004). The second approach to impostor modelling uses a single speaker-independent model trained on speech from a large number of speakers. This approach is usually referred to as general model, world model or universal background model (UBM) (Bimbot et al. 2004). The main advantage of the UBM approach is that a single speaker-independent model is trained and then used for all the speakers in the pool. This approach has become the predominant approach used in speaker verification systems. Generally, these two approaches can be applied to any speaker modelling technique (Reynolds 2002).

3 One-class classification

When employing binary classification we attempt to train a known speaker against anything that is ‘not’ from the speaker. This is an unfortunate scenario, as to sample everything that is ‘not’ is an impossible task. We are training with a class that is statistically well-sampled versus a class that is not. This statistical imbalance in the training set may lead to the creation of a system that does not generalise well when run against non-training data.

The area of OCC is well adapted to such problems; one builds a model that creates a boundary around the well-sampled target distribution that rejects all but a small percentage f of target examples and hence hopes to be able to identify $(100 - f)\%$ of the target while rejecting as many of the outlier class as possible. Most OCCs will produce a score for a given example and if it lies above a given threshold it is classified as a member of the target class.

The data we have is a sequence of windows of Mel cepstral coefficients where each example represents 20 ms of speech and there is an overlap between examples of 10 ms. Each individual example (time slice) is not particularly informative to predict the class of the data. It is assumed that groups of slices are taken from the same single source speaker. An approach that is used in the n-class problem is to take many slices accounting for a given period of speech and aggregate the scores to strengthen the evidence that this *set* of time windows comes from a given source (Reynolds 1995). Here we propose a similar strategy, which is explained in Sect. 3.2.

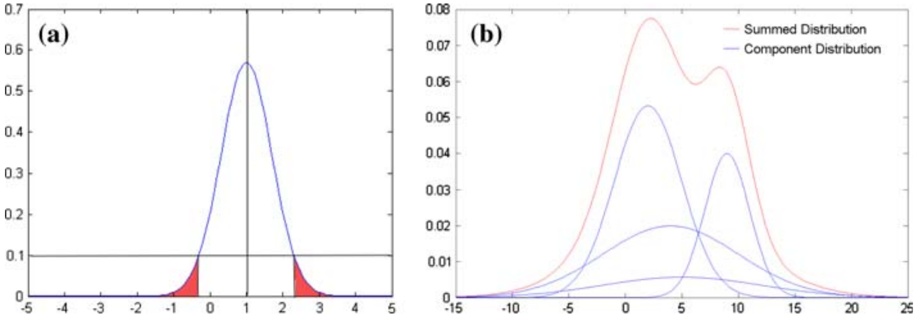

Fig. 2 In (**a**) a single Gaussian is used to model the underlying target data, the value of the gaussian function is thresholded so that when a value of less than the threshold is found the item will be rejected. Whereas (**b**) (a GMM) shows by using many individual Gaussian models and weighting them, more complex distribution shapes can be formed

3.1 Selection of classifiers used

In this work we chose four OCCs to compare: a single Gaussian, a GMM (Fig. 2), a Nearest Neighbour based approach and an approach inspired by Support Vector Machines (SVM) the Support Vector Domain Description (SVDD) (Tax and Duin 1999).

3.1.1 Single Gaussian

A simple model for any problem is to assume the data is drawn from the Gaussian distribution (Tax 2001). This model assumes that the data fits a unimodal convex data description. The function that determines the score, where μ is the mean of all the 'target' points, Σ it the covariance matrix of the target points and d is the dimensionality of the problem given by the classifier is:

$$p(\mathbf{x}, \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|} \exp\left(\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{z} - \mu)\right)$$

3.1.2 Gaussian mixture models

GMMs can model more complex underlying distributions. As the name suggests GMMs are the combination of several Gaussian Models (a weighted sum). The underlying Gaussians of the GMM have been shown to represent the characteristic spectral shapes of the phonetic sounds that make up a person's voice (Reynolds 1995). In these experiments we use only diagonal covariance matrices for each Gaussian. It is argued in Reynolds (1995) that this limits the computational complexity of the problem and that by adding more underlying mixtures of diagonal covariance, is equivalent to modelling with fewer full covariance matrices. The function that determines the score, where p is the Gaussian of an individual model, the α_i's are mixture weights which sum to one, and k is the number of individual Gaussian models used is given by:

$$p_{GMM}(\mathbf{x}) = \sum_{i=1}^{k} \alpha_i p(\mathbf{x}, \mu_i, \Sigma_i) \tag{1}$$

3.1.3 Support vector domain description

The SVDD finds a sphere of minimum radius that encloses all of the target data. This is cast as a minimisation problem where one finds a radius **R** and centre **c** such that the following minimisation problem is solved. Data that lies further than a given distance from the centre of the sphere is labelled as an outlier.

$$\begin{aligned} &\min \mathbf{R}^2 s.t. \\ &\langle \mathbf{x}_i - \mathbf{c}, \mathbf{x}_i - \mathbf{c} \rangle \leq \mathbf{R}^2 \quad \forall \mathbf{x}_i \in \text{target} \end{aligned}$$

By replacing the inner product in the above problem with a 'kernel' function more flexible decision boundaries may be found. Once the optimisation problem is solved, new points that are further than a given distance from the centre are labelled as outliers. Details of the mathematical derivation and details of the method can be found in Tax and Duin (1999). The kernel function that is used in this work is the Gaussian kernel:

$$K(\mathbf{x}, \mathbf{x}') = e^{-\|\mathbf{x}-\mathbf{x}'\|^2/2\sigma^2}$$

The σ parameter is selected for the kernel depending on the classification task at hand. It is often known as the width parameter and controls the flexibility of the decision boundary. If σ is set too high the model will tend to under-fit the data and if it is set too small it will over-fit the data.

A variation of the algorithm finds a minimum hyper-sphere that rejects all but some given fraction of the data. Upon the completion of the aforementioned optimisation problem a corresponding set of weights α_i that correspond to the weight of each training example in the solution settles to a global minimum. What is interesting is that for most of the items $\mathbf{x}_i$ in the training set their weight α_i will be zero, and so they are not used in the classification. The elements $\mathbf{x}_i$ in the training set that do not have zero weights are known as *support vectors* and new incoming data can be classified using kernel operations on these support vectors only (Tax and Duin 1999).

Through these support vectors and their corresponding weights it is possible to calculate the radius of the sphere in the space that the kernel function defines.

3.1.4 Nearest neighbour

We carried out a comparison of several nearest neighbour techniques to identify the variation producing the best results on an individual time window, as determined by the highest area under the ROC curve (AUC) (Bradley 1997). This nearest neighbour method we chose takes the magnitude of the average directional vector to the k nearest neighbours as the measure to threshold against.

3.2 Aggregation of scores

Each of the above classifiers creates a score that is thresholded and if it is above a given threshold it is considered to be from the 'target' class and otherwise it is considered to be an 'outlier'. Since each individual score is a high-variance prediction of the class, it makes sense to aggregate a sequence of scores when making a prediction.

The component scores from classifiers do not directly represent the probability of the item belonging to a class so in order to combine scores it is better to convert the raw scores to probabilities. To achieve this, the score for the item belonging to the target and the score for it belonging to the outliers was normalised to sum to one.

To combine the probabilities for an individual classification we used a simple summation, a strategy that has been shown to give good results (Taniguchi and Tresp 1997). Although the alternative product rule follows directly from a Baysian viewpoint, under the assumption of independence, it can dramatically amplify errors as more slices are added (Kittler et al. 1998). The sum rule is much less sensitive to estimation error at the single slice level (Kittler et al. 1998) and hence was used for the evaluation presented here.

4 Evaluation

As explained in Sect. 2.3 there are two ways to model the impostor. When using likelihood sets, an individual background model per speaker needs to be trained, in order to achieve optimal performance. This is a poor solution as each speaker requires their own background set to achieve optimal performance (Bimbot et al. 2004). The preferred alternative is a Universal Base Model (UBM), which is global background model trained on a large volume of speech (1 + h of speech). The UBM should be reflective of the expected alternative speech to be encountered during recognition as an impostor model (Reynolds et al. 2000). The assumption that this speech will be of the 'expected alternative' would appear to be counter to the goal we wish to attain, namely 'to be able to verify a speaker without full knowledge of the impostor'. While we recognise that outlier examples will considerably help in the verification task (by tightening the decision boundary around the target speaker), We believe a comparative analysis of OCCs trained only on 'target' data is the most fair comparison to benchmark classifiers against each other. We expect that GMM's will outperform other classifiers at a cepstral level as it has been shown that the underlying Gaussian components of the model inherently model the underlying distributions of the phonetic sound production (Reynolds and Rose 1995).

4.1 Experimental setup

The CHAINS corpus (Cummins et al 2006) is the result of an effort to provide a speech database expressly designed to characterise speakers as individuals.[1] The corpus contains the recordings of 36 speakers obtained in two different sessions with a time separation of about two months taken in two different recording environments. Across the two sessions, each speaker provides recordings in six different speaking styles.

In the experiments conducted here we used one speaking style SOLO from the CHAINS corpus, where 16 speakers read a prepared text alone. We only used speech from one recording session so that we would not have to manage problems created by different channel effects between different microphones. The training set was made up of speakers reading ten sentences making up on average 24 s of speech per speaker. The test set was made up of speakers reading nine sentences later in the same session making up on average 16 s of speech per speaker. We used the preprocessed Mel Cepstral coefficients from train and test files in the CHAINS corpus. Each Mel Feature vector represents a 20 ms slice of time. As noted above, we train using only data on the target speaker to provide a fair comparison between classifiers.

In order to select the parameters of the base classifiers for each speaker we built them on the individual slice level first. The parameters for each classifier (number of mixture models for GMM, σ for the SVDD, etc.) were selected by using a consistency criterion based on the

[1] The corpus is freely available for research purposes from http://chains.ucd.ie.

method is described in detail in Tax and Muller (2004). This criterion seeks to reject $f\%$ (in our case 10%) of individual target slices and evaluates more and more complex parameters for the classifier until the model becomes statistically unstable. This process defines the most complex classifier, which can still reliably be trained on the data (Tax and Muller 2004). For k-NN the simplest model was already unstable, so for selection of k here we ran cross validation and found $k = 40$ to be the best parameter value. It should be noted that for the models other than k-NN the parameter selection was done for each individual speaker rather than on a global basis.

On the training set the summed scores of the individual slices from the target class were thresholded to reject 10% of the target class in order to provide a tight decision boundary around the target class.

4.2 Discussion of results

When an evaluation on a single 20 ms time window was done on the data we found a high variability of scores between speakers (Fig. 3). With more slices all classification strategies increased performance and this trend continuing at after 1 s of speech (Fig. 4). A very wide range of performances was found across all speakers (Fig. 5.).

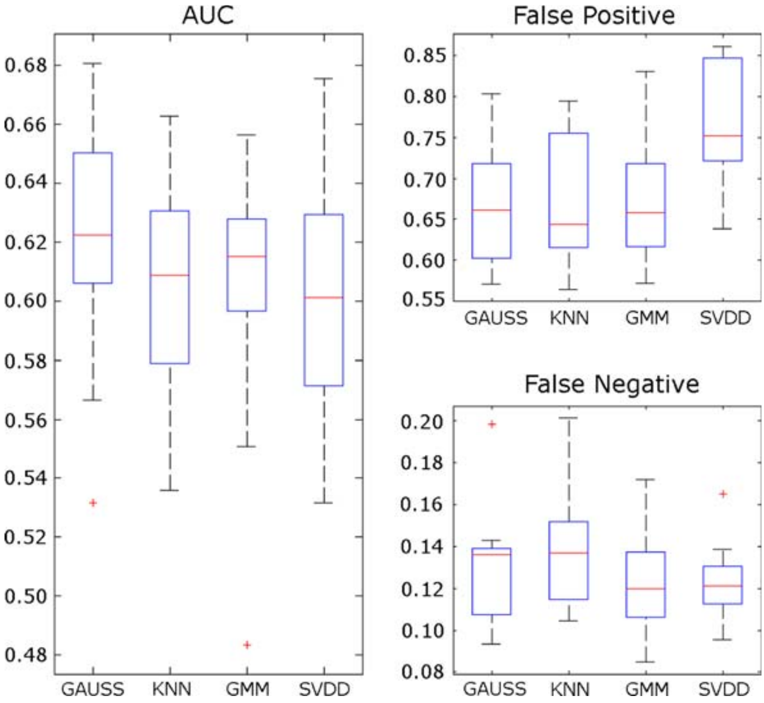

Fig. 3 Looking at each speaker in turn using only a single time slice to classify whether the speaker was an 'outlier' or a 'target' we found their associated AUC rates, False Positive Rates and False Negative Rates. The above diagram shows the box and whisker diagrams of the scores obtained from the 16 speakers on each classifier. It is clear that at a single time slice level classifiers have difficultly deciding whether the object comes from the 'target' or 'outlier' class. The best false positive rate over the 16 speakers was 0.56 where worst false positive error rate as high as 0.86 which can be observed for the SVDD

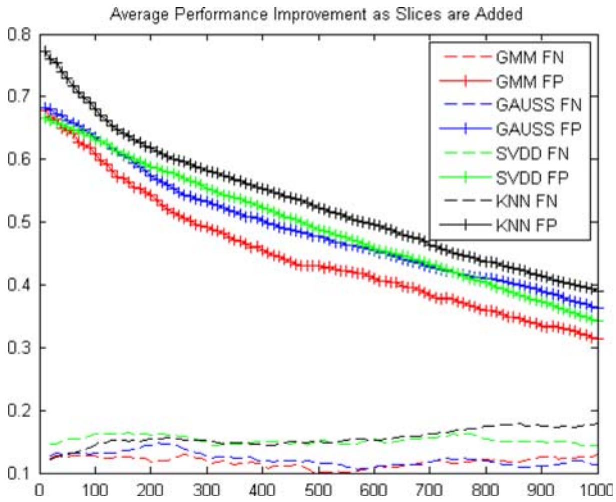

Fig. 4 The average performance increases as the number of slices is increased, this is seen to be continuing to rise even when 100 slices (∼1 s) of speech is used in the classification

By looking at the false negative scores it can be seen that on average none of the classifiers hit their trained target of 10% rejection during test and rejected more than this base amount on average. The GMM and single Gaussian model best fitted their target distribution between training and test sets. As mentioned in Reynolds and Rose (1995) the average speech spectrum contains speaker specific information and for this reason was not removed in these experiments. The average speech spectrum can vary considerably over even short periods of time (Reynolds and Rose 1995) and so this shift may account for the drift between the trained false negative rate differing from the values attained on the training set.

4.3 Error analysis

It is clear from Fig. 5 that the OCC approach to speaker verification is effective for some speakers but performs badly for other subjects. It is notable where errors are high that it is the false positive component of the error that is the problem. Perhaps the most remarkable aspect of the false positive rates is the variability: for some speakers a respectable figure of 10% of false positives is achieved but for other speakers the false positive figures are above 80%.

In order to explore this issue we performed some analysis on the data and the models in order to understand what caused the high FP error rates. Since high FP rates entail the acceptance of impostors as belonging to the positive class, our intuition would suggest that models with high FP rates are *flabby* models that cover large areas of the problem space. This hypothesis can most easily be explored in the context of the GMMs. In Eq. (1) we can see that the *spread* of the model is defined by the variance matrices Σ_i of each component model and also by the dispersion of model means μ_i. We have found that the variance in the component Gaussians rather than the dispersion of the Gaussian means correlates with the FP error rate. This can be seen in the graph on the top left of Fig. 6 which shows the correlation of the logarithm of the following score with the FP error:

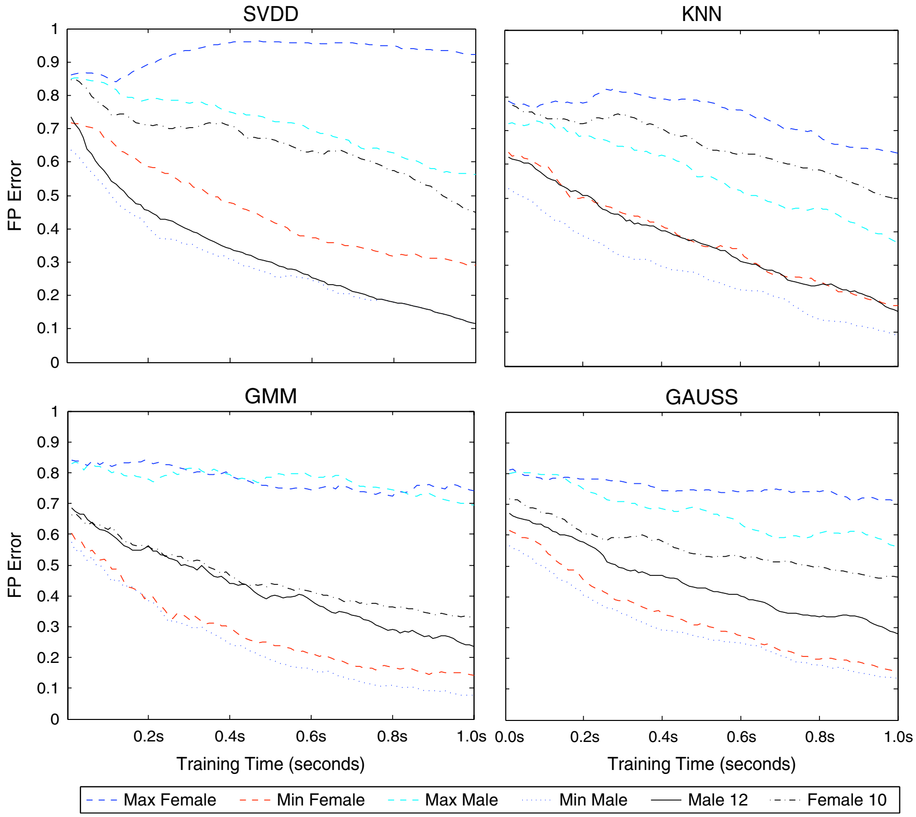

Fig. 5 It can be seen that for some individuals OCC techniques yielded good results when compared against other speakers. It is also noted that certain classification strategies preformed better for some speakers than for others indicating that model selection may also need to be considered when building an OCC for a given speaker

$$\text{Model variance score} = \sum_{i=1}^{k} \alpha_i \det(\Sigma_i) \tag{2}$$

The variance matrices Σ_i are diagonal matrices with each term representing the variance in a single dimension so the determinant is calculated by simply taking the product of the entries on the main diagonal. This simple measure effectively *predicts* the FP error, the dominant component of the overall error. This is particularly useful in OCC as it is not practical to do cross validation as an aid to model selection. In Fig. 6 we show some other measures that correlate (and thus *predict*) the FP error for the other classifiers. These other measures are summarised as follows:

- For the **SVDD** the radius of hypersphere as defined in Sect. 3 is the obvious measure of the model spread. However, it is not appropriate to compare the hypersphere radii for different speakers directly as they are based on different kernel widths—the kernel width is specialised for each speaker as part of the training process as outlined in Sect. 4.1. This parameter setting is disabled for the purpose of the analysis presented here: instead the kernel width is set to the average value for all speakers. So the results presented in Fig. 6

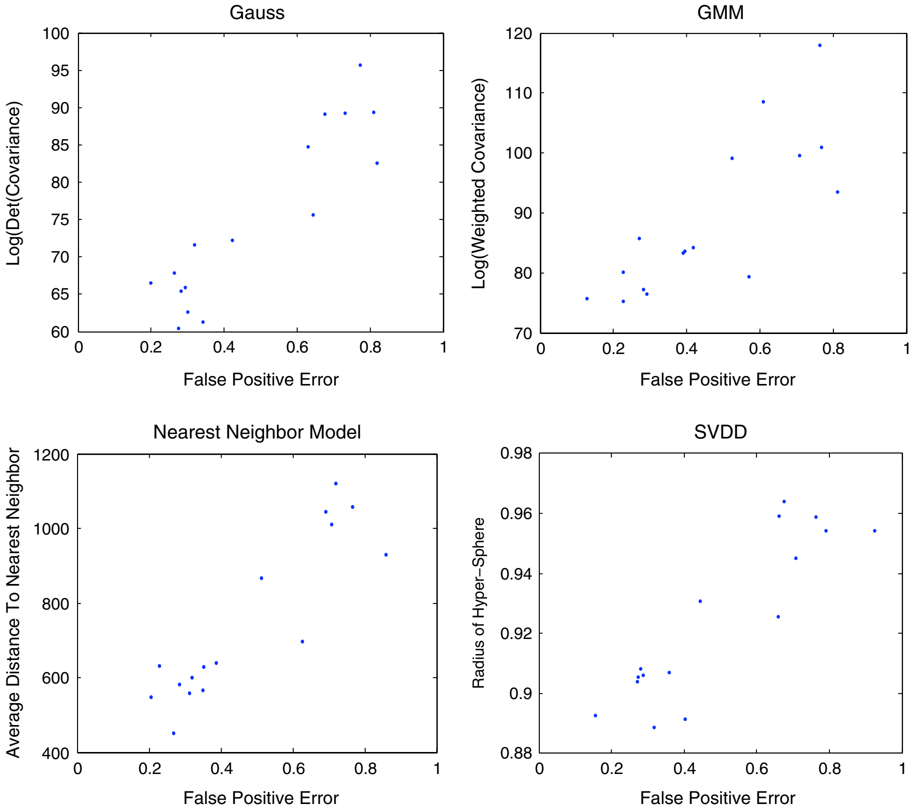

Fig. 6 The graphs in this figure illustrate how the *spread* in the models used by the different classifiers correlates with the FP error. The appropriate parameter for the different models are as follows; for the SVDD it is the radius of the hypersphere that captured all of the data for some fixed kernel width, in the case of the k-NN it is the average distance to the k nearest neighbours ($k = 40$), for the single Gaussian model it is the determinant of the covariance matrix, and for the Mixture of Gaussian's it is the weighted sum of the covariance matrices for each Gaussian in the model

are based on the same kernel width for all speakers. It can be seen that this hypersphere radius measure correlates well with the FP error.

- It would be expected that some measure of density would correlate with the FP error rate for the k-**NN** classifiers. Given that all the classifiers use the same size of training set, an appropriate measure of density would be the average distance to the k (40 in this case) nearest neighbours. Again, it can be seen in Fig. 6 that this measure correlates well with the FP error.
- The appropriate measure for the single **Gaussian** model is a simplification of the measure shown in Eq. (2). For a single Gaussian there is one covariance matrix Σ and we find that the determinant of this matrix correlates well with the FP error (see the graph on the top right in Fig. 6).

In this analysis the error figure used was the error found when 50 slices of speech were used. This analysis shows that, for some speakers, the models have a considerable spread in the input space and thus are prone to FPs. But if we consider the particular case of the singe Gaussian models it is clear that the problem is due to an inherent spread in the *data* rather than a problem in the process of building the models. If, for a single Gaussian model, $det(\Sigma)$ is large then that reflects a spread in the underlying data that cannot readily be fixed by the modelling process. This illustrates shortcomings in the feature extraction process whereby the extracted features do adequately separate the speakers. For this reason we are currently working on identifying an improved set of features to use for classification.

5 Conclusions and future work

The objective with this work was to assess whether state-of-the-art OCC techniques are effective for speaker verification. The evaluation presented here shows that GMM affords the best improvements on average of the four classification techniques examined. However to make an accurate prediction based on the techniques used thus far *only* using one class to train on the current feature set appears to be unrealistic due to the variability in performance between speakers.

It has been seen that different classification models perform better for different speakers and it would be interesting to try to tease out the reasons for these differences. Some speakers failed to perform well across all classification strategies (e.g. ‘irf07’ Fig. 5). These poor performing speakers may merely sit very close to one another in a region of feature space and so the spread of their underlying cepstral distributions overlaps more prominently than with other speakers. This could be investigated by looking at cross correlation matrices to see which speaker the false positives for a given individual comes from and in what percentage.

It has been suggested that an important future direction for speaker verification will be in the development of higher level speech features (Reynolds 2003) that capture not only the individual time slices but also the temporal information. While the GMM-UBM model is the best approach on a slice by slice level, the use of SVMs employing sequence kernels for speech recognition is an active research area. The combination of these classifiers with the GMM-UBM has shown considerable promise (Wan and Renals 2005). An investigation into an extended feature space would seem appropriate.

The next step in this evaluation is to compare this against a binary classification approach where a broad set of speakers is sampled to produce representative training examples of the non-class.

Acknowledgment This research is supported by Science Foundation Ireland Grant No. 05/IN.1/I24.

References

Bimbot F, Bonastre J, Fredouille C, Gravier G, Magrin-Chagnolleau I, Meignier S, Merlin T, Ortega-Garcia J, Petrovska-Delacretaz , Reynolds D D (2004) A tutorial on text-independent speaker verification. EURASIP J Appl Signal Process 4:430–451

Bradley AP (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognit 30(7):1145–1159

Cummins F, Grimaldi M, Leonard T, Simko J (2006) The CHAINS corpus: CHAracterizing INdividual Speakers. In: Proceedings of SPECOM’06, pp 431–435

Kittler J, Hatef M, Duin RPW, Matas J (1998) On combining classifiers. Pattern Anal Mach Intell IEEE Trans 20(3):226–239

Reynolds D (1995) Speaker identification and verification using Gaussian mixture speaker models. Speech Commun 17(1):91–108

Reynolds D (2002) An overview of automatic speaker recognition technology. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing

Reynolds DA (2003) Channel robust speaker verification via feature mapping. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03), vol 2, pp II–53–6

Reynolds DA, Rose RC (1995) Robust text-independent speaker identification using gaussian mixture speaker models. Speech Audio Process IEEE Trans 3(1):72–83

Reynolds DA, Quatieri TF, Dunn RB (2000) Speaker verification using adapted gaussian mixture models. Digital Signal Processing, pp 19–41

Taniguchi M, Tresp V (1997) Averaging regularized estimators. Neural Comput 9(5):1163–1178

Tax DMJ (2001) One-class classification. Ph.D. thesis, Delft University of Technology

Tax DMJ, Duin RPW (1999) Support vector domain description. Pattern Recogn Lett 20(11–13):1191–1199

Tax DMJ, Muller KR (2004) A consistency-based model selection for one-class classification. In: Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004), vol 3, pp 363–366

Wan V, Renals S (2005) Speaker verification using sequence discriminant support vector machines. Speech Audio Process IEEE Trans 13(2):203–210