Research Repository UCD

Title	An Evaluation of One-Class Classification Techniques for Speaker Verification
Authors(s)	Brew, Anthony, Grimaldi, Marco, Cunningham, Pádraig
Publication date	2007-08-13
Publication information	Brew, Anthony, Marco Grimaldi, and Pádraig Cunningham. An Evaluation of One-Class Classification Techniques for Speaker Verification. University College Dublin. School of Computer Science and Informatics, August 13, 2007.
Series	UCD CSI Technical Reports, UCD-CSI-2007-8
Publisher	University College Dublin. School of Computer Science and Informatics
Item record/more information	http://hdl.handle.net/10197/12361

Downloaded 2023-04-07T10:52:29Z

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)

© Some rights reserved. For more information

An Evaluation of One-Class Classification Techniques for Speaker Verification

Anthony Brew

Marco Grimaldi

Pádraig Cunningham

University College Dublin
Technical Report UCD-CSI-2007-8
August 13th, 2007

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. In this paper 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 show that of the one-class classification techniques, Gaussian Mixture Models shows the best performance on this task.

1 Introduction

In speaker recognition research two separate problem categories are identified; speaker identification and speaker verification [1]. 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 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 labeled 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 imposter is impossible to accurately model. Although we recognise the fact that use of an impostor model will dramatically improve performance, we believe the area merits a base evaluation, where OCCs are trained solely on target data so that direct comparisons can be fairly made. The evaluation presented here is carried out on the CHAINS corpus introduced by Cummins et al. [2].

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

2 Speaker Verification

Speaker recognition systems aim to extract, characterize and recognize 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* [3, 4]. 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 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 Figure 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. The big advantage of the OCC approach to speaker verification is that there is no need

Figure 1: Speaker Verification System.

for an impostor model as the classifier is trained on the speaker model only. 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 parametrization 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 [3].

In this work, conventional Mel Frequency Cepstral Coefficients (MFCCs) feature vectors are employed for speech parametrization. 25 MFCCs are used for speech parametrization, 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 Modeling

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 modeling of a speaker may be implemented according to various approaches, i.e. k nearest neighbour (k -NN), 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 [4]. From published results (e.g. [1]), HMM-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 [3, 4]. SVMs, on the other hand, have been the subject of recent studies [3] aimed at adapting this extremely powerful technique to the problem of speaker verification.

2.3 Impostor Modeling

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 [3, 4] – 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 recognized that this approach requires a speaker-specific background speaker set to obtain the best performance [3]. The second approach to impostor modeling 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) [3]. 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 modeling technique [4].

3 One-Class Classification (OCC)

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 sets of Mel cepstral coefficients where each set represents a time slice of 20ms of speech and there is an overlap between time slices of 10ms. Each individual 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

window over a sequence of slices comes from a given source [1]. Here we propose a similar strategy, which is explained in section 3.2.

3.1 Selection of Classifiers Used

In this work we chose four OCCs to compare: a single Gaussian, a GMM, a k -NN based approach and an approach based on SVMs, the support vector domain description (SVDD) [5].

Figure 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.

3.1.1 Single Gaussian:

A simple model for any problem is to assume the data is drawn from the Gaussian distribution [6]. 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 is the covariance matrix of the target points and d is the dimensionality of the problem given by the classifier is :

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

3.1.2 Gaussian Mixture Models (GMMs):

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 [1]. In these experiments we use only diagonal covariance matrices for each Gaussian. It is argued in [1] that this limits the computational complexity of the problem and adding more underlying mixtures of diagonal covariance, is equivalent to modeling 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:

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

3.1.3 Support Vector Domain Description (SVDD):

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 $\mathbf{R}$ and centre $\mathbf{c}$ such that the following minimisation problem is solved. Data that lies further than a given distance from the centre of the sphere is labeled as an outlier.

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

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 labeled as outliers. Details of the mathematical derivation and details of the method can be found in [5]. 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.

3.1.4 k -Nearest Neighbour:

We carried out a comparison of several nearest neighbour techniques to identify the variation producing the best results on a single time slice, as determined by the highest area under the ROC curve (AUC)[7]. The k -NN method we chose takes the magnitude of the average directional vector to the k nearest neighbours as its output to threshold against.

3.2 Aggregation of Scores

Each of the above classifiers produces a score, if the score 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 were normalised to sum to one.

To combine the probabilities for individual classification we used a simple summation, a strategy that has been shown to give good results [8]. Although the alternative product rule follows directly from a Bayesian viewpoint, under the assumption of independence, it can dramatically amplify errors as more slices are added [9]. The sum rule is much less sensitive to estimation error at the single slice level [9] and hence was used for the evaluation presented here.

4 Evaluation

The objective in the evaluation is to assess the performance of OCC techniques on the problem of speaker verification. As explained in section 2 the OCC will have the advantage that the problem of designing and constructing an impostor model is avoided.

Thus we are interested in the absolute performance of OCCs trained on ‘target’ data only. And we are interested in a comparative analysis of the OCCs described in section 3 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 [10].

4.1 Experimental Setup

The CHAINS corpus [2] is the result of an effort to provide a speech database expressly designed to characterize speakers as individuals¹. The corpus contains the recordings of 36 speakers obtained in two different sessions with a time separation of about two months taken on 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, 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 imposed by different channel effects between different microphones. The training set was made up of speakers reading ten sentences making up on average 24 seconds on speech per speaker. The test set was made up of speakers reading nine sentences later in the same session making up on average 16 seconds one speech per speaker. As noted above, we train only using target data to provide a fair comparison between classifiers.

In order to select the parameters of the base classifiers we built them on the individual slice level first. The parameters for each classifier (number of mixture models for GMM,

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

σ for the SVDD, etc.) were selected by using a consistency criterion. This criterion selects to reject $f\%$ (in our case 10%) of individual target slices and then uses 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 [11]. For k -NN the simplest model was already unstable, so to select k we ran cross validation and found a best value to use ($k = 40$). It should be noted that parameter selection was done for each individual speaker rather than on a global basis.

On the training set the summed scores of 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 slice was done on the data we found a high variability of scores between speakers (Fig3a.). With more slices all classification strategies increased performance and this trend continued even at the 10 second mark (Fig3b.). A very wide range of performances was found across all speakers (Fig4.). This is highlighted by the fact that, although on average the SVDD was outperformed by all other classifiers, it was the classifier which achieved the best performance on an individual speaker with a false positive rate over a 10 second window of 11.54%.

Figure 3: In (a) it is clear that, working with a single slice only, classifiers have difficulty deciding whether the object comes from the ‘target’ or ‘outlier’ class. Over the 16 speakers the best false positive rate found was 0.56 while false positive error rates as high as 0.86 can be observed for the SVDD. In (b) the average performance increases as the number of slices is increased, this is seen to be continuing to rise even when 1000 slices (~ 10 seconds) 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 Gaussian model best fitted their target distribution between training and test sets. As mentioned in [10] 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 [10] and so this shift may account for the drift between the trained false negative rate differing from the values attained on the training set.

Figure 4: 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 performed 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.

5 Conclusions and Future Work

The objective with this work was to assess whether state-of-the-art one-class classification techniques are effective for speaker verification. The evaluation presented here shows that GMM affords the best performance on average of the four classification techniques examined. As expected, the accuracy increases as the speech window used for classification is increased (i.e. more slices are added). While the accuracy on 10 seconds of speech is adequate for some speakers (i.e. $< 10\%$ FP), it is not adequate for all 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. ‘Female Max’ Fig4.). 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 the future direction of speaker verification will be in the use of higher level speech features [12] that capture not only the individual time slices but also the temporal information. While the GMM-UBM model is the best approach for a slice by slice level, active research is looking at the combination of sequence kernels for SVMs. The combination of these classifiers with the GMM-UBM has shown considerable promise [13]. 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. We will also employ limited outlier data for thresholding in the OCCs – this can be expected to improve on the performance of the OCCs presented here.

References

- [1] Reynolds, D.: Speaker identification and verification using Gaussian mixture speaker models. *Speech Communication* **17** (1995) 91–108
- [2] Cummins, F., Grimaldi, M., Leonard, T., Simko, J.: The CHAINS corpus: CHAracterizing INdividual Speakers. In: *Proc of SPECOM'06*. (2006) 431–435
- [3] Bimbot, F., Bonastre, J., Fredouille, C., Gravier, G., Magrin-Chagnolleau, I., Maignier, S., Merlin, T., Ortega-Garcia, J., Petrovska-Delacretaz, D., Reynolds, D.: A tutorial on text-independent speaker verification. *EURASIP Journal on Applied Signal Processing* **4** (2004) 430–451
- [4] Reynolds, D.: An overview of automatic speaker recognition technology. In: *Proc. Int. Conference on Acoustics, Speech, and Signal Processing*. (2002)
- [5] Tax, D.M.J., Duin, R.P.W.: Support vector domain description. *Pattern Recogn. Lett.* **20** (1999) 1191–1199
- [6] Tax, D.M.J.: One-class classification. PhD thesis, Delft University of Technology (2001)
- [7] Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition* **30** (1997) 1145–1159
- [8] Taniguchi, M., Tresp, V.: Averaging regularized estimators. *Neural Computation* **9** (1997) 1163–1178
- [9] Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **20** (1998) 226–239

- [10] Reynolds, D.A., Rose, R.C.: Robust text-independent speaker identification using gaussian mixture speaker models. *Speech and Audio Processing, IEEE Trans.* **3** (1995) 72–83
- [11] Tax, D.M.J., Muller, K.R.: A consistency-based model selection for one-class classification. In: *ICPR. Volume 3.* (2004) 363–366
- [12] Reynolds, D.A.: Channel robust speaker verification via feature mapping. In: *ICASSP.* (2003) II–53–6 vol.2
- [13] Wan, V., Renals, S.: Speaker verification using sequence discriminant support vector machines. *Speech and Audio Processing, IEEE Trans* **13** (2005) 203–210