INTERSPEECH 2009 BRIGHTON

MiniVectors: an Improved GMM-SVM Approach for Speaker Verification

Xavier Anguera

Multimedia Research Group, Telefonica Research, Barcelona, Spain

xanguera@tid.es

10.21437/Interspeech.2009-396

Abstract

The accuracy levels achieved by state-of-the-art Speaker Verification systems are high enough for the technology to be used in real-life applications. Unfortunately, the transfer from the lab to the field is not as straight-forward as could be: the best performing systems can be computationally expensive to run and need large speaker model footprints. In this paper, we compare two speaker verification algorithms (GMM-SVM Supervectors and Kharroubi's GMM-SVM vectors) and propose an improvement of Kharroubi's system that: (a) achieves up to 17% relative performance improvement when compared to the Supervectors algorithm; (b) is 24% faster in run time and (c) makes use of speaker models that are 94% smaller than those needed by the Supervectors algorithm.

Index Terms: speaker verification, support vector machines, supervectors

1. Introduction

Speaker verification technology has achieved accuracy levels that are high enough to apply it in real-life and commercial applications. In fact, nowadays biometric authentication systems typically include a speaker verification module, in addition to iris, face and fingerprint recognition. Other desirable and important characteristics for these algorithms to be applied in real-life scenarios include low computational cost and small fingerprints to store the speakers' models. Unfortunately, the best performing speaker verification algorithms proposed in the literature have usually made a trade-off between accuracy and computational and storage costs.

Among the alternative systems proposed for speaker verification in recent years [1], some model the speakers by means of Gaussian mixture models (GMM) [2], using Standard features like Mel-frequency cepstral coefficients (MFCC) or perceptual lineal predictors (PLP). These systems are simple to build and have a relatively small computational complexity and model footprint. However, their accuracy has far been outperformed by support vector machine (SVM)-based techniques, which started appearing after year 2000.

Given the suitability of GMM for speaker modeling and the power of SVM as classifiers, there have been many systems proposed over the years that combined both [3, 4, 5, 6, 7, 8]. Probably, Campbell's Supervectors algorithm [7] is one of the best known GMM-SVM systems today, where the means of speaker adapted GMM models are used as inputs for the SVM. The Supervectors method achieves very good verification performance but at the cost of large model fingerprint given the high dimensionality of the support vectors. Alternatively, the GMM-SVM algorithm proposed by Kharroubi in [8] utilizes support vectors of much lower dimensionality than the Supervectors. Its reported performance is slightly worse than that of the Supervectors algorithm for selected kernels, but uses only a fraction of the stored model footprint.

Along with the different algorithm proposals, there have been several performance enhancement methods to work in conjunction with the basic algorithms: feature mapping on the MFCC/PLP feature vectors, eigen-channel model adaptation on the GMM models, nuisance attribute projection (NAP) or factor analysis in order to remove unwanted channel/itersession variability (see [1] for references).

In this paper, we compare Campbell's Supervectors algorithm [7] with Kharroubi's GMM-SVM system [8] and propose a modification in the way vectors are obtained from the speaker GMM model in order to make Kharroubi's algorithm more efficient and intuitive. We call the proposed vectors Minivectors, in contrast with Campbell's Supervectors. Results show that the Minivectors achieve better performances than Supervectors and Kharroubi's system in all tested conditions while requiring less computational power and significantly less storage space for the speaker models. The focus of this paper is on reaching a deeper understanding of how two of the best performing state-of-the-art speaker verification algorithms work. Therefore, we chose to compare the plain algorithms in our analysis, without any of the previously mentioned enhancement methods.

In Section 2, we cover the SVM basics for the speaker verification task, followed by the description of the GMM-SVM Supervectors algorithm – Section 3, Kharroubi's GMM-SVM algorithm – Section 4 – and the proposed algorithm – Section 5. Finally, our experimental results are described in Section 6, followed by some conclusions and future work.

2. Support Vector Machines

Speaker verification has received an important boost in performance in recent years, partly thanks to the application of support vector machines (SVM) [9]. SVM are linear discriminative classifiers based on the Structural Risk Minimization theory [10]. In order to tackle non-linear classification problems, the input feature space is typically transformed to a higher dimensional space via a Kernel function, where it is possible to define a hyperplane to separate both classes. It has been proven that SVM can achieve a generalization performance equal or better than other classifiers, with less training data necessary.

The SVM two class classifier is constructed by the weighted sum of a kernel function in the following way

$$f(x) = \sum_{i=1}^{N} \alpha_i t_i K(x, v_i) + d \qquad (1)$$

where x is the input data; N is the number of support vectors; α_i and d are training parameters; v_i are the support vectors, obtained via an optimization process, such as Sequential Minimal Optimization (SMO) [11]; $K(\cdot,\cdot)$ is the kernel function; and t_i are the ideal outputs, with values ± 1 depending on whether the

accompanying support vectors belong to class 0 or 1. Overall, the parameters are subject to the constraint: $\sum_{i=1}^{N} \alpha_i t_i = 0$.

The kernel function must satisfy the Mercer's condition, such that:

$$K(x, v_i) = b(x)^t b(v_i) \quad (2)$$

where $b(\cdot)$ is the mapping that converts each data vector x from the input feature space into the high-dimensional SVM space (also called the expansion space). The SVM optimization process finds a hyperplane in the expansion space that can effectively separate between the two classes with maximum margin. Such hyperplane is defined by the support vectors, v_i, that are special data points chosen from the training data. At the evaluation stage, each input vector x is classified to class 0 or 1 acccording to the sign of $f(x)$.

Two of the most commonly used kernels in speaker verification are the linear kernel and the radial basis function (RBF) kernel, given by Equations 3 and 4, respectively.

$$K(x, v_i) = x \cdot v_i \quad (3)$$

$$K(x, v_i) = \exp[-\frac{1}{2\sigma}(x - v_i)^2] \quad (4)$$

where σ is the standard deviation of the radial basis function.

3. GMM-SVM Supervectors

The concept of GMM-SVM Supervectors for speaker verification was introduced by Campbell in 2006 [7] in order to combine the generative power of GMM with the discriminative properties of SVM. Since its proposal, GMM-SVM Supervectors have been successfully used by many researchers and have been combined with techniques such as nuisance attribute projection (NAP) [7] and factor analysis for further enhanced performance.

In this section, the basic GMM-SVM Supervectors technique is described. The SVM vectors for all of the systems described in this paper are derived from a universal background model (UBM). A UBM is a Gaussian Mixture Model (GMM) trained using acoustic data of different speakers, in order to model the acoustics of speech. Figure 1b illustrates the process behind the GMM-SVM Supervectors, which in enrolment phase proceeds as follows:

1. Acoustic feature vectors X are extracted from all available training utterances of the enroling speaker.
2. A GMM model λ with M Gaussians is obtained via MAP adaptation [12] (means only) from a UBM model.
3. A supervector V_X is constructed for speaker S by concatenating the N-dimensional means – typically normalized by the corresponding standard deviation of each of the Gaussian mixtures in the adapted GMM model. For a GMM model composed of M Gaussian mixtures, it results in a $M * N$ dimensional vector.
4. An SVM classifier is trained using the V_X vectors computed for the target speaker as positive examples (class $t = +1$), and a set of impostor speaker vectors V_I (common to all enrolment speakers) as negative examples (class $t = -1$).
5. The UBM model and the SVM parameters are stored as the speaker's fingerprint.

On verification stage, given an input speech utterance – converted to a feature vector sequence –, and a speaker model

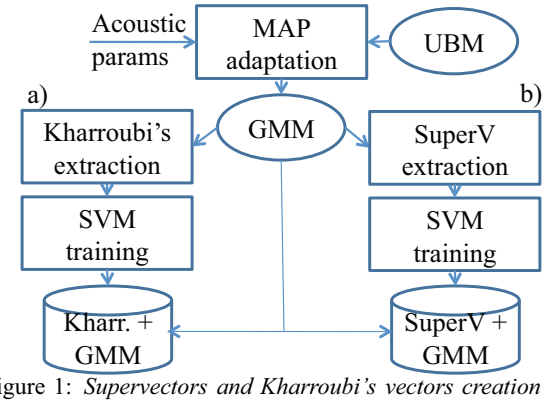

Figure 1: *Supervectors and Kharroubi's vectors creation diagram.*

to be verified, the steps $2 - 3$ of the procedure above are executed. The result is a Supervector, generated from the input sequence using the stored UBM model. The Supervector is evaluated against the speaker's SVM and a decision is made whether the enroled speaker and the input speech came from the same speaker (positive) or not (negative).

4. Kharroubi's GMM-SVM vectors

A few years before Campbell *et al.* proposed the Supervectors for speaker verification, Kharroubi *et al.* [8] proposed in 2001 a way to combine SVM with GMM models, as shown in Fig. 1a. In the enrolment phase, Kharroubi's process is as follows:

1. Acoustic feature vectors X are obtained for the enroling speaker (same as in the Supervectors method).
2. A GMM model λ with M Gaussian Mixtures is obtained from the UBM model $\bar{\lambda}$ via MAP adaptation of means.
3. For each speech feature sequence X, an SVM input vector V_X is initialized to have size $2 * M$ with all values initially set to 0. Each element in the vector, $V_X[m]$, represents each one of the Gaussian mixtures in the λ and $\bar{\lambda}$ models, respectively.
4. For each feature vector x_i in feature sequence X:
   - Find the Gaussian mixture $g_j \epsilon \lambda, \bar{\lambda}$ that maximizes the probability of the vector x_i, *i.e.*:

$$S_{max} = \max_{g_j} \log[\mathcal{P}(x_i|g_j)] \quad (5)$$

   - Accumulate in the appropriate vector position $V_X[m_{max}] = V_X[m_{max}] + S_{max}$ where $m_{max} = g_j$ if $g_j \epsilon \lambda$ and $m_{max} = g_j + M$ if $g_j \epsilon \bar{\lambda}$
5. Normalize the vector V_X by the number of frames in X
6. Train an SVM classifier using the V_X vectors computed for the target speaker as positive examples (class $t = +1$) and a set of impostor speaker vectors V_I (common to all enrolment speakers) as negative examples (class $t = -1$).
7. Save the GMM and UBM models, and the SVM parameters as the speaker's fingerprint

In the test phase, steps $4 - 5$ of the procedure above are carried out in order to create a vector V_Y (using a speaker's GMM and UBM models) from test feature vectors Y, which will be the input to the SVM. The vector V_Y is classified using the SVM in order to determine whether it was produced by the SVM speaker

or not. Note that there is no need here to recompute any GMM model, in contrast with the creation of Supervectors.

In both SVM methods, it is important to use a common UBM model for all speakers enroled in the system and the same set of impostor speakers when training the SVM classifiers.

5. Proposed Minivectors Algorithm

Kharroubi's algorithm offers a few interesting advantages when compared to the Supervectors method. In particular, significantly smaller speaker fingerprints and lower computational needs. However, the Supervectors algorithm proposes an intuitive process to generate the vectors: the Supervectors represent the speaker by the mean vectors, which is a direct mapping of the GMM model for the speaker.

In this section, we propose a modified version of Kharroubi's algorithm (which we call the Minivectors algorithm) that addresses the most important observed limitations of Kharroubi's original method:

- **Lack of normalization:** The resulting V_* vectors (*i.e.* any of the enrolment, impostor or test vectors) have values ranging from zero to some negative log likelihood value (not well defined) after its final normalization. This value is speaker dependent as it is computed from the evaluation of the GMM/UBM models. The use of non-normalized input vectors is suboptimum for training the SVM.
- **Log-likelihoods contradiction:** The best log-likelihood value $\log[\mathcal{P}(x_i|g_j)]$ is added to the appropriate position in the V_* vector. Such vector positions grow negatively in value by accumulating log-likelihood values which, in turn, are less negative the better the evaluated frame is matched by the selected Gaussian.
- **Indirect mapping:** The V_* vectors do not to represent a direct and intuitive mapping from all the information contained in each speaker's GMM model to the vector space, which could be thought of leading to some information loss as only the relative importance of the GMM Gaussians is stored in the vectors used in the SVM training.

Motivated by the limitations above, we propose a modification in the definition of the V_* vectors: instead of adding the best log-likelihood value to the best Gaussian's position in the V_* vectors, we propose adding $+1$ instead. By doing so, the final normalized vector will have values ranging from $[0, 1]$, much more appropriate for SVM training, thus addressing the first limitation. As the same value is added for every frame, there is no ambiguity on how well a Gaussian matches a frame and its contribution to the corresponding position in the vector position. Therefore, solving also the second limitation.

Finally, the impact that the proposed modification has on the third limitation is illustrated in Figure 2, where 3 vectors are shown from the training utterances of two different subjects: the first two vectors belong to the same speaker whereas the last vector is from a different speaker.

In order to reduce noise in the plots, the data of 10 V_* vectors from the same speaker has been averaged, and the final plot has been also low-pass filtered along the frames axis. Note how similar are the resulting plots for the same speaker and how different from the plot of the different speaker. In addition, note how the UBM model $\bar{\lambda}$ hardly gets any frames assigned to. However, when carrying out tests without the UBM part in the vectors, we obtained significantly worse performance. Thus, there seems to be valuable information carried by the UBM part of the vectors.

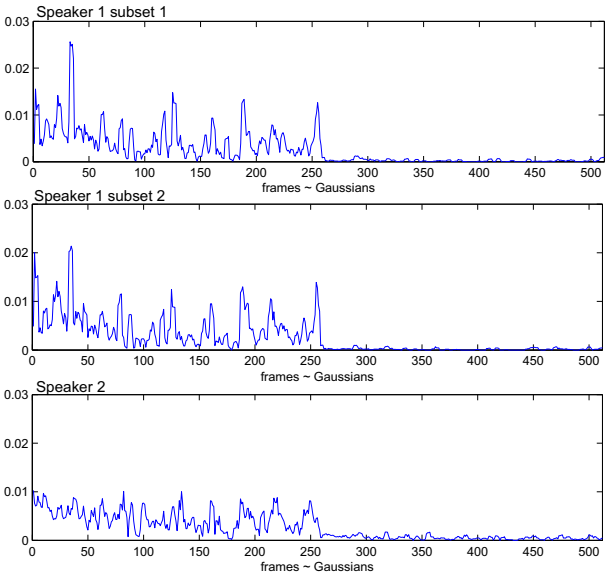

Figure 2: *SVM vectors comparison for* 2 *different speakers.*

6. Experiments

The experiments presented in this paper were conducted using the TelVoice [13] database, which consists of 59 speakers (39 male and 20 female) and 10 phone calls per speaker that took place in different sessions separated by up to one year. Each session consists of 10 spoken items, varying from isolated digits, strings of digits, connected digits, phrases, and free speech. Some items (like personal ID) the same values are spoken by multiple speakers to simulate impostor recordings.

All models are built from a 256 Gaussian UBM model trained from telephone speech gathered from a non-public database. In the algorithm tests, we use the cell phone recordings from the Telvoice database, where speakers say their (right or wrong) cell phone number. These recordings include both isolated digits and strings of digits and were selected to simulate a voice-based pin code authentication test. Adaptation to speaker models (via MAP adaptation) is achieved with 6 utterances of the speaker's cell phone number recorded over two different sessions. These same utterances were used as positive examples in training the SVM models, in addition to 100 impostor speaker utterances – different that the ones in the tests – used as negative examples. The SVM classifiers were trained by means of the *sequential minimal optimization* algorithm proposed by [11].

We ran both true and impostor trials. True trials were collected from the recordings of each speaker's own cell phone number for sessions not included in the training, with a total of 872 tests. Impostor trials used all available recordings of impostors uttering the tested speaker's cell phone number (a total of 873 tests). The features used were MFCC with 26 features extracted every 12ms with a 20ms Hamming windowing.

6.1. Experimental Results

Results of comparison tests between the three systems are shown in Tables 1, 2 and Fig. 3. Fig. 3 depicts the DET plots and DCF points for the three compared algorithms using linear and RBF kernels. Table 1 contains the equal error rate (EER) and the resulting detection cost function (DCF) with $C_{miss} = C_{fa} = 1$ and equal priory probabilities. Note how the proposed algorithms achieves the best performance using either

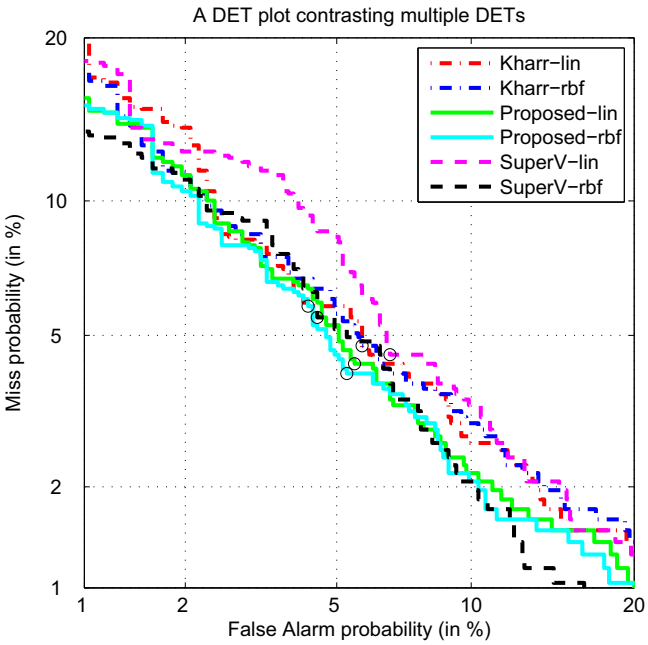

Figure 3: *DET plots for the algorithms presented.*

kernel function. Supervectors outperform Kharroubi's vectors with RBF kernels whereas Kharroubi's vectors achieve better performance than Supervectors with linear kernels.

Table 1: *DCF and EER results for the compared algorithms.*

System	Kernel	DCF	EER
SuperVectors	linear	5.57%	6.09%
Karroubi's vectors	linear	5.06%	5.52%
Proposed vectors	linear	**4.88%**	**5.06%**
SuperVectors	RBF	5.00%	5.17%
Karroubi's vectors	RBF	5.23%	5.40%
Proposed Vectors	RBF	**4.65%**	**4.83%**

Finally, Table 2 compares the three algorithms with respect to their processing times and the speaker model footprint. All ratios are relative to the Supervectors system using linear kernel. The three approaches have very similar computational needs during training. However, the proposed approach is 24% less computationally expensive (in runtime) than the Supervectors in test. Kharroubi's algorithm follows with a 14% reduction in computational needs when compared to the Supervectors. Both Kharroubi's vectors and the proposed algorithm achieve such speedup because: a) they do not need a GMM model adaptation at test time; and b) the vector's dimensionality is much smaller ($2 * 256 = 512$ *vs* $26 * 256 = 6656$ for the Supervectors). Such reduction on the vectors' size has a dramatic impact on the speaker model footprints, which are about 94% smaller than the Supervectors footprints.

Table 2: *Comparative analysis of computational needs and footprint size.*

System	Kernel	Training	test	footprint
SuperVectors	linear	1	1	1
Karroubi's vectors	linear	1.02	0.85	0.060
Proposed vectors	linear	1.00	0.76	0.061
SuperVectors	RBF	0.99	1.06	1.25
Karroubi's vectors	RBF	1.02	0.88	0.065
Proposed Vectors	RBF	1.01	0.76	0.061

7. Conclusions and future work

State-of-the-art Speaker verification systems have achieved accuracy levels which makes them usable in real-life and commercial application. Unfortunately, some systems have high computational requirements and big speaker models footprint which jeopardize such implementation. In this work we compare two speaker verification systems based on GMM-SVM, namely the Supervectors from Campbell *et al.* and Kharroubi's *et al.* system, and propose a modification of Kharroubi's algorithm, which we call Minivectors, that performs similarly to state-of-the-art while keeping a very small footprint and smaller computational requirements at test time. Tests on a speaker verification database indicate the proposed algorithm is 24% faster in runtime and uses speaker models which are 94% smaller in footprint to those needed by the Supervectors algorithm. Immediate future work is to test the algorithm on bigger and standardized databases, like those used for the NIST SRE evaluation.

8. Acknowledgements

We would like to thank Marc Ferras and Nuria Oliver for their much appreciated feedback on the content and during the writing of this paper.

9. References

[1] N. Brummer, L. Burget, J. Cernocky, O. Glembek, F. Grezl, M. Karafiat, D. A. V. Leeuwen, P. Matejka, P. Schwarz, and A. Strasheim, "Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 15, pp. 2072–2084, 2007.

[2] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.

[3] V. Wan, "Speaker verification using support vector machines," Ph.D. dissertation, University of Sheffield, 2003.

[4] S. Fine, J. Navrtil, and R. A. Gopinath, "A hybrid gmm/svm approach to speaker identification," in *in PRoc. ICASSP*, vol. 1, 2001, pp. 417–420.

[5] W. M. Campbell, "A svm/hmm system for speaker recognition," in *in PRoc. ICASSP*, Hong Kong, April 2003, pp. 156–159.

[6] Q. Le and S. Bengio, "Client dependent gmm-svm models for speaker verification," in *ICANN/ICONIP, also in Lecture Notes in Computer Science*, vol. Volume 2714/2003, 2003, p. 181.

[7] W. Campbell, D. Sturim, D. Reynolds, and A. Solomonoff, "Svm based speaker verification using a gmm supervector kernel and nap variability compensation," in *in Proc. ICASSP*, vol. 1, 2006, p. H.

[8] J. Kharroubi, D. Petrovska-Delacrtaz, and G. Chollet, "Combining GMMs with suport vector machines for text-independent speaker verification," in *in Proc. Eurospeech*, 2001, pp. 1761–1764.

[9] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining andKnowledge Discovery*, vol. 2, no. 2, pp. 1–47, 1998.

[10] V. Vapnick, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[11] J. C. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Microsoft Research, MSR-TR-98-14, 1998.

[12] C. H. Lee and J.-L. Gauvain, "Speaker adaptation based on map estimation of hmm parameters," in *in Proc. ICASSP*, vol. 2, 2003, pp. 558–561.

[13] L. Rodrguez-Liares, C. Garca-Mateo, and J. L. Alba-Castro, "On combining classifiers for speaker authentication," *Pattern Recognition Journal*, vol. 36, pp. 347–359, 2003.