Organizing Gaussian mixture models into a tree for scaling up speaker retrieval

Jamal Rougui, Marc Gelgon, D. Aboutajdine, Noureddine Mouaddib, M. Rziza

Organizing Gaussian mixture models into a tree for scaling up speaker retrieval

J.E Rougui [a,b], M. Gelgon [a] , D. Aboutajdine [b], N. Mouaddib [a], M. Rziza [b],

[a] *Ecole polytechnique de l'université de Nantes*
INRIA Atlas project-team, LINA (FRE CNRS 2729),
BP 50609 44306 Nantes Cedex 03, France
Tel : +33 2 40 68 32 57, Fax : +33 2 40 68 32 32

[b] *Groupe Signaux Communications et Multimedia*
Faculté des Sciences Rabat-Agdal
4, Av Ibn Battouta, Rabat, Morocco.

Abstract

Numerous pattern recognition tasks set in the probabilistic framework face the following issue : it is expensive to evaluate the likelihood function for test data, when there are given very many candidate probabilistic models for explaining this data. We consider the application of this general and important problem to speaker recognition for indexing and retrieval purposes in radio archives. More precisely, we propose to reduce complexity at query time, by prior organization of speaker models into a hierarchy. This is very classically done for multi-dimensional vectors, but we propose herein a technique for building a hierarchy of probabilistic models, in the case these models take the form of a Gaussian mixture. From a closed-form approximation of Kullback-Leibler divergence between parent and children, an optimality criterion and an optimization technique are derived, from which we propose an efficient approach for building a tree of models, using clustering techniques (dendrogram-based or k-means-like). The proposed scheme is evaluated on real data.

Key words: Multimedia indexing and retrieval, speaker recognition, Gaussian mixture, tree-based indexing structure

Email address: marc.gelgon@univ-nantes.fr (M. Gelgon).

1 Context and goal

Enhanced content-based indexing, browsing and retrieval in large amounts of audio documents requires prior temporal structuring of this content and labelling of entities extracted, such as assigning the identity of a speaker to a temporal segment (a task also known as "speaker diarization"). A considerable amount of work has been put forward in this field, during the past ten years (Bimbot et al., 2004). In this paper, we focus on the task of text-independent speaker recognition, applied to spoken radio archives. The front-end to the contribution is a classical one. We partition the audio stream into speaker-homogeneous segments by detecting changes in speaker turns. Each speaker is characterised by a probability density estimate of its Mel-cepstral feature vectors (MFCC). This density is modelled as a Gaussian mixture model (GMM), as this provides an effective trade-off between ability to describe complex densities and ability to estimate correctly the parameters of this model from a limited amount of training data, which is especially challenging in the relatively high dimension spaces formed by Mel-cepstral coefficients (between 10 and 40, generally).

Ideally, indexing of an audio stream in carried out incrementally. In such a case, the task of speaker matching is encountered at two stages: when two temporally disconnected segments contain the same speaker and should be labelled as such, and when a user formulates a query. The need for incrementality, i.e. the ability to accommodate for new speakers in the database, or refine already enrolled speaker models as new information is made available, affects a design choice of the scheme: we make use of generative models, rather than techniques that discriminate between speakers.

A typical solution to speaker recognition consists in exploring exhaustively the set of the $S_1, \ldots, S_M$ enrolled speaker models and evaluating the likehood of the query data given each candidate model. The point this paper wishes to address here pertains to scaling up such a system to a large number of speakers, by organizing the set of candidate speaker models in the form of a tree, with a view to obtaining a sub-linear (i.e. $< O(M)$) computational cost at evaluation time. Clearly, the matter is to trade a significant speed up against minimal loss recognition accuracy, relatively to exhaustive search.

There exist alternative work directions for reducing cost: cepstral subspaces (Nishida and Ariki, 1998; Zhou and Hansen, 2002; Upendra et al., 2001), anchor models (Mami and Charlet, 2002; Sturim et al., 2001), that express speakers in a basis of reference speakers, or considering only a few dominant Gaussians in the mixture. These approaches propose speeding up by reducing the evaluation per speaker but remain $O(M)$; our work direction is orthogonal and complements it.

The task relates tightly to the classical issue of indexing structures for multi-dimensional data. The database community has put forward a considerable amount of contributions based on a variety of tree structures (Berrani et al., 2003; Zezula et al., 2006). The particularity of the current problem arises from the nature of the entities to index, namely probability distributions, for which classical indexing structures are inappropriate. Extending such structures to handle probabilistic representations is one of the most important current issues, since it has a major impact on the ability to scale up applications to large amounts of data.

The remainder of this paper is organized as follows. Section 2 provides the following preliminary material: given a set of sibling speaker models and their parent, how do we define the representativity of the parent with respect to its children ? Then, how do we build a parent that possesses an optimal representativity ? Section 3 exploits proposals made above to define several alternative techniques for grouping similar speakers and organizing a set of models into a tree. Section 4 reports experimental results, while we provide concluding remarks in section 5.

2 Child-to-parent relation

Let us consider a set of M enrolled speaker models, i.e. M Gaussian mixture models. The manner by which the parameters of these models are estimated is not central in the present proposal: it may be through conventional EM-based estimation (Bishop, 1995) or, more effectively from limited training data, a point estimate from Bayesian learning with universal background model as a prior density (Ben et al., 2004). It suffices to say here that model k is expressed as:

$$S_k(x) = \sum_{i=1}^{m_k} w_k^i N_k^i(x) \tag{1}$$

where $N_k^i(x)$ is a Gaussian component which mean is μ_k^i and covariance Σ_k^i, while w_k^i are scalar weights.

Let us assume recognition is based on maximum likelihood of the query data D, over the set of M candidate models (the scheme extends directly to maximum a posteriori). Exhaustive maximum likelihood search forms the baseline technique, against which we propose improvement.

We aim at forming a hierarchy of speaker models by grouping the M models bottom-up. To justify the criterion proposed below for parent-child similarity, let us consider the simplest tree, where two speakers S_1 and S_2 are represented by a single father S_{12}, which also takes the form of a Gaussian mixture model. This extends directly to an arbitrary number of children.

The cost reduction at query time, when the tree is explored root-to-leaf, is obtained by computing a single value $p(D|S_{12})$ instead of both $p(D|S_1)$ and $p(D|S_2)$. Consequently, S_{12} should thus be designed so that $p(D|S_{12})$ is as close as possible to both $p(D|S_1)$ and $p(D|S_2)$, in order to keep classification error as close as possible to that of exhaustive search. The number m_{12} of Gaussian components in S_{12} should also be clearly smaller than $m_1 + m_2$ to ensure computational cost reduction of evaluation.

The next two subsections respectively define (sec. 2.1) an optimality criterion for a parent, given its children, and (sec. 2.2) expose how we optimise this criterion to actually determine the parent model from given children.

2.1 Defining a low-cost, minimal KL loss measure between parent and child

The expected loss in log-likelihood caused by approximating both S_1 and S_2 by S_{12} is expressed as:

$$E_{S_k}\left[\ ln\ p(D|S_k)\ \right]\ -\ E_{S_k}\left[\ ln\ p(D|S_{12})\ \right], \text{where}\quad k = 1, 2 \tag{2}$$

Assuming all candidates are equally probable, the optimal mixture $\widehat{S_{12}}$ minimizing this loss is thus defined as:

$$\widehat{S_{12}}\ =\ \arg\min_{\mathcal{S}}\ \left[\ -\int S_1(x)\ ln\ S_{12}(x)\ dx - \int S_2(x)\ ln\ S_{12}(x)\ dx\right] \tag{3}$$

where integrals span the feature space and $\mathcal{S}$ is the search space, discussed below. This corresponds in fact to minimising the Kullback-Leibler divergence $KL(S_{1+2}\|S_{12})$ (Bishop, 1995), where $S_{1+2}(x)$ designates $\frac{1}{2}(S_1(x)+S_2(x))$, i.e.:

$$\widehat{S_{12}}\ =\ \arg\min_{\mathcal{S}}\ \left[\ -\int S_{1+2}(x)\ ln\ \frac{S_{12}(x)}{S_{1+2}(x)}\ dx\ \right] \tag{4}$$

A major issue for the practical computation of (4) is the lack of closed form for this divergence, in the case of Gaussian mixtures. To avoid expensive Monte-Carlo evaluation (Chen et al., 2005), we propose a closed form through the following approximation. Linearity of the integral applied to (3) provides:

$$\widehat{S_{12}}\ =\ \arg\min_{\mathcal{S}}\left[-\ \sum_{i}^{m_1+m_2} w_{1+2}^i \int N_{1+2}^i(x)\ ln\ S_{12}(x)\ dx\right] \tag{5}$$

In each term of the sum in (5), we approximate the mixture S_{12} by only one of its Gaussian components, selected as the best approximation to N_{1+2}^i, in

the KL sense. This leads to the following similarity measure, denoted below KL_m for KL_{modified}, between a reference model S_{1+2}, which contains too many components to be efficient, and its approximation S_{12}:

$$\begin{aligned} \widehat{S_{12}} &= \arg\min_{\mathcal{S}} \ [KL_m(S_{1+2}\|S_{12})] \\ &= \arg\min_{\mathcal{S}} \ \left[\sum_{i=1}^{m_1+m_2} w_{1+2}^i \ \min_{j=1}^{m_{12}} \ KL(N_{1+2}^i \| N_{12}^j) \right] \end{aligned} \tag{6}$$

The following expression is used for comparing a child model and its parent model in the tree:

$$KL_m(S_k\|S_{12}) = \sum_{i=1}^{m_k} w_k^i \ \min_{j=1}^{m_{12}} \ KL(N_k^i \| N_{12}^j), \ k = 1, 2 \tag{7}$$

This similarity measure can easily be computed at low-cost, since the Kullback divergence between two Gaussians, which parameters are (μ_1, Σ_1) and (μ_2, Σ_2), benefits from the following closed-form expression:

$$\frac{1}{2}(log\frac{|\Sigma_2|}{|\Sigma_1|} + Tr(\Sigma_2^{-1}\Sigma_1) + (\mu_1 - \mu_2)^T\Sigma_2^{-1}(\mu_1 - \mu_2) - \delta) \tag{8}$$

where δ is the dimension of the feature space. It may be demonstrated (Goldberger and Roweis (2004)) that optimising (6) amounts to finding an optimal discrete mapping π between the $m_1 + m_2$ components of S_1 and the m_{12} $(< m_1 + m_2)$ components of S_{12}. This involves reducing the number of components in the mixture S_{1+2} to build S_{12}, while minimizing density distorsion, in the KL_m sense. The search space $\mathcal{S}$ thus consists in all ways of grouping the $m_1 + m_2$ components into m_{12} groups.

2.2 Search for the optimal parent mixture

The search space cannot in practice be searched exhaustively if there are more than 10 components, which we typically encounter. Hence, we optimise locally criterion (7) with an iterative scheme detailed in Algorithm 1 below. It is adapted from a technique proposed by Goldberger and Roweis (2004), in the context of hierarchical clustering of Gaussians (rather than Gaussian mixtures). The procedure bears analogy with the classical k-means algorithm, in that it operates local optimization by alternatively assigning elements to groups and re-computing group representatives. In our context, the elements are the components of S_{1+2} and the representatives those of S_{12}.

As often done with k-means, the initial assignments π^0 from which local optimisation proceeds could be drawn randomly. Our context suggests a more

effective initialisation criteria in our context: since generally, Gaussian components coming from the same mixture are not redundant, we draw π^0 at random with the constraint that components arising from the same mixture are not initially grouped. The iterative scheme may still regroup them later, if the data drives it that way. As it is practically desirable to draw multiple starting points to retain the best local optimum, this strategy improves sampling of the search space.

3 Grouping speaker models

This section applies the child-to-parent relation criteria and optimization technique presented in the previous section to three ways of organizing speaker models into a search tree. Practically, the scope of this paper is restricted to a single intermediate layer between the root and the leaves, and may be viewed as clustering of speaker models.

3.1 Dendrogram-based grouping

We first present a transposition of the most classical data clustering to our problem, namely bottom-up hierarchical clustering, where each leaf is a Gaussian mixture model (see fig. 1):

1. a $M \times M$ similarity matrix is computed between models. Similarity between two mixtures S_1 and S_2 is computed as :

$$KL_m(S_1||S_2) + KL_m(S_2||S_1) \tag{15}$$

2. the two most similar models are grouped and summarized as one (here, not reducing S_{1+2} to S_{12}, to keep a richer representation), and so on until there remain only two nodes. The similarity matrix is updated after each merge operation.
3. the dendrogram-tree obtained is cut (dashed line in fig. 1) so that the number of nodes just above it is close to $log_2(M)$. These nodes inheritate from all their (grand)children, and the corresponding mixture model are determined by optimizing criterion (6). Doing so, a tree with a variable of children is formed, which we use for searching.

As usually with hierarchical clustering, this technique is not incremental and its complexity does not scale up well to large amounts of speakers (it does at evaluation time, hence its interest, but not at tree construction time). Since similarity between models is computed by means of KL_m, it only resorts to

Algorithm 1 Iterative optimisation algorithm for estimating the reduced model S_{12} (criterion (7))

Start from a constrained random initialisation $\hat{\pi}^0$ (or given, if available)
$it = 0$
repeat
1. Re-fit mixture S_{12}:
given the current component clustering $\hat{\pi}^{it}$, set initially or computed at the previous iteration, update mixture model parameters as follows:

$$\widehat{S_{12}}^{it} = \arg \min_{S_{12} \in \mathcal{S}_{m_{12}}} KL_m(S_{1+2}, S_{12}, \hat{\pi}^{it}) \tag{9}$$

where $\mathcal{S}_{m_{12}}$ is the space of all mixture with m_{12} components that may be formed by grouping components of M_c. This re-estimation in fact amounts to updating each component of S_{12} as follows. For component j, algebra leads to the following expressions:

$$\hat{w}_{12}^j = \sum_{i \in \pi^{-1}(j)} w_{1+2}^i \tag{10}$$

$$\hat{\mu}_{12}^j = \frac{\sum_{i \in \pi^{-1}(j)} w_{1+2}^i \mu_{1+2}^i}{\hat{w}_{12}^j} \tag{11}$$

$$\hat{\Sigma}_{12}^j = \frac{\sum_{i \in \pi^{-1}(j)} w_{1+2}^i (\Sigma_{1+2}^i + (\mu_{1+2}^i - \hat{\mu}_{12}^j)(\mu_{1+2}^i - \hat{\mu}_r^j)^T)}{\hat{w}_{12}^j} \tag{12}$$

where $\pi^{-1}(j)$ is a light notation for $\hat{\pi}^{-1,it}(j)$, the set of S_{1+2} that project onto component j in S_{12}.
2. Grouping components:
for mixture $\widehat{S}_{12}^{it}$ obtained in Step 1, we seek the mapping π^{it+1}, defined from $\{1, \ldots, m_1 + m_2\}$ into $\{1, \ldots, m_{12}\}$, which best groups components of S_{1+2} to build components of $\widehat{S}_{12}^{it}$, in the following sense :

$$\hat{\pi}^{it+1} = \arg \min_{\pi} KL_m(S_{1+2}, \widehat{S_{12}}, \pi) \tag{13}$$

In other words, each component i of S_{1+2} projects onto the closest component j of $\widehat{S}_{12}^{it}$, according to their Kullback divergence ((14) below). In this phase, we resort to exhaustive search among 'source' components, which has a low-cost, thanks to the availability of (8).

$$\pi^{it+1}(i) = \arg \min_j KL(N_{1+2}^i || N_{12}^j) \tag{14}$$

3.*it=it+1*
until convergence (i.e. $\pi^{it+1} = \pi^{it}$)
compute

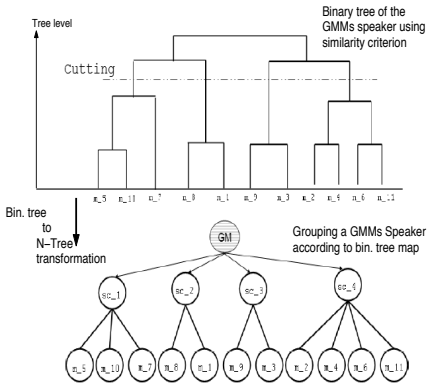

Fig. 1. Hierarchical clustering applied to a set of Gaussian mixture models. (top) A dendrogram is first built, then (bottom) cut to determine nodes (i.e. GMM) forming the intermediate layer

model parameters rather than data, and is hence practically fast and usable for a moderate number of speakers.

3.2 Iterative grouping

As an alternative to hierarchical clustering, we propose an iterative scheme analogous to the k-means procedure, for which data elements are mixture models. It is detailed in fig.2. The criterion to optimize here generalizes the simple parent-child relation optimality defined in section 2, over several parents :

$$\sum_{S_p \in parents} \quad \sum_{S_c \in \text{children of } S_p} KL_m(S_p \| S_c) \tag{16}$$

Algorithm 2 Iterative optimisation of parent model parameters

Start from random grouping of speaker models
repeat
1. Re-fit mixture of each parent using Algorithm 1.
This step involves itself a k-means-type algorithm that operates on Gaussian components.
(rather than on Gaussian mixtures as the present Algorithm 2 does).
2. Re-assign each child of the complete set to the most similar parent (in the $KL_m(S_{parent} \| S_{child})$ sense)
until convergence

An essential good property is that assignments of speaker models to groups may easily be questioned, in contrast to dendrogram-based grouping. Consequently, the iterative approach is amenable to incremental processing, i.e. it can accomodate new speaker models at leaves and update the intermediate

layer by re-optimizing (16) locally and, if required, it would be quite easy to extend the present scheme to allow the number of intermediate nodes to evolve over time.

3.3 Exploiting the approximation error in the tree structure

Let us consider a tree obtained by either of the approaches presented in the two previous subsections. Let S_p denote a parent node and $\{S_1, S_2, \dots\}$ its children.

The main point made in the paper so far is as follows : we proposed a technique for building S_p so that it explicitely tries to approach all of children models with respect to expected log-likelihood of data to be classified, i.e. it ensures that, for any child $log\ p(D|S_p) \approx log\ p(D|S_c)$. Computation savings come from that $log\ p(D|S_p)$ is the only likelihood that needs to be computed in the classification phase. This seems to us better founded than alternative approaches for fast processing of numerous speaker models, such as anchor models, where Euclidian distances are computed between likelihood vectors.

The further point we make here is that the resulting approximation error may not only be minimized, but also taken into account in order to search the tree more finely, in the classification phase, yet at approximately the same computational cost. Rather than replacing, for all children k, $log\ p(D|S_k)$ by $log\ p(D|S_p)$, the likelihood associated to each child node may be approximated as:

$$\underset{\text{child log-likelihood}}{log\ \tilde{p}(D|S_k)} \approx \underset{\text{parent log-likelihood}}{log\ p(D|S_p)} + \underset{\text{independent of data to classify}}{KL(S_p||S_k)}, k = 1, 2, \dots \quad (17)$$

The main point here is that $KL(S_p||S_k)$ can be pre-computed and is independent of the data to be classified. We advocate the use of the unscented transform (Julier, 1996) for the practical computation of $KL(S_p||S_k)$, as it is more accurate than KL_m used above. This approximation between Gaussian mixtures does not only consider the closest Gaussian, but summarizes each of them by concise statistics, leading to an overall light yet accurate computation. As a side remark, the properties of the unscented transform precluded its use in the model grouping phase.

Because likelihood approximations that are now individual, per child, this second point opens new possibilities for exploring the tree of models, for instance:

(1) searching exhaustively the set of children, by using $log\ \tilde{p}(D|S_k)$, or,
(2) by pre-computing the maximum and minimum error between a parent node and its children :

Exhaustive search	Recognition accuracy		
Query duration (sec)	5	10	15
ML	100%		
KL_m	75%	82.5%	85%

Table 1
Comparing performance of querying exhaustively the collection of speakers, based on maximum likelihood classification (ML line) or computation of KL_m between query and each candidate model (KL_m line). This is examined for 5,10 and 15 seconds queries.

$$Min_{ERR} = \min_k KL(S_p||S_k), \tag{18}$$

the corresponding cluster of speakers is characterised as having, with high probability, its log-likelihood within $[log\ p(D|S_p) + Min_{ERR}, log\ p(D|S_p) + Max_{ERR}]$, leading to again several possible search schemes.

4 Experimental results

All experiments reported below are applied to RealAudio streaming radio broadcast data, in French language. The 13 first MFCC features vectors and their temporal derivates are used. Temporal segmentation of the stream into segments is carried out with the BIC criterion (a classical approximation (Schwarz, 1978) to Bayesian hypothesis testing) over a 4 second sliding window. Individual speaker models are learned using Bayesian adaptation (Bimbot et al., 2004). The stream contains ordinary news programmes, including occasional short jingles than can quite reliably be removed, thanks to their acoustic properties in MFCC space, leaving essentially clean speech sections.

First experiments involve 20 speakers. Accuracy at query phase is evaluated as follows : 40 samples from the 20 speakers are provided for classification (2 per speaker).

We first report an experiment conducting exhaustive search, where query-to-model fitting is conducted by either using definition 15 or maximum likelihood (see table1). While maximum likelihood performs perfectly, KL_m far is less effective (through much faster)

4.1 Results for dendrogram-based hierarchical clustering

Two alternative criteria are compared for measuring similarity between speakers :

- the cross-likelihood, which requires resorting to the feature vectors, which is undesirable from computational cost viewpoint, but should be reliable,
- definition (15), a symmetric version of KL_m.

The trees obtained in these cases are shown in fig. 4.1. It appears that the tree build from KL_m similarity is well-balanced, actually better than the one based on cross-likelihood.

When exploring the tree root-to-leaf, query-to-model comparison is evaluated in two cases : (i) likelihood of the query data, given the model, or (ii) symmetric KL_m as in 15. In both cases, the tree was built using definition 15. Results are compared for 2 query lengths (5 and 10 seconds). Results are presened in tab.2. As in the previous experiment, the use of KL_m for querying, instead of ML, implies severe degradation. However, using ML, accuracy remains very satisfactory, and the approach remains beneficial in the sense that : (i) exploration is done through the tree rather than exhaustively, (ii) the tree is built using KL_m, thus quite fast.

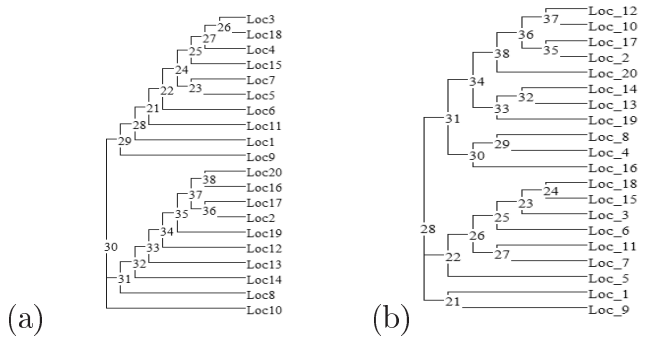

Fig. 2. (a) Binary tree generated using the cross-likelihood scoring matrix between 20 speaker Gaussian mixtures. (b) Binary tree generated using the symmetric KL_m between pairs of speaker models with an incremental perspective.

4.2 Results for a hierarchy build using iterative grouping of mixture models

Table 3 shows recognition accuracy obtained in the same conditions as previous, but the tree is built using the iterative scheme (Algorithm 2) rather than the dendrogram-based approach. The quality of the results are very similar as in the previous approach, which is encouraging, since this iterative approach

Dendrogram-based hierarchy	ML		KL	
Query duration	5	10	5	10
26 Gaussians	92.5%	95%	47.5%	40%
16 Gaussians	95%	95%	50%	45%

Table 2
Recognition accuracy in the case speakers are organized in the tree obtained at fig.4.1b (after cut). Two mixture complexities are considered.

Iterative grouping hierarchy	ML		KL	
Query duration	5	10	5	10
26 Gaussians	90%	92.5%	45%	55%
16 Gaussians	92.5%	90%	57.5%	60%

Table 3
Recognition accuracy in the case speakers are organised in the tree obtained by Algorithm 2. Two mixture complexities are considered.

is far more flexible than the dendrogram-based approach.

5 Conclusion

In this paper, we addressed the problem of scaling up speaker recognition to a large number of speakers, by organizing the set of speaker models into a search tree. The child-to-parent similarity may be measured and optimized iteratively, using an approximation of KL-divergence, that leads to a low-cost, tractable form. We define and evaluate two ways (dendrogram and iterative grouping) in which this similarity can be exploited, leading to results that loose little reliability with respect to exhaustive search, and offer promising perspectives for speed up. The iterative model grouping procedure is particularly interesting, as is very flexible for incremental processing of the data. There remain to generalise the proposal to more than two levels. Also, the estimated KL divergence between parents and children, which is computed anyway, could provide richer knowledge of the likelihood of data, given children, than is currently done by simply considering the likelihood of data, given the parent.

Acknowledgements

The authors are grateful to the French foreign office for funding this work, as part of the French-Moroccan research network on multimedia (Réseau STIC RTIM).

References

Ben, M., Gravier, G., Bimbot, F., 2004. Enhancing the robustness of bayesian methods for text-independent automatic speaker verification. In: Odyssey'04 Speaker and Language Recognition Workshop. pp. 34–39.

Berrani, S., Amsaleg, L., Gros, P., Nov. 2003. Robust content-based image searches for copyright protection. In: Proc. of ACM workshop on Multimedia databases. New Orleans, USA, pp. 70–77.

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

Bishop, C., 1995. Neural networks for Pattern Recognition. Oxford University Press.

Chen, M., Shao, M., Ibrahim, J., 2005. Monte Carlo Methods in Bayesian Computation. Springer.

Goldberger, J., Roweis, S., 2004. Hierarchical clustering of a mixture model. In: Proc. of Neural Information Processing Systems (NIPS'2004). pp. 505–512.

Julier, S., Nov. 1996. A general method for approximating a non linear transformation of probability distributions. Tech. rep., Oxford university, Dpt of Engineering Science.

Mami, Y., Charlet, D., Septembre 2002. Speaker identification by location in an optimal space of anchor models. In: International Conferences on Spoken Language Processing (ICSLP '02). Denver, Colorado, USA, pp. 1333–1336.

Nishida, M., Ariki, Y., 1998. Real time speaker indexing based on subspace method - application to tv news articles and debate. In: International Conference on Spoken Language Processing (ICSLP'1998). Sydney, Australia, pp. 1347–1350.

Schwarz, G., 1978. Estimation the dimension of a model. Annals of statistics 6, 461–464.

Sturim, D., Reynolds, D., Singer, D., Campbell, E., 2001. Speaker indexing in large audio databases using anchor models. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01). Salt Lake City, Utah, pp. 429–432.

Upendra, V., Navratil, J., Ramaswamy, G. N., Maes, S., May 2001. Very large

population text-independant speaker identification using transformation enhanced multi-grained models. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01). Salt Lake City, Utah, pp. 461–464.

Zezula, P., Amato, G., Dohnal, V., Batko, M., 2006. Similarity Search - The Metric Space Approach. Advances in Database Systems, Vol. 32, 2006, XVIII,. Springer.

Zhou, B., Hansen, J., 2002. Improved structural maximum likelihood eigenspace mapping for rapid speaker adaptation. In: International Conference on Spoken Language Processing (ICSLP'2002). Denver, Colorado, pp. 554–564.