Low-cost Distributed Learning of a Gaussian Mixture Model for Multimedia Content-based Indexing on a Peer-to-peer Network

A. Nikseresht and M. Gelgon
LINA FRE 2729 CNRS / ATLAS group, INRIA
Ecole polytechnique de l'université de Nantes
La Chantrerie, rue C.Pauc, BP 50609
44306 Nantes cedex 3 - France
email: firstname.lastname@univ-nantes.fr

Abstract
This work deals with estimation of a probability density, which is a common issue in multimedia pattern recognition. The originality comes from its computation in a distributed manner, since the study is motivated by the perspective of a multimedia indexing and retrieval peer-to-peer system over the internet. In a decentralized fashion, algorithms and data from various contributors would cooperate towards a collective statistical learning.

In this setting, aggregation of probabilistic Gaussian mixture models of the same class, but estimated on several nodes on different data sets, is a typical need, which we address herein. The proposed approach for fusion only requires moderate computation at each node and little data to transit between nodes. Both properties are obtained by aggregating models via their (few) parameters, rather than via multimedia data itself. Mixture models are in fact concatenated, then reduced to a suitable number of Gaussian components. An approximation on Kullback divergence leads to an iterative scheme for estimating this aggregated model. We provide experimental results on a speaker recognition task with real data, in a gossip propagation setting.

Categories and Subject Descriptors
H.3 [Information storage & retrieval]: Content analysis and indexing; I.5 [Pattern recognition]: Implementation—Special architecture

General Terms
Algorithms

Keywords
Probability density estimation, distributed computing, multimedia indexing and retrieval

1. INTRODUCTION
Advances in multimedia content-based retrieval are largely related to progress in pattern recognition techniques, on one side, and the design of media-specific observations from the data, on the other side. In this paper, we consider that elements of interest are classes, i.e. exhibit a variability that should be modelled, such as faces, speakers, spatio-temporal texture/video event [6]). This variability is to be learned, in a supervised manner. This is quite different from the retrieval of ”similar images”, which is the multimedia indexing task that has so far attracted most interest in the distributed setting [3].

We expect that services exploiting multimedia data on wide-scale networks has major perspectives of improvement, should they acquire this ”pattern recognition” dimension. Conversely, pattern recognition on multimedia data requires large amounts of training data, that we can hope to find on the internet. Work at the intersection of these fields includes, for instance, efficiency and data placement related to search of similar images [13] and collective learning from text data [17].

The present work takes the following viewpoint: let us consider that machine learning tools (for supervised learning) are made available online by various human contributors on nodes of a peer-to-peer network [12], e.g. as web services. Each service can receive multimedia data and then send the corresponding class identifier (e.g. name of speaker), and/or request, or share training data (i.e. (data,class identifier) pairs) for supervised learning. Multimedia data is placed on nodes and part of it is assumed to be labelled with its class, i.e. can contribute to supervised learning. In the example case of computer vision, two trends favour this perspective: on one side, joint text/image analysis, fed by a massive resource of web pages or video OCR [15] and, on the other side, recent advances in semi-supervised learning [16, 18], which enables learning a class from instances supplied in clutter (e.g. a face within a background). Overall, automatically building large amounts of distributed, partly labeled, multimedia data can be envisaged in the near future.

A peer-to-peer organization of participant nodes seems relevant, since:

- resources are dynamic: data and learning/classification services can join or leave the network at any time;
- a node is both client and server: it could learn from
another node and/or provide its knowledge to other nodes;

- resources are aggregated: ideally, the quality of the global service is due to its collective aspect;

- the system is decentralized: each contributor can supply data or learning tools, without any central administration.

This scenario naturally opens problems in a wide range of expertise, beyond the very point we address below. However, it justifies the design of scheme described below by a stimulating vision.

Our goal is statistical supervised learning of a class. Because the set of classes is not known a priori in our scenario (e.g., new speakers may be introduced), we consider generative rather than discriminant statistical models. More precisely, we focus on learning a probability density in a feature space, which we assume common to all contributors in the network, for the class to be learned. There are in fact cases where this strong hypothesis is reasonable, such as the widely used mel-cepstral acoustic features for the speaker recognition task, which we apply our proposal to. Furthermore, we focus on the case where all densities are Gaussian mixtures. This model form is indeed ubiquitous in modelling of multimedia data, due to its numerous good properties (good modelling properties vs. good behaviour in high dimension space, clean procedures for estimation and model complexity determination). They have for instance been used to model audio classes audio [14], images [9] or motion-based spatio-temporal events in videos [6].

In this paper, nodes in the network are willing to cooperate and share their knowledge. In practice, this means merging communicating this knowledge, merging it and dispatch it back. We propose a technique to achieve this in the important case of Gaussian mixtures. The proposal implies sending only a small amount of information on the network and low computation cost on nodes, leading to an overall very fast scheme. In fact, sharing of class representation is carried out through model parameters, while we do not require transmission of, nor computation on, the generally large amount of multimedia data (or feature vectors that represent it). Let us underline that the distributed learning phase, which we address in this paper, and the querying phase, can fully overlap, as mixture reduction keeps the class representation directly ready for query evaluation.

Finally, let us note that transmission of the data could be avoided in a much simpler way, by transmitting model parameters and sampling from them, but this implies considerable computation in large dimension spaces.

Fig. 1 provides a small-scale example. In the remainder of this paper, we detail the proposed approach for mixture aggregation (section 2) and provide a first validation in the example of a speaker recognition task (section 3). Section 4 provides concluding remarks.

2. LIGHT AGGREGATION OF MIXTURE MODELS

Let two nodes each carry different probabilistic Gaussian mixture models, denoted \( M_1(x) \) and \( M_2(x) \), associated to the same multimedia entity and hence hidden density \( p(x) \).

\[
M_k(x) = \sum_{i=1}^{m_k} w_k^i N_k^i(x), \quad k = 1, 2 \tag{1}
\]

where \( N_k^i(x) \) is a Gaussian component which mean is \( \mu_k^i \) and covariance \( \Sigma_k^i \) and the \( w_k^i \) are scalar weights. Model \( M_k \) is estimated on a data set of size \( n_k \) located on node \( k \).

\[
p(x) \text{ can be estimated by concatenating incoming mixtures as follows:}
\]

\[
M_s(x) = \frac{1}{n_1 + n_2} \left( n_1 \sum_{i=1}^{m_1} w_1^i N_1^i(x) + n_2 \sum_{i=1}^{m_2} w_2^i N_2^i(x) \right) \tag{2}
\]

However, the \( m_1 + m_2 \) components in \( M_s \) are generally largely redundant, which implies a useless increase in evaluation cost of (e.g.) likelihoods for this density at query time, when merges are chained (such as in the example gossip propagation developed in the experimental section of the paper). Consequently, scaling up the scheme requires transforming \( M_s \) into a reduced mixture \( M_r \) that preserves reasonably well the density while only having the necessary number of components for this. Doing so, the order of magnitude of the number of components is kept constant through propagation, although in detail it fluctuates to fit the complexity of the density.

2.1 Definition and optimization of the similarity between \( M_s \) and \( M_r \)

We seek a model \( M_r \) which maximizes the expected log-likelihood of data \( D \) assumed to be drawn from \( M_s \) (3). It can be shown [4] that this amounts to minimizing the Kullback-Leibler divergence \( KL(M_s \| M_r) \), defined by (4), which, in short, measures the loss of information due to the
approximation of $M_r$ by $M_r$.

$$M_r = \arg \max_{M_r} E_{M_r} \left[ \ln p(D|M_r) \right]$$  

$$M_r = \arg \min \left[ -\int M_r(x) \ln \frac{M_r(x)}{M(x)} \, dx \right]$$

A major issue is the lack of closed form for this divergence, in the case of Gaussian mixtures. We bypass this issue by resorting to a variation of this divergence, recently proposed by [7] in a different setting, consisting in minimizing the following similarity measure (5):

$$d(M_r, M_r) = \sum_{i=1}^{m_1+m_2} w_i^c \min_{j=1}^{m^c_r} KL(N^r_i\|N^j_r)$$

where $N^j_r$ (resp. $N^i_r$) is the $i^{th}$ component of $M_r$ (resp. of $M_r$).

This similarity measure exhibits two good properties:

- it can easily be computed at low-cost, since the Kullback divergence between two Gaussians, which parameters are $(\mu_1, \Sigma_1)$ and $(\mu_2, \Sigma_2)$, benefits from the following closed-form expression:

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

where $\delta$ is the dimension of the feature space.

- it preserves the following theoretical benefit of the original Kullback divergence: as the amount of data grows, minimizing $d(M_r, M_r)$ is equivalent to maximizing the expectation of the data log-likelihood (data being drawn from $M_r$), also assuming that all data drawn from a component of $M_r$ is assigned, in the mixture reduction process, to the same component in $M_r$.

Hence, following [7], we optimize locally criterion (5) with an iterative scheme, detailed in Algorithm 1 below, which could very roughly compared to a k-means algorithm applied to the components of the mixture. As, in multimedia applications, the number of Gaussian components is commonly of the order of 10, rather than several thousands for feature vectors, the (hard) convergence is quickly obtained.

### 2.2 Determining the number of components

An important point in the proposed approach is the determination of the number of Gaussian components in the reduced model $M_r$. The seminal study reported in [1] showed that estimating the Kullback divergence is in fact affected by a bias that grows with the number of parameters to be estimated, i.e. with the number of components. It also supplies a first-order approximation of this correction, which we apply here to the definition of $d(M_r, M_r)$, which hence becomes:

$$d(M_r, M_r) = \sum_{i=1}^{m_1+m_2} w_i \min_{j=1}^{m^c_r} KL(N^r_i\|N^j_r)$$

where $\nu_{M_r}$ is the number of independent parameters in the mixture. Our experimental results (not reported here) back the application of this approximation: the number of components obtained in practice appears very similar to that obtained by usual (AIC,BIC) model selection criteria on the model computed directly on all the data (i.e. discarding the distributed aspect of the learning process). We evaluate exhaustively from 1 to $m_1 + m_2$ the performance of each possible number of components in $M_r$, in independent trials. A faster alternative would be to compute this recursively downwards from $m_1 + m_2$ to 1, but experimental results suggest this can excessively prune the search space at early stages.

### 2.3 Initialization of the iterative optimization, incrementality of the scheme

The iterative technique proposed assumed some initialization for assigning the components of $M_r$ onto those of $M_r$. If no prior knowledge is available, random initialization can be used. Let us examine the case where at least one of the models being merged is itself the result of merging mixtures (‘prior mixtures’). There are in fact two approaches:

1. the prior mixtures could be ‘forgotten’, and we proceed as described above;

2. alternatively, the result of a merge operation can be a richer representation, composed of all prior components and their assignments to the reduced mixture. These assignments can be exploited to initialize, in part, assignments in fusions occurring downstream, hence reducing computational cost w.r.t random assignments, while generally improving the probabilistic modelling, since the overall process has more memory. In practice, propagation of components should be limited to a few successive fusions, to keep computational complexity constant.

To summarize, this paragraph sketches some possible benefits of the incremental capability of the iterative mixture reduction scheme.

### 3. EXPERIMENTAL RESULTS

We report experimental results obtained on real data. The example of distributed speaker recognition is taken throughout, but the technique directly applies to a wide range of multimedia classes. We first focus on a fusion, i.e. at local scale (section 3.1), then observe global performance, in the context of gossip-based mixture propagation (section 3.2).

#### 3.1 Experiments on a single fusion

In the first experiment, three nodes each have learnt a probability density for speaker ‘A’ in a common 13-dimension mel-cepstral feature space. The three corresponding mixtures merge simultaneously into a single mixture (i.e. (1) generalizes to merging more than two mixtures). Each node was provided with different training data from the same speaker and the duration of audio recordings was between 7 to 16 seconds. i.e. rather short for training. Independently of the technical contribution of this paper, each node should provide its mixture estimated from its data. To this end, the Expectation-Maximization local optimization algorithm is employed, but with some enhancement to limit local minima and ensure the three mixtures are reliable inputs. Each mixture also autonomously and automatically determines its number of components (in practice, using the common BIC criterion). All covariance matrices in the mixture are full (rather than spherical or diagonal). Let us point out that this first phase only provides the mixtures to the main contribution of the paper, they could be generated in another way.
mixture reduction, g. 2(c) provides numerical evidence to 4 components. To evaluate the effectiveness of the density after mixture reduction. A reference density is divergence is measured by a Monte-Carlo experiment with the three incoming densities, the concatenated density and direct mixtures are superimposed to the data, and the two latter are clearly very close.

We report a second experiment, applied to a different speaker. It again involves three nodes but, similarly to g. 1 and in contrast to the previous experiment, two nodes are merged, and then a third node is merged to their reduced mixture to form a final reduced mixture. The experiment is conducted in a 2-dimension space, for the sake of clarity of fig. 3(a), which shows all original feature vectors (i.e. the training data). Its purpose is more an illustration value than a demonstration of large scale effectiveness. The centres of the incoming mixtures, as well as the centres of the reduced and direct mixtures are superimposed to the data, and the two latter are clearly very close.

3.2 Application to combination and exchange by gossip information spreading

To evaluate the performance of the proposed technique in the distributed context described in the introduction of the paper, a mechanism is required for propagating mixture representations from the edge of the network (nodes that
own data) to all nodes. We employ here a simple gossip (or epidemic) spreading procedure, described in Algorithm 2. Despite its simplicity, it enables fast and robust propagation in an asynchronous, decentralized manner that fits well the peer-to-peer viewpoint. Our example networks are fully connected and hence each node owns data, but this is by no means a requirement.

A network of 13 nodes is used in the first experiment. Each node owns different data from the same speaker and independently estimates its own model.

To evaluate the capability of mixtures on the network to model data \( D \) from the class of interest (here, a speaker), the classical marginal likelihood is selected [11]. Data \( D \) used for this purpose is again different data, but also coming from the same speaker. The practical computation of the marginal likelihood of the data is carried out with the BIC criterion

\[
BIC(D|M) = - \log p(D|\hat{\theta}) + \frac{\nu \log(n)}{2}
\]

(13)

where mixture \( M \) is defined by a parameter vector \( \theta \), \( p(D|\theta) \) is the likelihood of the data for this model, \( \nu \) is the number of independent parameters in the mixture and \( \log(n) \) the size of the data set (the data set does not need to propagate in the network, but its size should propagate and cumulate).

Fig. 4(a) depicts, after each gossip cycle and on each node, the evolution of criterion (13), which should be minimized. The following observations can be made:

1. the process stabilizes around a “collective model”. Convergence cannot be established, as illustrated in the zoom fig. 5, due to the lack of an optimization criterion global to the network, which is the case in the prototypal example of computation of a mean [10, 5]. From a practical viewpoint, however, all nodes are rapidly assigned a mixture that is better (slightly or largely) than any of the original mixture, which implies improvement in recognition rates when the system is queried. This information is summarized in figs. 4(b) et (c), which plot the mean and variance of the BIC criterion, over the set of nodes.

2. the effectiveness of the collective model is quite close to that of a mixture model that could have been estimated directly on the whole data (dashed horizontal line).

It should also be underlined that the horizontal axis only indicates order and is non-linearly related to time, since gossiping is strongly parallel.

As illustrated in fig. 6, the scheme can easily handle a node that joins the network. In this example involving 20 nodes, a additional node joins after 50 cycles. Soon after it joins, it benefits from the previous exchanges. Indeed, the amount of data available for mixture estimation \((n_1 \text{ and } n_2)\) in eq. (2) cumulates as gossip progresses.

4. CONCLUSION

This work is a contribution towards the vision a multimedia indexing and retrieval system, which would be decentralized and deployed on a large scale. In this setting, algorithmic components are required, that induce low computa-
Algorithm 2 A gossip cycle for merging-sharing Gaussian mixture models

1. Select at random two nodes in the network, which models are \( M_i \) et \( M_j \) (practically, nodes should autonomously select their partners in a dialogue)
2. Concatenate \( M_i \) and \( M_j \) into a single model \( M_c \), then reduce this to \( M_r \)
3. Assign \( M_r \) to \( M_i \) and to \( M_j \)

Figure 4: In a 13 node network, evolution of the BIC criterion for each nodes is plotted as gossiping progresses. (a). A dashed line indicates the performance of the model estimated directly on the whole data set. (b,c) Mean and variance of the BIC criterion, over the set of nodes, are plotted as gossip progresses.

Figure 5: In the same type of experiment as previously, the temporal evolution of the BIC criterion at each node evaluated on a test set shows not to converge, due to the lack of optimization global to the network.

Figure 6: This experiment illustrates fast integration of a node joining a distributed learning process involving 20 nodes. Right from its first contact (cycle 73), the joining node strongly improves by catching the central trend of the network.
Acknowledgements
The authors are grateful to SFERE for funding this work in part, through Afshin Nikseresht’s Ph.D. grant and funding package.

5. REFERENCES