Parsimonious reduction of Gaussian mixture models with a variational-Bayes approach

Pierrick Bruneau, Marc Gelgon, Fabien Picarougne

Appears in Pattern Recognition (accepted August 2009)

Parsimonious reduction of Gaussian mixture models with a variational-Bayes approach *

Pierrick Bruneau[1,2], Marc Gelgon[1,2] and Fabien Picarougne[1]

(1) Nantes university, LINA (UMR CNRS 6241), Polytech'Nantes rue C.Pauc, La Chantrerie, 44306 Nantes cedex 3, France

(2) INRIA Atlas project-team

Tel : +33 2 40 68 32 02 Fax : +33 2 40 68 32 16

firstname.surname@univ-nantes.fr

Abstract

Aggregating statistical representations of classes is an important task for current trends in scaling up learning and recognition, or for addressing them in distributed infrastructures. In this perspective, we address the problem of merging probabilistic Gaussian mixture models in an efficient way, through the search for a suitable combination of components from mixtures to be merged. We propose a new Bayesian modelling of this combination problem, in association to a variational estimation technique, that handles efficiently the model complexity issue. A main feature of the present scheme is that it merely resorts to the parameters of the original mixture, ensuring low computational cost and possibly communication, should we operate on a distributed system. Experimental results are reported on real data.

*This work was funded by ANR Safimage, in particular through P. Bruneau's Ph.D. grant

1 Introduction

In this paper, we address the issue of probabilistic mixture model combination, in the case input and output models are Gaussian mixture models (GMM). This case is important, as this semi-parametric form is one of the most employed and versatile tool for modelling the density of multivariate continuous features. It is in particular employed for multimedia data, whether audio [24] or visual, static [15] or dynamic [11].

Aggregation of class models is a classical topic, but growing needs from many fields can be observed. Existing statistical learning and recognition tasks are being transposed onto distributed computing systems (cluster, P2P). Related applications include scaling up class-based multimedia retrieval systems [23] or estimation from sensor networks [21]. Data structures (e.g. tree-based) to handle masses of probabilistic models can also require merging these models [28].

In such contexts, one should be able to make learning sub-systems cooperate or compete, possibly in a decentralized fashion [4, 20]. This paper covers a central task among these : how multiple parametric models of the same class, but estimated from distributed sources, may merge into a single model, which parameters and complexity should be determined ? A sensible benchmark would be supplied through a model that would have been directly estimated on a centralized data source.

While a simple solution for a combined model would be obtained by a weighted sum of Gaussian mixtures, this would generally result in an unnecessarily high number of Gaussian components, with a view to capturing the underlying probability density. The scope of the paper is a new scheme for estimating, from such a possibly over-complex mixture, a mixture that is more parsimonious, yet attempts to preserve the ability to describe the underlying generative process. Preserving parsimony is particularly important if such combinations follow one after another, as one may face in large-scale cooperative multimedia class learning, in multi-target temporal tracking applications, or for building a tree index of models.

A straightforward solution would consist in sampling data from this combined mixture and re-estimating a mixture from this data, but this is generally not cost effective, especially in high dimensional spaces. In contrast, *our technique operates on the sole parameters of the over-complex mixture parameters*, ensuring lower cost for computation and communication, should the scheme operated in a distributed setting. In fact, parsimony is obtained through combination of Gaussian components. By employing a Bayesian formulation of the over-complex mixture parameter estimation and a variational approach to its resolution, the amount of compression and the suitable combination of Gaussian components may be jointly determined.

Gaussian mixture simplification through crisp combination of Gaussian components may, for small-size problems, be addressed through the Hungarian method to obtain a globally optimal combination [17]. Lower cost, local optima have been sought in [13], where the authors seek a combination that minimizes an approximation of Kullback-Leibler loss. Their technique may be viewed as a kind of k-means operating over components. As an alternative, a procedure akin to ascendent hierarchical clustering operating on Gaussian components is proposed in [26].

The search space considered in [28] is richer, as linear combinations of components are sought, rather than binary assignments, corresponding to a shift from k-means to maximum likelihood and EM operating on Gaussian components. However, these works leave open the central issue of the criterion and procedure for determining the desirable number of components.

Let us also contrast the present work from advances in combinations of classifiers [16] (ensemble methods [12], mixture of experts [6]). The present work does not combine opinions or decisions given by discriminant classifiers on a particular data set, but rather considers generative parametric class models and merges them at parameter level, without accessing data.

Bayesian estimation of mixture models is a well-known principle to solving the above issue, especially model complexity. In particular, the variational resolution provides a good trade-off between accuracy and computation efficiency, with a procedure known as Variational Bayes-EM [2] (VBEM hereafter), that compares favorably to more classical approximations of the posterior (BIC, Laplace). Yet, the standard use of VBEM is applied to data in $\mathbb{R}^n$. The central contribution of our paper is to demonstrate how simplification of an over-complex mixture may be carried out effectively by extending the Variational Bayes-EM principles to handling Gaussian components instead of real vectors. Fig 1 sketches this goal.

In section 2, we disclose this reformulation of the VBEM variational probability distribution that takes parameters rather than pointwise data as input. We show that this leads to coupled update equations, from which we derive an iterative EM-like algorithm. Yet, under the assumption of several non-redundant mixtures to be merged, it would make sense to prevent reunions of components originating from the same source. In section 3, a derivation taking this principle into account is described. Section 4 provides experimental results obtained by applying these algorithms to real data. We draw concluding remarks in section 5.

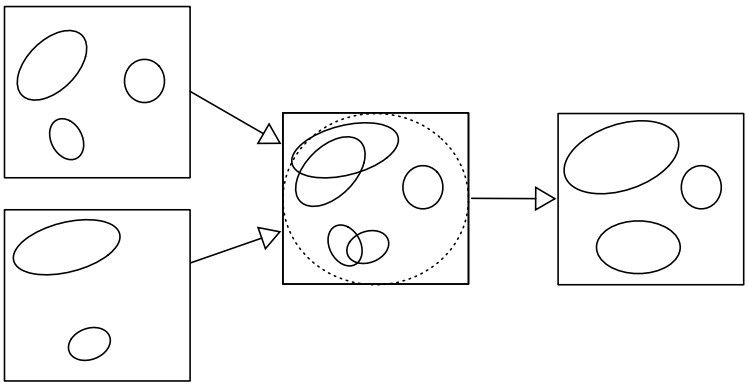


Figure 1: A toy-size illustration of the task addressed : two mixtures are added, then a suitable combination of components is sought, jointly with the task of determining how many components are required. The setting for prior distributions in the Bayesian estimation is shown as a dashed line.

2 Mixture simplification using the Variational Bayes EM principle

We first recall how the parameters and structure of a Gaussian mixture model may be estimated through a variational Bayes procedure, in the classical case of data in $\mathbb{R}^n$. We then show how this framework may be extended to achieve clustering in a space of Gaussian components.

2.1 Bayesian estimation of a mixture

We consider a set of data $X = [x_1, \ldots, x_N]^T$, to which we attempt to fit a probabilistic model parameterized by θ. The classical maximum likelihood estimation consists in maximizing the quantity $p(X|\theta)$, or equivalently the log-likelihood : $\mathcal{L}(X|\theta) = \ln p(X|\theta)$. This quantity can be interpreted as a measure for the *model fit*, i.e. how much the model is able to explain X. In the case θ is possibly of any complexity, maximizing $p(X|\theta)$ will always lead to the most complex model, and despite its perfect fit to the data, this model will lose most of its generalization power.

The Bayesian framework consists in treating the log-likelihood as a part of the marginal likelihood, or model evidence :

$$p(X) = \int p(X|\theta)\ p(\theta) d\theta \tag{1}$$

We stated earlier that the likelihood for over-complex models could grow infinitely : in eqn.

(1), these models are penalized.

Let us define Gaussian mixture models with the following notations:

$$p(x_n|\theta) = \sum_{k=1}^{K} \omega_k \mathcal{N}(x_n|\mu_k, \Lambda_k^{-1}) \tag{2}$$

where ω_k, μ_k and Λ_k are respectively the weight, mean vector and precision matrix for the component θ_k, and the full parameter set is denoted by $\theta = \{\theta_k\}$. We also define the following lightweight notations: $\Omega = \{\omega_k\}$, $\mu = \{\mu_k\}$ and $\Lambda = \{\Lambda_k\}$. The ω_k are under the constraint $\sum_k \omega_k = 1$.

Under i.i.d. assumption for X, we can conveniently decompose the global distribution:

$$p(Z|\Omega) = \prod_{n=1}^{N} \prod_{k=1}^{K} {\omega_k}^{z_{nk}} \tag{3}$$

$$p(X|Z, \mu, \Lambda) = \prod_{n=1}^{N} \prod_{k=1}^{K} \mathcal{N}(x_n|\mu_k, \Lambda_k^{-1})^{z_{nk}} \tag{4}$$

where Z is a set of binary variables denoting the component from which each element of X originates, i.e. $z_{nk} = 1 \equiv x_n$ i.i.d. from θ_k.

Various prior distributions for Gaussian mixtures were introduced in previous work. Roberts and al. [25] proposed improper flat priors. The chosen distributions had good non-informative properties, which lead to a simple analytic solution. The distributions induced by a GMM (equations (3) and (4)) naturally suggest the usage of conjugate priors. Indeed, the normal and multinomial distributions are members of the exponential family, and, as such, have a conjugate distribution. Using conjugates has a crucial advantage : the product of a likelihood function and its conjugate leads to an expression of the same functional form as the likelihood. As we will see further, in the variational framework this property highly simplifies the calculations, while preserving all the expressivity of the framework. Conjugates have extensively been used in the literature [2, 5, 10].

2.2 Variational Bayes estimation of the model

In this section, firstly we introduce general principles about variational methods, and then decline these for the case of a GMM. This constitutes a preliminary for our method.

We then derive its extension for handling Gaussian components instead of multidimensional data as input, and we explain how general VBEM properties enable automatic suppression of irrelevant Gaussian components in the mixture reduction process.

2.2.1 Review of general principles

We usually distinguish two kinds of random variables : latent variables and parameters. Latent variables scale with the data set (e.g. Z, that scales with X) while parameters are independent of the data set size (e.g. θ). In this section, let notation Y gather both latent variables and parameters.

For inference tasks, we usually specify a joint distribution $p(X, Y)$ over all variables. The purpose of a method is then to infer a posterior distribution $p(Y|X)$.

Instead of directly inferring $p(Y|X)$, we define a distribution q over Y, called *variational* distribution hereafter. $p(Y|X)$ remains unknown, and the purpose is to approximate it. The following scheme can be seen as an implementation of the principle described by equation 1. According to a simple application of Bayes' rule, the decomposition of the marginal likelihood into a lower bound and a Kullback-Leibler divergence holds :

$$\ln p(X) = \mathcal{L}(q) + KL(q \parallel p) \tag{5}$$

with :

$$\mathcal{L}(q) = \int q(Y) \ln \left\{ \frac{p(X,Y)}{q(Y)} \right\} dY \tag{6}$$

$$KL(q \parallel p) = -\int q(Y) \ln \left\{ \frac{p(Y|X)}{q(Y)} \right\} dY \tag{7}$$

As we stated previously, $\ln p(X)$ is a constant. This means that maximizing $\mathcal{L}(q)$ is equivalent to minimizing the divergence between $p(Y|X)$ and $q(Y)$. Solving this problem will therefore provide us with an approximation to $p(Y|X)$.

Tractability of further calculations are ensured by assuming it is possible to express $q(Y)$ in a factorized form :

$$q(Y) = \prod_{i=1}^{M} q_i(Y_i) \tag{8}$$

Under this formalism, we can rewrite (6) w.r.t. to a single term q_j :

$$\mathcal{L}(q) = \int q_j \ln \tilde{p}(X, Y_j) dY_j - \int q_j \ln q_j dY_j + \text{const} \tag{9}$$

with :

$$\ln \tilde{p}(X, Y_j) = \int \ln p(X, Y) \prod_{i \neq j} q_i dY_i \quad (10)$$

$$= \mathbb{E}_{i \neq j} [\ln p(X, Y)] + \text{const} \quad (11)$$

$E_{i \neq j}[.]$ denotes the expectation w.r.t. q_i terms for $i \neq j$.

The expression (9) is a negative KL divergence between q_j and $\tilde{p}(X, Y_j)$. This means that maximizing $\mathcal{L}(q)$ is equivalent to minimizing this KL divergence. This occurs when the two distributions are equal, we can therefore define q_j in its optimal setting :

$$\ln q_j^* = \mathbb{E}_{i \neq j}[\ln p(X, Y)] + \text{const} \quad (12)$$

Let us consider the more specific case of Gaussian mixtures. We previously defined the distributions (3) and (4), and now give the corresponding priors:

$$p(\Omega) = Dir(\Omega|\alpha_0) = C(\alpha_0) \prod_{k=1}^{K} \omega_k^{\alpha_0 - 1} \quad (13)$$

$$p(\mu, \Lambda) = p(\mu|\Lambda)p(\Lambda) \quad (14)$$

$$= \prod_{k=1}^{K} \mathcal{N}(\mu_k|m_0, (\beta_0 \Lambda_k)^{-1}) \; \mathcal{W}(\Lambda_k|W_0, \nu_0) \quad (15)$$

where Dir and $\mathcal{W}$ respectively denote the Dirichlet and Wishart distributions.

According to the associated graphical model (figure 2), expressions (4), (3), (13) and (15) define the following joint distribution:

$$p(X, Z, \Omega, \mu, \Lambda) = p(X|Z, \mu, \Lambda)p(Z|\Omega)p(\Omega)p(\mu|\Lambda)p(\Lambda) \quad (16)$$

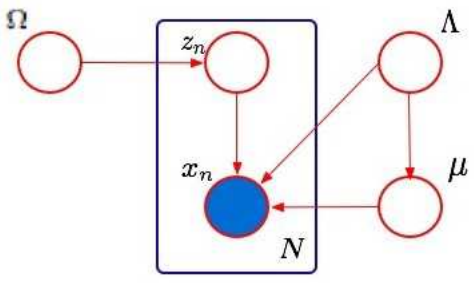


Figure 2: Graphical model associated with the Bayesian GMM estimation problem

We define a factorized variational distribution:

$$q(Z, \Omega, \mu, \Sigma) = q(Z)q(\Omega) \prod_k q(\mu_k, \Sigma_k) \tag{17}$$

Applying formula (12) for $q(Z)$, and identifying the obtained posterior to a multinomial functional form gives the following estimates :

$$r_{nk} = \frac{\rho_{nk}}{\sum_{j=1}^{K} \rho_{nj}} \tag{18}$$

with unnormalized log estimates:

$$\ln \rho_{nk} = \mathbb{E}[\ln \omega_k] + \frac{1}{2}\mathbb{E}[\ln \det(\Lambda_k)] - \frac{D}{2}\ln(2\pi) - \frac{1}{2}\mathbb{E}_{\mu_k,\Lambda_k}[(x_n - \mu_k)^T \Lambda_k (x_n - \mu_k)] \tag{19}$$

This update scheme uses the following moments evaluated w.r.t. the current θ estimates :

$$\begin{aligned}
\mathbb{E}_{\mu_k,\Lambda_k}[(x_n - \mu_k)^T \Lambda_k (x_n - \mu_k)] &= D\beta_k^{-1} + \nu_k (x_n - m_k)^T W_k (x_n - m_k) && (20)\\
\ln \tilde{\Lambda}_k \equiv \mathbb{E}[\ln \det(\Lambda_k)] &= \sum_{i=1}^{D} \psi\left(\frac{\nu_k + 1 - i}{2}\right) + D\ln 2 + \ln \det(W_k) && (21)\\
\ln \tilde{\omega}_k \equiv \mathbb{E}[\ln \omega_k] &= \psi(\alpha_k) - \psi(\hat{\alpha}) && (22)
\end{aligned}$$

For convenience, current r_{nk} estimates are used to define these synthetic statistics :

$$\begin{aligned}
N_k &= \sum_{n=1}^{N} r_{nk} && (23)\\
\bar{x}_k &= \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} x_n && (24)\\
S_k &= \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (x_n - \bar{x}_k)(x_n - \bar{x}_k)^T && (25)
\end{aligned}$$

Again applying formula (12), and identifying adequate functional forms, we obtain posterior model parameters estimates :

$$\alpha = (\alpha_k) \text{ and } \alpha_k = \alpha_0 + N_k \tag{26}$$

$$\beta_k = \beta_0 + N_k \quad (27)$$

$$m_k = \frac{1}{\beta_k}(\beta_0 m_0 + N_k \bar{x}_k) \quad (28)$$

$$W_k^{-1} = W_0^{-1} + N_k S_k + \frac{\beta_0 N_k}{\beta_0 + N_k}(\bar{x}_k - m_0)(\bar{x}_k - m_0)^T \quad (29)$$

$$\nu_k = \nu_0 + N_k \quad (30)$$

Cycling through these update equations implements an EM-like algorithm. More precisely :

- E step : compute expressions (18), (20), (21), (22), (23), (24) and (25)
- M step : compute expressions (26), (27), (28) and (29)

2.2.2 Handling components to cluster as virtual samples

The introduction of this paper discussed contexts related to multimedia class description sharing, in which recovering a common parsimonious mixture is a central issue. Now consider an arbitrary mixture defining L components, with parameters $\theta' = \{\Omega', \mu', \Lambda'\}$. Let us note that this mixture might be obtained by regrouping several descriptions. We then assume that X and Z' were i.i.d sampled from this distribution. It is therefore possible to regroup X according to the component from which its data was drawn. It leads us to the following formalism : $X = \{\hat{x}_1, \ldots, \hat{x}_L\}$ with $\text{card}(X) = N, \hat{x}_l = \{x_n | z'_{nl} = 1\} = \{x_{ln}\}$ and $\text{card}(\hat{x}_l) = \omega'_l N$. Now we express the likelihood (4) of such a dataset under a new and unknown model $\theta = \{\Omega, \mu, \Lambda\}$. Let us note that this new model comes with its specific latent variable $Z \neq Z'$. For the further developments to be tractable, we assume that $\forall x_n \in \hat{x}_l, z_{nk} = \text{const} = z_{lk}$. This can seem a strong assumption, but simplifying a model will be more likely about regrouping components, so in general it will hold. Thus we can rewrite expression (4) as follows :

$$p(X|Z, \mu, \Lambda) = \prod_{k=1}^{K} \prod_{l=1}^{L} p(\hat{x}_l | Z, \mu_k, \Lambda_k)^{z_{lk}} \quad (31)$$

$$p(X|Z, \mu, \Lambda) = \prod_{k=1}^{K} \prod_{l=1}^{L} \left[\prod_{n=1}^{\omega'_l N} \mathcal{N}(x_{ln} | \mu_k, \Lambda_k^{-1}) \right]^{z_{lk}} \quad (32)$$

$$\ln p(X|Z, \mu, \Lambda) = \sum_{k=1}^{K} \sum_{l=1}^{L} z_{lk} \left[\sum_{n=1}^{\omega'_l N} \ln \mathcal{N}(x_{ln} | \mu_k, \Lambda_k^{-1}) \right] \quad (33)$$

For N sufficiently large, we can make the following approximation :

$$\sum_{n=1}^{\omega'_l N} \ln \mathcal{N}(x_{ln}|\mu_k, \Lambda_k^{-1}) \simeq \omega'_l N \mathbb{E}_{\mu'_l,\Lambda'_l}\left[\ln \mathcal{N}(x|\mu_k, \Lambda_k^{-1})\right] \tag{34}$$

This statement is known as *virtual sampling*, and was introduced in [29, 28].

The expectation may be explicited :

$$\mathbb{E}_{\mu'_l,\Lambda'_l}\left[\ln \mathcal{N}(x|\mu_k, \Lambda_k^{-1})\right] = \int \mathcal{N}(x|\mu'_l, \Lambda_l'^{-1}) \ln \mathcal{N}(x|\mu_k, \Lambda_k^{-1})\, \mathrm{d}x \tag{35}$$

$$\begin{aligned}\mathbb{E}_{\mu'_l,\Lambda'_l}\left[\ln \mathcal{N}(x|\mu_k, \Lambda_k^{-1})\right] = -KL\left(\mathcal{N}(x|\mu'_l, \Lambda_l'^{-1}) \parallel \mathcal{N}(x|\mu_k, \Lambda_k^{-1})\right) \\ -H(\mathcal{N}(x|\mu'_l, \Lambda_l'^{-1}))\end{aligned} \tag{36}$$

with $KL(q_0 \parallel q_1)$ the KL divergence of q_1 from q_0 and $H(q_0)$ the entropy of q_0. These two terms benefit from closed-form expressions [7]. Thus by reinjecting (36) into (34), and then (34) into (33), we obtain the convenient following expression for $p(X|Z, \mu, \Lambda)$:

$$\begin{aligned}\ln p(X|Z, \mu, \Lambda) = N \sum_{k=1}^{K}\sum_{l=1}^{L} z_{lk}\omega'_l \\ \left[-KL\left(\mathcal{N}(x|\mu'_l, \Lambda_l'^{-1}) \parallel \mathcal{N}(x|\mu_k, \Lambda_k^{-1})\right) - H(\mathcal{N}(x|\mu'_l, \Lambda_l'^{-1}))\right]\end{aligned} \tag{37}$$

$$\begin{aligned}\ln p(X|Z, \mu, \Lambda) = N \sum_{k=1}^{K}\sum_{l=1}^{L} z_{lk}\omega'_l \\ \left[\frac{1}{2}\ln\det\Lambda_k - \frac{1}{2}\mathrm{Tr}(\Lambda_k\Lambda_l'^{-1}) - \frac{1}{2}(\mu'_l - \mu_k)^T\Lambda_k(\mu'_l - \mu_k) - \frac{d}{2}\ln(2\pi)\right]\end{aligned} \tag{38}$$

Here we notice that by considering an hypothetic data set originating from an arbitrary input model θ', it is possible to derive a limit expression for $\ln p(X|Z, \mu, \Lambda)$ that exhibits no dependence on the original data X and Z'. The formalism change also has consequences on (3) : as we previously stated that $z_{lk} = z_{nk}\ \forall x_n \in \hat{x}_l$, we can write :

$$p(Z|\Omega) = \prod_{n=1}^{N}\prod_{k=1}^{K}\omega_k^{z_{nk}} = \prod_{l=1}^{L}\prod_{k=1}^{K}\omega_k^{N\omega'_l z_{lk}} \tag{39}$$

Variational update equations are partially based on moments evaluated w.r.t $p(Z)$ and $p(X)$. Therefore cascading consequences occur relatively to the classical VBEM algorithm.

As a consequence of (38) and (39), the modified unnormalized estimates for $q(Z)$ obtained from application of formula (12) now are :

$$\begin{aligned}\ln(\rho_{lk}) = \frac{N\omega'_l}{2}\left(2\mathbb{E}[\ln\omega_k] + \mathbb{E}[\ln\det\Lambda_k] - d\ln(2\pi)\right) \\ -\frac{N\omega'_l}{2}\left(\mathbb{E}_{\mu_k,\Lambda_k}\left[\mathrm{Tr}(\Lambda_k\Lambda_l'^{-1}) + (\mu'_l - \mu_k)^T\Lambda_k(\mu'_l - \mu_k)\right]\right)\end{aligned} \tag{40}$$

leading to $\{r_{lk}\}$ estimates as in the classic scheme. The moment w.r.t μ_k and Λ_k is easily evaluated to give $\frac{d}{\beta_k} + \nu_k \left[\mathrm{Tr}(W_k \Lambda_l'^{-1}) + (\mu_l' - m_k)^T W_k (\mu_l' - m_k)\right]$.

Analogously to the classical scheme, for further convenience, we define the following synthetic statistics :

$$N_k = \sum_l^L N\omega_l' r_{lk} \tag{41}$$

$$\bar{x}_k = \frac{1}{N_k} \sum_l^L N\omega_l' r_{lk} \mu_l' \tag{42}$$

$$S_k = \frac{1}{N_k} \sum_l^L N\omega_l' r_{lk} (\mu_l' - \bar{x}_k)(\mu_l' - \bar{x}_k)^T \tag{43}$$

$$C_k = \frac{1}{N_k} \sum_l^L N\omega_l' r_{lk} \Lambda_l'^{-1} \tag{44}$$

Applying formula (12) for $q(\Omega)$ and $q(\mu, \Lambda)$, and using the synthetic statistics, we obtain the following update formulæ :

$$\alpha_k = \alpha_0 + N_k \tag{45}$$

$$\beta_k = \beta_0 + N_k \tag{46}$$

$$m_k = \frac{1}{\beta_k} (\beta_0 m_0 + N_k \bar{x}_k) \tag{47}$$

$$W_k^{-1} = W_0^{-1} + N_k S_k + N_k C_k + \frac{\beta_0 N_k}{\beta_0 + N_k} (\bar{x}_k - m_0)(\bar{x}_k - m_0)^T \tag{48}$$

$$\nu_k = \nu_0 + N_k \tag{49}$$

The classical VBEM algorithm is known to monotonically decreases the KL divergence between the variational pdf and the true posterior [5]. This is equivalent to maximizing the lower bound of the complete likelihood. As we can compute this lower bound, and as this bound should never decrease, we can test for convergence by comparing two successive values of the bound. For our derivation, only terms involving X or Z might change, these are the following :

$$\begin{aligned} \mathbb{E}[\ln p(X|Z,\mu,\Lambda)] = \frac{1}{2} \sum_k N_k \{ \ln \tilde{\Lambda}_k - \frac{d}{\beta_k} - \nu_k \mathrm{Tr}\left((S_k + C_k) W_k\right) \\ - \nu_k (\bar{x}_k - m_k)^T W_k (\bar{x}_k - m_k) - d \ln(2\pi) \} \end{aligned} \tag{50}$$

$$\mathbb{E}[\ln p(Z|\Omega)] = \sum_k N_k \ln \tilde{\omega}_k \tag{51}$$

Regarding the choice of the prior α_0 parameter, the strategy described in [5] still applies to our context. By choosing $\alpha_0 < 1$, the estimation process will favor a solution where at least one of the ω_k is 0. Therefore, by choosing a sufficiently large initial number K of components, we shall obtain a number of effective components K'. In the case of our method, we reduce the virtual sample : as we reduce it to the strictly necessary number of components, this is equivalent to suppressing redundancy in the input GMM.

3 Obtaining parsimony under constraints

Let us consider several data repositories, each one being the source of a Gaussian mixture fitted on the available data. The method proposed in section 2.2.2, named VBmerge hereafter, makes a weighted sum of all components from all sources in a single large mixture, and reduces it. Yet, doing so with a large number of sources has a drawback : as we obtain a globally very noisy model, the number of components is reduced drastically (see experimental results). Should we assume that each source produces a non-redundant Gaussian mixture, it would be sensible to penalize reductions that imply assigning components originating from the same source to the same target component.

3.1 Integrating constraints in the framework

Consequently, let us design a probabilistic model and derive the associated estimation algorithm, that takes into account this constraint to tackle the mixture merging question efficiently. Consider that the L components come from P distinct sources (necessarily, $L \geq P$). We denote a_{lp} the binary variable that denotes wether component l originates from source p or not. Let us define A the $L \times P$ matrix formed with a_{lp} values. As we know where each component originates from, A is a set of observed values.

We define a *pdf* over this new data set. The purpose of such a distribution is to model how much assignments of the L components violate or enforce the constraints defined by A, so it is sensible to restrict A dependencies to Z. Furthermore, A can be seen as originating from this distribution ; an assignment configuration (summarized by Z) enforcing the constraints would therefore result in a higher likelihood for the model. Before introducing the distribution, let us consider the $P \times K$ matrix $M = A^T Z$. One of its single terms m_{pk} measures how many components from a single source p are associated with the same target component k. Clearly, we want this amount to be as low as possible, so we model this constraint with a Poisson distribution parametrized with $\lambda = 1$ over each term. This will tend to favor rare events. Thus

the *pdf* over A is as follows :

$$p(A|Z) = p(M = A^T Z) = \prod_{p=1}^{P} \prod_{k=1}^{K} \frac{e^{-1}}{(1 + m_{pk})!} \tag{52}$$

The term 1 in eqn. (52) is added for conveniency, and causes no loss of generality. The joint distribution (16) is then augmented with eqn. (52).

Let us note that no additional term is added to the factorized distribution (17), and that, according to the general formulation (12), the term $p(A|Z)$ shall only influence the optimal setting for $q(Z)$. Therefore, update formulae from section 2.2.2 will remain unchanged except for the unnormalized estimates of Z (eqn (40)) :

$$\ln q^*(Z) = \sum_{l=1}^{L} \sum_{k=1}^{K} z_{lk} \ln \rho_{lk} - \sum_{k=1}^{K} \sum_{p=1}^{P} \ln(1 + m_{pk})! + \text{const} \tag{53}$$

Or equivalently :

$$\ln q^*(Z) = \sum_{l=1}^{L} \sum_{k=1}^{K} z_{lk} \ln \rho_{lk} - \sum_{k=1}^{K} \sum_{p=1}^{P} \sum_{i=1}^{m_{pk}} \ln(1 + i) + \text{const} \tag{54}$$

Let us denote $z_{.k}$ the set $\{z_{lk} | \ \forall l\}$ (and respectively $z_{l.}$). In the traditional scheme, $\ln q^*(Z)$ factorizes over l and k, giving rise to independent optimal z_{lk} estimates (more precisely, only unnormalized estimates are fully independent : each z_{lk} ultimately depends on $\rho_{l.}$ in order to obtain normalized r_{lk} values). Here this does not hold any more. All z_{lk} forming a single $z_{.k}$ are co-dependent : we must devise an alternate to the traditional E step.

We choose to define an order in the set of individuals, and approximate the overall co-dependent estimates by a one-pass scheme based on using already discovered estimates. This leads to the following approximation :

$$q(Z) = q(z_{1.}) q(z_{2.}|z_{1.}) q(z_{3.}|z_{1.} z_{2.}) \dots q(z_{L.}|z_{1.} \dots z_{L-1.}) \tag{55}$$

Our *E step* algorithm will proceed each term of the r.h.s. in increasing ranks order. We will describe the 2 first steps of the algorithm, leading to a general formulation. This iterated conditional scheme is closely related to ICM (iterated conditional modes) [3].

3.2 Initializing the scheme

Let us recall that $m_{pk} = \sum_{l=1}^{L} a_{lp} z_{lk}$. Our formulation allows us to restrict this sum to the current rank of the algorithm. For the first step we have :

$$\ln q^*(z_{1.}) = \sum_{k=1}^{K} z_{1k} \ln \rho_{1k} - \sum_{k=1}^{K} \sum_{p=1}^{P} \ln(1 + a_{1p} z_{1k}) + \text{const} \tag{56}$$

For a single z_{1k}, this leads to :

$$\ln q^*(z_{1k}) = z_{1k} \ln \rho_{1k} - \sum_{p=1}^{P} \ln(1 + a_{1p} z_{1k}) + \text{const} \tag{57}$$

Clearly, as such, this expression cannot give a multinomial law estimate. However, using a first order Taylor expansion for $\ln(1 + x)$, we obtain :

$$\ln q^*(z_{1k}) = z_{1k} \ln \rho_{1k} - \sum_{p=1}^{P} a_{1p} z_{1k} + \text{const} \tag{58}$$

$$\ln q^*(z_{1k}) = z_{1k} \ln \frac{\rho_{1k}}{e^{\sum_{p=1}^{P} a_{1p}}} + \text{const} \tag{59}$$

As each original component belongs to only one source,

$$\ln q^*(z_{1k}) = z_{1k} \ln \frac{\rho_{1k}}{e} + \text{const} \tag{60}$$

Giving a modified unnormalized estimate $\rho'_{1k} = \frac{\rho_{1k}}{e}$. This leads to the same normalized estimates as in the classical scheme (e denominator is constant and disappears).

3.3 A new general update formula for $\ln q^*(Z)$

Changing the rank of the restriction in eq. 56 leads to :

$$\begin{aligned} \ln q^*(z_{2.}|z_{1.}) = &\sum_{k=1}^{K} z_{2k} \ln \rho_{2k} - \sum_{k=1}^{K} \sum_{p=1}^{P} \ln(1 + a_{1p} z_{1k}) \\ &- \sum_{k=1}^{K} \sum_{p=1}^{P} \ln(1 + a_{1p} z_{1k} + a_{2p} z_{2k}) + \text{const} \end{aligned} \tag{61}$$

After considering a single k, and applying Taylor expansion supplies:

$$\begin{aligned} \ln q^*(z_{2k}|z_{1k}) = z_{2k} \ln \rho_{2k} &- \sum_{p=1}^{P} a_{1p} z_{1k} \\ &- \sum_{p=1}^{P} (a_{1p} z_{1k} + a_{2p} z_{2k}) + \text{const} \end{aligned} \tag{62}$$

Let us note $a_{i\max} = \arg\max_p a_{ip}$ and $z_{i\max} = \arg\max_k z_{ik}$. Using these notations, the previous expression can be factorized as following :

$$\ln q^*(z_{2k}|z_{1k}) = z_{2k}(\ln \rho_{2k} - 1 - 2\delta_{a_{1\max}, a_{2\max}} . \delta_{z_{1\max}, k}) + \text{const} \tag{63}$$

where δ is the Kronecker delta. This leads to a modified unnormalized estimate :

$$\rho'_{2k} = \frac{\rho_{lk}}{e^{1+2\delta_{a_{1\max},a_{2\max}}.\delta_{z_{1\max},k}}}$$

For any rank, same considerations lead to the following general formula :

$$\rho'_{jk} = \frac{\rho_{jk}}{e^{1+\sum_{i=1}^{j-1}(j-i+1).\delta_{a_{i\max},a_{j\max}}.\delta_{z_{i\max},k}}} \tag{64}$$

where j is the rank of the current item (i.e. original component).

3.4 Modified bound

Adding a term in our joint distribution implies modifying the bound discussed at the end of section 2.2.2. More specifically, the following term shall be added :

$$\mathbb{E}[\ln p(A \mid Z)] = \sum_{k=1}^{K}\sum_{p=1}^{P}\left[-1 - \sum_{i=0}^{\mathbb{E}[m_{pk}]}\ln(1+i)\right] \tag{65}$$

$$= -KP - \sum_{k=1}^{K}\sum_{p=1}^{P}\sum_{i=0}^{\mathbb{E}[m_{pk}]}\ln(1+i) \tag{66}$$

with

$$\mathbb{E}[m_{pk}] = \mathbb{E}\left[\sum_{l=1}^{L} a_{lp}z_{lk}\right] = \sum_{l=1}^{L} a_{lp}\mathbb{E}[z_{lk}] = \sum_{l=1}^{L} a_{lp}r_{lk} \tag{67}$$

In the classical VBEM scheme, this lower bound is strictly increasing during the estimation process. As we chose an approximate heuristic for our modified E step, this property does not hold any more : slight decreases can therefore be observed. But this does not change the principle of the algorithm : we still can use $\Delta(\text{bound}) < \text{threshold}$ as a stop criterion, the only difference being that now Δ might be negative.

4 Experiments

The framework presented here can be applied to numerous tasks. In this paper we propose two simple experimental settings :

- a case of distributed clustering, where several cluster structures will be merged,
- and a simple classification task performed over a database of images.

By doing so, we aim at showing :

- the respective interests of the methods presented in the paper (VBmerge or its constrained derivation),
- comparisons with alternative methods (resampling, k-nearest neighbors)

Task 1 : mixture estimation from distributed data

We consider a data set that is partitioned across several sites. On each of these sites, mixture model estimation is carried out independently, on local data. This scenario is typical of many distributed computing settings. Our point is to assess the quality of the model that can obtained by aggregating models fitted separately, especially compared to what would be obtained by fitting directly a GMM on the whole data set (impossible in real-world application, but a good figure-of-merit in evaluation phase). Secondly, a suitable initial value for K is sought. Indeed, K should be as big as possible to avoid the worst local minima, but it also should be as small as possible to limit the computational resources needed.

For these experiments, we used three UCI data sets : *Shuttle* used by the StatLog project, pen-based recognition of handwritten digits data [1] (named *Pendigits* hereafter), and the MAGIC Gamma telescope data set [8] (named *magic* hereafter). *Shuttle* has 9 numerical attributes, which are flight measurements obtained from a spacecraft. The status associated with each observation will stand for the class (or ground truth), and we have 5 different possible status in the data set. *Pendigits* has 16 numerical attributes, being obtained from positions of a stylus on a tablet. As the class is the digit drawn on the tablet, we have 10 classes. *magic* is defined over 10 numerical attributes. This data set contains background or positive signals, which form 2 distinct classes. We randomly selected 1000 items from each of these databases. We then followed the following protocol :

- global model fitting on the whole data set,
- separation of the 1000 observations into 10 subsamples,
- model fitting on each subsample,
- aggregation of separate models, and comparisons to the global model. The model obtained by resampling from the weighed average of the separate models is also considered. For the constrained VBmerge scheme, constraints are specified between components originating from the same submodel.

The ground truth classes might not conform to the Gaussian hypothesis (i.e. they do not originate from unknown Gaussians), so BIC scores [27] measuring a likely number of groups

are given for each data set as a reference. These scores were obtained with a classic EM algorithm for Gaussian mixtures. As fitting a Gaussian mixture on a data set is equivalent to building a cluster structure, we measured the posterior couple error (this measure penalizes cluster structures that gather data items from different true classes, and reciprocally), the number of final effective components (i.e. K' denoted previously) of each model, and the Jensen-Shannon (JS) divergence of separate, merged, or resampled models w.r.t. the overall model. The experiment was conducted with various values for K, and each result was averaged over 20 runs. JS divergence is a symmetric and normalized version of KL divergence. Average measures for models fitted on subsamples are given as a comparison. Results are provided in figures 3 and 4. Let us add a remark about the separation into subsamples : the subsampling is performed randomly and independently for each run. A bias is induced by this design choice. For example, the lack of smoothness of the curves presented on figures 3 and 4 is an artefact associated to this bias. Nevertheless, we believe that this choice, associated with averaging over 20 runs, leads to much more significant and robust results than what would be obtained with a "static" subsampling.

The following remarks may be drawn from these results :

- depending on the data set, the observed couple error would not be interpreted in the same way : the alternate schemes behave better for *magic*, worse for *pendigits*, and similarly for *Shuttle*. Yet this measure is not a very good reference, as it relies on the Gaussianity of the true classes.

- Measured BIC scores indicate the most likely number of groups is between 6 and 10 for *pendigits*, around 4 for *Shuttle* and around 3 for *pendigits*. Let us especially notice the tendency of the constrained scheme to over-estimate the number of groups.

- However, the divergences of the distributions obtained with VBmerge or its constrained version are generally better than those obtained with a resampling scheme, with a much lower algorithmic cost.

- For almost all curves there is an asymptotic behaviour : beyond a certain K, increasing it does not significantly change the expected result. By visually inspecting the graphs, we can set this number to 250. This number will be use for the variational procedures involved in the next section.

- The overhead in terms of model complexity for the constrained scheme is generally outweighed by a lower divergence w.r.t the global model. In cases computational resources

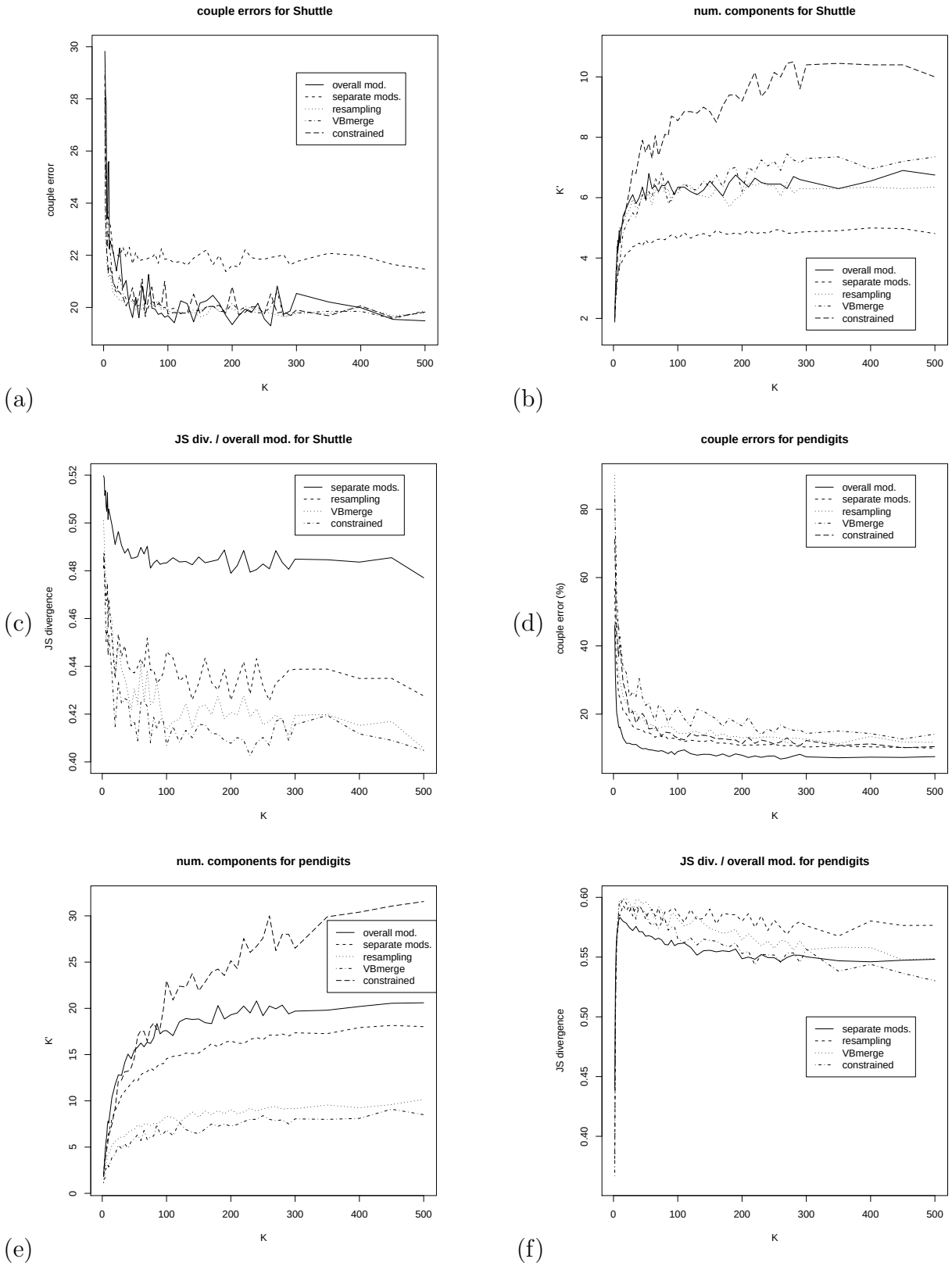
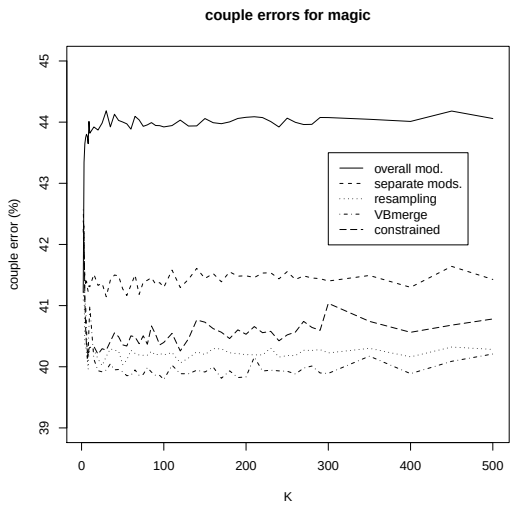
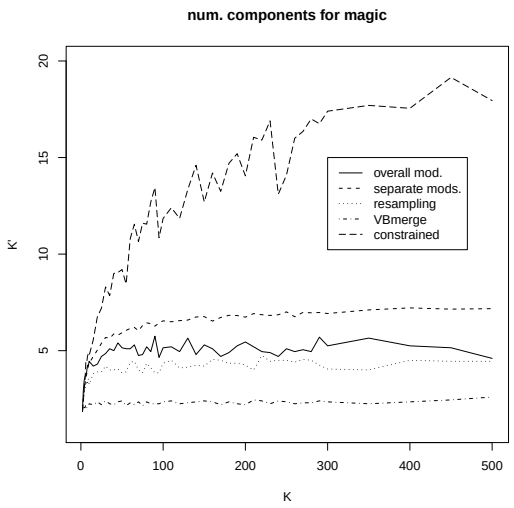


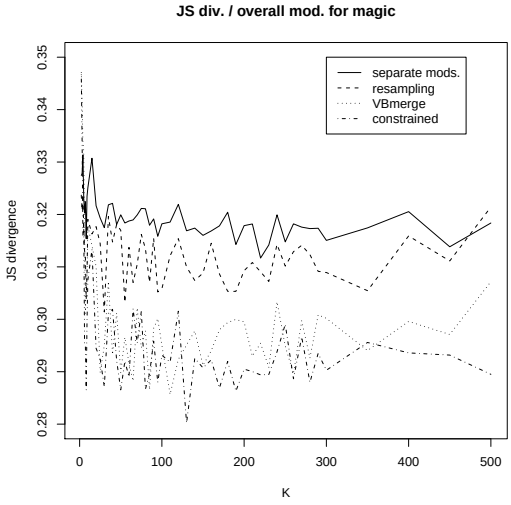
Figure 3: Results presented as a function of the initial K. a) couple error for *Shuttle* b) number of effective components for *Shuttle* c) JS divergence w.r.t the overall model for *Shuttle* d) couple error for *pendigits* e) number of effective components for *pendigits* f) JS divergence w.r.t the overall model for *pendigits*



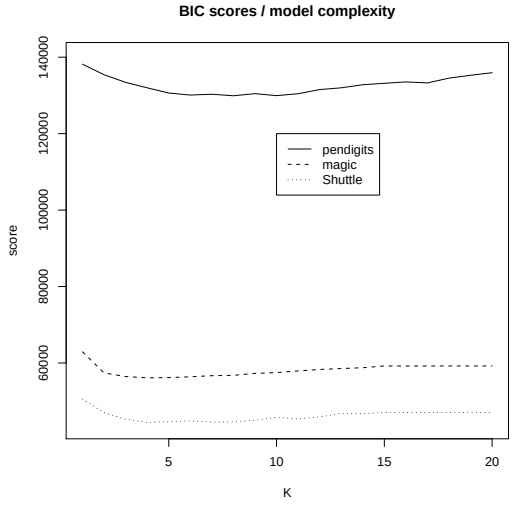
(g)



(h)



(i)



(j)

Figure 4: g) couple error for *magic* h) number of effective components for *magic* i) JS divergence w.r.t the overall model for *magic* j) BIC scores for the three data sets as a function of the number of components used

might be an issue, post-processing on the output of our constrained scheme may be considered. As shown in [9], the cost of VBmerge is almost linear w.r.t the number of input components : in this case it will be almost insignificant because this number will be already strongly reduced.

Task 2 : a naive classification task

Let us illustrate our scheme in the context of visual object recognition. This experiment does not aim at competing with current methods (e.g. vector-quantized SIFT/GIST words inputs Latent Dirichlet Allocation models [18]), but rather showing how the technique may be used in the framework of a tree-based class indexing scheme [28].

For this experiment, we selected 300 images from the 10 first categories in the Caltech-256 object category dataset [14] (30 images randomly chosen in each category). We consider each image as a data source, and fit a Gaussian mixture over its pixel data (L,a,b color space). Obtained individual Gaussian mixtures comprise 20.6 components on average.

The protocol is then similar to the one suggested by Vasconcelos [28] : we perform a leave-one-out classification task, where each image is taken alone and matched against the database. The matching database object is chosen as having the lowest JS divergence w.r.t the query image. With the VBmerge and constrained schemes, all other images are used to build one summary for each class. As a reference, we implemented a version where summaries are obtained with a resampling scheme, and also a k-nearest neighbor classifier which tries to match the images against the database. The best k was evaluated experimentally to 5. The experiment is performed for each image and each scheme, and the results were averaged over 20 runs. These are presented in figure 1.

Here we just wanted to underline that the results are not significantly different from those obtained with a classical classifier. Moreover, the objects in the chosen collection suffer from cluttered background, which makes the chosen representation space poorly distinctive. Observations used herein are crude. Yet, as learning visual vocabularies and topics from web-distributed training sets is currently attracting growing attention, a valuable perspective will consist in assessing the present technique with state-of-the-art observations (e.g. GIST descriptors [22], SIFT descriptors [19]).

The indicated computational times are the seconds taken to match an element to the database. We see that the constrained scheme leads to a very significant gain, with little error loss and model complexity overhead. To provide a fair comparison with k-NN, at each classification attempt with VBmerge, its constrained version or the resampling scheme, we

recompute the associated summaries (i.e. by adding the element from the previous attempt and removing the current one). Under the assumption of a static set of summaries, computational time would therefore be greatly reduced.

	classif. error (%)	average time (s)	number of components
k-NN (k=5)	68.7	8.09	
resampling	68.5	42.88	10.42
VBmerge	69.6	10.71	18.07
constrained	70.1	3.06	21.13

Table 1: Results for the classification task

5 Conclusion

Low-cost combination of multimedia class descriptions is a crux of future pattern recognition application of distributed infrastructures. In this paper, we described a novel approach dedicated to the mixture reduction phase involved in merging Gaussian mixture models, by transposing the variational Bayes framework to Gaussian components. We showed that operating through parameters provides considerable advantages in terms of cost efficiency, while trading off only little in terms of estimation accuracy.

We are considering several extensions of the present work. First, aggregating mixtures of PPCA are in fact a direct extension of our proposal, that should benefit handling of high dimensional spaces. A second task under way attempts to integrate, at the Gaussian component-level, some constraints that have been proposed in semi-supervised clustering (e.g. assign/don't assign to the same cluster). Thereby, in the mixture reduction process, the mixture from which each component originates would be taken into account. Finally, the counterpart of the present work for mixture of t-distributions would enable its application for robust probabilistic distributed clustering, with richer representations than standard consensus approaches.

References

[1] F. Alimoglu. Combining multiple classifiers for pen-based handwritten digit recognition. Technical report, Institute of Graduate Studies in Science and Engineering, 1996.

[2] H. Attias. A variational bayesian framework for graphical models. *Advances in Neural Information Processing Systems*, 2000.

[3] S. Basu, M. Bilenko, A. Banerjee, and R. J. Mooney. Probabilistic semi-supervised clustering with constraints. In *Semi-Supervised Learning*. MIT Press, 2006.

[4] M. Bechchi, G. Raschia, and N. Mouaddib. Merging distributed database summaries. In *Proc. ACM CIKM '07*, pages 419–428, Lisbon, November 2007.

[5] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

[6] Christopher M. Bishop and Markus Svensén. Bayesian hierarchical mixtures of experts. In Christopher Meek and Uffe Kjærulff, editors, *UAI*, pages 57–64. Morgan Kaufmann, 2003.

[7] R. E. Blahut. *Principles and Practice of Information Theory*. Addison-Wesley, 1987.

[8] R. K. Bock, A. Chilingarian, M. Gaug, F. Hakl, T. Hengstebeck, M. Jirina, J. Klaschka, E. Kotrc, P. Savicky, S. Towers, A. Vaiciulis, and W. Wittek. Methods for multidimensional event classification: a case study using images from a cherenkov gamma-ray telescope. *Nuclear instruments and methods in physics research. Section A, Accelerators, spectrometers, detectors and associated equipment*, 516(2-3):511–528, 2004.

[9] P. Bruneau, M. Gelgon, and F. Picarougne. Parsimonious variational-bayes mixture aggregation with a poisson prior. *To be published in EUSIPCO'09*, 2009.

[10] C. Constantinopoulos and M. K. Titsias. Bayesian feature and model selection for Gaussian mixture models. *IEEE Trans. on Pattern Analysis Machine Intelligence*, 28(6):1013–1018, 2006.

[11] R. Fablet, P. Bouthemy, and P. Perez. Non-parametric motion characterization using causal probabilistic models for video indexing and retrieval. *IEEE Trans. on Image Processing*, 11(4), apr 2002.

[12] Y. Freund. An adaptive version of the boost by majority algorithm. *Machine Learning*, 43(3):293–318, 2001.

[13] J. Goldberger and S. Roweis. Hierarchical clustering of a mixture model. *NIPS*, 2004.

[14] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[15] R. Hammoud and R. Mohr. Mixture densities for video objects recognition. In *International Conference on Pattern Recognition (ICPR'2000)*, pages 71–75, Barcelona, Spain, September 2000.

[16] J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(3):226–239, 1998.

[17] B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer, 2002.

[18] D. Larlus and F. Jurie. Latent mixture vocabularies for object categorization and segmentation. *Journal of Image & Vision Computing*, 27(5):523–534, apr 2009.

[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[20] A. Nikseresht and M. Gelgon. Gossip-based computation of a Gaussian mixture model for distributed multimedia indexing. *IEEE Transactions on Multimedia*, (3):385–392, March 2008.

[21] R. Nowak. Distributed EM algorithms for density estimation and clustering in sensor networks. *IEEE Trans. on Signal Processing*, 51(8), August 2003.

[22] A. Oliva and A. Torralba. Modeling the shape of the scene : a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[23] J. Ponce, M. Hebert, C. Schmid, and A. Zisserman. *Towards category-level object recognition*. Springer, 2006.

[24] D.A. Reynolds and R.C Rose. Text independent speaker identification using gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, 3(1):72–83, 1995.

[25] S. J. Roberts, D. Husmeier, I. Rezek, and W. Penny. Bayesian approaches to Gaussian mixture modelling. *IEEE Trans. on Pattern Analysis Machine Intelligence*, 20(11):1133–1142, 1998.

[26] A. Runnalls. A Kullback-Leibler approach to Gaussian mixture reduction. *IEEE Trans. on Aerospace and Electronic Systems*, 43(3):989–999, July 2007.

[27] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.

[28] N. Vasconcelos. Image indexing with mixture hierarchies. *Proceedings of IEEE Conference in Computer Vision and Pattern Recognition*, 1:3–10, 2001.

[29] N. Vasconcelos and A. Lippman. Learning mixture hierarchies. In *Neural Information Processing Systems (NIPS) Conference*, Denver, Colorado, September 1998.