PARSIMONIOUS VARIATIONAL-BAYES MIXTURE AGGREGATION WITH A POISSON PRIOR

Pierrick Bruneau, Marc Gelgon and Fabien Picarougne

Nantes university, LINA (UMR CNRS 6241), Polytech'Nantes
rue C.Pauc, La Chantrerie, 44306 Nantes cedex 3, France
INRIA Atlas project-team
firstname.surname@univ-nantes.fr

ABSTRACT

This paper addresses merging of Gaussian mixture models, which answers growing needs in e.g. distributed pattern recognition. We propose a probabilistic model over the parameter set, that extends the weighted bipartite matching problem to our mixture aggregation task. We then derive a variational-Bayes associated estimation algorithm, that ensure low cost and parsimony, as confirmed by experimental results.

1. INTRODUCTION

This paper addresses the issue combining several probabilistic mixture models of a single process. It focuses on the case input and output models are Gaussian mixture models (GMM), as this semi-parametric form is one of the most employed and versatile tool for modelling the density of multivariate continuous features. Alternatively, the same mixture scheme may be used for clustering data into Gaussian-shaped classes, in which case our task consists in a search for a consensus between data partitions. Whether for density estimation or clustering, our goal is to build a mixture that optimally describes the mixture ensemble. Both its parameters and number of components should be determined.

Aggregation of class models is a classical topic, both supervised (ensemble methods) and unsupervised. Growing interest comes from the transposition of existing statistical learning and recognition tasks onto distributed computing systems (cluster, P2P), which has motivated parsimonious model aggregation techniques [7], or sensor network.

A combined model could simply be obtained by a weighted sum of Gaussian mixtures, yet 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 preserves the ability to describe the underlying generative process. Parsimony is particularly important if such mixture combinations follow one after another.

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. Yet, this is interesting as a benchmark. 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, our technique seeks an optimal combination of Gaussian components, taking into account which mixture their arise from. 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. Lower cost, local optima have been sought in [5], 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, or a bipartite matching resolution between 2 sets of Gaussian components. As an alternative, a procedure akin to ascendent hierarchical clustering operating on Gaussian components is proposed in [8]. The search space considered in [9] 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.

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 [1] (VBEM hereafter). Yet, the standard use of VBEM is applied to data

THIS WORK WAS PARTLY FUNDED BY ANR SAFIMAGE (FRENCH MINISTRY OF UPPER EDUCATION AND RESEARCH) AND REGION PAYS DE LA LOIRE (MILES PROJECT)

in $\mathbb{R}^n$.

The central contribution of our paper is to define and 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. Besides conjugate priors on mixture parameters, the proposed probabilistic model includes a Poisson prior that discourages merges between Gaussian components supplied by the same mixture. This improves exploration of the search space significantly, i.e. reduces computation cost, over the simpler option of ignoring the origin of components in the reduction process.

Section 2 describes how the VBEM framework can be extended to parameter level, to conduct Bayesian clustering of Gaussian components. Section 3 extends this proposal by adapting the probabilistic model and deriving the associated estimation algorithm, including an initialization strategy. Section 4 provides experimental results and draws concluding remarks.

2. TRANSPOSING VARIATIONAL BAYESIAN TO PARAMETERS

Variational Bayesian EM framework is an iterative density modeling and clustering scheme based on a joint probability distribution function (*pdf*) defined over all variables and model parameters. Parameter modeling (a.k.a prior modeling) regularizes obtained estimates. More precisely, singularities are avoided, and the output number of significant (i.e. different from prior) components is as low as sensible [1, 3].

The joint distribution is defined by the following set of *pdf*s :

$$p(Z \mid \pi) = \prod_{n=1}^{N} \prod_{k=1}^{K} {\pi_k}^{z_{nk}} \tag{1}$$

$$p(X \mid Z, \mu, \Lambda) = \prod_{n=1}^{N} \prod_{k=1}^{K} \mathcal{N}(x_n \mid \mu_k, \Lambda_k^{-1})^{z_{nk}} \tag{2}$$

$$p(\pi) = \text{Dir}(\pi \mid \alpha_0) = C(\alpha_0) \prod_{k=1}^{K} \pi_k^{\alpha_0 - 1} \tag{3}$$

$$p(\mu, \Lambda) = p(\mu \mid \Lambda) p(\Lambda)$$
$$= \prod_{k=1}^{K} \mathcal{N}(\mu_k \mid m_0, (\beta_0 \Lambda_k)^{-1}) \, \mathcal{W}(\Lambda_k \mid W_0, \nu_0) \tag{4}$$

where X is a d-dimensional data set, Z the latent variables associating X with one of the K components in the model, $\theta = \{\theta_k\}$, $\theta_k = \{\pi_k, \mu_k, \Lambda_k\}$, $\pi = \{\pi_k\}$, $\mu = \{\mu_k\}$, $\Lambda = \{\Lambda_k\}$ are the model parameters, and $\{\alpha_k, \beta_k, \nu_k, W_k\}$ are the hyper-parameters (0 indices denoting prior values). Generally, prior parameters and hyper-parameters are set to uninformative values. Posterior values are obtained as an output.

This scheme was combined [4] with virtual samples [10] in order to merge and reduce a set of Gaussian mixtures. These mixtures might come from various sources and their addition involve high redundancy. Using this procedure (named VBMerge hereafter) we build a parsimonious and sensible representative.

Introducing virtual samples modifies the *pdf*s over X and Z. Let us remark that through virtual samples, the original distribution over X now depends solely on the input model $\theta' = \{\theta'_l\}$, $\theta'_l = \{\pi'_l, \mu'_l, \Lambda'_l\}$ (i.e. the set of Gaussian mixtures to reduce).

$$\ln p(X \mid Z, \mu, \Lambda) = \frac{N}{2} \sum_{k=1}^{K} \sum_{l=1}^{L} z_{lk} \pi'_l [\ln \det \Lambda_k - \text{Tr}(\Lambda_k {\Lambda'_l}^{-1}) - (\mu'_l - \mu_k)^T \Lambda_k (\mu'_l - \mu_k) - d \ln(2\pi)] \tag{5}$$

$$p(Z \mid \pi) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} = \prod_{l=1}^{L} \prod_{k=1}^{K} \pi_k^{N \pi'_l z_{lk}} \tag{6}$$

3. REDUCING A GAUSSIAN MIXTURE UNDER CONSTRAINTS

Let us consider several data repositories, each one being the source of a Gaussian mixture fitted on the available data. The previously proposed method [4] 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.

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 :

$$p(A|Z) = p(M = A^T Z) = \prod_{p=1}^{P} \prod_{k=1}^{K} \frac{e^{-1}}{(1 + m_{pk})!} \quad (7)$$

The term 1 is added for conveniency, and causes no loss of generality. The new global joint distribution is obtained by incorporating (7) to the set of *pdf*s { (3), (4), (5) , (6) }.

The classical VBEM algorithm (and its derivations) are based on two essential elements : updating estimates, and controlling convergence through a lower bound value. Readers are encouraged to see ([3] chapter 10) for thorough implementation details and theoretical justifications. We will focus on terms involving Z hereafter.

Let $q(Y)$ be a factorized variational distribution so that $q(Y) = \prod_j q_j(Y_j)$, and X a set of observed variables. Then the optimal factor q_j^* w.r.t. others kept fixed is obtained as :

$$\ln q_j^*(Y_j) = \mathbb{E}_{i \neq j}[\ln p(X, Y)] + \text{const} \quad (8)$$

Let us derive $\ln q^*(Z)$ for our modified joint distribution:

$$\ln q^*(Z) = \mathbb{E}_{\pi,\mu,\Lambda}[\ln p(A, X, Z, \pi, \mu, \Lambda)] + \text{const} \quad (9)$$

$$\ln q^*(Z) = \mathbb{E}_{\pi}[\ln p(Z|\pi)] + \mathbb{E}_{\mu,\Lambda}[\ln p(X|Z, \mu, \Lambda)] + \ln p(A|Z) + \text{const} \quad (10)$$

Following classic derivation ([3] chapter 10) adapted to the virtual sample context [10, 4], we obtain :

$$\begin{aligned} \ln q^*(Z) = N \sum_{l=1}^{L} \sum_{k=1}^{K} z_{lk} \pi'_l \\ [\ln \tilde{\pi}_k + \frac{1}{2} \ln \tilde{\Lambda}_k - \frac{d}{2} \ln(2\pi) - \frac{1}{2} [\frac{d}{\beta_k} \\ + \nu_k \text{Tr}(W_k \Lambda_l'^{-1}) + \nu_k (\mu'_l - m_k)^T W_k (\mu'_l - m_k)]] \\ - \sum_{k=1}^{K} \sum_{p=1}^{P} \ln(1 + m_{pk})! + \text{const} \end{aligned} \quad (11)$$

Or, equivalently :

$$\ln q^*(Z) = \sum_{l=1}^{L} \sum_{k=1}^{K} z_{lk} \ln \rho_{lk} - \sum_{k=1}^{K} \sum_{p=1}^{P} \sum_{i=0}^{m_{pk}} \ln(1+i) + \text{const} \quad (12)$$

with

$$\ln \rho_{lk} = N \pi'_l [\ln \tilde{\pi}_k + \frac{1}{2} \ln \tilde{\Lambda}_k - \frac{d}{2} \ln(2\pi) - \frac{1}{2} [\frac{d}{\beta_k} + \nu_k \text{Tr}(W_k \Lambda_l'^{-1}) + \nu_k (\mu'_l - m_k)^T W_k (\mu'_l - m_k)]] \quad (13)$$

$\tilde{\pi}_k = \mathbb{E}[\ln \pi_k]$, and $\tilde{\Lambda}_k = \mathbb{E}[\ln \det \Lambda_k]$

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 values r_{lk}). 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 :

$$q(Z) = q(z_{1.}) q(z_{2.}|z_{1.}) q(z_{3.}|z_{1.} z_{2.}) \dots q(z_{L.}|z_{1.} \dots z_{L-1.}) \quad (14)$$

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) [2].

3.1. 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 :

$$\ln q^*(z_{1.}) = \sum_{k=1}^{K} z_{1k} \ln \rho_{1k} - \sum_{k=1}^{K} \sum_{p=1}^{P} \ln(1 + a_{1p} z_{1k}) + \text{const} \quad (15)$$

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

$$\ln q^*(z_{1k}) = z_{1k} \ln \rho_{1k} - \sum_{p=1}^{P} \ln(1 + a_{1p} z_{1k}) + \text{const} \quad (16)$$

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

$$\ln q^*(z_{1k}) = z_{1k} \ln \rho_{1k} - \sum_{p=1}^{P} a_{1p} z_{1k} + \text{const} \quad (17)$$

$$\ln q^*(z_{1k}) = z_{1k} \ln \frac{\rho_{1k}}{e^{\sum_{p=1}^{P} a_{1p}}} + \text{const} \quad (18)$$

As each original component belongs to only one source,

$$\ln q^*(z_{1k}) = z_{1k} \ln \frac{\rho_{1k}}{e} + \text{const} \quad (19)$$

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).

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

Changing the rank of the restriction in eq. 15 leads to :

$$\ln q^*(z_{2.}|z_{1.}) = \sum_{k=1}^{K} z_{2k} \ln \rho_{2k} - \sum_{k=1}^{K}\sum_{p=1}^{P} \ln(1 + a_{1p}z_{1k}) - \sum_{k=1}^{K}\sum_{p=1}^{P} \ln(1 + a_{1p}z_{1k} + a_{2p}z_{2k}) + \text{const} \tag{20}$$

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

$$\ln q^*(z_{2k}|z_{1k}) = z_{2k} \ln \rho_{2k} - \sum_{p=1}^{P} a_{1p}z_{1k} - \sum_{p=1}^{P} (a_{1p}z_{1k} + a_{2p}z_{2k}) + \text{const} \tag{21}$$

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 :

$$\ln q^*(z_{2k}|z_{1k}) = z_{2k}(\ln \rho_{2k} - 1 - 2\delta_{a_{1\max},a_{2\max}}.\delta_{z_{1\max},k}) + \text{const} \tag{22}$$

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 :

$$\rho'_{jk} = \frac{\rho_{jk}}{e^{1+\sum_{i=1}^{j-1}(j-i+1).\delta_{a_{i\max},a_{j\max}}.\delta_{z_{i\max},k}}} \tag{23}$$

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

3.3. Modified bound

The bound is defined by a sum of expectations w.r.t the current variational distribution [3]. For it to be complete, we need to add a term associated to the distribution we introduced. The modified bound additional term is the following:

$$\mathbb{E}[\ln p(A \mid Z)] = \sum_{k=1}^{K}\sum_{p=1}^{P}\left[-1 - \sum_{i=0}^{\mathbb{E}[m_{pk}]} \ln(1+i)\right] \tag{24}$$

$$= -KP - \sum_{k=1}^{K}\sum_{p=1}^{P}\sum_{i=0}^{\mathbb{E}[m_{pk}]} \ln(1+i) \tag{25}$$

with

$$\mathbb{E}[m_{pk}] = \mathbb{E}\left[\sum_{l=1}^{L} a_{lp}z_{lk}\right] = \sum_{l=1}^{L} a_{lp}\mathbb{E}[z_{lk}] = \sum_{l=1}^{L} a_{lp}r_{lk} \tag{26}$$

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 $\Delta(\text{bound}) <$ threshold as a stop criterion, the only difference being that now Δ might be negative.

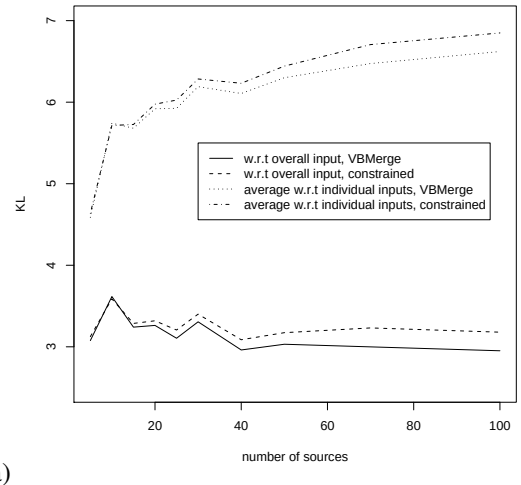
4. EXPERIMENTS AND CONCLUSION

We selected the 10 first categories in the Caltech-256 object category dataset [6], thus forming a set of 1243 images. We consider each image as a data source, and fit a Gaussian mixture over its pixel data ((L,a,b) color space, augmented by the pixel positions (x,y)). Obtained individual Gaussian mixtures comprise 18.1 components on average. We then randomly select x sources from the pool of images and perform the reduction. We measure :

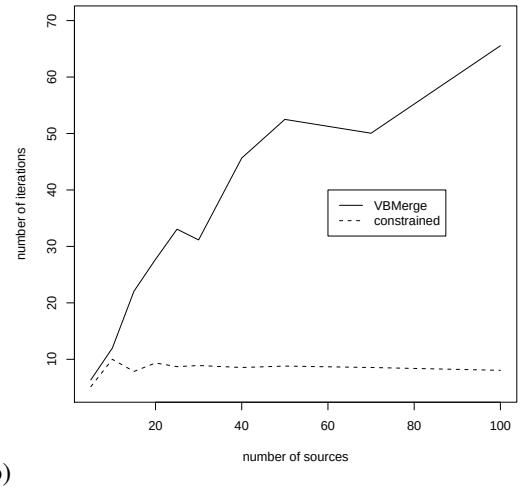
- the KL divergence of the reduced model w.r.t. the overall input (i.e. superimposed Gaussian mixtures), and w.r.t. each individual data source,
- the number of iterations before convergence,
- the number of significant components in the obtained model.

Results are reported in figure 1 for various numbers of input sources (i.e. images). From these results we draw the following conclusions :

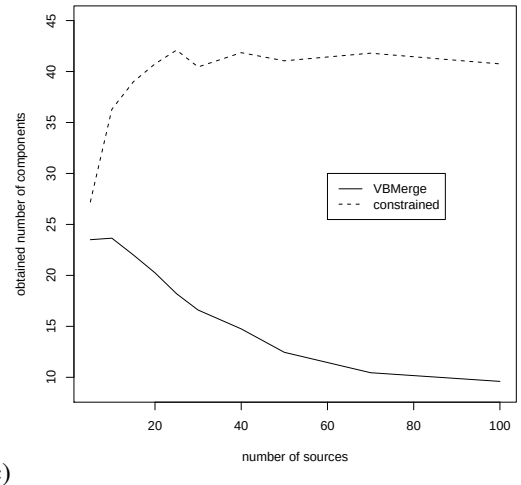
- the new technique provides reduced models that are equivalent to those of the baseline method, in the KL divergence sense. Occasionally, a slight loss was observed compared to the baseline method.
- as expected, our method prevents undesirable drastic reduction of the model : when reducing a set of 100 data sources (i.e. 1800 components on average), we obtain 40 components instead of 9. We see the "noise" effect on the results with VBMerge : as more and more components are considered, we introduce noise, which sometimes leads to simplistic reductions. The new technique thus supplies a trade-off between keeping the original structure and a slight signal loss.
- as the number of data sources increases, our method proves much faster than the baseline method. More precisely, the baseline method convergence becomes slower as the number of data sources augments. Intuitively, for the baseline method a lot of computational time is used to perform drastic reductions (leading to low improvements in terms of KL divergence), while in the new scheme, constraints allow us to stop the process as soon as possible.



(a)



(b)



(c) b

Figure 1: a : KL divergence of the reduced models w.r.t. the input sources, b : number of iterations before convergence, c : number of components in the reduced models.

Let us add a remark about algorithmic complexity : VBMerge iterations complexity is $o(L)$, while for the constrained derivation it is $o(L\ln(L))$. But on figure 1, we note that the number of iterations for VBMerge is linear w.r.t. L, while for our derivation it becomes $o(1)$. The loss for a single iteration is therefore largely outweighed by the gain in convergence time.

As a conclusion, we have proposed a novel scheme for merging Gaussian mixtures based on a variational-Bayes procedure. Introduction of the Poisson prior shows to significantly improve speed performance.

5. REFERENCES

[1] H. Attias. A variational Bayesian framework for graphical models. *Advances in Neural Information Processing Systems - MIT Press*, 12, 2000.

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

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

[4] P. Bruneau, M. Gelgon, and F. Picarougne. Parameter-based reduction of Gaussian mixture models with a variational-Bayes approach. In *19th International Conference on Pattern Recognition*, 2008.

[5] J. Goldberger and S. Roweis. Hierarchical clustering of a mixture model. *Advances in Neural Information Processing Systems - MIT Press*, 17, 2004.

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

[7] 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.

[8] A. Runnalls. A Kullback-Leibler approach to Gaussian mixture reduction. *IEEE Trans. on Aerospace and Electronic Systems*, 2006.

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

[10] N. Vasconcelos and A. Lippman. Learning mixture hierarchies. *Advances in Neural Information Processing Systems - MIT PressNeural Information Processing Systems*, II:606–612, 1998.