Infinite Gaussian Mixture Modeling with an Improved Estimation of the Number of Clusters

Avi Matza and Yuval Bistritz

The school of electrical engineering, Tel-Aviv University, Tel Aviv
avimatza@mail.tau.ac.il, bistritz@tauex.tau.ac.il

Abstract

Infinite Gaussian mixture modeling (IGMM) is a modeling method that determines all the parameters of a Gaussian mixture model (GMM), including its order. It has been well documented that it is a consistent estimator for probability density functions in the sense that, given enough training data from sufficiently regular probability density functions, it will converge to the shape of the original density curve. It is also known, however, that IGMM provides an inconsistent estimation of the number of clusters. The current paper shows that the nature of this inconsistency is an overestimation, and we pinpoint that this problem is an inherent part of the training algorithm. It stems mostly from a "self-reinforcing feedback" which is a certain relation between the likelihood function of one of the model hyperparameters (α) and the probability of sampling the number of components, that sustain their mutual growth during the Gibbs iterations. We show that this problem can be resolved by using *informative* priors for α and propose a modified training procedure that uses the inverse χ^2 for this purpose. The modified algorithm successfully recovers the "known" order in all the experiments with synthetic data sets. It also demonstrates good results when compared to other methods used to evaluate model order, using real-world databases. Furthermore, the improved performance is attained without undermining the fidelity of estimating the original PDFs and with a significant reduction in computational cost.

Introduction

The infinite Gaussian mixture model (IGMM) is a method for modeling multimodal distributions in a non-parametric Bayesian framework. It aims to provide a fully generative Gaussian mixture model (GMM), for arbitrary data, without a presumed number of Gaussians. The fact that IGMM is able to simultaneously infer both the number of Gaussian components and their parameters, makes it particularly attractive for clustering applications, where the primary goal is to determine the number of clusters in some unexplored data. The IGMM has been used for this purpose in bioinformatics (Medvedovic and Sivaganesan 2002), astronomy (Shin, Sekora, and Byun 2009), speech (Niekum and Barto 2011), (Kamper et al. 2014) and various other fields.

The IGMM parameters are typically obtained by unsupervised Bayesian inference techniques, in which a Dirichlet prior is assigned to the weights of the mixture. This approach falls within a family of modeling methods known as Dirichlet process mixture (DPM) modeling, see Ferguson (1973) and Antoniak (1974), that was subsequently applied to modeling, clustering and classification (Bouguila and Ziou 2009; Görür and Rasmussen 2010; Hu et al. 2013; Davy and Tourneret 2010; Dai and Storkey 2014; Meilă and Chen 2016) and others.

As a Dirichlet process mixture of Gaussians, the IGMM is a consistent estimator of probability density functions (PDF). Namely, given enough training data from a sufficiently regular PDF, the DPM model converges to the shape of the density (Ghosal, Ghosh, and Ramamoorthi 1999; Miller and Harrison 2013). However, consistency in estimating the shape of a density function does not necessarily mean a proper estimation of the number of components (the *order*) of the mixture (Miller and Harrison 2013, 2014, 2018; Lu 2017; Harshavardhan and Sreenivas 2010).

This inconsistency in order estimation is undesirable in density estimation tasks, as it means wasting computation effort on exploring insignificant regions in the parametric space and creating a model with higher than necessary number of parameters. It becomes unacceptable in clustering applications where the main goal is to estimate the number of clusters in some unfamiliar data.

It has been shown by West (1992) and Escobar and West (1995), following the work of Antoniak (1974), that the number of components in a DPM is affected mainly by a specific parameter of the Dirichlet prior distribution called the concentration parameter and denoted usually by α.

In spite of its significance, the selection of α has not received enough attention. Some studies simply set α to some adhoc value. For example, it was set to 1 in Medvedovic and Sivaganesan (2002) and in Shin, Sekora, and Byun (2009). Some other works, like Escobar and West (1992) and Rasmussen (2000), selected α with a prior distribution having some fixed hyper-parameters. For some more elaborated hyper-parameters selection methods of priors for α see Dorazio (2009) and Murugiah and Sweeting (2012).

This paper proposes a new training algorithm for IGMM, one that achieves quite accurate estimation of the "correct" number of components in synthetically generated data as well as real-world databases. First we present the baseline IGMM and explore the relationship between the concentration parameter α of the prior distribution for the model weights and the inferred model order. We expose a mechanism that sustains the increase in the number of weights sampled by the Gibbs algorithm and shortly discuss a few possible options to mitigate it. Section 4 presents the proposed modified algorithm followed by some evaluation tests results. Additional advantages of the modified IGMM includes: (i) flexibility to calibrate the algorithm roughly toward the anticipated order range using a single adjustable parameter that is shown to need only crude adjustment. (ii) good estimation of the PDF shape and (iii) reduced cost of computation.

The Baseline IGMM

The IGMM training algorithm as proposed by Rasmussen in (2000) and implemented by Mandel (2005), has the following hierarchical structure

Ground level - modeling the data. Consider a data set $\mathbf{Y} = \{y_1, y_2, \ldots, y_N\}$ of real scalars y_n that can be modelled by a mixture of K (finite yet unknown) Gaussians

$$p(y_n|\mathbf{M}_K) = \sum_{k=1}^{K} \pi_k \mathscr{N}(y_n|\mu_k, s_k^{-1}). \tag{1}$$

In the above we use $\mathscr{N}(\cdot)$ to denote a Gaussian (normal) PDF with mean μ_k and precision s_k and $\pi_k > 0$ are weights such that $\sum_{k=1}^{K} \pi_k = 1$. The collection of parameters of this GMM is denoted by $\mathbf{M}_K$ viz.

$$\mathbf{M}_K = \{\pi_k, \mu_k, s_k, k = 1, \ldots, K\} \tag{2}$$

First level - modeling the priors. The means, precisions and weights are modeled by

$$\mu_k|\lambda, r \sim \mathscr{N}(\lambda, r^{-1}) \tag{3}$$

$$s_k|\beta, w \sim \mathscr{G}(\beta, w^{-1}) \tag{4}$$

where $\mathscr{G}(\cdot)$ denotes the Gamma distribution. The mixing weights are modeled by a Dirichlet prior as follows

$$\pi_1, \ldots \pi_K|\alpha \sim \mathscr{D}\left(\frac{\alpha}{K}, \ldots, \frac{\alpha}{K}\right) = \frac{\Gamma(\alpha)}{\Gamma(\frac{\alpha}{K})^K} \prod_{k=1}^{K} \pi_k^{\alpha/K-1} \tag{5}$$

Second level - modeling the hyper parameters. The parameters λ and r (common to all the means) are modeled by

$$\lambda \sim \mathscr{N}(\mu_y, s_y^{-1}) \quad , \quad r \sim \mathscr{G}(1, s_y) \tag{6}$$

where μ_y and s_y denote the mean and precision of $\mathbf{Y}$ and are regarded as a set of given scalars. The parameters of $\mathscr{G}(\cdot)$ (common to all the precisions) are modeled by

$$\beta^{-1} \sim \mathscr{G}(1, 1) \quad , \quad w \sim \mathscr{G}(1, s_y^{-1}) \tag{7}$$

and the inverse of α is modeled by

$$\alpha^{-1} \sim \mathscr{G}(1, 1). \tag{8}$$

Actual model parameters are obtained by using the Gibbs sampling algorithm. The Gibbs sampling is a Markov chain Monte Carlo (MCMC) type algorithm that is used to obtain a sequence of samples for the parameters of a multivariable distribution. It is often used when the distribution does not have a closed form expression or when it does not render itself easily to direct sampling. These samples are obtained from the marginal posterior distributions of each parameter, in a sequential manner that can be shown to converge (Casella and George 1992). And then they are used to approximate the joint distribution. In the following we refer to the model and algorithm outlined above as the baseline IGMM. In the supplementary material we use synthetic databases to demonstrate that the baseline IGMM severely overestimates the number of components in a mixture (its *order*).

Hyper Parameter Statistics

Although several parameters have an impact on the number of estimated mixture components (K), the most significant parameter is the concentration parameter α in the Dirichlet process for the mixture weights in (5).

The probability of observing k distinct values when sampling N times a Dirichlet process with concentration parameter α (corresponding to the probability that there are K different values of π) can be expressed by

$$p(k|\alpha, N) = c_n(k) N! \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha + N)} \; ; k \in \{1, 2, \ldots, N\} \tag{9}$$

where $\Gamma(\cdot)$ is the Gamma function and $c_n(k)$ is $P(k|\alpha, N)$ at $\alpha = 1$, see (West 1992) and (Antoniak 1974). $c_n(k)$ can be computed by taking the absolute value of the Stirling numbers of the first kind $|S_n^{(k)}|$ that for large N can be approximated by

$$|S_n^{(k)}| \approx \frac{(N-1)!}{(k-1)!}(\gamma + \log(N))^{k-1} \tag{10}$$

where γ is Euler's constant. Therefore, the distribution of k given α and assuming large N is approximately proportional to

$$p(k|\alpha, N) \underset{\sim}{\propto} \frac{\alpha^k (\gamma + \log(N))^{k-1}}{(k-1)!} \frac{N!(N-1)!\Gamma(\alpha)}{\Gamma(\alpha + N)} . \tag{11}$$

Figure 1 presents the shapes of $p(k|\alpha, N)$ for $\alpha = 1, \ldots, 8$ (with N set arbitrarily to 60 and curves scaled by a proportion constant for convenience). In this figure, the left most curve (marked with "+"s) corresponds to $\alpha = 1$ and the following curves from left to right correspond to increasing values for α. It is clear that, indeed, higher values of α are associated with higher values of k.

Let us examine the opposite direction. Namely, let us see how k affects α. The posterior distribution of α given k (and N) is proportional to the conditional likelihood of α, (designated as $\mathcal{L}(\alpha|k, N)$) times some prior distribution assigned to it ($p_r(\alpha)$),

$$p(\alpha|k, N) \propto \mathcal{L}(\alpha|k, N) p_r(\alpha). \tag{12}$$

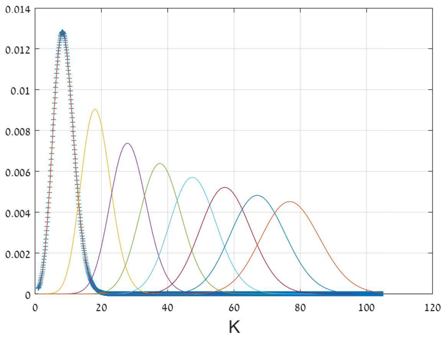

Figure 1: Approximated probability of k ($\alpha = 1, \ldots, 8$).

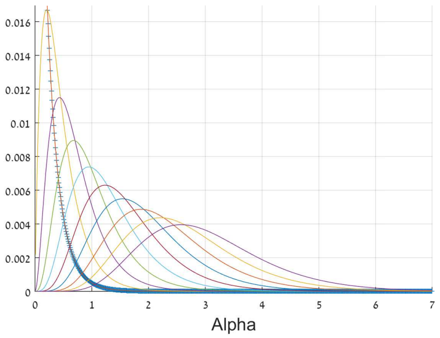

Figure 2: The likelihood of α for $k = 1, \ldots, 10$.

We need to examine the impact of each of the two terms in this product. The asymptotical likelihood of α given in (11), suggests the following approximation, per each k (and a fixed N)

$$\mathcal{L}(\alpha|k, N) \underset{\sim}{\propto} \alpha^k \frac{\Gamma(\alpha)}{\Gamma(\alpha + N)}. \tag{13}$$

Figure 2 illustrates the shape of the curves for $k = 1, \ldots, 10$ (scaled by a constant). The leftmost graph (also marked by "+") corresponds to $k = 1$ and the following curves (left to right) represent the likelihood of α for the subsequent values of k. The curves show that as k increases, the likelihood of sampling higher values for α increases as well.

So we see that increased values of α increase the probability of sampling higher values for k and that as k increases, the likelihood of sampling larger values for α increases as well. The combination of these two observations identifies a "self-reinforcing feedback" mechanism that sustains successive overestimation of k. A more formal analysis of the behaviour of $p(k|\alpha)$ and $\mathcal{L}(\alpha|k, N)$ is presented in the supplementary material.

This raises the question of how this "self-reinforcing feedback" mechanism might be mitigated in order to avoid the order overestimation it "causes". We explored various plausible remedies. For example in the supplementary material we examine changing the priors for α and it becomes apparent that the specific shape of the prior distribution does not have a significant impact on the overestimation issue. A well established fact in Bayesian estimation is that the support range of a prior has a significant impact on the posterior distribution. This means that setting an informative prior for α could be useful thorough selecting a prior such that α is confined to some restricted range of values. If, for example, we could limit α to values not higher than $\alpha = 1$, then, as illustrated in Figure 1, the values sampled for k will reside mostly between 1 and 20 (and most likely around 10). On the other hand, if the prior distribution admits higher values, say up to $\alpha = 8$, then all the curves in Figure 1 represent viable options with the consequence that values of k of 100 and beyond may be sampled. In other words, selecting an informative prior with limited admissible range of α may provide the remedy to the order overestimation.

Although informative priors can be assign using many distributions (including ones that were used before), when selecting a practical distribution we have a few additional requirement, it should (i) lead to conditional posterior distributions of model parameters that admits easily to Gibbs sampling (ii) admit easy manipulation of the prior information. In this paper we offer the $inverse\ \chi^2$ PDF given by

$$p_{inverse\ \chi^2}(\alpha|\theta) = \frac{\alpha^{(-\frac{\theta}{2}-1)} \exp\left(-\frac{1}{2\alpha}\right)}{\Gamma(\frac{\theta}{2}) 2^{\frac{\theta}{2}}} \quad , \quad \alpha > 0. \tag{14}$$

The $inverse\ \chi^2$ was chosen as it is a special case of the inverse Gamma PDF and thus retains the tractability of many of the posterior parameters for the Gibbs procedure. Furthermore, while it does not have a limited support, it involves only a single calibration hyper parameter (θ) which considerably simplifies prior regulation when compared for example with PDFs having two parameters as suggested for example in Dorazio (2009) and in Murugiah and Sweeting (2012).

The Modified IGMM

In this section we consolidate the insights and observations presented so far into proposing a modified IGMM. Assume we have a set of observed data $\mathbf{Y} = \{y_1, y_2, \ldots, y_N\}$ obtained from some unknown number of sources. This data can be modelled by the GMM of order K in (1) with a set of parameters $\mathbf{M}_K$ as in (2). The means μ_k are normally distributed (3) with mean λ and precision r common to all the mixture components.

The λ and r are hyper parameters modeled by normal and Gamma distribution depending on mean μ_y and the precision s_y obtained from the data $\mathbf{Y}$. We assign to $\mathbf{Y}$ a set of stochastic indicator variables $\mathbf{C} = \{c_1, c_2, \ldots, c_N\}$ where $c_n \in \{1, \ldots, K\}$ indicates which component "generated" y_n. Then the conditional posterior distribution for the means becomes

$$p(\mu_k|\mathbf{Y}, \mathbf{C}, s_k, \lambda, r) = \mathcal{N}\left(\frac{\bar{y}_k \ell_k s_k + \lambda r}{\ell_k s_k + r}, \frac{1}{\ell_k s_k + r}\right). \tag{15}$$

In the above ℓ_k, $\sum_{k=1}^{K} \ell_k = N$, denotes the number of data points that belong to mixture k (the occupation number) and

$\overline{y}_k$ denotes their mean. The conditional posterior distributions for the hyper parameters become

$$\begin{aligned}&p(\lambda|\mu_1,\ldots,\mu_K,r) = \\ &\mathscr{N}\left(\tfrac{\mu_y s_y + r\sum_{k=1}^{K}\mu_k}{s_y+Kr}, \tfrac{1}{s_y+Kr}\right)\end{aligned} \tag{16}$$

and

$$\begin{aligned}&p(r|\mu_1,\ldots,\mu_K,\lambda) = \\ &\mathscr{G}\left(K+1, \left(\tfrac{1}{K+1}(s_y^{-1} + \textstyle\sum_{k=1}^{K}(\mu_k-\lambda)^2)\right)^{-1}\right).\end{aligned} \tag{17}$$

The precision parameters s_k are modelled by Gamma distribution with second level hyper parameters β and w as in (4). The parameters β and w are assumed to be common to all components and modelled as in (7). Consequently, the conditional posteriors for s_k and w are

$$\begin{aligned}&s_k|\mathbf{Y},\mathbf{C},\mu_k,\beta,w \sim \\ &\mathscr{G}\left(\beta+\ell_k, \left(\tfrac{1}{\beta+\ell_k}(w\beta + \textstyle\sum_{n:c_n=k}(y_n-\mu_k)^2)\right)^{-1}\right) \\ &w|s_1,\ldots,s_K,\beta \sim \\ &\mathscr{G}\left(\beta K+1, \left(\tfrac{1}{\beta K+1}(s_y + \beta\textstyle\sum_{k=1}^{K}s_k)\right)^{-1}\right).\end{aligned} \tag{18}$$

In the baseline algorithm β posterior distribution did not have a form of a standard density functions thus the training algorithm resorted to adaptive rejection sampling (ARS) (Gilks and Wild 1992). When driving the modified IGMM posterior distributions we ended up with a similar issue. We studied a few possible distribution functions and concluded that β posterior distribution can be best approximated by using the standard generalised inverse Gaussian (GIG) distribution, given by

$$\begin{aligned}&p_{GIG}(x|\psi,\rho,\xi) = \\ &\left(\tfrac{\psi}{\rho}\right)^{\frac{\xi}{2}}\tfrac{x^{\xi-1}}{2\mathcal{K}_{\xi}(\sqrt{\psi\rho})}\exp\left(-\tfrac{1}{2}\left(\tfrac{\rho}{x}+\psi x\right)\right)\end{aligned} \tag{19}$$

where $\mathcal{K}_{\xi}(\cdot)$ denotes the modified Bessel function of the second kind and the corresponding β posterior approximation is

$$\begin{aligned}&p(\beta|s_1,\ldots,s_K,w) \approx \\ &p_{GIG}(\beta|\textstyle\sum_{k=1}^{K}(s_k w - ln(s_k w)), 1, \tfrac{K-1}{2}).\end{aligned} \tag{20}$$

The full derivation of the posterior and the resemblance to the *GIG* distribution are provided in the supplementary material.

Remaining is the consideration of the weights π_k. These parameters are Dirichlet distributed as in (5) with the hyper parameter α, the so-called *concentration parameter* that we investigated in the previous section. Assuming that $p(\alpha^{-1})$ has χ^2 distribution, then α has an inverse chi square distribution $p_{inverse\ \chi^2}(\alpha)$ defined in (14) which results in

$$\begin{aligned}&p(\alpha|\theta,K) \propto \frac{\alpha^{(-\frac{\theta}{2}-1)}\exp\left(-\frac{1}{2\alpha}\right)}{\Gamma(\frac{\theta}{2})2^{\frac{\theta}{2}}}\frac{\alpha^K\Gamma(\alpha)}{\Gamma(N+\alpha)} \propto \\ &\frac{\alpha^{(K-\frac{\theta}{2}-1)}\Gamma(\alpha)\exp\left(-\frac{1}{2\alpha}\right)}{\Gamma(N+\alpha)}\end{aligned} \tag{21}$$

(the full derivation can be found in the supplementary material).

At this point, the Gibbs algorithm encounters an obstacle. We notice that α can't be sampled directly because it does not have a closed-form PDF. In order to avoid using ARS, we followed Escobar and West in (1995), note that the ratio of the Gamma functions in (21) can be replaced with

$$\frac{\Gamma(\alpha)}{\Gamma(N+\alpha)} = \frac{(\alpha+N)\mathcal{B}(\alpha+1,N)}{\alpha\Gamma(N)} \tag{22}$$

where $\mathcal{B}(\alpha+1,N)$ is the standard Beta function

$$\mathcal{B}(u,v) = \int_0^1 z^{u-1}(1-z)^{v-1}dz\,. \tag{23}$$

After a few algebraic manipulations (provided in the supplementary material) we get

$$\begin{aligned}&p(\alpha|z,K) \propto \Psi_z p_{GIG}(\alpha|-2ln(z),1,K-\tfrac{\theta}{2})+ \\ &(1-\Psi_z)p_{GIG}(\alpha|-2ln(z),1,K-\tfrac{\theta}{2}-1).\end{aligned} \tag{24}$$

This expression is suitable for Gibbs sampling, eliminating the previous need for a secondary sampling (Gilks and Wild 1992). The ratio of the weights, defined via Ψ_z, is calculated using

$$\frac{\Psi_z}{1-\Psi_z} = N\sqrt{-2ln(z)}. \tag{25}$$

In order to complete the sampling of α, we still need the conditional marginal PDF of the auxiliary parameter z. It can be noticed from (23) that z can be considered as

$$p(z|\alpha) \propto z^{\alpha}(1-z)^{N-1}\,, \tag{26}$$

which corresponds to a standard Beta distribution with $\frac{1}{\mathcal{B}(\alpha+1,N)}$ as the proportion constant. Thus, the auxiliary parameter z can also be obtained by Gibbs sampling. Up until this point, K, which represents the number of components in the model, was assumed to be a finite value. It can be shown that the conditional posteriors of all model parameters (prepared for the Gibbs sampler), except the set of indicators $\mathbf{C}$, remain the same when K in their expressions represent, instead, the number of *occupied* mixture components (rather than *all* the possible components). The conditional posterior of the indicators can be obtained using some algebraic manipulations and the expression for the posterior probably of attaining new components is

$$\begin{aligned}&p(c_n = k|\mathbf{C}_{-n},\alpha,\mu_k,s_k) \propto \\ &\tfrac{\ell_{-n,k}}{N-1+\alpha}s_k^{\frac{1}{2}}\exp\left(-\tfrac{s_k(y_n-\mu_k)^2}{2}\right).\end{aligned} \tag{27}$$

The combined probability for all the other (not n) indicators to belong to other mixture components is

$$\begin{aligned}&p(c_n \neq c_{n'},\forall n \neq n'|\mathbf{C}_{-n},\alpha,\lambda,r,\beta,w) \propto \\ &\tfrac{\alpha}{N-1+\alpha}\textstyle\int p(y_n|\mu_k,s_k)p(\mu_k,s_k|\lambda,r,\beta,w)d\mu_k ds_k\end{aligned} \tag{28}$$

where $\mathbf{C}_{-n}$ presents all the indicators excluding c_n, $\ell_{-n,k}$ presents the number of data samples, excluding y_n, that are associated with component k of the mixture and the subscript $-n$ stands for "all indices except n".

Unfortunately, the integral for $p(c_n \neq c_{n'})$ is not analytically tractable. Following Neal (2000) we approximate it in two steps as follows. First we sample priors of μ_k and s_k and use them to compute y_n probabilities. Next we use these probabilities to draw new indicators from a multinomial distribution. This procedure provides a convenient way to generate new mixture components. Finally, it is noted that the correspondence between the indicators, π_k and ℓ_k (the number of data points that belong to the component k) is given by

$$p(\pi_k|\mathbf{C}) = \frac{\ell_k}{\sum_{k=1}^{K} \ell_k}. \tag{29}$$

The Proposed Procedure

In the following we formalize the training procedure, using the modifications introduced so far. Note that the procedure has a free parameter, the θ of the inverse chi square in (25) that can be used to crudely adjust the expected number of components. Notation wise, we use variables without a superscript to designate the input values at the beginning of the iteration step, and the superscript ν ("new") for the value that the iteration assigns to them.

- **Initialization.** The algorithm begins with a single mixture component. All data points are allocated to $K = 1$ thus $c_n = 1$ (for $n = 1, \dots, N$), $\ell_1 = N$ and $\overline{y}_1 = \overline{\mathbf{Y}}$ (i.e. average over all the data). Calculate the s_y and μ_y for the available data as defined in (6). Sample initial values for the parameters for λ, r, w, β, s_k, α and z using (6),(7),(4),(14) and (26). Note that (14) requires an appropriate setting of θ. This value should be selected such that it corresponds to the number of components expected in the model.
- **An $M \rightarrow M^\nu$ iteration.**

  Step 1. Sample μ_k, λ, r, s_k, w, β from their posterior distributions using equations (15-18,20) and assign them to μ_k^ν, λ^ν, r^ν, s_k^ν, w^ν, β^ν respectively.

  Step 2. Sampling of the hyper parameter α^ν:
  - Calculate the ratio $\Psi_z^\nu/(1-\Psi_z^\nu)$ using (25) and determine Ψ_z^ν.
  - Sample an auxiliary parameter ϱ from a uniform distribution [0:1].
  - If $\varrho < \Psi_z^\nu$, use the parameters from the first p_{GIG} distribution in (24) to sample α^ν. Otherwise use the parameters from the second distribution for the same purpose.

  Step 3. Sample auxiliary parameter z^ν using (26) and calculate $\ell_{-n,k}^\nu$.

  Step 4. Obtaining probabilities for a new set of indicators:

  Determine intermediate $\mathbf{C}_e^{tmp}$ from the posterior probabilities $p(c_n = k|\mathbf{C}_{-n}, \alpha, \mu_k, s_k)$ in (27) (for all $k \in \{1, \dots, K\}$).

  Next, calculate the probability of $\mathbf{C}_{oth}^{tmp}$ (the remaining entries), in two steps:
  - First, use the prior distributions (3) and (4) to draw temporary mixture parameters μ^{tmp}, s^{tmp}. Then use them to calculate the probabilities $p(y_n|\mu^{tmp}, s^{tmp})$ for each sample (the probability of y_n to come from a new mixture component).
  - Next, multiply the results by the first part of (28) to obtain the conditional posterior probabilities for getting all other entries, $\mathbf{C}_{oth}^{tmp}$.

  Finally, allocate the probabilities of all the other entries to one additional component.

  Step 5. Sampling a new set of indicators $\mathbf{C}^\nu = \{c_1^\nu, \dots c_N^\nu\}$:

  Use the probabilities for $\mathbf{C}_e^{tmp}$ and $\mathbf{C}_{oth}^{tmp}$ to randomly draw N indicators (c_n) from a multinomial distribution, one for each data sample, thus obtaining an updated $\mathbf{C}^\nu$.

  Step 6. Adding or discarding mixture components:
  - If data points were allocated to the new extra component, increase the value of K by one. Use (μ^{tmp}, s^{tmp}) drawn earlier as the mean and precision of the new mixture component.
  - If one or more existing mixture components were not assigned data points in the current cycle, remove them from the mixture and reduce the value of K accordingly.

  Denote the resulting updated number of mixture components K^ν.

  Step 7. For each $k \in \{1, \dots, K^\nu\}$, calculate ℓ_k^ν and use (29) to calculate π_k^ν.

  Step 8. Update all parameters: Rename all the new (ν) parameters as current parameters. Namely, remove $^\nu$ from all parameters $\mathbf{M}^\nu \rightarrow \mathbf{M}$ (as defined in (2)), and similarly $\{\lambda^\nu, r^\nu, \beta^\nu, w^\nu, \alpha^\nu, \mathbf{C}^\nu, K^\nu, \ell_k^\nu, z^\nu\} \rightarrow \{\lambda, r, \beta, w, \alpha, \mathbf{C}, K, \ell_k, z\}$.

  Go to Step 1
- **Termination.** Steps 1-8 present one iteration cycle that produces one new sampled GMM with its parameters $\mathbf{M}_K$ (2). This cycle is repeated till enough sampled models are obtained (in our tests we used 12,000 samples). Afterward, MAP is used to determine the value of K. If the goal is to determine the number of clusters in the data then this last MAP step terminates the algorithm. Else, if the task also requires estimation of the PDF, some additional MAP decisions may be applied to extract the best GMM parameters from all the models with the relevant K.

It is important to note that the use of χ^2 prior for α should not be regarded as providing by itself the total cure for the overestimation issue. However, the modified algorithm provides better means to mitigate this problem. Further discussion regarding the impact of the selected θ on model order estimation as well as a general method on how to judicially select it is provided in the next section.

Evaluating the Modified IGMM

In order to evaluate the performance of the modified algorithm we compare its results to the ones obtained by the baseline algorithm. This is done by executing both with 4 synthetic data sets designated as p_1, p_2, p_3 and p_4. Each

Data set	True K	Base K	Base KL$_{mc}$	Modif. K	Modif. KL$_{mc}$
p_1	6	23 (21)	0.027	6 (94)	0.003
p_2	3	64 (38)	0.082	3 (84)	0.041
p_3	20	43 (32)	13.37	20 (62)	13.51
p_4	30	63 (36)	39.95	30 (61)	36.28

Table 1: Comparison the baseline and the modified IGMM

data set contained 10,000 samples generated by randomly sampling four GMMs as follows:

- The p_1 data set was created using a GMM as in (1) of order $K = 6$ with $\pi = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$, $\mu = (-15, -8, -3, 3, 8, 15)$ and $s^{-1} = (1, 1, 1, 1, 1, 1)$.
- The p_2 data set was created using a GMM of order $K = 3$ with $\pi = (\frac{3}{10}, \frac{1}{2}, \frac{1}{5})$, $\mu = (-1, 0, 3)$ and $s^{-1} = (\frac{3}{2}, 1, \frac{1}{2})$.
- The p_3 data set was created using a GMM of order $K = 20$ with weights π_k, drawn from a uniform distribution $U(0, 10)$ (scaled to sum up to 1), means μ_k drawn from $\mathscr{N}(0, 30)$ and precision s_k drawn from $U(0, 10)$.
- The p_4 data set was created using a GMM of order $K = 30$ with randomly generated parameters like in data set p_3.

Both algorithms were executed 500 times for each data set, and K that was obtained most, and its % (in the parenthesis) are presented in the Table 1. Along with order estimation, the symmetric Kullback-Leibler divergence between the known GMMs and the corresponding inference is presented. It was evaluated through Monte Carlo simulation using one of the models with an appropriate K. It is clear that in all the experiments, the modified algorithm excels in estimating the order of the models. Furthermore the resulting models are also adequate as was verified by both the low divergence scores presented above, and reaffirmed by a visual inspection of their correspondence to the original histograms, presented in the supplementary material.

The Impact of the Calibration Parameter

One of the reasons the *inverse* χ^2 PDF was selected as a prior for α was the fact that it depends on a single parameter θ (14) and as such can be easily manipulated. The mean value of $p_{inverse\ \chi^2}(\alpha|\theta)$ is given by $\frac{1}{\theta-2}$ (for $\theta > 2$). This fact can be used as a rule of thumb to set a judicious choice for θ when there is some vague prior knowledge regarding the expected number of clusters (mixture components) in the explored database. In this subsection we show that the specific setting of θ is not critical and that order estimation results are quite tolerant in this regard.

We executed the modified training algorithm using p_1, p_2, p_3, p_4 with different values of θ. For each data set and per every selection of θ we obtained 50,000 samples of K. Figure 3 presents the results in the form of box-plots. The x axis presents the different values of θ and the y axis the order sampled (K) for these values. For each box, the central mark presents the median, and its floor and ceiling its 25th and 75th percentiles, respectively. The 2 vertical lines indicate the range in which the results are still considered valid, and outliers are plotted individually using the '+' symbol.

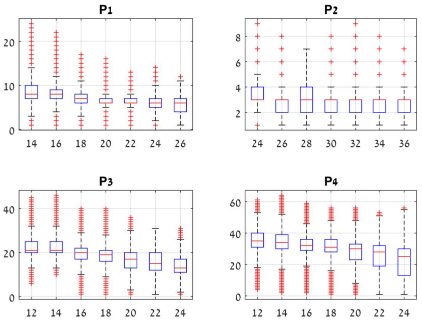

Figure 3: Box plots for p_1 to p_4 with various values of θ.

For p_1, it is seen that for $\theta = 14, 16$, K is centered around 8, for $\theta = 18$ K is centered around 7 and for $\theta = 20, 22, 24, 26$, K is centered around 6. Thus the correct value $K = 6$ resides, in all cases, between the 25th and 75th percentiles and for $\theta = 20, 22, 24, 26$ it forms the central value. For p_2, all the values of θ around 30 (± 6) resulted in the correct value $K = 3$ as the central value. For p_3 the correct value $K = 20$ is between the 25th and 75th percentiles for values of θ ranging from 12 to 22. It is the central value for $\theta = 16$ and is estimated just slightly higher ($K = 21$) for $\theta = 12, 14$. Similar behavior is observed for p_4, were for small values of θ (between 12 and 18), K is estimated higher than its actual value ($K = 30$) and for higher θ, the central K values are less than 30.

Since the papers main concern is the “self-reinforcing feedback” issue, the resulting overestimation, and how it can be resolved, we used in the previous tests θ values that demonstrate that this overestimation can indeed be mitigated. One should note, however, that a non-judicious selection of θ may result not only in overestimation but also in underestimation. Fortunately, the results presented in this section suggest that one needs only a vague prior knowledge regarding the expected number of clusters in the explored data, set θ accordingly and the modified algorithm will, most likely, converge to the correct order.

Reduced Computational Cost

It is worth noting, that the proposed algorithm also offers a significant reduction in computational cost. The reduction stems from obtaining closed form representations for some of the posteriors. These representations eliminate the need for a secondary sampling (using ARS), that was necessary in the baseline algorithm. This circumvents several hundreds of iterations per the creation of each model.

We used the same test environment for both the baseline and the modified IGMM in all our experiments. They were executed using *matlab* on a core i7 cpu at 2.8 GHz with 16GB ram. Both the baseline and the modified algorithms were executed several times using the reported four synthetic data sets. The average running time (per data set) measured for the baseline algorithm was approximately 996 seconds where the average time for the modified algorithm was 223 seconds. Namely, the proposed procedure is about 4 times faster than the baseline algorithm.

Comparison with Other Methods

In the previous sections we demonstrated that the modified IGMM was able to avoid over estimation and successfully determined model order in several test scenarios. Let us compare our performance with a few other order estimation methods. Most often, GMM is trained by the expectation maximization (EM) algorithm, so our first two methods are drawn from this realm. EM inference of a GMM typically requires an initial setting of K, in order to determine the best model order we executed the training algorithm multiple times with different values of K ($K = 2, \ldots, 25$) and used two methods to select the "best" model among them.

The first method is the Akaike Information Criteria (AIC), in Akaike (1973), expressed as

$$AIC(\lambda_k) = -2\mathcal{L}(\mathbf{Y}|M_k) + 2\nu_k, \quad (30)$$

where $\mathbf{Y}$ is the training data, $\mathcal{L}(\mathbf{Y}|M_k)$ is the log likelihood function and ν_k is the number of free parameters in the model. The second is the minimum description length (MDL) criterion which can be seen in Rissanen (1978)

$$MDL(\lambda_k) = -\mathcal{L}(\mathbf{Y}|M_k) + \frac{\nu_k log(N)}{2}, \quad (31)$$

were N represent the number of available training samples.

A third method we use for comparison is based on competitive learning which is a form of unsupervised learning were components compete each other for the right to be updated. One of the main competitive learning algorithms applied to GMM is the rival penalization competitive learning (RPEM) presented by Cheung (2005). We follow here the procedure presented in Matza and Bistritz (2011).

Our comparison is based on a clustering task of 3 well know databases, the galaxy, enzyme and acidity databases. Table 2 presents the number of clusters determined, for each database, by the various methods. The first raw presents the range of the "true" number of clusters, the most likely values are in parenthesis. These values were set based on the results reported in the literature, for the galaxy database they were set based on results from Escobar and West (1995); Richardson and Green (1997); Fraley and Raftery (2007); Griffin (2010) and others. For the enzyme database results are based on Bechtel et al. (1993); Richardson and Green (1997); Bilancia and Pollice (1999); Griffin (2010), and for the acidity database on Richardson and Green (1997); Griffin (2010); Das and Bhattacharya (2014).

Note that the modified IGMM was executed, again, 500 times per database. For the galaxy database the value $K = 4$ was inferred 98% of the time, for the enzyme database $K = 5$ was inferred 88% of the times and in case of the acidity database $K = 4$ was obtained in *all* our 500 repeated runs of the algorithm.

We can see that all the compared methods, except for the baseline IGMM, were able to infer K quite successfully and their estimation is aligned with the range of values that corresponds to these data bases. For example, RPEM estimated $K = 7$ for the galaxy database which is well within the range of $1 - 15$ deduced in other papers. AIC with $K = 9$ and MDL with $K = 11$ are also well within bounds for this database. Similar results can be observed when looking into the inference results of the other two data bases. One should note, however, that in most cases these results are close to the edge of the acceptable ranges. The modified IGMM results, on the other hand, are closer to the center of these ranges and corresponds better to the more likely values.

	Galaxy	**Enzyme**	**Acidity**
True K	$1-15$ $(5-6)$	$2-10$ $(3-4)$	$1-13$ $(2-4)$
Baseline IGMM	52	54	48
Modified IGMM	4	5	4
GMM_{AIC}	9	10	9
GMM_{MDL}	11	9	10
RPEM	7	9	10

Table 2: Comparing modified IGMM to other methods

Conclusion

The paper attended to an observed order overestimation problem in previous implementations of the IGMM that hinders the use of this unsupervised hierarchical Bayes training procedure for clustering problems. The order overestimation was pinpointed to certain "self-reinforcing feedback" relations between the likelihood function of the concentration parameter of the Dirichlet prior assigned to the weights (α), and the probability of sampling values for K, that sustains the growth of the two of them during the Gibbs iterations. We showed that this failure can not be resolved by a simple replacement of priors but requires using informative priors for α. We proposed an alternative training algorithm that uses $inverse\ \chi^2$ as prior and involves an adjustable parameter (θ) that, upon crude calibration toward the anticipated range of orders, has recovered the "true" number of mixture components in all the experiments held with synthetic data sets. When tested using real-world databases, and compared to other methods used to evaluate model order, the algorithm demonstrated very good performance as well.

While other informative priors could be used, inverse χ^2 was selected since it is the simplest among several possible choices and it has the advantage of having a single tuning parameter with a robust behaviour. The improved order estimation was attained without impairing the accurate estimation of the PDF. Furthermore, the modified IGMM circumvents the need for a secondary intermediate sampling algorithm (ARS) resulting in a much simpler training procedure with a significantly lower cost of computation. The modified IGMM presented here may be used to model data in various classification tasks. It is expected to be particularly attractive in tasks where the investigated data is scarce and has few clusters whose number has to be estimated accurately.

References

Akaike, H. 1973. Information theory as an extension of the maximum likelihood principle–In: Second International Symposium on Information Theory (Eds) BN Petrov, F. *Csaki. BNPBF Csaki Budapest: Academiai Kiado* .

Antoniak, C. E. 1974. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The annals of statistics* 2(6): 1152–1174.

Bechtel, Y. C.; Bonaiti-Pellie, C.; Poisson, N.; Magnette, J.; and Bechtel, P. R. 1993. A population and family study N-acetyltransferase using caffeine urinary metabolites. *Clinical Pharmacology & Therapeutics* 54(2): 134–141.

Bilancia, M.; and Pollice, A. 1999. Bayesian estimation of finite mixtures of Gaussian mixtures. *Metron* 57: 1–2.

Bouguila, N.; and Ziou, D. 2009. A Dirichlet process mixture of generalized Dirichlet distributions for proportional data modeling. *IEEE Transactions on Neural Networks* 21(1): 107–122.

Casella, G.; and George, E. I. 1992. Explaining the Gibbs sampler. *The American Statistician* 46(3): 167–174.

Cheung, Y.-m. 2005. Maximum weighted likelihood via rival penalized EM for density mixture clustering with automatic model selection. *IEEE Transactions on Knowledge and Data Engineering* 17(6): 750–761.

Dai, A. M.; and Storkey, A. J. 2014. The supervised hierarchical Dirichlet process. *IEEE transactions on pattern analysis and machine intelligence* 37(2): 243–255.

Das, M.; and Bhattacharya, S. 2014. Transdimensional transformation based Markov Chain Monte Carlo. *arXiv preprint arXiv:1403.5207* .

Davy, M.; and Tourneret, J.-Y. 2010. Generative supervised classification using dirichlet process priors. *IEEE transactions on pattern analysis and machine intelligence* 32(10): 1781–1794.

Dorazio, R. M. 2009. On selecting a prior for the precision parameter of Dirichlet process mixture models. *Journal of Statistical Planning and Inference* 139(9): 3384–3390.

Escobar, M. D.; and West, M. 1995. Bayesian density estimation and inference using mixtures. *Journal of the american statistical association* 90(430): 577–588.

Ferguson, T. S. 1973. A Bayesian analysis of some nonparametric problems. *The annals of statistics* 1(2): 209–230.

Fraley, C.; and Raftery, A. E. 2007. Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of classification* 24(2): 155–181.

Ghosal, S.; Ghosh, J. K.; and Ramamoorthi, R. 1999. Posterior consistency of Dirichlet mixtures in density estimation. *The annals of statistics* 27(1): 143–158.

Gilks, W. R.; and Wild, P. 1992. Adaptive rejection sampling for Gibbs sampling. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 41(2): 337–348.

Görür, D.; and Rasmussen, C. E. 2010. Dirichlet process gaussian mixture models: Choice of the base distribution. *Journal of Computer Science and Technology* 25(4): 653–664.

Griffin, J. E. 2010. Default priors for density estimation with mixture models. *Bayesian Analysis* 5(1): 45–64.

Harshavardhan, S.; and Sreenivas, T. V. 2010. Robust mixture modeling using t-distribution: Application to speaker ID. In *Eleventh Annual Conference of the International Speech Communication Association*.

Hu, W.; Li, X.; Tian, G.; Maybank, S.; and Zhang, Z. 2013. An incremental DPMM-based method for trajectory clustering, modeling, and retrieval. *IEEE transactions on pattern analysis and machine intelligence* 35(5): 1051–1065.

Kamper, H.; Jansen, A.; King, S.; and Goldwater, S. 2014. Unsupervised lexical clustering of speech segments using fixed-dimensional acoustic embeddings. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, 100–105.

Lu, J. 2017. Hyperprior on symmetric Dirichlet distribution. *arXiv preprint arXiv:1708.08177* .

Matza, A.; and Bistritz, Y. 2011. Speaker recognition with rival penalized EM training. In *2011 IEEE International Workshop on Machine Learning for Signal Processing*, 1–6. IEEE.

Medvedovic, M.; and Sivaganesan, S. 2002. Bayesian infinite mixture model based clustering of gene expression profiles. *Bioinformatics* 18(9): 1194–1206.

Meilă, M.; and Chen, H. 2016. Bayesian non-parametric clustering of ranking data. *IEEE transactions on pattern analysis and machine intelligence* 38(11): 2156–2169.

Mendel, M. 2005. Implementing the Infinite GMM. *Within the project in Tony Jebara's Machine Learning course* (cs4771), downloadable at http://mr-pc.org/work/.

Miller, J. W.; and Harrison, M. T. 2013. A simple example of Dirichlet process mixture inconsistency for the number of components. In *Advances in neural information processing systems*, 199–206.

Miller, J. W.; and Harrison, M. T. 2014. Inconsistency of Pitman-Yor process mixtures for the number of components. *The Journal of Machine Learning Research* 15(1): 3333–3370.

Miller, J. W.; and Harrison, M. T. 2018. Mixture models with a prior on the number of components. *Journal of the American Statistical Association* 113(521): 340–356.

Murugiah, S.; and Sweeting, T. 2012. Selecting the precision parameter prior in Dirichlet process mixture models. *Journal of Statistical Planning and Inference* 142(7): 1947–1959.

Neal, R. M. 2000. Markov chain sampling methods for Dirichlet process mixture models. *Journal of computational and graphical statistics* 9(2): 249–265.

Niekum, S.; and Barto, A. G. 2011. Clustering via dirichlet process mixture models for portable skill discovery. In

Advances in neural information processing systems, 1818–1826.

Rasmussen, C. E. 2000. The infinite Gaussian mixture model. In *Advances in neural information processing systems*, 554–560.

Richardson, S.; and Green, P. J. 1997. On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: series B (statistical methodology)* 59(4): 731–792.

Rissanen, J. 1978. Modeling by shortest data description. *Automatica* 14(5): 465–471.

Shin, M.-S.; Sekora, M.; and Byun, Y.-I. 2009. Detecting variability in massive astronomical time series data–I. Application of an infinite Gaussian mixture model. *Monthly Notices of the Royal Astronomical Society* 400(4): 1897–1910.

West, M. 1992. *Hyperparameter estimation in Dirichlet process mixture models*. Duke University ISDS Discussion Paper # 92-A03.