Generalized Gaussian Mixture Models as a Nonparametric Bayesian Approach for Clustering Using Class-Specific Visual Features

Tarek Elguebaly , Nizar Bouguila [??]

Concordia Institute for Information Systems Engineering, Faculty of Engineering and Computer Science, Concordia University, Montreal, Qc, Canada H3G 2W1

Abstract

Recently, there has been a growing interest in the problem of learning mixture models from data. The reasons and motivations behind this interest are clear, since finite mixture models offer a formal approach to the important problems of clustering and data modeling. In this paper, we address the problem of modeling non-Gaussian data which are largely present, and occur naturally, in several computer vision and image processing applications via the learning of a generative infinite generalized Gaussian mixture model. The proposed model, which can be viewed as a Dirichlet process mixture of generalized Gaussian distributions, takes into account the feature selection problem, also, by determining a set of relevant features for each data cluster which provides better interpretability and generalization capabilities. We propose then an efficient algorithm to learn this infinite model parameters by estimating its posterior distributions using Markov Chain Monte Carlo (MCMC) simulations. We show how the model can be used, while comparing it with other models popular in the literature, in several challenging applications involving photographic and painting images categorization, image and video segmentation, and infrared facial expression recognition.

Key words: Mixture models, generalized Gaussian, feature selection, nonparametric Bayes, MCMC, Gibbs sampling, photographic, painting, segmentation, infrared images.

* Corresponding author
Email addresses: t_elgue@encs.concordia.ca (Tarek Elguebaly), bouguila@ciise.concordia.ca (Nizar Bouguila).

1 Introduction

The problem of clustering data into homogenous groups is widely studied and has many applications in a variety of areas such as image processing, data mining, computer vision and bioinformatics [**?**]. Given its importance many approaches have been proposed in the past. Finite mixture models have become increasingly popular as a formal approach to clustering by assuming that the data are originated from different sources where the data arising from each particular source are modeled by a certain probability density function [**?**]. Such an approach to clustering raises, however, several fundamental problems: Which distribution should be considered to model the data? What order (i.e. number of clusters) should be selected? Should we consider all the features? How we should estimate the mixture parameters? The main goal of this paper is to summarize all these challenging interrelated problems in one unified model.

One of the most fundamental and widely used statistical models is the mixture of Gaussians which is generally justified for asymptotic reasons (i.e. the sample is supposed to be sufficiently large) [**?**]. However, it has been observed that the Gaussian distribution is generally an inappropriate choice to model data in complex real life applications [**?**] and especially in the case of image processing problems where we often deal with small samples [**?**]. For instance, the distribution of intensity levels in natural images is well-known to be far from Gaussian [**?**,**?**,**?**,**?**]. Many studies have shown that the generalized Gaussian distribution (GGD), that we will consider in this paper, can be a good alternative to the Gaussian thanks to its shape flexibility which allows the modeling of a large number of non-Gaussian signals [**?**,**?**,**?**,**?**]. The GGD contains the Laplacian, the Gaussian and asymptotically the uniform distribution as special cases [**?**] and has been used in many challenging problems (see, for instance,[**?**,**?**,**?**]). A standard method to learn finite mixture models is maximum likelihood which generally estimates the parameters through the expectation maximization (EM) framework. The EM algorithm enables us to update the mixture parameters with respect to a data set. The EM, however, is not guaranteed to lead to the best global optimal solution, depends heavily on the choice of initial parameters, and produces models that generally overfits the data which leads to suboptimal generalization performances [**?**,**?**]. A solution to these problems can be provided by Bayesian approaches which consider the average result computed over several models by taking into account model uncertainty [**?**,**?**,**?**] and then enhances generalization performance [**?**,**?**]. Bayesian methods have been extensively used in machine learning and signal processing because they provide a strong theoretical framework to design clustering algorithms as well as a formal approach to incorporate prior knowledge about the problem at hand (see, for instance, [**?**,**?**,**?**,**?**]). A lot of research has been devoted also to the automatic selection of the number of clusters which best describe a given data set (see [**?**,**?**,**?**], for instance, and references therein).

Mixture models are parametric since a particular form has to be chosen for the components densities. At the same time, mixture models can be viewed as nonparametric, since it is possible to increase the number of components as new data arrive. The number of components can be actually supposed to increase to infinity [?]. Thus, mixtures models provide actually the best of both worlds (i.e. parametric and nonparametric approaches). In this paper, we are interested in the nonparametric aspect of mixture models and in particular Bayesian nonparametric approaches for modeling and selection using mixture of Dirichlet processes [?] which have been shown to be a powerful alternative to determine the number of clusters [?,?,?]. In contrast with classic Bayesian approaches which suppose an unknown finite number of mixture components, nonparametric Bayesian approaches assume infinitely complex models (i.e. an infinite number of components) and have witnessed considerable theoretical and computational advances in recent years [?,?,?,?,?,?,?,?,?]. Reviews and in-depth coverage of nonparametric Bayesian approaches can be found in [?,?]. Thus, our approach builds on and extends work on finite generalized Gaussian mixtures [?] to the infinite case. To our knowledge, there has been no previous consideration of nonparametric Bayesian learning for the generalized Gaussian mixture.

The majority of research in mixture models has been primarily concerned with the estimation of parameters and the selection of the number of clusters. Such an approach has several limitations because a priori all features are typically assumed to have the same weight. This is actually an important problem, since the main goal is not only the determination of clusters and their parameters but also providing the most parsimonious model to accurately describe the data which are typically highly dimensional in the number of variables. Moreover, it is noteworthy that the way employed by humans in clustering and recognition is based on formulating few selected features (i.e. humans pick up just the relevant information and ignore the irrelevant [?]) and cluster the data on the basis of these features [?]. Hence, a crucial preprocessing step is usually feature selection which generally provides more comprehensible parsimonious statistical models. Indeed, some studies have shown that two completely different patterns can be made similar by increasing the number of redundant features that encode them [?]. In conventional approaches, feature selection is treated as a separate preprocessing step. It is important to differentiate between feature selection and feature extraction. Unlike feature selection techniques, feature extraction approaches such as principal components analysis, transform the original features into a new dimension-reduced space. The main drawback of feature extraction approaches is that the physical meaning of the original features is generally lost [?]. In our case, and following recent approaches (see, for instance, [?,?,?,?]), feature selection is performed simultaneously with the learning of clusters by incorporating the notion of feature relevancy into our infinite model. It is noteworthy that the proposed work is different from some of our recent efforts, namely [?,?,?,?], that have been dedicated to finite mixture models learning and applications. In fact, [?]

proposes a Bayesian algorithm for the learning of finite one-dimensional generalized Gaussian mixture models. In [**?**] a hybrid generative discriminative Bayesian approach based on the generation of support vector machine kernels from finite Beta-Liouville mixtures for the classification of proportional vectors sequences has been proposed. The work in [**?**] has been devoted to image segmentation using maximum likelihood estimation of Dirichlet-based finite mixture models while the work in [**?**] has concerned reversible jump MCMC learning of finite Beta mixture models for the modeling of one-dimensional data.
The remainder of this article is structured as follows: In Section 2 we introduce the main formalism of our model; in Section 3 we derive the posterior distributions over the model parameters and we provide a detailed description of the learning approach; in Section 4, a simulation study is conducted to demonstrate the effectiveness of the proposed approach via a set of challenging applications; and Section 5 provides a summary of the main results and concludes the paper.

2 A Hierarchical Bayesian Model for Clustering and Feature Selection

In this section, we first introduce our simultaneous feature selection and clustering approach and then we show how it can be represented as a Bayesian Hierarchical model.

2.1 The Modeling Approach

Let $\mathcal{X} = \{\vec{X}_1, \ldots, \vec{X}_N\}$ be an unlabeled data set where each vector $\vec{X}_i$ is composed of a set of continuous features representing a given object (e.g. image, video, document, etc.). It is common to assume that the data contains signals from various sources and generated from a finite mixture model:

$$p(\vec{X}_i|\Theta_M) = \sum_{j=1}^{M} p_j p(\vec{X}_i|\theta_j) \tag{1}$$

where M is the number of components (i.e. sources) which determines the structure of the model, $\Theta_M = (\vec{P}, \vec{\theta})$, $\vec{\theta} = (\theta_1, \ldots, \theta_M)$, $\vec{P} = (p_1, \ldots, p_M)$ is the vector of the components weights which are positive and sum to one, and $p(\vec{X}_i|\theta_j)$ are the components distributions which we take as multidimensional generalized Gaussians. In dimension D, by supposing that the features are conditionally independent, the generalized Gaussian density can be defined

by [**?**]:

$$p(\vec{X}_i|\vec{\mu},\vec{\sigma},\vec{\lambda}) = \prod_{d=1}^{D} p(X_{id}|\mu_d,\sigma_d,\lambda_d) = \prod_{d=1}^{D} \frac{\lambda_d \sqrt{\frac{\Gamma(3/\lambda_d)}{\Gamma(1/\lambda_d)}}}{2\sigma_d \Gamma(1/\lambda_d)} \exp\left(-A(\lambda_d)\left|\frac{X_{id}-\mu_d}{\sigma_d}\right|^{\lambda_d}\right) \tag{2}$$

in which $A(\lambda_d) = \left(\frac{\Gamma(3/\lambda_d)}{\Gamma(1/\lambda_d)}\right)^{\lambda_d/2}$, $\Gamma(.)$ denotes the Gamma function, $\vec{\mu} = (\mu_1,\ldots,\mu_D)$, $\vec{\sigma} = (\sigma_1,\ldots,\sigma_D)$ and $\vec{\lambda} = (\lambda_1,\ldots,\lambda_D)$. μ_d and σ_d are the pdf location and standard deviation parameters in the d^{th} dimension. The generalized Gaussian has been shown to efficiently take into account the non-Gaussian character of the natural image ensemble [**?**] thanks to the flexibility of its shape. The parameter λ_d controls the tails of the pdf and determines whether it is peaked or flat. Smaller values of λ_d correspond to heavy tailed distributions, when $\lambda = 2$ we have the Gaussian distribution, when $\lambda = 1$, we have the Laplacian pdf, when $\lambda >> 1$ the distribution tends to a uniform pdf, and when $\lambda < 1$ the pdf tends to be more peaked around the mean and to have heavier tails [**?**]. Notice that by selecting generalized Gaussians for the mixture components, the generic parameter θ_j in Eq. **??** becomes $(\vec{\mu}_j,\vec{\sigma}_j,\vec{\lambda}_j)$. It is noteworthy that the model in Eq. **??** supposes actually that the D features have the same importance and carry pertinent information which is not generally the case, since many of which can be irrelevant for the targeted application. This is especially true in the case of image processing and computer vision applications which generally generate high-dimensional feature vectors and thus grew out the need to have efficient feature weighting and selection procedures [**?**,**?**,**?**,**?**,**?**]. Examples include the challenging problems of object detection and visual scenes categorization where an important step is to determine which are the relevant features that express structure common to a given object or visual scene class [**?**,**?**]. In fact, using all the dimensions in general will not only result in poor modeling, but also incurs excessive costs for estimating an excessive number of model parameters, some of which are potentially irrelevant [**?**]. It is natural, then, to assume that different features may have different weights according to each data cluster [**?**,**?**] which can be expressed as following in the case of mixture models [**?**]

$$p(\vec{X}_i|\Theta) = \sum_{j=1}^{M} p_j \prod_{d=1}^{D} \left(\rho_{jd} p(X_{id}|\theta_{jd}) + (1-\rho_{jd}) p(X_{id}|\theta_{jd}^{irr})\right) \tag{3}$$

where $\Theta = (\Theta_M,\vec{\rho},\vec{\theta}^{irr})$, $\vec{\theta}^{irr} = (\theta_1^{irr},\ldots,\theta_M^{irr})$, $\theta_{jd} = (\mu_{jd},\sigma_{jd},\lambda_{jd})$, $\theta_{jd}^{irr} = (\mu_{jd}^{irr},\sigma_{jd}^{irr},\lambda_{jd}^{irr})$, and $\vec{\rho} = (\vec{\rho}_1,\ldots,\vec{\rho}_M)$ such that $\vec{\rho}_j = (\rho_{j1},\ldots,\rho_{jD})$ where each $0 \leq \rho_{jd} \leq 1$ represents the saliency of feature d for component j (i.e. the probability that feature d is relevant for component j). The previous model has actually a sound interpretation. Indeed, it considers that the features are not with equal importance and makes a distinction between those that are relevant and those which are irrelevant. Conceptually we assume that relevant

features have been generated from $p(X_{id}|\theta_{jd})$ and irrelevant features have been generated from another distribution $p(X_{id}|\theta_{jd}^{irr})$ taken also as a generalized Gaussian. We finally note that the previous model is reduced to the one in Eq. **??** when all the feature are considered as relevant.

2.2 Bayesian Hierarchical Model

In the context of Bayesian inference, the most important step is the determination of the posterior which is actually proportional to the model joint distribution [**?**,**?**] which is given by the following in our case

$$\begin{aligned} p(\vec{P}, Z, \vec{\rho}, z, \vec{\theta}, \vec{\theta}^{irr}, \mathcal{X}) = & \, p(\vec{P})p(Z|\vec{P})p(\vec{\rho}|\vec{P}, Z)p(z|\vec{\rho}, \vec{P}, Z)p(\vec{\theta}|\vec{P}, Z, \vec{\rho}, z) \\ & \times p(\vec{\theta}^{irr}|\vec{P}, Z, \vec{\rho}, z, \vec{\theta})p(\mathcal{X}|\vec{P}, Z, \vec{\rho}, z, \vec{\theta}, \vec{\theta}^{irr}) \end{aligned} \tag{4}$$

where $Z = (Z_1, \ldots, Z_N)$ represents the missing allocation variables and $z = (z_1, \ldots, z_N)$ are missing binary vectors to identify if a given feature is relevant or not. Each Z_i indicates from which cluster each vector $\vec{X}_i$ arose (i.e. $Z_i = j$ means that $\vec{X}_i$ comes from component j). Each $p_j = p(Z_i = j)$ represents the *a priori* probability that the vector $\vec{X}_i$ was generated by component j, and it follows from Bayes' theorem [**?**,**?**] that $p(Z_i = j|\vec{X}_i)$, the probability that vector i is in cluster j, conditional on having observed $\vec{X}_i$ is given by

$$\begin{aligned} p(Z_i = j|\vec{X}_i) &= \frac{p_j \prod_{d=1}^{D} \left(\rho_{jd} p(X_{id}|\theta_{jd}) + (1-\rho_{jd}) p(X_{id}|\theta_{jd}^{irr})\right)}{\sum_{j=1}^{M} p_j \prod_{d=1}^{D} \left(\rho_{jd} p(X_{id}|\theta_{jd}) + (1-\rho_{jd}) p(X_{id}|\theta_{jd}^{irr})\right)} \\ &\propto p_j \prod_{d=1}^{D} \left(\rho_{jd} p(X_{id}|\theta_{jd}) + (1-\rho_{jd}) p(X_{id}|\theta_{jd}^{irr})\right) \end{aligned} \tag{5}$$

As for z, we have $z_i = (\vec{z}_{i1}, \ldots, \vec{z}_{iM})$, $\vec{z}_{ij} = (z_{ij1}, \ldots, z_{ijD})$ where each z_{ijd} indicates if feature d in vector $\vec{X}_i$ is relevant for cluster j or not (i.e. $z_{ijd} = 1$, if the feature d is relevant for cluster j and $z_{ijd} = 0$, otherwise). Each $\rho_{jd} = p(z_{ijd} = 1)$ represents the *a priori* probability that the feature d is relevant for component j, and it is straightforward to show that $p(z_{ijd} = 1, Z_i = j|\vec{X}_i)$, the probability that a feature d is relevant for cluster j, conditional on having observed $\vec{X}_i$ is given by

$$\begin{aligned} p(z_{ijd} = 1, Z_i = j|\vec{X}_i) &= \frac{\rho_{jd} p(X_{id}|\theta_{jd})}{\rho_{jd} p(X_{id}|\theta_{jd}) + (1-\rho_{jd}) p(X_{id}|\theta_{jd}^{irr})} p(Z_i = j|\vec{X}_i) \\ &\propto \rho_{jd} p(X_{id}|\theta_{jd}) p(Z_i = j|\vec{X}_i) \end{aligned} \tag{6}$$

and we can deduce that

$$p(z_{ijd}=0, Z_i=j|\vec{X}_i) = \frac{(1-\rho_{jd})p(X_{id}|\theta^{irr}_{jd})}{\rho_{jd}p(X_{id}|\theta_{jd}) + (1-\rho_{jd})p(X_{id}|\theta^{irr}_{jd})} p(Z_i=j|\vec{X}_i)$$
$$\propto (1-\rho_{jd})p(X_{id}|\theta^{irr}_{jd})p(Z_i=j|\vec{X}_i) \quad (7)$$

It is worth mentioning that if we condition on Z and z, the distribution of $\mathcal{X}$ is simply given by

$$p(\mathcal{X}|\vec{P}, Z, \vec{\rho}, z, \vec{\theta}, \vec{\theta}^{irr}) = p(\mathcal{X}|\vec{\theta}, \vec{\theta}^{irr}, Z, z)$$
$$= \prod_{i=1}^{N}\prod_{d=1}^{D}\left[\left(p(X_{id}|\theta_{Z_id})\right)^{z_{id}}\left(p(X_{id}|\theta^{irr}_{Z_id})\right)^{1-z_{id}}\right] \quad (8)$$

We impose further common conditional independencies, so that: $p(\vec{\rho}|\vec{P}, Z) = p(\vec{\rho})$, $p(z|\vec{\rho}, \vec{P}, Z) = p(z|\vec{\rho})$, $p(\vec{\theta}|\vec{P}, Z, \vec{\rho}, z) = p(\vec{\theta})$, $p(\vec{\theta}^{irr}|\vec{P}, Z, \vec{\rho}, z, \vec{\theta}) = p(\vec{\theta}^{irr})$, $p(\vec{\theta}|Z, \vec{P}) = p(\vec{\theta})$, thus

$$p(\vec{P}, Z, \vec{\rho}, z, \vec{\theta}, \vec{\theta}^{irr}, \mathcal{X}) = p(\vec{P})p(Z|\vec{P})p(\vec{\rho})p(z|\vec{\rho})p(\vec{\theta})p(\vec{\theta}^{irr})p(\mathcal{X}|\vec{\theta}, \vec{\theta}^{irr}, Z, z) \quad (9)$$

A serious practical problem now is to choose the prior distributions which describe our prior opinion about the model parameters. In our case, we suppose that $\vec{\theta}$, $\vec{\theta}^{irr}$, $\vec{\rho}$ and $\vec{P}$ follow priors depending on hyperparameters, drawn from independent hyperpriors, Λ, Λ^{irr}, δ and η, respectively. In addition, our prior distributions are that $\vec{\mu}_j$, $\vec{\sigma}_j$, $\vec{\lambda}_j$, $\vec{\mu}^{irr}_j$, $\vec{\sigma}^{irr}_j$ and $\vec{\lambda}^{irr}_j$ are all drawn independently:

$$p(\vec{\theta}|\Lambda) = \prod_{j=1}^{M} p(\vec{\sigma}_j|\Lambda_{j|\sigma})p(\vec{\mu}_j|\Lambda_{j|\mu})p(\vec{\lambda}_j|\Lambda_{j|\lambda}) \quad (10)$$

$$p(\vec{\theta}^{irr}|\Lambda^{irr}) = \prod_{j=1}^{M} p(\vec{\sigma}^{irr}_j|\Lambda^{irr}_{j|\sigma})p(\vec{\mu}^{irr}_j|\Lambda^{irr}_{j|\mu})p(\vec{\lambda}^{irr}_j|\Lambda^{irr}_{j|\lambda}) \quad (11)$$

where $\Lambda = (\Lambda_1, \ldots, \Lambda_M)$, $\Lambda_j = (\Lambda_{j|\sigma}, \Lambda_{j|\mu}, \Lambda_{j|\lambda})$, $\Lambda^{irr} = (\Lambda^{irr}_1, \ldots, \Lambda^{irr}_M)$ and $\Lambda^{irr}_j = (\Lambda^{irr}_{j|\sigma}, \Lambda^{irr}_{j|\mu}, \Lambda^{irr}_{j|\lambda})$. To add more flexibility to the model, it is common to assume that the hyperparameters η, δ, Λ^{irr} and Λ themselves follow distributions $p(\eta)$, $p(\delta)$, $p(\Lambda^{irr})$ and $p(\Lambda)$, respectively, that we shall develop in the next section. The joint distribution of all our model's variables is then expressed by the following factorization

$$p(\vec{P}, Z, \vec{\rho}, z, \vec{\theta}, \vec{\theta}^{irr}, \eta, \delta, \Lambda, \Lambda^{irr}, \mathcal{X}) = p(\eta)p(\delta)p(\Lambda)p(\Lambda^{irr}) \quad (12)$$
$$\times p(\vec{P}|\eta)p(Z|\vec{P})p(\vec{\rho}|\delta)p(z|\vec{\rho})p(\mathcal{X}|\vec{\theta}, \vec{\theta}^{irr}, Z, z)\prod_{j=1}^{M}\Bigg[p(\vec{\sigma}_j|\Lambda_{j|\sigma})p(\vec{\mu}_j|\Lambda_{j|\mu})$$
$$\times p(\vec{\lambda}_j|\Lambda_{j|\lambda})p(\vec{\sigma}^{irr}_j|\Lambda^{irr}_{j|\sigma})p(\vec{\mu}^{irr}_j|\Lambda^{irr}_{j|\mu})p(\vec{\lambda}^{irr}_j|\Lambda^{irr}_{j|\lambda})\Bigg]$$

3 Nonparametric Bayesian Learning

In this section, we first present the priors of our model. After specifying prior distributions, it is important to consider how to update these priors with information brought by the data to obtain the posterior distributions. After developing these posteriors, we describe the nonparametric approach by extending the model to the infinite case. The MCMC posterior inference and the complete learning algorithm are also given.

3.1 Priors and Posteriors

3.1.1 Conditional Posterior Distributions of $\vec{\mu}_j$ and $\vec{\mu}_j^{irr}$

We consider independent Normal priors with common hyperparameters ψ and ε^2 as the mean and variance, respectively, for the different μ_{jd} and μ_{jd}^{irr}:

$$p(\vec{\mu}_j|\psi,\varepsilon^2) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi}\varepsilon} \exp\Big(\frac{-(\mu_{jd}-\psi)^2}{2\varepsilon^2}\Big) \tag{13}$$

And $p(\vec{\mu}_j^{irr}|\psi,\varepsilon^2)$ has the same form as $p(\vec{\mu}_j|\psi,\varepsilon^2)$. So, the generic hyperparameters $\Lambda_{j|\mu}$ and $\Lambda_{j|\mu}^{irr}$ become (ψ,ε) and according to the previous equation and our joint distribution in Eq. **??**, the full conditional posterior distributions for $\vec{\mu}_j$ and $\vec{\mu}_j^{irr}$, giving the rest of the parameters, are:

$$p(\vec{\mu}_j|\ldots) \propto p(\vec{\mu}_j|\psi,\varepsilon^2)p(\mathcal{X}|\vec{\theta},\vec{\theta}^{irr},Z,z) \qquad p(\vec{\mu}_j^{irr}|\ldots) \propto p(\vec{\mu}_j^{irr}|\psi,\varepsilon^2)p(\mathcal{X}|\vec{\theta},\vec{\theta}^{irr},Z,z) \tag{14}$$

The hyperparameters ψ and ε are given Normal and inverse Gamma priors, respectively:

$$p(\psi|\epsilon,\chi^2) = \frac{1}{\sqrt{2\pi}\chi} \exp\Big(\frac{-(\psi-\epsilon)^2}{2\chi^2}\Big) \qquad p(\varepsilon^2|\varphi,\varrho) = \frac{\varrho^{\varphi}\exp(-\varrho/\varepsilon^2)}{\Gamma(\varphi)\varepsilon^{2(\varphi+1)}} \tag{15}$$

Thus, according to Eqs. **??**, **??** and **??**, we obtain the following posteriors

$$p(\psi|\ldots) \propto p(\psi|\epsilon,\chi^2)\prod_{j=1}^{M} p(\vec{\mu}_j|\psi,\varepsilon^2)p(\vec{\mu}_j^{irr}|\psi,\varepsilon^2) \qquad p(\varepsilon^2|\ldots) \propto p(\varepsilon^2|\varphi,\varrho)\prod_{j=1}^{M} p(\vec{\mu}_j|\psi,\varepsilon^2)p(\vec{\mu}_j^{irr}|\psi,\varepsilon^2) \tag{16}$$

3.1.2 Conditional Posterior Distributions of $\vec{\sigma}_j$ and $\vec{\sigma}_j^{irr}$

Independent Gamma priors with common hyperpriors ι, υ, as the shape and rate parameters, are given for $\vec{\sigma}_j$ and $\vec{\sigma}_j^{irr}$

$$p(\vec{\sigma}_j|\iota,\upsilon) \sim \prod_{d=1}^{D} \frac{\sigma_{jd}^{\iota-1}\upsilon^{\iota}\exp\Big(-\upsilon\sigma_{jd}\Big)}{\Gamma(\iota)} \tag{17}$$

And $p(\vec{\sigma}_j^{irr}|\iota,\upsilon)$ has the same form as $p(\vec{\sigma}_j|\iota,\upsilon)$. So, the generic hyperparameters $\Lambda_{j|\sigma}$ and $\Lambda_{j|\sigma}^{irr}$ become (ι,υ) and according to the previous equation and our joint distribution in Eq. **??**, the full conditional posterior distributions for $\vec{\sigma}_j$ and $\vec{\sigma}_j^{irr}$, giving the rest of the parameters, are:

$$p(\vec{\sigma}_j|\ldots) \propto p(\vec{\sigma}_j|\iota,\upsilon)p(\mathcal{X}|\vec{\theta},\vec{\theta}^{irr},Z,z) \qquad p(\vec{\sigma}_j^{irr}|\ldots) \propto p(\vec{\sigma}_j^{irr}|\iota,\upsilon)p(\mathcal{X}|\vec{\theta},\vec{\theta}^{irr},Z,z) \tag{18}$$

The hyperparameters ι and υ are given inverse Gamma and Gamma priors, respectively:

$$p(\iota|\vartheta,\varpi) \sim \frac{\varpi^{\vartheta}\exp(-\varpi/\iota)}{\Gamma(\vartheta)\iota^{\vartheta+1}} \qquad p(\upsilon|\tau,\omega) \sim \frac{\upsilon^{\tau-1}\omega^{\tau}\exp\Big(-\omega\upsilon\Big)}{\Gamma(\tau)} \tag{19}$$

Thus, according to Eqs. **??**, **??** and **??**, we obtain the following posteriors

$$p(\iota|\ldots) \propto p(\iota|\vartheta,\varpi)\prod_{j=1}^{M} p(\vec{\sigma}_j|\iota,\upsilon)p(\vec{\sigma}_j^{irr}|\iota,\upsilon) \qquad p(\upsilon|\ldots) \propto p(\upsilon|\tau,\omega)\prod_{j=1}^{M} p(\vec{\sigma}_j|\iota,\upsilon)p(\vec{\sigma}_j^{irr}|\iota,\upsilon) \tag{20}$$

3.1.3 Conditional Posterior Distributions of $\vec{\lambda}_j$ and $\vec{\lambda}_j^{irr}$

For the parameters $\vec{\lambda}_j$ and $\vec{\lambda}_j^{irr}$ we placed independent Gamma priors with common hyperparameters κ and ς:

$$p(\vec{\lambda}_j|\kappa,\varsigma) = \prod_{d=1}^{D} \frac{\lambda_{jd}^{\kappa-1}\varsigma^{\kappa}\exp\Big(-\varsigma\lambda_{jd}\Big)}{\Gamma(\kappa)} \tag{21}$$

And $p(\vec{\lambda}_j^{irr}|\kappa,\varsigma)$ has the same form as $p(\vec{\lambda}_j|\kappa,\varsigma)$. So, the generic hyperparameters $\Lambda_{j|\lambda}$ and $\Lambda_{j|\lambda}^{irr}$ become (κ,ς) and according to the previous equation and our joint distribution in Eq. **??**, the full conditional posterior distributions for $\vec{\lambda}_j$ and $\vec{\lambda}_j^{irr}$, giving the rest of the parameters, are:

$$p(\vec{\lambda}_j|\ldots) \propto p(\vec{\lambda}_j|\kappa,\varsigma)p(\mathcal{X}|\vec{\theta},\vec{\theta}^{irr},Z,z) \qquad p(\vec{\lambda}_j^{irr}|\ldots) \propto p(\vec{\lambda}_j^{irr}|\kappa,\varsigma)p(\mathcal{X}|\vec{\theta},\vec{\theta}^{irr},Z,z) \tag{22}$$

The hyperparameters κ and ς are given inverse Gamma and Gamma priors, respectively:

$$p(\kappa|\alpha,\phi) \sim \frac{\phi^{\alpha}\exp(-\phi/\kappa)}{\Gamma(\alpha)\kappa^{\alpha+1}} \qquad p(\varsigma|\nu,\beta) \sim \frac{\varsigma^{\nu-1}\beta^{\nu}\exp\left(-\beta\varsigma\right)}{\Gamma(\nu)} \tag{23}$$

Thus, according to Eqs. **??**, **??** and **??**, we obtain the following posteriors

$$p(\kappa|\ldots) \propto p(\kappa|\alpha,\phi)\prod_{j=1}^{M} p(\vec{\lambda}_j|\kappa,\varsigma)p(\vec{\lambda}_j^{irr}|\kappa,\varsigma) \qquad p(\varsigma|\ldots) \propto p(\varsigma|\nu,\beta)\prod_{j=1}^{M} p(\vec{\lambda}_j|\kappa,\varsigma)p(\vec{\lambda}_j^{irr}|\kappa,\varsigma) \tag{24}$$

3.1.4 Conditional Posterior Distribution of $\vec{\rho}$

We know that each ρ_{jd} is defined in the compact support [0,1], thus we consider for it a Beta distribution, with parameters δ_1 and δ_2 common to all classes and all dimensions, as a prior, which give us

$$p(\vec{\rho}|\delta) = \left[\frac{\Gamma(\delta_1+\delta_2)}{\Gamma(\delta_1)\Gamma(\delta_2)}\right]^{MD}\prod_{j=1}^{M}\prod_{d=1}^{D}\rho_{jd}^{\delta_1-1}(1-\rho_{jd})^{\delta_2-1} \tag{25}$$

So, the generic hyperparameter δ become (δ_1,δ_2). Recall that $\rho_{jd} = p(z_{jd}=1)$ and $1-\rho_{jd} = p(z_{jd}=0)$, $d=1,\ldots,D$, $j=1,\ldots,M$ thus each z_{jd} follows a D-variate Bernoulli distribution and we have

$$p(z|\vec{\rho}) = \prod_{i=1}^{N}\prod_{j=1}^{M}\prod_{d=1}^{D}\rho_{jd}^{z_{ijd}}(1-\rho_{jd})^{1-z_{ijd}} = \prod_{j=1}^{M}\prod_{d=1}^{D}\rho_{jd}^{f_{jd}}(1-\rho_{jd})^{N-f_{jd}} \tag{26}$$

where $f_{jd} = \sum_{i=1}^{N}\mathbb{I}_{z_{ijd}=1}$. Then, according to Eqs. **??**, **??** and **??**, we have

$$p(\vec{\rho}|\ldots) \propto p(\vec{\rho}|\delta)p(z|\vec{\rho}) \propto \prod_{j=1}^{M}\prod_{d=1}^{D}\rho_{jd}^{f_{jd}+\delta_1-1}(1-\rho_{jd})^{N-f_{jd}+\delta_2-1} \tag{27}$$

The hyperparameters δ_1 and δ_2 are given Gamma priors with common hyperparameters $(\varphi_\delta,\varrho_\delta)$ which give us the following posteriors

$$p(\delta_1|\ldots) \propto p(\delta_1|\varphi_\delta,\varrho_\delta)p(\vec{\rho}|\delta) \qquad p(\delta_2|\ldots) \propto p(\delta_2|\varphi_\delta,\varrho_\delta)p(\vec{\rho}|\delta) \tag{28}$$

3.2 The Infinite Model

An important issue now is the determination of the number of clusters which has been widely studied from both Bayesian and deterministic perspectives by supposing that the number of components is bounded (see, for instance, [**?**,**?**]).

An alternative approach that has attracted a lot of attention recently is to define mixture of distributions with a countably infinite number of components [**?**]. The attractive features of nonparametric Bayesian approaches, which can incorporate infinitely many parameters, have been widely exploited and are well documented and will not be repeated here (see, for instance, [**?**,**?**,**?**,**?**]). In the following, we explain the main idea behind this approach in the case of mixture models.
According to Eq. **??**, the only terms that involve $\vec{P}$ whose dimensionality is M are $p(Z|\vec{P})$ and $p(\vec{P}|\eta)$. Recall that $p_j = p(Z_i = j), j = 1, \ldots, M$, thus

$$p(Z|\vec{P}) = \prod_{j=1}^{M} p_j^{n_j} \tag{29}$$

where $n_j = \sum_{i=1}^{N} \mathbb{I}_{Z_i=j}$ is the number of vector in cluster j. The distribution $p(\vec{P}|\eta)$ is taken as a symmetric Dirichlet with parameters $\frac{\eta}{M}$. Because the Dirichlet is a conjugate prior to the multinomial, we can marginalize out $\vec{P}$ from Eq. **??**:

$$\begin{aligned} p(Z|\eta) &= \int_{\vec{P}} p(Z|\vec{P}) p(\vec{P}|\eta) d\vec{P} = \frac{\Gamma(\eta)}{\prod_{j=1}^{M} \Gamma(\frac{\eta}{M})} \int_{\vec{P}} \prod_{j=1}^{M} p_j^{n_j + \frac{\eta}{M} - 1} d\vec{P} \\ &= \frac{\Gamma(\eta)}{\Gamma(\eta + N)} \prod_{j=1}^{M} \frac{\Gamma(\frac{\eta}{M} + n_j)}{\Gamma(\frac{\eta}{M})} \end{aligned} \tag{30}$$

which can be considered as a prior on Z. We have also

$$p(\vec{P}|Z, \eta) = \frac{p(Z|\vec{P}) p(\vec{P}|\eta)}{p(Z|\eta)} = \frac{\Gamma(\eta + N)}{\prod_{j=1}^{M} \Gamma(\frac{\eta}{M} + n_j)} \prod_{j=1}^{M} p_j^{n_j + \frac{\eta}{M} - 1} \tag{31}$$

which is a Dirichlet distribution with parameters $(n_1 + \frac{\eta}{M}, \ldots, n_M + \frac{\eta}{M})$ from which we can show that:

$$p(Z_i = j|\eta, Z_{-i}) = \frac{n_{-i,j} + \frac{\eta}{M}}{N - 1 + \eta} \tag{32}$$

where $Z_{-i} = \{Z_1, \ldots, Z_{i-1}, Z_{i+1}, \ldots, Z_N\}$, $n_{-i,j}$ is the number of vectors, excluding $\vec{X}_i$, in cluster j. The main idea behind countably infinite mixture models relies on observing that by taking the limit of $p(Z_i = j|\eta, Z_{-i})$ as $M \to \infty$ gives us [**?**,**?**]

$$p(Z_i = j|\eta, Z_{-i}) = \begin{cases} \frac{n_{-i,j}}{N-1+\eta} & \text{if } n_{-i,j} > 0 \text{ (cluster } j \text{ is represented)} \\ \frac{\eta}{N-1+\eta} & \text{if } n_{-i,j} = 0 \text{ (cluster } j \text{ is not represented)} \end{cases} \tag{33}$$

Thus, a vector $\vec{X}_i$ is allocated to an existing (i.e. represented) cluster with a certain probability proportional to the number of vectors already assigned

to this cluster and it is affected to a new (i.e. not represented) cluster with probability proportional to the hyperparameter η. It is noteworthy that infinite mixture models takes implicitly into account the notion of online learning and then the fact that features relevancy may change as new data arrive which is crucial in several machine vision applications for instance [**?**]. Having the conditional priors in Eq. **??**, the conditional posteriors are obtained by combining these priors with the likelihood of the data [**?**,**?**]

$$p(Z_i = j|\ldots) = \begin{cases} \frac{n_{-i,j}}{N-1+\eta}\prod_{d=1}^{D}\left(\rho_{jd}p(X_{id}|\theta_{jd}) + (1-\rho_{jd})p(X_{id}|\theta_{jd}^{irr})\right) & \text{if } j \text{ is represented} \\ \int \frac{\eta p(\vec{Y}_i|\vec{m}_j,\vec{s}_j,\xi^{irr},z_i)p(\theta_j|\Lambda_j)p(\theta_j^{irr}|\Lambda_j^{irr})p(\vec{\rho}_j|\delta)}{N-1+\eta} d\theta_j d\theta_j^{irr} d\vec{\rho}_j & \text{if } j \text{ is not represented} \end{cases} \quad (34)$$

Concerning the hyperparameters η [1], we have chosen an inverse gamma prior with parameters (χ_η, κ_η) for it:

$$p(\eta|\chi_\eta,\kappa_\eta) \sim \frac{{\kappa_\eta}^{\chi_\eta}\exp(-\kappa_\eta/\eta)}{\Gamma(\chi_\eta)\eta^{\chi_\eta+1}} \quad (35)$$

which gives with Eq. **??** the following posterior (for more details, see [**?**])

$$p(\eta|\ldots) \propto \frac{{\kappa_\eta}^{\chi_\eta}\exp(-\kappa_\eta/\eta)}{\Gamma(\chi_\eta)\eta^{\chi_\eta+1}} \frac{\eta^M\Gamma(\eta)}{\Gamma(N+\eta)} \quad (36)$$

3.3 Complete Algorithm

Several Monte carlo methods for sampling mixture posteriors have been developed in the past [**?**,**?**]. The most widely applied approach is the Gibbs sampling (see [**?**], for instance, for interesting discussions) that we will use to sample from the obtained model posteriors as follows:

- Generate Z_i from Eq. **??** and then update n_j, $j = 1,\ldots,M$, $i = 1,\ldots,N$.
- Update the number of represented components M.
- $p_j = \frac{n_j}{N+\eta}$, $j = 1,\ldots,M$ and the mixing parameters of unrepresented components are given by $p_U = \frac{\eta}{\eta+N}$.
- Generate the $\vec{z}_{ij}$ from a D-variate Bernoulli distribution with parameters $p(z_{ijd} = 1, Z_i = j|\vec{X}_i)$.
- Generate ρ_d from Eq. **??**, $d = 1,\ldots,D$.

[1] This parameter plays an important role in controlling the weights of the mixture components and then the number of clusters. Indeed, it is possible to show that the number of clusters increase at a rate proportional to $\eta\log N$ and then it is crucial to suppose that it is unknown and then follows a prior distribution [**?**].

- Generate $\vec{\mu}_j$ and $\vec{\mu}_j^{irr}$ from the posteriors in Eq. **??**, $\vec{\sigma}_j$ and $\vec{\sigma}_j^{irr}$ from the posteriors in Eq. **??**, $\vec{\lambda}_j$ and $\vec{\lambda}_j^{irr}$ from the posteriors in Eq. **??**, $j = 1, \ldots, M$.
- Update the hyperparameters: Generate ψ, ε^2, ι, υ, κ, ς, δ_1, δ_2 and η according to the posteriors in Eqs. **??**, **??**, **??**, **??** and **??**, respectively.

Note that, for the initialization step we start by assuming that all the vectors are in the same cluster and that all the features are relevant, and we generate the parameters by sampling from their prior distributions. It is noteworthy also that although an infinite model appears complex because of the number of involved parameters, it allows actually straightforward posterior inference with MCMC simulation as it is clear from the previous algorithm. The above algorithm can be viewed actually as a self-refinement process that starts with an initial set of data and feature relevancy. From this initial state, the process strives to find features that are discriminative for each cluster, and then refine the clusters by determining the cluster label of each vector using these relevant features. Via this self-refinement process, the accuracy of the whole data representation is gradually improved. All the conditional posterior distributions are straightforward to sample from (especially the posteriors of ρ_d and $\vec{z}_{ij}$ which have known forms). Indeed, sampling from Eqs. **??**, **??**, **??**, **??** and **??** is based on adaptive rejection sampling (ARS) [**?**]. The sampling of the vectors Z_i (Eq. **??**) is based on an approach, originally proposed in [**?**]. For simulations from the posteriors of $\vec{\mu}_j$, $\vec{\mu}_j^{irr}$, $\vec{\sigma}_j$, $\vec{\sigma}_j^{irr}$, $\vec{\lambda}_j$ and $\vec{\lambda}_j^{irr}$, we apply the well known random walk Metropolis-Hastings (M-H) algorithm (i.e. log-normal proposals with scale ζ^2). An important problem when using MCMC techniques is the convergence assessment which has been the topic of extensive rigorous studies in the past (see, for instance, [**?**,**?**,**?**]). Several systematic approaches for establishing convergence of MCMC have been proposed and one of them that we follow is the diagnostic approach proposed by Raftery and Lewis [**?**,**?**], that has been shown to often work well in practice. This approach is based on a single long-run of the Gibbs sampler.

4 Experimental Results

Performance evaluation for our model is conducted using a set of challenging experiments involving distinguishing paintings from photographs, image and video segmentation, and infrared facial expression recognition. The main goal of these experiments is to compare our infinite model (IGGM+FS) with other models that have been used in the literature namely finite Gaussian mixture (GM) [**?**], finite Gaussian mixture with feature selection (GM+FS) [**?**], finite generalized Gaussian (GGM) [**?**], finite generalized Gaussian with feature selection (GGM+Fs) [**?**], infinite Gaussian mixture (IGM) [**?**], infinite Gaussian mixture with feature selection (IGM+FS) and infinite generalized Gaussian

(IGM) [**?**]. It is noteworthy that the IGM+FS approach is deduced from our approach by setting the λ_{jd} values to 2 which reduces the generalized Gaussian to the Gaussian. In these applications our specific choice for the hyperparameters is $\varphi_\delta = 2, \varrho_\delta = 0.5, \chi_\eta = 2, \kappa_\eta = 1, \epsilon = 0, \chi^2 = 1, \varphi = 2, \varrho = 0.5, \vartheta = 2, \varpi = 1, \tau = 2, \omega = 0.5, \alpha = 2, \phi = 1, \nu = 0.5$ and $\beta = 2$. In order to conduct a sensitivity test we have used different values of hyperparameters in the following intervals :$\varphi_\delta \in [1.8, 2.2], \varrho_\delta \in [0.3, 0.7], \chi_\eta \in [1.8, 2.2], \kappa_\eta \in [0.8, 1.2], \epsilon \in [0, 0.4], \chi^2 \in [0.8, 1.2], \varphi \in [1.8, 2.2], \varrho \in [0.3, 0.7], \vartheta \in [1.8, 2.2], \varpi \in [0.8, 1.2], \tau \in [1.8, 2.2], \omega \in [0.3, 0.7], \alpha \in [1.8, 2.2], \phi \in [0.8, 1.2], \nu \in [0.3, 0.7]$ and $\beta \in [1.8, 2.2]$. Having the outputs of our algorithm when changing the hyperparameters in hand we applied a student t test to determine the robustness of the posteriors results with respect to our hyperparameters.

4.1 Distinguishing Paintings from Photographs

4.1.1 Image Description

Distinguishing paintings from real photographs is an important and challenging (even for a human observer) problem in several applications such as content-based image retrieval, web site filtering (e.g. distinguishing pornographic images from nude paintings) [**?**,**?**], categorization [**?**] and content-based access to art paintings [**?**]. However, very few works have been proposed in the past [**?**,**?**,**?**,**?**] as compared, for instance, to the problem of distinguishing photographs and computer-generated graphics [**?**,**?**,**?**]. In particular, the authors in [**?**,**?**] found that an important step is the extraction of the right visual features derived from the edge, color and gray-scale-texture information. In particular, the following distinguishing features have been derived and found efficient. Four scalar-valued, called visual features, were defined namely color edges vs. intensity edges (E_g), spatial variation of color (R), number of unique colors (U) and pixel saturation (S). Pixel distribution in RGBXY space ($\vec{s}$) and gray-scale-texture have been considered, also.
Color edges vs. intensity edges feature is defined as $E_g = \frac{\text{\#pixels: intensity, not color edge}}{\text{total number of edge pixels}}$ [**?**,**?**]. The spatial variation of color, R, is defined as the average over all image pixels of the sums of the areas of the facets of the pyramids determined by three normals at each pixel. These normals are obtained by determining at each pixel the orientation of the plane that best fits a 5×5 neighborhood centered on that pixel in the RGB domain. The number of unique colors, U, is defined as the number of unique colors of an image normalized by the total number of pixels. The pixel saturation, S, is defined as the ratio between the count in the highest bin (20) and the lowest (1) of the mean saturation histogram derived from the image represented in the HSV color space. The pixel distribution in RGBXY space represents an image by a five dimensional vectors $\vec{s}$ of the singular values of its RGBXY pixel covariance matrix (i.e.

the image representation in the RGB color space enhanced by adding the two spatial coordinates, x and y to the RGB vector of each pixel). Finally, the gray-scale-texture feature is a description of the image by a feature vector of 32 dimensions which represent the mean and standard deviation of the Gabor responses across image locations for filtered images obtained by considering four different scales and four orientations (0, 90, 45, 135 degrees). Using all these features images can be represented by 41-dimensional vectors which can be used for the categorization task (i.e. paintings vs. photographs).

4.1.2 Results

The performance of our infinite mixture model was evaluated on the data set considered in [**?**,**?**] which contains 6000 photographs, with 568 $\times$ 506 pixels mean size and standard deviation equal to 144 $\times$ 92 pixels, and 6000 paintings with mean size and standard deviation equal to 534 $\times$ 497 and 171 $\times$ 143 pixels, respectively. In this data set, the painting class includes conventional canvas paintings, murals and frescoes, but excludes line drawings and computed-generated images. On the other hand, the class photographs includes exclusively three-dimensional real-world scenes color images. Figure **??** shows examples of images from both classes. From these images, and like [**?**,**?**],

Fig. 1. Sample images from each group. Row 1: Paintings, Row 2: Photographs.

we have generated 36 training sets where each set consists of 1000 paintings and 1000 photographs and the corresponding testing sets consist of the remaining images (i.e. 5000 paintings and 5000 photographs). Having the training data in hand, we apply our algorithm, presented in Section **??**, to the training vectors in both classes. After this stage, each class in the database is represented by an infinite generalized Gaussian mixture. Finally, in the classification stage each unknown image is assigned to the class increasing more its loglikelihood. Table **??** summarizes the classification results when considering different learning approaches and scenarios. According to this table, we can see clearly that both considered infinite models (IGM and IGGM) outperform the classification approach used in [**?**,**?**] and based on neural networks. Moreover, the results are improved further when feature selection is considered. It is noteworthy that we have applied a sensitivity test using different values of the hyperparameters with the student t test and found that the difference was not statistically significant (for photographs: $0.527497 \leq P\text{-value} \leq 0.544464$, and for paintings: $0.515763 \leq P\text{-value} \leq 0.537951$).

Table 1
Average classification accuracies (%) (± standard deviation) obtained using different approaches for distinguishing paintings from photographs. IGM: infinite Gaussian mixture, IGM + FS: infinite Gaussian mixture with feature selection, IGGM: infinite generalized Gaussian mixture, IGGM + FS: infinite generalized Gaussian mixture with feature selection.

Approach	Photographs	Paintings
[?,?] using $\{E_g, U, R, S\}$	71.00 (±4.00)	72.00 (±5.00)
[?,?] using RGBXY space	81.00 (±3.00)	81.00 (±3.00)
[?,?] using gray-scale-texture feature	78.00 (±4.00)	79.00 (±4.00)
[?,?] using all features	92.00 (±2.00)	94.00 (±3.00)
IGM using $\{E_g, U, R, S\}$	69.00 (±5.50)	70.50 (±6.25)
IGM + FS using $\{E_g, U, R, S\}$	73.00 (±4.50)	74.25 (±6.00)
IGM using RGBXY space	80.75 (±4.25)	80.50 (±4.00)
IGM + FS using RGBXY space	83.00 (±3.75)	83.75 (±3.25)
IGM using gray-scale-texture feature	76.25 (±3.75)	77.00 (±3.25)
IGM + FS using gray-scale-texture feature	82.75 (±3.25)	82.00 (±3.50)
IGM using all features	89.00 (±4.75)	90.00 (±5.25)
IGM + FS using all features	93.75 (±3.25)	93.50 (±3.00)
IGGM using $\{E_g, U, R, S\}$	72.50 (±3.50)	73.50 (±4.00)
IGGM + FS using $\{E_g, U, R, S\}$	77.50 (±2.50)	78.50 (±3.00)
IGGM using RGBXY space	82.75 (±3.50)	81.50 (±3.50)
IGGM + FS using RGBXY space	87.25 (±2.25)	86.75 (±2.75)
IGGM using gray-scale-texture feature	80.25 (±2.75)	80.00 (±3.25)
IGGM + FS using gray-scale-texture feature	84.25 (±2.75)	84.75 (±1.75)
IGGM using all features	92.00 (±3.75)	93.00 (±3.25)
IGGM + FS using all features	97.25 (±1.50)	96.75 (±1.75)

4.2 Image and Video Segmentation

Image and video segmentation is one of the major and basic steps in digital multimedia processing which has the objective of extracting information from an image or a sequence of images (video). It consists in partitioning the given image (or video) into homogeneous spatial (spatiotemporal in the case of videos) regions enjoying similar properties such as texture, color, boundary, and intensity. It is via segmentation that regions of interest are extracted

for subsequent processing such as object detection and recognition and for further applications such as content-based image retrieval. Various methods have been proposed in the literature and tremendous advancements have been made within the past decade (see, for instance, [**?**,**?**]). Contributions, however, continue to be made in the formulation of new mathematical and statistical approaches and in the developments of new algorithms. The discussion of all these previous approaches is clearly beyond the scope of this paper. As a formal well-established approach to clustering, finite mixture models have been widely used for image segmentation. In particular finite generalized Gaussian mixture models have provided excellent segmentation results [**?**,**?**]. The effectiveness of the segmentation to yield meaningful regions depends greatly on the choice of the number of clusters. This problem has been tackled in [**?**] and [**?**] using minimum message length (MML) and Bayes factor criteria. Here we propose the application of our infinite model in order to reduce further over/under-segmentation problems. Our model takes also into account the fact that, usually, in video/image segmentation, some features are noisy, redundant, or irrelevant for the segmentation. The presence of these irrelevant features introduces a bias to distances between objects which may effect the homogeneity of regions as discussed in some recent works that have shown that feature weighting and selection generally improve segmentation results [**?**,**?**]. Color and texture are important segmentation cues [**?**,**?**], thus we have used for each pixel a 27 features vector that combines both information. For color information, we have chosen the RGB color space, as for texture information 24 features calculated from the color correlogram of the pixel neighborhood, as defined in [**?**], have been considered.

The images used in our experiments are from the Berkeley benchmark [**?**]. We have chosen this dataset because a ground truth (GT) (i.e., segmentation performed manually) is provided for each image in the dataset. We have compared the segmentation results obtained using our proposed approach (IGGM+FS) with those obtained using: 1) The Gaussian mixture with MML and without feature selection (GM); 2) The infinite Gaussian mixture without feature selection (IGM); 3) The generalized Gaussian mixture with MML and without FS (GGM); 4) The infinite generalized Gaussian mixture without feature selection (IGGM); 5) The Gaussian mixture with MML and feature selection (GM+FS); 6) The infinite Gaussian mixture with feature selection (IGM+FS); 7) The generalized Gaussian mixture with MML and with FS (GGM+FS). In order to have a quantitative evaluation of the performance, we have used two objective criteria namely Boundary localization error (E_1) and the over/under-segmentation error (E_2). E_1 measures the misalignment of regions between a tested segmentation (TS) and the GT and is defined as [**?**]:

$$E_1 = \frac{1}{N} \sum_{(u,v)} \min\{E_{(u,v)}(TS, GT), E_{(u,v)}(GT, TS)\} \tag{37}$$

(a)

(b) (c) (d) (e)

(f) (g) (h) (i)

Fig. 2. Segmentation results for the first image from the Berkeley dataset. (a) GT, (b) GM, (c) IGM, (d) GGM, (e) IGGM, (f) GM+FS, (g) IGM+FS, (h) GGM+FS, (i) IGGM+FS.

(a)

(b) (c) (d) (e)

(f) (g) (h) (i)

Fig. 3. Segmentation results for the second image from the Berkeley dataset. (a) GT, (b) GM, (c) IGM, (d) GGM, (e) IGGM, (f) GM+FS, (g) IGM+FS, (h) GGM+FS, (i) IGGM+FS.

(a)

(b) (c) (d) (e)

(f) (g) (h) (i)

Fig. 4. Segmentation results for the third image from the Berkeley dataset. (a) GT, (b) GM, (c) IGM, (d) GGM, (e) IGGM, (f) GM+FS, (g) IGM+FS, (h) GGM+FS, (i) IGGM+FS.

where $E_{(u,v)}(TS, GT) = \frac{|S_{TS} - S_{GT}|}{|S_{TS}|}$ and $E_{(u,v)}(GT, TS) = \frac{|S_{GT} - S_{TS}|}{|S_{GT}|}$. $S_{TS}(u, v)$ and $S_{GT}(u, v)$ are the segments (where the segment is defined as a connected set of 5 pixels or more) containing the pixel (u, v) in the TS and the GT, respectively. Note that the symbol (–) means the set difference operator and N is the number of pixels in the image. E_2 measures the amount of over/under-segmentation produced by each TS when compared to the GT. E_2 is defined as the sum of the number of segments in the GT that are over-segmented in the TS, and the number of segments in the TS that are over-segmented in the GT as suggested in [**?**].

Figures **??**, **??**, and **??** show the segmentation results for 3 different images from the Berkeley dataset when applying the 8 different methods. Table **??**

shows the different values of E_1 and E_2 for each model when considering the whole dataset. We can conclude that the IGGM+FS outperformed all

Table 2
Errors (E_1 and E_2) calculation for the Berkeley dataset.

Errors (E_1; E_2)							
GM	IGM	GGM	IGGM	GM+FS	IGM+FS	GGM+FS	IGGM+FS
(0.21; 23)	(0.21; 21)	(0.19; 20)	(0.17; 18)	(0.14; 15)	(0.13; 13)	(0.09; 11)	(0.08; 9)

other methods in both performance criteria. This can be explained by the fact that the generalized Gaussian is more flexible and by the fact that adopting Bayesian approach allows to account for the effect of uncertainty in the modeling parameters on the subsequent segmentation. We can see clearly also that using feature selection in both generalized Gaussian and Gaussian mixture models yields better performance than without using feature selection which is actually expected and meets the conclusions reached in some previous works [**?**].

In the case of videos, the segmentation problem is more challenging and depends on different factors such as lighting conditions, partial occlusion, rotation in depth and scale changes [**?**]. Moreover, the segmentation model has to be adapted in time to take into account the dynamical nature of the video scenes. Indeed, the number of regions, the saliency of the used features and the segmentation model's parameters can change from one frame to another. The majority of the previous works that have used mixture models for video segmentation assume a fixed number of components and just update the component's parameters. Here we use our infinite model, which takes implicitly into account the updating problem, by considering the same visual feature described in the image segmentation part and by adopting the formulation proposed in [**?**]. We have investigated our model via two widely used videos (Akiyo and Suzie). In order to demonstrate the robustness of our method we have used the two objective criteria (E_1 and E_2) introduced above. Figures **??** and **??** show two examples of video segmentation using different tested approaches. From each video, we show a frame drawn randomly from the sequence. Table **??** shows the segmentation results in terms of E_1 and E_2. We can see clearly the improvement brought by the proposed model against the compared ones. These results are confirmed visually in the segmentations shown in figures **??** and **??**, where the quality of object segmentation is clearly improved using the proposed approach.

4.3 Infrared Facial Expression Recognition

Face expression analysis and recognition has been one of the fastest growing areas of computer vision over the last few years [**?**,**?**], due primarily to the rapidly increasing demand for emotion analysis, biometrics, and image retrieval to ensure security and safety. Face expression recognition (FER) is

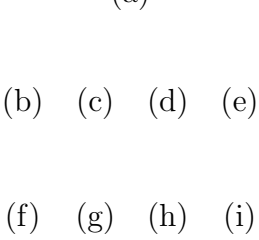

Fig. 5. Sample image from Akiyo video. (a) Sample frame, (b) GM, (c) IGM, (d) GGM, (e) IGGM, (f) GM+FS, (g) IGM+FS, (h) GGM+FS, (i) IGGM+FS.

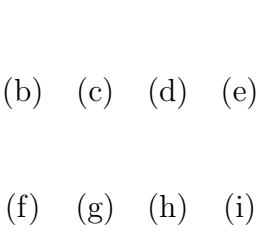

Fig. 6. Sample image from Suzie video. (a) Sample frame, (b) GM, (c) IGM, (d) GGM, (e) IGGM, (f) GM+FS, (g) IGM+FS, (h) GGM+FS, (i) IGGM+FS.

Table 3
Errors (E_1 and E_2) calculation for the 2 tested videos when using our infinite model (IGGM+FS), finite Gaussian mixture (GM), finite Gaussian mixture with feature selection (GM+FS), finite generalized Gaussian (GGM), finite generalized Gaussian with feature selection (GGM+FS), infinite Gaussian mixture (IGM), infinite Gaussian mixture with feature selection (IGM+FS) and infinite generalized Gaussian (IGM).

Errors (E_1; E_2)		
Video	Akiyo	Suzie
Size	300 frames	150 frames
GM	(0.23; 22.50)	(0.25; 27.40)
IGM	(0.22; 21.70)	(0.24; 24.90)
GGM	(0.22; 20.50)	(0.24; 23.70)
IGGM	(0.20; 19.40)	(0.21; 20.40)
GM+FS	(0.16; 17.80)	(0.18; 19.10)
IGM+FS	(0.15; 16.20)	(0.16; 15.60)
GGM+FS	(0.12; 13.20)	(0.12; 11.30)
IGGM+FS	(0.11; 11.70)	(0.11; 10.80)

interested in applying machine vision and pattern recognition algorithms on both still images and/or image video sequences in order to extract emotional

content from visual patterns of a person's face. Most of FER systems rely on videos as their input [**?**,**?**], however, video sequences are not always available in every real world situation. Thus, different FER image based approaches have been developed [**?**,**?**] in order to offer a reliable alternative. FER process can be divided into three main tasks: region of interest selection, feature extraction, and image classification. Region of interest (ROI) selection is used to identify areas where feature extraction will take place (e.g. the entire face [**?**], eyes, and mouth [**?**]). In order to decrease the dimensionality of different ROI, they are usually represented in terms of low-level feature vectors in lower dimensional feature space [**?**,**?**]. Image classification task identifies the emotional state of the input person face by searching a database of known different emotional expressions.

Facial analysis systems relying on visual spectrum have received relatively more attention compared to the thermal infrared one. This was justified by both the higher cost of thermal sensors, the lack of widely available IR image databases and the quality of the produced images (lower resolution and higher image noise). Recently, however, thermal imagery of human faces has been established as a valid biometric signature and several approaches have been proposed thanks to the advances of infrared imaging technology [**?**,**?**].

In this section, the aim is to implement an infrared multi-class face expression recognition algorithm based on our infinite model (IGGM+FS). The proposed FER system can be divided into three steps: face localization, facial feature estimation, feature selection and emotions identification. Normally, the position of the face is not centered within the image and can change greatly. Figure **??** shows examples of faces with different expressions taken from different poses. Face localization is used to identify the approximate position of each subject's face within the image. Infrared face localization is based on the idea that higher image intensities correspond to region with higher temperature which correspond to the face in our case. First, we have used a thresholding operation over the entire image which makes facial pixel intensities more prominent. Now for the n pixels remaining (with values over the threshold value) taken as position vectors $p = (p_x,p_y)$ we can compute the geometric centroid μ=(μ_x, μ_y) and μ_j=$\frac{\sum_{i=1}^{n} \mu_{j,i}}{n}$. In order to calculate μ_x we look for the facial pixels with the lowest thermal content along μ_y which will be centered at the person nose [**?**]. Figure **??** shows how by applying thresholding on the thermal image of a person's face we can easily locate the center even under different poses. Most of the research done until now considers that emotional information is centered around the eyes and mouth areas on the face. So in order to localize our facial features we have chosen an area of 120×130 centered around the image centroid μ which should be appropriate to our dataset as argued in [**?**]. In order to estimate the important features in this area we used the method of [**?**] that allows us to get 75 key points. Figure **??** shows an image from our dataset with the 75 interest points detected on the person's face. After detecting interest points we have applied the K mean approach as implemented in

[?] in order to identify eyes and mouth areas on the face. From figure ?? we can see clearly that this method was able to identify these areas effectively. The texture information in these area has been then represented using the texture descriptors proposed and used in [?].
In our experiments, we have performed face recognition using images from the Iris thermal face which is a subset of the Object Tracking and Classification Beyond the Visible Spectrum (OTCBVS) database. Images are gray-scale infrared of 320×240 each and represent different persons under different expressions and poses. We used 756 images for 28 different persons with three different expressions: surprise, happy, and angry. Figure ?? shows images from different classes (with different emotions). We have used 9 images for each person as training set and the rest as testing set. This gave us 252 and 504 images for training and testing, respectively. We have also applied the 7 other methods introduced above. Tables ??.a, ??.b, ??.c, ??.d, ??.e, ??.f, ??.g, and ??.h are the corresponding confusion matrices.
In order to evaluate the quality of clustering we have used two different criteria: accuracy and normalized mutual information. Accuracy is a simple and transparent evaluation measure that computes the percentage of images correctly clustered to the total number of images. Normalized mutual information (NMI) [?] is a novel criterion used for classifier evaluation based on information theory that is always between 0 and 1 (The larger this value, the better the clustering performance):

$$NMI(\Omega, \mathtt{C}) = \frac{I(\Omega, \mathtt{C})}{[H(\Omega) + H(\mathtt{C})]/2} \tag{38}$$

where $\mathtt{C} = (\mathtt{C}_1, \ldots, \mathtt{C}_M)$ are the classes that represent the data (in this application M=3) and $\Omega = (\Omega_1, \ldots, \Omega_K)$ are the clusters identified by the classification algorithm. I is the mutual information given by:

$$I(\Omega, \mathtt{C}) = \sum_{m=1}^{M} \sum_{k=1}^{K} P(\Omega_k \cap \mathtt{C}_m) \log \frac{P(\Omega_k \cap \mathtt{C}_m)}{P(\Omega_k) P(\mathtt{C}_m)} \tag{39}$$

where $P(\Omega_k)$, $P(\mathtt{C}_m)$, and $P(\Omega_k \cap \mathtt{C}_m)$ are the probabilities of an image being in cluster Ω_k, class $\mathtt{C}_m$, and in the intersection of Ω_k and $\mathtt{C}_m$, respectively. H is the entropy where

$$\begin{aligned} H(\Omega) &= -\sum_{k=1}^{K} P(\Omega_k) \log P(\Omega_k) \\ H(\mathtt{C}) &= -\sum_{m=1}^{M} P(\mathtt{C}_m) \log P(\mathtt{C}_m) \end{aligned} \tag{40}$$

Table ?? shows the different accuracies and NMI for the dataset when applying the eight methods. According to this table it is clear that the IGGM+FS

Fig. 7. Sample images from each group. Row 1: Surprise, Row 2: Happy, Row 3: Angry.

Fig. 8. Processing steps shown for sample images from each group. Row 1: sample images, Row 2: Thresholding, Row 3: Center location, Row 4: Interest points detection, Row 5: Regions of interest extraction.

outperformed all other methods. Note that, the student t test has shown that the difference between the categorization accuracies of our infinite model when changing its hyperparameters is not statistically significant ($0.4548 \leq P\text{-value} \leq 0.7878$).

5 Conclusion

In this paper we have proposed a hierarchical infinite mixture model of generalized Gaussian distributions for visual learning based on non-parametric Bayesian estimation. The specific choice of infinite mixture models is motivated by the fact that they combine flexibility in modeling, clarity of interpretation and intuitive analysis which is crucial in statistical inference from image data generally supposed to be generated from different sources. We have shown that fully Bayesian models provide a rigorous framework for challenging applications due to its ability to handle uncertainties associated with the involved data, by incorporating prior knowledge, and to its deep foundation on probability inference. According to the results, it is clear that performing feature selection in tandem with infinite mixture models leads to excellent clustering results and avoids overfitting. The experiments show clearly the broad applicability and generality of the proposed approach which is able to infer at the same time both meaningful clusters and meaningful features.
The adoption of generalized Gaussian is supported by various studies in image statistics which have shown that the statistics of natural images are generally non-Gaussian. The Bayesian clustering approach via infinite mixture models

Table 4
Confusion matrices for the infrared facial expression recognition application using: (a) GM, (b) IGM, (c) GGM, (d) IGGM, (e) GM+FS, (f) IGM+FS, (g) GGM+FS, (h) IGGM+FS.

	Happy	Angry	Surprise
Happy	101	35	32
Angry	23	131	14
Surprise	35	15	118

(a)

	Happy	Angry	Surprise
Happy	109	31	28
Angry	22	137	9
Surprise	28	13	127

(b)

	Happy	Angry	Surprise
Happy	119	27	22
Angry	21	141	6
Surprise	24	11	133

(c)

	Happy	Angry	Surprise
Happy	124	23	21
Angry	21	143	4
Surprise	22	9	137

(d)

	Happy	Angry	Surprise
Happy	115	27	26
Angry	20	139	9
Surprise	26	11	131

(e)

	Happy	Angry	Surprise
Happy	121	26	21
Angry	18	144	6
Surprise	23	7	138

(f)

	Happy	Angry	Surprise
Happy	126	23	19
Angry	9	157	2
Surprise	21	5	142

(g)

	Happy	Angry	Surprise
Happy	133	21	14
Angry	7	161	0
Surprise	16	3	149

(h)

is clearly attractive in part due to recent advances in MCMC techniques which allows straightforward posteriors computation. However, it is also hindered by the very high computational cost. A possible future work could be the development of a variational approach for the learning of the proposed model which shall allow tremendous savings in time and computation over MCMC methods.

Table 5
Accuracies and NMI when applying the 8 methods for the infrared facial expression recognition.

	Accuracy(%)	NMI
IGGM+FS	87.90 (±2.15)	0.6249 (±0.0143)
GGM+FS	84.33 (±2.60)	0.5389 (±0.0173)
IGM+FS	79.96 (±3.35)	0.4407 (±0.0127)
GM+FS	76.39 (±4.65)	0.3713 (±0.0261)
IGGM	80.15 (±3.10)	0.4439 (±0.0206)
GGM	77.98 (±2.90)	0.4010 (±0.0193)
IGM	74.01 (±4.20)	0.3348 (±0.0201)
GM	69.44 (±4.70)	0.2668 (±0.0182)

Acknowledgment

The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors would like to thank Dr. Riad Hammoud for stimulating discussions on the problem of distinguishing paintings from photographs and for providing the related data sets.