Noname manuscript No.
(will be inserted by the editor)

Bayesian Learning of Inverted Dirichlet Mixtures for SVM Kernels Generation

Taoufik Bdiri and Nizar Bouguila

Taoufik Bdiri is with the Electrical and Computer Engineering Department (ECE), Concordia University, Montreal, QC H3G 1T7, Canada (email: t_bdiri@encs.concordia.ca).
Nizar Bouguila is with the Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montreal, QC H3G 1T7, Canada (email: bouguila@ciise.concordia.ca).

Abstract We describe approaches for positive data modeling and classification using both finite inverted Dirichlet mixture models and support vector machines (SVMs). Inverted Dirichlet mixture models are used to tackle an outstanding challenge in SVMs namely the generation of accurate kernels. The kernels generation approaches, grounded in ideas from information theory, that we consider allow the incorporation of data structure and its structural constraints. Inverted Dirichlet mixture models are learned within a principled Bayesian framework using both Gibbs sampler and Metropolis-Hastings for parameter estimation and Bayes factor for model selection (i.e. determining the number of mixture's components). Our Bayesian learning approach uses priors, that we derive by showing that the inverted Dirichlet distribution belongs to the family of exponential distributions, over the model parameters, and then combines these priors with information from the data to build posterior distributions. We illustrate the merits and the effectiveness of the proposed method with two real-world challenging applications namely object detection and visual scenes analysis and classification.

Key words Mixture models, SVM, hybrid models, inverted Dirichlet, Bayesian inference, Bayes factor, model selection, Gibbs sampling, kernels, object detection, image databases

1 Introduction

The increasing availability of large volumes of data has caused an urgent demand for the deployment of statistical models for the analysis of these data [1–4]. Several trends have emerged and there has been a large growth in the number of statistical methods. A review of many of these methods and techniques can be found in [5]. Mixture models are among the most widely used statistical approaches and provide a principled approach for performing inference on heterogenous data in which vectors are supposed to be drawn from different distributions [6]. Indeed, finite mixture models have become arguably among the representations of choice to model data and uncertainty and have been successfully applied in many pattern recognition, computer vision and data mining applications [6–10]. A significant weakness of many learning models that have considered mixtures of distributions is their reliance on the Gaussian assumption. In fact, it is well-known that inference in statistical models is generally sensitive to modeling assumptions especially the choice of probability density functions. For instance, we have shown in our previous works that other distributions namely the Dirichlet and the generalized Dirichlet offer better modeling capabilities in the case of proportional data [11–13]. In this paper, we tackle another problem, namely positive data modeling and classification. Indeed, such data arise naturally in many real applications [14,15]. In particular, we propose the consideration of inverted Dirichlet mixture models which have been largely ignored in the past despite the fact that they offer both flexibility and ease of use as we shall show.

One of the hard problems in the case of finite mixture models is how to effectively learn the mixture model parameters from incomplete data (i.e. in the presence of missing variables). Several approaches have been proposed in the past. The maximum likelihood approach, based on the iterative expectation-maximization (EM) framework [16], is perhaps the most popular technique. It is well-known, however, that the maximum likelihood approach depends on the used initialization algorithm and does not allow the selection of the appropriate number of mixture components which is of central importance [10]. In [17], we introduced an EM-based approach that combines the standard EM algorithm for parameters optimization and the minimum message length (MML) criterion for model selection. In recent years, however, there has been much interest in Bayesian learning of finite mixture models to resolve inherent problems re-

lated to EM-based approaches. Indeed, several recent researches advocated the use of Bayesian techniques which have been considered in a variety of applications (see, for instance, [18–20,13]). This can be justified by the fact that Bayesian inference handles uncertainty in a natural manner and embodies Occam's razor the principle that favors simple but accurate models. With Bayesian approaches, we need fundamentally to combine our prior beliefs about the parameters with the data to obtain posteriors from which we obtain samples using Markov Chain Monte Carlo (MCMC) techniques [21]. Thus, the learning approach that we propose in this paper capitalizes on this trend providing a reliable framework for positive data clustering and a richer inference than the previously proposed EM-based algorithm [14,17]. An important problem that we address within the proposed Bayesian framework is the determination of the number of mixture components that best describe a given data set. The problem is challenging and has been the subject of extensive research in the past (see, for instance, [22]). In our framework we tackle it using Bayes factors. For more details about Bayesian learning and Bayes factors the reader is strongly encouraged to refer to [23].
The adoption of finite mixture models can be viewed also as an approach that strives to generate classification rules from examples using the well-known Bayes rule. The main idea is to compare the *a posteriori* probabilities of all classes and assign each vector to the class with the highest probability. Several other approaches have been proposed for the design of statistical classifiers and this topic has been the subject of intense study in the pattern recognition and machine learning communities [24]. These approaches can be grouped into two families namely generative and discriminative techniques. Generative approaches (e.g. finite mixture models) are widely adopted when the number of training data is small. On the other hand, if the task to be performed is classification, and a large number of training examples are available, then discriminative techniques are generally deployed to learn classification rules [25]. SVM is perhaps one of the most important techniques which has been widely applied in problem solving in several areas due to their potential to greatly increase classification accuracy and generalization capabilities [26,27]. As a discriminative approach the goal of SVM is to find surfaces that better separate the different data classes. The main ingredient of SVM, which was developed based on the structural risk minimization principle from statistical learning theory, is the kernel trick which allows efficient discrimination in non-linearly separable input feature spaces. Thus, a main problem and one of the important challenges, when using SVM, is the choice of the kernel function which has to be suitable for the data to classify and the general task to solve [28,29]. Classic kernels include linear, polynomial and radial basis function (RBF) kernels. Unfortunately, these kernels cannot be applied in applications where the objects to classify are represented by sequences (or bags of vectors) which may not have the same length. Bags of vectors occur frequently and naturally in several applications. For instance, an image or a video can be represented by several feature vectors. The importance of this problem has motivated intense research and the main natural question that have arisen is wether we can use the data itself to build SVM kernels. As a result, approaches, often referred to as hybrid generative discriminative learning, have been proposed and have provided a systematic and a reproducible properly motivated approach to classification [30]. Hybrid approaches allow the conversion of data of non-fixed lengths into fixed length, and seek to get the best of both approaches and then decrease generalization and prediction errors. In this work, we also propose and derive some kernels for the classification of objects represented by sequences of positive vectors using a hybrid of inverted Dirichlet mixture models and SVM. Our approaches build on the recent progress made in hybrid generative discriminative learning techniques [31] such as Fisher kernel [30], Kullback-Leibler Divergence Kernels [32], Rényi and Jensen-Shannon Kernels [33] and product kernels [34]. To illustrate the merits of our approach we shall consider several challenging applications. In particular, we show that Bayesian methods, coupled with MCMC computational techniques can be successfully applied in the analysis of complex data sets and the generation of accurate SVMs kernels [1].
The plan of this paper is as follows. In Section 2 we describe our generative model namely the inverted Dirichlet mixture and state a fully Bayesian approach for its learning. In Section 3, SVM kernels generated from the inverted Dirichlet mixture are proposed for sequences classification. Then we present our experimental results in Section 4. Finally, we conclude the paper with a brief discussion and a summary of the work in Section 5.

2 Bayesian Model Specification

In this section, we first briefly present the inverted Dirichlet mixture model. An important step that we discuss next is how to set up appropriate prior structure for the model and then compute the necessary posteriors for sampling.

2.1 The Inverted Dirichlet Mixture Model

Let $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ denotes N D-dimensional positive vectors, $\mathbf{X}_i$, of measurements on N objects (e.g. images, documents) to be clustered. We define our model to be a mixture of inverted Dirichlet distributions. Mixture

[1] It is noteworthy that some Bayesian interpretations of SVMs have been proposed in the past (see, for instance, [35–37]).

models allow the representation of a probability distribution as a linear superposition of component distributions [14]:

$$p(\mathbf{X}_i|\Theta) = \sum_{j=1}^{M} p_j p(\mathbf{X}_i|\boldsymbol{\alpha}_j) \tag{1}$$

where j labels the component, p_j denotes the weight of component j (the weights are positive and sum to one), M is the number of mixture components that must be inferred from the data, $\Theta = (\{p_j\}, \{\boldsymbol{\alpha}_j\})$ denotes the set of all involved parameters, and $\boldsymbol{\alpha}_j$ represents the parameters of the inverted Dirichlet distribution representing cluster j:

$$p(\mathbf{X}_i|\boldsymbol{\alpha}_j) = \frac{\Gamma(|\boldsymbol{\alpha}_j|)}{\prod_{d=1}^{D+1}\Gamma(\alpha_{jd})} \prod_{d=1}^{D} X_{id}^{\alpha_{jd}-1} (1 + \sum_{d=1}^{D} X_{id})^{-|\boldsymbol{\alpha}_j|} \tag{2}$$

where $X_{id} > 0, d = 1, 2, \ldots, D$, $\boldsymbol{\alpha}_j = (\alpha_{j1}, \ldots, \alpha_{jD+1})$ is the vector of parameters and $|\boldsymbol{\alpha}_j| = \sum_{d=1}^{D+1} \alpha_{jd}$, $\alpha_{jd} > 0, d = 1, 2, \ldots, D+1$.

In mixture learning frameworks (both EM-based and Bayesian), the estimation problem is generally phrased as a missing data problem, tackled using the EM algorithm, where the complete data are $\{(\mathbf{X}_i, \mathbf{Z}_i)\}$, where $\mathbf{Z}_i = (Z_{i1}, \ldots, Z_{iM})$ is called the membership vector. The hidden variables $Z_{ij}, j = 1, \ldots, M$ are indicators representing which inverted Dirichlet generated which vector such that $Z_{ij} = 1$ if $\mathbf{X}_i$ belongs to cluster j and $Z_{ij} = 0$, otherwise. The E- step of the EM computes the posterior probabilities $\hat{Z}_{ij}$ given by

$$\hat{Z}_{ij} = \frac{p_j p(\mathbf{X}_i|\boldsymbol{\alpha}_j)}{\sum_{j=1}^{M} p_j p(\mathbf{X}_i|\boldsymbol{\alpha}_j)} \tag{3}$$

and the M- step then maximizes the expected value of the complete data likelihood. The main problem of EM-based algorithms is the dependency on initialization. Bayesian approaches have been proposed as an alternative to likelihood-based inference [38,39]. In this work we shall consider Bayesian inference by considering the parameters vector Θ of the model to be random variables. We are mainly motivated by the success of Bayesian analysis in handling difficult problems in statistical analysis in general and the problem of mixtures learning in particular. The main goal is to determine the conditional distribution of Θ given the training data $\mathcal{X}$ (i.e. posterior distribution) and the structure of the model M (i.e. the number of clusters in this case). We must therefore select a prior distribution $p(\Theta)$ and then compute the resulting posterior distribution $p(\Theta|\mathcal{X})$ via the Bayes' theorem [40]:

$$p(\Theta|\mathcal{X}) = \frac{p(\mathcal{X}|\Theta)p(\Theta)}{p(\mathcal{X})} \propto p(\mathcal{X}|\Theta)p(\Theta) \tag{4}$$

where $p(\mathcal{X}|\Theta)$ is the likelihood of the data given the model parameters. In the following section, we state our prior distributions and then compute the related posteriors from which the mixture model's parameters will be generated.

2.2 Priors and Posteriors

Results can be very sensitive to the choice of the prior. A number of techniques have been suggested to the problem of assessing prior distributions (see, for instance, [41, 42]). Generally, however, techniques which are satisfying in some situations may not work well in other situations. Thus, we have preferred a formal approach, to develop a conjugate prior [2], based on the fact that the inverted Dirichlet belongs to the exponential family of distributions which has several good mathematical properties and has been largely studied in the literature [43,44]. Using this formal approach, the prior for the $\boldsymbol{\alpha}_j$ is given by the following (see Appendix 1):

$$p(\boldsymbol{\alpha}_j) \propto \exp\Bigg[\sum_{d=1}^{D} \rho_d \alpha_{jd} - \rho_{D+1}|\boldsymbol{\alpha}_j| + \kappa\Bigg(\log\big(\Gamma(|\boldsymbol{\alpha}_j|)\big) - \sum_{d=1}^{D+1}\log\Gamma(\alpha_{jd})\Bigg)\Bigg] \tag{5}$$

where $(\rho_1, \ldots, \rho_{D+1}, \kappa)$ is the set of hyperparameters governing the prior. Having this prior, the posterior distribution $p(\boldsymbol{\alpha}_j|\mathcal{Z}, \mathcal{X})$, where $\mathcal{Z} = (\mathbf{Z}_1, \ldots, \mathbf{Z}_N)$, associated with a class j is then

$$\begin{aligned} p(\boldsymbol{\alpha}_j|\mathcal{Z}, \mathcal{X}) &\propto p(\boldsymbol{\alpha}_j) \prod_{Z_{ij}=1} p(\mathbf{X}_i|\boldsymbol{\alpha}_j) \\ &\propto \exp\Bigg[\sum_{d=1}^{D}\Bigg(\rho_d + \sum_{Z_{ij}=1}\log X_{id}\Bigg)\alpha_{jd} \\ &+ \Bigg(\rho_{D+1} + \sum_{Z_{ij}=1}\log(1 + \sum_{d=1}^{D} X_{id})\Bigg)\alpha_{jD+1} \\ &+ (\kappa + n_j)\Bigg(\log\big(\Gamma(|\boldsymbol{\alpha}_j|)\big) - \sum_{d=1}^{D+1}\log(\Gamma(\alpha_{jd}))\Bigg)\Bigg] \end{aligned} \tag{6}$$

We can see clearly that the posterior and the prior distributions have the same form, then $p(\boldsymbol{\alpha}_j)$ is really a conjugate prior on $\boldsymbol{\alpha}_j$. As for the parameters p_j, we know that they are defined on the simplex $\{\mathbf{P} = (p_1, \ldots, p_M) : \sum_{j=1}^{M-1} p_j < 1\}$, then a natural choice, as a prior, is a Dirichlet distribution with parameters $\eta = (\eta_1, \ldots, \eta_M)$:

$$p(\mathbf{P}) \propto \prod_{j=1}^{M} p_j^{\eta_j - 1} \tag{7}$$

[2] A conjugate prior is a prior that shares the same parametric form as the posterior.

We know also that

$$p(\mathcal{Z}|\mathbf{P}) = \prod_{i=1}^{N} p(\mathbf{Z}_i|\mathbf{P}) = \prod_{i=1}^{N}\prod_{j=1}^{M} p_j^{Z_{ij}} = \prod_{j=1}^{M} p_j^{n_j} \quad (8)$$

where $n_j = \sum_{i=1}^{N} \mathbb{I}_{Z_{ij}=j}$. Thus,

$$p(\mathbf{P}|\mathcal{Z}) \propto p(\mathcal{Z}|\mathbf{P})p(\mathbf{P}) \propto \prod_{j=1}^{M} p_j^{\eta_j+n_j-1} \quad (9)$$

Having our posteriors, it is possible now to give the complete mixture estimation algorithm. Many computationally intensive Bayesian statistical analyzes have became practical thanks to the recent development of MCMC techniques such as the Gibbs sampler [45,46] that we adopt here. The steps of our Gibbs sampler are:

Algorithm 1

1. Initialization
2. Step t: For t=1,...
   (a) Generate $\mathbf{Z}_i^{(t)} \sim \mathcal{M}(1; \hat{Z}_{i1}^{(t-1)}, \ldots, \hat{Z}_{iM}^{(t-1)})$
   (b) Compute $n_j^{(t)} = \sum_{i=1}^{N} \mathbb{I}_{Z_{ij}^{(t)}=j}$
   (c) Generate $\mathbf{P}^{(t)}$ from Eq. 9
   (d) Generate $\boldsymbol{\alpha}_j^{(t)}$ $(j = 1, \ldots, M)$ from Eq. 6 using the Metropolis-Hastings (M-H) algorithm [47]

where $\mathcal{M}(1; \hat{Z}_{i1}^{(t-1)}, \ldots, \hat{Z}_{iM}^{(t-1)})$ denotes a Multinomial of order one with parameters $\hat{Z}_{i1}^{(t-1)}, \ldots, \hat{Z}_{iM}^{(t-1)})$. It is noteworthy that a M-H algorithm is used here to simulate from the $\boldsymbol{\alpha}_j$ posterior since it has not a usual known form. The M-H algorithm is now routinely used in these kind of situations and has been the topic of extensive theoretical and experimental studies in the past (see, for instance, [47]). As all the α_{jd} are positive, we have used a random walk M-H algorithm with log-normal distribution, as a proposal, with variance v^2. More details about the M-H algorithm can be found in [48]. A problem of critical importance when considering MCMC techniques is the convergence assessment [49–52]. Many techniques have been proposed and applied with success [53]. In our case we have assessed convergence to the stationary distribution using a diagnostic approach, based on a single long-run of the Gibbs sampler, proposed in [54], that has been shown to often work well in practice.

2.3 Model Selection

Generally complex models fit well the data, but will have poor generalization [3] capabilities (i.e. overfitting problem). This is especially true for high-dimensional real data involving images, videos, speech and text. It is important then to have a certain compromise between goodness of fit and the complexity of the learned model. Many works have been concerned with the development of model selection (i.e. the problem of choosing the best value of M) procedures and have been based generally on penalized likelihood criteria (see, for instance, [57]). A model selection approach that have been successful in several mixture-based applications is the consideration of the Bayes factor. The Bayes factor has been widely studied in the past, is efficient and is not sensitive to the initial values (see, for instance, [58–60]).

Suppose that we have two models M_{k_1} and M_{k_2} in our candidate set. Choosing M_{k_1} instead of M_{k_2} is determined by the Bayes factor $B_{k_1k_2}$ given by the following equation where all the candidates are supposed to have the same prior probability:

$$B_{k_1k_2} = \frac{p(\mathcal{X}|M_{k_1})}{p(\mathcal{X}|M_{k_2})} \quad (10)$$

where $p(\mathcal{X}|M_{k_1})$ is the marginal likelihood (known also as the evidence or the partition function in statistical physics) for model M_{k_1} given by

$$p(\mathcal{X}|M_{k_1}) = \int p(\mathcal{X}|\Theta_{k_1}, M_{k_1})p(\Theta_{k_1})d\Theta_{k_1} \quad (11)$$

where $p(\mathcal{X}|\Theta_{k_1}, M_{k_1})$ is the likelihood function. The previous marginal is usually approximated by the Laplace method as [58]:

$$p(\mathcal{X}|M_{k_1}) \approx (2\pi)^{N_{k_1}/2}|\Sigma|^{1/2}p(\mathcal{X}|\hat{\Theta}_{k_1}, M_{k_1})p(\hat{\Theta}_{k_1}) \quad (12)$$

where $\hat{\Theta}$ is the posterior mode, N_{k_1} is the number of free parameters in the mixture model, and Σ is the Hessian matrix which is actually asymptotically equal to the posterior covariance matrix evaluated at the posterior mode $\hat{\Theta}$. Note that the previous equation could be also approximated by $N^{N_{k_1}/2}p(\mathcal{X}|\hat{\Theta}_{k_1}, M_{k_1})$ [58], which gives us the MDL criterion [61]:

$$\log(p(\mathcal{X}|M_{k_1})) \approx \log(p(\mathcal{X}|\hat{\Theta}_{k_1}, M_{k_1})) - \frac{N_{k_1}}{2}\log(N) \quad (13)$$

Having the model selection approach, the complete Bayesian learning algorithm of the finite inverted Dirichlet mixture is as follows:

Algorithm 2

For each candidate value of M_{k_1}:

1. Apply Algorithm 1
2. Select the optimal model M* such that:

$$M^* = \arg\max_{M_{k_1}} \log p(\mathcal{X}|M_{k_1})$$

3 Deriving SVM Kernels for Positive Sequence Data

SVMs are among the most successful and well-established developments within the pattern recognition and machine learning communities. For more details and discussions about SVMs, the reader is referred to [62] and

[3] Some researchers have used the adjective "projectability", also (see, for instance, [55,56]).

references therein. A problem of prime importance when using SVM is the choice of appropriate kernels [63,64]. Generally classic kernels (e.g. RBF, polynomial, linear) have been widely used to classify objects when each object is represented by a single vector. However, many real world-world problems involve the representation of objects by non-standard data structures [65] such as sequences of vectors which cannot be handled using these widely used classic kernels. Thus, a crucial problem is to develop kernel functions defined on these kind of data. The problem of generating SVM kernels to classify sequences has been investigated by many researchers. In particular, researchers have in recent years intensified their study of approaches to generate kernels directly from the tackled data in an effort to surmount this barrier. The most successful techniques have been based on deriving kernels from probability distributions in order to extend SVMs ability to handle diverse forms of structured inputs such as bags of vectors, graphs and strings. There is a large and interesting literature on these techniques, so-called hybrid generative discriminative, which has been driven mainly by recent applications, such as objects categorization using local features [66], which have necessitated new approaches. The goal of this section is to present some techniques confined to generating SVM kernels to handle the classification of bags (or sequences) of positive vectors for which the classic widely used kernels are not appropriate.

3.1 Fisher Kernel

Suppose we are given a set of multimedia objects $\mathcal{O} = \{O_1, \ldots, O_T\}$ to classify, where each object O_t is described by a set of positive vectors $\mathcal{X}_t = (\mathbf{X}_1, \ldots, \mathbf{X}_{N_t})$. It is noteworthy that the number of vectors N_t needed to describe each vector may be different from one object to another (i.e. each set has its own size). It is clear that classic kernels cannot be used in this case to classify our objects. It is possible however to represent adequately each object as a finite inverted Dirichlet mixture model. To simplify the notation without loss of generality let $\mathcal{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$ and $\mathcal{X}' = (\mathbf{X}'_1, \ldots, \mathbf{X}'_N)$ be two sets, representing two different multimedia objects O and O' in $\mathcal{O}$, and modeled by two finite inverted Dirichlet mixtures $p(\mathbf{X}|\Theta)$ and $p'(\mathbf{X}'|\Theta')$, respectively. We can define the likelihoods of each both objects as $p(\mathcal{X}|\Theta) = \prod_{i=1}^{N} p(\mathbf{X}|\Theta)$ and $p'(\mathcal{X}'|\Theta') = \prod_{i=1}^{N'} p'(\mathbf{X}'|\Theta')$, respectively. The Fisher kernel was proposed initially in [30] and the main idea is to exploit the geometric structure on the statistical manifold by mapping each individual sequence into a single feature vector, defined in the gradient log-likelihood space. The resulted feature vector is called the Fisher score and defined as $U_{\mathcal{X}} = \frac{\partial p(\mathcal{X}|\Theta)}{\partial \Theta}$, where each component is the derivative of the log-likelihood with respect to a particular parameter of the mixture model. The kernel is then defined as $\mathcal{K}(\mathcal{X}, \mathcal{X}') = U_{\mathcal{X}} F(\Theta)^{-1} U_{\mathcal{X}'}$, where $F(\Theta)$ is the Fisher information matrix which role is less significant and then can be approximated by the identity matrix [30]. Thus, we can develop the Fisher kernel in the case of inverted Dirichlet mixtures by computing the gradient of $\log p(\mathcal{X}|\Theta)$ with respect to the model parameters p_j, by taking into account the fact that the p_j sum to one, and α_{jd}:

$$\frac{\partial \log p(\mathcal{X}|\Theta)}{\partial \alpha_{jd}} = \sum_{i=1}^{N} \Big[\hat{Z}_{ij} \big(\Psi(|\boldsymbol{\alpha}_j|) - \Psi(\alpha_{jd}) + \log(X_{id}) - \log(1 + \sum_{d=1}^{D} X_{id}) \big) \Big] \qquad j = 1, \ldots, M \quad d = 1, \ldots, D$$

$$\frac{\partial \log p(\mathcal{X}|\Theta)}{\partial \alpha_{jD+1}} = \sum_{i=1}^{N} \Big[\hat{Z}_{ij} \big(\Psi(|\boldsymbol{\alpha}_j|) - \Psi(\alpha_{jD+1}) - \log(1 + \sum_{d=1}^{D} X_{id}) \big) \Big]$$

$$\frac{\partial \log p(\mathcal{X}|\Theta)}{\partial p_j} = \sum_{i=1}^{N} \left[\frac{\hat{Z}_{ij}}{p_j} - \frac{\hat{Z}_{i1}}{p_1} \right] \qquad j = 2, \ldots, M$$

3.2 Probability Product kernels

Let $\mathcal{O}$ and $\mathcal{O}'$ two multimedia objects modeled by two finite inverted Dirichlet mixtures $p(\mathbf{X}|\Theta) = \sum_{j=1}^{M} p_j p(\mathbf{X}|\alpha_j)$ and $p'(\mathbf{X}'|\Theta') = \sum_{j=1}^{M'} p'_j p(\mathbf{X}'|\alpha'_j)$, respectively, defined on the space $[0, +\infty[$. Probability product kernels were proposed initially in [34] by replacing the kernel computation in the original sequence space by computation in the probability density function (PDF) space (i.e. the kernel becomes a measure of similarity between probability distributions) as the following: $\mathcal{K}(\mathcal{X}, \mathcal{X}') \Rightarrow \mathcal{K}_\rho(p(\mathbf{X}), p'(\mathbf{X})) = \int_0^{+\infty} p(\mathbf{X})^\rho p'(\mathbf{X})^\rho d\mathbf{X}$, where ρ is a parameter. The two important special cases of probability product kernels are the Bhattacharyya kernel obtained with $\rho = 1/2$ and the expected likelihood kernel obtained with $\rho = 1$ [34]. Here we take $\rho = 1/2$ which gives us the Bhattacharyya kernel for which it is possible to find a closed form expression when the mixture model is reduced to one inverted Dirichlet distribution (see Appendix 2):

$$\mathcal{K}_{\frac{1}{2}}(p(\mathbf{X}|\boldsymbol{\alpha}), p'(\mathbf{X}|\boldsymbol{\alpha'})) = \frac{\prod_{d=1}^{D+1} \Gamma(\frac{1}{2}\alpha_d + \frac{1}{2}\alpha'_d) \sqrt{\Gamma(\sum_{d=1}^{D+1} \alpha_d) \Gamma(\sum_{d=1}^{D+1} \alpha'_d)}}{\Gamma\big(\sum_{d=1}^{D+1} (\frac{1}{2}\alpha_d + \frac{1}{2}\alpha'_d)\big) \sqrt{\prod_{d=1}^{D+1} \Gamma(\alpha_d) \Gamma(\alpha'_d)}} \tag{14}$$

As for a mixture model we can use the following heuristic [34]:

$$\mathcal{K}_{\frac{1}{2}}(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')) = \sum_{j=1}^{M} \sum_{j'=1}^{M'} p_j p'_j K_{\frac{1}{2}}(p(\mathbf{X}|\theta_j), p'(\mathbf{X}|\theta'_j)) \tag{15}$$

It is noteworthy that the Bhattacharyya kernel has a cubic complexity [66], but has the main advantage in terms of nonlinear flexibility [34].

3.3 Information Divergence Kernels

Several information divergence-based kernels have been proposed in [33]. The first one is based on the symmetric Kullback-Leibler (KL) divergence and is given by:

$$\mathcal{K}_{KL}(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')) = \exp\Big[-AJ\big(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')\big)\Big] \tag{16}$$

where $A > 0$ is a kernel parameter included for numerical stability, and

$$\begin{aligned} &J\big(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')\big) \\ &= KL\big(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')\big) + KL\big(p'(\mathbf{X}|\Theta'), p(\mathbf{X}|\Theta)\big) \end{aligned} \tag{17}$$

is the symmetric KL divergence between $p(\mathbf{X}|\Theta)$ and $p'(\mathbf{X}|\Theta')$. The KL divergence has a closed-form expression in the case of the inverted Dirichlet distribution and is given by (see Appendix 3):

$$\begin{aligned} KL\big(p(\mathbf{X}|\theta), p'(\mathbf{X}|\theta')\big) &= \log\left[\frac{\Gamma(|\boldsymbol{\alpha}|)\prod_{d=1}^{D+1}\Gamma(\alpha'_d)}{\Gamma(|\boldsymbol{\alpha}'|)\prod_{d=1}^{D+1}\Gamma(\alpha_d)}\right] \\ + \sum_{d=1}^{D+1}(\alpha_d - \alpha'_d)\Psi(\alpha_d) &+ (\alpha_{d+1} - \alpha'_{d+1})\Psi(|\boldsymbol{\alpha}|) \end{aligned} \tag{18}$$

The Rényi kernel is another approach, based on the symmetric Rényi divergence [67], that has been proposed in [33]

$$\begin{aligned} &\mathcal{K}(\mathcal{X}, \mathcal{X}') \Rightarrow \mathcal{K}_R\big(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')\big) \\ &= \exp\Big[-AR\big(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')\big)\Big] \end{aligned} \tag{19}$$

where

$$\begin{aligned} &R\big(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')\big) = \\ &= \frac{1}{\sigma-1}\log\int_0^{+\infty} p(\mathbf{X}|\Theta)^{\sigma} p'(\mathbf{X}|\Theta')^{1-\sigma} d\mathbf{X} \\ &+ \frac{1}{\sigma-1}\log\int_0^{+\infty} p'(\mathbf{X}|\Theta')^{\sigma} p(\mathbf{X}|\Theta)^{1-\sigma} d\mathbf{X} \end{aligned} \tag{20}$$

where $\sigma > 0$ and $\sigma \neq 1$ is the order of Rényi divergence. By substituting Eq. 20 into Eq. 19, we obtain the following

$$\begin{aligned} \mathcal{K}_R(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')) &= \Big[\int_0^{+\infty} p(\mathbf{X}|\Theta)^{\sigma} p'(\mathbf{X}|\Theta')^{1-\sigma} d\mathbf{X} \\ \times \int_0^{+\infty} p'(\mathbf{X}|\Theta')^{\sigma} p(\mathbf{X}|\Theta)^{1-\sigma} d\mathbf{X}\Big]^{\frac{A}{1-\sigma}} \end{aligned} \tag{21}$$

In the case of an inverted Dirichlet distribution, we can find a closed-form expression for the Rényi divergence (See Appendix 4):

$$\begin{aligned} &\int_0^{+\infty} p(\mathbf{X}|\theta)^{\sigma} p'(\mathbf{X}|\theta')^{1-\sigma} d\mathbf{X} \\ &= \left[\frac{\Gamma(|\boldsymbol{\alpha}|)}{\prod_{d=1}^{D+1}\Gamma(\alpha_d)}\right]^{\sigma}\left[\frac{\Gamma(|\boldsymbol{\alpha}'|)}{\prod_{d=1}^{D+1}\Gamma(\alpha'_d)}\right]^{1-\sigma} \\ &\times \left[\frac{\Gamma(\prod_{d=1}^{D+1}\Gamma(\alpha'_d - \sigma\alpha'_d + \sigma\alpha_d))}{\sum_{d=1}^{D+1}(\alpha'_d - \sigma\alpha'_d + \sigma\alpha_d)}\right] \end{aligned} \tag{22}$$

The last kernel is the Jensen-Shannon (JS) Kernel, generated according to the Jensen-Shannon divergence [68], and is given by [33]

$$\begin{aligned} &\mathcal{K}_{JS}(\mathcal{X}, \mathcal{X}') \Rightarrow \mathcal{K}(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')) \\ &= \exp\Big[-AJS_{\omega}(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta'))\Big] \end{aligned} \tag{23}$$

where

$$\begin{aligned} JS_{\omega}(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')) &= H\big[\omega p(\mathbf{X}|\Theta) + (1-\omega)p'(\mathbf{X}|\Theta')\big] \\ &- \omega H\big[p(\mathbf{X}|\Theta)\big] - (1-\omega)H\big[p'(\mathbf{X}|\Theta')\big] \end{aligned}$$

where ω is a parameter and

$$H[p(\mathbf{X}|\Theta)] = -\int_0^{+\infty} p(\mathbf{X}|\Theta)\log p(\mathbf{X}|\Theta) d\mathbf{X} \tag{24}$$

is the Shannon entropy and we can show that is given by the following in the case of the inverted Dirichlet distribution (See Appendix 5)

$$\begin{aligned} H[p(\mathbf{X}|\theta)] &= -\log\Gamma(|\boldsymbol{\alpha}_j|) + \sum_{d=1}^{D+1}\log\Gamma(\alpha_{jd}) \\ &- \sum_{d=1}^{D}(\alpha_{jd} - 1)(\Psi(\alpha_d) - \Psi(|\boldsymbol{\alpha}|)) + |\boldsymbol{\alpha}_j|\Psi(|\boldsymbol{\alpha}|) \end{aligned} \tag{25}$$

4 Experimental Results

In this section, we investigate our approach using real life challenging applications. The first goal of these applications is to verify the capabilities of our Bayesian learning algorithm as compared to a learning algorithm, based on maximum likelihood (ML) estimation and minimum message length selection (ML+MML), previously proposed in [17]. The second goal is to compare the performance of our inverted Dirichlet mixture model and the widely used Gaussian mixture, and to investigate our hybrid generated kernels. In the first real application, we tackle the problem of object detection. The second real application involves the problem of visual scenes analysis and classification. For the hybrid approaches we use the SVM implementation provided in

the LIBSVM [4]. library and train one-versus-all classifiers. Moreover, following [33], the parameters A and ω were selected from $A \in \{2^{-10}, 2^{-9}, \ldots, 2^{4}\}$ and $\omega \in \{0.1, 0.2, \ldots, 0.8\}$, respectively, and the C-SVM formulation, $C \in \{2^{-2}, 2^{-1}, \ldots, 2^{12}\}$ was used. It is noteworthy that in the case of finite mixture models closed form expressions do not exist for the information divergence-based probabilistic kernels, thus we have used Monte carlo approximation with 15 000 generated points. Our practical experiences have indicated that these choices are appropriate.

4.1 Object Detection

In this section we address the problem of detecting objects in images which has several interesting applications such as automatic images semantic annotation and filtering. In particular, following [69], we consider the problems of detecting sky and vegetation in images using color and texture features. The main purpose of these experiments is to compare the performance of the widely used Gaussian mixture model with the performance of the inverted Dirichlet mixture learned using both ML+MML and Bayesian learning. It is noteworthy that a comparison with the numerous generative and discriminative approaches that have been proposed in the past is beyond the scope of this paper. The considered detection framework can be summarized as follows. First a given image is divided into 16×16 sub-blocks and then each sub-block is classified as containing the object (vegetation, sky) we are trying to detect or not (non-vegetation, non-sky). As in [69], each block is described by considering: 1) the color extracted by using 6 color moment features namely mean and variance in LUV color space, 2) the texture extracted by using 56 Gabor features namely mean and variance in 4 orientations and 7 scale features, and 3) the 2-dimensional block position described as its center coordinates. Thus, each sub-block is represented as a 64-dimensional vector.
In order to build a sky detector, we consider sky sub-blocks, taken as positive examples, extracted manually from 500 images collected from the web and non-sky blocks extracted from 1000 images collected also from the web. Figure 1 displays examples of images from which sky sub-blocks are extracted. As explained above these sub-block are described as 64-dimensional feature vectors used then to train an inverted Dirichlet mixture model representing sky and another mixture model representing non-sky blocks. Given these two mixture models and a test image, the well-known Bayes' rule can be used to assign the test image to the sky or non-sky classes. The vegetation detector is build in the same way using both vegetation sub-blocks, extracted from 500 images, and non-vegetation sub-blocks extracted from 1000 images. However, it is noteworthy that, as recommended in [69], we do not use the position features for vegetation detection (i.e. each sub-block is described by a 62-dimensional vector of only color and texture features) since generally there are not restrictions on where vegetation may occur in a given image. Figure 2 displays examples of images from which vegetation sub-blocks are extracted. Figure 3 shows the number of components

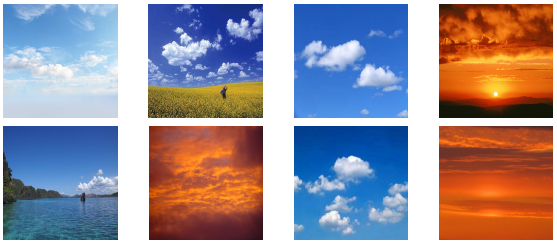

Fig. 1 Examples of images from which sky blocks are extracted.

Fig. 2 Examples of images from which vegetation blocks are extracted.

selected, using our inverted Dirichlet mixture learned in a Bayesian way (IDM-B) and when using EM and MML (IDM-EM), to model sky, non-sky, vegetation and non-vegetation sub-blocks.
Table 1 shows the classification accuracies when using IDM-B, IDM-EM, Bayesian Gaussian mixture (GM-B) and Gaussian mixture with EM and MML (GM-EM). According to this table, it is clear that the inverted Dirichlet mixture outperforms significantly, according to a student's t-test, the Gaussian mixture. We can notice also that Bayesian learning provides better results than ML+MML approach for both the inverted Dirichlet and Gaussian mixtures.

4.2 Visual Scenes Analysis and Classification

4.2.1 Problem Statement The proliferation of images requests their accurate organization and management to enable increased efficiency of retrieval and browsing to

[4] C-C. Chang and C-J. Lin, LIBSVM: A Library for Support Vector Machines. Available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

Table 1 Accuracies (%) for the sky and vegetation detection problems using different methods.

Method	IDM-B	IDM-EM	GM-B	GM-EM
Vegetation vs. Non-vegetation	95.61%	93.98%	90.32%	88.57%
Sky vs. Non-sky	96.16%	94.79%	92.94%	90.05%

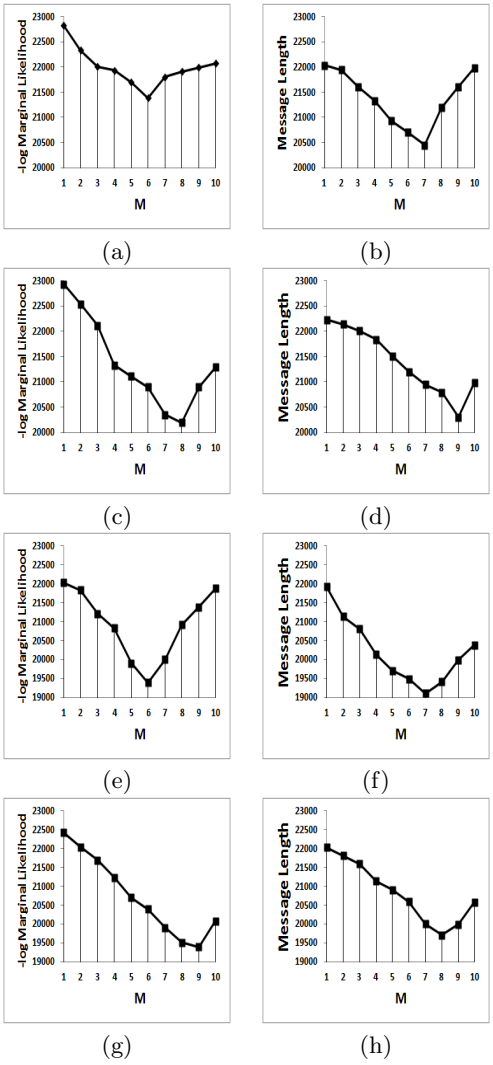

Fig. 3 Number of components selected using IDM-B (a,c,e,g) and IDM-EM (b,d,f,h) to represent sky (row 1), non-sky (row 2), vegetation (row 3) and non-vegetation (row 4) sub-blocks.

meet the needs of the end user. The problem of scenes classification entails the assignment of an unknown scene to one of several classes of scenes based on a set of visual features extracted from the scene. Scenes classification provide important contextual information and cues which can be used later for objects detection and recognition for instance [70] and in many other applications ranging from content-based image retrieval to tracking. Kernel-based methods [71,72] and in particular SVMs have been widely used for this problem. One crucial ingredient for image classification is the extraction of accurate features, that describes accurately a given scene [73], to represent the images. A very wide range of recent approaches have proposed the use of local patches (or image subregions) from which local features are extracted to represent images (see, for instance, [74–76]). The main motivation was the fact that local features provide a compact representation of objects that is robust to the large variation, because of illumination conditions and viewpoints, changes in scale, translation and affine deformations for instance [77], that can be seen between images belonging to the same class. In particular these local features have been used to construct visual vocabularies. A visual vocabulary can be viewed as an intermediate level representation for visual objects obtained through the quantization of a set of local features such as SIFT features which have been developed with an eye toward their use to provide accurate local presentation of objects [78]. Unfortunately, this approach causes the loss of important information about the image or object since it is based on the quantization of the features to obtain a fixed-length representation of the image or object [66]. A better approach is to consider all the set of features representing each scenes which can be handled using the hybrid kernels that we have developed in section 3 [31]. The main goal of this section is to investigate our generated kernels via the description of a given visual scene as a bag of vectors rather than as one vector of features.

4.2.2 Data Set and Results We have considered the publicly available ETH-80 data set [79] which is composed of 3280 images grouped into eight object classes with 10 unique objects and 41 views of each (see figure 4). Interest points in these images have been detected using the Harris detector and then represented using SIFT features [78]. Each image was represented then as a bag of orderless SIFT feature vectors extracted at detected interest points (these feature vectors describe the local image properties for a given particular small area in the image) that can be modeled as a finite inverted Dirichlet mixture model, using the algorithm in section 2, from which our kernels can be generated. As in [66] a one-versus-all SVM classifier is trained for each generated kernel by measuring the performance via cross-validation where all five views of an object are held out at one. The generated kernels that we have investigated and compared in our experiments were those generated from the inverted Dirichlet mixture namely Fisher kernel (FK-IDM), KL divergence kernel (KL-IDM), Rényi Kernel (R-IDM), Bhattacharyya kernel (B-IDM) and Jensen-

Fig. 8 Sample images from each group in the first data set. (a) Highway, (b) Inside of cities, (c) Tall buildings, (d) Streets, (e) Forest, (f) Coast, (g) Mountain, (h) Open country, (i) Suburb residence, (j) Bedroom, (k) Kitchen, (l) Livingroom, (m) Office..

Fig. 9 Additional categories in the second data set. (a)-(c) Store, (d)-(f) Industrial.

Fig. 4 Examples of images from each of the eight classes in the ETH-80 data set (apple, pear, tomato, cow, dog, horse, cup and car).

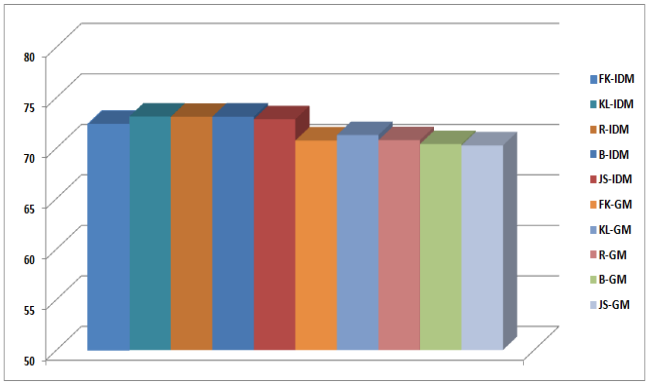

Fig. 5 Average classification performance (%) for the ETH-80 data set obtained using different generated kernels by considering inverted Dirichlet and Gaussian mixtures with different number of components obtained automatically using Bayesian learning.

Shannon (JS-IDM). In the same way probabilistic kernels were generated from Gaussian mixture (FK-GM, KL-GM,R-GM, B-GM and JS-GM) learned on the images data set. Figure 5 summarizes the obtained classification results when considering these different generated kernels. It is noteworthy that the results shown in this figure have been obtained by selecting automatically the number of components representing each image using the Bayes factor (i.e. the number of components M representing different images may be different). Classification results when we constrain the number of components to be the same for all the mixture models (we take M as the smallest number of components selected when modeling the different images) are shown in figure 6. We have also conducted another experiment where we assumed that each image may be represented by a single distribution (i.e. $M = 1$). Figure 7 summarizes the classification results in this case. In these three set of experiments (with different M, with same M, and with $M = 1$), the best results were obtained by considering the KL-IDM kernel (73.11%± 0.28, 72.94 % ± 0.34, and 70.71% ± 0.21, respectively). We can note that these results are close to the 73% accuracy obtained for instance in [66]. We can note also that the results when M is determined automatically and when it is fixed to be the same for all mixture models are comparable (i.e. the differences are not statistically significant) which shows that, when in-

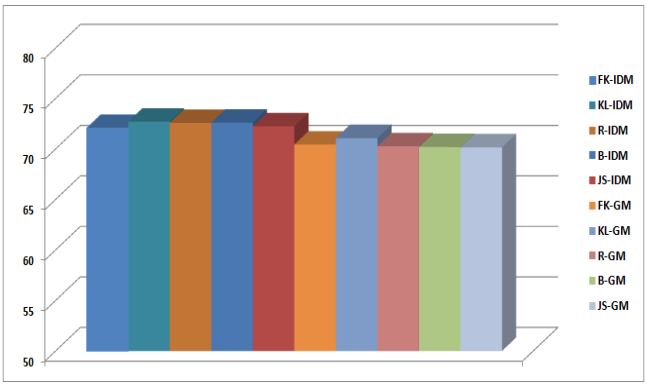

Fig. 6 Average classification performance (%) for the ETH-80 data set obtained using different generated kernels by assuming that the different mixture models representing the different images have the same number of components.

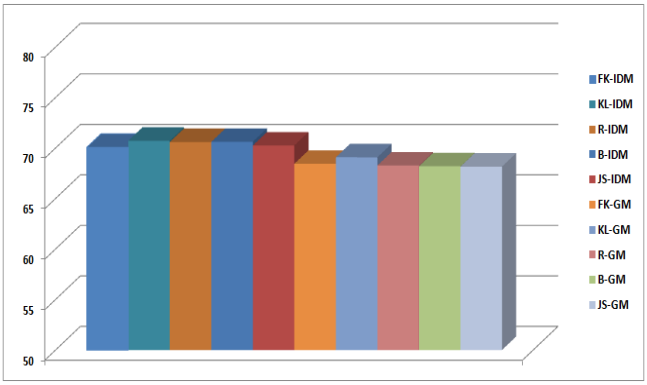

Fig. 7 Average classification performance (%) for the ETH-80 data set obtained using different generated kernels by assuming that each image may be represented by a single distribution (i.e. $M = 1$).

tegrated with SVM, the mixture models do not need to be complex to reach good classification results. However, reducing the mixture models to single distribution by assuming that $M = 1$ has caused a small deterioration of the classification accuracy.

We have considered two other well-known data sets, also. The first data set contains 13 categories of natural scenes [80] and consists of 13 categories: coasts (360 images), forest (328 images), mountain (374 images), open country (410 images), highway (260 images), inside of cities (308 images), tall buildings (356 images), streets (292 images), suburb residence (241 images), bedroom (174 images), kitchen (151 images), livingroom (289 images), and office (216 images). Figure 8 shows examples of images from this data set. The second data set contains 15 categories [81] and consists of the 13 categories of the second data set plus 626 other images divided into two categories (see figure 9): store (315 images) and industrial (311 images). We divided each of these data sets 10 times randomly into two separate halves, half for training and half for testing. Classification results for the 13 categories data set are summarized in figure 10. The best result was obtained using the KL-IDM kernel (76.03 % $\pm$ 0.29). The results in the case of the 15 categories data set are shown in figure 10 and the best performance was reached again using the KL-IDM kernel (71.19 % $\pm$ 0.33). The obtained classification results are better than the results obtained in the past using the well-known bag-of-visual words approach. This can be explained and interpreted by the fact that the constructed generative kernels respect local image structure in contrast with quantization which does not take into account the spatial information [82].

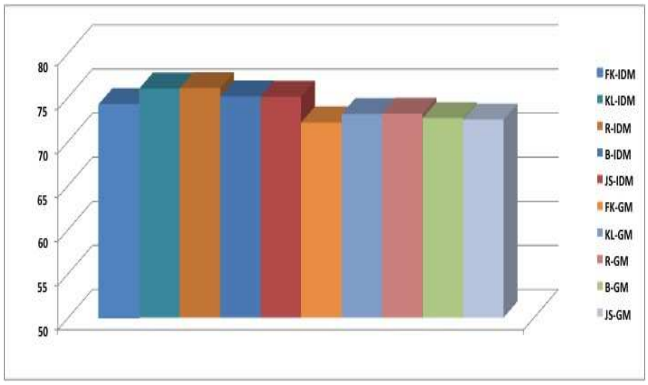

Fig. 10 Average classification performance (%) for the 13 categories data set obtained using different generated kernels by considering inverted Dirichlet and Gaussian mixtures with different number of components obtained automatically using Bayesian learning.

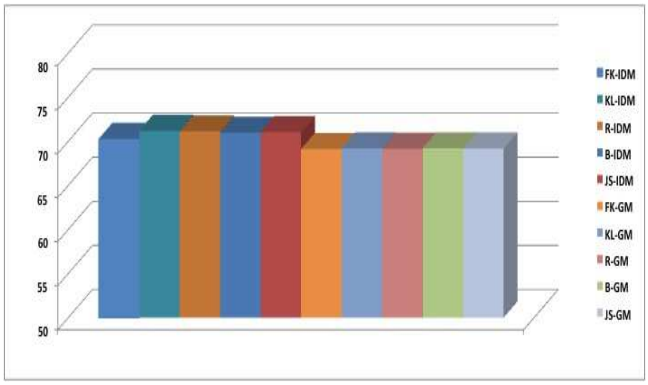

Fig. 11 Average classification performance (%) for the 15 categories data set obtained using different generated kernels by considering inverted Dirichlet and Gaussian mixtures with different number of components obtained automatically using Bayesian learning.

5 Conclusion

In this paper we have tackled the problems of modeling and clustering of positive data, defined in multidimensional space, into homogeneous groups. Unlike other approaches that have relied on the heavy assumption that the data are Gaussian, which is not true in may real-world applications, the data in this work are represented using finite mixtures of inverted Dirichlet distributions which provide plausible explanations. The consideration of finite mixture models is motivated by the central role they play in many data mining problems which has led to a variety of techniques for dealing with incomplete data. Several challenging problems have been investigated within the developed statistical framework. In particular, we have proposed a principled purely Bayesian algorithm for the learning of inverted Dirichlet mixtures. Our Bayesian model is based on representing the uncertainty in the inverted Dirichlet parameters via a formal prior that we derive by showing that the inverted Dirichlet belongs to the exponential family of distributions. The Bayesian machinery provides consistent inference where the plausibility of each model within a possible set of models, given the extracted data, is summarized by the joint posterior distribution of the model parameters and evaluated using Bayes factors. As exact inference in purely Bayesian approaches is not tractable to compute, we have used approximation MCMC methods namely Gibbs and Metropolis-Hastings sampling. Using the developed Bayesian learning framework, we have also considered elegant approaches for kernels generation that are theoretically and experimentally well-founded. Indeed, real data generated from challenging real life applications have been used for experimental purposes and the results, discussed in details, suggest that the proposed approaches are practical and unambiguously demonstrate the utility of the developed kernels. A number of possible avenues for future work suggest itself. Future works could be devoted, for instance, to the investigation of learning approaches that have tried to provide compromises between Bayesian and EM-based approaches such as [83]. Moreover, in many complicated problems, the slow convergence of MCMC approaches still posts the great challenge. To overcome this problem, variational techniques could be investigated. Another future work could be devoted to feature selection which is actually a fundamental problem in machine learning and pattern recognition. Indeed, in several situations some of the variables may be more relevant than others, some variables may be totally irrelevant for the task at hand and some variables may be just linear combinations of other variables. Finally, it is possible to investigate online learning within the developed framework.

6 Appendix

In this Appendix, we give more details about the calculations made. We do not spell out every calculation step explicitly, since the main goal is to present the most important issues.

6.1 Appendix 1: Proof of Eq. 5

if a S-parameter density p belongs to the exponential family, then we can write it as the following [84,48]

$$p(\mathbf{X}|\theta) = H(\mathbf{X}) \exp\big(G(\theta)^{tr} T(\mathbf{X}) + \Phi(\theta)\big) \tag{26}$$

where $G(\theta) = (G_1(\theta), \ldots, G_S(\theta)), T(\mathbf{X}) = (T_1(\mathbf{X}), \ldots, T_S(\mathbf{X}))$ and tr denotes the transpose. In this case a conjugate prior on θ is given by [48]

$$p(\theta) \propto \exp(\sum_{l=1}^{S} \rho_l G_l(\theta) + \kappa \Phi(\theta)) \tag{27}$$

where $\rho = (\rho_1, \ldots, \rho_S) \in \mathbb{R}^S$ and $\kappa > 0$ are referred as hyperparameters. The inverted Dirichlet distribution can be written as an exponential density. In fact, we can easily show that

$$p(\mathbf{X}_i|\alpha_j) = \frac{\Gamma(|\boldsymbol{\alpha}_j|)}{\prod_{d=1}^{D+1}\Gamma(\alpha_{jd})} \prod_{d=1}^{D} X_{id}^{\alpha_{jd}-1} (1+\sum_{d=1}^{D} X_{id})^{-|\boldsymbol{\alpha}_j|}$$

$$= \exp\Bigg[\log\big(\Gamma(|\boldsymbol{\alpha}_j|)\big) - \sum_{d=1}^{D+1} \log\Gamma(\alpha_{jd}) + \sum_{d=1}^{D} \alpha_{jd} \log X_{id}$$

$$- \sum_{d=1}^{D} \log X_{id} - |\boldsymbol{\alpha}_j| \log(1+\sum_{d=1}^{D} X_{id}) \Bigg]$$

Then by letting

$$S = D+1$$

$$\Phi(\boldsymbol{\alpha}_j) = \log\big(\Gamma(|\boldsymbol{\alpha}_j|)\big) - \sum_{d=1}^{D+1} \log\Gamma(\alpha_{jd})$$

$$G_d(\boldsymbol{\alpha}_j) = \alpha_{jd}, \quad d = 1, \ldots, D \quad G_{D+1}(\boldsymbol{\alpha}_j) = -|\boldsymbol{\alpha}_j|$$

$$T_d(\mathbf{X}) = \log X_{id}, \quad d = 1, \ldots, D \quad T_{D+1}(\mathbf{X}) = \log(1+\sum_{d=1}^{D} X_{id})$$

$$H(\mathbf{X}) = \exp\Big(-\sum_{d=1}^{D} \log X_{id} \Big)$$

The prior is

$$p(\boldsymbol{\alpha}_j) \propto \exp\Bigg[\sum_{d=1}^{D} \rho_d \alpha_{jd} - \rho_{D+1} |\boldsymbol{\alpha}_j|$$

$$+ \kappa \Big(\log\big(\Gamma(|\boldsymbol{\alpha}_j|)\big) - \sum_{d=1}^{D+1} \log\Gamma(\alpha_{jd}) \Big) \Bigg]$$

6.2 Appendix 2: Proof of Eq. 14

It is possible to compute the Bhattacharyya kernel in a closed form for densities that belong to the exponential family of distributions, i.e densities that can be written in the form given by Eq. 26. In this case the Bhattacharyya kernel is given by

$$\mathcal{K}_{\frac{1}{2}}(p(\mathbf{X}|\theta),p'(\mathbf{X}|\theta')) = \exp\left(\frac{1}{2}\Phi(\theta)+\frac{1}{2}\Phi(\theta')-\Phi(\frac{1}{2}\theta+\frac{1}{2}\theta')\right) \tag{28}$$

In the case of the inverted Dirichlet distribution, we have $\theta = \boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_{D+1})$ and $\Phi(\boldsymbol{\alpha}) = \log\big(\Gamma(|\boldsymbol{\alpha}|)\big) - \sum_{d=1}^{D+1}\log\Gamma(\alpha_d)$ Thus, according to Eq. 28

$$\begin{aligned}
&K_{\frac{1}{2}}(p(\mathbf{X}|\theta),p'(\mathbf{X}|\theta')) = \exp\Bigg[-\log\big(\Gamma(\frac{1}{2}|\boldsymbol{\alpha}|+\frac{1}{2}|\boldsymbol{\alpha}'|)\big)\\
&+\sum_{d=1}^{D+1}\log\Gamma(\frac{1}{2}\alpha_d+\frac{1}{2}\alpha'_d)+\frac{1}{2}\log\big(\Gamma(|\boldsymbol{\alpha}|)\big)-\frac{1}{2}\sum_{d=1}^{D+1}\log\Gamma(\alpha_d)\\
&+\frac{1}{2}\log\big(\Gamma(|\boldsymbol{\alpha}'|)\big)-\frac{1}{2}\sum_{d=1}^{D+1}\log\Gamma(\alpha'_d)\Bigg]\\
&=\frac{\prod_{d=1}^{D+1}\Gamma(\frac{1}{2}\alpha_d+\frac{1}{2}\alpha'_d)\sqrt{\Gamma(\sum_{d=1}^{D+1}\alpha_d)\Gamma(\sum_{d=1}^{D+1}\alpha'_d)}}{\Gamma\big(\sum_{d=1}^{D+1}(\frac{1}{2}\alpha_d+\frac{1}{2}\alpha'_d)\big)\sqrt{\prod_{d=1}^{D+1}\Gamma(\alpha_d)\Gamma(\alpha'_d)}}
\end{aligned}$$

6.3 Appendix 3: Proof of Eq. 18

The K-L divergence between two exponential distributions (See Eq. 26) is given by [85]

$$KL(p(\mathbf{X}|\theta),p'(\mathbf{X}|\theta')) = \Phi(\theta)-\Phi(\theta')+[G(\theta)-G(\theta')]^{tr}E_\theta[T(\mathbf{X})] \tag{29}$$

where E_θ is the expectation with respect to $p(\mathbf{X}|\theta)$. Moreover, we have the following [85]

$$E_\theta[T(\mathbf{X})] = -\Phi'(\theta) \tag{30}$$

Thus,

$$E_\theta\big[\log X_{id}\big] = -\frac{\partial\Phi(\theta)}{\partial\alpha_d} = \Psi(\alpha_d)-\Psi(|\boldsymbol{\alpha}|) \qquad d=1,\ldots,D \tag{31}$$

$$E_\theta\big[\log(1+\sum_{d=1}^{D}X_{id})\big] = -\frac{\partial\Phi(\theta)}{\partial-|\boldsymbol{\alpha}|} = \Psi(|\boldsymbol{\alpha}|) \tag{32}$$

By substituting the previous two equations into Eq. 29, we obtain

$$\begin{aligned}
&KL(p(\mathbf{X}|\theta),p'(\mathbf{X}|\theta')) = \log\big(\Gamma(|\boldsymbol{\alpha}_j|)\big)-\sum_{d=1}^{D+1}\log\Gamma(\alpha_d)\\
&-\log\big(\Gamma(|\boldsymbol{\alpha}'|)\big)+\sum_{d=1}^{D+1}\log\Gamma(\alpha'_d)\\
&+\sum_{d=1}^{D}(\alpha_d-\alpha'_d)(\Psi(\alpha_d)-\Psi(|\boldsymbol{\alpha}|))+(|\boldsymbol{\alpha}|-|\boldsymbol{\alpha}'|)\Psi(|\boldsymbol{\alpha}|)\\
&=\log\left[\frac{\Gamma(|\boldsymbol{\alpha}|)\prod_{d=1}^{D+1}\Gamma(\alpha'_d)}{\Gamma(|\boldsymbol{\alpha}'|)\prod_{d=1}^{D+1}\Gamma(\alpha_d)}\right]+\sum_{d=1}^{D+1}(\alpha_d-\alpha'_d)\Psi(\alpha_d)\\
&+(\alpha_{d+1}-\alpha'_{d+1})\Psi(|\boldsymbol{\alpha}|)
\end{aligned}$$

6.4 Appendix 4: Proof of Eq. 22

In the case of the inverted Dirichlet distribution, we can show that

$$\begin{aligned}
&\int_0^{+\infty}p(\mathbf{X}|\theta)^\sigma p'(\mathbf{X}|\theta')^{1-\sigma}d\mathbf{X}\\
&=\left[\frac{\Gamma(|\boldsymbol{\alpha}|)}{\prod_{d=1}^{D+1}\Gamma(\alpha_d)}\right]^\sigma\left[\frac{\Gamma(|\boldsymbol{\alpha}'|)}{\prod_{d=1}^{D+1}\Gamma(\alpha'_d)}\right]^{1-\sigma}\\
&\times\int_0^{+\infty}\left[\prod_{d=1}^{D}X_d^{\alpha_d-1}(1+\sum_{d=1}^{D}X_{id})^{-|\boldsymbol{\alpha}|}\right]^\sigma d\mathbf{X}\\
&\times\int_0^{+\infty}\left[\prod_{d=1}^{D}X_d^{\alpha'_d-1}(1+\sum_{d=1}^{D}X_{id})^{-|\boldsymbol{\alpha}'|}\right]^{1-\sigma}d\mathbf{X}\\
&=\left[\frac{\Gamma(|\boldsymbol{\alpha}|)}{\prod_{d=1}^{D+1}\Gamma(\alpha_d)}\right]^\sigma\left[\frac{\Gamma(|\boldsymbol{\alpha}'|)}{\prod_{d=1}^{D+1}\Gamma(\alpha'_d)}\right]^{1-\sigma}\\
&\times\int_0^{+\infty}\left[\prod_{d=1}^{D}X_d^{\alpha'_d-\sigma\alpha'_d+\sigma\alpha_d-1}(1+\sum_{d=1}^{D}X_{id})^{-\sigma|\boldsymbol{\alpha}|+\sigma|\boldsymbol{\alpha}'|-|\boldsymbol{\alpha}'|}\right]d\mathbf{X}\\
&=\left[\frac{\Gamma(|\boldsymbol{\alpha}|)}{\prod_{d=1}^{D+1}\Gamma(\alpha_d)}\right]^\sigma\left[\frac{\Gamma(|\boldsymbol{\alpha}'|)}{\prod_{d=1}^{D+1}\Gamma(\alpha'_d)}\right]^{1-\sigma}\\
&\times\left[\frac{\Gamma(\prod_{d=1}^{D+1}\Gamma(\alpha'_d-\sigma\alpha'_d+\sigma\alpha_d))}{\sum_{d=1}^{D+1}(\alpha'_d-\sigma\alpha'_d+\sigma\alpha_d)}\right]
\end{aligned}$$

6.5 Appendix 4: Proof of Eq. 25

$$H[p(\mathbf{X}|\theta)] = -\int_0^{+\infty} p(\mathbf{X}|\theta)\log p(\mathbf{X}|\theta)d\mathbf{X} \quad (33)$$

$$= -\int_0^{+\infty} p(\mathbf{X}|\theta)\Bigg[\log\Gamma(|\boldsymbol{\alpha}_j|) - \sum_{d=1}^{D+1}\log\Gamma(\alpha_{jd}) + \sum_{d=1}^{D}(\alpha_{jd}-1)\log X_d - |\boldsymbol{\alpha}_j|\log(1+\sum_{d=1}^{D}X_d)\Bigg]d\mathbf{X}$$

$$= -\Bigg[\log\Gamma(|\boldsymbol{\alpha}_j|) - \sum_{d=1}^{D+1}\log\Gamma(\alpha_{jd}) + \sum_{d=1}^{D}(\alpha_{jd}-1)E_\theta[\log X_d] - |\boldsymbol{\alpha}_j|E_\theta[\log(1+\sum_{d=1}^{D}X_d)]\Bigg]$$

By substituting Eqs. 31 and 32 into the previous equation, we obtain the following

$$H[p(\mathbf{X}|\theta)] = -\log\Gamma(|\boldsymbol{\alpha}_j|) + \sum_{d=1}^{D+1}\log\Gamma(\alpha_{jd}) - \sum_{d=1}^{D}(\alpha_{jd}-1)(\Psi(\alpha_d)-\Psi(|\boldsymbol{\alpha}|)) + |\boldsymbol{\alpha}_j|\Psi(|\boldsymbol{\alpha}|) \quad (34)$$

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 the anonymous referees and the associate editor for their helpful comments.

References

1. T. K. Ho and H. S. Baird. Large-Scale Simulation Studies in Image Pattern Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1067–1079, 1997.
2. L. Saitta and F. Neri. Learning in the "Real World". *Machine Learning*, 30:133–163, 1998.
3. P. Domingos. MetaCost: A General Method for Making Classifiers Cost-Sensitive. In *Proc. of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 155–164, 1999.
4. S. Y. Sohn. Meta Analysis of Classification Algorithms for Pattern Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1137–1144, 1999.
5. V. Cherkassky and F. Mulier. *Learning from Data: Concepts, Theory, and Methods*. Wiley, second edition, 2007.
6. G.J. McLachlan and D. Peel. *Finite Mixture Models*. New York: Wiley, 2000.
7. C. Fraley and A. E. Raftery. MCLUST: Software for Model-Based Cluster Analysis. *Journal of Classification*, 16:297–306, 1999.
8. D. Ghosh and A. M. Chinnaiyan. Mixture Modelling of a Gene Expression Data from Microarray Experiments. *Bioinformatics*, 18(2):275–286, 2002.
9. D. Keysers, F. J. Och and H. Ney. Maximum Entropy and Gaussian Models for Image Object Recognition. In *Proc. of the DAGM-Symposium*, pages 498–506, 2002.
10. Z. Lu, Y. Peng and H. H. S. Ip. Gaussian Mixture Learning via Robust Competetive Agglomeration. *Pattern Recognition Letters*, 31(7):539–547, 2010.
11. N. Bouguila, D. Ziou and J. Vaillancourt. Unsupervised Learning of a Finite Mixture Model Based on the Dirichlet Distribution and its Application. *IEEE Transactions on Image Processing*, 13(11):1533–1543, 2004.
12. N. Bouguila and D. Ziou. A Hybrid SEM Algorithm for High-Dimensional Unsupervised Learning Using a Finite Generalized Dirichlet Mixture. *IEEE Transactions on Image Processing*, 15(9):2657– 2668, 2006.
13. N. Bouguila, D. Ziou and R. I. Hammoud. On Bayesian Analysis of a Finite Generalized Dirichlet Mixture Via a Metropolis-within-Gibbs Sampling. *Pattern Analysis and Applications*, 12(2):151–166, 2009.
14. T. Bdiri and N. Bouguila. Learning Inverted Dirichlet Mixtures for Positive Data Clustering. In *Proc. of 3th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC)*, volume 6743 of *Lecture Notes in Computer Science*, pages 265–272. Springer, 2011.
15. T. Bdiri and N. Bouguila. An infinite mixture of inverted dirichlet distributions. In Bao-Liang Lu, Liqing Zhang, and James T. Kwok, editors, *ICONIP (2)*, volume 7063 of *Lecture Notes in Computer Science*, pages 71–78. Springer, 2011.
16. J. G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, 1997.
17. T. Bdiri and N. Bouguila. Positive Vectors Clustering Using Inverted Dirichlet Finite Mixture Models. *Expert Systems With Applications*, 39(2):1869–1882, 2012.
18. A. Racine, A. P. Grieve, H. Fluhler and A. F. M. Smith. Bayesian Methods in Practice: Experiences in the Pharmaceutical Industry (with discussion). *Applied Statistics*, 35(2):93–150, 1986.
19. D. Barber and C. K. I. Williams. Gaussian Processes for Bayesian Classification via Hybrid Monte Carlo. In *Advances in Neural Information Processing Systems (NIPS)*, pages 340–346, 1996.
20. D. K. Agarwal and A. E. Gelfand. Slice Sampling for Simulation Based Fitting of Spatial Data Models. *Statistics and Computing*, 15:61–69, 2005.
21. X-L. Meng and S. Schilling. Fitting Full-Information Item Factor Models and an Empirical Investigation of Bridge Sampling. *Journal of the American Statistical Association*, 91(435):1254–1267, 1996.
22. S. Dudoit and J. Fridlyand. A Prediction-Based Resampling Method for Estimating the Number of Clusters in a Dataset. *Genome Biology*, 3(7):1–21, 2002.
23. R.E. Kass and A.E. Raftery. Bayes Factors. *Journal of the American Statistical Association*, 90:773–795, 1995.
24. D. Miller, A. V. Rao, K. Rose and A. Gersho. A Global Optimization Technique for Statistical Classifier Design. *IEEE Transactions on Signal Processing*, 44(12):3108–3122, 1996.
25. J. Liu, J-Q. Song and Y-L. Huang. A Generative/Discriminative Hybrid Model: Bayes Perceptron

Classifier. In *Proc. of International Conference on Machine Learning and Cybernetics (ICLMC)*, pages 2767–2772, 2007.
26. B. schölkopf, P. Bartlett, A. Smola and R. Williamson. Shrinking the Tube: A New Support Vector Regression Algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, pages 330–336, 1998.
27. T. Joachims. Estimating the Generalization Performance of an SVM Efficiently. In *Proc. of the International Conference on Machine Learning (ICML)*, pages 431–438, 2000.
28. N. Cristianini, C. Campbell and J. Shawe-Taylor. Dynamically Adapting Kernels in Support Vector Machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 204–210, 1998.
29. S. Fine and K. Scheinberg. Efficient SVM Training Using Low-Rank Kernel Representations. *Journal of Machine Learning Research*, 2:243–264, 2001.
30. T. S. Jaakkola and D. Haussler. Exploiting Generative Models in Discriminative Classifiers. In *Proc. of Advances in Neural Information Systems (NIPS)*, pages 487–493. MIT Press, 1998.
31. N. Bouguila. Bayesian Hybrid Generative Discriminative Learning Based on Finite Liouville Mixture Model. *Pattern Recognition*, 44(6):1183–1200, 2011.
32. P. J. Moreno, P. P. Ho and N. Vasconcelos. A Kullback-Leibler Divergence Based Kernel for SVM Classification in Multimedia Applications. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2003.
33. A. B. Chan, N. Vasconcelos and P. J. Moreno. A Family of Probabilistic Kernels Based on Information Divergence. Technical Report SVCL-TR 2004/01, University of California, San Diego, 2004.
34. T. Jebara and R. Kondor. Bhattacharyya Expected Likelihood Kernels. In *Proc. of the Annual Conference on Computational Learning Theory (COLT)*, pages 57–71, 2003.
35. J. T-Y. Kwok. Moderating the Outputs of Support Vector Machine Classifiers. *IEEE Transactions on Neural Networks*, 10(5):1018–1031, 1999.
36. P. Sollich. Probabilistic Interpretations and Bayesian Methods for Support Vector Machines. In *Proc. of the International Conference on Artificial Neural Networks (ICANN)*, pages 91–96, 1999.
37. P. Sollich. Bayesian Methods for Support Vector Machines: Evidence and Predictive Class Probabilities. *Machine Learning*, 46:21–52, 2002.
38. E. A. Thompson. Likelihood and Linkage: From Fisher to the Future. *Annals of Statistics*, 24(2):449–465, 1996.
39. L. Stewart. Hierarchical Bayesian Analysis Using Monte Carlo Integration: Computing Posterior Distributions When there are Many Possible Models. *The Statistician*, 36(2/3):211–219, 1987.
40. D. A. Binder. Approximations to Bayesian Clustering Rules. *Biometrika*, 68(1):275–285, 1981.
41. R. L. Winkler. The Quantification of Judgment: Some Methodological Suggestions. *Journal of the American Statistical Association*, 62(320):1105–1129, 1967.
42. C. S. S. von Holstein. Two Techniques for Assessment of Subjective Probability Distributions - An Experimental Study. *Acta Psychologica*, 35(6):478–494, 1971.
43. D. Haughton. On the Choice of a Model to Fit Data from an Exponential Family. *Annals of Statistics*, 16(1):342–355, 1988.
44. D. Haughton. Size of the Error in the Choice of a Model to Fit Data from an Exponential Family. *Sankhya: The Indian Journal of Statistics, Series A*, 51(1):45–58, 1989.
45. A. Frigessi, J. Gåsemyr and H. Rue. Antithetic Coupling of Two Gibbs Sampler Chains. *Annals of Statistics*, 28(4):1128–1149, 2000.
46. G. Casella. Mixture Models, Latent Variables and Partitioned Importance Sampling. *Statistical Methodology*, 1(1-2):1–18, 2004.
47. G. O. Roberts and J. S. Rosenthal. Optimal Scaling of Discrete Approximations to Langevin Diffusions. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 60(1):255–268, 1998.
48. C.P. Robert. *The Bayesian Choice*. Springer-Verlag, 2001.
49. J. S. Rosenthal. Rates of Convergence for Data Augmentation on Finite Sample Spaces. *The Annals of Applied Probability*, 3(3):819–839, 1993.
50. G. O. Roberts and N. G. Polson. On the Geometric Convergence of the Gibbs Sampler. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 56(2):377–384, 1994.
51. C. J. Geyer and E. A. Thompson. Annealing Markov Chain Monte Carlo with Applications to Ancestral Inference. *Journal of the American Statistical Association*, 90(431):909–920, 1995.
52. J. S. Rosenthal. Rates of Convergence for Gibbs Sampling for Variance Component Models. *The Annals of Statistics*, 23(3):740–761, 1995.
53. G. O. Roberts and R. L. Tweedie. Bounds on Regeneration Times and Convergence Rates for Markov Chains. *Stochastic Processes and Their Applications*, 80:211–229, 1999.
54. A. E. Raftery and S. M. Lewis. One Long Run with Diagnostics: Implementation Startegies for Markov Chain Monte Carlo. *Statistical Science*, 7(4):493–497, 1992.
55. T. K. Ho and E. M. Kleinberg. Building Projectable Classifiers of Arbitrary Complexity. In *Proc. of the International Conference on Pattern Recognition (ICPR)*, pages 880–885, 1996.
56. E. M. Kleinberg. An Overtraining-Resistant Stochastic Modeling Method for Pattern Recognition. *Annals of Statistics*, 24(6):2319–2349, 1996.
57. B. G. Leroux. Consistent Estimation of a Mixing Distribution. *Annals of Statistics*, 20(3):1350–1360, 1992.
58. A. E. Raftery. Approximate Bayes Factors and Accounting for Model Uncertainity in Generalized Linear Models. *Biometrika*, 83(2):251–266, 1996.
59. X-L. Meng and W. H. Wong. Simulating Ratios of Normalizing Constants via a Simple Idendity: A Theorectical Exploration. *Statistica Sinica*, 6:831–860, 1996.
60. M-H. Chen and Q-M. Shao. Estimating Ratios of Normalizing Constants for Densities with Different Dimensions. *Statistica Sinica*, 7:607–630, 1997.
61. J. Rissanen. Modeling by Shortest Data Description. *Automatica*, 14:465–471, 1978.
62. K. P. Bennett and C. Campbell. Support Vector Machines: Hype or Hallelujah? *SIGKDD Explorations*, 2(2):1–13, 2000.

63. T. Van Gestel, J. A. K. Suykens, J. De Brabanter, B. De Moor and J. Vandewalle. Kernel Canonical Correlation Analysis and Least Squares Support Vector Machines. In *Proc. of International Conference on Artificial Neural Networks (ICANN)*, pages 384–389, 2001.
64. N. Cristianini, J. Shawe-Taylor, A. Elisseeff and J. Kandola. On Kernel-Target Alignment. In *Advances in Neural Information Processing Systems (NIPS)*, pages 367–373, 2001.
65. B. Hammer and B. J. Jain. Neural Methods for Non-Standard Data. In *Proc. of the European Symposium on Artificial Neural Networks (ESANN)*, pages 281–292, 2004.
66. K. Grauman and T. Darrell. The Pyramid Match Kernel: Efficient Learning with Sets of Features. *Journal of Machine Learning Research*, 8:725–760, 2007.
67. A. Rènyi. On Measures of Entropy and Information. In *Proc. of Berkeley Symposium on Mathematical Statistics and Probability*, pages 547–561, 1960.
68. J. Lin. Divergence Measure Based on Shannon Entropy. *IEEE Transactions on Information Theory*, 37(14):145–151, 1991.
69. A. Vailaya and A. Jain. Detecting sky and vegetation in outdoor images. In *Proc. of the International Society for Optical Engineering (SPIE)*, pages 411–420, 2000.
70. A. Torralba. Contextual Priming for Object Detection. *International Journal of Computer Vision*, 53(2):169–191, 2003.
71. Y-Y. Lin, T-L. Liu and C-S. Fuh. Local Ensemble Kernel Learning for Object Category Recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.
72. A. Vedaldi and S. Soatto. Quick Shift and Kernel Methods for Mode Seeking. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 705–718, 2008.
73. L. Itti, C. Koch and E. Neibur. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.
74. R. Fergus, P. Perona and A. Zisserman. A Sparse Object Category Model for Efficient Learning and Exhaustive Recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 380–387, 2005.
75. A. Bosch, A. Zisserman and X. Muñoz. Representing Shape with a Spatial Pyramid Kernel. In *Proc. of the 6th ACM international conference on Image and video retrieval (CIVR)*, pages 401–408. ACM, 2007.
76. B. Fulkerson, A. Vedaldi and S. Soatto. Localizing Objects with Smart Dictionaries. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 179–192, 2008.
77. A. Vedaldi and S. Soatto. Features for Recognition: Viewpoint Invariance for Non-Planar Scenes. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pages 1474–1481, 2005.
78. D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
79. B. Leibe and B. Schiele. Analyzing Appearance and Contour Based Methods for Object Categorization. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 409–415, 2003.
80. L. Fei-Fei and P. Perona. A Bayesian Hierarchical Model for Learning Natural Scene Categories. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 524–531, 2005.
81. S. Lazebnik, C. Schmid and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2169–2178, 2006.
82. F. Perronnin and C. Dance. Fisher Kernels on Visual Vocabularies for Image Categorization. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.
83. N. Friedman. The Bayesian Structural EM Algorithm. In *Proc. of the 4th Conference on Uncertainity in Artificial Intelligence (UAI)*, pages 129–138, 1998.
84. S. R. Dalal and W. J. Hall. Approximating Priors by Mixtures of Natural Conjugate Priors. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 45(2):278–286, 1983.
85. L. D. Brown. *Fundamentals of Statistical Exponential Families with Applications in Statistical Decision Theory.* Hayward, CA: Institute of Mathematical Statistics, 1986.