

Image Modeling and Segmentation Using Incremental Bayesian Mixture Models

Constantinos Constantinopoulos and Aristidis Likas*

Department of Computer Science, University of Ioannina, GR 45110 Ioannina, Greece
ccostas@cs.uoi.gr, arly@cs.uoi.gr

Abstract. Many image modeling and segmentation problems have been tackled using Gaussian Mixture Models (GMM). The two most important issues in image modeling using GMMs is the selection of the appropriate low level features and the specification of the appropriate number of GMM components. In this work we deal with the second issue and present an approach for GMM-based image modeling employing an incremental variational algorithm for Bayesian GMM training that automatically specifies the number of mixture components. Experimental results on natural and texture images indicate that the method yields reasonable models without requiring the a priori specification of the number of components.

1 Introduction

Several approaches have been proposed for statistical image representation, ie. the modeling of an image based on the distribution of various features at pixel or window level. Simpler approaches focus on the use of histograms, while more sophisticated statistical modeling tools such as mixture models have also been considered [1]. More specifically, the assumption under the GMM framework is that an image is considered as a set of regions (segments) where each region is represented by a Gaussian distribution and the set of all regions in an image is represented by a GMM.

To build the GMM for an image, first a dataset $X = \{f_n\}$ is constructed that contains one feature vector $f_n \in R^d$ either for each image pixel n (as is the case in this work) or for appropriately selected image windows (patches). Typical low level features used are related to color and texture. Then an efficient training method is applied to the dataset X in order to obtain the final GMM for the image.

Let g be a mixture with M Gaussian components

$$g(f) = \sum_{j=1}^M \pi_j \mathcal{N}(f; \mu_j, T_j) \quad (1)$$

where $\pi = \{\pi_j\}$ are the mixing coefficients (priors), $\mu = \{\mu_j\}$ the means (centers) of the components, and $T = \{T_j\}$ the precision (inverse covariance) matrices.

* This work was partially supported by Interreg IIIA (Greece-Italy) grant I2101005.

After training, it is possible to segment the image, ie. assign a group of pixels to each GMM component by finding for the feature vector f of each pixel the component j with maximum posterior, (ie. with maximum $\pi_j \mathcal{N}(f; \mu_j, T_j)$).

An issue to be considered is how to impose the requirement for spatial smoothness, ie. that in most cases neighboring pixels should be assigned to the same component (cluster). Two approaches have been proposed. The first imposes an MRF prior on the posteriors [2]. The MRF approach is more difficult to handle from the learning point of view, and requires all image pixels to be included in the training set. It provides as outcome the posterior probabilities for each pixel. In addition, no effective method has been proposed for automatically determining the number of GMM components in the MRF framework.

In this work we focus on the second way to impose spatial smoothness [1]. According to this approach, the spatial location (x, y) of a pixel is considered as an additional feature to be included in the feature vector f describing this pixel, for example $f_n = (L, a, b, x, y)_n$ for the n -th pixel with image coordinates (x, y) and color vector (L, a, b) . Thus the spatial distance between two pixels contributes significantly to the total distance between the corresponding feature vectors. In this way, adjacent pixels tend to have similar cluster labels since their spatial distance is small, thus spatial smoothing is achieved. On the other hand, this approach forces large image areas (with approximately the same color) that can be considered as one segment to be splitted into smaller segments, because spatially distant pixels cause the distance of the corresponding feature vectors to be large, thus they cannot be modeled by a single Gaussian component. However, this fragmentation problem can be easily resolved through a simple postprocessing stage that merges image regions corresponding to GMM components with similar mean feature values. In addition, to further enforce spatial smoothness, it is possible to apply the MRF-based GMM model afterwards starting from the GMM solution provided by the second approach.

The advantage of using the second approach for spatial smoothness is that it results in a GMM that takes as input the image location (x, y) and can assign feature labels (for example color) to every location (x, y) independent of whether this pixel has been used for training or not. This has three interesting consequences. First it is easy to obtain a model for a specific region of the image by considering only the mixture components that are active in this region. For example we can easily derive a mixture model for the color density in an arbitrary image region. Second it is straightforward to marginalize the (x, y) coordinates and obtain a GMM for the distribution of the low level features. Finally, it is easy to assign segment labels to image locations (x, y) not used for training, simply by computing the component j with highest posterior. This allows train the mixture model using only a subset of the pixels.

Despite those advantages, a major difficulty is introduced that relates to the specification of the number of mixture components. For example, it is possible that an image with four colors cannot be modeled using a GMM with four components, especially in the case where the image segments with the same color have large size or are disconnected. In this case the GMM requires more

components to accurately model the image. Therefore *it is difficult to specify in advance the number of GMM components*, thus it is essential to use GMM training methods that incorporate a built-in mechanism for automatically assessing the number of components. This is the case with our method [3] used in this work that is described next.

2 The Incremental Variational Bayesian Method

In this section we describe a Bayesian method for Gaussian mixture learning [3] that is deterministic, does not depend on the initialization, and resolves adequately the model selection problem, ie. the specification of the number of components. The method is an incremental one: it starts with one component and progressively adds components to the model. The procedure for component addition is based on a *splitting test* applied to each of the existing mixture components. According to this test, *a component is splitted into two sub-components* and then variational Bayesian learning is applied to the specific pair of components, while the rest components remain “fixed”. To apply this method, it is necessary to define a Bayesian Gaussian mixture model by imposing priors on the parameters π , μ_j and T_j .

The effect of variational Bayesian learning is that a *competition* takes place between the two sub-components. If the data distribution in the region of the splitted component strongly suggests the existence of more than one clusters, then both sub-components will “survive” and the number of model components will be increased. Otherwise, the competition among the two components will cause one of them to be eliminated and the initial component will be recovered. This strategy of incremental component addition also facilitates the specification of the parameters of the priors, since it can be based on the parameters of the component to be splitted.

In order to apply this idea, a modification of the typical Bayesian mixture model [4] is required that is described in the graphical model of Figure 1 and is explained next. Assume that we wish to restrict the competition to a subset containing s of the GMM components while the others $M - s$ components are “fixed”. The proposed modification is to impose a prior only on the $M - s$ “fixed” mixing coefficients $\tilde{\pi}$. Let $X = \{f_n\}$ the set of training points containing the feature vectors of the image pixels. The hidden variables $Z = \{z_{jn}\}$ capture the missing information of which component has generated a given data point f_n . More specifically, $z_{jn} = 1$ if component j is responsible for generating x_n , otherwise $z_{jn} = 0$. Therefore it holds that:

$$p(X|Z, \mu, T) = \prod_{n=1}^N \prod_{j=1}^M [\mathcal{N}(f_n; \mu_j, T_j)]^{z_{jn}} \quad (2)$$

The distribution of Z is a product of multinomials

$$p(Z|\pi, \tilde{\pi}) = \prod_{n=1}^N \prod_{j=1}^s \pi_j^{z_{jn}} \prod_{j=s+1}^M \tilde{\pi}_j^{z_{jn}} \quad (3)$$

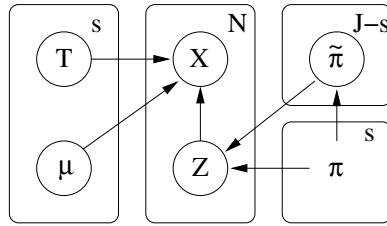


Fig. 1. The graphical model

given the subset $\tilde{\pi} = \{\tilde{\pi}_j\}$ of “fixed” mixing coefficients and the subset $\pi = \{\pi_j\}$ of “free” mixing coefficients. For notational convenience and assuming M mixing components, we can always rearrange the indexes so that the first s components are the “free” ones.

The typical Bayesian framework assumes conjugate Dirichlet priors over the entire set of mixing coefficients. Therefore, in order to define the modified Bayesian GMM, it is necessary to define the conditional joint distribution $p(\tilde{\pi}|\pi)$ of the “fixed” mixing coefficients given the “free” which can be shown to be a non-standard Dirichlet with parameters α_j ($j = s + 1, \dots, M$):

$$p(\tilde{\pi}|\pi) = \left(1 - \sum_{j=1}^s \pi_j\right)^{-M+s} \frac{\Gamma(\sum_{j=s+1}^M \alpha_j)}{\prod_{j=s+1}^M \Gamma(\alpha_j)} \times \prod_{j=s+1}^M \left(\frac{\tilde{\pi}_j}{1 - \sum_{k=1}^s \pi_k}\right)^{\alpha_j - 1} \quad (4)$$

It must be emphasized that we do not impose prior on the “free” mixing coefficients. Completing the specification of the Bayesian model we assume Gaussian and Wishart priors for μ and T respectively [4]:

$$p(\mu) = \prod_{j=1}^s \mathcal{N}(\mu_j | 0, \beta \mathcal{I}) \quad (5)$$

$$p(T) = \prod_{j=1}^s \mathcal{W}(T_j | \nu, V). \quad (6)$$

Learning in the Bayesian framework can be achieved through maximization of the marginal likelihood of X given π , that is obtained by integrating out the *hidden variables* $\theta = \{Z, \mu, T, \tilde{\pi}\}$ as follows:

$$p(X|\pi) = \sum_Z \int p(X, \theta|\pi) d\mu dT d\tilde{\pi}. \quad (7)$$

Since this integral is intractable, the *Variational Bayes* methodology is adopted, where we maximize a *lower bound* \mathcal{L} of the logarithmic marginal likelihood $\log p(X|\pi)$:

$$\mathcal{L}[q, \pi] = \sum_Z \int q(\theta) \log \frac{p(X, \theta | \pi)}{q(\theta)} d\theta \tag{8}$$

where q is an arbitrary distribution that approximates the posterior distribution $p(\theta | X, \pi)$. The maximization of \mathcal{L} is performed in an iterative way, where at each iteration two steps take place (in analogy to the EM approach): first maximization of the bound with respect to q , and subsequently maximization of the bound with respect to π . To implement this maximization with respect to q the mean-field approximation [4] has been adopted, which assumes that q is constrained to be a product of the form: $q(\theta) = q_Z(Z)q_\mu(\mu)q_T(T)q_{\tilde{\pi}}(\tilde{\pi})$. The resulting update equations for the parameters of the q distributions (E-step) and the parameters π (M-step) are omitted and described in detail in [3].

Using the above idea, the incremental algorithm for constructing the GMM proceeds as follows. Mixture components are sequentially added to the mixture model using the following component splitting procedure: one of the mixture components is selected and is appropriately splitted in two components. The resulting two components are considered as “free” and the rest as “fixed” according to the terminology introduced previously. Next we set the precision prior $p(T)$ based on the characteristics of the splitted component, and apply variational learning as described in the previous section. In case that the two components provide a much better fit to the data in their region, then both components are

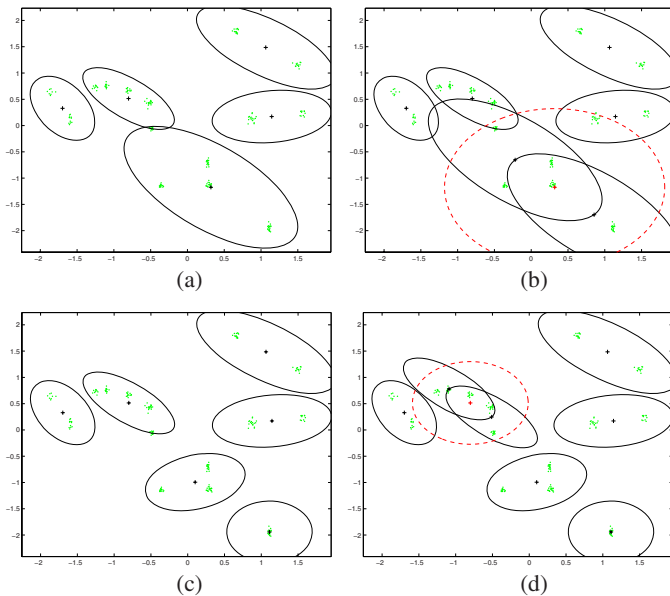


Fig. 2. Four steps of the incremental training procedure. The expected covariance w.r.t. the Wishart prior is depicted with a dashed line. (a) An intermediate solution with 5 components. (b) One component is splitted in two. (c) The mixture after variational learning. (d) Another component is selected and splitted.

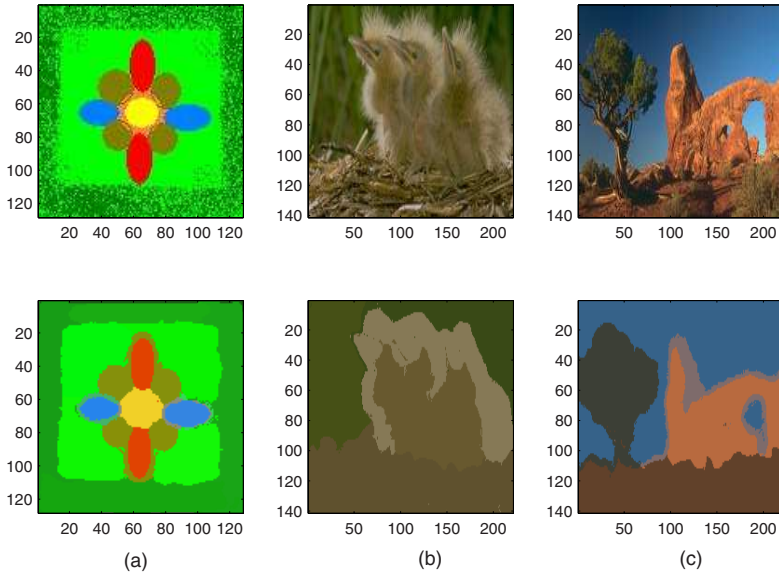


Fig. 3. Segmentation of (a) an artificial image, (b) and (c) natural images from BSBS. Top row: original images. Bottom row: segmented images.

retained in the mixture model, otherwise the update equations will eliminate one of them. The splitting test is applied sequentially to all components and the method terminates when all mixture components have been unsuccessfully tested for splitting. In the case where a successful split is encountered, then the number of mixture components increases and a new round of split tests for all components is initialized.

To illustrate the details of splitting, assume that some component \hat{j} has to be splitted, with density $\mathcal{N}(f; \mu_{\hat{j}}, T_{\hat{j}})$. The idea is that in order to form the new mixture, we remove component \hat{j} and insert two new components with densities $\mathcal{N}(f; \mu_{\hat{j}1}, T_{\hat{j}1})$ and $\mathcal{N}(f; \mu_{\hat{j}2}, T_{\hat{j}2})$ respectively. We have selected to place the centers of the two components along the dimension of the principal axis of the covariance $T_{\hat{j}}^{-1}$ and at opposite directions with respect to the center $\mu_{\hat{j}}$. The mixing coefficients of the two components are set equal $\pi_{\hat{j}1} = \pi_{\hat{j}2} = \pi_{\hat{j}}/2$, and their parameters are set according to: $\mu_{\hat{j}1} = \mu_{\hat{j}} + \sqrt{\lambda} u$, $\mu_{\hat{j}2} = \mu_{\hat{j}} - \sqrt{\lambda} u$, $T_{\hat{j}1} = T_{\hat{j}}$ and $T_{\hat{j}2} = T_{\hat{j}}$, where λ is the maximum eigenvalue of $T_{\hat{j}}^{-1}$ and u the corresponding eigenvector. An important issue in the proposed method is the specification of the scale parameter V of the prior $\mathcal{W}(\nu, V)$ over the precision matrices, based on the splitted component. We set $\nu = d$ (which is the minimum allowed value) and $V = \nu \lambda \mathcal{I}$, where λ is the highest eigenvalue of $T_{\hat{j}}^{-1}$. The value of β was set to 10^{-10} . An example of component splitting is illustrated in Figure 2.

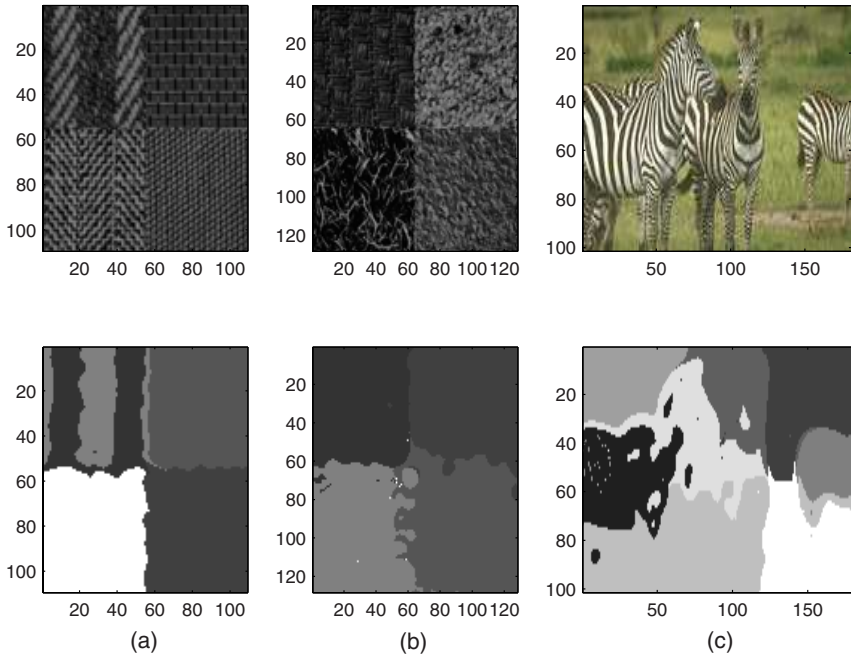


Fig. 4. Segmentation using texture feature: (a) and (b) artificial images, (c) BSDS image. Top row: original images. Bottom row: segmented images.

3 Experimental Results

To illustrate the performance of the proposed method, we have conducted experiments using both artificially generated and natural images. For each image the following steps were taken. First the dataset containing the feature vectors for a subset of 5000 pixels was constructed and next the feature vectors were preprocessed so that each feature distribution has zero mean and unit standard deviation. The resulting dataset was then used to build the GMM for the image. Using the resulting GMM, the 'segmented' image is produced. In the case where color features are used, the segmented image is produced by assigning to each pixel (x, y) the mean color value of the corresponding GMM component. In the case of texture images we use an arbitrary different color for each segment.

For segmentation using artificial color features we used the (L, a, b) representation along with the (x, y) coordinates. Figure 3.(a) illustrates the segmentation result using an artificial color image. Figure 3.(b),(c) displays segmentation results with two natural images from the Berkeley Segmentation Data Set (BSDS) [5]. For these images we also added a texture feature, namely the polarity feature pl [1] to the feature vector describing a pixel, i.e. $f_n = (L, a, b, pl, x, y)_n$. Finally Figure 4 provides results for two artificial texture images and one BSDS image, using only the polarity feature, i.e. $f_n = (pl, x, y)_n$.

From the experimental results it is clear that, although only a subset of the pixels are used for GMM training, the method yields segmentations of satisfactory quality, that are spatially smooth and estimates reasonably well the number of image segments, without producing significant oversegmentation.

4 Conclusions

We have presented an efficient approach for image modeling and segmentation using GMMs that is based on a recently proposed Bayesian technique for estimating the components of a GMM [3]. The approach is fully automatic, makes no assumptions regarding the required number of GMM components and provides solutions that are spatially smooth. Future work will focus on a more systematic testing of the method in the case where other texture-related features are used and in the case where feature vectors are defined at the patch level instead of the pixel level. Also we plan to test the efficiency of the method in image retrieval tasks where GMMs are used as models of the stored images and the query image.

References

1. Carson, C., Belongie, S., Greenspan, H., Malik, J.: Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(8), 1026–1038 (2002)
2. Li, S.Z.: *Markov Random Field Modelling in Computer Vision*. Springer, Heidelberg (2001)
3. Constantinopoulos, C., Likas, A.: Bayesian gaussian mixture learning based on variational component splitting. *IEEE Transactions on Neural Networks* 18(3), 745–755 (2007)
4. Corduneanu, A., Bishop, C.M.: Variational Bayesian model selection for mixture distributions. In: *Artificial Intelligence and Statistics 2001*, pp. 27–34. Morgan Kaufmann, San Francisco (2001)
5. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, pp. 416–423 (2001)