SAR IMAGES AS MIXTURES OF GAUSSIAN MIXTURES

Peter Orbanz and Joachim M. Buhmann

Institute of Computational Science, ETH Zurich

ABSTRACT

We consider the problem of image segmentation by clustering local histograms with parametric mixture-of-mixture models. These models represent each cluster by a single mixture model of simple parametric components, typically truncated Gaussians. Clustering requires unsupervised inference of the model parameters, for which we derive a nested variant of the EM algorithm. This learning procedure is designed to deal with the large number of hidden variables required by the model. Results are presented for application of the algorithm to unsupervised segmentation of synthetic aperture radar (SAR) images.

1. INTRODUCTION

Image segmentation defines a fundamental problem in image analysis, which can be addressed by applying clustering (or grouping) methods to image data [1]. These methods attempt to group data into a user-specified number of groups, called the clusters, by means of unsupervised learning techniques. A widely used clustering technique assumes a parametric model for each cluster and thus models the data distribution by a parametric mixture model, often a mixture of Gaussians. This approach works if the individual cluster distributions can be approximated by Gaussians, but it is less well applicable if the cluster distributions deviate from the Gaussian shape or are multimodal. More complicated cluster distributions can be modeled by mixture-of-mixtures models, i. e. mixture models where each component itself is composed of a parametric mixture model. Models of this type have been introduced as mixture of experts in the supervised learning context by Jordan and Jacobs [2]. Gaussian mixture-of-mixtures models have previously been applied to image segmentation in [3], where the model is optimized by deterministic annealing, and model components are coupled between clusters to decrease computational complexity. In this paper, we consider an optimization of the model which avoids coupling constraints between clusters. Each cluster is represented by a Gaussian mixture of two or three components, which allows us to approximate non-Gaussian unimodal densities (such as gamma or logarithmic gamma densities for SAR image processing), as well as bimodal densities. Unsupervised learning of mixture models is usually performed by the Expectation-Maximization (EM) algorithm [4, 5]. We propose a nested EM algorithm to account for the hierarchical structure of the mixture-of-mixtures model. EM iterations for mixture model optimization alternate between assigning data to clusters based on the current configuration of the model, and adjusting the model by maximum likelihood (ML) estimation based on the current assignment of the data. Since the components themselves are mixture models, we perform ML estimation by executing an inner EM algorithm for each cluster within the estimation step. A nested version of the EM algorithm has been discussed in a different context by van Dyk [6], who suggests nesting to improve convergence rates of E-steps which rely on Monte-Carlo integration. Our algorithm, on the other hand, is nesting the M-step to perform estimation for the components of the hierarchical mixture model.

2. SEGMENTATION APPROACH

Our approach to image segmentation is based on histogram clustering. The features extracted from the image are histogram representations of the local data distributions in the neighborhood of image pixels. The histograms are grouped into a pre-specified number of clusters, each of which is modeled by a parametric mixture model.

For a grayscale input image, we extract local histograms from the image at the sites of an equidistant grid. The local histogram at a given grid point is extracted by centering a window at the respective pixel, selecting all pixels within the window and sorting their grayscale values into a histogram. This procedure results in a set of histograms $\mathbf{n}_i = (n_{i1}, \ldots, n_{iN_{\text{Bins}}})$. Here $i = 1, \ldots, N_{\text{Sites}}$ indexes the grid points and N_{Bins} is the number of histogram bins, so n_{ij} denotes the counts in bin j of histogram i. We assume that all histograms contain an identical total number N_{Counts} of counts.

The data is modeled by a *mixture-of-mixtures model*, i. e. a finite mixture model the component densities of which are themselves represented by finite mixtures. In this work, all component mixture densities consist of an identical num-

0-7803-9134-9/05/$20.00

ber N_{Modes} of Gaussian components:

$$p\left(x|\Theta\right) = \sum_{\tau=1}^{N_{\text{Clusters}}} c_\tau p_\tau\left(x\right) = \sum_{\tau=1}^{N_{\text{Clusters}}} c_\tau \left(\sum_{\alpha=1}^{N_{\text{Modes}}} c_\alpha^\tau g_\alpha^\tau\left(x\right) \right) , \tag{1}$$

where $g_\alpha^\tau\left(x\right) = g\left(x|\mu_\alpha^\tau, \sigma_\alpha^\tau\right)$ denotes a normal density and Θ the full set of Gaussian parameters. c_τ, c_α^τ are the priors of the segments and the modes, respectively. We expect the local image histograms to be uni- or at most bimodal, so we are interested only in cases where the number of inner components is small (typically $N_{\text{Modes}} = 2, 3$).

Since the range of digital image data is restricted to a finite intensity interval, we have to truncate the Gaussians. These distributions are referred to as *rectified distributions* in the literature [7]. Rectification somewhat complicates parameter estimation, because a ML estimator for a Gaussian mean or variance parameter is not a valid ML estimator for the rectified Gaussian.

3. INFERENCE ALGORITHM

EM algorithms for mixture models: Unsupervised inference of mixture models is usually conducted by the EM algorithm [4, 5]. Assignments of sites (indexed by i) to clusters (indexed by τ) are encoded by *hidden variables*, which we denote by $M_{i\tau}$. These assume binary values, with $M_{i\tau} = 1$ if site i is assigned to cluster τ and $M_{i\tau} = 0$ otherwise, satisfying the constraint $\sum_\tau M_{i\tau} = 1$. The EM algorithm relaxes binary assignments to real numbers in $[0, 1]$ by computing expectations. For optimization of the standard mixture model of the form $\sum_\tau c_\tau p_\tau\left(x|\Theta_\tau\right)$, the algorithm iterates the following two steps: Compute expectations $\mathbb{E}\left[M_{i\tau}\right] = \frac{c_\tau p_\tau(x_i|\Theta_\tau)}{\sum_\nu c_\nu p_\nu(x_i|\Theta_\nu)}$ (E-step) and maximize $Q(\Theta, \tilde{\Theta}) := \mathbb{E}_{M|\mathbf{x},\tilde{\Theta}}\left[l\left(\Theta\right)\right]$ with respect to Θ (M-step), where $\tilde{\Theta}$ denotes the parameter values computed during the previous step. The latter expression can be shown to be

$$Q(\Theta, \tilde{\Theta}) = \sum_{i,\tau} \mathbb{E}_{M|\mathbf{x},\tilde{\Theta}}\left[M_{i\tau}\right] \log\left(c_\tau p_\tau\left(x_i|\Theta_\tau\right)\right) , \tag{2}$$

which is computed by substituting the values obtained during the E-step for the expected assignments.

EM for histogram data: We assume our input data to be a set $\mathbf{n} = \left(\mathbf{n}_1, \ldots, \mathbf{n}_{N_{\text{Sites}}}\right)$ of histograms, drawn i. i. d. from a source modeled by a parametric density of the form (1). Denote by I_j the interval in the data domain corresponding to bin j. For a histogram drawn from cluster τ, the probability of a data value to fall into bin I_j is $p_j^\tau\left(\Theta\right) = \int_{I_j} p_\tau\left(x|\Theta_\tau\right) dx$. Given the probabilities of occurrence $p_1^\tau\left(\Theta\right), \ldots, p_{N_{\text{Bins}}}^\tau\left(\Theta\right)$, the probability for any one histogram $\mathbf{n}_i$ to occur is multinomially distributed according to

$$p_\tau\left(\mathbf{n}_i|\Theta\right) := N_{\text{Counts}}! \prod_{j=1}^{N_{\text{Bins}}} \frac{p_j^\tau\left(\Theta\right)^{n_{ij}}}{n_{ij}!} . \tag{3}$$

Including assignment variables for the EM algorithm, $\mathbf{n}$ and $\mathbf{M}$ are jointly distributed according to

$$p\left(\mathbf{n}, \mathbf{M}|\Theta\right) := \prod_i \sum_\tau M_{i\tau} c_\tau p_\tau\left(\mathbf{n}_i|\Theta_\tau\right) . \tag{4}$$

The resulting log-likelihood is

$$\begin{aligned} l\left(\Theta\right) &= \sum_{i=1}^{N_{\text{Sites}}} \left(\log\left(N_{\text{Counts}}!\right) - \sum_{j=1}^{N_{\text{Bins}}} \log\left(n_{ij}!\right) \right) \\ &+ \sum_{i,\tau} M_{i\tau} \log\left(c_\tau\right) + \sum_{i,j,\tau} M_{i\tau} n_{ij} \log\left(p_j\left(\Theta_\tau\right)\right) , \end{aligned} \tag{5}$$

using the standard EM trick of drawing a sum over normalized binary assignments through the logarithm. Since the first sum in the log-likelihood is a constant of the input data, we may drop it for the EM target function:

$$Q(\Theta, \tilde{\Theta}) := \sum_{i,\tau} \mathbb{E}\left[M_{i\tau}\right] \left(\log c_\tau + \sum_j n_{ij} \log\left(p_j\left(\Theta_\tau\right)\right) \right) .$$

Nested EM algorithm for the hierarchical model: Each component of our model (1) is again a Gaussian mixture. Optimization of the model requires a ML estimation for a simple Gaussian mixture model in the M-step. Therefore, we perform the M-step by executing an EM algorithm for each component mixture model. The approach requires hierarchical assignments: Variables for the outer EM loop, which indicate cluster assignments and will again be denoted $M_{i\tau}$, and a complete set of assignment variables for each inner mixture, denoted $M_{i\alpha}^\tau$, where i indicates the site, τ the cluster and α the Gaussian mode. Additionally, for the inner EM algorithm, we drop the assumption that each site is assigned to a model component: A site not assigned to the cluster in question ($M_{i\tau} = 0$ for the current cluster τ) should not be taken into account by the inner loop. Thus, $M_{i\alpha}^\tau = 1$ indicates that site i is assigned to component α of cluster τ iff $M_{i\tau} = 1$. We define effective inner assignment indicators $L_{i\alpha}^\tau$ by

$$L_{i\alpha}^\tau := M_{i\alpha}^\tau \cdot M_{i\tau} . \tag{6}$$

To make the algorithmic treatment feasible, we assume statistical independence of $M_{i\alpha}^\tau$ and $M_{i\tau}$.

The outer algorithm computes expectations in the E-step according to

$$\mathbb{E}\left[M_{i\tau}\right] = \frac{c_\tau p_\tau\left(\mathbf{n}_i|\Theta_\tau\right)}{\sum_\nu c_\nu p_\nu\left(\mathbf{n}_i|\Theta_\nu\right)} . \tag{7}$$

The M-step computes the mixture weights c_τ from the outer assignments as $c_\tau = \sum_i \mathbb{E}\left[M_{i\tau}\right] / N_{\text{Sites}}$.

The inner loop consists of one EM algorithm for each cluster, which is initialized by the final inner model parameters obtained for the current cluster by previous execution

of the inner loop (i. e. during the previous step of the outer algorithm). The E-step computes expectations as

$$\mathbb{E}\left[M_{i\alpha}^{\tau}\right] = \frac{c_{\alpha}^{\tau} p_{\alpha}^{\tau}\left(\mathbf{n}_i|\Theta_{\alpha}^{\tau}\right)}{\sum_{\nu} c_{\nu}^{\tau} p_{\nu}^{\tau}\left(\mathbf{n}_i|\Theta_{\nu}^{\tau}\right)} \;, \tag{8}$$

where, in our case, $p_{\alpha}^{\tau}\left(\mathbf{n}_i|\Theta_{\alpha}^{\tau}\right) = g\left(\mathbf{n}_i|\mu_{\alpha}^{\tau}, \sigma_{\alpha}^{\tau}\right)$. Since we assume independence, we can compute expectations for the effective inner assignments $L_{i\alpha}^{\tau}$ as

$$\mathbb{E}\left[L_{i\alpha}^{\tau}\right] = \mathbb{E}\left[M_{i\alpha}^{\tau}\right] \cdot \mathbb{E}\left[M_{i\tau}\right] \;. \tag{9}$$

The M-steps require one target function for each cluster:

$$Q^{\tau}(\Theta^{\tau}, \tilde{\Theta}^{\tau}) = \sum_{i,\alpha} \mathbb{E}\left[L_{i\alpha}^{\tau}\right] \log\left(c_{\alpha}^{\tau} p_{\alpha}^{\tau}\left(\mathbf{n}_i|\Theta_{\alpha}^{\tau}\right)\right) \;. \tag{10}$$

By substituting histogram probabilities as in (5), we obtain

$$\begin{aligned} Q^{\tau}(\Theta^{\tau}, \tilde{\Theta}^{\tau}) \;&=\; \sum_{i} J\left(\mathbf{n}_i\right) + \sum_{i,\alpha} \mathbb{E}\left[L_{i\alpha}^{\tau}\right] \log\left(c_{\alpha}^{\tau}\right) \\ &+\; \sum_{i,\alpha,j} n_{ij} \mathbb{E}\left[L_{i\alpha}^{\tau}\right] \log\left(p_{\alpha j}^{\tau}\left(\Theta_{\alpha}^{\tau}\right)\right) , \end{aligned} \tag{11}$$

where $J\left(\mathbf{n}_i\right)$ denotes the constant term depending only on the input data, which can again be neglected for optimization purposes. Of the two remaining terms, one depends only on the inner mixture weights c_{α}^{τ} and one on the mode parameters Θ_{α}^{τ}. Therefore, the two terms can be optimized independently. Solving for the mixture weights gives

$$c_{\alpha}^{\tau} := \frac{\sum_i \mathbb{E}\left[L_{i\alpha}^{\tau}\right]}{\sum_{i,\alpha} \mathbb{E}\left[L_{i\alpha}^{\tau}\right]} = \frac{\sum_i \mathbb{E}\left[L_{i\alpha}^{\tau}\right]}{c_{\tau}} \;. \tag{12}$$

ML estimation for the Gaussian parameters during the inner M-step has to be conducted by numerical optimization of the last term in (11), because ML equations for rectified Gaussians lack closed-form solutions. The last term of (11) may be regarded, up to histogram normalization, as a cross-entropy between the average cluster data distribution and the discretized cluster model distribution. This can be turned into the negative Kullback-Leibler divergence between the two discrete distributions by adding the average data distribution's entropy (cf. [8]). ML estimation is therefore equivalent to minimization of the KL divergence between the data and the discretized model. Instead of computing ML estimators for the rectified model, we minimize the KL divergence on the restricted domain.

As a stopping criterion for both the outer and inner EM algorithm, we can threshold the change in assignments between consecutive steps. During the first steps of the algorithm, however, the assignments in the outer loop are still subject to large changes. It turns out that, by gradually increasing the number of inner iterations with each outer step, we can obtain results comparable (and sometimes superior) to a thresholding approach. The outer loop can then be interpreted as a generalized EM algorithm [5], since the M-step (the inner EM loop) is not designed to fully maximize the log-likelihood, only to increase it.

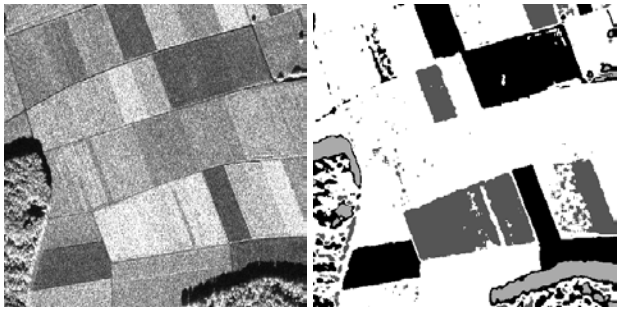

Fig. 1. SAR image and segmentation solution obtained with 4 clusters and 3 modes per cluster.

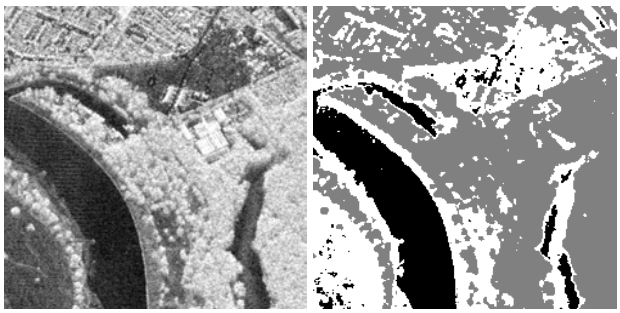

Fig. 2. SAR image and segmentation solution for 3 clusters, 3 modes per clusters.

4. APPLICATION TO SAR DATA

SAR image segmentation is an interesting application for mixture-of-mixtures models, because SAR data is known to be distributed in a characteristic fashion. The gamma distribution (and several other, closely related distributions) have been suggested as parametric models for this data [9]. A gamma distribution can be approximated roughly by a single Gaussian, but very closely by a mixture of two or more Gaussians. For certain parameter configurations, gamma distributions are monotonously decreasing rather than peaked; these cases can be closely approximated by the right tail of

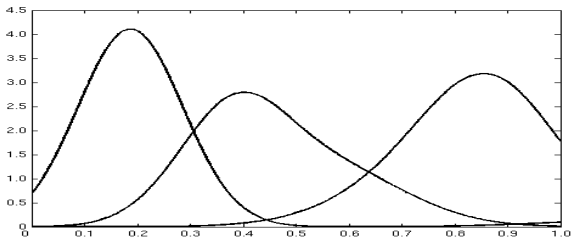

Fig. 3. Cluster distributions (i. e. summed inner Gaussian mixtures) for the clustering solution in Fig. 2. Two modes closely resemble Gaussians, one is clearly non-Gaussian.

a Gaussian when using a rectified model. If we assume that each segment is roughly gamma distributed, we can thus apply our algorithm to SAR image segmentation by clustering local histograms extracted from a SAR image using Gaussian mixtures. Figs. 1 and 2 show segmentation solutions obtained by our algorithm on two different SAR images. The locally correlated structure of the errors is typical for histogram data with overlapping windows: Due to the size of the histogram window, local deviations from the average distribution of the segment enter in all histograms within a certain neighborhood, which are then erroneously assigned. Fig. 3 provides a plot of the summed Gaussian mixtures modelling the three clusters for the solution in Fig. 2. The middle mode shows how a Gaussian mixture can model a distribution of typical gamma shape.

SAR image data is often processed by logarithmic transforms. Deriving appropriate model distributions for this processed data has proven rather difficult (see, for example, [10]). The shape of the resulting distribution is roughly of reversed gamma shape, i. e. it resembles a gamma distribution of inverse skewness. The Gaussian mixtures of our model are just as suited to model this kind of data as to model gamma distributions, since a Gaussian mixture approximating a gamma distribution can be turned into a distribution of reversed gamma shape by simply shifting components.

5. DISCUSSION AND FURTHER WORK

The algorithm converges despite the large number of hidden variables. In general, the performance of EM algorithms is known to deteriorate as the number of hidden variables (and thus their total entropy) increases. The special property of our algorithm is that each of the inner EM algorithms works only on a fixed subset of hidden variables, leaving all others untouched; therefore, only the entropy of the hidden variables in the subset is relevant for the performance of the corresponding inner loop. Since these subsets are pairwise disjoint, sequential execution of the inner algorithms performs a consecutive series of steps on orthogonal subspaces of the space spanned by the hidden variables, as a refinement of the alternating series of orthogonal maximization steps performed by a standard EM algorithm.

In theory, the hierarchical structure of both the model and our algorithm could be extended to a nesting depth greater than two, but due to the increasingly complicated structure of the hierarchy of hidden variables and the question of model identifiability, optimization of models with more than two layers is unlikely to be reliable. The image segmentation approach was described here only for grayscale or single-channel image data. It can easily be extended to multiple channels (without increasing the number of hidden variables) by using marginal histograms for each channel. In this case, the EM target function becomes a simple sum over the channels. Since short inner iterations seem to be advisable during the first few loops of the overall algorithm, one might consider a stopping criterion for the inner loop depending on the change of assignments in the outer loop, so optimization in the inner loop becomes increasingly precise as cluster assignments become more reliable. Further work will address the issue of how the model complexity may be adjusted in a data-dependent manner.

Acknowledgment: The SAR images are shown courtesy of Infoterra GmbH, Friedrichshafen, Germany (Fig. 1) and Definiens AG, Munich, Germany (Fig. 2).

6. REFERENCES

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review.," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264– 323, 1999.

[2] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation*, vol. 6, pp. 181–214, 1994.

[3] L. Hermes, T. Zöller, and J. M. Buhmann, "Parametric distributional clustering for image segmentation," in *Computer Vision - ECCV 2002*. 2002, vol. 3 of *LNCS 2352*, pp. 577–591, Springer.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Stat. Soc. B*, vol. 39, no. 1, pp. 1–38, 1977.

[5] G. J. McLachlan and T. Krishnan, *The EM algorithm and extensions*, John Wiley & Sons, 1997.

[6] D. A. van Dyk, "Nesting EM algorithms for computational efficiency," *Statistica Sinica*, vol. 10, pp. 203–225, 2000.

[7] N. D. Socci, D. D. Lee, and H. S. Seung, "The rectified gaussian distribution," in *Advances in Neural Information Proscessing Systems*, M. I. Jordan, M. Kearns, and S. Solla, Eds. 1998, pp. 350–356, MIT Press.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, 1991.

[9] C. Oliver and S. Quegan, *Understanding Synthetic Aperture Radar Images*, Artech House, 1998.

[10] H. Xie, L. E. Pierce, and F. T. Ulaby, "Statistical properties of logarithmically transformed speckle," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 40, no. 3, 2002.