Acknowledgment

The author thanks R. Riedi, L. Barreira, and L. Olsen for helpful comments on the multifractal formalism. He is also thankful to anonymous reviewers and G. Dorffner for helpful comments on the manuscript.

References

[1] G. Mayer-Kress, R. Bargar, and I. Choi, "Musical structures in data from chaotic attractors," Tech. Rep. CCSR-92-14, Center for Complex Syst. Res., Beckman Inst., Univ. Illinois, Urbana-Champaign, 1992.

[2] C. L. Berthelsen, J. A. Glazier, and M. H. Skolnik, "Global fractal dimension of human DNA sequences treated as pseudorandom walks," *Phys. Rev. A,* vol. 45, no. 12, pp. 8902–8913, 1992.

[3] H. E. Stanley, S. V. Buldyrev, A. L. Goldberger, Z. D. Goldberger, S. Havlin, R. N. Mantegna, S. M. Ossadnik, C.-K. Peng, and M. Simons, "Statistical mechanics in biology: How ubiquitous are long-range correlations?," *Phys. A,* vol. 205, pp. 214–253, 1994.

[4] J. Jeffrey, "Chaos game representation of gene structure," *Nucl. Acids Res.,* vol. 18, no. 8, pp. 2163–2170, 1990.

[5] M. F. Barnsley, *Fractals Everywhere.* New York: Academic, 1988.

[6] J. L. Oliver, P. Bernaola-Galván, J. Guerrero-Garcia, and R. Román Roldan, "Entropic profiles of DNA sequences through chaos-game-derived images," *J. Theor. Biol.,* no. 160, pp. 457–470, 1993.

[7] R. Roman-Roldan, P. Bernaola-Galvan, and J. L. Oliver, "Entropic feature for sequence pattern through iteration function systems," *Pattern Recognit. Lett.,* vol. 15, pp. 567–573, 1994.

[8] S. Basu, A. Pan, C. Dutta, and D. J, "Chaos game representation of proteins," *J. Mol. Graph. Model.,* vol. 15, no. 5, pp. 279–289, 1997.

[9] V. V. Solovyev, S. V. Korolev, and H. A. Lim, "A new approach for the classification of functional regions of DNA sequences based on fractal representation," *Int. J. Genom. Res.,* vol. 1, no. 1, pp. 109–128, 1993.

[10] A. Fiser, G. E. Tusnady, and I. Simon, "Chaos game representation of protein structures," *J. Mol. Graph.,* vol. 12, no. 4, pp. 302–304, 1994.

[11] K. A. Hill and S. M. Singh, "The evolution of species-type specificity in the global DNA sequence organization of mitochondrial genomes," *Genome,* vol. 40, pp. 342–356, 1997.

[12] W. Li, "The study of correlation structures of DNA sequences: A critical review," *Comput. Chem.,* vol. 21, no. 4, pp. 257–272, 1997.

[13] R. Roman-Roldan, P. Bernaola-Galvan, and J. L. Oliver, "Application of information theory to DNA sequence analysis: A review," *Pattern Recognit.,* vol. 29, no. 7, pp. 1187–1194, 1996.

[14] Y. Pesin, *Dimension Theory in Dynamical Systems: Rigorous Results and Applications.* Chicago, IL: Univ. of Chicago Press, 1997.

[15] P. Tiňo and M. Köteleš, "Extracting finite state representations from recurrent neural networks trained on chaotic symbolic sequences," *IEEE Trans. Neural Networks,* vol. 10, pp. 284–302, Mar. 1999.

[16] R. Kenyon and Y. Peres, "Measures of full dimension on affine invariant sets," *Ergod. Theory Dynam. Syst.,* vol. 16, pp. 307–323, 1996.

[17] A. I. Khinchin, *Mathematical Foundations of Information Theory.* New York: Dover, 1957.

[18] A. Renyi, "On the dimension and entropy of probability distributions," *Acta Math. Hung.,* no. 10, pp. 193, 1959.

[19] P. Grassberger, "Information and complexity measures in dynamical systems," in *Information Dynamics,* H. Atmanspacher and H. Scheingraber, Eds. New York: Plenum, 1991, pp. 15–33.

[20] J. P. Crutchfield and K. Young, "Computation at the onset of chaos," in *Complexity, Entropy, and the Physics of Information, SFI Studies in the Sciences of Complexity,* W. H. Zurek, Ed. Reading, MA: Addison-Wesley, 1990, pp. 223–269, vol. 8.

[21] C. Beck and F. Schlogl, *Thermodynamics of Chaotic Systems.* Cambridge, U.K.: Cambridge Univ. Press, 1995.

[22] J. L. McCauley, *Chaos, Dynamics and Fractals: An Algorithmic Approach to Deterministic Chaos.* Cambridge, U.K.: Cambridge Univ. Press, 1994.

[23] R. H. Riedi, "Conditional and relative multifractal spectra," *Fractals,* vol. 5, no. 1, pp. 153–168, 1997.

[24] K. J. Falconer, *Fractal Geometry: Mathematical Foundations and Applications.* New York: Wiley, 1990.

[25] L. Staiger, "Quadtrees and the Hausdorff dimension of pictures," in *Workshop Geometrical Problems Image Processing, Georgental, GDR,* 1989, pp. 173–178.

[26] K. Culik, II and S. Dube, "Affine automata and related techniques for generation of complex images," *Theor. Comput. Sci.,* vol. 116, no. 2, pp. 373–398, 1993.

[27] ——, "Rational and affine expressions for image description," *Discrete Appl. Math.,* vol. 41, pp. 85–120, 1993.

[28] Y. Pesin and H. Weiss, "On the dimension of deterministic and cantor-like sets, symbolic dynamics, and the Eckmann–Ruelle conjecture," *Commun. Math. Phys,* vol. 182, no. 1, pp. 105–153, 1996.

[29] P. Moran, "Additive functions of intervals and Hausdorff dimension," in *Proc. Cambridge Philos. Soc.,* 1946, vol. 42, pp. 15–23.

[30] Y. Pesin and H. Weiss, "A multifractal analysis of equilibrium measures for conformal expanding maps and Moran-like geometric constructions," *J. Stat. Phys.,* vol. 86, no. 1/2, pp. 233–275, 1997.

[31] L. Barreira, Y. Pesin, and J. Schmeling, "On a general concept of multifractality: Multifractal spectra for dimensions, entropies, and Lyapunov exponents multifractal rigidity," *Chaos: Interdiscipl. J. Nonlinear Sci.,* vol. 7, no. 1, pp. 27–53, 1996.

[32] D. Ron, Y. Singer, and N. Tishby, "The power of amnesia," *Mach. Learn.,* vol. 25, 1996.

[33] M. J. Weinberger, J. J. Rissanen, and M. Feder, "A universal finite memory source," *IEEE Trans. Inform. Theory,* vol. 41, pp. 643–652, May 1995.

[34] P. Tiňo and G. Dorffner, "Constructing finite-context sources from fractal representations of symbolic sequences," Tech. Rep. TR-98-18, Austrian Res. Inst. Artif. Intell., 1998.

A Kurtosis-Based Dynamic Approach to Gaussian Mixture Modeling

Nikos Vlassis and Aristidis Likas

***Abstract*— We address the problem of probability density function estimation using a Gaussian mixture model updated with the expectation-maximization (EM) algorithm. To deal with the case of an unknown number of mixing kernels, we define a new measure for Gaussian mixtures, called total kurtosis, which is based on the weighted sample kurtoses of the kernels. This measure provides an indication of how well the Gaussian mixture fits the data. Then we propose a new dynamic algorithm for Gaussian mixture density estimation which monitors the total kurtosis at each step of the EM algorithm in order to decide dynamically on the correct number of kernels and possibly escape from local maxima. We show the potential of our technique in approximating unknown densities through a series of examples with several density estimation problems.**

***Index Terms*— Expectation-maximization (EM) algorithm, Gaussian mixture modeling, number of mixing kernels, probability density function estimation, total kurtosis, weighted kurtosis.**

I. Introduction

The Gaussian mixture model [1] has been proposed as a general model for estimating an unknown probability density function, or simply density. The virtues of the model lie mainly in its good

Manuscript received September 17, 1998.

N. Vlassis is with the RWCP, Autonomous Learning Functions SNN, Department of Computer Science, University of Amsterdam, 1098 SJ Amsterdam, The Netherlands (e-mail: vlassis@wins.uva.nl).

A. Likas is with the Department of Computer Science, University of Ioannina, 45110 Ioannina, Greece (e-mail: arly@cs.uoi.gr).

Publisher Item Identifier S 1083-4427(99)05344-8.

approximation properties and the variety of estimation algorithms that exist in the literature [1], [2]. The model assumes that the unknown density can be written as a weighted finite sum of Gaussian kernels, with different mixing weights and different parameters, namely, means and covariance matrices. Then, depending on the estimation algorithm, an optimum vector of these parameters is sought that optimizes some criterion. Most often, the estimation of the parameters of the mixture is carried out by the maximum likelihood method, aiming at maximizing the likelihood of a set of samples drawn independently from the unknown density.

One of the algorithms often used for Gaussian mixture modeling is the expectation-maximization (EM) algorithm, a well-known statistical tool for maximum likelihood problems [3]. The algorithm provides iterative formulae for the estimation of the unknown parameters of the mixture, and can be proven to monotonically increase in each step the likelihood function. However, the main drawbacks of EM is that it requires an initialization of the parameter vector near the solution, and also it assumes that the total number of mixing kernels is known in advance.

To overcome the above limitations, we propose in this paper a novel dynamic algorithm for Gaussian mixture modeling that starts with a small number of kernels K (usually $K = 1$) and performs EM steps in order to maximize the likelihood of the data, while at the same time monitors the value of a new measure of the mixture, called *total kurtosis*, that indicates how well the Gaussian mixture fits the input data. This new measure is computed from the individual *weighted sample kurtoses* of the mixing kernels, defined by analogy to the weighted means and variances of the kernels and first introduced in [4] for on-line density estimation. Based on the progressive change of the total kurtosis, our algorithm performs kernel splitting and increases the number of kernels of the mixture. This splitting aims at making the absolute value of the total kurtosis as small as possible.

By performing dynamic kernel allocation, the proposed algorithm is capable of finding a good estimation of the number of kernels of the mixture, while it does not require any prior initialization near the solution. As experiments indicate, our approach seems to be superior to the original EM algorithm in approximating an unknown density. This fact renders it a good alternative for Gaussian modeling, especially in cases where little information about the density to be approximated is available beforehand.

In the neural networks literature, a feed-forward network that implements a Gaussian mixture is the probabilistic neural network [5]. The network uses one Gaussian kernel for each input sample, while the variance of each kernel is constant and known and the mixing weights are equal to the reciprocal of the total number of inputs. The network can be regarded as a distributed implementation of the Parzen windows method [6]. Some of the limitations of the original network model were relaxed in subsequent works [7]–[10], leading to network models that implement some variants of the EM algorithm. However, in most approaches the number K of kernels of the mixture is considered known in advance, and it turns out that the automatic estimation of K is a difficult problem [1], [11].

Statistical methods or neural network models for estimating the number of kernels of a Gaussian mixture have been proposed in the literature [1], [9], [12], [13]. However, most of them usually cannot satisfy the necessary regularity conditions for estimating the asymptotic distributions of the underlying tests, and thus have to resort to costly heuristic techniques, e.g., Monte Carlo bootstrapping, for obtaining a solution.

In Section II we review the use of Gaussian mixtures as models for probability density estimation, and show how the EM algorithm can be used for obtaining maximum likelihood solutions. In Section III we describe our algorithm. We first define the new measure of total kurtosis that is needed by the algorithm and then show how the algorithm can make use of this quantity to produce better solutions. We consider the univariate case only. Work is under progress to extend the definition and use of total kurtosis in higher dimensions. Section IV gives experimental results from the application of the algorithm to density estimation problems, while Section V summarizes and gives hints for future research.

II. Gaussian Mixtures and the EM Algorithm

A. Gaussian Mixtures

We say that a random variable x has a finite mixture distribution when its probability density function $p(x)$ can be written as a finite weighted sum of known densities, or simply kernels. In cases where each kernel is the Gaussian density, we say that x follows a Gaussian mixture. For the univariate case and for a number K of Gaussian kernels, the unknown mixture can be written

$$p(x) = \sum_{j=1}^{K} \pi_j p(x|j) \tag{1}$$

where $p(x|j)$ stands for the univariate Gaussian $N(\mu_j, \sigma_j)$

$$p(x|j) = \frac{1}{\sigma_j \sqrt{2\pi}} \exp\left[\frac{-(x-\mu_j)^2}{2\sigma_j^2}\right] \tag{2}$$

parametrized on the mean μ_j and the variance σ_j^2. In order for $p(x)$ to be a probability density function with integral 1 over the input space, the additional constraints on the weights π_j of the mixture must hold

$$\sum_{j=1}^{K} \pi_j = 1, \quad \pi_j \geq 0. \tag{3}$$

The Gaussian mixture model is general and under regular conditions it may approximate any continuous function having a finite number of discontinuities [11].

For the estimation problem we assume a training set $X = (x_1, \cdots, x_n)$, of n independent and identically distributed samples of the random variable x, taking values from an input space, e.g., in the univariate case the real line $\mathbb{R}$. Training aims at finding the number of kernels K and the optimum vector θ^* of the $3K$ parameters of the mixture

$$\theta^* = (\pi_1^*, \mu_1^*, \sigma_1^*, \ldots, \pi_K^*, \mu_K^*, \sigma_K^*) \tag{4}$$

that maximizes the likelihood function

$$\theta^* = \arg\max_{\theta} L(\theta), \quad L(\theta) = \prod_{i=1}^{n} p(x_i). \tag{5}$$

Although efficient methods exist for the estimation of the $3K$ parameters of the mixture from a set of samples of x, the automatic estimation of K remains a difficult problem [11].

B. The EM Algorithm

The EM algorithm [2], [3], [14] is a powerful statistical tool for finding maximum likelihood solutions to problems involving observed and hidden variables. The algorithm applies in cases where we ask for maximum likelihood estimates for some observed variables X but we do not know the exact form of their probability density function. Instead, we can compute the joint density of these variables and some hidden variables Y.

At each EM step the algorithm computes the quantity

$$Q\left(\theta|\theta^{(t)}\right) = E_Y\left[\log p(X, Y|\theta)|X, \theta^{(t)}\right] \tag{6}$$

which is a function of the parameter vector θ and which is obtained by averaging the logarithm of the joint density of X and Y, conditioned on θ, over the hidden variables Y, given the observations X and the current estimate of the parameter vector $\theta^{(t)}$ (E step). Then, the next estimate $\theta^{(t+1)}$ of the parameter vector is computed as the value that maximizes the quantity Q (M step). Alternating these two steps, the EM algorithm can be shown [2] to monotonically increase the likelihood of the observations X, thus yielding an optimum θ^* in the maximum likelihood sense.

The problem of estimating an unknown Gaussian mixture by maximizing the likelihood of the parameter vector θ can be regarded as a problem with hidden variables and thus solved by using the EM algorithm. In this case, the hidden variables are the kernels the input samples statistically belong to, while each EM step provides an improved estimate of the parameters π_j, μ_j, and σ_j of each kernel j, $j = 1, \cdots, K$. These iterative formulae can be shown [2] to be

$$\pi_j^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} P(j|x_i), \tag{7}$$

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{n} P(j|x_i)x_i}{\sum_{i=1}^{n} P(j|x_i)} \tag{8}$$

$$\sigma_j^{2\,(t+1)} = \frac{\sum_{i=1}^{n} P(j|x_i)\left(x_i - \mu_j^{(t+1)}\right)^2}{\sum_{i=1}^{n} P(j|x_i)} \tag{9}$$

where

$$P(j|x_i) = \frac{\pi_j^{(t)} p\left(x_i|j;\, \mu_j^{(t)},\, \sigma_j^{(t)}\right)}{\sum_{k=1}^{K} \pi_k^{(t)} p\left(x_i|k;\, \mu_k^{(t)},\, \sigma_k^{(t)}\right)}. \tag{10}$$

It is not difficult to see that the weights π_j satisfy the conditions (3) after applying the above formulae for all kernels.

It is useful here to make a qualitative analysis of the above formulae. In each EM step we use the posterior probability $P(j|x_i)$ that a sample x_i belongs statistically to kernel j when the prior probability of the latter is π_j, and which is computed in (10) by applying the continuous version of Bayes' theorem. This quantity $P(j|x_i)$ is computed for every kernel j from the previous estimates of the parameters π_j, μ_j, and σ_j for this kernel, and for input x_i. Then it is summed over all input samples x_i for the estimation of the new priors π_j (7), it is summed weighted by the inputs x_i for the estimation of the new means (8), and it is summed weighted by the square distances of the input samples x_i to the new means for the estimation of the new variances σ_j^2 (9). By analogy to the formulae of the sample moments in statistics, the estimated parameters in each EM step of the algorithm can be regarded as weighted sample moments of the random variable x, with the weights being the posterior probabilities $P(j|x_i)$.

C. Limitations of EM

In problems of Gaussian mixtures, due to the nonlinearities of the underlying densities with respect to their parameters μ and σ, the likelihood function exhibits almost surely local maxima or saddle points. It is desirable, therefore, that a maximum likelihood estimation method should be able to escape from such local maxima.

Although the EM algorithm monotonically increases in each step the likelihood of the observations, it cannot ensure convergence of the parameter vector θ^* to a global maximum (satisfying appropriate nonsingularity conditions) of the parameter space [2]. The algorithm may easily get stuck in a local maximum or saddle point. Moreover, so far we have assumed a known number of mixing kernels and thus have obtained the iterative solutions (7)–(9). In practice this is hardly true: K is usually unknown and has also to be estimated from the input samples. These two constraints hamper severely the efficiency of the EM algorithm.

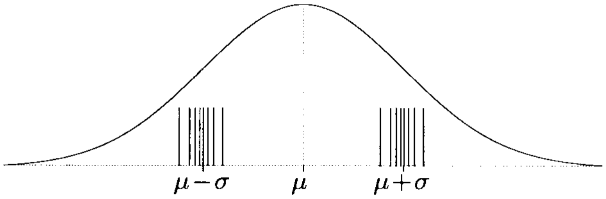

Fig. 1. Wrong fitting: the EM algorithm tries to fit the input samples (vertical bars) to a Gaussian kernel, while the samples actually follow a bimodal distribution.

On the other hand, for the estimation of the number of mixing kernels we cannot use the maximum likelihood method. Maximizing the likelihood with respect to the number of kernels leads to using one kernel for each input sample of X, with the kernel mean equal to the sample. Apparently, such a solution would lead to a large number of kernels, equal to the cardinality of X, giving rise to overfitting.

In general, it appears that the EM algorithm for Gaussian mixtures suffers from the problems of local maxima and an unknown number of kernels. In order to solve these two problems, we need a measure of the quality of the approximation at any instant as an indicator of how well the model fits the data. A bad fitting would necessitate a change of the parameter vector, or even an increase of the dimensionality of the parameter space. In the following we touch these issues.

III. The Kurtosis-Based Algorithm for Gaussian Mixture Modeling

A. Total Kurtosis Measure

Assuming that the random variable x follows a Gaussian mixture with a predefined number of kernels K, the EM algorithm yields the iterative update equations (7)–(9) for the estimation of the parameters of the mixture. In these equations, the posterior probability (10) specifies the probability that a sample x_i statistically belongs to a kernel j, thus each input sample can be regarded as originating from one of the kernels j with probability $P(j|x_i)$.

Based on this quantity, the fitting procedure tries to optimally distribute the K given kernels over the input space, in such a way that most of the details of the unknown density are correctly captured. However, in cases where the number of kernels is not adequate for accurately approximating the input density in some parts, the fitting algorithm, using a limited number of kernels, will unavoidably underestimate the density in those parts and give poor results. It turns out that although the parameters of the kernels, i.e., means and variances, are correctly estimated from the weighted sample moments shown in (8) and (9), it is probable that the fit is not adequate. In Fig. 1 we show such a case: the input samples (vertical bars) follow a bimodal distribution while EM tries to fit them to a single Gaussian kernel. In this case, there is no way to use the first two moments, i.e., mean and variance, to reveal this hidden multimodality.

A solution to the problem is to resort to higher moments in order to decide whether a kernel j fits adequately the samples lying in its vicinity. Under the assumption that $p(x|j)$ in (1) is the Gaussian density (2), it is not difficult to verify that the following equation holds irrespective of the values of μ_j and σ_j

$$\int_{-\infty}^{\infty} \left(\frac{x-\mu_j}{\sigma_j}\right)^4 p(x|j)\,dx = 3. \tag{11}$$

Using Bayes' rule we can express $p(x|j)$ in the above formula in terms of the mixture $p(x)$

$$p(x|j) = \frac{P(j|x)}{\pi_j}\,p(x) \tag{12}$$

and (11) becomes

$$\int_{-\infty}^{\infty} \left(\frac{x-\mu_j}{\sigma_j}\right)^4 \frac{P(j|x)}{\pi_j}\,p(x)\,dx = 3. \tag{13}$$

The left part of this equation can be approximated by Monte Carlo integration [11], [15] from the training data $x_i, i = 1, \cdots, n$, which are independently sampled from $p(x)$ to give

$$\frac{1}{n\pi_j}\sum_{i=1}^{n}\left(\frac{x_i-\mu_j}{\sigma_j}\right)^4 P(j|x_i) = 3. \tag{14}$$

To make this result more intuitive, we can substitute in the above formula the mixing weights π_j computed in each step of the EM algorithm from (7) to arrive at the result

$$\frac{\sum_{i=1}^{n}\left(\frac{x_i-\mu_j}{\sigma_j}\right)^4 P(j|x_i)}{\sum_{i=1}^{n} P(j|x_i)} = 3. \tag{15}$$

Based on that, we define the *weighted kurtosis* κ_j of a kernel j of the mixture as

$$\kappa_j = \frac{\sum_{i=1}^{n}\left(\frac{x_i-\mu_j}{\sigma_j}\right)^4 P(j|x_i)}{\sum_{i=1}^{n} P(j|x_i)} - 3 \tag{16}$$

which, for a true Gaussian mixture, should be zero for all components, and which can be regarded as the "weighted" equivalent of the original definition of the kurtosis of a distribution [15]

$$\text{Kurt} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i-\mu}{\sigma}\right)^4 - 3 \tag{17}$$

where μ and σ the sample mean and variance, respectively, of the random variable x.

For a mixture of well-separated kernels where the posteriors $P(j|x_i)$ for a kernel j are almost one for samples belonging to the correct kernel and almost zero for distant samples, the weighted kurtosis κ_j approximates the original kurtosis measure (17). On the other hand, if for some kernel j the distribution of the samples in its vicinity is non-Gaussian, the associated weighted kurtosis κ_j deviates from zero to a positive or negative number.

To test how large this deviation is for the whole mixture, a new measure is needed that weighs the deviation κ_j of each kernel according to its importance π_j for the whole mixture. In this sense, we define a new quantity called *total kurtosis* as

$$K_T = \sum_{j=1}^{K} \pi_j |\kappa_j| \tag{18}$$

which is a weighted average of the individual weighted kurtoses of the kernels of the mixture. The absolute values are needed to compensate for the individual kurtoses taking positive or negative values.

The total kurtosis K_T can be regarded as a measure of how well a Gaussian mixture fits the data, since a low value (near zero) indicates that each individual kernel fits naturally the samples in its vicinity, therefore the mixture constitutes a good approximation to the unknown density that generated the samples. On the other hand, a large value of the total kurtosis means that there are kernels that do not fit adequately their corresponding samples, or their parameters (mean and variance) are not properly adjusted.

The measure of kurtosis is important since the value of the likelihood alone does not provide much information regarding the effectiveness of the fit. For example, by using the EM algorithm we arrive at a solution of maximum likelihood for a specific number of kernels, but we cannot be sure whether the solution constitutes an acceptable approximation to the unknown density; the two densities may differ significantly based on other distance measures [16]. On the other hand, we know that a lower bound for the total kurtosis is the zero value. Therefore, we can expect that the lower the total kurtosis value of the obtained solution is, the better is the approximation of the unknown density.

As an example, consider the approximation of two unknown densities using a Gaussian mixture with $K = 10$ kernels. The first density was a Gaussian mixture with four components, while the second was a uniform density. The maximization of the likelihood provided solutions with log-likelihood values $-12\,202$ and -11732, respectively. These values contain no information of how well the obtained solutions approximate the corresponding densities. As expected, the first solution accurately approximated the known Gaussian mixture, while the second solution was only a coarse approximation to the uniform density. The total kurtosis of the solutions was 0.03 and 0.31, respectively, for the two cases. This means that the total kurtosis value revealed that the first approximation was accurate, while the second approximation was coarse. In general, solutions of lower total kurtosis, provide better fit to the samples compared to solutions with higher total kurtosis. In the following section we propose an algorithm that is based on EM and which automatically increases the number of kernels based on the value of the total kurtosis of the mixture.

B. The Proposed Algorithm

Based on the definition of the total kurtosis (18), we have developed a new algorithm for Gaussian mixture density estimation that uses the EM algorithm for parameter estimation and automatically adjusts the number of kernels using criteria based on the total kurtosis of the mixture. The proposed algorithm is based on the idea that we should try to maximize the likelihood by performing EM steps that in general lead to a decrease of the total kurtosis value.

More specifically, we start with a small number K of kernels (usually K is selected from one to three) and perform EM steps using the K kernels. These EM steps adjust the parameters of the kernels so that the likelihood is increasing and the total kurtosis is decreasing. This procedure is continued until either a local maximum of the likelihood is encountered or the total kurtosis reaches a minimum value and starts increasing. We distinguish between these two cases. In the first case a local maximum of the likelihood is also a local minimum of the total kurtosis, since no further update of the parameters is possible. If this happens, we check the value of the total kurtosis and, if it is sufficient low, we accept the solution, otherwise we consider that the solution is inadequate. Then, using the current local maximum parameters of the kernels, we create a new initial point for the EM algorithm by splitting one of the kernels in two as it will be described later in this section.

In the second case where the total kurtosis starts increasing without the likelihood having reached yet a local maximum, we consider that the EM algorithm has made its best in trying to fit each Gaussian to the given samples. Consequently, more kernels are needed to approximate the samples better. Therefore, if an EM step leads to an increase in the value of the total kurtosis, this is considered as the event that triggers the split of a kernel in two kernels in order to provide the capability for better approximation of the unknown density by the Gaussian mixture. After splitting, the $K+1$ kernels continue to be updated at each step using the EM algorithm. In general, the splitting of the kernels leads to a decrease in the value of total kurtosis. Nevertheless, in some cases it is possible that, due to improper initialization of the two new kernels, the value of the kurtosis temporarily increases for some steps, until the kernels move to the right positions, and the kurtosis starts decreasing again. For this reason, for a number of EM steps after a split, no further splits are permitted, since we allow the kernels to move to their appropriate positions even if this temporarily leads to an increase of the total kurtosis.

Once we have decided when to perform kernel splitting by monitoring the value of the total kurtosis after each EM step, it remains to specify which kernel will be selected for splitting.

A deviation of the total kurtosis K_T from zero implies that the weighted kurtosis κ of one or more kernels deviates also from the zero value. Therefore, a reasonable selection criterion is to split the kernel j that contributes most significantly to the high value of the total kurtosis, i.e., we select the kernel with the highest value of $\pi_j|\kappa_j|$. The two new kernels that are created have means equal to $\mu_j + \sigma_j$ and $\mu_j - \sigma_j$ respectively, and variances both equal to σ_j. Their priors are set to $\pi_j/2$ so that (3) still holds.

Finally, we must also specify termination criteria for the proposed method. Since the EM algorithm converges for fixed number of kernels, we must specify criteria for disabling kernel splitting in future steps. Consequently, the EM algorithm will converge to a maximum value of the likelihood. A criterion of this kind is a measure of the effectiveness of the split: we store the value of the total kurtosis at the time of a split and the corresponding value at the time of the next split. If the difference is very small, we consider that splitting is no longer effective and from that time on, we keep the number of kernels fixed and perform EM steps until reaching a local maximum of the likelihood.

The algorithm just described that dynamically adds kernels based on the value of the total kurtosis, has the attractive feature that it requires no initial knowledge about the number K of the kernels of the mixture. It starts using a small number of kernels and adds kernels in the mixture dynamically, while the algorithm evolves. Moreover, it requires no initialization at a point θ which is already near the optimum solution, while, by dynamically increasing the number of kernels, it is also capable of potentially escaping from local maxima of the likelihood function, thus yielding a better approximation to the unknown density.

The complete algorithm is summarized below. In the following description ϵ_1, ϵ_2 are user defined variables, $limit$ denotes the number of steps after the last splitting during which a new splitting is not allowed, $nosteps$ denotes the number of steps after the last splitting and K_{split} denotes the total kurtosis value at the time of the last split. Moreover if the variable $enableSplitting$ is set equal to 1 then no further kernel splitting is allowed.

```
1) Initialization: Set the initial number $K$ of kernels and initialize
   the parameters of the kernels (means and variances).
2) Compute the initial value of the total kurtosis $K_T^{old}$ and the
   initial value of the likelihood $L^{old}$.
3) Set $nosteps := 0$, $K_{split} := K_T^{old}$, $enableSplitting := 1$.
   a) Perform an EM step. Set $nosteps := nosteps + 1$.
   b) Compute the new value of the total kurtosis $K_T^{new}$ and
      the new value of the likelihood $L^{new}$.
   c) Check for convergence: if $|L^{new} - L^{old}| < \epsilon_1$ go to
      step (f).
   d) If $(K_T^{new} > K_T^{old})$ and $(nosteps > limit)$ and
      $(enableSplitting = 1)$ then
        • Perform kernel splitting.
        • $K := K + 1$, $nosteps := 0$.
        • $K_T^{old} := K_T^{new}$, $L^{old} := L^{new}$
        • Compute $K_T^{new}$, $L^{new}$
        • If $|K_{split} - K_T^{new}| < \epsilon_2$ then $enableSplitting :=$
          $0$.
        • Set $K_{split} := K_T^{new}$
   e) Go to step (a)
   f) If ($K_T^{new}$ is not small enough) and $(enableSplitting =$
      $1)$ then
        • Perform kernel splitting.
        • $K := K + 1$, $nosteps := 0$.
        • $K_T^{old} := K_T^{new}$, $L^{old} := L^{new}$
        • Compute $K_T^{new}$, $L^{new}$
        • If $|K_{split} - K_T^{new}| < \epsilon_2$ then $enableSplitting :=$
          $0$.
        • Set $K_{split} := K_T^{new}$
        • Go to step (a)
4) end.
```

IV. Examples

To assess the effectiveness of our approach we have conducted experiments with data drawn independently from known distributions, which in turn we tried to approximate using our algorithm and the conventional EM algorithm. After training, we tested the accuracy of the obtained approximations with respect to the true densities.

In every problem considered, we have created a data set of $n = 5000$ points drawn independently from the corresponding density to be approximated. In all experiments the EM algorithm started with the means of the K kernels being uniformly distributed within the range of the data, while the deviance σ of each kernel was set equal to 0.5. On the other hand, our algorithm always started with $K = 1$ kernel with mean in the center of the data range and σ also equal to 0.5.

We have considered three one-dimensional problems:

1) a Gaussian mixture density with four kernels;
2) a Gaussian mixture density with five kernels;
3) a density with two Gaussian and two uniform kernels.

In all experiments, since the original density $g(x)$ is known, we could compute the theoretically optimal log-likelihood $\tilde{L}$ for the given set of samples x_i, $i = 1, \cdots, n$, drawn from the respective density

$$\tilde{L} = \sum_{i=1}^{n} \log g(x_i). \tag{19}$$

Example 1: In this experiment we have generated samples using the following Gaussian mixture density

$$\begin{aligned} g(x) = & 0.25N(-7, 0.5) + 0.25N(-3, 1) \\ & + 0.25N(3, 1) + 0.25N(7, 0.5) \end{aligned} \tag{20}$$

where $N(\mu, \sigma)$ is the normal distribution with mean μ and standard deviation σ. The value of the theoretical log-likelihood was $\tilde{L} = -12\,201$.

Fig. 2 displays the original density $g(x)$ as well as the obtained solutions using our approach and the EM algorithm. The EM algorithm was applied on $K = 4$ kernels and provided accurate solution

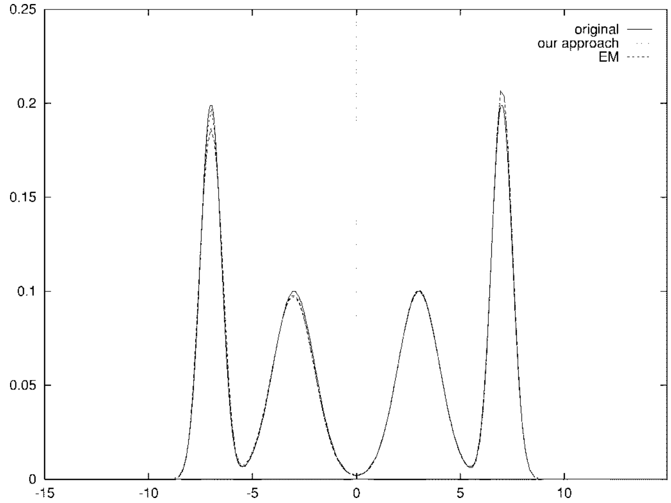

Fig. 2. Approximation of a Gaussian mixture density with four kernels.

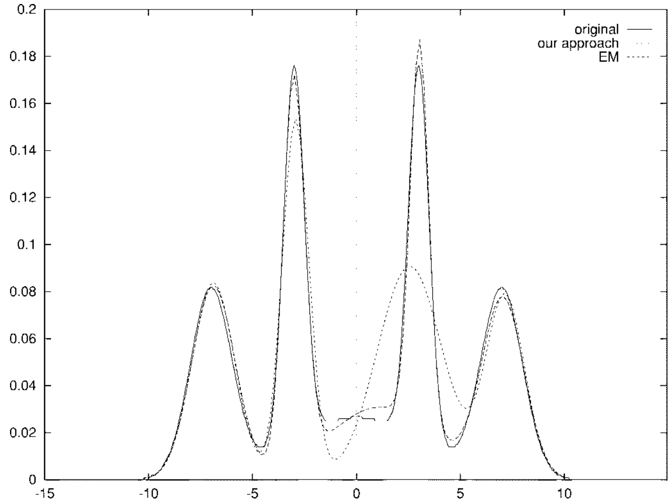

Fig. 3. Approximation of a Gaussian mixture density with five kernels.

with $L = -12\,201.4$ and total kurtosis $K_T = 0.03$. Our algorithm was able to identify the correct number of kernels and provided an accurate solution with $K = 4$ kernels having $L = -12\,201.9$ and $K_T = 0.033$.

Example 2: In this experiment we have generated samples using the following density:

$$\begin{aligned} g(x) = &0.2N(-7,\, 1) + 0.2N(-3,\, 0.5) + 0.2N(0,\, 3) \\ &+ 0.2N(3,\, 0.5) + 0.2N(7,\, 1). \end{aligned} \tag{21}$$

The value of the theoretical log-likelihood was $\tilde{L} = -13\,382.6$.

Fig. 3 displays the original density $g(x)$ as well as the obtained solutions using our approach and the EM algorithm. The EM algorithm was applied on $K = 5$ kernels and was stuck in a local maximum with $L = -13\,718$ and total kurtosis $K_T = 0.35$. On the contrary, our algorithm provided a much better solution with $K = 6$ kernels having $L = -13\,385$ and $K_T = 0.062$.

Example 3: Finally we have conducted experiments with the following density consisting of two Gaussian and two uniform kernels

$$\begin{aligned} g(x) = &0.25N(-7,\, 0.5) + 0.25U(-3,\, -1) \\ &+ 0.25U(1,\, 3) + 0.25N(7,\, 0.5) \end{aligned} \tag{22}$$

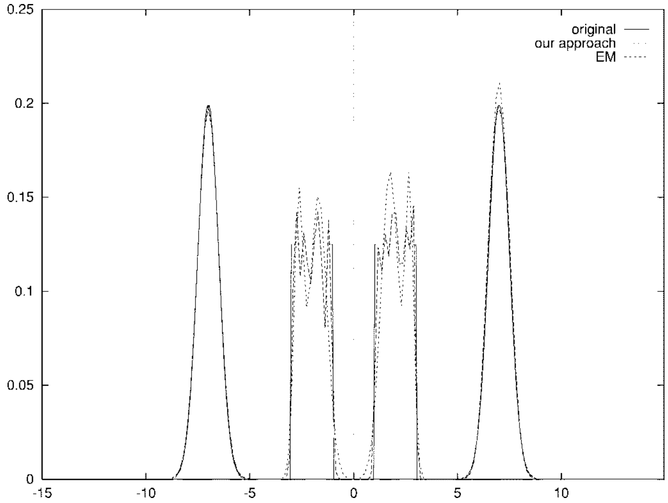

Fig. 4. Approximation of a mixture with two Gaussian and two uniform kernels.

where $U(a,\, b)$ denotes the uniform density in $[a,\, b]$. The value of the theoretical log-likelihood was $\tilde{L} = -10\,492$.

The obtained solutions are shown in Fig. 4. Our algorithm provided a solution with $K = 12$ kernels having $L = -10\,577$ and total kurtosis $K_T = 0.15$. The EM algorithm was also tested with $K = 12$ kernels, but again the obtained solution was worse compared to ours: the log-likelihood value was $L = -10\,730$ and the value of the total kurtosis was $K_T = 0.31$.

As a conclusion, we can state that the dynamic allocation of new kernels which is guided by monitoring the value of the total kurtosis makes the proposed algorithm an efficient method for Gaussian mixture density estimation that yields considerable improvement over the classical EM algorithm. Moreover, as shown in examples 1 and 2, our algorithm has the ability to approximately identify the number of kernels of an unknown Gaussian mixture density, which is of major importance in many applications.

V. Conclusion

We proposed a new method for Gaussian mixture modeling which is based on the EM algorithm, and which dynamically adjusts the number of the kernels of the mixture. We defined a new quantity, called total kurtosis, to be used in the algorithm as an indicator of how well a Gaussian mixture fits the data. The algorithm performs EM steps and updates the parameters of the mixture, while at the same time monitors the value of the total kurtosis and increases the number of kernels in the case where this quantity starts increasing. The increase of the number of kernels is performed through splitting of the kernel that contributes more significantly to the value of the total kurtosis. In this sense the proposed algorithm proceeds by performing both likelihood maximization and kurtosis minimization. The increase in the number of kernels stops when no further progress in the minimization of the kurtosis seems possible. Experimental results on several test problems indicate that our approach constitutes a promising alternative to the EM algorithm.

In this work we have examined the univariate case. Current work focuses on a multidimensional definition of the weighted kurtosis (and the total kurtosis) and the application of the proposed algorithm to density estimation problems of higher dimensionality. Moreover, we aim at testing the effectiveness of the algorithm on several applications where an EM approach has already been employed, e.g., classification, time series, etc.

References

[1] D. M. Titterington, A. F. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions.* New York: Wiley, 1985.

[2] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Rev.,* vol. 26, pp. 195–239, Apr. 1984.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc. B,* vol. 39, pp. 1–38, 1977.

[4] N. Vlassis, G. Papakonstantinou, and P. Tsanakas, "Mixture density estimation based on maximum likelihood and test statistics," *Neural Process. Lett.,* vol. 9, pp. 63–76, Feb. 1999.

[5] D. F. Specht, "Probabilistic neural networks," *Neural Networks,* vol. 3, pp. 109–118, 1990.

[6] E. Parzen, "On the estimation of a probability density function and mode," *Ann. Math. Statist.,* vol. 33, pp. 1065–1076, 1962.

[7] H. G. C. Tråvén, "A neural network approach to statistical pattern classification by 'semiparametric' estimation of probability density functions," *IEEE Trans. Neural Networks,* vol. 2, pp. 366–377, May 1991.

[8] R. L. Streit and T. E. Luginbuhl, "Maximum likelihood training of probabilistic neural networks," *IEEE Trans. Neural Networks,* vol. 5, no. 5, pp. 764–783, 1994.

[9] S. Shimoji, "Self-organizing neural networks based on Gaussian mixture model for PDF estimation and pattern classification," Ph.D. dissertation, Univ. Southern California, Los Angeles, 1994.

[10] N. Vlassis, A. Dimopoulos, and G. Papakonstantinou, "The probabilistic growing cell structures algorithm," in *Proc. ICANN'97, Int. Conf. Artificial Neural Networks,* Lausanne, Switzerland, Oct. 1997, pp. 649–654.

[11] B. D. Ripley, *Pattern Recognition and Neural Networks.* Cambridge, U.K.: Cambridge Univ. Press, 1996.

[12] G. J. McLachlan, "On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture," *Appl. Statist.,* vol. 36, pp. 318–324, 1987.

[13] W. D. Furman and B. G. Lindsay, "Testing for the number of components in a mixture of normal distributions using moment estimators," *Comput. Statist. Data Anal.,* vol. 17, pp. 473–492, 1994.

[14] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions.* New York: Wiley, 1997.

[15] W. H. Press, S. A. Teukolsky, B. P. Flannery, and W. T. Vetterling, *Numerical Recipes in C,* 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[16] S. Ingrassia, "A comparison between the simulated annealing and the EM algorithms in normal mixture decompositions," *Stat. Comput.,* vol. 2, pp. 203–211, 1992.

Analysis of Modularly Composed Nets by Siphons

MuDer Jeng and Xiaolan Xie

***Abstract*—This paper uses siphons to analyze the class of Petri nets constructed by a modular approach in [5] for modeling manufacturing systems with shared resources. A resource point of view is taken. First the behavior of each resource is modeled using resource control nets, strongly connected state machines with one place being marked initially. Interactions among the resources are modeled through merging of common transition subnets. This paper provides conditions, expressed in terms of siphons, under which reversibility and liveness of the integrated model are obtained. Relations between siphons and circular-wait are formally established. Superiority of the siphon-based analysis over a previous analysis using circular wait is shown.**

***Index Terms*—Analysis, Petri nets, synthesis.**

I. Introduction

Modular approach is an efficient way to cope with the complexity in modeling a large-scale system. It consists in decomposing it into simple subsystems called modules, modeling each module and integrating the module models together to obtain the model of the whole system.

A major concern, when modeling a real-life system, is to check whether the Petri net model has desired qualitative properties such as liveness, boundedness, and reversibility. As long as manufacturing systems are concerned, the liveness ensures that blocking will never occur, the boundedness guarantees that the number of in-process parts is bounded, the reversibility enables the system to come back to its initial state from whatever state it reaches.

Due to the complexity of real-life systems, classical property checking methods such as coverability tree, invariant analysis and algebraic analysis (see [10]) hardly apply. There are two classes of methods for analyzing a large Petri net model. The first one is the reduction of Petri nets while preserving properties. Reduction rules have been proposed [2], [9]. The main disadvantage of this approach lies in the difficulty of finding reducible subnets.

The second class of methods includes synthesis methods which build the models systematically and progressively such that the desired properties are preserved all along the design process. Two synthesis approaches: top-down approach and bottom-up approach, have been proposed.

The top-down approach begins with an aggregate model of the system which is refined progressively to introduce more and more details. The basic refinement is the substitution of a place or a transition by a so-called well-formed block [12], [13], [15]. Conditions, under which the desired properties are preserved, are given. This approach is well suited to model systems composed of almost independent sub-systems. However, this approach loses its efficiency in case of strongly coupled sub-systems since it is impossible to find small aggregate models.

The bottom-up approach [1], [4], [5], [7], [8], [11], [14] starts from sub-system models and integrate them by merging some places and/or transitions. The disadvantage of the general bottom-up approach lies

Manuscript received June 26, 1998; revised April 1, 1999.

M. Jeng is with the Department of Electrical Engineering, National Taiwan Ocean University, Keelung, Taiwan, R.O.C. (e-mail: b0162@ind.ntou.edu.tw).

X. Xie is with INRIA/MACSI Team, ENIM-Ile du Saulcy, 57045 Metz, France (e-mail: xie@loria.fr).

Publisher Item Identifier S 1083-4427(99)05343-6.