Incremental Learning of Temporally-Coherent Gaussian Mixture Models

Ognjen Arandjelović Roberto Cipolla
Department of Engineering, Cambridge, CB2 1PZ, UK
{oa214,cipolla}@eng.cam.ac.uk

Abstract

In this paper we address the problem of learning Gaussian Mixture Models (GMMs) incrementally. *Unlike previous approaches which universally assume that new data comes in blocks representable by GMMs which are then merged with the current model estimate, our method works for the case when novel data points arrive* one-by-one*, while requiring little additional memory. We keep only two GMMs in the memory and no historical data. The current fit is updated with the assumption that the number of components is fixed, which is increased (or reduced) when enough evidence for a new component is seen. This is deduced from the change from the oldest fit of the same complexity, termed the* Historical GMM*, the concept of which is central to our method. The performance of the proposed method is demonstrated qualitatively and quantitatively on several synthetic data sets and video sequences of faces acquired in realistic imaging conditions.*

1 Introduction

The Gaussian Mixture Model (GMM) is a semi-parametric method for high-dimensional density estimation. It is used widely across different research fields, with applications to computer vision ranging from object recognition [3], shape [2] and face appearance modelling [7] to colour-based tracking and segmentation [11], to name just a few. It is worth emphasizing the key reasons for its practical appeal: (i) its flexibility allows for the modelling of complex and nonlinear pattern variations [7], (ii) it is simple and efficient in terms of memory, (iii) a principled model complexity selection is possible, and (iv) there are theoretically guaranteed to converge algorithms for model parameter estimation.

Virtually all previous work with GMMs has concentrated on non time critical applications, typically in which model fitting (i.e. model parameter estimation) is performed offline, or using a relatively small training corpus. On the other hand, the recent trend in computer vision is oriented towards real-time applications (for example for human-computer interaction and on-the-fly model building) and modelling of increasingly complex patterns which inherently involves large amounts of data. In both cases, the usual batch fitting becomes impractical and an incremental learning approach is necessary.

Problem challenges Incremental learning of GMMs is a surprisingly difficult task. One of the main challenges of this problem is the model complexity selection which is required to be dynamic by the very nature of the incremental learning framework. Intuitively, if all information that is available at any time is the current GMM estimate, a single novel point *never* carries enough information to cause an increase in the number of Gaussian components. Another closely related difficulty lies in the *order* in which new data arrives

BMVC 2005 doi:10.5244/C.19.59

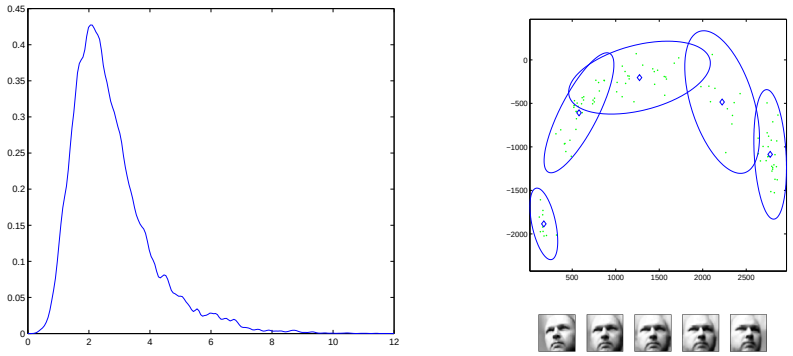
[9]. If successive data points are always badly correlated, then a large amount of data has to be kept in memory if accurate model order update is to be achieved.

1.1 Related Previous Work

The most common way of fitting a GMM is using the Expectation-Maximization (EM) algorithm [4]. Starting from an estimate of model parameters, soft membership of data is computed (the *Expectation* step) which is then used to update the parameters in the maximal likelihood (ML) manner (the *Maximization* step). This is repeated until convergence, which is theoretically guaranteed. In practice, initialization is frequently performed using the K-means clustering algorithm [1, 5].

Incremental approaches Incremental fitting of GMMs has already been addressed in the machine learning literature. Unlike the proposed method, most of the existing methods assume that novel data arrives in *blocks* as opposed to a single datum at a time. Hall *et al.* [8] merge Gaussian components in a pair-wise manner by considering volumes of the corresponding hyperellipsoids. A more principled method was recently proposed by Song and Wang [14] who use the W statistic for covariance and the Hotelling's T^2 statistic for mean equivalence. However, they do not fully exploit the available probabilistic information by failing to take into account the *evidence* for each component at the time of merging. Common to both [8] and [14] is the failure to make use of the existing model when the GMM corresponding to new data is fitted. What this means is that even if some of the new data is already explained well by the current model, the EM fitting will try to explain it in the context of other novel data, affecting the accuracy of the fit as well as the subsequent component merging. The method of Hicks *et al.* [10] (also see [9]) does not suffer from the same drawback. The authors propose to first "concatenate" two GMMs and then determine the optimal model order by considering models of *all* low complexities and choosing the one that gives the largest penalized log-likelihood. A similar approach of combining Gaussian components was also described by Vasconcelos and Lippman [15].

Model order selection Broadly speaking, there are three classes of approaches for GMM model order selection: (i) EM-based using validation data, (ii) EM-based using model validity criteria, and (iii) dynamic algorithms. The first approach involves random partitioning of the data to training and validation sets. Model parameters are then iteratively estimated from training data and the complexity that maximizes the posterior of the validation set is sought. This method is typically less preferred than methods of the other two groups, being wasteful both of the data and computation time. The most popular group of methods is EM-based and uses the posterior of all data, penalized with model complexity. Amongst the most popular are the Minimal Description Length (MDL) [12], Bayesian Information (BIC) [13] and Minimal Message Length (MML) [19] criteria. Finally, there are methods which combine the fitting procedure with dynamic model order selection. Briefly, Zwolinski and Yang [20], and Figueredo and Jain [6] overestimate the complexity of the model and reduce it by discarding "improbable" components. Vlassis and Likas [18] use a weighted sample kurtoses of Gaussian kernels, while Verbeek *et al.* introduce a heuristic greedy approach in which mixture components are added one at the time [16].



(a) (b)

Figure 1: ***TC-GMMs:*** *(a) Average distribution of Euclidean distance between temporally consecutive faces across video sequences of faces in unconstrained motion. The distribution peaks at a low, but greater-than-zero distance, which is typical of Temporally-Coherent GMMs analyzed in this paper. Both too low and too large distances are infrequent, in this case the former due to the time gap between the acquisition of consecutive video frames, the latter due to the smoothness of face shape and texture. (b) A typical sequence projected to the first three principal components estimated from the data, the corresponding MDL EM fit and the component centres visualized as images. On average, we found that over 80% of pairs of successive faces have the highest likelihood of having been generated by the same Gaussian component.*

2 Incremental GMM Estimation

A GMM with M components in a D-dimensional embedding space is defined as:

$$\mathscr{G}(\mathbf{x};\theta) = \sum_{j=1}^{M} \alpha_j \mathscr{N}(\mathbf{x};\mu_j,\mathbf{C}_j) \tag{1}$$

where $\theta = (\{\alpha_i\},\{\mu_i\}\{\mathbf{C}_i\})$ is the set of model parameters, α_i being the prior of the i-th Gaussian component with the mean μ_i and covariance $\mathbf{C}_i$:

$$\mathscr{N}(\mathbf{x};\mu,\mathbf{C}) = \frac{1}{(2\pi)^{D/2}\sqrt{|\mathbf{C}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{C}^{-1}(\mathbf{x}-\mu)\right) \tag{2}$$

2.1 Temporally-Coherent GMMs

We assume *temporal coherence* on the order in which data points are seen. Let $\{\mathbf{x}_t\} \equiv \{\mathbf{x}_0,\ldots,\mathbf{x}_T\}$ be a stream of data, its temporal ordering implied by the subscript. The assumption of an underlying Temporally-Coherent GMM (TC-GMM) on $\{\mathbf{x}_t\}$ is:

$$\begin{aligned} \mathbf{x}_0 &\sim \mathscr{G}(\mathbf{x};\theta) \\ \mathbf{x}_{t+1} &\sim p_S(\|\mathbf{x}_{t+1}-\mathbf{x}_t\|)\cdot\mathscr{G}(\mathbf{x};\theta) \end{aligned}$$

where p_S is a unimodal density. Intuitively, while data is distributed according to an underlying Gaussian mixture, it is also expected to vary smoothly with time, see Figure 1.

Algorithm 1 Incremental TC-GMM

Input: current GMM $\mathscr{G}_N(\mathbf{x};\theta_1)$,
historical GMM $\mathscr{G}_N^{(h)}(\mathbf{x};\theta^{(h)})$,
novel observation $\mathbf{x}$.

Output: updated GMM $\mathscr{G}_M(\mathbf{x};\theta_2)$.

1. **Fixed-complexity update:** update$(\mathscr{G}_N,\mathbf{x})$
2. **Model splitting:** $\mathscr{G}_M$ = split-all$(\mathscr{G}_N,\mathscr{G}_N^{(h)})$
3. **Pair-wise component merging:**

   for all $(i,j)\in(1..N,1..N)$

4. **Expected description length:**

   $[L_1,L_2]$ =DL$\{$merge$(\mathscr{G}_M,i,j)$, split$(\mathscr{G}_M,i,j)\}$

5. **Complexity update**

   $\mathscr{G}_M = L_1 < L_2\ ?$
   merge$(\mathscr{G}_M,i,j)$: split$(\mathscr{G}_M,i,j)$

6. **end**

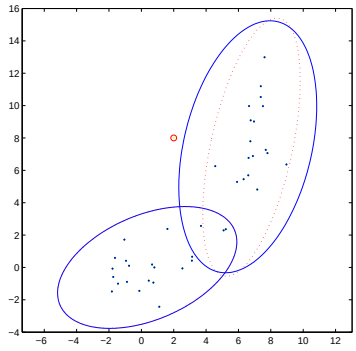


Figure 2: ***Fixed complexity update:*** *The mean and the covariance of each Gaussian component are updated according to the probability that it generated the novel observation (red circle). Old covariances are shown as dashed, the updated ones as solid ellipses corresponding to component parameters, while historical data points are displayed as blue dots.*

2.2 Method Overview

The proposed method consists of a three-stage model update each time a new data point becomes available, see Algorithm 1. At each time step: (i) model parameters are updated under the constraint of fixed complexity, (ii) new Gaussian components are postulated by model splitting and (iii) components are merged to minimize the expected model description length. We keep in memory only two GMMs and *no historical data*. One is the current GMM estimate, while the other is the oldest model od the same complexity after which no permanent new cluster creation took place – we term this the *Historical GMM*.

2.3 GMM Update for Fixed Complexity

In the first stage of our algorithm, the current GMM $\mathscr{G}(\mathbf{x};\theta)$ is updated under the *constraint of fixed model complexity*, i.e. fixed number of Gaussian components. We start with the assumption that the current model parameters are estimated in the ML fashion in a local minimum of the EM algorithm:

$$\alpha_i = \frac{\sum_j p(i|\mathbf{x}_j)}{N} \qquad \mu_i = \frac{\sum_j \mathbf{x}_j p(i|\mathbf{x}_j)}{\sum_j p(i|\mathbf{x}_j)} \qquad \mathbf{C}_i = \frac{\sum_j (\mathbf{x}_j-\mu_i)(\mathbf{x}_j-\mu_i)^T p(i|\mathbf{x}_j)}{\sum_j p(i|\mathbf{x}_j)} \tag{3}$$

where $p(i|\mathbf{x}_j)$ is the probability of the i-th component conditioned on data point $\mathbf{x}_j$. Similarly, for the updated set of GMM parameters θ^* it holds:

$$\alpha_i^* = \frac{\sum_j p^*(i|\mathbf{x}_j) + p^*(i|\mathbf{x})}{N+1} \qquad \mu_i^* = \frac{\sum_j \mathbf{x}_j p^*(i|\mathbf{x}_j) + \mathbf{x} p^*(i|\mathbf{x})}{\sum_j p^*(i|\mathbf{x}_j) + p^*(i|\mathbf{x})} \tag{4}$$

$$\mathbf{C}_i^* = \frac{\sum_j (\mathbf{x}_j - \mu_i^*)(\mathbf{x}_j - \mu_i^*)^T p^*(i|\mathbf{x}_j) + (\mathbf{x} - \mu_i^*)(\mathbf{x} - \mu_i^*)^T p^*(i|\mathbf{x})}{\sum_j p^*(i|\mathbf{x}_j) + p^*(i|\mathbf{x})} \tag{5}$$

The key problem is that the probability of each component conditioned on the data changes *even for historical data* $\{\mathbf{x}_j\}$. In general, the change in conditional probabilities can be arbitrarily large as the novel observation $\mathbf{x}$ can lie anywhere in the $\mathbb{R}^D$ space. However, the expected correlation between temporally close points, governed by the underlying TC-GMM model allows us to make the assumption that component likelihoods do not change much with the inclusion of novel information in the model:

$$p^*(i|\mathbf{x}_j) = p(i|\mathbf{x}_j) \tag{6}$$

This assumption is further justified by the two stages of our algorithm that follow (Sections 2.4 and 2.5) – a large change in probabilities $p(i|\mathbf{x}_j)$ occurs only when novel data is not well explained by the current model. When enough evidence for a new Gaussian components is seen, model complexity is increased, while old component parameters switch back to their original value. Using (6), a simple algebraic manipulation of (3)-(4), omitted for clarity, and writing $\sum_j p(i|\mathbf{x}_j) \equiv E_i$, leads to the following:

$$\alpha_i^* = \frac{E_i + p(i|\mathbf{x})}{N+1} \qquad \mu_i^* = \frac{\mu_i E_i + \mathbf{x} p(i|\mathbf{x})}{E_i + p(i|\mathbf{x})} \tag{7}$$

$$\mathbf{C}_i^* = \frac{(\mathbf{C}_i + \mu_i \mu_i^T - \mu_i \mu_i^{*T} - \mu_i^* \mu_i^T + \mu_i^* \mu_i^{*T}) E_i + (\mathbf{x} - \mu_i^*)(\mathbf{x} - \mu_i^*)^T p(i|\mathbf{x})}{E_i + p(i|\mathbf{x})} \tag{8}$$

It can be seen that the update equations depend only on the parameters of the old model and the sum of component likelihoods, but *no historical data*. Therefore the additional memory requirements are of the order $O(M)$, where M is the number of Gaussian components. Constant-complexity model parameter update is illustrated in Figure 2.2.

2.4 Model Splitting

One of the greatest challenges of incremental GMM learning is the dynamic model order selection. In the second stage of our algorithm, new Gaussian clusters are postulated based on the parameters of the current parameter model estimate $\mathscr{G}$ and the *Historical GMM* $\mathscr{G}^{(h)}$, which is central to our idea. As, by definition, no permanent model order changes occurred between the Historical and the current GMMs, they have the same number of components and, importantly, the 1-1 correspondence between them is known (the current GMM is merely the Historical GMM that was updated under the constraint of fixed model complexity). Therefore, for each pair of corresponding components $(\mu_i, \mathbf{C}_i)$ and $(\mu_i^{(h)}, \mathbf{C}_i^{(h)})$ we compute the 'difference' component, see Figure 3 (a-c). Writing (3) for the Historical and the current GMMs, and using the assumption in (6) the i-th differ-

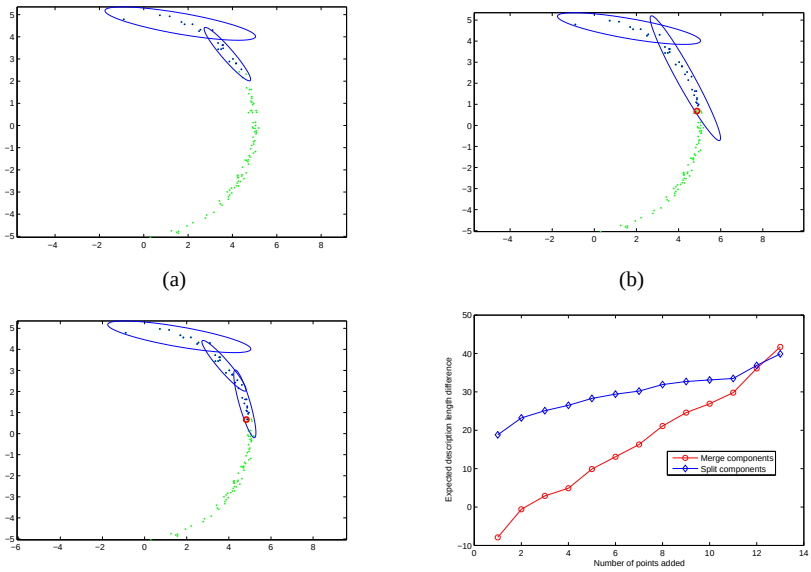


Figure 3: ***Dynamic model order selection:*** *(a) Historical GMM. (b) Current GMM before the arrival of novel data. (c) New data point (red circle) causes the splitting of a Gaussian component, resulting in a 3-component mixture. (d) The contribution to the expected model description length for merging and splitting of the component, as the number of novel data points is increased.*

ence component parameters become:

$$\alpha_i^{(n)} = \frac{E_i - E_i^{(h)}}{N - N^{(h)}} \qquad \mu_i^{(n)} = \frac{\mu_i E_i - \mu_i^{(h)} E_i^{(h)}}{E_i - E_i^{(h)}} \tag{9}$$

$$\mathbf{C}_i^{(n)} = \frac{\mathbf{C}_i E_i - (\mathbf{C}_i^{(h)} + \mu_i^{(h)} \mu_i^{(h)^T}) E_i^{(h)} + (\mu_i^{(h)} \mu_i^T + \mu_i \mu_i^{(h)^T}) E_i^{(h)} - \mu_i \mu_i^T E_i}{E_i - E_i^{(h)}} + \tag{10}$$
$$\mu_i^{(n)} \mu_i^T + \mu_i \mu_i^{(n)^T} - \mu_i^{(n)} \mu_i^{(n)^T}$$

2.5 Component Merging

In the proposed method, dynamic model complexity estimation is based on the MDL criterion. Briefly, MDL assigns to a model a cost related to the amount of information necessary to encode the model and the data *given* the model. This cost, known as the description length $L(\theta|\{\mathbf{x}_i\})$, is equal to the likelihood of the data under that model penalized by the model complexity, measured as the number of free parameters N_E:

$$L(\theta|\{\mathbf{x}_i\}) = \frac{1}{2} N_E \log_2(N) - P(\{\mathbf{x}_i\}|\theta) \tag{11}$$

In the case of an M-component GMM with full covariance matrices in $\mathbb{R}^D$ space, free parameters are $(M-1)$ for priors, MD for means and $MD(D+1)/2$ for covariances:

$$N_E = M - 1 + MD + M\frac{D(D+1)}{2} \tag{12}$$

The problem is that for the computation of $P(\{\mathbf{x}_i\}|\theta)$ historical data $\{\mathbf{x}_i\}$ is needed – which is unavailable. Instead of $P(\{\mathbf{x}_i\}|\theta)$, we propose to compute the *expected* likelihood of the same number of data points and, hence, use the expected description length as the model order selection criterion. Consider two components with the corresponding multivariate Gaussian densities $p_1(\mathbf{x}) \sim \mathcal{N}(\mathbf{x};\mu_1,\mathbf{C}_1)$ and $p_2(\mathbf{x}) \sim \mathcal{N}(\mathbf{x};\mu_2,\mathbf{C}_2)$. The expected likelihood of N_1 points drawn from the former and N_2 from the latter given model $\alpha_1 p_1(\mathbf{x}) + \alpha_2 p_2(\mathbf{x})$ is:

$$E\left[P(\{\mathbf{x}_j\}|\theta_S)\right] = \left(\int p_1(\mathbf{x})(\alpha_1 p_1(\mathbf{x}) + \alpha_2 p_2(\mathbf{x}))d\mathbf{x}\right)^{N_1} \left(\int p_2(\mathbf{x})(\alpha_1 p_1(\mathbf{x}) + \alpha_2 p_2(\mathbf{x}))d\mathbf{x}\right)^{N_2} \tag{13}$$

where integrals of the type $\int p_i(\mathbf{x})p_j(\mathbf{x})d\mathbf{x}$ are recognized as the Bhattacharyya distance, which is for Gaussian distributions easily computed as:

$$d_B(p_i,p_j) = \int p_i(\mathbf{x})p_j(\mathbf{x})d\mathbf{x} = \frac{\exp(-K/2)}{(2\pi)^{D/2}|\mathbf{C}_i\mathbf{C}_j\mathbf{C}|^{1/2}} \tag{14}$$

where:

$$\mathbf{C} = \left(\mathbf{C}_i^{-1} + \mathbf{C}_j^{-1}\right)^{-1} \quad \mu = \mathbf{C}(\mathbf{C}_i^{-1}\mu_i + \mathbf{C}_j^{-1}\mu_j) \quad K = \mu_i\mathbf{C}_i^{-1}\mu_i^T + \mu_j\mathbf{C}_j^{-1}\mu_j^T - \mu\mathbf{C}^{-1}\mu^T$$

On the other hand, consider the case when the two components are merged i.e. replaced by a single Gaussian component with the corresponding density $p(\mathbf{x})$. Then we compute the expected likelihood of N_1 points drawn from $p_1(\mathbf{x})$ and N_2 points drawn from $p_2(\mathbf{x})$, given model $p(\mathbf{x})$:

$$E\left[P(\{\mathbf{x}_j\}|\theta_M)\right] = \left(\int p(\mathbf{x})p_1(\mathbf{x})d\mathbf{x}\right)^{N_1} \cdot \left(\int p(\mathbf{x})p_2(\mathbf{x})d\mathbf{x}\right)^{N_2} \tag{15}$$

Substituting the expected evidence and model complexity in (11) we get:

$$\Delta E[L] = E[L_S] - E[L_M] = \frac{1}{4}D(D+1)\log_2(N_1+N_2) - E[P(\{\mathbf{x}_j\}|\theta_S)] + [P(\{\mathbf{x}_j\}|\theta_S)] \tag{16}$$

Then the condition for merging is simply $\Delta E[L] > 0$, see Figure 3 (d). Merging equations are virtually the same as (9) and (10) for model splitting, so we do not repeat them.

3 Empirical Evaluation

The proposed method was evaluated on several synthetic data sets and video sequences of faces in unconstrained motion, acquired in realistic imaging conditions and localized using the Viola-Jones face detector [17], see Figure 1 (b). Two synthetic data sets that we illustrate its performance on are:

1. 100 points generated from a Gaussian with a diagonal covariance matrix in radial coordinates: $r \sim \mathcal{N}(\bar{r} = 5, \sigma_r = 0.1)$, $\phi \sim \mathcal{N}(\bar{\phi} = 0, \sigma_\phi = 0.7)$
2. 80 points generated from a uniform distribution in x and a Gaussian noise perturbed sinusoid in y coordinate : $x \sim \mathcal{U}(\min x = 0, \max x = 10)$, $y \sim \mathcal{N}(\bar{y} = \sin x, \sigma_y = 0.1)$

Temporal ordering was imposed by starting from the data point with the minimal x coordinate and then iteratively choosing as the successor the nearest neighbour out of yet unused points. The initial GMM parameters, the final fitting results and the comparison with the MDL-EM fitting are shown in Figure 4. In the case of face motion video sequences, temporal ordering of data is inherent in the acquisition process. An interesting fitting example is shown and compared with the MDL-EM batch approach in Figure 5.

Qualitatively, both in the case of synthetic and face data it can be seen that our algorithm consistently produces meaningful GMM estimates. Quantitatively, the results are comparable with the widely accepted EM fitting with the underlying MDL criterion, as witnessed by the description lengths of the obtained models.

Failure modes On our data sets two types phenomena in data sometimes caused unsatisfactory fitting results. The first, one inherently problematic to our algorithm, is when newly available data is well explained by the Historical GMM. Referring back to Section 2.4, it can be seen in (9) and (10) that this data contributes to the confidence of creating a *new* GMM component whereas it should not. The second failure mode was observed when the assumption of temporal coherence (Section 2.1) was violated, e.g. when our face detector failed to detect faces in several consecutive video frames. While this cannot be considered an inherent fault of our algorithm, it does point out that ensuring temporal coherence of data is not always a trivial task in practice.

In conclusion, while promising, a more comprehensive evaluation on different sets of real data is needed to fully understand the behaviour of the proposed method.

4 Conclusions and Future Work

A novel algorithm for incremental learning of Temporally-Coherent Gaussian mixtures was introduced. Promising performance was empirically demonstrated on synthetic data and face appearance streams extracted from realistic video, and qualitatively and quantitatively compared with the standard EM-based fitting.

Acknowledgements We are grateful to Dr. Peter Hall whose suggestions greatly helped us improve this paper. Funding was provided by Toshiba Corporation and Trinity College.

References

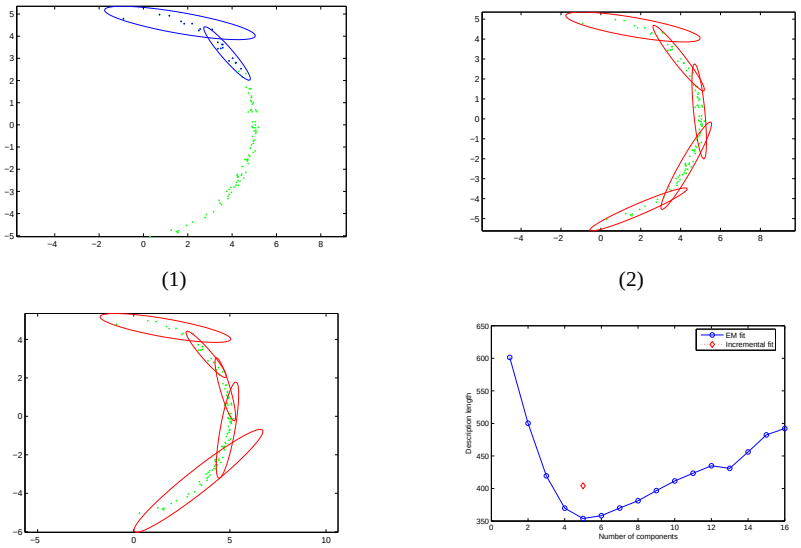
[1] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[2] T. Cootes and C. Taylor. A mixture model for representing shape. *IVC*, 1999.

[3] J. Dahmen, D. Keysers, H. Ney, and M. O. Güld. Statistical image object recognition using mixture densities. *Journal of Mathematical Imaging and Vision*, 14(3):285–296, 1999.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

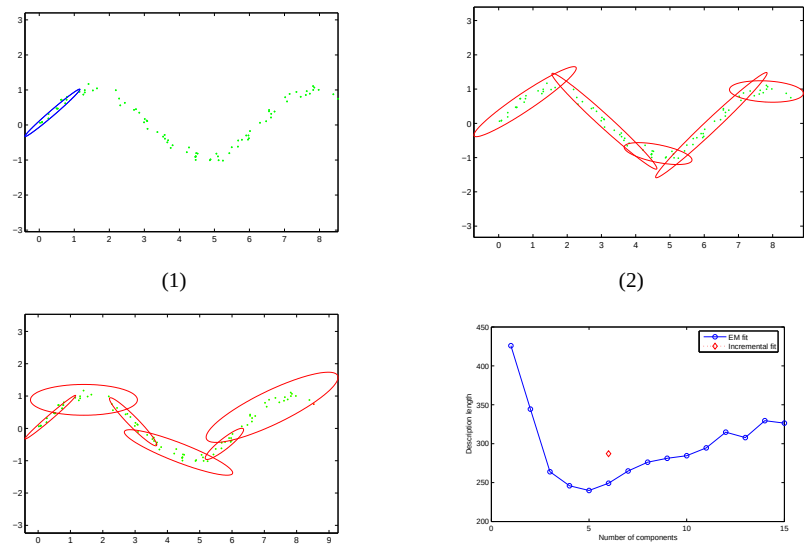
[5] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wily & Sons, Inc., New York, 2nd edition, 2000.



(1) (2)

(3) (4)

(a) Synthetic data set 1



(1) (2)

(3) (4)

(b) Synthetic data set 2

Figure 4: ***Synthetic data sets:*** *(1s) Data (dots) and the initial model (visualized as ellipses corresponding to the parameters of the Gaussian components). (2s) MDL-EM GMM fit. (3s) Incremental GMM fit. (4s) Description length of GMMs fitted using EM and the proposed incremental algorithm (shown is the description length of the final GMM estimate). Our method produces qualitatively meaningful results which are also qualitatively comparable with the best fits obtained using the usual batch method.*

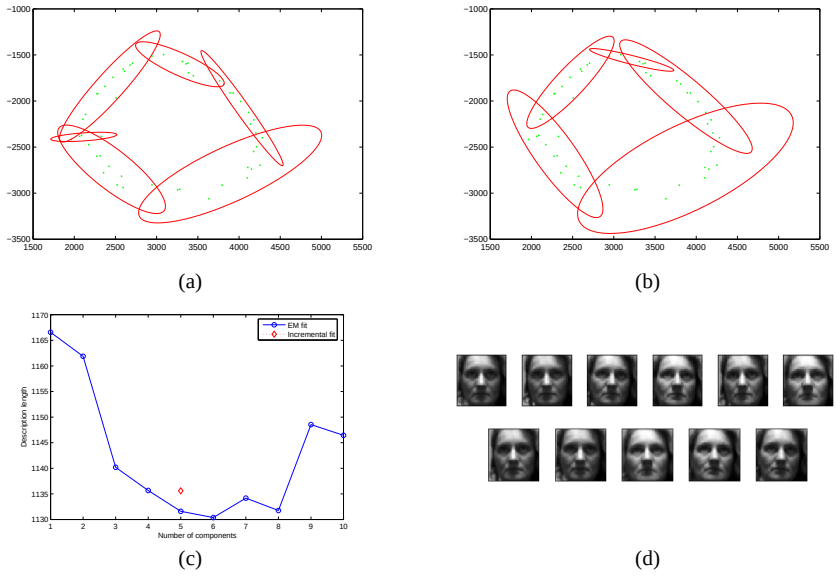


Figure 5: ***Face motion data set:*** *Data (dots) and (a) MDL-EM GMM fit. (b) Incremental GMM fit. (c) Description length of GMMs fitted using EM and the proposed incremental algorithm (shown is the description length of the final GMM estimate). (d) GMM component centres visualized as images for the MDL-EM fit (top) and the incremental algorithm (bottom).*

[6] M. Figueiredo and A. Jain. Unsupervised learning of finite mixture models. *PAMI*, 2003.

[7] R. Gross, J. Yang, and A. Waibel. Growing Gaussian mixture models for pose invariant face recognition. *International Conference on Pattern Recognition*, 1:1088–1091, 2000.

[8] P. Hall, D. Marshall, and R. Martin. Merging and splitting eigenspace models. *PAMI*, 2000.

[9] P. M. Hall and Y. Hicks. A method to add Gaussian mixture models. *Tech. Report, University of Bath*, 2004.

[10] Y. A. Hicks, P. M. Hall, and A. D. Marshall. A method to add Hidden Markov Models with application to learning articulated motion. *BMVC*, 2003.

[11] Y. Raja, S. J. McKenna, and S. Gong. Segmentation and tracking using colour mixture models. *ICCV*, pages 607–614, 1998.

[12] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[13] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[14] M. Song and H. Wang. Highly efficient incremental estimation of Gaussian mixture models for online data stream clustering. *Intelligent Computing: Theory And Applications*, 2005.

[15] N. Vasconcelos and A. Lippman. Learning mixture hierarchies. *NIPS*, 1998.

[16] J. J. Verbeek, N. Vlassis, and B. Kröse. Efficient greedy learning of Gaussian mixture models. *Neural Computation*, 5(2):469–485, 2003.

[17] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

[18] N. Vlassis and A Likas. A kurtosis-based dynamic approach to Gaussian mixture modeling. *Systems, Max, and Cybernetics – Part A: Systems and Humans*, 24(9):393–399, 1999.

[19] C. S. Wallace and D. L. Dowe. Minimum message length and Kolmogorov complexity. *Computer Journal*, 42(4):270–283, 1999.

[20] M. Zwolinski and Z. R. Yang. Mutual information theory for adaptive mixture models. *PAMI*, 2001.