# Gaussian Mixture Models for Time Series Modelling, Forecasting, and Interpolation

Emil Eirola[1] and Amaury Lendasse[123]

[1] Department of Information and Computer Science, Aalto University, FI–00076 Aalto, Finland
emil.eirola@aalto.fi
[2] IKERBASQUE, Basque Foundation for Science, 48011 Bilbao, Spain
[3] Computational Intelligence Group, Computer Science Faculty, University of the Basque Country, Paseo Manuel Lardizabal 1, Donostia/San Sebastián, Spain

**Abstract.** Gaussian mixture models provide an appealing tool for time series modelling. By embedding the time series to a higher-dimensional space, the density of the points can be estimated by a mixture model. The model can directly be used for short-to-medium term forecasting and missing value imputation. The modelling setup introduces some restrictions on the mixture model, which when appropriately taken into account result in a more accurate model. Experiments on time series forecasting show that including the constraints in the training phase particularly reduces the risk of overfitting in challenging situations with missing values or a large number of Gaussian components.

**Keywords:** time series, missing data, Gaussian mixture model

## 1 Introduction

A time series is one of the most common forms of data, and has been studied extensively from weather patterns spanning centuries to sensors and microcontrollers operating on nanosecond scales. The features and irregularities of time series can be modelled through various means, such as autocovariance analysis, trend fitting, or frequency-domain methods. From a machine learning perspective, the most relevant tasks tend to be prediction of one or several future data points, or interpolation for filling in gaps in the data. In this paper, we study a model for analysing time series, which is applicable to both tasks.

For uniformly sampled stationary processes, we propose a versatile methodology to model the features of the time series by embedding the data to a high-dimensional regressor space. The density of the points in this space can then be modelled with Gaussian mixture models [1]. Such an estimate of the probability density enables a direct way to interpolate missing values in the time series and conduct short-to-medium term prediction by finding the conditional expectation of the unknown values. Embedding the time series in a higher-dimensional space imposes some restrictions on the possible distribution of points, but these constraints can be accounted for when fitting the Gaussian mixture models.

The suggested framework can readily be extended to situations with several related time series, using exogenous time series to improve the predictions of a target series. Furthermore, any missing values can be handled by the Gaussian mixture model in a natural manner.

This paper is structured as follows. Section 2 presents the procedure for modelling time series by Gaussian mixture models, the constraints on the Gaussian mixture model due to time series data are discussed in Section 3, and some experiments showing the effect of selecting the number of components and introducing missing values are studied in Section 4.

## 2 Mixture Models for Time Series

Given a time series $z$ of length $n$, corresponding to a stationary process:

$$z_0,\, z_1,\, z_2,\, \ldots,\, z_{n-2},\, z_{n-1},$$

by choosing a regressor length $d$ we can conduct a delay embedding [2] and form the design matrix $\mathbf{X}$,

$$\mathbf{X} = \begin{bmatrix} z_0 & z_1 & \cdots & z_{d-1} \\ z_1 & z_2 & \cdots & z_d \\ \vdots & \vdots & & \vdots \\ z_{n-d} & z_{n-d+1} & \cdots & z_{n-1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_{n-d} \end{bmatrix}. \tag{1}$$

The rows of $\mathbf{X}$ can be interpreted as vectors in $\mathbb{R}^d$. We can model the density of these points by a Gaussian mixture model, with the probability density function

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{2}$$

where $\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the probability density function of the multivariate normal distribution, $\boldsymbol{\mu}_k$ represents the means, $\boldsymbol{\Sigma}_k$ the covariance matrices, and $\pi_k$ the mixing coefficients for each component $k$ ($0 < \pi_k < 1$, $\sum_{k=1}^{K} \pi_k = 1$).

Given a set of data, the standard approach to training a Gaussian mixture model is the EM algorithm [3,4] for finding a maximum-likelihood fit. The log-likelihood of the $N$ data points is given by

$$\log \mathcal{L}(\theta) = \log p(\mathbf{X} \mid \theta) = \sum_{i=1}^{N} \log \left( \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right), \tag{3}$$

where $\theta = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ is the set of parameters defining the model. The log-likelihood can be maximised by applying the EM algorithm. After some initialisation of parameters, the E-step is to find the expected value of the log likelihood function, with respect to the conditional distribution of latent variables $\mathbf{Z}$ given the data $\mathbf{X}$ under the current estimate of the parameters $\theta^{(t)}$:

$$Q(\theta \mid \theta^{(t)}) = \mathrm{E}_{\mathbf{Z}|\mathbf{X},\theta^{(t)}} \left[ \log \mathcal{L}(\theta; \mathbf{X}, \mathbf{Z}) \right] \tag{4}$$

This requires evaluating the probabilities $t_{ik}$ that $\boldsymbol{x}_i$ is generated by the $k$th Gaussian using the current parameter values:

$$t_{ik}^{(t)} = \frac{\pi_k^{(t)} \mathcal{N}(\boldsymbol{x}_i \mid \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)})}{\sum_{j=1}^{K} \pi_j^{(t)} \mathcal{N}(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j^{(t)}, \boldsymbol{\Sigma}_j^{(t)})} \,. \tag{5}$$

In the M-step, the expected log-likelihood is maximised:

$$\theta^{(t+1)} = \arg\max_{\theta} \, Q(\theta \mid \theta^{(t)}) \,, \tag{6}$$

which corresponds to re-estimating the parameters with the new probabilities:

$$\boldsymbol{\mu}_k^{(t+1)} = \frac{1}{N_k} \sum_{i=1}^{N} t_{ik}^{(t)} \boldsymbol{x}_i \,, \tag{7}$$

$$\boldsymbol{\Sigma}_k^{(t+1)} = \frac{1}{N_k} \sum_{i=1}^{N} t_{ik}^{(t)} (\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(t+1)})(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(t+1)})^T \,, \tag{8}$$

$$\pi_k^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} t_{ik}^{(t)} \,. \tag{9}$$

Here $N_k = \sum_{i=1}^{N} t_{ik}^{(t)}$ is the effective number of samples covered by the $k$th component. The E and M-steps are alternated repeatedly until convergence. As the algorithm tends to occasionally converge to sub-optimal solutions, the procedure can be repeated to find the best fit.

### 2.1 Model Structure Selection

The selection of the number of components $K$ is crucial, and has a significant effect on the resulting accuracy. Too few components are not able to model the distribution appropriately, while having too many components causes issues of overfitting.

The number of components can be selected according to the Akaike information criterion (AIC) [5] or the Bayesian information criterion (BIC) [6]. Both are expressed as a function of the log-likelihood of the converged mixture model:

$$\mathrm{AIC} = -2 \log \mathcal{L}(\theta) + 2P \,, \tag{10}$$

$$\mathrm{BIC} = -2 \log \mathcal{L}(\theta) + \log(N) P \,, \tag{11}$$

where $P = Kd + \frac{1}{2} Kd(d+1) + K - 1$ is the number of free parameters. The EM algorithm is run for several different values of $K$, and the model which minimises the chosen criterion is selected. As $\log(N) > 2$ in most cases, BIC more aggresively penalises an increase in $P$, generally resulting in a smaller choice for $K$ than by AIC.

### 2.2 Forecasting

The model readily lends itself to being used for short-to-medium term time series prediction. For example, if a time series is measured monthly and displays some seasonal behaviour, a Gaussian model could be trained with a regressor size of 24 (two years). This allows us to take the last year's measurements as the 12 *first* months, and determine the conditional expectation of the following 12 months.

The mixture model provides a direct way to calculate the conditional expectation. Let the input dimensions be partitioned into past values $P$ (known) and future values $F$ (unknown). Then, given a sample $\boldsymbol{x}_i^P$ for which only the past values are known and a prediction is to be made, calculate the probabilities of it belonging to each component

$$t_{ik} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i^P \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_i^P \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}\,, \tag{12}$$

where $\mathcal{N}(\boldsymbol{x}_i^P \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the *marginal* multivariate normal distribution probability density of the observed (i.e., past) values of $\boldsymbol{x}_i$.

Let the means and covariances of each component also be partitioned according to past and future variables:

$$\boldsymbol{\mu}_k = \begin{bmatrix} \boldsymbol{\mu}_k^P \\ \boldsymbol{\mu}_k^F \end{bmatrix}, \quad \boldsymbol{\Sigma}_k = \begin{bmatrix} \boldsymbol{\Sigma}_k^{PP} & \boldsymbol{\Sigma}_k^{PF} \\ \boldsymbol{\Sigma}_k^{FP} & \boldsymbol{\Sigma}_k^{FF} \end{bmatrix}. \tag{13}$$

Then the conditional expectation of the future values with respect to the component $k$ is given by

$$\tilde{\boldsymbol{y}}_{ik} = \boldsymbol{\mu}_k^F + \boldsymbol{\Sigma}_k^{FP} (\boldsymbol{\Sigma}_k^{PP})^{-1} (\boldsymbol{x}_i^P - \boldsymbol{\mu}_k^P) \tag{14}$$

in accordance with [7, Thm. 2.5.1]. The total conditional expectation can now be found as a weighted average of these predictions by the probabilities $t_{ik}$:

$$\hat{\boldsymbol{y}}_i = \sum_{k=1}^{K} t_{ik} \tilde{\boldsymbol{y}}_{ik}\,. \tag{15}$$

It should be noted that the method directly estimates the full vector of future values at once, in contrast with most other methods which would separately predict each required data point.

### 2.3 Missing Values and Imputation

The proposed method is directly applicable to time series with missing values. Missing data in the time series become diagonals of missing values in the design matrix. The EM-algorithm can in a natural way account for missing values in the samples [8,9].

An assumption here is that data are Missing-at-Random (MAR) [10]:

$$P(M \mid x_{\text{obs}}, x_{\text{mis}}) = P(M \mid x_{\text{obs}})\,,$$

i.e., the event $M$ of a measurement being missing is independent from the value it would take ($x_{\text{mis}}$), conditional on the observed data ($x_{\text{obs}}$). The stronger assumption of Missing-Completely-at-Random (MCAR) is not necessary, as MAR is an ignorable missing-data mechanism in the sense that maximum likelihood estimation still provides a consistent estimator [10].

To conduct missing value imputation, the procedure is the same as for prediction in Section 2.2. The only difference is that in this case the index set $P$ contains all known values for a sample (both before and after the target to be predicted), while $F$ contains the missing values that will be imputed.

### 2.4 Missing-data Padding

When using an implementation of the EM algorithm that is able to handle missing values, it is reasonable to consider that every value before and after the recorded time series consists is missing. This can be seen as "padding" the design matrix $\mathbf{X}$ with missing values (marked as '?'), effectively increasing the number of samples available for training from $n-d+1$ to $n+d-1$ (cf. Eq. (1)):

$$\mathbf{X} = \begin{bmatrix} ? & ? & \dots & ? & z_0 \\ ? & ? & \dots & z_0 & z_1 \\ \vdots & \vdots & & \vdots & \vdots \\ ? & z_0 & \dots & z_{d-3} & z_{d-2} \\ z_0 & z_1 & \dots & z_{d-2} & z_{d-1} \\ \vdots & \vdots & & \vdots & \vdots \\ z_{n-d} & z_{n-d+1} & \dots & z_{n-2} & z_{n-1} \\ z_{n-d+1} & z_{n-d+2} & \dots & z_{n-1} & ? \\ \vdots & \vdots & & \vdots & \vdots \\ z_{n-1} & ? & \dots & ? & ? \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_{d-2} \\ \boldsymbol{x}_{d-1} \\ \vdots \\ \boldsymbol{x}_{n-1} \\ \boldsymbol{x}_n \\ \vdots \\ \boldsymbol{x}_{n+d-2} \end{bmatrix} \tag{16}$$

Fitting the mixture model using this padded design matrix has the added advantage that the sample mean and variance of (the observed values in) each column is guaranteed to be equal. The missing-data padding can thus be a useful trick even if the time series itself features no missing values, particularly if only a limited amount of data is available.

## 3 Constraining the Global Covariance

The Gaussian mixture model is ideal for modelling arbitrary continuous distributions. However, embedding a time series to a higher-dimensional space cannot lead to an arbitrary distribution. For instance, the mean and variance for each dimension should equal the mean and variance of the time series. In addition, all second-order statistics, such as covariances, should equal the respective autocovariances of the time series. These restrictions impose constraints on the mixture

model, and accounting for them appropriately should lead to a more accurate model when fitting to data.

In the EM algorithm, we estimate means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$, and mixing coefficients $\pi_k$ for each component $k$, and then the global mean and covariance of the distribution defined by the model is

$$\boldsymbol{\mu} = \sum_{k=1}^{K} \pi_k \boldsymbol{\mu}_k \,, \qquad \boldsymbol{\Sigma} = \sum_{k=1}^{K} \pi_k \left( \boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T \right) - \boldsymbol{\mu}\boldsymbol{\mu}^T \,. \tag{17}$$

However, the global mean and covariance correspond to the mean and autocovariance matrix of the time series. This implies that the global mean for each dimension should be equal. Furthermore, the global covariance matrix should be symmetric and Toeplitz ("diagonal-constant"):

$$\boldsymbol{\Sigma} \approx \mathbf{R}_z = \begin{bmatrix} r_z(0) & r_z(1) & r_z(2) & \dots & r_z(d-1) \\ r_z(1) & r_z(0) & r_z(1) & \dots & r_z(d-2) \\ \vdots & \vdots & \vdots & & \vdots \\ r_z(d-1) & r_z(d-2) & r_z(d-3) & \dots & r_z(0) \end{bmatrix}$$

where $r_z(l)$ is the autocovariance of the time series $z$ at lag $l$.

In practice, these statistics usually do not exactly correspond to each other, even when training the model on the missing-data padded design matrix discussed in Section 2.4. Unfortunately, the question of how to enforce this constraint in each M-step has no trivial solution. Forcing every component to have an equal mean and Toeplitz covariance structure by its own is one possibility, but this is far too restrictive.

Our suggestion is to calculate the M-step by Eqs. (7–9), and then modify the parameters as little as possible in order to achieve the appropriate structure. As $\theta = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k\}_{k=1}^K$ contains the parameters for the mixture model, let $\Omega$ be the space of all possible values for $\theta$, and $T \subset \Omega$ be the subset such that all parameter values $\theta \in T$ correspond to a global mean with equal elements, and Toeplitz covariance matrix by Eq. (17).

When maximising the expected log-likelihood with the constraints, the M-step should be

$$\theta^{(t+1)} = \arg\max_{\theta \in T} Q(\theta \mid \theta^{(t)}) \,, \tag{18}$$

but this is not feasible to solve exactly. Instead, we solve the conventional M-step

$$\theta' = \arg\max_{\theta \in \Omega} Q(\theta \mid \theta^{(t)}) \,, \tag{19}$$

and then project this $\theta'$ onto $T$ to find the closest solution

$$\theta^{(t+1)} = \arg\min_{\theta \in T} d(\theta, \theta') \tag{20}$$

for some interpretation of the distance $d(\theta, \theta')$. If the difference is small, the expected log-likelihood $Q(\theta^{(t+1)} \mid \theta^{(t)})$ should not be too far from the optimal

$\max_{\theta \in T} Q(\theta \mid \theta^{(t)})$. As the quantity is not maximised, though it can be observed to increase, this becomes a Generalised EM (GEM) algorithm. As long as an increase is ensured in every iteration, the GEM algorithm is known to have similar convergence properties as the EM algorithm [3,4].

Define the distance function between sets of parameters as follows:

$$d(\theta, \theta') = \sum_k \|\boldsymbol{\mu}_k - \boldsymbol{\mu}'_k\|^2 + \sum_k \|\mathbf{S}_k - \mathbf{S}'_k\|_F^2 + \sum_k (\pi_k - \pi'_k)^2 \,, \qquad (21)$$

where $\mathbf{S}_k = \boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T$ are the second moments of the distributions of each component and $\|\cdot\|_F$ is the Frobenius norm. Using Lagrange multipliers, it can be shown that this distance function is minimised by the results presented below in Eqs. (22) and (23).

### 3.1 The Mean

After an iteration of the normal EM-algorithm by Eqs. (7–9), find the vector with equal components which is nearest to the global mean $\boldsymbol{\mu}$ as calculated by Eq. (17). This is done by finding the mean $m$ of the components of $\boldsymbol{\mu}$, and calculating the discrepancy $\boldsymbol{\delta}$ of how much the current mean is off from the equal mean:

$$m = \frac{1}{d} \sum_{j=1}^{d} \mu_j, \qquad \boldsymbol{\delta} = \boldsymbol{\mu} - m\mathbf{1} \,,$$

where $\mathbf{1}$ is a vector of ones. Shift the means of each component to compensate, as follows:

$$\boldsymbol{\mu}'_k = \boldsymbol{\mu}_k - \frac{\pi_k}{\sum_{j=1}^{K} \pi_j^2} \boldsymbol{\delta} \qquad \forall k \,. \qquad (22)$$

As can be seen, components with larger $\pi_k$ take on more of the "responsibility" of the discrepancy, as they contribute more to the global statistics. Any weights which sum to unity would fulfil the constraints, but choosing the weights to be directly proportional to $\pi_k$ minimises the distance in Eq. (21).

### 3.2 The Covariance

After updating the means $\boldsymbol{\mu}_k$, recalculate the covariances around the updated values as

$$\boldsymbol{\Sigma}_k \leftarrow \boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T - \boldsymbol{\mu}'_k \boldsymbol{\mu}'^T_k \qquad \forall k \,.$$

Then, find the nearest (in Frobenius norm) Toeplitz matrix $\mathbf{R}$ by calculating the mean of each diagonal of the global covariance matrix $\boldsymbol{\Sigma}$ (from Eq. (17)):

$$r(0) = \frac{1}{d} \sum_{j=1}^{d} \Sigma_{j,j}, \quad r(1) = \frac{1}{d-1} \sum_{j=1}^{d-1} \Sigma_{j,j+1}, \quad r(2) = \frac{1}{d-2} \sum_{j=1}^{d-2} \Sigma_{j,j+2}, \quad \text{etc.}$$

The discrepancy $\mathbf{\Delta}$ from this Toeplitz matrix is

$$\mathbf{\Delta} = \mathbf{\Sigma} - \mathbf{R}, \quad \text{where } \mathbf{R} = \begin{bmatrix} r(0) & r(1) & r(2) & \dots & r(d-1) \\ r(1) & r(0) & r(1) & \dots & r(d-2) \\ \vdots & \vdots & \vdots & & \vdots \\ r(d-1) & r(d-2) & r(d-3) & \dots & r(0) \end{bmatrix}.$$

In order to satisfy the constraint of a Toeplitz matrix for the global covariance, the component covariances are updated as

$$\mathbf{\Sigma}'_k = \mathbf{\Sigma}_k - \frac{\pi_k}{\sum_{j=1}^{K} \pi_j^2} \mathbf{\Delta} \qquad \forall k\,, \tag{23}$$

the weights being the same as in Eq. (22). Eqs. (22) and (23), together with $\pi'_k = \pi_k$, minimise the distance in Eq. (21) subject to the constraints.

### 3.3 Heuristic Correction

Unfortunately the procedure described above seems to occasionally lead to matrices $\mathbf{\Sigma}'_k$ which are not positive definite. Hence an additional heuristic correction $c_k$ is applied in such cases to force the matrix to remain positive definite:

$$\mathbf{\Sigma}''_k = \mathbf{\Sigma}_k - \frac{\pi_k}{\sum_{k=1}^{K} \pi_k^2} \mathbf{\Delta} + c_k \mathbf{I} \qquad \forall k. \tag{24}$$

In the experiments, the value $c_k = 1.1|\lambda_{k0}|$ is used, where $\lambda_{k0}$ is the most negative eigenvalue of $\mathbf{\Sigma}'_k$. The multiplier needs to be larger than unity to avoid making the matrix singular.

A more appealing correction would be to only increase the negative (or zero) eigenvalues to some acceptable, positive, value. However, this would break the constraint of a Toeplitz global covariance matrix, and hence the correction must be applied to all eigenvalues, as is done in Eq. (24) by adding to the diagonal.

### 3.4 Free Parameters

The constraints reduce the number of free parameters relevant to calculating the AIC and BIC. Without constraints, the number of free parameters is

$$P = \underbrace{Kd}_{\text{means}} + \underbrace{\frac{1}{2}Kd(d+1)}_{\text{covariances}} + \underbrace{K-1}_{\text{mixing coeffs}}\,,$$

where $K$ is the number of Gaussian components, and $d$ is the regressor length. There are $d-1$ equality constraints for the mean, and $\frac{1}{2}d(d-1)$ constraints for the covariance, each reducing the number of free parameters by 1. With the constraints, the number of free parameters is then

$$P' = \underbrace{(K-1)d+1}_{\text{means}} + \underbrace{\frac{1}{2}(K-1)d(d+1)+d}_{\text{covariances}} + \underbrace{K-1}_{\text{mixing coeffs}}\,.$$

The leading term is reduced from $\frac{1}{2}Kd^2$ to $\frac{1}{2}(K-1)d^2$, in effect allowing one additional component for approximately the same number of free parameters.

### 3.5 Exogenous Time Series or Non-contiguous Lag

If the design matrix is formed in a different way than by taking consecutive values, the restrictions for the covariance matrix will change. Such cases are handled by forcing any affected elements in the matrix to equal the mean of the elements it should equal. This will also affect the number of free parameters.

As this sort of delay embedding may inherently have a low intrinsic dimension, optimising the selection of variables could considerably improve accuracy.

## 4 Experiments: Time Series Forecasting

To show the effects of the constraints and the number of components on the prediction accuracy, some experimental results are shown here. The studied time series is the Santa Fe time series competition data set A: Laser generated data [11]. The task is set at predicting the next 12 values, given the previous 12. This makes the regressor size $d = 24$, and the mixture model fitting is in a 24-dimensional space. The original 1000 points of the time series are used for training the model, and the continuation (9093 points) as a test set for estimating the accuracy of the prediction. Accuracy is determined by the mean squared error (MSE), averaging over the 12 future values for all samples. No variable selection is conducted, and all 12 variables in the input are used for the model. The missing-data padded design matrix of Section 2.4 is used for the training, even when the time series otherwise has no missing values.

### 4.1 The Number of Components

Gaussian mixture models were trained separately for 1 through 30 components, each time choosing out of 10 runs the best result in terms of log-likelihood. In order to provide a perspective on average behaviour, this procedure was repeated 20 times both with and without the constraints detailed in Section 3.

The first two plots in Fig. 1 show the MSE of the prediction on the training and test sets, as an average of the 20 repetitions. It is important to note that the model fitting and selection was conducted by maximising the log-likelihood, and not by attempting to minimise this prediction error. Nevertheless, it can be seen that the training error decreases when adding components, and is consistently lower than the test error, as expected. Notably, the difference between training and test errors is much smaller for the constrained mixture model than the unconstrained one. Also, the training error is consistently decreasing for both models when increasing the number of components, but for the test error this is true only for constrained model. It appears that the unconstrained model results in overfitting when used with more than 10 components. For 1 to 10 components,

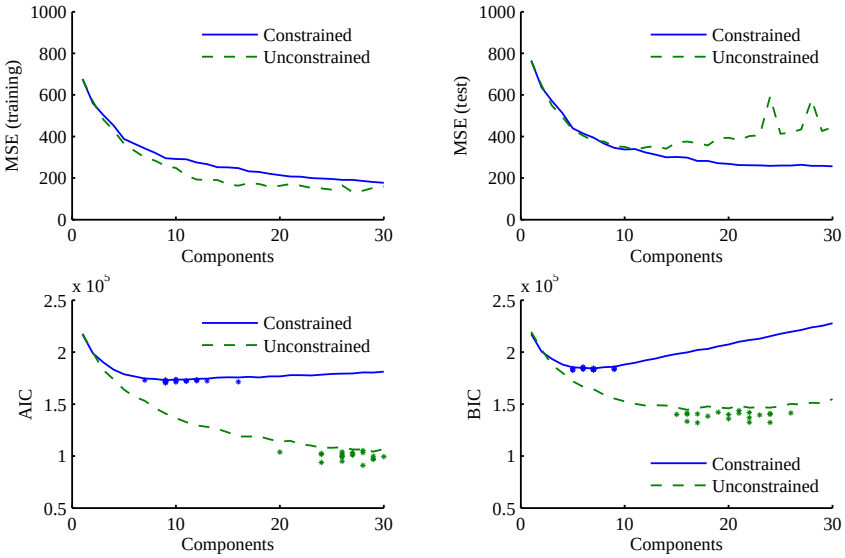


**Fig. 1.** Results on the Santa Fe A Laser time series data, including the average MSE of the 12-step prediction on the training and test sets, AIC and BIC values for both the constrained and unconstrained mixture models for 1 through 30 components.

there is no notable difference in the test error between the two models, presumably because around 10 components are required for a decent approximation of the density. However, for 10 or more components, the constraints provide a consistent improvement in the forecasting accuracy.

The third and fourth plots in Fig. 1 shows the evolution of the average AIC and BIC of the converged model. The line plots show the average value of the criterion, and the asterisks depict the minimum AIC (or BIC) value (i.e., the selected model) for each of the 20 runs. As results on the test set are not available in the model selection phase, the number of components should be chosen based on these criteria. As the log-likelihood grows much faster for the unconstrained model, this results in a consistently larger number of components as selected by both criteria. Comparing the AIC and BIC, it is clear that BIC tends to choose fewer components, as expected. However, the test MSE for the constrained model keeps decreasing even until 30 components, suggesting that both criteria may be exaggerating the penalisation in this case when increasing the model size.

### 4.2 Missing Data

To study the effect of missing values, the modelling of the Santa Fe Laser time series is repeated with various degrees of missing data (1% through 50%). In the training phase, missing data is removed at random from the time series before forming the padded design matrix. To calculate the testing MSE, missing values

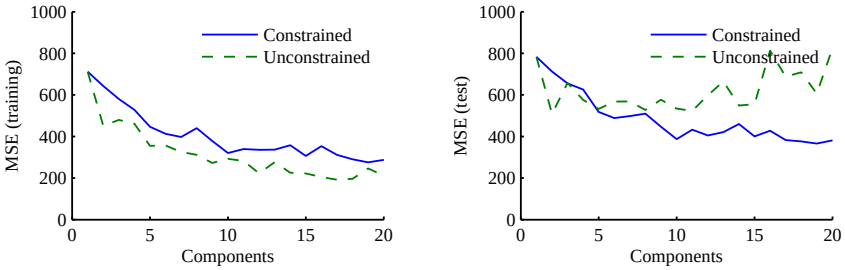


**Fig. 2.** Results on the Santa Fe A Laser time series data with 10% missing values, including the average MSE of the 12-step prediction on the training and test sets for both the constrained and unconstrained mixture models for 1 through 20 components.

are also removed from the inputs (i.e., the past values from which predictions are to be made) at the same probability. The MSE is then calculated as the error between the forecast and the actual time series (with no values removed).

The training and test MSE for 10% missing values are shown in Fig. 2. The behaviour is similar to the corresponding plots in Fig. 1, although the difference in the testing MSE appears more pronounced, and for a lower number of components. This supports the notion that the constraints help against overfitting.

Fig. 3 shows the number of components selected by AIC and BIC, and the corresponding test MSEs, for various degrees of missing values. As expected, the forecasting accuracy deteriorates with an increasing ratio of missing data. The number of components selected by the AIC remains largely constant, and the constrained model consistently performs better. The BIC, on the other hand, seems to selected far too few components for the constrained model (the MSE plots in Figs. 1 and 2 suggest five components are far from sufficient), resulting in a reduced forecasting accuracy.

Figs. 1 and 3 reveal largely similar results between using AIC and BIC for the unconstrained case. However, for the constrained model, BIC is clearly too restrictive, and using AIC leads to more accurate results.

## 5 Conclusions

Time series modelling through Gaussian mixture models is an appealing method, capable of accurate short-to-medium term prediction and missing value interpolation. Certain restrictions on the structure of the model arise naturally through the modelling setup, and appropriately including these constraints in the modelling procedure further increases its accuracy. The constraints are theoretically justified, and experiments support their utility. The effect is negligible when there are enough samples or few components such that fitting a mixture model is easy, but in more challenging situations with a large number of components or missing values they considerably reduce the risk of overfitting.

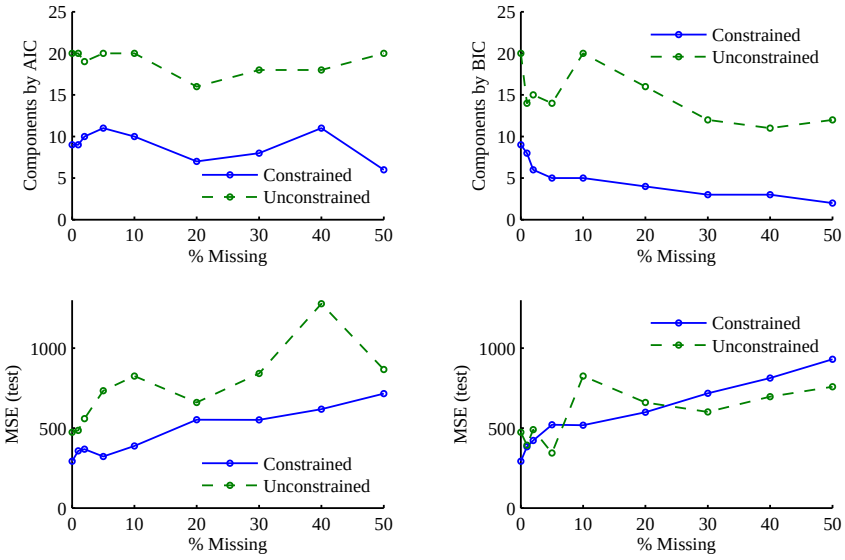


**Fig. 3.** Results on the Santa Fe A Laser time series data for various degrees of missing values, including the number of components selected by AIC (left) and BIC (right) and the resulting MSEs of the corresponding test set predictions.

## References


1. McLachlan, G., Peel, D.: Finite Mixture Models. Wiley Series in Probability and Statistics. John Wiley & Sons, New York (2000)
2. Kantz, H., Schreiber, T.: Nonlinear Time Series Analysis. Cambridge nonlinear science series. Cambridge University Press (2004)
3. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological) **39**(1) (1977) pp. 1–38
4. McLachlan, G., Krishnan, T.: The EM Algorithm and Extensions. Wiley Series in Probability and Statistics. John Wiley & Sons, New York (1997)
5. Akaike, H.: A new look at the statistical model identification. Automatic Control, IEEE Transactions on **19**(6) (December 1974) 716–723
6. Schwarz, G.: Estimating the dimension of a model. The annals of statistics **6**(2) (1978) 461–464
7. Anderson, T.W.: An Introduction to Multivariate Statistical Analysis. Third edn. Wiley-Interscience, New York (2003)
8. Ghahramani, Z., Jordan, M.: Learning from incomplete data. Technical report, Lab Memo No. 1509, CBCL Paper No. 108, MIT AI Lab (1995)
9. Hunt, L., Jorgensen, M.: Mixture model clustering for mixed data with missing information. Computational Statistics & Data Analysis **41**(3–4) (2003) 429–440
10. Little, R.J.A., Rubin, D.B.: Statistical Analysis with Missing Data. Second edn. Wiley-Interscience (2002)
11. Gershenfeld, N., Weigend, A.: The Santa Fe time series competition data (1991) `http://www-psych.stanford.edu/~andreas/Time-Series/SantaFe.html`.