# Mixtures of ARMA Models for Model-Based Time Series Clustering *

Yimin Xiong Dit-Yan Yeung
Department of Computer Science, Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong

## Abstract

*Clustering problems are central to many knowledge discovery and data mining tasks. However, most existing clustering methods can only work with fixed-dimensional representations of data patterns. In this paper, we study the clustering of data patterns that are represented as sequences or time series possibly of different lengths. We propose a model-based approach to this problem using mixtures of* autoregressive moving average *(ARMA) models. We derive an* expectation-maximization *(EM) algorithm for learning the mixing coefficients as well as the parameters of the component models. Experiments were conducted on simulated and real datasets. Results show that our method compares favorably with another method recently proposed by others for similar time series clustering problems.*

## 1. Introduction

*Clustering* is the unsupervised process of grouping data patterns into clusters so that patterns within a cluster bear strong similarity to one another but are very dissimilar to patterns in other clusters. Clustering problems are central to many knowledge discovery and data mining tasks. Many clustering techniques have been studied for data patterns that are represented as points in multidimensional spaces of fixed dimensionality. In this paper, we deal with sequential patterns such as sequences and time series possibly of different lengths.

*Distance-based* methods and *model-based* methods are two major classes of clustering methods. They are analogous to other nonparametric and parametric methods, respectively, in that the former category (i.e., distance-based or nonparametric methods) assumes only some weak structure of the data, but the latter category (i.e., model-based or parametric methods) assumes some strong structure. For time series data, model-based methods provide a principled approach for handling the problem of modeling and clustering time series of different lengths. In this paper, we will focus on model-based time series clustering methods. In particular, *mixture models* [9] will be used.

*A longer version of this paper can be found in `http://www.cs.ust.hk/faculty/dyyeung/paper/ps/yeung.icdm2002long.ps`.

## 2. Related work

Finite mixtures of Markov chains [2] have been proposed for clustering time series. The *expectation-maximization* (EM) algorithm [3] is used to learn the mixing coefficients as well as the parameters of the component models. The number of clusters can be determined by comparing different choices of the number based on some scoring scheme. Another approach to the clustering of time series modeled by Markov chains is called Bayesian clustering by dynamics (BCD) [12] which can best be seen as a hybrid approach with both model-based and distance-based flavors.

While simple Markov chains are good enough for some applications, some time series can be modeled better using *hidden Markov models* (HMM) [11] due to their ability of handling temporal and spatial uncertainties simultaneously. Finite mixtures of HMMs have been studied. Similar to mixtures of Markov chains, the EM algorithm can also be used for HMM mixtures [14, 8]. To trade accuracy for efficiency, the $k$-means algorithm (used in [10]) and the rival penalized competitive learning (RPCL) algorithm (used in [7]) have also been used in place of EM.

In addition to Markov chains and HMMs, *autoregressive moving average* (ARMA) and *autoregressive integrated moving average* (ARIMA) models have also been used extensively for time series analysis [1]. Kwok *et al.* [6] applied mixtures of ARMA models as well as their special cases, mixtures of autoregressive (AR) models, for time series modeling and forecasting. However, clustering applications based on such mixture models were not studied by them. More recently, a method was proposed by Kalpakis *et al.* for clustering ARIMA time series [5]. This method is similar to the BCD method in that it is a hybrid method with both model-based and distance-based characteristics.

In the next section, we will propose a new time series clustering method based on mixtures of ARMA models.

## 3. Mixtures of ARMA models

The ARIMA model introduced by Box and Jenkins [1] is a combination of three types of time series data processes, namely, autoregressive, integrated, and moving average processes. A stationary ARIMA model with autoregressive order $p$ and moving average order $q$ is commonly denoted as ARMA$(p, q)$. Given a time series $\mathbf{x} = \{x_t\}_{t=1}^n$, the fitted ARMA$(p, q)$ model takes the form

$$x_t = \phi_0 + \sum_{j=1}^{p} \phi_j x_{t-j} + \sum_{j=1}^{q} \theta_j e_{t-j} + e_t, \qquad t = 1, 2, \ldots, n,$$

where $n$ is the length of the time series, $\phi_0$ is a constant term, $\{\phi_1, \phi_2, \ldots, \phi_p, \theta_1, \theta_2, \ldots, \theta_q\}$ is the set of AR$(p)$ and MA$(q)$ coefficients, and $\{e_t\}_{t=1}^n$ is a sequence of independent and identically distributed (IID) Gaussian white noise terms with variance $\sigma^2$. From [1], we can express the natural logarithm of the conditional likelihood function as

$$\ln P(\mathbf{x}|\boldsymbol{\Phi}) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{t=1}^{n} e_t^2,$$

where $\boldsymbol{\Phi} = \{\phi_0, \phi_1, \phi_2, \ldots, \phi_p, \theta_1, \theta_2, \ldots, \theta_q, \sigma^2\}$ is the set of all model parameters and $e_t$ must be estimated recursively.

We now extend standard ARMA models to mixtures of ARMA models, or simply called ARMA mixtures, for time series clustering. Let us assume that the time series data are generated by $M$ different ARMA models, which correspond to the $M$ clusters of interest denoted as $\omega_1, \omega_2, \ldots, \omega_M$. Let $P(\mathbf{x}|\omega_k, \boldsymbol{\Phi}_k)$ denote the conditional likelihood function or density function of component model $k$, with $\boldsymbol{\Phi}_k$ being the set of parameters for the model. Let $P(\omega_k)$ be the prior probability that a time series comes from model $k$. The conditional likelihood function of the mixture model can be expressed in the form of a mixture density as $P(\mathbf{x}|\boldsymbol{\Theta}) = \sum_{k=1}^{M} P(\mathbf{x}|\omega_k, \boldsymbol{\Phi}_k)P(\omega_k)$, where $\boldsymbol{\Theta} = \{\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2, \ldots, \boldsymbol{\Phi}_M, P(\omega_1), P(\omega_2), \ldots, P(\omega_M)\}$ represents the set of all model parameters for the mixture model. For a time series $\mathbf{x}$, it is assigned to cluster $\omega_k$ with posterior probability $P(\omega_k|\mathbf{x})$, where $\sum_{k=1}^{M} P(\omega_k|\mathbf{x}) = 1$.

Suppose we are given a set $\mathbf{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ of $N$ time series. Under the usual assumption that different time series are conditionally independent given the underlying model parameters, we can express the likelihood of $\mathbf{D}$ as $P(\mathbf{D}|\boldsymbol{\Theta}) = \prod_{i=1}^{N} P(\mathbf{x}_i|\boldsymbol{\Theta})$. Model parameter learning amounts to finding the *maximum a posteriori* (MAP) parameter estimate given the data set $\mathbf{D}$, i.e., $\widehat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} \left[P(\mathbf{D}|\boldsymbol{\Theta})P(\boldsymbol{\Theta})\right]$. If we take a noninformative prior on $\boldsymbol{\Theta}$, learning degenerates to *maximum likelihood estimation* (MLE), i.e., $\widehat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} P(\mathbf{D}|\boldsymbol{\Theta})$. This MLE problem can be solved efficiently using EM, which will be discussed in detail in the next section.

## 4. EM learning algorithm

The EM algorithm is an iterative approach to MLE or MAP estimation problems with incomplete data. It has been widely used for many applications, including clustering and mixture density estimation problems [13].

The likelihood of $\mathbf{D}$ can be rewritten as a function of the parameter vector $\boldsymbol{\Theta}$ for a given data set $\mathbf{D}$, i.e., $L(\boldsymbol{\Theta}; \mathbf{D}) = P(\mathbf{D}|\boldsymbol{\Theta}) = \prod_{i=1}^{N} P(\mathbf{x}_i|\boldsymbol{\Theta})$. Assuming a noninformative prior on $\boldsymbol{\Theta}$, the goal of the EM algorithm is to find $\boldsymbol{\Theta}$ that maximizes the likelihood $L(\boldsymbol{\Theta}; \mathbf{D})$ or the log-likelihood $\ell(\boldsymbol{\Theta}; \mathbf{D}) = \sum_{i=1}^{N} \ln P(\mathbf{x}_i|\boldsymbol{\Theta})$.

Since $\mathbf{D}$ is the incomplete data, we assume the missing data to be $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\}$, such that $\mathbf{D}$ and $\mathbf{Z}$ form the complete data $(\mathbf{D}, \mathbf{Z})$. Thus the complete-data log-likelihood function is $\ln P(\mathbf{D}, \mathbf{Z}|\boldsymbol{\Theta})$. If we knew the missing data (and hence the complete data), parameter estimation would be straightforward. Without knowing the missing data, however, the EM algorithm has to iterate between the Expectation step (E-step) and the Maximization step (M-step). In the E-step, we calculate the expected value $Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}(t))$ of the complete-data log-likelihood with respect to the unknown data $\mathbf{Z}$ given the observed data $\mathbf{D}$ and the current parameter estimate $\boldsymbol{\Theta}(t)$, i.e., $Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}(t)) = E\left[\ln P(\mathbf{D}, \mathbf{Z}|\boldsymbol{\Theta}) \mid \mathbf{D}, \boldsymbol{\Theta}(t)\right]$. In the M-step, we try to maximize $Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}(t))$ with respect to $\boldsymbol{\Theta}$ to find the new parameter estimate $\boldsymbol{\Theta}(t+1)$.

In the context of using ARMA mixtures for clustering, the missing data correspond to the unknown cluster or group membership of each time series $\mathbf{x}_i$. The log-likelihood $\ell(\boldsymbol{\Theta}; \mathbf{D})$ can thus be expressed as $\ell(\boldsymbol{\Theta}; \mathbf{D}) = \sum_{i=1}^{N} \ln P(\mathbf{x}_i|\omega_{\mathbf{z}_i}, \boldsymbol{\Phi}_{\mathbf{z}_i}) + \sum_{i=1}^{N} \ln P(\omega_{\mathbf{z}_i})$. Given the observed data $\mathbf{D}$ and the current parameter estimate $\boldsymbol{\Theta}(t)$, the expectation of the complete-data log-likelihood becomes

$$\begin{aligned} Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}(t)) = &\sum_{i=1}^{N}\sum_{k=1}^{M} P(\omega_k|\mathbf{x}_i, \boldsymbol{\Theta}(t)) \ln P(\mathbf{x}_i|\omega_k, \boldsymbol{\Phi}_k) \\ &+ \sum_{i=1}^{N}\sum_{k=1}^{M} P(\omega_k|\mathbf{x}_i, \boldsymbol{\Theta}(t)) \ln P(\omega_k). \end{aligned}$$

The EM algorithm iteratively maximizes $Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}(t))$ until convergence. For each iteration, we compute the posterior probabilities $P(\omega_k|\mathbf{x}_i, \boldsymbol{\Theta}(t))$ and $Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}(t))$ using the current parameter estimate $\boldsymbol{\Theta}(t)$ in the E-step, and update the parameter estimate by maximizing $Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}(t))$ with respect to $\boldsymbol{\Theta}$ to obtain $\boldsymbol{\Theta}(t+1)$ in the M-step.

## 5. Results on simulated datasets

As in [5], experiments were conducted on both simulated and real datasets. Instead of handling ARIMA time series directly, a preprocessing step of differencing was first applied to convert each nonstationary ARIMA time series into the corresponding stationary ARMA time series. Moreover, as discussed in [1], ARMA models can be converted into equivalent AR models. Thus, for simplicity, we in fact used mixtures of AR models in all our experiments, although the EM algorithm presented above can be used for general ARMA mixtures.

The cluster similarity measure [4] was used to evaluate and compare the clustering results obtained by Kalpakis *et al.*'s method (abbreviated in the tables below as CEP for cepstral coefficients) and our method (abbreviated as MAR for mixtures of AR models).

We first study the simpler scenario with simulated time series data generated by a known number of AR models. We consider two cases separately. The first case involves AR models with the same noise variance, and the second case involves AR models with different noise variances.

In the first experiment, we used two AR(1) models with their AR coefficients uniformly distributed in the ranges $(0.30 \pm 0.01)$ and $(0.60 \pm 0.01)$, respectively. The noise variance was 0.01 for both models. Each model generated 15 time series to form the dataset. As expected, both our MAR method and the CEP method worked very well because the two groups of time series are easily separable. The cluster similarity measure was always equal to 1.

We further conducted more experiments on time series generated by two closer AR(1) models. As before, the AR coefficient of one model was uniformly distributed in the range $(0.30 \pm 0.01)$, but that for the other model was set to four different ranges in four different experiments, varying from $(0.55 \pm 0.01)$ to $(0.40 \pm 0.01)$. In each experiment, each model generated 15 time series to form the dataset. Both methods were run 10 times on each dataset. The minimum, average, and maximum values of the cluster similarity measure over 10 trials were recorded. Table 1 summarizes the results obtained by the two methods. Our method is slightly better than CEP when the two AR(1) models are farther from each other, but CEP becomes slightly better when the range of AR coefficient of one model decreases to $(0.40 \pm 0.01)$, which is very close to that of the other model.

We repeated the experiments above under the same setup, except that the two AR(1) models had the same AR coefficient distribution range of $(0.30 \pm 0.01)$ but different noise variances of 0.01 and 0.02, respectively. Our method gives perfect clustering of the two groups of time series, but CEP, which makes no use of the noise variances, gives very poor results on this dataset with the cluster similarity values being $(0.51/0.59/0.67)$.

In the experiments above the number of component models was specified in advance. We further improve our algorithm by running the basic EM algorithm multiple times with an increasing number of component models until at least one redundant component model is found. When the number of component models specified is equal to or less than the actual number of clusters, the basic EM algorithm will converge. However, if the number of component models specified is larger than the actual number of clusters, the EM algorithm will not converge within a reasonably large number of iterations. Moreover, some component models will learn to become very similar to each other. Based on these characteristics, we can decide whether too many component models are specified. Hence the correct number of clusters can be determined and returned.

**Table 1. Clustering results for time series generated by two AR(1) models with the same noise variance but different AR coefficient distribution ranges**

| Range of AR coefficient | Cluster similarity (min/avg/max) | |
|---|---|---|
| | MAR | CEP |
| $(0.55 \pm 0.01)$ | (0.93/0.99/1.00) | (0.93/0.98/1.00) |
| $(0.50 \pm 0.01)$ | (0.83/0.93/0.97) | (0.80/0.93/0.97) |
| $(0.45 \pm 0.01)$ | (0.80/0.88/0.93) | (0.71/0.86/0.93) |
| $(0.40 \pm 0.01)$ | (0.63/0.77/0.90) | (0.63/0.79/0.93) |

This set of experiments based on simulated datasets allows us to explore the strengths and weaknesses of the two methods under different controlled settings. While our method, like other EM-based methods, generally degrades in clustering performance when the underlying clusters are very close to each other, it is better than Kalpakis *et al.*'s distance-based method under more general situations. Specifically, our method is significantly better when the models have different noise variances. It is also more flexible in determining the number of clusters automatically.

## 6. Results on real datasets

For comparison, we conducted further experiments with the same four real datasets used by Kalpakis *et al.* [5]. The same preprocessing steps used by them were also applied to the datasets to remove the nonstationarity in the data. Moreover, due to differences in level and scale, a normalization step was applied to generate normalized data so that the time series values fall in the range $[0, 1]$. All the experiments were conducted on both normalized and unnormalized data. The cluster similarity values for the four real datasets are shown in Table 2.

Compared with the CEP method, our method can give the same (for two datasets) or better (for another two datasets) results when unnormalized data are used. Our method always works better on unnormalized data because

**Table 2. Clustering results for real datasets**

| Dataset | Normalized | | Unnormalized | |
|---|---|---|---|---|
| | MAR | CEP | MAR | CEP |
| Personal income | 0.78 | 0.84 | 0.90 | 0.84 |
| ECG | 0.80 | 0.94 | 0.94 | 0.94 |
| Temperature | 0.58 | 0.65 | 1.00 | 0.65 |
| Population | 0.62 | 0.64 | 0.64 | 0.64 |

the variance information can be utilized in separating the clusters. However, both our method and the CEP method, due to their nature of modeling stationary ARMA processes only, do not learn the differences in trend of the time series and hence cannot give very satisfactory results for the population dataset. It should be noted, however, that the trends of the two groups of population time series are actually visually distinguishable. Extension of our method to address this issue will be discussed in the next section.

## 7. Conclusion and future work

In this paper, we have proposed a model-based method for clustering univariate ARIMA time series. This mixture-model method, based on mixtures of ARMA models, uses an EM algorithm to learn the mixing coefficients as well as the parameters of the component models. In addition, the number of clusters in the data can be determined automatically. Experimental results on both simulated and real datasets show that this method is generally effective in clustering time series, and that it compares favorably with the hybrid method proposed recently by Kalpakis *et al.* for similar time series clustering problems.

Our method can be improved in a number of aspects. One aspect is related to parameter initialization for the EM algorithm, which may affect the convergence speed of the algorithm and the quality of the solution found. Currently our method sets the initial prior probabilities of the clusters to be equal, and randomly picks $M$ different time series to initialize the $M$ component models. A possible improvement is to initialize the parameters of the mixture model based on the clustering results of some faster but less accurate method. This is analogous to the use of $k$-means for finding the initial parameter values for an EM algorithm.

Computational speedup can be achieved by pruning some models if their posterior probabilities become very close to 0, indicating that their significance is negligible. One problem with our method, like other EM-based methods, is that its clustering performance can degrade significantly when the underlying clusters are very close to each other. A possible extension to model ARIMA time series without removing the nonstationarity may also be explored.

## References


[1] G. Box and G. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden Day, San Francisco, CA, USA, 1970. Revised 1976.

[2] I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White. Visualization of navigation patterns on a web site using model-based clustering. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 280–284, Boston, MA, USA, 20–23 August 2000.

[3] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.

[4] M. Gavrilov, D. Anguelov, P. Indyk, and R. Motwani. Mining the stock market: Which measure is best? In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 487–496, Boston, MA, USA, 20–23 August 2000.

[5] K. Kalpakis, D. Gada, and V. Puttagunta. Distance measures for effective clustering of ARIMA time-series. In *Proceedings of the IEEE International Conference on Data Mining*, pages 273–280, San Jose, CA, USA, 29 November - 2 December 2001.

[6] H. Kwok, C. Chen, and L. Xu. Comparison between mixture of ARMA and mixture of AR model with application to time series forecasting. In *Proceedings of the Fifth International Conference on Neural Information Processing*, pages 1049–1052, Kitakyushu, Japan, 21–23 October 1998.

[7] M. Law and J. Kwok. Rival penalized competitive learning for model-based sequence clustering. In *Proceedings of the Fifteenth International Conference on Pattern Recognition*, volume 2, pages 195–198, Barcelona, Spain, 3–7 September 2000.

[8] C. Li and G. Biswas. A Bayesian approach to temporal data clustering using hidden Markov models. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 543–550, Stanford, CA, USA, 29 June - 2 July 2000.

[9] G. McLachlan and K. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, NY, USA, 1988.

[10] M. Perrone and S. Connell. $K$-means clustering for hidden Markov models. In *Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition*, pages 229–238, Amsterdam, Netherlands, 11–13 September 2000.

[11] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[12] M. Ramoni, P. Sebastiani, and P. Cohen. Bayesian clustering by dynamics. *Machine Learning*, 47(1):91–121, 2002.

[13] R. Redner and H. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239, 1984.

[14] P. Smyth. Clustering sequences with hidden Markov models. In *Advances in Neural Information Processing Systems 9*, pages 648–654. MIT Press, 1997.