Clustering Continuous Time Series

Paola Sebastiani SEBAS@MATH.UMASS.EDU
Department of Mathematics and Statistics, University of Massachusetts, Amherst, MA 01003 USA

Marco Ramoni MARCO_RAMONI@HARVARD.EDU
Children's Hospital Informatics Program, Harvard Medical School, Boston, MA 02115 USA

Abstract

This paper presents a Bayesian algorithm to cluster continuous time series. We assume that the series are generated by stationary processes and can be modeled as autoregressive equations. The algorithm applies an agglomerative clustering procedure to discover the most probable set of clusters describing the essential dynamics of these time series. This search across the exponential space of possible clusters is made feasible by a distance-based heuristic strategy. The algorithm is tested on empirical and real data.

1. Introduction

The time series in Figure 1 describe the annual yield of grain produced between 1852 and 1925 in 17 plots of the Broadbalk field at the Rothamstead experimental station. The left picture depicts the grain yield evolution in the first nine plots, while the right picture depicts the annual yield grain in the last eight plots. The field was divided in 17 plots — the vertical strips in Figure 2 — and each plot was treated with a particular fertilizer. The main object of this experiment was to determine whether grain could be grown continuously by means of artificials alone or with no manure, and also to compare the results obtained by chemicals on the one hand and by farmyard manure on the other. We want to analyze these data in an unsupervised way, by grouping the 17 time series on the basis of their similarity to identify the treatments having the same effect. This is a classical problem in the analysis of time series and the solution depends on the meaning of *similar time series.* A time series is the realization of stochastic processes and we define two or more time series similar when they are generated by the same process. Therefore, deciding whether two time series are similar is equivalent to deciding whether they are observations of the same process. This recognition task can be described as a clustering problem: given a batch of time series, we cluster them so that each cluster contains series generated by the same process. Current approaches treat time series of continuous values as vectors, and they group the series using one of the traditional clustering methods (Kazikawa et al., 1998). These approaches are distance-based, they require the specification of a preset number of clusters or a threshold, and they overlook the correlation between observations.

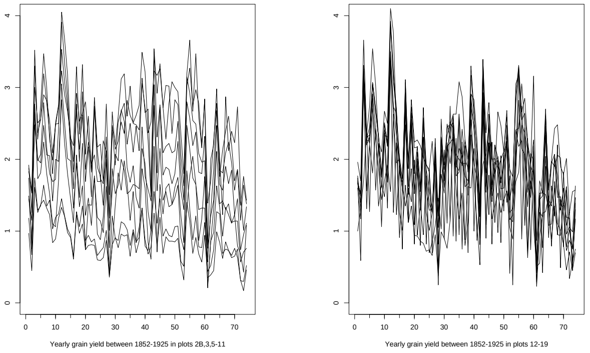

Figure 1. Annual yield of grain produced in plots 2B, 3, 5–11 (left) and plots 12–19 (right) between 1852 and 1925 in the Broadbalk Experiment at the Rothamsted experimental station.

The solution we propose is an algorithm for Bayesian clustering of continuous time series. Given a batch of stationary time series, we model the process generating each of them as an autoregressive model of order p, say AR(p). An AR(p) models the autoregressive structure of a stationary time series as a Markov process of order p (Box & Jenkins, 1976). Then, we cluster those time series with a high posterior probability of being generated by the same AR(p) model. The distinguished feature of our method is to describe clustering as a statistical model so that the clustering task is solved

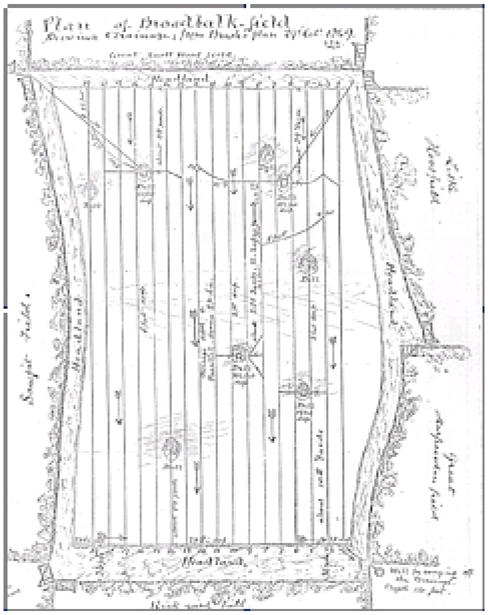

Figure 2. A sketch of the Broadbalk field at the Rothamsted experimental station. The vertical strips represent the 17 plots, labeled 2B, 3, 5–19, from right to left. `http://www.res.bbsrc.ac.uk/era/broadbalk_1.html`.

as a Bayesian model selection problem: the clustering model we look for is the most likely partition of the time series, given the data at hand. In principle, we just need to evaluate the posterior probability of all possible clustering models and select that one with maximum posterior probability. However, the number of clustering models grows exponentially with the number of time series and a heuristic search is needed to make the method feasible. The solution we developed uses a measure of similarity to drive the search process in a subspace of all possible clusters. An important feature of this heuristic search is to provide a stopping rule, so that clustering can be done without assuming a preset number of clusters.

2. Autoregressive Models

Let $S = \{y_{-p}, y_{-p+1}, \ldots, y_t, \ldots y_n\}$ be a stationary time series of continuous values. The series follows an AR(p) model, with mean 0, if the value of the time series at time t is a linear function of the values observed in the previous p steps. More formally:

$$y_t \,|\, \beta = \sum_{j=1}^{p} \beta_j y_{t-j} + \epsilon_t \qquad (1)$$

where $\beta_1, \ldots, \beta_p$ are autoregressive coefficients, and ϵ_t is the error, that we assume normally distributed, with expected value $E(\epsilon_t) = 0$, and variance $V(\epsilon_t) = \sigma^2 = \tau^{-1}$, for any t. The value p is the order of the autoregression, and specifies that y_t is independent of $(y_{-p}, \ldots, y_{t-p-1})$, given $(y_{t-p}, \ldots, y_{t-1})$.

The time series is stationary if it is invariant by temporal translation: if we observe the series in a different time interval, we expect to see the same evolution about the process mean, except for sampling variability. There are several methods to check whether a series is stationary. Particularly, the roots of the polynomial $f(u) = 1 - \sum_{j=1}^{p} \beta_j u^j$ must have moduli greater than unity. If some of the roots have moduli smaller than unity, the process is non-stationary but typically some transformation of the variables is sufficient to transform a non-stationary process into a stationary one (Box & Jenkins, 1976).

The model described by Equation (1) represents the deviation of the process from its mean. To generalize the model to include a non-zero mean μ for each y_t, we can add an intercept term β_0, so that $y_t \,|\, \beta = \beta_0 + \sum_{j=1}^{p} \beta_j y_{t-j} + \epsilon_t$ and $\beta_0 / (1 - \sum_{j=1}^{p} \beta_j)$ is the mean of the process. We will find convenient to write the model in matrix form

$$y \,|\, \beta = X\beta + \epsilon$$

where y is the vector $(y_1, \ldots, y_n)^T$; X is the $n \times q$ matrix with tth row given by $(y_{t-1}, \ldots, y_{t-p})$ for a model with no intercept and $(1, y_{t-1}, \ldots, y_{t-p})$ for a model with intercept term; $\beta = \{\beta_0, \beta_1, \ldots, \beta_p\}$ is a vector of autoregressive coefficients and ϵ is a vector of uncorrelated errors. The value q is either p or $p+1$ according to whether or not there is an intercept term.

Given a series S, two tasks need to be performed: estimating the parameters β and τ from data, and computing the autoregressive order best fitting the data. Both tasks can be done by using Bayesian methods. Suppose first we know p. The Bayesian estimation of the parameters β and τ consists of updating their prior distribution into the posterior distribution by Bayes' Theorem. So, with $f(\beta, \tau)$ denoting the prior density, we need to compute the posterior density which is given by the formula: $f(\beta, \tau | S, p) = f(S | \beta, \tau, p) f(\beta, \tau) / f(S | p)$. The quantity $f(S | \beta, \tau, p)$ is the *likelihood function*, and $f(S|p)$ is the *marginal likelihood*: the former is the joint density of the data, conditional on the parameters β, τ and p, the latter is the averaged likelihood function in which the parameters β and τ are integrated out. With the recommended Jeffrey's prior $f(\beta, \tau) = \tau^{-2}$, $\tau > 0$, (O'Hagan, 1994), we can solve this integration analytically.

Conditional on the first p values of the series, the likelihood function is

$$f(S \mid \beta, \tau) = \sqrt{\frac{\tau^n}{(2\pi)^n}} \exp\left(-\frac{\tau(y - X\beta)^T(y - X\beta)}{2}\right).$$

Suppose X is of full rank, and define $\hat{\beta} = (X^TX)^{-1}X^Ty$ and $\mathsf{rss}= y^T(I_n - X(X^TX)^{-1}X^T)y$. Then, it is well known (Box & Jenkins, 1976) that the marginal likelihood is

$$f(S \mid p) = \frac{\left(\frac{\mathsf{rss}}{2}\right)^{(q+2-n)/2}\Gamma\left(\frac{n-q-2}{2}\right)}{(2\pi)^{(n-q)/2}\det(X^TX)^{1/2}}. \tag{2}$$

The posterior distribution of τ and β is normal-gamma, with $\beta \mid y, \tau \sim N(\hat{\beta}, [\tau(X^TX)]^{-1})$, and $\tau \mid y \sim \text{Gamma}\,(\mathsf{rss}/2, (n-q-2)/2)$. Bayesian estimates of β and σ^2 are $\hat{\beta}$ and $\mathsf{rss}/(n-q-2)$.

Essential to the above solution is the knowledge of the autoregressive order p. Given a series S, we can find the autoregressive order best fitting the data by using a Bayesian model selection approach. Suppose we want to choose the best order p from the set $1, 2, \ldots, k$ and, a priori, each order has probability $P(j)$. Then, the Bayesian solution selects the order p with maximum posterior probability. Routine calculations show that, when the prior probabilities are uniform, the posterior probability $P(j|S)$ is proportional to the marginal likelihood $f(S|p)$, so that the search for the autoregressive order best fitting the data is equivalent to searching for the order with maximum marginal likelihood.

3. Clustering

Suppose now we have a batch of time series $S = \{S_1, S_2, ..., S_m\}$, generated by an unknown number of stationary $\text{AR}(p)$ models with a common autoregressive order p and different autoregressive coefficients. We want to cluster the time series in S according to their dynamics. The task of the algorithm is two-fold: finding the set of clusters that gives the best partition of the data and assigning each time series S_i to one and only one cluster. Contrary to common practice, we do not want to specify, a priori, a preset number of clusters. Formally, the clustering method regards a partition as an unobserved discrete variable C with states $C_1, \ldots, C_c$. Each state C_k of the variable C labels, in the same way, the time series generated by the same $\text{AR}(p)$ model with coefficients β_k and, hence, it represents a cluster of time series. The number c is unknown but it is bounded above by the total number of time series in the data set S. The clustering algorithm tries to re-label those time series that are likely to have been generated by the same $\text{AR}(p)$ model and merges the initial states $C_1, \ldots, C_m$ of the variable C into a subset $C_1, \ldots, C_c$, with $c \leq m$.

The specification of the number c of states of the variable C and the assignment of one of its states to each time series S_i define a statistical model M_c. This allows us to regard the clustering task as a Bayesian model selection problem, in which the model we seek is the most probable way of clustering time series, given the data. If $P(M_c)$ is the prior probability of each model M_c, by Bayes' Theorem its posterior probability is $P(M_c \mid S) \propto P(M_c)f(S \mid M_c)$ and we select the clustering model with maximum posterior probability. We show next that, under some assumptions on the sample space, the adoption of a particular parameterization for the model M_c and the specification of a uniform prior lead to a simple, closed-form expression for the marginal likelihood $f(S \mid M_c)$.

Conditional on the model M_c and, hence, on a specification of the number of states of the variable C and of the labeling of the original time series, we suppose that the marginal distribution of C is multinomial, with cell probabilities $\theta_k = P(C = C_k \mid \theta)$. Furthermore, we suppose that, conditional on $C = C_k$, the batch of m_k time series $\{S_{kj}\}$ in cluster C_k is independent of the batch of time series $\{S_{lj}\}$ in any other cluster C_l, and that the time series generated by the same $\text{AR}(p)$ model in each C_k are mutually independent. We denote by β_k the vector of autoregressive coefficients of the $\text{AR}(p)$ model in cluster C_k, and suppose that each of these series can be represented as

$$y_{kj} \mid \beta_k, \tau_k = X_{kj}\beta_k + \epsilon_{kj}.$$

The index k indicates cluster membership and ϵ_{kj} is a vector of uncorrelated errors, which we assume normally distributed, with $E(\epsilon_{kjt}) = 0$ and $V(\epsilon_{kjt}) = \tau_k^{-1}$, for any t. The fact that series assigned to the same cluster C_k are characterized by the same vector of autoregression coefficients β_k and by the same variance $\sigma_k^2 = \tau_k^{-1}$ allows us to represent the whole batch of series $\{S_{kj}\}$ in cluster C_k as $y_k \mid \beta_k, \tau_k = X_k\beta_k + \epsilon_k$, where the vector y_k and the matrix X_k are defined as

$$y_k = \begin{pmatrix} y_{k1} \\ \vdots \\ y_{km_k} \end{pmatrix} \quad X_k = \begin{pmatrix} X_{k1} \\ \vdots \\ X_{km_k} \end{pmatrix}$$

Let β denote the set of parameter vectors $\{\beta_k\}$, and let τ denotes the set of parameters τ_k for $k = 1, \ldots, c$. Then, by the independence of series assigned to different clusters, the overall likelihood function is $f(S \mid \theta, \beta, \tau) = \prod_{k=1}^{c} \theta_k^{m_k} f(y_k \mid X_k, \beta_k, \tau_k)$, where m_k is the number of series in C_k.

We now define a prior distribution for θ as a Dirichlet $D(\alpha_1, ..., \alpha_c)$ (O'Hagan, 1994), and $f(\beta, \tau) = \prod_k \tau_k^{-2}$ to β and τ. Then, using the result in (2), and standard results on Dirichlet integration, it is easy to show that the marginal likelihood $f(S \mid M_c) = \int f(S \mid \theta, \beta, \tau) f(\theta) f(\beta, \tau) d\theta d\beta d\tau$ is

$$f(S \mid M_c) = \frac{\Gamma(\alpha)}{\Gamma(\alpha + \sum_k m_k)} \times \prod_{k=1}^{c} \frac{\Gamma(\alpha_k + m_k)}{\Gamma(\alpha_k)} \frac{\left(\frac{\mathsf{rss}_k}{2}\right)^{(q+2-n_k)/2} \Gamma\left(\frac{n_k-q-2}{2}\right)}{(2\pi)^{(n_k-q)/2} \det(X_k^T X_k)^{1/2}} \quad (3)$$

where $\alpha = \sum_k \alpha_k$ is the overall cluster prior precision, n_k is the dimension of the vector y_k, and $\mathsf{rss}_k = y_k^T (I_n - X_k (X_k^T X_k)^{-1} X_k^T) y_k$ is the residual sum of squares in cluster C_k.

Once the, *a posteriori*, most likely partition has been selected, each cluster C_k is associated with parameters β_k modeling the autoregression equation, and precision τ_k. The posterior distribution of $\beta_k \mid \tau_k, y_k$ is $N(\hat{\beta}_k, [\tau_k(X_k^T X_k)]^{-1})$, where $\hat{\beta}_k = (X_k^T X_k)^{-1} X_k^T y_k$, while the posterior distribution of $\tau_k \mid y_k$ is Gamma $(\mathsf{rss}_k/2, (n_k - q - 2)/2)$. So, $\hat{\beta}_k$ is the estimate of β_k, and $(n_k - q - 2)/\mathsf{rss}_k$ is the estimate of the within-cluster precision τ_k. The estimate of the probability of $C = C_k$ is $\hat{p}_k = (\alpha_k + n_k)/(\alpha + \sum_k m_k)$.

In practical applications, we use symmetric prior distributions for the parameters θ with a common prior precision α. The initial m hyper-parameters α_k are set equal to α/m and, when two time series are assigned to the same cluster C_k, their hyper-parameters are summed up. Thus, the hyper-parameters of a cluster corresponding to the merging of m_k time series will be $m_k\alpha/m$. In this way, the specification of the prior hyper-parameters requires only the prior global precision α, which measures the confidence in the prior model. The current implementation of the algorithm assumes that the series follow stationary autoregressive models of a given order p and checks stationarity at the end of the clustering process.

4. Search

In principle, the clustering method described in the previous section requires only the computation of the posterior probability of each clustering model and the choice of the model with maximum posterior probability. However, since the number of possible partitions grows exponentially with the number of series, we need a heuristic method to make the search feasible.

Our method uses a measure of similarity between AR(p) models to efficiently guide the search process in a subset of all possible clustering models. Since all AR(p) models have the same order, this similarity measure is an estimate of the symmetric Kullback-Liebler divergence (Jeffreys, 1946) between marginal posterior distributions of the autoregressive coefficients $\beta_k \mid S$. This similarity is estimated by computing the symmetric Kullback-Liebler divergence for every pair of parameters β_k, β_j, assuming normal distribution conditional on the cluster precisions τ_k and τ_j. These precisions are replaced by their posterior estimates.

The algorithm starts by transforming the time series in S into a set of m AR(p) models, using the procedure described in the previous sections, and computes the set of $m(m-1)/2$ pairwise distances between posterior distributions of the parameters. Then, the algorithm merges the two closest AR(p) models and evaluates whether the resulting model M_c, in which the two closest AR(p) models are assigned to the same cluster, is more probable than the model M_s in which they are distinct. If the probability $P(M_c \mid S)$ is larger than $P(M_s \mid S)$, the algorithm updates the set of series by replacing the two series with the cluster resulting from their merging. Consequently, the algorithm updates the ordered set of distances, and the procedure is iterated on the new set. If the probability $P(M_c \mid S)$ is not larger than $P(M_s \mid S)$, the algorithm tries to merge the second best, the third best, and so on, until the set of pairs is empty and, in this case, returns the most probable partition found so far. The rationale behind this heuristic is that merging closest AR(p) models first should speed up the search for the clustering model with maximum posterior probability.

5. Evaluation

To assess the accuracy of the algorithm, we carried out four experiments. In each experiment we generated three batches of 30 time series each, each batch having series of length n, for $n = 25, 50, 100$. In the first experiment, we generated the time series from three AR(3) models with different autoregressive coefficients and different variances. In the second experiment, the generating models were three AR(3) models with different autoregressive coefficients but similar variances. To further increase the similarity of the generated series, in the third experiment the generating models were three AR(3) models with autoregressive coefficients constrained to give the same process mean. Finally, to test the robustness of the algorithm to the common autoregressive order assumption, in the fourth experiment we generated the time series from AR(1), AR(2), and AR(3) models. We then run the clustering algorithm on each batch of time series using five different autoregressive orders: $p = 1, \ldots, 5$, and three values of the global cluster precision: $\alpha = 1, 2, 3$.

	Experiment 1			Experiment 2			Experiment 3			Experiment 4		
	25	50	100	25	50	100	25	50	100	25	50	100
AR(1)	3	5	4	4	5	5	4	4	3	3	3	3
AR(2)	3	3	3	2	3	3	3	3	3	3	3	3
AR(3)	3	3	3	2	3	3	3	3	3	3	3	3
AR(4)	2	3	3	2	3	3	3	3	3	3	3	3
AR(5)	2	3	3	1	3	3	3	3	3	3	3	3
AR(1)	.03	.00	.00	.17	.07	.03	.17	.10	.00	.00	.00	.00
AR(2)	.03	.00	.00	.43	.00	.03	.20	.10	.00	.00	.00	.00
AR(3)	.03	.00	.00	.37	.03	.00	.13	.03	.00	.00	.00	.00
AR(4)	.33	.00	.00	.37	.00	.00	.10	.03	.00	.00	.00	.00
AR(5)	.33	.00	.00	.67	.07	.00	.17	.07	.00	.00	.03	.00

Table 1. Number of clusters found by the algorithm in the 12 experiments (top) and average impurity rate (bottom), for five different autoregressive orders.

We evaluate the algorithm performance using the number of clusters found and an average cluster impurity rate, which is defined as follows. In each experimental condition, the series are generated by three different models, so that a perfect clustering would partition the 30 series in each experimental condition into three groups G_1, G_2 and G_3, each group consisting of series generated by the same model. Therefore, for each cluster C_k found by the algorithm we count the number of series belonging to each of the three groups, say m_{k1}, m_{k2}, and m_{k3}, identify the maximum m_{kj} and label the cluster as group j. The *impurity rate* of cluster C_k is defined as the number $1 - m_{kj} / \sum_i m_{ki}$, and varies between 0 and 2/3. The value 0 is taken when the cluster consists only of series generated by the same group. The maximum is taken when $m_{k1} = m_{k2} = m_{k3}$, so that the cluster mixes series belonging to the three groups in equal proportion, and it is impossible to label the cluster. In the special case in which two of the three groups are equally represented in the cluster, we choose one of the two at random. The *average cluster impurity rate* is then a weighted average of the cluster impurity rates and turns out to be the ratio between the total number of series assigned to the wrong group, and the total number of series in the batch.

The algorithm reproduces essentially the same results for the different choices of α. Summaries of the experimental results are in Table 1, for $\alpha = 1$. In the first experiment, in which 30 series were generated from three AR(3) models, the accuracy of the algorithm to both identify the correct number of clusters and assign the series correctly to each cluster is very good. When the autoregressive order is larger than 1 and the series are at least 50 step long, the algorithm always partitions the series into three clusters, with zero impurity rate, so that each of the three clusters merges series generated from the same model. When the autoregressive order is 1 and the series are 50 step long, the algorithm returns five clusters with zero impurity rate: the ten series $S_1 - S_{10}$ generated by the $\text{AR}(3)_1$ model are partitioned in two clusters $C_1 = \{S_1 - S_6, S_9 - S_{10}\}$ and $C_2 = \{S_7 - S_8\}$; similarly, the ten series $S_{11} - S_{20}$ generated by the $\text{AR}(3)_2$ model are partitioned in two clusters $C_3 = \{S_{11} - S_{13}, S_{15} - S_{17}, S_{19} - S_{20}\}$ and $C_4 = \{S_{14} - S_{18}\}$; the last cluster merges all series $S_{21} - S_{30}$ generated by the $\text{AR}(3)_3$ model. When the series are 100 step long and an order $p = 1$ is used, the algorithm finds four clusters, again with zero impurity rate. Thus, although the algorithm fails to return the correct partition, it does not mix series generated by different models. Only when the series are short, the algorithm is unable to partition the series correctly and two groups of series are merged in the same cluster when the autoregressive order is $p = 4$ or 5 so that the impurity rate in 1/3.

In the second experiment, in which the series were generated from autoregressive models with different coefficients but same variance, the task of the algorithm should be more difficult. When the series are 25 step long, the algorithm is unable to reconstruct the correct partition of time series and merges the series into four clusters when $p = 1$, and two or one clusters when $p > 1$. The impurity rate ranges between .67, when all series are merged into one cluster, and .17, when the series are merged into four clusters and five series are assigned to the wrong cluster. The algorithm partitions correctly the series when they are sufficiently long and the autoregressive order is at least 2. The impurity rate is now slightly larger than in the first experiment, with 1 or 2 series allocated to the wrong cluster. However, the algorithm does a reasonably good job in partitioning the time series and the overall accuracy increases with the length of the series. For example, when $p = 3$ and the series are 25 step long the algorithm finds two clusters: $C_{1,25} = \{S_1 - S_{10}, S_{12}, S_{21} - S_{23}, S_{25} - S_{30}\}$ and $C_{2,25} = \{S_{11}, S_{13} - S_{20}, S_{24}\}$. Thus, $C_{1,25}$ merges all series generated by the $\text{AR}(3)_1$ with one generated by the $\text{AR}(3)_2$ and 9 generate by the $\text{AR}(3)_3$. When the series are 50 time step long, the model consists of three clusters $C_{1,50} = \{S_1 - S_9\}$, $C_{2,50} = \{S_{11} - S_{20}\}$ and $C_{3,50} = \{S_{10}, S_{21} - S_{30}\}$. Thus, the cluster $C_{1,25}$ looses the series S_{12}, now assigned to $C_{2,50}$, and is split in two, $C_{1,50}$ and $C_{3,50}$. Particularly, $C_{3,50}$ absorbs the series S_{24}, previously assigned to $C_{2,25}$. When the series are 100 step long, the algorithm partitions the series in the correct way.

In the third experiment, the generating models have autoregressive coefficients constrained to reproduce the same process means. The algorithm identifies always 3 clusters, for all autoregressive orders greater than 1. The impurity rate decreases with the length of the series: when the series are 25 step long, at most 6 series are assigned to the wrong cluster, with an impurity rate of 0.20; when the series are 50 step long, at most 3 series are assigned to the wrong cluster, and the impurity rate is 0.1. The partition is perfect when the series are 100 step long. In the last experiment, the series were generated from three models with different autoregressive order. The algorithm partitions the series correctly, for every autoregressive order used, except for one case in which one series is assigned to the wrong cluster.

Facts emerging from this small experimental evaluation are a *monotonic discriminatory ability* of the algorithm, that is, an accuracy increasing with the length of the series; a robustness of the algorithm to misspecification of the autoregressive order; and a robustness of the algorithm when the batch consists of series generated by models with different autoregressive orders.

6. Application

We apply the clustering algorithm described in the previous section to the time series of grain yield produced in the seventeen plots of the Broadbalk experiment. The plots are labeled 2B, 3, 5–19, and they correspond to the seventeen vertical strips in Figure 2. Plot 2B is the first strip on the right side of the Figure, followed — from right to left — by plots 3, 5 and so on.

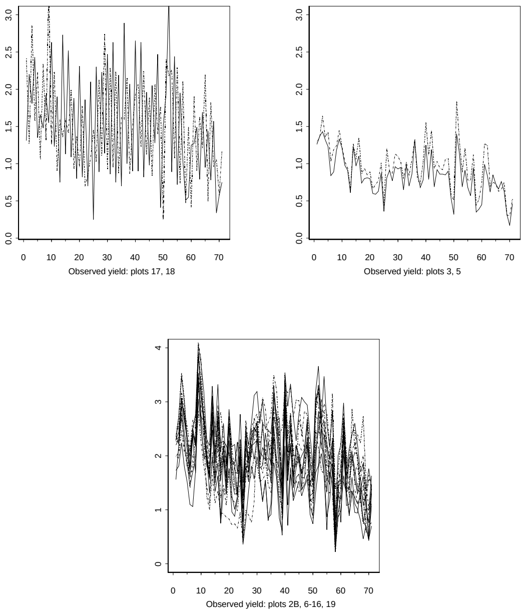

Figure 3. Clusters found by the algorithm for the time series in Figure 1.

We first searched, for each time series, the best autoregressive order fitting the data. This search was carried out by using the Bayesian method described in Section 2. All but two series — those representing the grain yield in plots 17 and 18 — were best fitted by autoregressive models of order 3. For the two remaining series, the best order was four. However, since the majority of the time series was best described by an AR(3) model, we cluster the 17 time series assuming $p = 3$. We also assumed $\alpha = 1$, $f(\beta_k, \tau_k) = \tau^{-2}$, and uniform prior on all clustering models.

Results of the clustering algorithm are displayed in Figure 3. The algorithm finds three clusters: C_1 in the top-left picture, C_2 in the top-middle picture, and C_3 in the top-right picture. Cluster C_1 groups the series of grain yield collected in plots 17 and 18. Estimates of the autoregressive coefficients are $\hat{\beta}_0 \approx 1.13$, $\hat{\beta}_1 \approx -0.05$, $\hat{\beta}_2 \approx 0.52$ and $\hat{\beta}_3 \approx -0.21$, with a mean $\hat{\mu} = 1.53$. Cluster C_2 groups the series of grain yield collected in plots 3 and 5. The estimates of the autoregressive coefficients are $\hat{\beta}_0 \approx 0.33$, $\hat{\beta}_1 \approx 0.36$, $\hat{\beta}_2 \approx 0.10$ and $\hat{\beta}_3 \approx 0.16$, with a mean $\hat{\mu} = 0.9$. Cluster C_3 groups the time series of the yields in the remaining 14 plots. Estimates of the autoregression coefficients are $\hat{\beta}_0 \approx 0.62$, $\hat{\beta}_1 \approx 0.34$, $\hat{\beta}_2 \approx 0.27$ and $\hat{\beta}_3 \approx 0.06$, and mean $\hat{\mu} = 1.88$.

At first glance, the series in each cluster seem to share

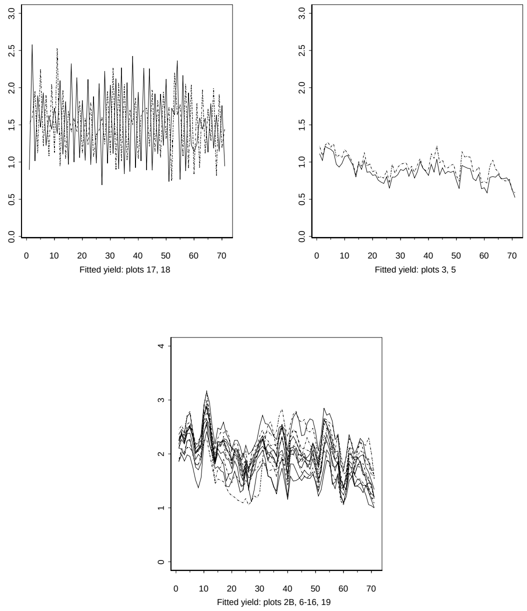

Figure 4. Fitted values for the time series assigned to the three clusters by the algorithm.

the same dynamics. Further information about the treatments of the 17 plots and their positions can help to evaluate the clusters found by the algorithm. First of all we notice that plots 3 and 5 are adjacent, as well as the plots 17 and 18. In both cases, the time series are assigned to the same cluster. Plots 17 and 18 were treated with the same combination of fertilizer, superphosphate, sulphate of potash, magnesium, and nitram in proportions changed seasonally. So, the fact that the time series of the grain yield produced by these two plots are assigned to the same cluster is consistent with the similar treatment received by the two plots.

Plot 3 is the only one which was not treated with any chemical fertilizer, while plot 5 was treated with superphosphate, sulphate of potash, and magnesium. However, some authors point out that adjacent plots could be contaminated (Howard, 1945), so that plot 3 could actually and non-intentionally have received a combination of the treatments given to plots 2B — farmyard manure — and plot 5. This contamination may justify the similar dynamics shown by the two series and their merging in the same cluster. The merging of the time series for the yields in plots 6–16 and 19 into cluster 3 is consistent with the treatment given to the plots: all treatments are a mixture of different chemicals and nitram is always present, although in different proportions. The yield of plot 2B is also merged in the same cluster, this plot is the only one to receive an organic treatment, but the yield evolution is very similar to that of the others assigned to the same cluster. This result is consistent with the view that, under Rothamsted conditions, satisfactory yields of wheat can be obtained by means of chemicals only and that no outstanding advantage follows the use of farmyard manure (Howard, 1945).

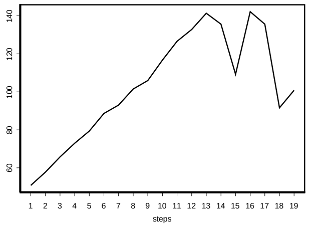

Figure 5. Change of the marginal likelihood, in logarithmic scale, at each step of the agglomerative search procedure.

During the analysis we assumed the time series were all generated by AR(3) models. Plots of the observed and fitted values in Figure 4 provide an overall assessment of the robustness of the result with respect to this assumption. Fitted values reproduce the original time series very satisfactory. Since the search for the best autoregressive order showed that the evolutions of grain yield in plots 17 and 18 were best described by AR(4), we also run our clustering algorithm assuming that the series were generated by AR(4). The clusters found were the same, so that the fact that in the original clusters the two series for plots 17 and 18 were merged would again suggest a robustness of the algorithm to mispecification of the autoregressive order.

Finally, note that the three clusters of time series were found by the search algorithm in just nineteen steps. Figure 5 shows the increase of the log-marginal likelihood at each step of the agglomerative search procedure. In the first fourteen steps there is a linear increase of the marginal likelihood. Thus, merging the time series which belong to the clusters with nearest autoregressive coefficients increases the marginal likelihood. In the next steps merging the closest clusters does not always increase the marginal likelihood, so that the merging of the second nearest clusters is evaluated and accepted. This is so until step 17, when the algorithm has merged the fourteen time series into three clusters. At this point, the three possible merging of two clusters at a time are evaluated and, since they all result in a decrease of the marginal likelihood, the algorithm stops and returns the three clusters so found.

7. Discussion and Related Work

Both the evaluation on synthetic data and the previous application show that our clustering algorithm does a good job in summarizing a batch of time series into clusters characterized by similar dynamics. The summary is achieved by using simple models — autoregressive equations — which are able to capture the essential dynamics of the original time series.

An essential feature of our clustering method is that it is model-based: a clustering is represented as a statistical model and the search for the best clustering of time series becomes a model selection problem. Traditional clustering methods are distance-based so that the decision as to whether merging two time series into the same cluster would require to treat a time series as a vector, and to merge two time series if their distance exceeds a prespecified threshold. In our algorithm, the distance between parameter estimates is used only to drive the search process, while the decision of merging is based on the posterior probability of the clustering model. In fact, our application shows that the distance-based clustering would fail to reproduce the partition found by our algorithm.

Model-based clustering of time series has been proposed by Ridgeway (1997) and Cadez et al. (2000), and their approach does not use a threshold to decide when merging, but requires a preset number of clusters. Our approach differs in three main aspects. By explicitly modeling cluster membership, our algorithm does not use mixture models and, therefore, does not need iterative procedures to compute a scoring metric for a set of clusters. By using an agglomerative clustering procedure and a heuristic search, our algorithm does not need a preset number of clusters and, furthermore, the scoring metric used by our algorithm is the marginal likelihood and not the maximized likelihood function. When applied to cluster discrete Markov chains (Ramoni et al., 2000a), our clustering procedure proved to be superior to the mixture-model approach of Cadez et al. (2000), and we expect this higher accuracy to hold even for continuous time series, as long as autoregressive models provide a reasonable approximation of the dynamics of the time series.

Current limitations of our new algorithm are the fact that it is limited to univariate time series, and assumes AR(p) models with a common order p. We anticipate that the algorithm can be generalized to multivariate time series by generalizing the approach in Ramoni et al. (2000b). At first glance, modeling time series as autoregressive equations may appear to limit the applicability of our method. Often, however, non-stationary time series can be transformed into stationary ones by making some transformation as, for example, computing the first differences of the log-transformed series. The empirical evaluations we carried out suggest a robustness of the algorithm with respect the the "common p" assumption, at least when the batch consists of series of similar orders. In practical applications, analysis of the fitted values or residual sum of squares provides an overall assessment of the results and a comparative measure for sets of clusters found with different autoregressive orders.

References

Box, G. E. P., & Jenkins, G. M. (1976). *Time series analysis: Forecasting and control.* San Francisco, CA: Holden-Day. 2nd edition.

Cadez, I., Gaffney, S., & Smyth, P. (2000). A general probabilistic framework for clustering individuals. *ACM KDD-2000 Proceedings.*

Howard, A. (1945). *Farming and gardening for health or disease.* London, UK: Faber and Faber. (with the assistance of Louise E. Howard).

Jeffreys, H. (1946). An invariant form for the prior probability in estimation procedures. *Proceedings of the Royal Society, London, A*, *186*, 453–461.

Kazikawa, Y., Shumway, R. H., & Taniguchi, M. (1998). Discrimination and clustering for multivariate time series. *Jornal of the American Statistical Association*, *93*, 310–340.

O'Hagan, A. (1994). *Bayesian inference.* Kendall's Advanced Theory of Statistics. London, UK: Arnold.

Ramoni, M., Sebastiani, P., & Cohen, P. R. (2000a). Bayesian clustering by dynamics. Submitted.

Ramoni, M., Sebastiani, P., & Cohen, P. R. (2000b). Multivariate clustering by dynamics. *Proceedings of the Seventeeth National Conference on Artificial Intelligence* (pp. 633–638). San Francisco, CA: Morgan Kaufmann.

Ridgeway, G. (1997). *Finite discrete Markov process clustering* (Technical Report MSR-TR-97-24). Microsoft Research, Redmond, WA.