Appendix 1

Goodness of Fit Measures

This appendix introduces three types of goodness of fit measures for clustering models, which evaluate the goodness of the fitted model for individual series, for individual clusters, and for the entire clustering model, respectively. By providing a global goodness of fit measure of the clustering model, these scores can also be used as a "safety" measure against the fact that the algorithm does not explore exhaustively the space of clustering models but it follows a heuristic search strategy.

The intuition behind the scores is that, by summarizing a batch of time series into groups, clustering induces a loss of the information conveyed by the data. Such a loss of information has two main components: one is caused by the autoregressive assumption, while the other component is caused by the merging of time series into clusters. Both components can be accounted for by using log-scores, which were originally introduced by ref. 1 for assessing the predictive capability of a probability distribution, and are becoming increasingly popular as model assessment tools refs 2 and 3.

Suppose the model M_c returned by the algorithm consists of c clusters, $C_1, \ldots, C_c$, each C_k merging m_k series. Then, each series assigned to cluster C_k is modeled as an autoregressive equation, with coefficients β_k estimated by $\hat{\beta}_k = (X_k^T X_k)^{-1} X_k^T y_k$, and variance σ_k^2 estimated by $\hat{\sigma}_k^2 = \text{RSS}_k/(n_k - q - \gamma)$. Therefore, for each series S_h assigned to this cluster C_k, the value $y_{h,i}$ of the series at the ith time step, conditional on the past p values, has a normal distribution with mean $\hat{y}_{k,h,i} = \hat{\beta}_{k,0} + \sum_j \hat{\beta}_{k,j} y_{h,i-j}$ and variance $\hat{\sigma}_k^2$. The log-score for this value is minus the logarithm of its density function and is proportional to

$$s_{k,h,i} = \log(2\pi) + \log(\hat{\sigma}_k^2) + \hat{\sigma}_k^2 (y_{k,h,i} - \hat{y}_{k,h,i})^2.$$

By summing over the i, we obtain the *cumulative series score*

$$s_{k,h} = n_h - p \log(2\pi) + n_h - p \log(\hat{\sigma}_k^2) + \frac{1}{\hat{\sigma}_k^2} \sum_i (y_{h,i} - \hat{y}_{k,h,i})^2.$$

The cumulative score penalizes the goodness of fit of the autoregressive equation, with large scores indicating poor performance and a large loss of data information. This quantity can therefore be used as a comparative measure, across different cluster sets. We can also define a monitoring measure, to detect series that are possibly misallocated by the algorithm. The monitoring measure is built by summing the standardized scores. It is easy to show that the expected value of $s_{k,h,i}$, where the expectation is taken with respect to the distribution of $y_{h,i}$, is

$$E(s_{k,h,i}) = \log(2\pi) + \log(\hat{\sigma}_k^2),$$

and the variance is $V(s_{k,h,i}) = 1/2$, so that

$$z_{k,h,i} = \sqrt{2}(s_{k,h,i} - E(s_{k,h,i})) = \frac{\sqrt{2}}{\hat{\sigma}_k^2}(y_{h,i} - \hat{y}_{k,h,i})^2 - 1)$$

is the standardized log-score of the series S_h at time i. By definition, $z_{k,h,i}$ has zero expectation and unit variance. By summing the standardized scores over the $n_h - p$ free values of the series, we compute the *series monitor*

$$z_{k,h} = \frac{\sqrt{2}}{\hat{\sigma}_k^2} \left(\sum_i (y_{h,i} - \hat{y}_{k,h,i})^2 - (n_h - p) \right),$$

which measures the ability of the autoregressive model in cluster C_k to reproduce the observed series S_h. For large $n_h - p$, we can use the results in ref. 4 to show that the series monitor has approximately

a normal distribution, with zero expectation and variance $(n_h - p)$. For a fixed significance level a, one can then use values of $z_{k,h}$ within the limits $\pm\sqrt{n_h - p}z_{a/2}$, where $z_{a/2}$ is the $(1 - a/2)$ percentile of the standard normal distribution, as indication of goodness of fit of the cluster C_k autoregressive model for the series S_h. On the other hand, values of $z_{k,h}$ outside these limits signal the series on which either the autoregressive assumption or the cluster assignment fail to provide a good fitting. In this case, a visual inspection of the signaled series can suggest reasons for the lack of fit.

By summing the series score over the m_k series assigned to cluster C_k, we obtain

$$s_k = \sum_h s_{k,h} = \frac{1}{\hat{\sigma}_k^2} \sum_{i,k} (y_{h,i} - \hat{y}_{k,h,i})^2 + n_k \log(2\pi) + n_k \log(\hat{\sigma}_k^2),$$

where n_k is the length of the vector y_k in cluster C_k and is therefore $\sum_h (n_h - p)$. Since $\sum_{i,k}(y_{h,i} - \hat{y}_{k,h,i})^2 = (n_k - q - \gamma)\hat{\sigma}_k^2)$, the above score simplifies to the *cluster score*

$$s_k = n_k(1 + \log(2\pi) + \log(\hat{\sigma}_k^2)) - q - \gamma,$$

from which, by summing over the c clusters, we compute the *model score*

$$s_m = \sum_k s_k = -c(q + \gamma) + (1 + \log(2\pi)) \sum_k n_k + \sum n_k \log(\hat{\sigma}_k^2).$$

By writing $\hat{\sigma}_k^2 = \text{RSS}_k/(n_k - q - \gamma)$, the score s_m becomes

$$s_m = -c(q + \gamma) + (1 + \log(2\pi)) \sum_k n_k - \sum n_k \log(n_k - q - \gamma) + \sum n_k \log(\text{RSS}_k)$$

and measures the loss of data information of the clustering model. When the prior is uniform, as it happens in our case, the score s_m becomes

$$s = cq + \sum_k n_k[\log(n_k - q) - \log(\text{RSS}_k)] - (1 + \log(2\pi)) \sum_k n_k.$$

1. Good, I. J. (1952) *J. Roy. Stat. Soc. B* **14**, 107–114.
2. Bernardo, J. M. & Smith, A. F. M. (1994) *Bayesian Theory* (Wiley, New York).
3. Cowell, R. G., Dawid, A. P., Lauritzen, S. L. & Spiegelhalter, D. J. (1999) *Probabilistic Networks and Expert Systems* (Springer, New York).
4. Seillier-Moiseiwitsch, F. & Dawid, A.P. (1993) *J. Amer. Statist. Assoc.* **88**, 355–359.

Appendix 2

Experimental Evaluation

This appendix presents the results of a controlled experiment to evaluate the accuracy of the algorithm when it is applied to a batch of time series generated by different AR(p) models. The results show an overall accuracy that appears to suffer only when the time series in the batch are very short and have very similar dynamics.

Methods

To assess the accuracy of the algorithm, we carried out four experiments and varied two factors in each of them. In the first experiment, we generated 30 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. Table 2 gives the parameter specification of the AR(p) models used in each experiment.

The factors we varied were the length of each time series, 25, 50, and 100 time steps, and the number of time series generated by each AR(p) model, either 10 series from each generating model or 5, 10, and 15. These two factors yield 6 conditions for each experiment, for a total of 24 batches of time series. In the last experiment, in which series were unevenly generated by models with 3 different autoregressive orders, we generated 15 series from the AR(1) model, 10 from the AR(2) model, and 5 from the AR(3) model.

We then run the clustering algorithm on each batch of time series using five different autoregressive orders, $p = 1, \ldots, 5$, and used the model score defined in Appendix 1 to identify the clustering model with minimum score in each set of five. To assess the effect of the prior hyper-parameters, we repeated the procedure with three different priors on β, τ: $\gamma = 1, 2, 3$; and three values of the global cluster precision: $\alpha = 1, 2, 3$. We evaluate the algorithm performance using the number of clusters found and an average cluster impurity rate, 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 is

$$r = \frac{\sum_k m_k (1 - m_{kj}/\sum_i m_{ki})}{\sum_k m_k},$$

where m_k is the number of series in cluster C_k. Since $\sum_i m_{ki} = m_k$, the quantity r is simply the ratio between the total number of series assigned to the wrong group, and the total number of series in the batch.

Table 2: Parameter specification of the AR(p) models used in the four experiments.

Model	Experiment 1					Experiment 2				
parameters	β_0	β_1	β_2	β_3	σ^2	β_0	β_1	β_2	β_3	σ^2
AR(3)$_1$	1.13	-0.05	0.52	-0.21	0.26	1.13	-0.05	0.52	-0.21	0.27
AR(3)$_2$	0.33	0.36	0.10	0.16	0.07	0.33	0.36	0.10	0.16	0.27
AR(3)$_3$	0.62	0.34	0.27	0.06	0.34	0.62	0.34	0.27	0.06	0.27
Model	Experiment 3					Experiment 4				
parameters	β_0	β_1	β_2	β_3	σ^2	β_0	β_1	β_2	β_3	σ^2
AR(3)$_1$	1.13	-0.05	0.52	-0.21	0.26	1.13	-0.05	-	-	0.27
AR(3)$_2$	0.57	0.36	0.10	0.16	0.07	0.33	0.36	0.10	-	0.07
AR(3)$_3$	0.50	0.34	0.27	0.06	0.34	0.62	0.34	0.27	0.06	0.34

Results

The algorithm reproduces essentially the same results for all different choices of α and γ. Summaries of the experimental results are in Tables 3 and 4, for $\alpha = 1$ and $\gamma = 2$. Table 3 reports the number of clusters found by the algorithm for each of the five autoregressive orders, in each of the 24 experiments. Table 4 reports the average cluster impurity rate for each clustering model found with one of the five autoregressive models, in each of the 24 experimental conditions. In both tables, figures in bold face in each column denote the number of clusters and the average impurity rate of the model selected by the model score in each experimental condition.

Experiment 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 steps long, the algorithm partitions the series into three clusters for all autoregressive orders, in both the balanced and unbalanced case. In all cases, the impurity rate is zero, so that each of the three clusters merges series generated from the same model. In the balanced case, when the autoregressive order is 1 and the series are 50 steps long, the algorithm returns five clusters with zero impurity rate: the 10 series $S_1 - S_{10}$ generated by the 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 10 series $S_{11} - S_{20}$ generated by the 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 AR(3)$_3$ model. When the series are 100 steps long and an order $p = 1$ is used, the algorithm finds four clusters in the balanced case and five clusters in the unbalanced case, 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, in the balanced case, 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 unbalanced case, the impurity rate is larger, as the number of series assigned to the wrong cluster increases. For example, when the autoregressive order is $p = 3$, and 5 series of length 25 are generated from the AR(3)$_1$ model, 10 from the AR(3)$_2$ model, and 15 from the AR(3)$_3$ model, the algorithm finds two clusters: one merging the 5 series generated

Table 3: Number of clusters found by the algorithm in the 24 experiments. The top half of the table reports the results for the balanced case, and the bottom half reports the results for the unbalanced case. Each row reports the number of clusters found by the algorithm for the autoregressive order specified in the first column. Figures in bold face are the cluster models selected by the model scores from the clustering models found with the five different autoregressive orders.

	Experiment 1			Experiment 2			Experiment 3			Experiment 4		
Balanced	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
Unbalanced	25	50	100	25	50	100	25	50	100	25	50	100
AR(1)	3	3	5	3	4	5	3	4	4	3	3	3
AR(2)	2	3	3	**3**	3	3	3	3	3	**3**	**3**	**3**
AR(3)	2	3	**3**	2	3	**3**	4	**4**	**3**	3	3	3
AR(4)	2	**3**	3	2	**3**	3	3	4	3	3	3	3
AR(5)	**2**	3	3	2	3	3	**4**	5	3	3	3	3

from the $\text{AR}(3)_1$ model with the 15 series generated from the $\text{AR}(3)_3$ model and one series generated from the $\text{AR}(3)_2$ model; the second cluster merges the remaining 9 series generated from the $\text{AR}(3)_2$ model. As the length of the series increases, the correct partition is identified.

On average, the algorithm exhibits a robustness with respect to misspecification of the autoregressive order. Furthermore, when the model score is used to compare the different clustering models found for different autoregressive order and to select one, the correct partition is always identified in the balanced case. In the unbalanced case, the model score selects the clustering model found with $p = 5$, when the series are short, and the clustering model found with $p = 4$ when the series are 50 steps long. When the series are 100 steps long, the model score identifies the correct partition. Interestingly, the clustering model identified by the model score when the series are 25 steps long has score -2.52, while the score of the clustering model when $p = 2$ is -2.45, it is -2.49 for $p = 3$ and -2.48 for $p = 4$. Thus, although both clustering models found for $p = 2$ and $p = 5$ merge the series generated by the $\text{AR}(3)_1$ and the $\text{AR}(3)_3$ models in one cluster, the adoption of an autoregressive order $p = 5$ is less lossy. The scores of the clustering models found with $p = 3, 4$ are larger than those of the model found with $p = 2, 5$, and reflect the larger impurity rates.

Experiment 2. 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. The algorithm partitions correctly the series when they are sufficiently long, and the model score signals the correct partition in either the balanced or unbalanced case. When the series are only 50 steps long, the performance of the algorithm is similar to that in the first experiment: in either the balanced or unbalanced case, the number of clusters is 3 when 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 in

Table 4: Average impurity rate in the 24 experiments. Figures in bold face are the average impurity rates of the clusters selected by the model score from the clustering models found with the five different autoregressive orders.

	Experiment 1			Experiment 2			Experiment 3			Experiment 4		
Balanced	25	50	100	25	50	100	25	50	100	25	50	100
AR(1)	0.03	0.00	0.00	0.17	0.07	0.03	0.17	0.10	0.00	0.00	0.00	0.00
AR(2)	0.03	0.00	0.00	0.43	0.00	0.03	0.20	0.10	0.00	0.00	0.00	0.00
AR(3)	**0.03**	**0.00**	**0.00**	**0.37**	**0.03**	**0.00**	0.13	0.03	0.00	**0.00**	**0.00**	**0.00**
AR(4)	0.33	0.00	0.00	0.37	0.00	0.00	**0.10**	**0.03**	**0.00**	0.00	0.00	0.00
AR(5)	0.33	0.00	0.00	0.67	0.07	0.00	0.17	0.07	0.00	0.00	0.03	0.00
Unbalanced	25	50	100	25	50	100	25	50	100	25	50	100
AR(1)	0.07	0.03	0.00	0.10	0.10	0.00	0.07	0.07	0.00	0.00	0.00	0.00
AR(2)	0.17	0.03	0.00	**0.07**	0.10	0.00	0.17	0.07	0.00	**0.00**	**0.00**	**0.00**
AR(3)	0.20	0.00	**0.00**	0.37	0.10	**0.00**	0.17	**0.07**	**0.00**	0.00	0.00	0.00
AR(4)	0.23	**0.00**	0.00	0.53	**0.03**	0.00	0.20	0.07	0.00	0.03	0.00	0.00
AR(5)	**0.17**	0.00	0.00	0.40	0.17	0.00	**0.20**	0.03	0.00	0.03	0.00	0.00

the balanced case, and up to 5 series allocated to the wrong cluster in the unbalanced case. The model score identifies the correct order in the balanced case, and chooses the order $p = 4$ in the unbalanced case, which corresponds to the clustering model with the smallest impurity rate.

When the series are 25 steps long, the algorithm is unable to reconstruct the correct partition of time series in the balanced case, and merges the series into four clusters when $p = 1$, and two or one clusters when $p > 1$. The impurity rate ranges between 0.67, when all series are merged into one cluster, and 0.17, when the series are merged into four clusters and five series are assigned to the wrong cluster. In the unbalanced case, the number of clusters found by the algorithm is two or three. The model score identifies the correct autoregressive order in the balanced case, and the order $p = 2$ in the unbalanced case: this is the clustering model with the smallest impurity rate. Again, the algorithm exhibits an accuracy that increases with the length of the series and a robustness to misspecification of the autoregressive order. When coupled with the model score to select the clustering model with the best fit, the whole procedure does a reasonably good job in partitioning the time series and the overall accuracy increases with the length of the series. For example, the clustering model selected by the model score in the balanced case when the series are 25 steps long consists of 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 steps long, the clustering model selected by the model score 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 $C_{1,25}$ loses 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 steps long, the algorithm partitions the series in the correct way, and the model score identifies the correct autoregressive order.

Experiment 3. In the third experiment, the generating models have autoregressive coefficients constrained to reproduce the same process means. In the balanced case, 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 steps long, at most 6 series are assigned to the wrong cluster, with an impurity rate of 0.20; when the series are 50 steps 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 steps long. The model score signals $p = 4$ as best autoregressive order in all three sets of series of different length and this is the unique case in which the score fails to identify the correct partition even when the series are long. In the unbalanced case, the algorithm creates the correct partition when the series are 100 steps long and the model score signals the correct autoregressive order, but it fails to reproduce the correct partition when the series are 25 or 50 steps long. The impurity rate decreases with the length of the series, and the model score identifies the correct autoregressive order with series of length 50, while it favors larger orders with short series.

Experiment 4. In the last experiment, the series were generated from three models with different autoregressive order. In the balanced case, 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. Interestingly, the model score selects, in all cases, the clustering model fitted with order $p = 3$. In the unbalanced case, the number of cluster created by the algorithm is always three, for all five autoregressive orders, and in two cases one series is assigned to the wrong cluster. Now the model score signals the clustering models found with order 2 as those giving the best fit. In all three experimental conditions, we generated 15 series from the AR(1), 10 from the AR(2), and 5 from the AR(3), so that the model scores identify an average order. 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 the ability of the method to handle time series generated by models with different autoregressive orders. The model score seems to be very effective to signal the best partition returned by the algorithm for different autoregressive orders, particularly when the series are reasonably long.