Department of Economics

Cluster Analysis of Panel Datasets Using Non-Standard Optimisation of Information Criteria

George Kapetanios

Working Paper No. 535 May 2005 ISSN 1473-0278

Cluster Analysis of Panel Datasets using Non-Standard Optimisation of Information Criteria

George Kapetanios*
Queen Mary, University of London

November 26, 2004

Abstract

Panel datasets have been increasingly used in economics to analyse complex economic phenomena. One of the attractions of panel datasets is the ability to use an extended dataset to obtain information about parameters of interest which are assumed to have common values across panel units. However, the assumption of poolability has not been studied extensively beyond tests that determine whether a given dataset is poolable. We propose an information criterion method that enables the distinction of a set of series into a set of poolable series for which the hypothesis of a common parameter subvector cannot be reject and a set of series for which the poolability hypothesis fails. The method can be extended to analyse datasets with multiple clusters of series with similar characteristics. We discuss the theoretical properties of the method and investigate its small sample performance in a Monte Carlo study.

Keywords: Panel datasets, Poolability, Information Criteria, Genetic Algorithm, Simulated Annealing

JEL Codes: C12, C15, C23

*Department of Economics, Queen Mary, University of London, Mile End Road, London E1 4NS. email: G.Kapetanios@qmul.ac.uk

1 Introduction

Panel datasets have been used extensively in the econometric literature to enable more accurate analysis of complex economic phenomena. However, the validity of the assumption of poolability, i.e. the validity of the assumption that panel units described by a given model have a common parameter subvector for that model, has not received great attention in the literature. Work in this area has concentrated on whether a given dataset is poolable as a whole, i.e, whether the null hypothesis $H_0 : \beta_j = \beta$, $j = 1, \ldots, N$ holds, where β is the assumed common parameter subvector of the N cross-sectional units of the dataset. In that vein a common approach, discussed, in some detail, in Baltagi (2001), is to use an extension of the Chow (1960) parameter stability test on the pooled dataset. Other tests for this null hypothesis have been developed by Ziemer and Wetzstein (1983) and Baltagi, Hidalgo, and Li (1996).

However, if such tests reject the researcher is left with little idea of how to proceed. In other words if we reject this null hypothesis we do not know which series caused the rejection. It would be of some interest if a method were available that would enable the distinction of the set of series into a group of poolable and a group of nonpoolable series. Such methods seem indeed possible and this paper is proposing one. Our method uses an information criterion search to distinguish between poolable and nonpoolable series. If more than one series are actually poolable then the use of panel methods to investigate the properties of this set of series is indeed more efficient compared to univariate methods.

An alternative method that sorts poolable from nonpoolable series has been recently suggested by Kapetanios (2003). That paper used a sequence

of tests to achieve the sorting. The current methodology is useful in a more general context. A possibility that has only recently been seriously considered in the econometric literature is the possibility that panel datasets are in fact made up of smaller panels or clusters of series with the same characteristics (see e.g. Paap, Frances, and Van Dijk (2003)). Methodologies such as latent panel analysis may be useful but may face computational problems for large datasets.

Our suggested methodology is relatively simple. For a given split of the panel datasets into series that belong to a cluster and series that do not we can easily estimate the cluster model and the individual models for the series that are assumed not to belong to the cluster, and get the value of an information criterion. This defines a mapping from a given split of the panel series to the value of an information criterion. Then, the chosen split is the one that maximises an information criterion. This however poses a serious computational problem. For a set of N series there exist 2^N splits and therefore the problem quickly becomes impossible if all splits are to be evaluated. To bypass this problem we suggest the use of nonstandard optimisation algorithms. We suggest simulated annealing and genetic maximisation. These are especially well suited to dealing with problems of optimising functions whose domains are discrete sets. Further, we extend the analysis to splits involving more than one clusters to which each series may belong. The problem is conceptually similar to that of one cluster.

Our approach is related to a large literature on clustering in other disciplines. We note the work by Cantu-Paz and Kamath (2003), who use similar algorithms for determining decision trees in computer science, and by Kakazawa, Shumway, and Taniguchi (1998) on clustering with an application to seismology. In economics we note the work of Durlauf and Johnson (1995)

on the classification of growth regressions and the work of Vahid (1999) on an alternative clustering algorithm for panel datasets.

The paper is organised as follows: Section 2 sets out in detail the problem we would like to address. Section 3 presents details on the maximisation algorithms we consider. Section 4 presents a Monte Carlo exercise. Finally, Section 5 concludes.

2 Theory

Let us consider the following panel data model

$$y_{j,t} = \alpha_j + \beta_j x_{j,t} + \epsilon_{j,t}, \quad j = 1, \ldots, N, \quad t = 1, \ldots, T. \tag{1}$$

where $x_{j,t}$ is a k-dimensional vector of predetermined variables. This is a standard panel data model where we do not need to specify the nature of the cross sectional individual effect α_j. Our discussion carries through both for fixed and random effect models. We would like to investigate the restriction

$$\beta_j = \beta, \quad \forall j \tag{2}$$

We now define the object we wish to estimate. To simplify the analysis we assume that there exists one cluster of series with equal $\beta_j = \beta$. For the time being we will assume that there exists just one cluster of series with equal β_j and all the rest of the series have different β_j. The more general case is straightforward to deal with and will be discussed later. For every series $y_{j,t}$ (and associated set of predetermined variables $x_{j,t}$) define the binary object $\mathcal{I}_j$ which takes the value 0 if $\beta_j = \beta$ and 1 if $\beta_j \neq \beta$. Then, $\boldsymbol{\mathcal{I}} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)'$. We wish to estimate $\boldsymbol{\mathcal{I}}^0$ where $\boldsymbol{\mathcal{I}}^0$ denotes the true split. We denote the estimate by $\hat{\boldsymbol{\mathcal{I}}}$.

For every possible split of the dataset, $\boldsymbol{\mathcal{I}}$, one can obtain the value of an information criterion. The generic form of such a criterion is usually

$$IC(\boldsymbol{\mathcal{I}}) = 2L(\boldsymbol{\mathcal{I}}) - C_T(\boldsymbol{\mathcal{I}}) \tag{3}$$

where $L(\boldsymbol{\mathcal{I}})$ is the log-likelihood of the model associated with the split $\boldsymbol{\mathcal{I}}$ and $C_T(\boldsymbol{\mathcal{I}})$ is the penalty term associated with it. The three most usual penalty terms are $2\tilde{m}$, $ln(T)\tilde{m}$ and $2ln(ln(T))\tilde{m}$ associated with the Akaike, Schwartz and Hannan-Quinn information criteria. $\tilde{m}$ is the number of free parameters associated with the modelling of the whole dataset. Note that $\tilde{m}$ depends on $\boldsymbol{\mathcal{I}}$. For example, if for some $\boldsymbol{\mathcal{I}}$ there are N_1 series belonging to a cluster and $N_2 = N - N_1$ series which should be modelled individually, then $\tilde{m} = (N_2 + 1)k$. It is straightforward under relatively weak conditions on $x_{j,t}$ and $\epsilon_{j,t}$, and using the results of say, Sin and White (1996), to show that the split which maximises $IC(.)$ will converge to $\boldsymbol{\mathcal{I}}^0$ with probability approaching one as $T \to \infty$ as long as $C_T(\boldsymbol{\mathcal{I}}) \to \infty$ and $C_T(\boldsymbol{\mathcal{I}})/T \to 0$.

More specifically, the assumptions needed for the results of Sin and White (1996) to hold are mild and can be summarised as follows, assuming estimation of the models is undertaken in the context of Gaussian or pseudo maximum likelihood: (i) Assumption A of Sin and White (1996) requires measurability, continuity and twice differentiability of the log-likelihood function and a standard identifiability assumption; (ii) A uniform weak law of large numbers for the log-likelihood of each observation and its second derivative; (iii) A central limit theorem for the first derivative of the log-likelihood of each observation. (ii) and (iii) above can be obtained by assuming, e.g., that $x_{j,t}$ are weakly dependent, say, near epoque dependent, processes and $\epsilon_{j,t}$ are martingale difference processes. Hence, it is clear that consistency of model selection as long as the penalty related conditions hold is straightforwardly obtained.

The problem is of course how to maximise the information criterion. For small panels, evaluating the information criterion for all splits may be feasible. But as soon as N exceeds say 30 or 40 units, this strategy is bound to fail. Since, $\boldsymbol{\mathcal{I}}$ is a binary sequence there exist 2^N splits to be evaluated. For example, when $N = 50$ and optimistically assuming that 100000 splits can be evaluated per second, we still need about 357 years for an evaluation of all splits. We may alternatively treat this as a maximisation problem. Nevertheless, clearly standard maximisation algorithms do not apply. We resort to two very powerful non-standard maximisation algorithm classes: simulated annealing and genetic algorithms. These is discussed in the next section.

The above method can be extended to more complicated problems where there may be more than one clusters. We next see how we can go about formally casting the problem. Assume the presence of k clusters each with a different value of β, say $\beta^{(i)}$, $i = 1, \ldots, k$. Then, we define the object $\tilde{\mathcal{I}}_j$ which takes the value i if the unit j belongs to the i-th cluster, i.e. $\beta_j = \beta^{(i)}$, or zero if the unit j does not belong to a cluster. Then, we want to estimate $\tilde{\boldsymbol{\mathcal{I}}} = (\tilde{\mathcal{I}}_1, \ldots, \tilde{\mathcal{I}}_N)'$. Once again we wish to maximise an information criterion. Note that the number of splits now is given by $(k+1)^N$. Hence, the optimisation problem is even more difficult. The above assumes that we know the number of clusters. This may be considered restrictive. An obvious extension is to maximise the information criterion for every k clusters and then choose the value of k that maximises the criterion over $k = 1, \ldots, K$ for some maximum number of clusters K.

3 Nonstandard Optimisation Algorithms

In the previous section we saw how we translated the problem of clustering individual series from panel datasets to a problem of maximising an information criterion. On the one hand the space where the information criterion is defined is discrete and hence standard optimisation methods cannot be applied. On the other hand, standard grid search which is usually implemented to maximise the information criterion, as in, e.g., lag selection, is clearly infeasible due to the computational burden of the problem. One alternative is to resort to nonstandard optimisation algorithms that do not require neither smoothness nor continuity for the algorithm to converge.

3.1 Simulated Annealing

Simulated annealing is a generic term used to refer to a family of powerful optimisation algorithms. In essence, it is a method that uses the objective function to create a nonhomogeneous Markov chain that asymptotically converges to the maximum of the objective function. It is especially well suited for functions defined in discrete spaces like the information criteria considered here. Below, we give a description of the algorithm together with the necessary arguments that illustrate its validity in our context. We describe the operation of the algorithm when the domain of the function (information criterion) is the set of binary strings i.e. $\{\boldsymbol{\mathcal{I}} = (\mathcal{I}_1, \ldots, \mathcal{I}_N)' | \mathcal{I}_i \in \{0, 1\}\}$.

Each step of the algorithm works as follows starting from an initial string $\boldsymbol{\mathcal{I}}_0$.

1. Using $\boldsymbol{\mathcal{I}}_i$ choose a neighboring string at random, denoted $\boldsymbol{\mathcal{I}}^*_{i+1}$. We discuss the definition of a neighborhood below.

2. If $IC(\boldsymbol{\mathcal{I}}_i) < IC(\boldsymbol{\mathcal{I}}^*_{i+1})$, set $\boldsymbol{\mathcal{I}}_{i+1} = \boldsymbol{\mathcal{I}}^*_{i+1}$. Else, set $\boldsymbol{\mathcal{I}}_{i+1} = \boldsymbol{\mathcal{I}}^*_{i+1}$ with

probability $e^{(IC(\boldsymbol{\mathcal{I}}^*_{i+1})-IC(\boldsymbol{\mathcal{I}}_i))/T_i}$ or set $\boldsymbol{\mathcal{I}}_{i+1} = \boldsymbol{\mathcal{I}}_i$ with probability $1 - e^{(IC(\boldsymbol{\mathcal{I}}^*_{i+1})-IC(\boldsymbol{\mathcal{I}}_i))/T_i}$.

Heuristically, the term T_i gets smaller making it more difficult, as the algorithm proceeds, to choose a point that does not increase $IC(.)$. The issue of the neighborhood is extremely relevant. What is the neighborhood? Intuitively, the neighborhood could be the set of strings that differ from the current string by one element of the string. But this may be too restrictive. We can allow the algorithm to choose at random, up to some maximum integer (say h), the number of string elements at which the string at steps i and $i+1$ will differ. So the neighborhood is all strings with up to h different bits from the current string. Another issue is when to stop the algorithm. There are a number of alternatives in the literature. We have chosen to stop the algorithm if it has not visited a string with higher $IC(.)$ than the current maximum for a prespecified number of steps (B_v) (Steps which stay at the same string do not count) or if the number of overall steps exceeds some other prespecified number (B_s). All strings visited by the algorithm are stored and the best chosen at the end rather than the final one.

The simulated annealing algorithm has been proven by Hajek (1988) (see also Del Moral and Miclo (1999)) to converge asymptotically, i.e. as $i \to \infty$, to the maximum of the function almost surely as long as $T_i = T_0/ln(i)$ for some T_0 for sufficiently large T_0. In particular, for almost sure convergence to the maximum it is required that $T_0 > d^*$. d^* denotes the maximum depth of all local maxima of the function $IC(.)$. Heuristically, the depth of a local maximum, $\boldsymbol{\mathcal{I}}_1$, is defined as the smallest number $E > 0$ such that the function never falls below $IC(\boldsymbol{\mathcal{I}}_1) - E$ during its trajectory from[1] this maximum

[1]A trajectory from $\boldsymbol{\mathcal{J}}_1$ to $\boldsymbol{\mathcal{J}}_2$ is a set of strings, $\boldsymbol{\mathcal{J}}_{11}, \boldsymbol{\mathcal{J}}_{12}, \ldots, \boldsymbol{\mathcal{J}}_{1p}$, such that (i) $\boldsymbol{\mathcal{J}}_{11} \in N(\boldsymbol{\mathcal{J}}_1)$, (ii) $\boldsymbol{\mathcal{J}}_{1p} \in N(\boldsymbol{\mathcal{J}}_2)$ and (iii) $\boldsymbol{\mathcal{J}}_{1i+1} \in N(\boldsymbol{\mathcal{J}}_{1i})$ for all $i = 1, \ldots, p$, where $N(\boldsymbol{\mathcal{J}})$ denotes the set of strings that make up the neighborhood of $\boldsymbol{\mathcal{J}}$.

to any other local maximum, $\boldsymbol{\mathcal{I}}_2$, for which $IC(\boldsymbol{\mathcal{I}}_1) < IC(\boldsymbol{\mathcal{I}}_2)$.

In the case of a single cluster the problem is already defined in terms of binary strings. However, when more clusters are considered a slight modification is needed. In particular, we write $\tilde{\mathcal{I}}_j$ in its binary form, so for example, 3 is written as 11 and so on. This poses a slight problem since for even numbers of clusters, k, the simulated annealing algorithm may move to a neighbouring point which denotes cluster $k+1$ which does not exist. We solve this problem by ensuring that any string that contains reference to cluster $k+1$ is penalised by setting the information criterion to an extreme negative value.

3.2 The genetic algorithm (GA)

Once again, we describe the operation of the algorithm when the domain of the function (information criterion) is the set of binary strings. The motivating idea of genetic algorithms is to start with a population of binary strings which then evolve and recombine to produce new populations with 'better' characteristics, i.e. higher values for the information criterion. We start with an initial population represented by a $N \times m$ matrix made up of 0's and 1's. Columns represent strings. m is the chosen size of the population. Denote this population (matrix) by $\mathbf{P}_0$. The genetic algorithm involves defining a transition from $\mathbf{P}_i$ to $\mathbf{P}_{i+1}$. The algorithm has the following steps:

1. For $\mathbf{P}_i$ create a $m \times 1$ 'fitness' vector, $\mathbf{p}_i$, by calculating for each column of $\mathbf{P}_i$ its 'fitness'. The choice of the 'fitness' function is completely open and depends on the problem. For our purposes it is the information criterion. Normalise $\mathbf{p}_i$, such that its elements lie in $(0, 1)$ and add up to 1. Denote this vector by $\mathbf{p}_i^*$. Treat $\mathbf{p}_i^*$ as a vector of probabilities and resample m times out of $\mathbf{P}_i$ with replacement, using the vector $\mathbf{p}_i^*$

as the probabilities with which each string with be sampled. So 'fit' strings are more likely to be chosen. Denote the resampled population matrix by $\mathbf{P}^1_{i+1}$.

2. Perform cross over on $\mathbf{P}^1_{i+1}$. For cross over we do the following: Arrange all strings in $\mathbf{P}^1_{i+1}$, in pairs (assume that m is even). Denote a generic pair by $(a^\alpha_1, a^\alpha_2, \ldots, a^\alpha_n)$, $(a^\beta_1, a^\beta_2, \ldots, a^\beta_n)$. Choose a random integer between 2 and $n-1$. Denote this by j. Replace the pair by the following pair: $(a^\alpha_1, a^\alpha_2, \ldots, a^\alpha_j, a^\beta_{j+1}, \ldots, a^\beta_n)$, $(a^\beta_1, a^\beta_2, \ldots, a^\beta_j, a^\alpha_{j+1}, \ldots, a^\alpha_n)$. Perform cross over on each pair with probability p_c. Denote the new population by $\mathbf{P}^2_{i+1}$. Usually p_c is set to some number around 0.5-0.6.

3. Perform mutation on $\mathbf{P}^2_{i+1}$. This amounts to flipping the bits (0 or 1) of $\mathbf{P}^2_{i+1}$ with probability p_m. p_m is usually set to a small number, say 0.01. After mutation the resulting population is $\mathbf{P}_{i+1}$.

These steps are repeated a prespecified number of times (B_g). Each set of steps is refereed to as generation in the genetic literature. If a string is to be chosen this is the one with maximum fitness. For every generation we store the identity of the string with maximum 'fitness'. At the end of the algorithm the string with the highest information criterion value over all members of the populations and all generations is chosen. One can think of the transition from one string of maximum fitness to another as a Markov Chain. So this is a Markov Chain algorithm. In fact, the Markov chain defined over all possible strings is time invariant but not irreducible as at least the $m-1$ least fit strings will never be picked. To see this note that in any population there will be a string with more fitness than that of the $m-1$ worst strings. There has been considerable work on the theoretical properties of genetic algorithms. Hartl and Belew (1990) and Del Moral and Miclo (1999) have shown that with probability approaching one, the population at the n-th

generation will contain the global maximum as $n \to \infty$. For more details see also Del Moral, Kallel, and Rowe (2001).

4 Monte Carlo Study

4.1 Monte Carlo Setup

In this section we carry out a Monte Carlo investigation of our new method. We consider two setups: A single cluster setup and a setup with three clusters. Let

$$y_{j,t} = \phi_j y_{j,t-1} + \epsilon_{j,t}, j = 1, \ldots, N, \quad t = 1, \ldots, T \tag{4}$$

where $\epsilon_{j,t} \sim N(0,1)$. We investigate the new method along a number of different dimensions for the above model. Namely, we consider variations in N, T and ϕ_j. More specifically, we consider $T \in \{100, 200, 400\}$ and $N \in \{25, 50, 75\}$.

For ϕ_j we consider the following setup: $\phi_j = 0.8$ with probability δ over j and $\phi_j \sim U(\gamma_1, \gamma_2)$ with probability $1-\delta$. This is a general setup designed to address a number of issues not widely discussed in the literature. Obviously, the degree of variation in ϕ_j under the alternative hypothesis is of great importance. Further, the choice of δ is likely to affect the performance of the new method. We set $\delta \in \{0.25, 0.5, 0.75\}$.

We choose $\gamma_1 = -0.9$ and $\gamma_2 = 0.4$. The performance measure we use is the estimated probability of classifying a series as nonpoolable. This should tend to zero for poolable series and to one for nonpoolable series. Denote the number of Monte Carlo replications by B. This probability is calculated as

follows in our experiments.

$$\hat{P}(\hat{\mathcal{I}}_u = 1 | \mathcal{I}_u = s) = \frac{1}{N_s B} \sum_{b=1}^{B} \sum_{\mathcal{I}_q = s} \hat{\mathcal{I}}_q^b \tag{5}$$

where $N_s = N(1-\delta)s + N\delta(1-s)$ and u denotes a generic series.

For multiple clusters we consider an extension of the above setup. In particular, every series has an equal chance to be in any of the three clusters or in no cluster. For clusters 1-3 the parameter ϕ_j is set to 0.9, 0.5 and 0.1 respectively. Otherwise, ϕ_j is uniformly distributed in $(-0.25, -0.95)$.

For the search algorithms we choose the following parameters: $B_v = 500$, $B_s = 5000$, $m = 100$ and $B_g = 150$, $h = 1$, $T_0 = 10$, $p_c = 0.6$, $p_m = 0.01$. This implies that for the simulated annealing we evaluate the loss function 5000 times whereas for the genetic algorithm we do so 15000. We view the parameters for SA as reasonable, whereas the parameters for GA are relatively low. We nevertheless choose them for GA since otherwise the Monte Carlo analysis becomes prohibitively expensive. We carry out 250 Monte Carlo replications. Again anything significantly more than that is prohibitively expensive. Note that the results reported here took more than 1 1/2 months of computer time on a personal computer with 3 Ghz processor speed.

4.2 Results

In Tables 1-6 we report results for the experiments dealing with one cluster. In these tables we report the probability of finding a series nonpoolable both when it is in fact nonpoolable and when it is poolable. In Tables 7 and 8 we report results for the three cluster experiments. There, for every (N, T) pair we report estimated probabilities that a series with a given cluster identifier

will be found to belong to some cluster or to no cluster at all. More specifically, our results are reported in the form of 4×4 matrices. The i-th row of this matrix, $i = 1, 2, 3$, reports on series that belong to the i-th cluster. For the fourth row we report results for series that do not belong in a cluster. So, for example, row 1 reports on series that belong to the cluster identified as cluster 1. The first column reports the estimated probability that a series that belongs to cluster 1 will be found to belong to cluster 1, and so on for columns 2 and 3. Column 4 reports the estimated probability that a series belonging to cluster 1 will be found to belong to no cluster. Note that the information criterion procedure names clusters in an ad hoc basis. That means that the information criterion method may name cluster 1 as cluster 2. We identify clusters by the way the dataset is constructed. So, for, say, $N = 80$, the first 20 series in the dataset have AR coefficient 0.9. We call this cluster 1 and it is clearly different to cluster 2 in which series have AR coefficients equal to 0.5. The procedure may find that a lot of series in the group of the first 20 series belong to the same cluster. The procedure may then give an ad hoc name to this cluster as, say, cluster 2. We rename this cluster as cluster 1 in the reporting of the results.

Results are revealing. Overall, simulated annealing seems to work better than the genetic algorithm. We see that for the one cluster case results dramatically depend on the proportion of poolable series in the dataset. Whereas, for a high proportion of poolable series, the algorithms work satisfactorily, when there are few poolable series, the algorithms are less able to distinguish between poolable and nonpoolable series.

Moving on to multiple clusters we see that there are variations in performance depending mainly on the search algorithm and on the information criterion used. BIC performs best, with HQ second and AIC last. Given the

overall tendency of the algorithms to report that series actually belonging to a cluster do not belong to one this result is reasonable, since it is less parsimonious not to belong to a cluster and hence more parsimonious criteria such as BIC will do better. The first cluster's parameter is further away from the average parameter for series not belonging to a cluster and hence it is reasonable that series belonging to the first cluster will be identified best. When the number of observations increases there is further overall tendency for series to appear not to belong to clusters. For the AIC this is expected, since it tends to choose overparametrised, and hence less parsimonious, models asymptotically. The SA algorithm always tends to err in favour of finding a series either belonging to a cluster closer (in terms of parameters) to the non-cluster group of series than its actual cluster or belonging to the non-cluster group. This is easily seen by the upper triangularity of the matrices of results in the tables. The GA algorithm can err either way and in general is less accurate. Given its larger computational cost, we can conclude that SA is to be preferred for empirical analysis.

5 Conclusion

The use of panel datasets for the investigation of a number of economic phenomena has been increasing recently. Both the availability of larger datasets and the development of new estimation methods methods specifically designed for panel datasets can account for this.

An important advantage of panel methods is their ability to improve inference compared to single unit methods. Nevertheless, this implies that the parameter restrictions implied by the panel structure are valid. Poolability tests exists to help with this problem but if they reject the null hypothesis of poolability the researcher is often uncertain about the cause of the rejection,

or in particular about the identity of the series that caused this rejection. In other words a method that could distinguish poolable from nonpoolable series within a panel dataset would be of interest to empirical researchers.

This paper has suggested such a method. It is based on the use of information criteria to evaluate splits of the dataset between poolable and nonpoolable series. The method extends to cases where the dataset may contain more than one cluster of series with similar characteristics. Maximisation of the information criterion is difficult using standard grid search methods since the number of possible splits increases exponentially with the number of series considered. Hence, we resort to nonstandard optimisation methods. We find that in number of situations the methods work quite well. Further research can illustrate both the use of the new method in empirical contexts and the potential for alternative maximisation algorithms to give rise to methods that improve upon the results reported here.

References

Baltagi, B. H. (2001): *Econometric Analysis of Panel Data*. Wiley.

Baltagi, B. H., J. Hidalgo, and Q. Li (1996): "A Nonparametric Test for Poolability Using Panel Data," *Journal ofEconometrics*, 75(1), 345–367.

Cantu-Paz, E., and C. Kamath (2003): "Inducing oblique decision trees with evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, 7(1), 54–68.

Chow, A. (1960): "Tests of Equality Between Sets of Coefficients in two Linear Regressions," *Econometrica*, 28, 591–605.

Del Moral, P., L. Kallel, and J. Rowe (2001): "Modelling Genetic Algorithms with Interacting Partical Systems," *Revista de Matematica, Teoria y Aplicaciones*, 8(2).

Del Moral, P., and L. Miclo (1999): "On the Convergence and the Applications of Generalised Simulated Annealing," *SIAM Journal of Control and Optimisation*, 37(4), 1222–1250.

Durlauf, S., and P. Johnson (1995): "Multiple Regimes and Cross-Country Growth Behaviour," *Journal of Applied Econometrics*, 10, 365–384.

Hajek, B. (1988): "Cooling Schedules for Optimal Annealing," *Mathematics of Operations Research*, 13(2), 311–331.

Hartl, H. R. F., and R. K. Belew (1990): "A Global Convergence Proof for a Class of Genentic Algorithms," *Technical Report, Technical University of Vienna*.

Kakazawa, Y., R. H. Shumway, and M. Taniguchi (1998): "Discrimination and Clustering for Multivariate Time series," *Journal of the American Statistical Association*, 93, 328–340.

Kapetanios, G. (2003): "Determining the Poolability Properties of Individual Series in Panel Datasets," *Queen Mary, University of London Working Paper no. 499*.

Paap, R., P. H. Frances, and D. Van Dijk (2003): "Does Africa grow slower than Asia and Latin America?," *Erasmus University; Econometric Institute, Research Report 2003-07/A*.

Sin, C. Y., and H. White (1996): "Information Criteria for Selecting

Possibly Misspecified Parametric Models," *Journal of Econometrics*, 71(1–2), 207–225.

Vahid, F. (1999): "Partial Pooling: A Partial Answer to 'To Pool Or Not To Pool'," in *Cointegration, Causality and Forecasting: Festschrift in Honor of Clive W. J. Granger*, ed. by R. Engle, and H. White.

Ziemer, R. F., and M. E. Wetzstein (1983): "A Stein Rule Method for Pooling Data," *Economics Letters*, 11(1), 137–143.

Table 1: One cluster, SA, AIC

%Poolab	(N,T)	100	200	400
0.25	25	$\binom{0.698}{0.772}$	$\binom{0.532}{0.861}$	$\binom{0.508}{0.873}$
	50	$\binom{0.709}{0.808}$	$\binom{0.748}{0.830}$	$\binom{0.667}{0.878}$
	75	$\binom{0.810}{0.798}$	$\binom{0.839}{0.832}$	$\binom{0.743}{0.863}$
0.50	25	$\binom{0.410}{0.894}$	$\binom{0.444}{0.875}$	$\binom{0.372}{0.944}$
	50	$\binom{0.587}{0.842}$	$\binom{0.424}{0.920}$	$\binom{0.402}{0.922}$
	75	$\binom{0.458}{0.905}$	$\binom{0.618}{0.886}$	$\binom{0.551}{0.905}$
0.75	25	$\binom{0.298}{0.944}$	$\binom{0.348}{0.944}$	$\binom{0.301}{0.944}$
	50	$\binom{0.404}{0.921}$	$\binom{0.424}{0.926}$	$\binom{0.377}{0.926}$
	75	$\binom{0.443}{0.920}$	$\binom{0.474}{0.910}$	$\binom{0.434}{0.929}$

For the notation $\binom{a}{b}$ we have that a gives the probability that a poolable series will be classified as nonpoolable, whereas b gives the probability that a nonpoolable series will be classified as nonpoolable.

Table 2: One cluster, SA, BIC				
%Poolab	(N,T)	100	200	400
0.25	25	$\binom{0.849}{0.664}$	$\binom{0.794}{0.694}$	$\binom{0.786}{0.747}$
	50	$\binom{0.744}{0.700}$	$\binom{0.628}{0.788}$	$\binom{0.624}{0.829}$
	75	$\binom{0.901}{0.667}$	$\binom{0.734}{0.777}$	$\binom{0.787}{0.791}$
0.50	25	$\binom{0.235}{0.870}$	$\binom{0.517}{0.759}$	$\binom{0.261}{0.894}$
	50	$\binom{0.331}{0.847}$	$\binom{0.440}{0.827}$	$\binom{0.411}{0.853}$
	75	$\binom{0.399}{0.833}$	$\binom{0.417}{0.842}$	$\binom{0.367}{0.863}$
0.75	25	$\binom{0.132}{0.917}$	$\binom{0.135}{0.917}$	$\binom{0.123}{0.907}$
	50	$\binom{0.094}{0.944}$	$\binom{0.171}{0.912}$	$\binom{0.143}{0.917}$
	75	$\binom{0.247}{0.880}$	$\binom{0.175}{0.910}$	$\binom{0.145}{0.914}$
See notes in Table 1				

Table 3: One cluster, SA, HQ				
%Poolab	(N,T)	100	200	400
0.25	25	$\binom{0.659}{0.741}$	$\binom{0.841}{0.741}$	$\binom{0.778}{0.772}$
	50	$\binom{0.808}{0.740}$	$\binom{0.688}{0.784}$	$\binom{0.765}{0.821}$
	75	$\binom{0.857}{0.737}$	$\binom{0.810}{0.799}$	$\binom{0.591}{0.856}$
0.50	25	$\binom{0.274}{0.912}$	$\binom{0.158}{0.944}$	$\binom{0.265}{0.912}$
	50	$\binom{0.376}{0.878}$	$\binom{0.627}{0.829}$	$\binom{0.440}{0.862}$
	75	$\binom{0.558}{0.821}$	$\binom{0.635}{0.823}$	$\binom{0.706}{0.842}$
0.75	25	$\binom{0.181}{0.944}$	$\binom{0.202}{0.944}$	$\binom{0.149}{0.944}$
	50	$\binom{0.218}{0.944}$	$\binom{0.218}{0.921}$	$\binom{0.240}{0.898}$
	75	$\binom{0.215}{0.944}$	$\binom{0.285}{0.917}$	$\binom{0.276}{0.917}$
See notes in Table 1				

Table 4: One cluster, GA, AIC				
%Poolab	(N,T)	100	200	400
0.25	25	$\binom{0.921}{0.741}$	$\binom{0.810}{0.812}$	$\binom{0.714}{0.846}$
	50	$\binom{0.940}{0.733}$	$\binom{0.880}{0.748}$	$\binom{0.859}{0.758}$
	75	$\binom{0.927}{0.680}$	$\binom{0.930}{0.698}$	$\binom{0.924}{0.681}$
0.50	25	$\binom{0.402}{0.944}$	$\binom{0.346}{0.944}$	$\binom{0.406}{0.944}$
	50	$\binom{0.464}{0.933}$	$\binom{0.482}{0.940}$	$\binom{0.444}{0.940}$
	75	$\binom{0.497}{0.904}$	$\binom{0.496}{0.905}$	$\binom{0.463}{0.902}$
0.75	25	$\binom{0.339}{0.944}$	$\binom{0.377}{0.944}$	$\binom{0.380}{0.944}$
	50	$\binom{0.458}{0.944}$	$\binom{0.408}{0.944}$	$\binom{0.437}{0.944}$
	75	$\binom{0.449}{0.938}$	$\binom{0.450}{0.944}$	$\binom{0.444}{0.944}$
See notes in Table 1				

Table 5: One cluster, GA, BIC				
%Poolab	(N,T)	100	200	400
0.25	25	$\binom{0.635}{0.710}$	$\binom{0.802}{0.710}$	$\binom{0.730}{0.778}$
	50	$\binom{0.842}{0.697}$	$\binom{0.919}{0.707}$	$\binom{0.915}{0.745}$
	75	$\binom{0.944}{0.656}$	$\binom{0.944}{0.684}$	$\binom{0.860}{0.704}$
0.50	25	$\binom{0.235}{0.907}$	$\binom{0.218}{0.926}$	$\binom{0.137}{0.944}$
	50	$\binom{0.338}{0.936}$	$\binom{0.380}{0.940}$	$\binom{0.353}{0.933}$
	75	$\binom{0.431}{0.887}$	$\binom{0.414}{0.901}$	$\binom{0.418}{0.896}$
0.75	25	$\binom{0.164}{0.944}$	$\binom{0.167}{0.944}$	$\binom{0.143}{0.944}$
	50	$\binom{0.265}{0.940}$	$\binom{0.279}{0.944}$	$\binom{0.262}{0.944}$
	75	$\binom{0.351}{0.941}$	$\binom{0.345}{0.944}$	$\binom{0.333}{0.941}$
See notes in Table 1				

Table 6: One cluster, GA, HQ				
%Poolab	(N,T)	100	200	400
0.25	25	$\binom{0.921}{0.673}$	$\binom{0.746}{0.762}$	$\binom{0.944}{0.731}$
	50	$\binom{0.944}{0.719}$	$\binom{0.859}{0.740}$	$\binom{0.889}{0.746}$
	75	$\binom{0.880}{0.695}$	$\binom{0.915}{0.701}$	$\binom{0.944}{0.700}$
0.50	25	$\binom{0.231}{0.944}$	$\binom{0.226}{0.944}$	$\binom{0.261}{0.944}$
	50	$\binom{0.438}{0.931}$	$\binom{0.391}{0.931}$	$\binom{0.378}{0.942}$
	75	$\binom{0.436}{0.884}$	$\binom{0.447}{0.884}$	$\binom{0.469}{0.905}$
0.75	25	$\binom{0.228}{0.944}$	$\binom{0.251}{0.944}$	$\binom{0.246}{0.944}$
	50	$\binom{0.338}{0.944}$	$\binom{0.327}{0.944}$	$\binom{0.306}{0.944}$
	75	$\binom{0.373}{0.941}$	$\binom{0.361}{0.944}$	$\binom{0.376}{0.944}$
See notes in Table 1				

Table 7: 3 clusters, SA													
N	T	AIC				BIC				HQ			
		0.65	0.01	0.01	0.33	0.84	0.02	0.01	0.13	0.75	0.02	0.01	0.22
	100	0.00	0.64	0.01	0.34	0.01	0.79	0.00	0.20	0.00	0.75	0.01	0.24
		0.00	0.00	0.32	0.68	0.00	0.00	0.35	0.64	0.00	0.00	0.33	0.67
		0.01	0.00	0.00	0.99	0.01	0.01	0.04	0.94	0.01	0.01	0.02	0.97
		0.64	0.00	0.02	0.33	0.80	0.03	0.01	0.15	0.76	0.02	0.01	0.22
25	200	0.00	0.64	0.02	0.35	0.00	0.77	0.02	0.21	0.00	0.75	0.01	0.24
		0.00	0.00	0.30	0.70	0.00	0.00	0.32	0.68	0.00	0.00	0.32	0.68
		0.00	0.00	0.00	1.00	0.00	0.00	0.03	0.97	0.00	0.00	0.02	0.98
		0.65	0.01	0.01	0.32	0.78	0.07	0.01	0.14	0.78	0.02	0.01	0.19
	400	0.00	0.64	0.02	0.34	0.00	0.71	0.04	0.25	0.00	0.75	0.02	0.24
		0.00	0.00	0.29	0.71	0.00	0.00	0.32	0.68	0.00	0.00	0.30	0.70
		0.00	0.00	0.00	1.00	0.00	0.00	0.03	0.97	0.00	0.00	0.01	0.99
		0.58	0.02	0.01	0.39	0.80	0.04	0.01	0.15	0.70	0.04	0.01	0.26
	100	0.00	0.56	0.01	0.43	0.01	0.77	0.01	0.22	0.00	0.65	0.01	0.34
		0.00	0.00	0.29	0.70	0.00	0.00	0.32	0.68	0.00	0.00	0.30	0.69
		0.00	0.00	0.02	0.97	0.01	0.01	0.03	0.95	0.00	0.00	0.03	0.96
		0.56	0.05	0.01	0.39	0.74	0.08	0.00	0.17	0.69	0.05	0.01	0.25
50	200	0.00	0.52	0.01	0.46	0.00	0.68	0.02	0.30	0.00	0.62	0.02	0.36
		0.00	0.00	0.28	0.72	0.00	0.00	0.30	0.70	0.00	0.00	0.29	0.71
		0.00	0.00	0.02	0.98	0.00	0.00	0.04	0.96	0.00	0.00	0.03	0.97
		0.56	0.05	0.01	0.37	0.75	0.08	0.00	0.17	0.68	0.07	0.01	0.24
	400	0.00	0.53	0.04	0.43	0.00	0.66	0.02	0.32	0.00	0.63	0.03	0.34
		0.00	0.00	0.27	0.73	0.00	0.00	0.30	0.70	0.00	0.00	0.27	0.73
		0.00	0.00	0.01	0.99	0.00	0.00	0.03	0.97	0.00	0.00	0.03	0.97
		0.53	0.03	0.01	0.44	0.76	0.05	0.00	0.18	0.64	0.06	0.00	0.30
	100	0.00	0.52	0.02	0.46	0.01	0.72	0.02	0.26	0.00	0.60	0.02	0.38
		0.00	0.00	0.27	0.73	0.00	0.00	0.29	0.70	0.00	0.00	0.30	0.70
		0.00	0.00	0.03	0.97	0.01	0.01	0.03	0.94	0.00	0.01	0.03	0.96
		0.52	0.07	0.01	0.40	0.71	0.10	0.00	0.19	0.65	0.09	0.01	0.25
75	200	0.00	0.42	0.03	0.55	0.00	0.64	0.02	0.34	0.00	0.54	0.02	0.44
		0.00	0.00	0.27	0.73	0.00	0.00	0.32	0.68	0.00	0.00	0.29	0.71
		0.00	0.00	0.02	0.97	0.00	0.00	0.05	0.95	0.00	0.00	0.04	0.96
		0.52	0.07	0.02	0.39	0.74	0.08	0.00	0.18	0.65	0.10	0.01	0.24
	400	0.00	0.44	0.03	0.53	0.00	0.68	0.03	0.29	0.00	0.53	0.03	0.45
		0.00	0.00	0.25	0.75	0.00	0.00	0.30	0.70	0.00	0.00	0.28	0.72
		0.00	0.00	0.01	0.99	0.00	0.00	0.04	0.96	0.00	0.00	0.04	0.96

See explanation of layout in page 13

Table 8, 3 clusters, GA													
N	T	AIC				BIC				HQ			
		0.38	0.09	0.06	0.47	0.45	0.07	0.07	0.42	0.42	0.08	0.06	0.45
	100	0.06	0.38	0.08	0.49	0.07	0.42	0.08	0.43	0.05	0.40	0.07	0.47
		0.06	0.05	0.32	0.57	0.06	0.06	0.36	0.53	0.05	0.06	0.34	0.55
		0.06	0.05	0.04	0.86	0.06	0.08	0.04	0.82	0.06	0.07	0.04	0.82
		0.40	0.09	0.06	0.45	0.43	0.06	0.07	0.45	0.40	0.08	0.07	0.46
25	200	0.05	0.37	0.06	0.51	0.05	0.41	0.08	0.46	0.06	0.38	0.07	0.49
		0.06	0.04	0.33	0.58	0.05	0.05	0.33	0.57	0.05	0.05	0.32	0.58
		0.06	0.06	0.05	0.83	0.08	0.07	0.03	0.82	0.07	0.05	0.03	0.86
		0.41	0.07	0.07	0.45	0.41	0.08	0.06	0.45	0.42	0.08	0.06	0.44
	400	0.05	0.37	0.07	0.52	0.05	0.41	0.07	0.48	0.05	0.38	0.09	0.48
		0.05	0.05	0.32	0.59	0.05	0.05	0.32	0.58	0.06	0.05	0.31	0.58
		0.06	0.05	0.04	0.85	0.06	0.06	0.04	0.83	0.05	0.05	0.04	0.86
		0.39	0.11	0.09	0.41	0.40	0.09	0.10	0.41	0.38	0.12	0.09	0.41
	100	0.10	0.35	0.10	0.45	0.08	0.38	0.11	0.42	0.10	0.36	0.11	0.43
		0.08	0.08	0.31	0.53	0.11	0.09	0.31	0.49	0.09	0.09	0.32	0.50
		0.10	0.09	0.06	0.75	0.09	0.10	0.07	0.74	0.09	0.10	0.06	0.75
		0.38	0.12	0.08	0.43	0.39	0.11	0.08	0.42	0.38	0.11	0.08	0.42
50	200	0.09	0.35	0.11	0.45	0.09	0.37	0.10	0.44	0.10	0.37	0.10	0.44
		0.08	0.08	0.30	0.54	0.09	0.09	0.31	0.52	0.08	0.09	0.29	0.54
		0.08	0.09	0.06	0.76	0.09	0.09	0.07	0.75	0.10	0.11	0.07	0.73
		0.37	0.11	0.10	0.42	0.38	0.11	0.09	0.43	0.38	0.09	0.08	0.45
	400	0.09	0.37	0.07	0.47	0.09	0.38	0.08	0.45	0.09	0.36	0.11	0.44
		0.08	0.08	0.30	0.54	0.08	0.08	0.34	0.51	0.08	0.09	0.31	0.52
		0.11	0.10	0.06	0.73	0.09	0.09	0.07	0.75	0.10	0.10	0.06	0.73
		0.35	0.13	0.11	0.41	0.36	0.14	0.11	0.38	0.36	0.14	0.10	0.40
	100	0.13	0.33	0.13	0.42	0.11	0.36	0.11	0.42	0.13	0.35	0.12	0.41
		0.11	0.11	0.30	0.49	0.11	0.12	0.31	0.46	0.12	0.11	0.29	0.48
		0.12	0.11	0.09	0.68	0.11	0.11	0.09	0.69	0.11	0.11	0.08	0.70
		0.35	0.14	0.11	0.40	0.36	0.14	0.11	0.39	0.35	0.13	0.11	0.41
75	200	0.12	0.34	0.11	0.43	0.12	0.36	0.13	0.40	0.11	0.35	0.12	0.42
		0.12	0.11	0.29	0.48	0.11	0.11	0.29	0.49	0.12	0.11	0.30	0.47
		0.11	0.12	0.08	0.69	0.11	0.09	0.09	0.71	0.11	0.11	0.08	0.70
		0.36	0.15	0.11	0.38	0.36	0.12	0.11	0.41	0.36	0.13	0.10	0.42
	400	0.11	0.35	0.11	0.43	0.12	0.34	0.13	0.41	0.12	0.33	0.12	0.42
		0.10	0.11	0.30	0.49	0.12	0.11	0.29	0.48	0.10	0.10	0.30	0.50
		0.11	0.09	0.09	0.71	0.10	0.11	0.08	0.72	0.12	0.10	0.08	0.69

See explanation of layout in page 13

This working paper has been produced by
the Department of Economics at
Queen Mary, University of London

Department of Economics
Queen Mary, University of London
Mile End Road
London E1 4NS
Tel: +44 (0)20 7882 5096
Fax: +44 (0)20 8983 3580
Web: www.econ.qmul.ac.uk/papers/wp.htm