Adaptive estimation for locally stationary autoregressions

Karolos K. Korkas† and Piotr Fryzlewicz

London School of Economics, UK

†Presenting author. E-mail: k.k.korkas@lse.ac.uk, p.fryzlewicz@lse.ac.uk

Introduction

In this paper we focus on the problem of detecting structural breaks in locally stationary autoregressive processes. We explore the simplest type of deviation from stationarity which is piecewise constant autoregressive models. We assume an AR(p) model

$$y_t = \phi_1(t)y_{t-1} + \phi_2(t)y_{t-2} + ... + \phi_p(t)y_{t-p} + \epsilon_t$$

where the coefficients ϕ_i, $i = 1, ..., p$ are piecewise constant functions of time and the noise sequence $\{\epsilon_t\}$ is iid with mean 0 and variance σ^2. The order of the AR process is assumed to be known, but the number of breakpoints and their locations are unknown and need to be estimated. We are not directly interested for the AR coefficients since their estimation is straightforward after having identified both the number and the location of the breakpoints by using a parametric method such as least-squares regression or maximum likelihood estimation within every segment.

Other breakpoint detection methods for piecewise constant autoregressive (weakly or strongly dependent) processes include the SLEX model of [11] which searches for a segmentation that has the minimum cost among all the possible blocks but assumes a dyadic structure; Auto-PARM of [4] which looks for the model that minimises the Minimum Description Length; and [3] who use the empirical wavelet coefficients as obtained from a least squares minimisation problem. For more recent developments the reader should advise [9] who propose a method that minimises a penalised contrast function (similar to a penalised loss function) which controls for the number of breakpoints; and [2] who apply the binary segmentation algorithm on local wavelets periodograms of a dependent process in order to de-correlate it and bring it closer to normality by adopting the Locally Stationary Wavelets (LSW) of [12].

The Fused Lasso Methodology

In this section we present a non-parametric model selection technique, the Fused Lasso regression of [14]. This method is an extension of the Lasso method developed in a seminal paper by [13]. Fused Lasso can solve the signal+noise problem (Fused Lasso Signal Approximator - FLSA) by transforming the predictor matrix accordingly ($X = I$ where I is a matrix with all the elements equal to zero except the diagonal). The fusion penalty coincides with the total variation penalty and is a novel approach in penalised least squares problem because it uses simultaneously the total variation and the $L1$-norm penalties, which favours solutions that are both sparse and blocky.

Sparsity can be achieved by the Lasso penalty which shrinks irrelevant coefficients towards zero (see [13]). Assume that we have p predictors x_{ij}, $j = 1, 2, ..., p$ and the response variable y_i. Predictors are also standardised to have zero mean and unit variance in order to be comparable in size. Lasso finds those β_js that minimise the following Lagrange function

$$f(\beta) = \frac{1}{2}\left(\sum_{i=1}^{n} y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| \quad (1)$$

where λ is a tuning parameter - smaller values means more coefficients are set to zero.

The authors of [14] extended the Lasso to include not only sparsity of the coefficients but also sparsity in their differences (Fused Lasso). The Fused Lasso makes sense only when there is a natural ordering of the coefficients. Simply, the coefficients are closely related to their neighbours. In this case the loss function takes the following form

$$f(\beta) = \frac{1}{2}\left(\sum_{i=1}^{n} y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda_1\sum_{j=1}^{p}|\beta_j| + \lambda_2\sum_{j=2}^{p}|\beta_j - \beta_{j-1}| \quad (2)$$

The Fused Lasso AR(p) estimator

In this section we consider the following AR(p) model

$$y_t = \phi_1(t)y_{t-1} + \phi_2(t)y_{t-2} + ... + \phi_p(t)y_{t-p} + \epsilon_t$$

where ϕ_{tj} for $j = 1, ..., p$ are piecewise constant functions of time where breakpoints b_i for every ϕ_{ij} occur not necessarily at the same time.

The loss function in the AR(p) case in matrix notation is now

$$f(\phi) = ||\mathbf{y} - \mathbf{X}\phi||_2^2 + \lambda_2||\mathbf{D}\phi||$$

where $\mathbf{y}$ is the T response matrix; $\mathbf{X}$ is the $T \times pT$ design matrix whose partition contains p diagonal matrices of size $T \times T$; ϕ is the $pT \times 1$ coefficient matrix; and $\mathbf{D}$ is the penalty matrix whose special form is described shortly after.

The AR(p) case and in general any time-varying coefficient model can be written to take a similar form with the FLSA problem by noticing that one replaces the ones in the diagonal of the design matrix with the values of y_{tj} for $j = 1, ..., p$. In the next chapter we explore the AR(1) piecewise constant model where we explore the similarities with the signal+noise model. In the multivariate case the design matrix which contains the lagged values of the autoregressive process y_t has the following form

$$\mathbf{X} = \begin{pmatrix} y_{t1} & y_{t2} & ... & y_{tp} \end{pmatrix}$$

where y_{tj} is the $T \times T$ diagonal matrix that contains the lagged values $j = 1, ..., p$ of y_t.

The penalty matrix $\mathbf{D}$ has the following form

$$\mathbf{D} = \begin{pmatrix} \mathbf{D_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{D_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{D_p} \end{pmatrix} \quad (3)$$

with

$$\mathbf{D_j} = \begin{pmatrix} -1 & 1 & 0 & ... & 0 & 0 \\ 0 & -1 & 1 & ... & 0 & 0 \\ . & . & . & ... & . & . \\ 0 & 0 & 0 & ... & -1 & 1 \end{pmatrix}$$

The loss function $f(\phi)$ is not differentiable due to the absolute values of the penalties. However, it is strictly convex and thus a unique minimum always exists.

Estimation Methodologies - Proximal Gradient Optimization

In order to deal with the non-differentiability of the loss function, [7] propose the proximal-gradient method using an auxiliary matrix that smoothes the loss function. According to the authors, approximation of the smoothed function is sufficiently close to the original objective function and can work efficiently under the $n << p$ paradigm.

Authors define an auxiliary vector α with length $T \times (T-1)$ and domain $Q = \{||a||_1 \leq 1$ such that

$$||\beta^T\mathbf{D}||_1 = \max_{\alpha \in Q} \alpha\beta^T\mathbf{D} \quad (4)$$

According to (4) the penalty term can be seen as the inner product of the auxiliary vector α and the linear mapping of β via a linear operator $\Gamma(\beta) = \beta^T\mathbf{D}$. Yet, it remains a non-smooth function of β, hence, optimisation is still not feasible. To deal with this, they add an auxiliary convex function $d(a)$ defined on Q such that:

$$f_\mu(\beta) = \max_{\alpha \in Q} \alpha\beta^T\mathbf{D} - \mu d(a) \quad (5)$$

where μ is a smoothness parameter.

The algorithm of [7] utilises the optimal solution of (5) and propose $d(\alpha) = \frac{1}{2}||\alpha||_2^2$.

Figure (1) plots the non-smooth and smooth objective function assuming different values of μ. It is obvious to see that $f_0(\beta) = \max_{\alpha \in Q} \alpha\beta^T\mathbf{D}$ is lower bounded by $f_\mu(\beta)$. The gap G between the objective functions is $G = \max_{\alpha \in Q} d(a) = \max_{\alpha \in Q} \frac{1}{2}||\alpha||_2^2 = T/2$. Hence, $f_0(\beta) - f_\mu(\beta) \leq \mu T/2$ so μ has the role of adjusting the gap between the smooth and non-smooth function. To achieve efficient convergence [7] set $\mu = \epsilon/2G$. Having smoothed the non-differentiable objective function authors continues with a gradient optimisation algorithm.

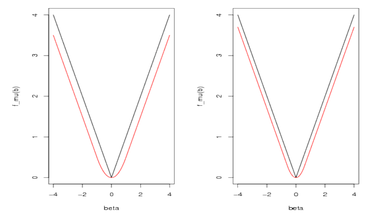

Figure 1: Non-smooth (black) vs smooth (red) objective function when the smoothness parameter μ is 1 (left) and 0.6 (right)

Solution Path of the Generalised Lasso

The authors in [15] propose a path algorithm to solve the fused lasso problem. Especially for the dimensional case where $X = I$, the methods resembles the top-down approach of the Unbalanced Haar technique of [6]. We adapt their method to solve the AR(1) case because it is faster than [7] and as a path algorithm is easier to build a stopping rule. From here on we assume that $\lambda_1 = 0$ and $\lambda_2 = \lambda$.

The authors consider an argument from [8] who transform the dual problem

$$\min_{\phi \in R^n} \frac{1}{2}||y - X\phi||_2^2 + \lambda||D\phi||_1$$

where D is the $(N-1) \times N$ penalty matrix as defined at (3) into a simpler one with no linear transformations by re-writing the dual problem as

$$\min_{\phi \in R^n, z \in R^m} \frac{1}{2}||y - X\phi||_2^2 + \lambda||z||_1$$

subject to $D\phi = z$.

Following similar arguments with [15] and [8], the dual problem of the optimisation problem is reduced

$$\min_{u \in R^m} \frac{1}{2}||y - (DX^{-1})^T u||_2^2 \quad (6)$$

subject to $||u||_\infty \leq \lambda$.

The primal and dual solution are related by

$$\hat{\phi}_\lambda = X^{-1}(y - \tilde{D}^T\hat{u}_\lambda) \quad (7)$$

where $\tilde{D} = DX^{-1}$.

As can be seen from the structure of the problem, λ has an inverse consequence in the dual problem. This means that when $\lambda = \infty$ the dual problem is an unconstrained problem where the solutions $\hat{u}_{\lambda=\infty}$ are equivalent to the OLS solutions.

The authors propose a path algorithm for the signal+noise case given that only for this case the boundary lemma holds. The lemma is as follows

The boundary lemma: *For the 1-dimensional fused lasso (FLSA) we have that for any coordinate i, the solution $\hat{u}_\lambda$ of (6) satisfies*

$\hat{u}_{\lambda_0,i} = \lambda_0 \Rightarrow \hat{u}_{\lambda_0,i} = \lambda \ \ for \ \ all \ \ \lambda \ \in [0, \lambda_0]$

and

$\hat{u}_{\lambda_0,i} = -\lambda_0 \Rightarrow \hat{u}_{\lambda_0,i} = -\lambda \ \ for \ \ all \ \ \lambda \ \in [0, \lambda_0]$

Simply, the lemma states that for decreasing λ the coordinate u_i stays within the boundary i.e. $u_i = \lambda$ and thus at every iteration we need to solve only for the interior coordinates. The boundary lemma is the equivalent of proposition 2 of [5] which states that when two values $\hat{\beta}$ fuse then for increasing λ those values remain always fused. The boundary lemma is about the fusion of the dual solutions $\hat{u}$ for decreasing λ. We prove that the boundary lemma holds for the AR(1) case.

Having established that we present the authors' path algorithm, noticing that the penalty matrix is transformed to $\tilde{D} = DX^{-1}$ which is summarised below:

For $k = 0, 1, n-2$,

- Compute the solution at λ_k using the following least squares estimator

$$\hat{u}_{\lambda_k,-\mathcal{B}} = (\tilde{D}_{-\mathcal{B}}(\tilde{D}_{-\mathcal{B}})^T)^{-1}\tilde{D}_{-\mathcal{B}}(y - \lambda_k(\tilde{D}_{\mathcal{B}})^T s)$$

- Locate the next hitting time t_i and λ_{k+1} using the following

$$t_i = \frac{[\tilde{D}_{-\mathcal{B}}(\tilde{D}_{-\mathcal{B}})^T)^{-1}\tilde{D}_{-\mathcal{B}}y]_i}{[\tilde{D}_{-\mathcal{B}}(\tilde{D}_{-\mathcal{B}})^T)^{-1}\tilde{D}_{-\mathcal{B}}(\tilde{D}_{\mathcal{B}})^T s]_i \pm 1} \quad (8)$$

and

$$\lambda_{k+1} = \max_i(t_i)$$

- Add coordinate i to $\mathcal{B}$ and its sign to s.

Set $\mathcal{B}$ contains the coordinates that are currently on the boundary, or simply the coordinate that has the maximum distance with its next point $i+1$. In addition, $D_{-\mathcal{B}}$ is the penalty matrix which does not contain the row corresponding to point i.

From the above formulas such as (7) one can see that the invertibility of matrix X is a necessary condition, but it can exhibit a very unstable behaviour at the computational level. We propose to transform the response matrix y and the design matrix X by using the Moore-Penrose pseudoinverse of a matrix (we denote it X^+) instead of the inverse. The pseudo-inverse of a diagonal matrix will return 0 for the rows where some of the data are exactly equal to zero or fall below a tolerance level. Ideally, we want $\frac{y_{t+1}}{y_t}$ to be as large as possible. For that reason, and in order to enforce invertibility of the diagonal matrix X, we test whether every element of the diagonal X^+ is below a certain tolerance level ϵ. For those values that this is satisfied we replace the zeros with a value C. In our simulations we set $\epsilon = 10^{-3}$ and $C = 2var(y_t)$ and the performance was very good.

Model selection and choice of λ_2

An important issue in estimation of the number of breakpoints is a stopping rule. This involves a tuning process and it is controlled by the λ penalty parameter. Given the multiresolution criterion of Davis and Kovac (2001) and due to the multiscale nature of the algorithm we choose the estimated piecewise constant coefficient such that the multiresoltuion sum of the estimated residuals $\sum_{t=1}^{T}(y_t - \hat{\phi}_t y_{t-1})$ is bounded by $\sigma_n\sqrt{\tau \log n}$.

In addition to the above, we propose two information criteria, the Cp statistic from [15]) and the Bayesian Information Criterion (BIC) as applied to structural break estimation by [1]. The criteria are given below respectively,

$$C_p(\lambda) = \sum_{t=1}^{T}(y_t - \phi_t^{\lambda_k}y_{t-1})^2 - T\hat{\sigma}^2 + 2\hat{\sigma}^2 df(\phi^{\lambda_k})$$

and

$$BIC(\lambda) = ln\left(\sum_{t=1}^{T}(y_t - \phi_t^{\lambda_k}y_{t-1})^2\right) + df(\phi^{\lambda_k})ln(T)/T$$

where $\phi_t^{\lambda_k}$ are the estimated slopes at iteration k, $\hat{\sigma}^2$ is the estimate of σ_ϵ^2 assuming that there are no breaks, and $df(\phi^{\lambda_k})$ are the degrees of freedom at λ_k. The degrees of freedom measures the complexity of the model and here it describes the effective number of parameters that can be used in the fit which here is the number of the estimated segments.

For decreasing λ_k the sum of the estimated residuals are monotonically decreasing and hence the the minimum Cp or $BIC(\lambda_2)$ value will be found somewhere between the critical points. Ideally, that implies that at every iteration we can stop the algorithm as soon as we get a value of C_p or BIC that is larger from the one obtained in the previous iteration. In order to deal with the increasing complexity we simply allow the algorithm to run several steps and then choose the estimated coefficients that returns the global minimum for either criterion. For our setup we simply chose to run the algorithm $K_{max} = 20$ times so that the maximum of breakpoints estimated is K_{max}. This is motivated by [10] who suggest plotting the loss function against the number of breakpoints. This approach looks for the point where the slope changes significantly i.e. it ceases to decrease too fast. Figure (2) plots the estimated variance of the residuals at every step against the number of steps. We observe that the curve stabilises around the true variance without significant drops. In an extensive simulation we find that around 20 iterations are enough in order to get a good estimate of the variance without risking underestimation of the breakpoints.

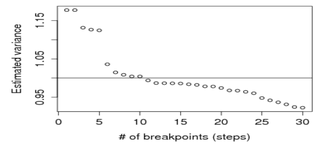

Figure 2: Plot of the estimated variance at every step against the number of the breakpoints (steps).The horizontal line is the variance used to simulate the AR(1) process

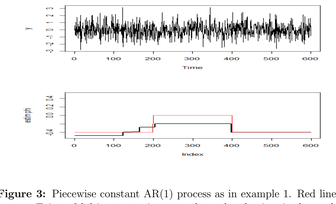

Figure 3: Piecewise constant AR(1) process as in example 1. Red line is the true coefficient. Multisegmentation around true breakpoints is observed.

Simulation Study

We present two different scenarios that are shown below. For the AR(2) case we use the Proximal Gradient method of [7] and for the AR(1) the path algorithm of [15].

In both cases we use the multiresolution criterion to select the optimum segmentation of the series. The two scenarios are:

Example 1:

$$y_t = \begin{cases} -0.4y_{t-1} + \epsilon_t, \ \epsilon_t \sim N(0,1) & \text{for } 1 \leq t \leq 200 \\ \epsilon_t, \ \epsilon_t \sim N(0,1) & \text{for } 201 \leq t \leq 400 \\ -0.4y_{t-1} + \epsilon_t, \ \epsilon_t \sim N(0,1) & \text{for } 401 < t \leq 600 \end{cases}$$

Example 2:

$$y_t = \begin{cases} 0.9y_{t-1} + \epsilon_t, \ \epsilon_t \sim N(0,1) & \text{for } 1 \leq t \leq 100 \\ 1.69y_{t-1} - 0.81y_{t-2} + \epsilon_t, \ \epsilon_t \sim N(0,1) & \text{for } 101 < t \leq 200 \end{cases}$$

We note that the Fused Lasso estimator introduces a bias whose magnitude is a function of the penalty parameter λ, the size of the segments and their relative positions. This can be seen from figures (3) and (4) where estimated coefficients stand between the true ones.

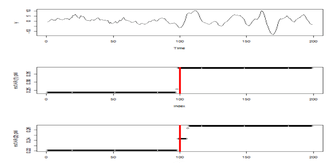

Figure 4: Piecewise constant AR(2) process as in example 2. Vertical red line is the true breakpoint.

References

[1] J. Bai and P. Perron (2003) "Computation and Analysis of Multiple Structural Change Models", *Journal of Appllied Econometrics*, **18**, 1–22.

[2] H. Cho and P. Fryzlewicz (2012) "Multiscale and multilevel technique for consistent segmentation of nonstationary time series", *Statistica Sinica*, **22**, 207–229.

[3] R. Dahlhaus, M. Neumann and R. Von Sachs (1999) "Nonlinear Wavelet Estimation of Time-Varying Autoregressive Processes", *Bernoulli*, **5**, 873–906.

[4] P.L. Davies and A. Kovac (2001) "Local extremes, runs, strings and multiresolution" (with discussion), *Ann. Statist.*, **29**, 1–65.

[5] J. Friedman, T. Hastie, H. Hoefling, and R. Tibshirani (2007) "Pathwise coordinate optimization", *Annals of Applied Statistics*, **1(2)**, 302–332.

[6] P. Fryzlewicz (2007) "Unbalanced Haar technique for nonparametric function estimation", *J. Am. Stat. Assoc.* , **102**, 1318–1327.

[7] H. Chen, S. Kim, Q. Lin, J. Carbonell and E.P. Xing (2010) "Graph-Structured Multi-task regression and an effcient optimization method for General Fused Lasso", Working paper.

[8] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky (2009) "$\lambda - 1$ trend filtering", *SIAM Review*, **51(2)**, 339–360.

[9] M. Lavielle and G. Teyssire (2006) "Adaptive detection of multiple change-points in asset price volatility", *Long Memory in Economics*, 129-156.

[10] M. Lavielle and G. Teyssire (2006) "Detection of multiple change-points in multivariate time series", *Lithuanian Mathematical Journal*, **46(3)**, 287-306.

[11] H. Ombao, J. A. Raz, R. Von Sachs, and B.A Malow (2001) "Automatic Statistical Analysis of Bivariate Nonstationary Time Series", *J. Am. Stat. Assoc.* , **96**, 543–560.

[12] G. P. Nason, R. von Sachs, and G. Kroisandt, (2000) "Wavelet processes and adaptive estimation of the evolutionary wavelet spectrum" , *J. Roy. Stat. Soc. B*, **62**, 271-292.

[13] R. Tibshirani (1996) "Regression shrinkage and selection via the lasso", *J. Roy. Stat. Soc. B*, **58**, 267–288.

[14] R. Tibshirani, M. Saunders, J. Zhu, and S. Rosset (2005) "Sparsity and smoothness via the fused lasso", *J. Roy. Stat. Soc. B*, **67**, 91–108.

[15] R. Tibshirani and J. Taylor (2011) " The Solution Path of the Generalized Lasso", *Annals of Statistics*, **39**, 1335–1371.