A FAST SUBSAMPLING METHOD FOR NONLINEAR DYNAMIC MODELS

HONG, H.[a], SCAILLET, O.[b] and TAMER, E.[c] [1]

[a] Department of Economics, Princeton University, USA and IRES, UCL, Belgium. doubleh@princeton.edu.

[b] HEC, Université de Genève, and FAME, Switzerland. scaillet@hec.unige.ch.

[c] Department of Economics, Princeton University, USA. tamer@princeton.edu.

This version: November 2001 (First version: October 2001)

[1] We would like to thank the participants at the CRDE conference on resampling methods in econometrics for stimulating comments. The first and second author gratefully acknowledge financial support from the Belgian Program on Interuniversity Poles of Attraction (PAI nb. P4/01), while this research has also been supported by the Swiss National Science Foundation through the National Center of Competence Financial Valuation and Risk Management. Part of this research was done when the second author was visiting THEMA and IRES. Downloadable at http://www.hec.unige.ch/professeurs/SCAILLET_Olivier/pages_web/Home_Page_of_Olivier_Scaillet.htm

A FAST SUBSAMPLING METHOD FOR NONLINEAR DYNAMIC MODELS

HONG, H., SCAILLET, O. and TAMER, E.

Abstract

We highlight a fast subsampling method that can be used to provide valid inference in nonlinear dynamic econometric models. This method is based on the subsampling theory proposed by Politis and Romano (1992,1994) which computes an estimator on subsamples of the data and uses these estimates to construct valid inference under very weak assumptions. Fast subsampling directly exploits score functions computed on each subsample and avoids recomputing the estimators for each of them thereby reducing computational time considerably. This method is used to obtain the limit distribution of estimators, possibly simulation based, that admit an asymptotic linear representation with both known and unknown rates of convergence. It can also be used for bias reduction and variance estimation. Monte Carlo experiments demonstrate the desirable performance and vast improvement in numerical speed of the fast subsampling method.

Résumé

Nous présentons une méthode rapide de sous-échantillonnage qui peut être utilisée pour donner une inférence correcte dans des modèles économétriques dynamiques nonlinéaires. Cette méthode est fondée sur la théorie de sous-échantillonnage proposée par Politis et Romano (1992,1994) qui calcule un estimateur sur des sous-échantillons des données et utilise ces estimateurs pour construire une inférence valide sous des hypothèses très faibles. Le sous-échantillonnage rapide exploite directement les fonctions de score calculées sur chaque sous-échantillon et évite de recalculer les estimateurs sur chacun d'eux ce qui réduit le temps de calcul considérablement. Cette méthode est utilisée afin d'obtenir la distribution limite d'estimateurs, fondés éventuellement sur des méthodes simulées, qui admettent une représentation asymptotique linéaire de taux de convergence connu ou inconnu. Elle peut aussi être utilisée à des fins de réduction de bias et d'estimation de variance. Des simulations de Monte Carlo démontrent la performance désirée et l'amélioration considérable en vitesse de calcul de la méthode rapide de sous-échantillonnage.

Keywords: Subsampling, Nonlinear dynamic models, Simulation based estimators.

Mots-clés: Sous-échantillonnage, Modèles nonlinéaires dynamiques, Estimateurs fondés sur des simulations.

JEL Classification: C12, C15, C22, C52.

1 Introduction

Bootstrap, jackknife and other resampling methods have become in recent years a burgeoning area in both theoretical and applied statistics, and are clearly beginning to impact developments in econometric methodology as well as various applied scientific fields. Their primary asset is to provide powerful statistical tools which are easy to implement and work where other more classical tools fail. Among these computer-intensive methods, a very general approach to constructing asymptotically valid inference procedures has been proposed by Politis and Romano (1992,1994). Their approach is based on subsampling and benefits from a wider applicability than the bootstrap and the jackknife. The main idea of subsampling is to evaluate a statistic of interest at subsamples of the data, and to use these subsampled values to build up an estimated sampling distribution. In particular, the consistency properties of this sampling distribution hold for dependent data under very weak assumptions and even in situations where the bootstrap collapses. This attractive feature is the reason why we focus on subsampling methods. Indeed we wish to provide a setting large enough to cover various estimation methods and modelling frameworks.

In several econometric models, estimation of the parameter is computer-intensive and hence requires a considerable amount of time. This is due to complex nonlinearities of the model, or the need to rely on simulations when the model is too difficult to estimate in a direct way, such as in latent factor models, stochastic volatility models and diffusion processes. The objective of the paper is to offer a *fast* procedure that obtains valid inference for parameters of interest while maintaining the attractiveness of the subsampling method. The idea is to avoid estimating the parameter on each subsample but rather use the asymptotic linear representation of the estimator and evaluate that on subsamples of the data. This has been suggested by Heagerthy and Lumley (2000) in variance estimation for asymptotically normally distributed estimators obtained as the root of an additive estimating function. In this paper we propose to use the subsampled values as the building blocks of an *entire estimated sampling distribution*. This allows for constructing confidence regions without the requirement of asymptotic normality. We only need existence of a limit law. Overall, we focus on models with dependent observations where

we consider the stationary and unit roots cases. Deriving asymptotic variances for inference purposes is more challenging under serial correlation. Our framework also includes very general estimating functions, not necessarily additive, which may involve simulated data. Simulation based nonadditive estimating functions are for example involved in the indirect inference method introduced by Gouriéroux, Monfort and Renault (1993). We also develop the case of unknown rate of convergence and eventually show how fast subsampling can be used in bias reduction and variance estimation.

The basis for the material concerning the building of confidence regions and subsampling distribution estimation is laid in Politis and Romano (1994). The design of subsampling intervals in autoregressive models with linear time trend is addressed in Romano and Wolf (2001), and the case of unknown rate of convergence is analyzed in Bertail, Politis, and Romano (1999). For more on the subsampling theory, we refer the reader to the excellent monograph of Politis, Romano and Wolf (PRW) (1999).

Finally let us point out that fast subsampling is the subsampling version of the estimating function bootstrap of the bootstrap literature (see e.g. Shao and Tu (1995), Davidson and McKinnon (1999), Hu and Kalbfleisch (2000), Goncalves and White (2000), Andrews (2001)). Indeed this fast bootstrap also avoids solving an optimisation problem for each resample by relying on resampling the gradient of the nonlinear objective function, hence reducing computational cost of bootstrapping.

The paper is organized as follows. In Section 2 we first outline our framework and recall the standard subsampling methodology. Then we describe how fast subsampling works and how it can be used in building confidence intervals. We consider stationary strong mixing processes and a broad category of direct estimators and simulation based indirect estimators admitting an asymptotically linear representation. We also discuss the modifications to be made in the unit root case. Section 3 explains how fast subsampling can be exploited in bias reduction and variance estimation. Section 4 is devoted to the case of unknown rate of convergence. In Section 5 we show on a set of Monte Carlo experiments that fast subsampling achieves desirable performance and improves computational speed when compared with standard subsampling. Proofs are gathered in an appendix.

2 Framework and subsampling methods

2.1 Stationary case

We consider a real-valued strictly stationary process $\{X_t, t \in \mathbb{Z}\}$, and denote the joint probability law governing the infinite sequence by P. By stationarity all finite-dimensional marginal distributions are shift-invariant. We consider a parametric model indexed by some parameter θ, and the goal is to construct a confidence interval for θ on the basis of observing $\{X_t; t = 1, ..., T\}$. This sequence of observations is assumed to satisfy the weak dependence condition of α-mixing. Let $\hat{\theta}_T$ be an estimator of θ, the parameter of interest, based on the full sample. We assume that this estimator converges to a true value, pseudo true value or indirect pseudo true value, depending on the estimation technique and whether the parameric model is well specified or not (see e.g. WHITE (1982), GOURIEROUX, MONFORT, and TROGON (1984), DHAENE, GOURIEROUX and SCAILLET (1998)). This value is denoted by θ_0.

The usual subsampling method consists in approximating the sampling distribution of an estimator by recomputing it on subsamples of smaller size on the observed data. In the case of stationary time series these subsamples are choosen to be blocks of size b of consecutive observations, the first one being $\{X_1, \ldots, X_b\}$, and the last one $\{X_{T-b+1}, \ldots, X_T\}$. This gives $q = T - b + 1$ blocks. For each block, an estimate of the parameter of interest is computed. Let us define $\hat{\theta}_{T,b,t}$, the estimator of θ based on $\{X_t, \ldots, X_{t+b-1}\}$, and $J_b(P)$, the sampling distribution of $\tau_b(\hat{\theta}_{T,b,1} - \theta_0)$ where τ_b is an appropriate normalizing constant. Since each block is a true subsample of size b from P, the distribution of $\tau_b(\hat{\theta}_{T,b,t} - \theta_0)$ is $J_b(P)$. Hence we expect the empirical distribution of the $T - b + 1$ values of $\tau_b(\hat{\theta}_{T,b,t} - \theta_0)$ to be a good approximation of $J_T(P)$ the sampling distribution of $\tau_T(\hat{\theta}_T - \theta_0)$. Since θ_0 is unknown it will be replaced by $\hat{\theta}_T$ which converges at a faster rate if $\tau_b/\tau_T \to 0$. Then by defining in the univariate parameter case

$$L_{T,b}(x) = \frac{1}{T-b+1} \sum_{t=1}^{T-b+1} \mathbb{1}\{\tau_b(\hat{\theta}_{T,b,t} - \hat{\theta}_T) \leq x\},$$

and assuming $\tau_b/\tau_T \to 0$, $b/T \to 0$ and $b \to \infty$ as $T \to \infty$, it can be shown (see PRW (1999) Theorem 3.2.1) that $L_{T,b}(x)$ converges to the cdf $J(x, P)$ of the limit law, if it exists,

of $J_T(P)$ at point x. Note that the extension to the multivariate case is straightforward by applying an adequate norm on the difference $\tau_b(\hat{\theta}_{T,b,t} - \hat{\theta}_T)$.

As already mentioned, this standard subsampling procedure is suitable if the computational burden associated with the estimation of the parameter of interest is not heavy. This will not be the case if the dynamic model involves lots of nonlinearities, which typically slows down numerical optimization routines, or if an indirect estimation method based on simulations is called for. Note that the computations of the $T - b + 1$ values of $\hat{\theta}_{T,b,t}$ may take a long time in both situations, and refrain the applied econometrician from implementing the subsampling method. We propose hereafter to deal with this problem by directly using estimating (score) functions computed on each subsample.

Assume that $\hat{\theta}_T$ solves the equation $\psi_T(\theta) = 0$, where $\psi_T(\theta)$ is a function involving the observed data $\{X_1, \ldots, X_T\}$, and possibly S simulated samples $\{\tilde{X}_1^s(\theta), \ldots, \tilde{X}_T^s(\theta)\}$, $s = 1, \ldots, S$, obtained from a simulation of the parametric model. This equation may correspond to the first order condition of direct estimation methods, such as maximum likelihood method and generalised method of moments, or indirect estimation methods, such as indirect inference, efficient method of moments, simulated method of moments and simulated pseudo maximum likelihood. In the case of M-estimation, $\psi_T(\theta)$ is equal to the empirical average of the score vector, and is thus additive. A Taylor expansion of $\psi_T(\hat{\theta}_T)$ around θ_0 leads in general to:

$$\tau_T(\hat{\theta}_T - \theta_0) = A_0 \, \tau_T \psi_T(\theta_0) + o_p(1),$$

for some matrix A_0. Note that our method is in fact valid for any statistics, which admit a linear representation of the form:

$$\hat{\theta}_T = \theta(P) + \frac{1}{T} \sum_{t=1}^{T} g_P(X_t) + o_P(1/\tau_T),$$

where g_P corresponds to an appropriately defined notion of derivative, Fréchet derivative for example, of $\theta(\cdot)$ at P (see Section 1.6 of PRW (1999) for further details on linear approximations to statistical functionals).

Assume now that we can consistently estimate A_0 by $\hat{A}_T$. The idea is to approximate $\tau_T \psi_T(\theta_0)$ by $\tau_b \psi_{T,b,t}(\theta_0)$ based on $\{X_t, \ldots, X_{t+b-1}\}$, and $\{\tilde{X}_t^s(\theta_0), \ldots, \tilde{X}_{t+b-1}^s(\theta_0)\}$, $s =$

$1, \ldots, S$. Since each block is a subsample of size b, the distribution of $A_0\,\tau_b\psi_{T,b,t}(\theta_0)$ is $J_b(P)$. Hence we expect the empirical distribution of the $T-b+1$ values of $A_0\,\tau_b\psi_{T,b,t}(\theta_0)$ to be a good approximation of $J_T(P)$. If $\tau_b/\tau_T \to 0$, we may replace the unknown A_0, θ_0 by $\hat{A}_T$, $\hat{\theta}_T$, respectively, and build in the univariate parameter case:

$$L^*_{T,b}(x) = \frac{1}{T-b+1}\sum_{t=1}^{T-b+1} \mathbb{1}\{\hat{A}_T\,\tau_b\psi_{T,b,t}(\hat{\theta}_T) \leq x\}.$$

Let us introduce the following assumption.

Assumption 1

(i) $\hat{\theta}_T = \theta_0 + o_p(1)$.

(ii) $\hat{A}_T = A_0 + o_p(1)$.

(iii) $\tau_T(\hat{\theta}_T - \theta_0) = A_0\,\tau_T\psi_T(\theta_0) + o_p(1)$.

(iv) There exists a limit law $J(P)$ *such that* $J_T(P)$ *converges weakly to* $J(P)$.

(v) The α*-mixing sequence corresponding to* $\{X_t\}$ *is such that* $\alpha_X(m) \to 0$ *as* $m \to \infty$.

In part (i) above, we require that the parameter of interest can be consistently estimated. This is clearly a necessary minimal requirement for what follows. We also ask in (ii) to have a consistent estimator for A_0. Such estimators are available in most cases. The crucial assumption is (iii). The estimator needs to admit an asymptotic linear representation where the sampling distribution of $A_0\tau_T\psi_T(\theta_0)$ becomes $J_T(P)$ as T increases. This assumption excludes estimators like the sample minimum/maximum or non-regular likelihood problems where the support of the likelihood depends on the parameter. The last assumption specifies the weak dependence properties of the data.

The next theorem states that $L^*_{T,b}(x)$ converges to $J(x,P)$ when $\tau_b/\tau_T \to 0$, $b/T \to 0$ and $b \to \infty$ as $T \to \infty$ where $L_T(b)$ is the sampling distribution of $\tau_T(\hat{\theta}_T - \theta_0)$.

Theorem 1 *Let Assumption 1 hold and assume that* $\tau_b/\tau_T \to 0$, $b/T \to 0$ *and* $b \to \infty$ *as* $T \to \infty$.

1. *If* x *is a continuity point of* $J(\cdot, P)$, *then* $L^*_{T,b}(x) \to J(x,P)$ *in probability as* $T \to \infty$.
2. *If* $J(\cdot, P)$ *is continuous, then* $\sup_x |L^*_{T,b}(x) - J(x,P)| \to 0$ *in probability as* $T \to \infty$.
3. *For* $\alpha \in (0,1)$, *let*

$$c_{T,b}(1-\alpha) = \inf\{x : L^*_{T,b}(x) \geq 1-\alpha\}.$$

and

$$c(1-\alpha, P) = \inf\{x : J(x, P) \geq 1-\alpha\}.$$

If $J(\cdot, P)$ is continuous at $c(1-\alpha, P)$, then

$$P[\tau_T(\hat{\theta}_T - \theta_0) \leq c_{T,b}(1-\alpha)] \to 1-\alpha \quad as \quad T \to \infty.$$

Note that the rate of convergence in most parametric econometric models is $\tau_T = \sqrt{T}$ and so assumptions on b then simplify to $b/T \to 0$ and $b \to \infty$. For some simulation based estimation methods, for example simulated pseudo maximum likelihood, the rate of convergence is $\sqrt{TS}$ which yields the simplification $b/(TS) \to 0$ and $b \to \infty$.

Similar results hold for two-sided equal-tailed or symmetric subsampling confidence intervals, as well as for studentised version of statistics (see PRW (1999) for further developments). The multivariate parameter case can also be easily handled.

Besides the fast subsampling method can be applied to test statistics which depend on the difference between a constrained estimator and an unconstrained estimator of θ. Let us keep the notation $\hat{\theta}_T$ for the unconstrained estimator, and use $\bar{\theta}_T$ for the estimation under the restriction $g(\theta) = 0$. Under the null hypothesis $H_0 : g(\theta_0) = 0$, we have (see Gourieroux and Monfort (1995) Sections 17 an 18):

$$\tau_T(\hat{\theta}_T - \bar{\theta}_T) = (Id - M'(\theta_0))\tau_T(\hat{\theta}_T - \theta_0) + o_p(1),$$

where

$$M(\theta_0) = Id - \frac{\partial g'(\theta_0)}{\partial \theta}\left(\frac{\partial g(\theta_0)}{\partial \theta'} A_0 \frac{\partial g'(\theta_0)}{\partial \theta}\right) \frac{\partial g(\theta_0)}{\partial \theta'} A_0$$

is the projection matrix associated to the orthogonal projection onto the subspace that is orthogonal to the subspace generated by the columns $\partial g'(\theta_0)/\partial\theta$ with repect to the scalar product defined by A_0. We may then think of approximating $\tau_T(\hat{\theta}_T - \theta_0)$ by $A_0\, \tau_b \psi_{T,b,t}(\theta_0)$ as done in the previous lines. This will allow to build subsampling procedures for Hausman-Wald type tests.

2.2 Unit root case

Assumption 1 is devoted to the case of a matrix $\hat{A}_T$ which converges to a non random limit A_0. It may happen that $\hat{A}_T$ diverges, but $\tau_T \hat{A}_T$ converges to a random limit. A leading example is the unit root process: $X_t = X_{t-1} + \epsilon_t$, where $\{\epsilon_t\}$ is a strictly stationary white noise innovation sequence with variance σ^2. Let us consider the OLS estimator $\hat{\rho}_T$ derived from the estimating function

$$\psi_T(\rho) = \frac{1}{T} \sum_{t=2}^{T} X_{t-1}(X_t - \rho X_{t-1}).$$

We know that $\hat{\rho}_T$ converges to $\rho_0 = 1$, but

$$\hat{A}_T = \left(\frac{1}{T} \sum_{t=2}^{T} X_{t-1}^2 \right)^{-1}$$

diverges. Nevertheless $T\hat{A}_T$ converges to the random limit $1/\left(\sigma^2 \int_0^1 W^2(r)dr\right)$, while $\psi_T(\rho_0)$ converges to the random limit $\sigma^2(W^2(1) - 1)/2$, where $W(\cdot)$ denotes a standard Brownian motion. Hence $\hat{A}_T\, T\psi_T(\hat{\rho}_T)$ admits the random limit $(W^2(1)-1)/(2\int_0^1 W^2(r)dr)$.

To manage this situation as well as more general martingale difference innovation sequences we may substitute the following assumption for Assumption 1. Theorem 1 will then still hold but with $L^*_{T,b}(x)$ defined as

$$L^*_{T,b}(x) = \frac{1}{T-b+1} \sum_{t=1}^{T-b+1} \mathbb{1}\{\hat{A}_{T,b,t}(\hat{\theta}_T)\, \tau_b \psi_{T,b,t}(\hat{\theta}_T) \leq x\}.$$

Assumption 2

(i) $\hat{\theta}_T = \theta_0 + o_p(1)$.

(ii) $\tau_T(\hat{\theta}_T - \theta_0) = \hat{A}_T(\theta_0)\tau_T\psi_T(\theta_0) + o_p(1)$.

(iii) There exists a limit law $J(P)$ such that $J_T(P)$ converges weakly to $J(P)$, and for every continuity point x of $J(P)$, we have $\dfrac{1}{T-b+1} \sum_{t=1}^{T-b+1} J_{b,t} \to J(x,P)$ *for any sequence T, b with $T, b \to \infty$ and $b/T \to 0$.*

(iv) The α-mixing sequence corresponding to $\{Z_{T,b,t}\}$ is such that $T^{-1}\sum_{h=1}^{T} \alpha_{T,b}(h) \to 0$ as $T \to \infty$, where $Z_{T,b,t} = \hat{A}_{T,b,t}(\theta_0)\, \tau_b \psi_{T,b,t}(\theta_0)$.

Assumptions (iii) and (iv) correspond to the assumptions needed for Theorem 12.2.1 in PRW (1999), which covers the random walk case as a particular example. From a numerical point of view this second method will be slower than the previous one since we need to recompute $\hat{A}_{T,b,t}(\hat{\theta}_T)$, namely the inverse of the Hessian matrix on each subsample. However $\hat{\theta}_T$ stays fixed across subsamples.

Finally note that application of Theorem 1 requires the knowledge of the rate of convergence. The rate of convergence of the OLS estimator $\hat{\rho}_T$ is given by T in the unit root case and by $\sqrt{T}$ in the stationary case. In order to avoid the issue of choosing between the two cases, we may consider a studentised statistic, namely the usual t-statistic for $\hat{\rho}_T$, as advocated by ROMANO and WOLF (2001). Indeed this statistic has a proper limit distribution no matter what the value of ρ_0. The necessary theory to apply fast subsampling in a studentised setting can be derived from PRW (1999) Section 12.2.2 and ROMANO and WOLF (2001).

3 Bias reduction and variance estimation

The fast subsampling methodology may be a useful bias reduction device. Indeed we may approximate the bias $Bias(\hat{\theta}_T) = E[\hat{\theta}_T - \theta_0]$ by the empirical counterpart:

$$\widehat{Bias}(\hat{\theta}_T) = \frac{\tau_b}{\tau_T}\left[\frac{1}{q}\sum_{t=1}^{q}\hat{A}_T\psi_{T,b,t}(\hat{\theta}_T)\right], \qquad q = T - b + 1,$$

and put

$$\hat{\theta}_{T,BC} = \hat{\theta}_T - \widehat{Bias}(\hat{\theta}_T).$$

The estimator $\hat{\theta}_{T,BC}$ will be a bias corrected estimator of the parameter of interest in virtue of the next theorem similar to PRW (1999) Theorem 3.7.1. Denote by $m_{T,b}$, μ_T, μ, the respective means of $L^*_{T,b}$, J_T and J.

Theorem 2 *Assume Assumption 1 and that $\tau_b/\tau_T \to 0$, $b/T \to 0$ and $b \to \infty$ as $T \to \infty$. Also assume that $\mu_T \to \mu$, and $E|\hat{\psi}^*_{T,b,1}(\theta_0)|^{2+\delta} < C$, where δ, C are two positive constants independent of T and $\psi^*_{T,b,t}(\theta_0) = \psi_{T,b,t}(\theta_0)/\sqrt{Var[\psi_{T,b,1}(\theta_0)]}$. Then $|m_{T,b} - \mu_T| \to 0$ in probability.*

Subsampling techniques may also provide variance estimator. Since the variance of $\tau_T\hat{\theta}_T$ can be approximated by the variance of $L^*_{T,b}$, an estimator of the variance of $\hat{\theta}_T$ is simply in the univariate case:

$$\hat{V}ar(\hat{\theta}_T) = \frac{\tau_b^2}{\tau_T^2}\frac{1}{q}\sum_{t=1}^{q}\left(\hat{A}_T\psi_{T,b,t}(\hat{\theta}_T) - \hat{A}_T\bar{\psi}_{T,b}\right)^2$$

with $q = T - b + 1$ and $\bar{\psi}_{T,b} = \dfrac{1}{q}\sum_{t=1}^{q}\psi_{T,b,t}(\hat{\theta}_T)$.

The consistency of $\tau_T^2\hat{V}ar(\hat{\theta}_T)$ is a direct consequence of Lemma 3.8.1 in PRW (1999). This lemma establishes the convergence to $\sigma_\infty^2 = \lim_{T\to\infty} Var(\hat{\theta}_T)$ in L^2 norm.

4 Unknown rates of convergences

In the previous section, the rate of convergence τ_T was assumed to be known. Sometimes it may not be the case, and the only available information is $\tau_T = T^\beta$ for some $\beta > 0$. Then the methods described in PRW (1999) Section 8 can be used to obtain consistent estimate for β and subsequently be used in a second step inference procedure. For b denoting the subsample size, and $q = T - b + 1$, define

$$L_{T,b}\left(x|\tau_b\right) = \frac{1}{q}\sum_{t=1}^{q}\mathbb{I}\{\hat{A}_T\,\tau_b(\psi_{T,b,t}(\hat{\theta}_T) - \psi_T(\hat{\theta}_T)) \le x\},$$

and

$$L_{T,b}\left(x|1\right) = \frac{1}{q}\sum_{t=1}^{q}\mathbb{I}\{\hat{A}_T(\psi_{T,b,t}(\hat{\theta}_T) - \psi_T(\hat{\theta}_T)) \le x\}.$$

This implies that $L_{T,b}\left(x|\tau_b\right) = L_{T,b}\left(\tau_b^{-1}x|1\right) \equiv \nu$, and therefore

$$x = L_{T,b}^{-1}\left(\nu|\tau_b\right) = \tau_b\left(\tau_b^{-1}x\right) = \tau_b L_{T,b}^{-1}\left(\nu|1\right),$$

where G^{-1} denotes the quantile transformation of a given distribution G. Since $L_{T,b}(x|\tau_b)$ will converge uniformly to $J(x,P)$ (cf. Statement 2 of Theorem 1), it can be shown (PRW (1999) Lemma 8.2.1) that if $J(x,P)$ is continuous and strictly increasing on its support,

$$L_{T,b}^{-1}\left(\nu|\tau_b\right) = J^{-1}\left(\nu,P\right) + o_p\left(1\right),$$

or equivalently

$$\tau_b L_{T,b}^{-1}(\nu|1) = J^{-1}(\nu, P) + o_p(1).$$

The last expression implies that $L_{T,b}^{-1}(\nu|1)$ is approximately proportional to $\tau_b^{-1} = (b^\beta)^{-1}$. Therefore it should be possible to determine β by constructing several subsampling distributions. Indeed let us take b_1 and b_2. We get

$$\log\left(\frac{b_1}{b_2}\right)^{-1}\left(\log\left(L_{T,b_2}^{-1}(\nu|1)\right) - \log\left(L_{T,b_1}^{-1}(\nu|1)\right)\right) = \beta + o_p\left(\log\left(\frac{b_1}{b_2}\right)^{-1}\right).$$

The idea is then to average the left hand side over several ν_j to estimate β. In fact if we consider different b_i, $i = 1, \ldots, \mathcal{J}$, and take points $\nu_{2j} \in (0.5, 1)$ and $\nu_{1j} \in (0, 0.5)$, for $j = 1, \ldots, \mathcal{J}$, the estimator β will consist in

$$\beta_{\mathcal{I},\mathcal{J}} = -\frac{\sum_{i=1}^{\mathcal{I}}(y_{i,\cdot} - \bar{y})\left(\log b_i - \bar{\log}\right)}{\sum_{i=1}^{\mathcal{I}}\left(\log b_i - \bar{\log}\right)^2}$$

where

$$y_{i,\cdot} = \frac{1}{\mathcal{J}}\sum_{j=1}^{\mathcal{J}}\left(\log\left(L_{T,b_i}^{-1}(\nu_{2j}|1)\right) - \log\left(L_{T,b_i}^{-1}(\nu_{1j}|1)\right)\right),$$

$$\bar{y} = \frac{1}{\mathcal{I}\mathcal{J}}\sum_{i=1}^{\mathcal{I}}\sum_{j=1}^{\mathcal{J}}\left(\log\left(L_{T,b_i}^{-1}(\nu_{2j}|1)\right) - \log\left(L_{T,b_i}^{-1}(\nu_{1j}|1)\right)\right),$$

and

$$\bar{\log} = \frac{1}{\mathcal{I}}\sum_{i=1}^{\mathcal{I}}\log b_i.$$

Theorem 8.2.2 in PRW (1999) establishes that $\beta_{\mathcal{I},\mathcal{J}} = \beta + o_p(1/\log T)$, while Theorem 8.3.1 shows the asymptotic validity of the second step procedure, namely subsampling with an estimated rate of convergence $\hat{\tau}_T = T^{\beta_{\mathcal{I},\mathcal{J}}}$. Their adaptation to our setting is straightforward.

5 Monte Carlo experiments

This section aims to examine the small sample performance of the fast subsampling method via Monte Carlo experiments. We follow closely PRW (1999) Sections 9.5 and 12.5 for

their design. This will ease comparison with previous studies on the performance of subsampling methods. Performance is assessed for nominal 95% two sided confidence intervals by measuring estimated true coverage probability. We first analyse results for AR(1) processes and linear regressions. Then we study ARCH(1) processes. Finally the unit root case is analysed.

5.1 AR(1) process

The first Monte Carlo study is concerned with the parameter ρ of a simple stationary AR(1) process:

$$X_t = \rho X_{t-1} + \epsilon_t,$$

where $\{\epsilon_t\}$ is white noise. We use the estimating equation (PML estimation of order one with Gaussian pseudo family):

$$\psi_T(\rho) = \frac{1}{T}\sum_{t=2}^{T} X_{t-1}(X_t - \rho X_{t-1}) = 0,$$

and

$$\hat{A}_T = \left(\frac{1}{T}\sum_{t=2}^{T} X_{t-1}^2\right)^{-1}.$$

Here $\hat{\rho}_T$ takes the form of the standard OLS estimator.

In the following we wish first to examine the impact of block size on performance. We take a sample length $T = 256$, and block sizes $b = 4, 8, 16, 32$ and 64. The coverage probability is estimated as the proportion of 95% confidence intervals that contain the true value ρ_0. This proportion is computed on 3000 random samples. Innovations are chosen to be i.i.d. from either a standard normal distribution or an exponential distribution with parameter one shifted to have mean zero. The true values of the parameter of interest are $\rho_0 = 0.2, 0.5, 0.8, 0.95$ and -0.5. Note that we adopt the same design as PRW (1999), but their study is dedicated to the mean.

Tables 1 and 2 give the estimated coverage probabilities for equal tailed and symmetric intervals both under the fast subsampling method and the standard subsampling method, for the normal and exponential errors, respectively. It is clear that the fast subsampling

method does equally well and sometimes outperforms the standard subsampling method. Block sizes 4, 8 and 16, all give very precise coverage for both types of confidence intervals. It can be noticed that given our sample size $T = 256$, performance of both subsampling methods deteriorates for the large block sizes $b = 32$ and $b = 64$. This is not surprising since a large number of independent blocks is typically required to obtain precise coverage in subsampling. With sample size 256 and block size 64, we only have four roughly independent sample observations in the empirical distribution of the subsample statistics. Although overlapping blocks are used, these are highly dependent with each other. The drop in coverage is thus easy to understand. For b too close to T all statistics $\hat{\theta}_{T,b,t}$ in standard subsampling will almost equal to $\hat{\theta}_T$ resulting in the subsampling distribution being too tight (see for example the discussion in Section 9 of PRW (1999)). The same phenomenon will occur in fast subsampling since the subsample scores $\psi_{T,b,t}(\hat{\theta}_T)$ will only differ from each other by a small amount. Given that we draw here from independent normal and exponential errors, a block size $b = 4$ already gives a good distributional approximation. The crucial need for a large number of independent blocks has been confirmed by increasing the sample size to 4096 (results not reported here). In that case block size 64 gives very similar results to the first three rows of Tables 1 and 2.

In principle, a numerical comparison in terms of computational speed is not interesting in this first set of Monte Carlo experiments since both techniques involve close amount of computational load. In fact, this can be seen from the last two columns. The computational times are essentially identical for the two methods. Computational times are total computational times over the 3000 simulations in seconds.

5.2 Linear regressions

The second set of Monte Carlo experiments is concerned with multivariate least squares linear regression. We examine the model:

$$y_t = x_t'\beta + \epsilon_t,$$

where $x_t' = (1, x_{1t}, \ldots, x_{4t})$. We are interested in constructing confidence intervals for the second coordinate of β. The true value β_0 is set equal to zero. The estimator $\hat{\beta}_T$ is the

least squares estimator, and the fast subsampling method relies on

$$\hat{A}_T \sqrt{b} \psi_{T,b,t}(\hat{\beta}_T),$$

with

$$\psi_{T,b,t}(\hat{\beta}_T) = \frac{1}{b} \sum_{h=0}^{b-1} x_{1t+h} \left(y_{t+h} - x'_{t+h} \hat{\beta}_T \right),$$

and

$$\hat{A}_T = \left(\frac{1}{T} \sum_{t=1}^{T} x_{1t}^2 \right)^{-1}.$$

Note that we do not need to use the full score vector, but only its second coordinate.

Four data generating processes are employed:

$$\begin{aligned}
\text{AR(1)-HOMO}: &\quad x_{jt} = \rho x_{jt-1} + \eta_{jt}, \quad \epsilon_t = \rho \epsilon_{t-1} + \eta_t^\epsilon \\
\text{AR(1)-HET}: &\quad x_{jt} = \rho x_{jt-1} + \eta_{jt}, \quad \tilde{\epsilon}_t = \rho \tilde{\epsilon}_{t-1} + \eta_t^\epsilon, \quad \epsilon_t = |x_{2t}| \tilde{\epsilon}_t \\
\text{AR(1)-SEASON}: &\quad x_{jt} = \rho x_{jt-1} + \eta_{jt}, \quad \epsilon_t = \rho \epsilon_{t-1} + a_t \eta_t^\epsilon, \\
\text{MA(1)-HOMO}: &\quad x_{jt} = \eta_{jt} + \delta \eta_{jt-1}, \quad \epsilon_t = \eta_t^\epsilon + \delta \eta_{t-1}^\epsilon,
\end{aligned}$$

where $\{\eta_{jt}\}$ and $\{\eta_t^\epsilon\}$ are independent i.i.d. innovation sequences, and $\{a_t\}$ is made of repeated sequences $\{1, 1, 1, 2, 3, 1, 1, 1, 1, 2, 4, 6\}$. Innovation distribution is either standard normal or centered exponential with variance one. Values for the parameters ρ and δ are 0.2, 0.5, 0.8, 0.95, and -0.5. Sample size and block sizes are again $T = 256$, and $b = 8, 16, 32, 64$. Coverage probability is assessed on 3000 random samples for each scenario.

Tables 2 to 10 report results for all above configurations. It is not possible to use block size 4 for the standard subsampling method since it would be inverting a singular matrix. On the other hand, it is still possible to use the fast subsampling method with $b = 4$ and very precise coverage rates have been obtained (results not reported here). In fact, even with block size 8, the standard subsampling method needs inversion of a near to singular matrix, which yields a coverage rate very close to 1 as we can see from the tables. On the other hand, the fast subsampling method does fairly well. For block sizes 16 to 64, both methods give comparable and reasonable coverage rates (previous remarks for large block sizes also apply here). Note that the standard subsampling tends in general to give higher

estimated coverage probabilities than the fast one. The improvement in the computational speed offered by the fast subsampling method is obvious from the two last columns. This is essentially due to avoiding inversion of a new matrix for each subsample. This time consuming numerical operation is needed to get $\hat{\beta}_{T,t,b}$ in the standard subsampling method. In fact in most cases the fast subsampling method takes less than half the time that the standard subsampling method asks for.

5.3 ARCH(1) process

We analyse a simple Gaussian ARCH(1) model:

$$\begin{aligned} X_t &= \epsilon_t \sqrt{h_t}, \\ h_t &= \beta_1 + \beta_2 X_{t-1}^2, \end{aligned}$$

where $\{\epsilon_t\}$ is a sequence of i.i.d. normal disturbances. We analyse the second coordinate of $\theta = (\beta_1, \beta_2)'$. The estimator $\hat{\theta}_T$ is obtained through an ML criterion. The fast subsampling method is run using the score and the outer product version of the information matrix. Parameter β_1 takes value 0.5, while β_2 takes value in 0.2, 0.5, 0.8 and 0.95. We keep a sample size $T = 256$ and block sizes $b = 4, 8, 16, 32$ and 64. We draw 3000 samples each time. A quasi-Newton optimization (routine dfpmin.c from numerical recipes in C) is used to perform estimation of $\theta = (\beta_1, \beta_2)'$.

Variation of performance according to block size is given in Table 11. Due to the nonlinear nature of the problem, the coverage performance is not as good as previously. On the other hand, the requirement of nonlinear optimization highlights the advantage of the fast subsampling method over the standard subsampling method in terms of both precision and computational speed. In fact we find that the standard subsampling method does not perform very well. Estimated coverage probabilities are far away from the nominal ones. On the contrary, the fast subsampling method gives very reasonable approximation to the nominal coverage probability, although as we expect for a nonlinear problem, the approximation is not as good as in the two previous simple linear examples. Furthermore, as shown in the last two columns of the tables, the gain in numerical speed is striking. In

fact, the fast subsampling method reduces the computational time essentially by a factor of 20. The explanation is that a numerical optimisation routine is no more necessary in the fast subsampling method once $\hat{\theta}_T$ has been obtained.

5.4 Unit root process

In Table 12 we finally report the results for the unit root process discussed in Section 2.2 with an innovation sequence $\{\epsilon_t\}$ taken as i.i.d. standard normal. As in ROMANO and WOLF (2001), we also consider in Table 13 the case $\epsilon_t = Z_{t-1}Z_t$ where the Z_t are i.i.d. standard normal. In the latter case the innovations are a martingale difference sequence but dependent. Performances of the standard subsampling method and the fast subsampling method are compared for the OLS estimator $\hat{\rho}_T$ and its associated t-statistic for $T = 256$, $b = 4, 8, 16, 32, 64$ and 3000 samples.

Fast subsampling method does fairly well in both cases, while the standard subsampling method tends to produce confidence intervals that are either too wide or too tight. Numerical speed is similar for both methods as expected.

6 Conclusion

We have proposed a fast subsampling method. The method directly exploits estimating (score) functions computed on each subsample and avoids recomputing the estimators for each of them. Fast subsampling is easy to perform, and achieves satisfactory performance while improving considerably numerical speed. These advantages should be of interest for applied researchers using nonlinear and dynamic models to conduct effective inference.

APPENDIX

Proof of Theorem 1

The proof parallels the proof of Theorem 3.2.1 in PRW (1999).

Let us define

$$U_T(x) = \frac{1}{q}\sum_{t=1}^{q} \mathbb{1}\{A_0\,\tau_b \psi_{T,b,t}(\theta_0) \leq x\},$$

with $q = T - b + 1$.

Since $E[U_T(x)] = J_b(x, P)$ and Assumption (iv) holds we only need to show that $\mathrm{Var}[U_T(x)]$ tends to zero as T goes to infinity. Define:

$$s_{q,h} = \frac{1}{q}\sum_{t=1}^{q-h} \mathrm{Cov}[I_{b,t}, I_{b,t+h}],$$

where $I_{b,t} = \mathbb{1}\{A_0\,\tau_b \psi_{T,b,t}(\theta_0) \leq x\}$.

Then

$$\begin{aligned} \mathrm{Var}[U_T(x)] &= \frac{1}{q}(s_{q,0} + 2\sum_{h=1}^{q-1} s_{q,h}) \\ &= \frac{1}{q}(s_{q,0} + 2\sum_{h=1}^{b-1} s_{q,h} + 2\sum_{h=b}^{b-1} s_{q,h}), \end{aligned}$$

which can be rewritten $M^* + M$, with $M^* = \frac{1}{q}(s_{q,0} + 2\sum_{h=1}^{b-1} s_{q,h})$ and $M = \frac{2}{q}\sum_{h=b}^{b-1} s_{q,h}$. Since $|M^*| = O(b/q)$, M^* converges to zero. M will also converge to zero due to Assumption (v). Hence the first statement is shown. Proofs of the second and third statements are identical to the lines in PRW (1999).

Proof of Theorem 2

The proof mimicks the proof of Theorem 3.7.1 in PRW (1999).

We have

$$E\left[\frac{\tau_b}{q}\sum_{t=1}^{q} A_0 \psi_{T,b,t}(\theta_0)\right] = \mu_b,$$

and $|\mu_b - \mu_T| \to 0$. Hence we only need to show that $\mathrm{Var}[m_{T,b}]$ tends to zero as T goes to infinity. This is deduced from

$$\text{Var}\left[\frac{\tau_b}{q}\sum_{t=1}^{q} A_0\psi_{T,b,t}(\theta_0)\right] = \tau_b^2 A_0^2\left[\frac{1}{q}\text{Var}[\psi_{T,b,1}(\theta_0)] + \sum_{\substack{t,t'=1\\ t\neq t'}}^{q}\text{Cov}[\psi_{T,b,t}(\theta_0), \psi_{T,b,t'}(\theta_0)]\right],$$

which leads to

$$|\text{Var}\left[\frac{\tau_b}{q}\sum_{t=1}^{q}\hat{A}_0\psi_{T,b,t}(\theta_0)\right]| \leq \frac{C_1}{q}\sum_{t=1}^{q-1}|\text{Cov}[\psi^*_{T,b,1}(\theta_0), \psi^*_{T,b,1+t}(\theta_0)]|,$$

using $\tau_b^2 = O(1/\text{Var}[\psi_{T,b,1}(\theta_0)])$ and some constant C_1. The convergence is then obtained by application of Davydov's mixing inequality.

Table 1: **Monte Carlo Simulations I: AR(1) process**

Sample size=256, nsim=3000, Exponential innovations

	equal tail		symmetric		time	
	standard	fast	standard	fast	standard	fast
	$\rho = 0.20$					
$b = 4$	0.937333	0.926333	0.946000	0.915333	4.924219	1.074409
$b = 8$	0.915667	0.931333	0.935333	0.909333	1.209093	1.245713
$b = 16$	0.877667	0.907333	0.909333	0.907667	1.299703	4.668300
$b = 32$	0.837667	0.862000	0.874667	0.863333	1.601945	4.304005
$b = 64$	0.769000	0.785667	0.800667	0.793667	4.061635	2.068239
	$\rho = 0.50$					
$b = 4$	0.937667	0.935000	0.961667	0.924667	1.069319	1.072294
$b = 8$	0.901000	0.933333	0.928333	0.918667	1.205699	4.755776
$b = 16$	0.877000	0.898667	0.913000	0.905333	1.300249	1.331772
$b = 32$	0.845333	0.866667	0.876000	0.868333	4.397887	4.301340
$b = 64$	0.772667	0.779333	0.810333	0.787000	1.940223	5.932426
	$\rho = 0.80$					
$b = 4$	0.970000	0.936000	0.987667	0.927667	1.056460	1.062079
$b = 8$	0.930000	0.919667	0.973000	0.922667	1.192468	1.235280
$b = 16$	0.887000	0.890333	0.950333	0.915333	1.290944	4.675033
$b = 32$	0.849333	0.846000	0.911000	0.862333	1.595596	4.306672
$b = 64$	0.789000	0.783000	0.835667	0.786333	4.060219	2.067516
	$\rho = 0.95$					
$b = 4$	0.997333	0.944333	0.997667	0.951333	1.050101	1.063628
$b = 8$	0.985333	0.918667	0.991000	0.943000	1.186962	4.765824
$b = 16$	0.948333	0.870000	0.980000	0.924667	1.285051	1.321090
$b = 32$	0.901333	0.841333	0.945000	0.872000	4.408379	1.689679
$b = 64$	0.810333	0.766667	0.864000	0.792000	4.062242	2.065544
	$\rho = -0.50$					
$b = 4$	0.961667	0.959333	0.977000	0.942333	1.056712	4.936795
$b = 8$	0.890667	0.962667	0.958000	0.940667	1.190533	1.239859
$b = 16$	0.851667	0.932333	0.919333	0.922667	1.286913	4.670065
$b = 32$	0.842000	0.867667	0.892333	0.865667	1.591273	4.303820
$b = 64$	0.769333	0.781667	0.810000	0.786667	4.064861	2.066407

Table 2: **Monte Carlo Simulations I: AR(1) process**

Sample size=256, nsim=3000, Normal innovations

	equal tail		symmetric		time	
	standard	fast	standard	fast	standard	fast
	$\rho = 0.20$					
$b = 4$	0.924000	0.960667	0.933000	0.949667	1.122589	1.123548
$b = 8$	0.881333	0.947667	0.902000	0.934333	1.266405	4.704120
$b = 16$	0.889000	0.925333	0.909333	0.919000	1.348243	1.383722
$b = 32$	0.856667	0.892000	0.873333	0.890000	4.352598	4.251291
$b = 64$	0.795000	0.787333	0.807000	0.797667	4.009319	2.117345
	$\rho = 0.50$					
$b = 4$	0.938333	0.952333	0.958000	0.946667	1.112640	1.113706
$b = 8$	0.890333	0.942667	0.934333	0.945000	1.249227	4.711506
$b = 16$	0.868000	0.920000	0.910667	0.918667	1.343252	1.378262
$b = 32$	0.864000	0.875000	0.888667	0.876333	4.354286	4.254739
$b = 64$	0.790333	0.804333	0.819667	0.806000	4.009632	2.116778
	$\rho = 0.80$					
$b = 4$	0.975333	0.929333	0.991333	0.946333	1.106057	1.107263
$b = 8$	0.915667	0.904333	0.978333	0.931000	1.238486	4.718862
$b = 16$	0.880000	0.887000	0.950667	0.922333	1.337476	1.374065
$b = 32$	0.839333	0.860000	0.913333	0.877333	4.357022	4.258802
$b = 64$	0.775000	0.782667	0.832333	0.795667	4.012489	2.115786
	$\rho = 0.95$					
$b = 4$	0.997667	0.934667	0.998000	0.956667	1.099281	1.110624
$b = 8$	0.980000	0.886667	0.992667	0.935000	1.236251	4.718865
$b = 16$	0.955333	0.858333	0.981667	0.923000	1.332574	1.368064
$b = 32$	0.897333	0.836333	0.951667	0.886667	4.361264	4.262167
$b = 64$	0.806000	0.764667	0.858667	0.785000	4.014762	2.114612
	$\rho = -0.50$					
$b = 4$	0.942333	0.950000	0.954000	0.951000	1.111129	1.114435
$b = 8$	0.901333	0.946667	0.930667	0.942000	1.243929	4.708704
$b = 16$	0.877333	0.910667	0.918667	0.913667	1.338614	1.381167
$b = 32$	0.862333	0.873000	0.884000	0.877333	4.357562	4.251439
$b = 64$	0.787000	0.796333	0.801333	0.805333	4.014851	2.118232

Table 3: **Monte Carlo Simulations II: linear regressions**

Sample size=256, nsim=3000, Exponential innovations, AR(1)-HOMO

	equal tail		symmetric		time	
	standard	fast	standard	fast	standard	fast
	$\beta = 0.20$					
$b = 8$	0.999667	0.945667	1.000000	0.928333	22.221629	11.344490
$b = 16$	0.980667	0.914000	0.987333	0.908667	28.620526	14.373252
$b = 32$	0.927667	0.881333	0.935000	0.888333	45.486909	20.195992
$b = 64$	0.825667	0.791667	0.846333	0.800000	73.163079	26.323384
	$\beta = 0.50$					
$b = 8$	0.996667	0.946000	0.998667	0.919667	22.214799	11.350676
$b = 16$	0.967000	0.907333	0.977333	0.903000	28.612346	14.377046
$b = 32$	0.909333	0.869667	0.925667	0.864333	48.531839	17.800292
$b = 64$	0.806000	0.785333	0.842667	0.787333	73.173226	29.672025
	$\beta = 0.80$					
$b = 8$	0.957333	0.906667	0.977000	0.871000	19.930448	11.349569
$b = 16$	0.871333	0.880000	0.912000	0.870333	31.251467	11.619685
$b = 32$	0.836333	0.838000	0.883333	0.833333	48.498704	20.204919
$b = 64$	0.783667	0.737000	0.840333	0.744000	71.108355	26.319230
	$\beta = 0.95$					
$b = 8$	0.736333	0.725333	0.787333	0.684000	20.075695	11.351645
$b = 16$	0.621333	0.770333	0.698667	0.746667	29.005545	14.381656
$b = 32$	0.655667	0.719667	0.742667	0.712000	45.745910	20.211683
$b = 64$	0.628667	0.591667	0.742333	0.592333	73.060444	26.308645
	$\beta = -0.50$					
$b = 8$	0.997333	0.947667	0.999333	0.929000	22.286640	11.345515
$b = 16$	0.971333	0.923000	0.976667	0.913000	28.622931	14.377980
$b = 32$	0.918000	0.889333	0.925000	0.883333	45.479240	20.203520
$b = 64$	0.830667	0.776333	0.847333	0.791333	73.148033	26.314054

Table 4: **Monte Carlo Simulations II: linear regressions**

Sample size=256, nsim=3000, Normal innovations, AR(1)-HOMO

	equal tail		symmetric		time	
	standard	fast	standard	fast	standard	fast
	$\beta = 0.20$					
$b = 8$	0.999000	0.937333	0.999000	0.930000	22.074718	11.070202
$b = 16$	0.962667	0.911333	0.967667	0.912000	31.143926	14.130420
$b = 32$	0.910667	0.875000	0.913000	0.885333	45.685491	20.000170
$b = 64$	0.826000	0.809667	0.836667	0.816333	71.100299	29.411436
	$\beta = 0.50$					
$b = 8$	0.994333	0.929667	0.996000	0.910000	22.001972	8.919144
$b = 16$	0.935333	0.915000	0.952000	0.917333	31.155503	14.133083
$b = 32$	0.887333	0.867333	0.909000	0.872667	48.347879	20.000216
$b = 64$	0.812667	0.793333	0.834667	0.798667	71.096207	29.418522
	$\beta = 0.80$					
$b = 8$	0.921000	0.894333	0.947333	0.874000	20.184567	11.086873
$b = 16$	0.828333	0.884667	0.872333	0.872667	31.010423	14.146991
$b = 32$	0.820000	0.838000	0.868000	0.840333	48.309074	20.007986
$b = 64$	0.757333	0.747333	0.819667	0.755000	71.098799	26.578197
	$\beta = 0.95$					
$b = 8$	0.661000	0.735667	0.716667	0.699667	20.351245	11.087034
$b = 16$	0.580333	0.771333	0.652000	0.740333	29.242647	14.142852
$b = 32$	0.602667	0.711333	0.696667	0.711333	48.071880	20.019547
$b = 64$	0.618667	0.608000	0.728333	0.621000	71.212956	29.421738
	$\beta = -0.50$					
$b = 8$	0.994667	0.939000	0.996667	0.928000	22.083764	11.079350
$b = 16$	0.951333	0.911000	0.958667	0.908667	31.145863	14.138691
$b = 32$	0.899333	0.873667	0.907000	0.877667	48.328521	20.007705
$b = 64$	0.804667	0.797000	0.820667	0.805000	71.109755	26.585338

Table 5: **Monte Carlo Simulations II: linear regressions**

Sample size=256, nsim=3000, Exponential innovations, AR(1)-HET

	equal tail		symmetric		time	
	standard	fast	standard	fast	standard	fast
	$\beta = 0.20$					
$b = 8$	0.852333	0.947000	0.863667	0.878667	22.153899	11.295578
$b = 16$	0.762000	0.907000	0.819667	0.891000	28.675303	14.353669
$b = 32$	0.760333	0.841667	0.808000	0.834000	48.451778	17.771825
$b = 64$	0.689000	0.737667	0.744000	0.748333	73.116664	29.616406
	$\beta = 0.50$					
$b = 8$	0.886667	0.953333	0.891333	0.884667	22.141849	8.706913
$b = 16$	0.796333	0.911667	0.830333	0.894667	31.336670	14.353665
$b = 32$	0.761000	0.844667	0.802333	0.828333	45.562938	20.227019
$b = 64$	0.717000	0.740000	0.774333	0.735333	73.129074	26.383580
	$\beta = 0.80$					
$b = 8$	0.861667	0.931000	0.864000	0.838333	20.002799	11.294232
$b = 16$	0.788000	0.898667	0.803333	0.865667	31.202536	14.353185
$b = 32$	0.753667	0.832000	0.795667	0.810000	45.592056	20.224496
$b = 64$	0.702000	0.713333	0.762667	0.704667	71.032494	26.425313
	$\beta = 0.95$					
$b = 8$	0.707000	0.752333	0.709667	0.655000	20.233897	11.297375
$b = 16$	0.694333	0.783333	0.713667	0.722667	29.028903	11.642453
$b = 32$	0.709333	0.718333	0.745667	0.701000	48.178326	20.229754
$b = 64$	0.620667	0.554000	0.681333	0.559000	71.008642	26.372976
	$\beta = -0.50$					
$b = 8$	0.866333	0.964667	0.867000	0.901000	22.211854	11.294618
$b = 16$	0.837000	0.929667	0.847333	0.920667	28.654514	14.353748
$b = 32$	0.813000	0.864000	0.835333	0.855333	48.452197	17.774360
$b = 64$	0.746000	0.751333	0.780333	0.750667	73.115564	29.632741

Table 6: **Monte Carlo Simulations II: linear regressions**

Sample size=256, nsim=3000, Normal innovations, AR(1)-HET

	equal tail		symmetric		time	
	standard	fast	standard	fast	standard	fast
	$\beta = 0.20$					
$b = 8$	0.962333	0.956000	0.963333	0.936333	22.025650	11.051928
$b = 16$	0.909333	0.918333	0.911000	0.912000	31.058884	14.092693
$b = 32$	0.878667	0.883000	0.880333	0.875333	48.186742	18.006162
$b = 64$	0.796333	0.798333	0.821333	0.809667	71.127459	29.261800
	$\beta = 0.50$					
$b = 8$	0.950667	0.955000	0.951333	0.922333	23.924014	11.048342
$b = 16$	0.898333	0.922333	0.902333	0.912667	28.899060	14.100695
$b = 32$	0.854333	0.870667	0.870333	0.863667	48.158899	18.063143
$b = 64$	0.789667	0.780333	0.813000	0.780000	74.632910	26.688544
	$\beta = 0.80$					
$b = 8$	0.875000	0.925667	0.874000	0.863333	20.239452	11.073107
$b = 16$	0.816333	0.903667	0.831333	0.872667	29.031416	14.131946
$b = 32$	0.793000	0.842000	0.821000	0.826000	48.170007	20.023867
$b = 64$	0.725333	0.730000	0.771667	0.737667	71.087483	29.325195
	$\beta = 0.95$					
$b = 8$	0.678667	0.757667	0.679333	0.665000	23.609314	11.055319
$b = 16$	0.675667	0.790667	0.693000	0.722000	29.289869	14.094036
$b = 32$	0.684000	0.701333	0.727333	0.697000	46.108904	20.005530
$b = 64$	0.615667	0.577333	0.687667	0.571333	71.222939	29.279025
	$\beta = -0.50$					
$b = 8$	0.933333	0.965333	0.931333	0.931333	22.016554	11.026871
$b = 16$	0.888667	0.931000	0.900333	0.917000	31.092572	14.044001
$b = 32$	0.869000	0.874667	0.877333	0.877667	48.137248	20.013409
$b = 64$	0.771000	0.790333	0.785333	0.796000	71.110701	29.304502

Table 7: **Monte Carlo Simulations II: linear regressions**

Sample size=256, nsim=3000, Exponential innovations, AR(1)-SEASON

	equal tail		symmetric		time	
	standard	fast	standard	fast	standard	fast
	$\beta = 0.20$					
$b = 8$	0.998667	0.975667	0.999667	0.933333	21.370292	12.629717
$b = 16$	0.985667	0.945333	0.983000	0.929667	31.009842	12.569247
$b = 32$	0.939667	0.895667	0.936667	0.880667	52.751286	19.197753
$b = 64$	0.848000	0.821333	0.855667	0.821333	76.593835	31.325954
	$\beta = 0.50$					
$b = 8$	0.995333	0.973000	0.997667	0.927333	21.399071	12.615772
$b = 16$	0.968333	0.934667	0.975000	0.918667	31.010740	12.569459
$b = 32$	0.918333	0.888333	0.927667	0.878667	49.253593	22.828511
$b = 64$	0.831333	0.798667	0.842333	0.801333	76.487474	31.334471
	$\beta = 0.80$					
$b = 8$	0.945667	0.925000	0.966667	0.874333	24.462900	9.372431
$b = 16$	0.878667	0.896000	0.917333	0.877333	31.141159	15.432206
$b = 32$	0.854000	0.844000	0.895000	0.843667	49.293170	19.175169
$b = 64$	0.778333	0.748667	0.824667	0.758667	79.499229	31.339434
	$\beta = 0.95$					
$b = 8$	0.733333	0.738667	0.776667	0.691333	24.340965	9.366512
$b = 16$	0.618333	0.776333	0.689667	0.748000	34.619020	12.564055
$b = 32$	0.647000	0.728000	0.744000	0.725333	52.473108	19.165086
$b = 64$	0.623000	0.615000	0.733667	0.619333	79.383994	28.663418
	$\beta = -0.50$					
$b = 8$	0.997000	0.973000	0.995667	0.927000	21.305777	9.370282
$b = 16$	0.970000	0.934000	0.972333	0.923000	33.005734	15.433096
$b = 32$	0.926333	0.903667	0.929667	0.889000	49.257912	19.170534
$b = 64$	0.832667	0.781000	0.855000	0.789000	79.495583	31.335386

Table 8: **Monte Carlo Simulations II: linear regressions**

Sample size=256, nsim=3000, Normal innovations, AR(1)-SEASON

	equal tail		symmetric		time	
	standard	fast	standard	fast	standard	fast
	$\beta = 0.20$					
$b = 8$	0.999667	0.971667	0.999667	0.939000	24.500464	9.632896
$b = 16$	0.971000	0.943333	0.969333	0.936000	34.724897	12.840410
$b = 32$	0.922000	0.897333	0.924000	0.887000	52.374974	22.572527
$b = 64$	0.810333	0.805667	0.825333	0.805667	77.146013	28.814146
	$\beta = 0.50$					
$b = 8$	0.993000	0.965000	0.996000	0.924333	21.578373	12.353445
$b = 16$	0.949000	0.924333	0.964333	0.910667	31.287991	15.159032
$b = 32$	0.903000	0.883000	0.912333	0.869667	52.375243	19.438273
$b = 64$	0.815333	0.786667	0.833000	0.792000	77.149086	31.197714
	$\beta = 0.80$					
$b = 8$	0.919000	0.926667	0.950333	0.885333	24.234948	12.366362
$b = 16$	0.830667	0.901333	0.883667	0.888000	31.449808	15.163945
$b = 32$	0.822333	0.843667	0.862333	0.831667	52.320000	19.420533
$b = 64$	0.761667	0.751333	0.825667	0.763333	77.160251	31.172236
	$\beta = 0.95$					
$b = 8$	0.668667	0.723667	0.717000	0.688667	24.097808	12.374219
$b = 16$	0.577000	0.772667	0.650667	0.741000	31.667165	15.163683
$b = 32$	0.608000	0.725667	0.707333	0.719667	52.107257	19.424637
$b = 64$	0.626333	0.614667	0.743333	0.621000	80.722773	31.182769
	$\beta = -0.50$					
$b = 8$	0.994000	0.971000	0.994667	0.933333	21.515967	12.361406
$b = 16$	0.960667	0.939000	0.957667	0.927667	31.292004	15.168712
$b = 32$	0.913667	0.896333	0.912000	0.882000	52.362998	19.445435
$b = 64$	0.813667	0.803000	0.827000	0.803667	77.181672	31.210887

Table 9: **Monte Carlo Simulations II: linear regressions**

Sample size=256, nsim=3000, Exponential innovations, MA(1)-HOMO

	equal tail		symmetric		time	
	standard	fast	standard	fast	standard	fast
	$\rho = 0.20$					
	$\beta = 0.20$					
$b = 8$	0.999000	0.936667	0.999333	0.925000	22.135509	11.235934
$b = 16$	0.979667	0.914333	0.986000	0.907333	28.597782	14.248934
$b = 32$	0.936333	0.876333	0.935333	0.877333	45.149513	18.163444
$b = 64$	0.817333	0.789333	0.834667	0.797000	72.063531	29.574989
	$\beta = 0.50$					
$b = 8$	0.998333	0.943667	0.999333	0.926333	22.076793	11.248199
$b = 16$	0.978333	0.922667	0.984333	0.914667	28.586218	14.312845
$b = 32$	0.926333	0.885000	0.935667	0.882000	45.206295	20.181945
$b = 64$	0.833000	0.786333	0.848000	0.790667	72.184503	26.496267
	$\beta = 0.80$					
$b = 8$	1.000000	0.951333	1.000000	0.925667	22.116943	11.241444
$b = 16$	0.975667	0.918667	0.980667	0.911333	31.364697	11.683778
$b = 32$	0.938000	0.868000	0.944333	0.869667	48.879892	17.810614
$b = 64$	0.846000	0.777000	0.856667	0.785333	72.245084	26.370749
	$\beta = 0.95$					
$b = 8$	0.999667	0.958333	0.999667	0.928000	19.818973	11.247910
$b = 16$	0.985333	0.936000	0.992000	0.915333	31.423614	11.677717
$b = 32$	0.934333	0.893333	0.942667	0.882667	45.089990	20.172078
$b = 64$	0.835000	0.780000	0.863000	0.769667	72.206920	29.587364
	$\beta = -0.50$					
$b = 8$	0.999667	0.957333	0.999667	0.939000	19.848966	11.239523
$b = 16$	0.981667	0.918000	0.978667	0.915000	31.422583	11.674821
$b = 32$	0.926333	0.876333	0.934000	0.878000	48.904007	17.843676
$b = 64$	0.832333	0.770000	0.848000	0.781667	72.163126	29.627338

Table 10: **Monte Carlo Simulations II: linear regressions**

Sample size=256, nsim=3000, Normal innovations, MA(1)-HOMO

	equal tail		symmetric		time	
	standard	fast	standard	fast	standard	fast
	$\beta = 0.20$					
$b = 8$	0.998333	0.933333	0.999333	0.930667	20.015759	9.075204
$b = 16$	0.960667	0.913333	0.966333	0.917000	31.168467	12.003598
$b = 32$	0.907333	0.875333	0.918333	0.880000	48.615643	18.132335
$b = 64$	0.814000	0.798333	0.829667	0.807667	70.121052	29.304408
	$\beta = 0.50$					
$b = 8$	0.997667	0.938667	0.998667	0.923333	20.060068	9.073609
$b = 16$	0.963000	0.920333	0.969000	0.914000	31.172958	14.001012
$b = 32$	0.904333	0.876000	0.912000	0.875667	48.614151	18.097987
$b = 64$	0.820667	0.789000	0.833333	0.794667	70.069833	26.693628
	$\beta = 0.80$					
$b = 8$	0.997333	0.946000	0.998000	0.924000	20.123753	9.088840
$b = 16$	0.961333	0.922333	0.970667	0.909667	31.091003	12.000105
$b = 32$	0.907333	0.872333	0.914667	0.870333	48.585880	18.102141
$b = 64$	0.806333	0.792000	0.834000	0.792667	70.073567	29.307689
	$\beta = 0.95$					
$b = 8$	0.997667	0.957333	0.999667	0.935333	20.180027	12.926023
$b = 16$	0.962333	0.944333	0.971333	0.922667	28.894624	14.005363
$b = 32$	0.909333	0.884667	0.921667	0.873000	48.595144	18.115928
$b = 64$	0.825333	0.803333	0.837667	0.792667	70.223251	29.311657
	$\beta = -0.50$					
$b = 8$	0.997333	0.936000	0.999000	0.929667	20.018498	12.922229
$b = 16$	0.959667	0.917000	0.968000	0.913000	28.839317	14.004709
$b = 32$	0.911667	0.888333	0.913667	0.885667	48.597131	18.103728
$b = 64$	0.813333	0.790667	0.821000	0.806000	70.109295	29.302486

Table 11: **Monte Carlo Simulations III: ARCH(1) process**

Sample size=256, nsim=3000

	equal tail		symmetric		time	
	standard	fast	standard	fast	standard	fast
	$\beta = 0.20$					
b=4	0.395000	0.818667	0.478000	0.794667	106.158012	9.441458
b=8	0.323000	0.822333	0.416667	0.800333	124.008540	7.191408
b=16	0.250333	0.782667	0.304000	0.778667	160.285605	11.606134
b=32	0.183333	0.770000	0.209333	0.778000	211.590072	13.494634
b=64	0.125000	0.696333	0.130667	0.719333	283.147487	16.326398
	$\beta = 0.50$					
b=4	0.399000	0.920667	0.451333	0.897000	101.461675	6.091064
b=8	0.331000	0.906000	0.384000	0.891667	109.097385	9.211740
b=16	0.269000	0.894333	0.293000	0.893333	132.371523	7.891285
b=32	0.225000	0.863000	0.225000	0.873000	164.017802	13.943144
b=64	0.145000	0.807667	0.149333	0.828000	215.731918	13.242809
	$\beta = 0.80$					
b=4	0.407667	0.950667	0.466333	0.933667	103.554542	8.105092
b=8	0.348000	0.938667	0.373667	0.930000	101.444544	9.415940
b=16	0.303000	0.928000	0.308000	0.933667	116.284817	7.737518
b=32	0.244000	0.918000	0.248667	0.925333	145.055172	12.128273
b=64	0.156667	0.875333	0.172333	0.897000	197.286815	13.052062
	$\beta = 0.95$					
b=4	0.418000	0.964667	0.459333	0.951667	100.516946	8.152153
b=8	0.350667	0.955000	0.380667	0.948333	99.042962	9.450073
b=16	0.289333	0.945333	0.302667	0.950333	111.131355	7.671819
b=32	0.231333	0.926333	0.253000	0.939333	140.831393	9.827453
b=64	0.135667	0.876333	0.159667	0.899333	190.037159	13.012083

Table 12: **Monte Carlo Simulations IV: unit root process**

Sample size=256, nsim=3000, Normal innovations

	equal tail		symmetric		time	
	standard	fast	standard	fast	standard	fast
	$\rho = 1$, OLS estimator					
$b = 4$	0.936667	0.895333	0.926000	0.870667	1.07750	1.05419
$b = 8$	0.982000	0.913333	0.970667	0.892333	4.79926	1.16436
$b = 16$	0.991000	0.915333	0.981333	0.904333	1.42551	1.36613
$b = 32$	0.998000	0.897000	0.993667	0.903667	4.19037	4.27615
$b = 64$	0.999667	0.855667	0.999333	0.880333	2.38357	5.73919
	$\rho = 1$, t-statistics					
$b = 4$	0.672667	0.891667	0.699333	0.887000	1.36296	4.68180
$b = 8$	0.733000	0.916000	0.759333	0.912667	5.08875	4.52817
$b = 16$	0.798000	0.912333	0.826667	0.916333	6.54767	4.24121
$b = 32$	0.826333	0.890000	0.869333	0.914333	4.43174	2.24512
$b = 64$	0.821333	0.838333	0.921333	0.886667	8.03645	5.00263

Table 13: **Monte Carlo Simulations IV: unit root process**

Sample size=256, nsim=3000, Martingale difference

	equal tail		symmetric		time	
	standard	fast	standard	fast	standard	fast
	$\rho = 1$, OLS estimator					
$b = 4$	0.963000	0.940000	0.948333	0.913667	1.10152	1.07395
$b = 8$	0.983333	0.954333	0.967333	0.925000	1.22514	1.18560
$b = 16$	0.994667	0.942333	0.987667	0.930000	4.55796	1.39896
$b = 32$	0.999333	0.927000	0.996667	0.923000	4.16829	4.24014
$b = 64$	1.000000	0.871000	1.000000	0.890333	2.39837	5.72064
	$\rho = 1$, t-statistics					
$b = 4$	0.688333	0.919667	0.6750000	0.915667	4.61876	1.34423
$b = 8$	0.732333	0.945000	0.738667	0.944000	5.04936	1.48389
$b = 16$	0.791667	0.929667	0.814667	0.941333	3.48277	4.22785
$b = 32$	0.824000	0.919333	0.880667	0.938667	7.54265	2.25628
$b = 64$	0.815333	0.845333	0.909667	0.893	8.01383	5.00687

REFERENCES :

- Andrews, D. (2001), "Higher-Order Improvements of a Computationally Attractive k-step Bootstrap for Extremum Estimators", forthcoming in *Econometrica*.

- Bertail, P., D. Politis and J. Romano (1999): "On Subsampling Estimation with Unknown Rate of Convergence", *Journal of the American Statistical Association*, **94**, 446, 569-579.

- Davidson, R. and J. McKinnon (1999), "Bootstrap Testing in Nonlinear Models", *International Economic Review*, **40**, 487–508.

- Dhaene, G., C. Gouriéroux and O. Scaillet (1998), "Indirect Encompassing and Instrumental Models", *Econometrica*, **66**, 673–688.

- Goncalves,S. and H. White (2000), "Maximum Likelihood and the Bootstrap for Nonlinear Dynamic models", DP UCSD.

- Gouriéroux, C. and A. Monfort (1995): *Statistics and Econometric Models*, Cambrdige University press, Cambridge.

- Gouriéroux, C., A. Monfort and E. Renault (1993): "Indirect Inference", *Journal of Applied Econometrics*, **8**, 85–118.

- Gouriéroux, C., A. Monfort and A. Trognon (1984): "Pseudo Maximum Likelihood Theory", *Econometrica*, **52**, 681–700.

- Heagerty, P. and T. Lumley (2000): "Window Subsampling on Estimating Functions with Application to Regression Models", *Journal of the American Statistical Association*, **95**, 449, 197-211.

- Hu, F. and J. Kalbfleisch (2000), "The Estimating Function Bootstrap", *Canadian Journal of Statistics*, **28**, 449–499.

- Politis, D., and J. Romano (1992): "A General Theory for Large Sample Confidence Regions based on Subsamples under Minimal Assumptions", Technical Report 399, Dpt of Statistics, Stanford University.

- Politis, D., and J. Romano (1994): "Large Sample Confidence Regions based on Subsamples under Minimal Assumptions", *Annals of Statitics*, **22**, 2031-2050.

- Politis, D., Romano, J. and M. Wolf (1999): *Subsampling*, Springer-Verlag, New-York.

- Romano, J. and M. Wolf (2001): "Subsampling Intervals in Autoregressive Models with Linear Time Trend", *Econometrica*, **69**, 1283-1314.

- Shao, J. and and D. Tu (1995), *The Jackknife and Bootstrap*, Springer-Verlag, New York.

- White, H. (1982), "Maximum Likelihood Estimation of Misspecified Models", *Econometrica*, **50**, 1–26.