Technical details concerning implementation of the method in Craig (2007a) for computing orthant probabilities for multivariate normal auto-regressive sequences

Peter Craig

July 15th 2007

Abstract

This document is an expanded version of section 3.2 of Craig (2007a) intended to make clear certain details of the algorithm implemented in Craig (2007b).

1 Introduction

We wish to compute $\mathrm{P}[X \geq 0] = \mathrm{P}[X_1 \geq 0, \ldots, X_p \geq 0]$ when $X_1, \ldots, X_p$ are an auto-regressive sequence. In order to make the calculation easier to structure, simplify the subsequent presentation and facilitate analysis of accuracy, we will restrict to the case of unit variances and express the calculation in terms of $W = X - \mu$. Then, writing $\rho_i = \mathrm{Corr}[X_i,\, X_{i+1}]$ and $\sigma_i^2 = 1 - \rho_i^2$, the sequence $W_1, \ldots, W_p$ is Markov and $W_{i+1} \,|\, W_i \sim \mathrm{N}(\rho_i W_i, \sigma_i^2)$.

Taking $\psi_1(w_1) = p(w_1) = \phi(w_1)$, we sequentially compute approximations to the functions $\psi_2, \ldots, \psi_p$ where

$$\begin{aligned}
\psi_{n+1}(w_{n+1}) &= \int_{-\mu_n}^{\infty} \cdots \int_{-\mu_1}^{\infty} p(w_1, \ldots, w_{n+1})\, dw_1 \ldots dw_n \\
&= \int_{-\mu_n}^{\infty} p(w_{n+1} \,|\, w_n) \int_{-\mu_{n-1}}^{\infty} \cdots \int_{-\mu_1}^{\infty} p(w_1, \ldots, w_n)\, dw_1 \ldots dw_{n-1} dw_n \\
&= \frac{1}{\sigma_n} \int_{-\mu_n}^{\infty} \phi\big((w_{n+1} - \rho_n w_n)/\sigma_n\big) \psi_n(w_n)\, dw_n \qquad (1) \\
&= \frac{1}{|\rho_n|\sigma_n} \int_{-|\rho_n|\mu_n}^{\infty} \phi\big((w_{n+1} - \mathrm{sign}(\rho_n) u)/\sigma_n\big) \psi_n(u/|\rho_n|)\, du \qquad (2)
\end{aligned}$$

so that $\mathrm{P}[X \geq 0]$ is obtained by integrating ψ_p from $-\mu_p$ to infinity.

In principle, calculation of (1) appears to require the numerical evaluation of a one-dimensional integral for each value of w_{n+1}. However, (2) is in the form of a convolution and may be amenable to calculation using the fast Fourier transform (FFT). The FFT approach is efficient except when σ_n is small; moreover there is an efficient linear filtering algorithm to handle small σ_n. When ρ_n is zero (or effectively so), (1) reduces to integrating ψ_n and restarting the process with ψ_{n+1} proportional to $\phi(w_{n+1})$.

For numerical calculations, we replace the lower limit of integration in (1) by $L_n = \max(-U, -\mu_n)$ and the upper limit by U. In doing so, we lose some probability from the final estimate of $\mathrm{P}[X \geq 0]$ but the loss is less than $\sum_i \mathrm{P}[|W_i| > U] = 2p\Phi(-U)$ where Φ is the standard normal distribution function. When linear filtering, we also truncate ϕ to zero outside the interval $[-U, U]$ and again lose less than $2p\Phi(-U)$. In practice, it should be easy to choose U so that the total loss is negligible; for double precision, $U = 8$ works well.

The strategy is to approximate ψ_n on a grid (sequence of equally spaced values) covering $[L_n, U)$ with a spacing Δ_n which we try to keep close to a fixed value Δ which is our basic control on the accuracy of approximation.

2 Using the fast Fourier transform

The Fourier transform of ψ_{n+1} is

$$\begin{aligned}\Psi_{n+1}(t) &= \int e^{itx}\psi_{n+1}(x)\,dx = \frac{1}{\sigma_n}\int e^{itv}\phi(v/\sigma_n)\,dv\int_{-\mu_n}^{\infty} e^{it\rho_n y}\psi_n(y)\,dy \\ &= e^{-\sigma_n^2 t^2/2}\Psi_n^+(\rho_n t)\end{aligned}$$

where Ψ_n^+ is the Fourier transform of the truncation of ψ_n to zero below $-\mu_n$. Thus we can compute ψ_{n+1} by finding the Fourier transform of the truncated version of ψ_n, stretching it by factor $1/\rho_n$, multiplying by $e^{-\sigma_n^2 t^2/2}$ and finally inverting the Fourier transform.

Numerically, we can approximate this operation using the fractional FFT of Bailey and Swarztrauber (1991) to compute Fourier transforms on suitable grids of values. The fractional FFT differs from the conventional FFT in that the output fundamental frequency (spacing of the grid of values of t) may be freely chosen rather than being determined by the spacing in the x-grid and the number of grid points. As described in detail in section 13.9 of Press et al. (1993), we may increase the accuracy of the calculation of a Fourier integral for a fixed grid spacing by using Filon (1928)'s method for Fourier integrals of continuous functions. The Filonized fractional FFT (FfFFT) is most efficient when the grid length is a power of 2.

The main numerical difficulty is that the Fourier transform of a truncated smooth function decays to zero very slowly (asymptotically at rate $1/t$). Hence the decay to zero of $e^{-\sigma_n^2 t^2/2}$ is the limiting control on the range over which we need to approximate Ψ_{n+1} in order to compute ψ_{n+1}. In effect, this also determines the range over which we must evaluate Ψ_n^+. For $\sigma_n = 1$, approximating Ψ_{n+1} on $[-U, U]$ should suffice but in general the range needs to be proportional to $1/\sigma_n$ which means we must either use a larger grid or lose precision by increasing the spacing; we do the former as it is easy to use the FfFFT to approximate Ψ_n^+ on a grid whose length is an integer multiple g_n of the length of the grid holding the truncation of ψ_n. In principle, g_n could become arbitrarily large. However, the difficulty only arises when when σ_n is near to zero. In such situations, the convolution may be computed directly by linear filtering of the function ψ_n evaluated on a grid.

For efficient use of the FfFFT, we work with grids whose lengths are integer multiples of some basic grid length G which we restrict to be a power of 2 so that our target grid-spacing is $\Delta = 2U/G$. Ψ_n^+ is computed on a grid of length $g_n G$ where we restrict g_n to be a power of 2 and we choose $g_n > 1/(2\sigma_n)$ to ensure that the resulting grid spacing for Ψ_{n+1} is less than 2Δ. We evaluate Ψ_n^+ on the grid starting at $-\rho_n U/\sigma_n$ with spacing $\rho_n\Delta/(\sigma_n g_n)$, obtain Ψ_{n+1} and use the inverse FfFFT to obtain ψ_{n+1} on an grid of length $g_n G$ starting at L_{n+1} with spacing Δ_{n+1}. Finally we throw away all except the first G values on this grid; in principle, one may save effort by using special FFT algorithms to compute only the first G values but the author's experience concurs with the literature which suggests that in practice there is no significant gain in efficiency and often a loss unless g_n is large (Frigo and Johnson (2006b) suggest order of magnitude 100).

3 Using linear filtering

Linear filtering is an efficient way to approximate (2) when σ_n is small since the integral will effectively only involve a small range of u for each x. Suppose we have already calculated an approximation to $\psi_n(x)$ so that $\hat{\psi}_{n,k} \approx \psi_n(x_{n,k})$ for $0 \le k < G$ where

$x_{n,k} = L_n + k\Delta_n$. Linear filtering means that we plan (for most values of k) to compute $\hat{\psi}_{n+1,k} = \sum_{m=0}^{M_n} c_{n,m}\hat{\psi}_{n,k_0+k+m}$ where k_0 (depending on n) is an offset between the grids; for negative ρ_n, replace $\hat{\psi}_{n,k_0+k+m}$ by $\hat{\psi}_{n,k_0-k+m}$. For this to be possible, the grids used for x and u must have the same spacing and so $\Delta_{n+1} = |\rho_n|\Delta_n$.

We could easily obtain coefficients $c_{n,m}$ by using an elementary quadrature formula such as Simpson's rule to approximate (2). However, because σ_n is small, the integrand in (2) is not really smooth enough for such a simple approach. The smoothness of the underlying multivariate normal probability density function means that ψ_n should be smooth apart from the step at $x = -\mu_n$, a view supported by some numerical experiments. This suggests using an approach to computing the convolution which takes advantage of the smoothness of ψ_n while allowing for the lack of smoothness in $\phi(./\sigma_n)$.

We replace ψ_n in (2) by a piecewise cubic interpolant. On each interval $(x_{n,k}, x_{n,k+1})$, approximate $\psi_n(x)$ by the cubic interpolant of $\hat{\psi}_n$ for the four nearest grid points. Hence, the contribution of the interval to $\hat{\psi}_{n+1,j}$ may be obtained as a linear combination of the values of $\hat{\psi}_n$ at those points.

For now, assume $\rho_n > 0$. On an interior $(0 < k < G-2)$ interval $[x_{n,k}, x_{n,k+1}]$, the interpolant of the grid of values of $\hat{\psi}_n$ is $\hat{\psi}_n(x) = \sum_{i=-1}^{2} \hat{\psi}_{n,k+i}\, l_i(t)$ where $t = (x - x_{n,k})/\Delta_n$ and $l_{-1}, \ldots, l_2$ are the Lagrange polynomials for cubic interpolation of a function evaluated at abscissae -1, 0, 1 and 2. Substituting into (1), the contribution to $\hat{\psi}_{n+1,j}$ from an interior interval is $(\Delta_n/\sigma_n)\sum_{i=-1}^{2} \zeta_{n,j-k,i}\hat{\psi}_{n,k+i}$ where

$$\begin{aligned}\zeta_{n,j-k,i} &= \frac{1}{\Delta_n}\int_{x_{n,k}}^{x_{n,k+1}} \phi\left(\frac{x_{n+1,j} - \rho_n y}{\sigma_n}\right) l_i\left(\frac{y - x_{n,k}}{\Delta_n}\right) dy \\ &= \int_0^1 \phi\left(\frac{t - (j-k) - \tilde{\mu}_n}{\tilde{\sigma}_n}\right) l_i(t)\, dt \end{aligned} \qquad (3)$$

with $\tilde{\sigma}_n = \sigma_n/\Delta_{n+1}$ and $\tilde{\mu}_n = (L_{n+1} - \rho_n L_n)/\Delta_{n+1}$. Truncating ϕ outside $[-U, U]$ means that $\zeta_{n,m,i} = 0$ unless $m + \tilde{\mu}_n \in (-U\tilde{\sigma}_n, 1 + U\tilde{\sigma}_n)$.

The contributions to $(\sigma_n/\Delta_n)\hat{\psi}_{n+1,j}$ from the edge-intervals, $k = 0$ and $k = G-2$, are respectively $\sum_{i=-1}^{2} \zeta^*_{n,j,i}\hat{\psi}_{n,i+1}$ and $\sum_{i=-1}^{2} \zeta^{**}_{n,j-(G-2),i}\hat{\psi}_{n,G-3+i}$ where $\zeta^*_{n,m,i}$ and $\zeta^{**}_{n,m,i}$ are obtained by substituting respectively $l_i(t-1)$ and $l_i(t+1)$ for $l_i(t)$ in (3).

Consider now all the contributions made by $(\Delta_n/\sigma_n)\hat{\psi}_{n,k}$ to $\hat{\psi}_{n+1,k+m}$. For $3 \le k \le G-4$, there is a contribution of $\zeta_{n,m+i,i}$ for $-1 \le i \le 2$. i.e. a total of $\xi_{n,m} = \sum_{i=-1}^{2} \zeta_{n,m+i,i}$. At the edges, the contributions change. For $k < 3$ the upper limit of the sum is $k-1$ and for $k > G-4$, the lower limit is $k-(G-3)$. Additional contributions at the edges are $\zeta^*_{n,k+m,k-1}$ for $k \le 3$ and $\zeta^{**}_{n,m-2+k-(G-4),k-(G-3)}$ for $k \ge G-4$.

It is possible to evaluate $\zeta_{n,m,i}$, $\zeta^*_{n,m,i}$, $\zeta^{**}_{n,m,i}$ and $\xi_{n,m}$ as expressions involving $\tilde{\mu}_n$, $\tilde{\sigma}_n$, m and the functions ϕ and Φ. In practice, one may use a package for symbolic mathematical computation such as Maple (Monagan et al., 2005) which can then also generate code in a suitable programming language. Dropping the subscript n, Maple showed that

$$\begin{aligned}\xi_0 = \frac{1}{6}\sum_{j=-2}^{2}(-1)^j\binom{4}{j+2}\Bigg[&(\tilde{\mu}+j)\tilde{\sigma}\big((\tilde{\mu}+j)^2 + 3\tilde{\sigma}^2 - 1\big)\Phi\left(\frac{\tilde{\mu}+j}{\tilde{\sigma}}\right) \\ &+ \tilde{\sigma}^2\big((\tilde{\mu}+j)^2 + 2\tilde{\sigma}^2 - 1\big)\phi\left(\frac{\tilde{\mu}+j}{\tilde{\sigma}}\right)\Bigg]\end{aligned} \qquad (4)$$

and ξ_m is easily obtained by substituting $\tilde{\mu} + m$ for $\tilde{\mu}$. For fixed $\tilde{\mu}$ and $\tilde{\sigma}$ and varying m, there are savings to be made by pre-computing or saving of Φ and ϕ between evaluations.

The one remaining problem is that and expression such as (4) is fine for direct calculation except when $\tilde{\sigma}$ is large when it suffers from high relative truncation errors. The solution is to use a series approximation for larger values of $\tilde{\sigma}$. Since the range of values of interest for $\tilde{\mu}$ is proportional to $\tilde{\sigma}$, we must first write $\tilde{\mu} = c\tilde{\sigma}$ and then find a series approximation. For double-precision, a 14th-order series in $1/\tilde{\sigma}$ (computed by Maple) gave excellent results for $\tilde{\sigma} > 4$ and direct evaluation of (4) achieved close to machine accuracy for smaller $\tilde{\sigma}$.

The details of linear filtering above assume $\rho_n > 0$. For negative ρ_n, the easy way to handle it is to exploit the fact that $\psi_{n+1}(-x)$ may be computed using the right-side of (2), omitting "sign(ρ_n)", simply by changing the sign of ρ_n. So we compute on the grid $-(L_{n+1} + (G-1)\Delta_{n+1}) + j\Delta_{n+1}$ having changed the sign of ρ_n and reverse the order of the computed values at the end; in effect we temporarily replace L_{n+1} by $-(L_{n+1} + (G-1)\Delta_{n+1})$ as well as temporarily changing the sign of ρ_n.

4 Overall strategy

We need to decide when to use the FFT and when use linear filtering. Computational effort for the latter should essentially be proportional to $2GU\tilde{\sigma}_n = G^2\Delta\sigma_n/\Delta_{n+1} \approx G^2\sigma_n$ assuming that $\Delta_{n+1} \approx \Delta$. The fractional FFT computes a number of FFTs each of length $N = g_nG \approx G/\sigma_n$ so that the computation time should essentially be proportional to $N\log_2 N$. The ratio of FFT time to linear filtering time is therefore approximately proportional to $S = (\log_2 G - \log_2 \sigma_n)/(G\sigma_n^2)$ and the decision as to which method to use should be based on the magnitude of this quantity. For our Pentium 4 hardware and using the FFTW library (Frigo and Johnson, 2006a), numerical experiments suggested that $S = .02$ is a suitable point for making the transition from FFT to linear filtering.

The algorithm has 3 control parameters which affect overall accuracy and efficiency: U, G and the threshold value S. The basic value of Δ is determined from U and G. To determine Δ_n, recall that, when linear filtering is used to compute ψ_{n+1}, Δ_{n+1} determines Δ_n whereas the FFT allows a free choice of Δ_n. Thus a simple strategy is to start with $\Delta_p = \Delta$ and to work backwards, setting $\Delta_n = \Delta$ whenever the computation of ψ_{n+1} uses the FFT. If at some point this leads to $\Delta_n > 2\Delta$, it may be necessary to use the FFT for some step where it would normally be more efficient to use linear filtering.

References

Bailey, D. H. and Swarztrauber, P. N. (1991) The fractional Fourier transform and applications. *SIAM Review*, **33**, 389–404.

Craig, P. (2007a) A new reconstruction of multivariate normal orthant probabilities.

——— (2007b) *orthants: Multivariate normal orthant probabilities.* URL `http://www.maths.dur.ac.uk/~dma0psc/orthants/`.

Filon, L. (1928) On a quadrature formula for trigonometric integrals. *Proceedings of the Royal Society of Edinburgh*, **49**, 38–47.

Frigo, M. and Johnson, S. G. (2006a) *FFTW Home Page.* URL `http://www.fftw.org/`.

——— (2006b) *Pruned FFTs with FFTW.* URL `http://www.fftw.org/pruned.html`.

Monagan, M., Geddes, K., Heal, K., Labahn, G., Vorkoetter, S., McCarron, J. and DeMarco, P. (2005) *Maple 10 Introductory Programming Guide*.

Press, W. H., Flannery, B. P., Teukolsky, S. A. and Vetterling, W. T. (1993) *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge: Cambridge University Press, 2nd edition.