Maximum Expected Utility via MCMC

Eric Jacquier[a]*, Michael Johannes[b], Nicholas Polson[c]

[a] *HEC Montreal and Sloan MIT, 100 Main Street, Cambridge MA 02138,*

[b] *Graduate School of Business, Columbia University, 3022 Broadway, NY, NY 10027*

[c] *Chicago Booth School of Business, 5807 South Woodlawn, Chicago, IL 60637.*

First draft: January 2006

This draft: December 2010

Abstract

In this paper we provide a new simulation-based approach to maximum expected utility (MEU) portfolio allocation problems. MEU requires computation of expected utility and its optimization over the decision variable. In portfolio problems, the expected utility is generally not analytically available. Traditional methods resort to gradient-based estimates of expected utility with its ensuing derivatives. This leads to computational inefficiencies which are particularly acute in portfolio problems with parameter uncertainty (a.k.a. estimation risk). Our simulation-based method avoids the calculation of derivative and also allows for functional optimization. The algorithm combines Markov Chain Monte Carlo (MCMC) with the insights of simulated annealing and evolutionary Monte Carlo. It can exploit conjugate utility functions and latent variables in the relevant predictive density for efficient simulation. We also show how slice sampling naturally allows for constraints in the portfolio weights. We illustrate our methodology with a portfolio problem with estimation risk and CARA utility.

JEL Classification: C1, C11, C15, G1

Key Words: MCMC, expected utility, portfolio choice, asset allocation, optimization, simulated annealing, evolutionary Monte Carlo, Bayesian learning, slice sampling.

[*] Jacquier is the corresponding author at jacquier@mit.edu. This paper has benefited from comments on a related paper by Greg Bauer, Ron Gallant, Lars Hansen, George Tauchen, and the participants of the Montreal Financial Econometrics conference.

1 Introduction

Maximum expected utility (MEU) problems require both the computation of expected utility and its optimization over the decision variable. In this paper, we provide a simulation-based approach to MEU portfolio problems incorporating parameter uncertainty. Under realistic distributional and utility function specifications, expected utility often has no analytical solution. Therefore, integration, via simulation together with optimization are simultaneously required. Traditional methods, such as gradient-based optimization with embedded numerical integration, can quickly become intractable. Our work builds upon Zellner and Chetty (1965) and many others who incorporate estimation risk into optimal portfolio design. This literature extends the classic results in Merton (1969) to incorporate parameter uncertainty. See e.g., Kandel and Stambaugh (1996), Barberis (2000), Brandt et al (2005), and Jacquier, Kane and Marcus (2005) for portfolio design with estimation uncertainty, Brandt (2010) and Jacquier and Polson (2011) for recent surveys.[1]

Consider the generic MEU problem of an agent solving $d^{\star} = \text{argmax}_d \mathbb{E}_X \left[U(X,d) \right]$ where $U(X,d)$ represents the agent's utility given a decision d, and X is a latent variable related to the stochastic uncertainty. The expectation $\mathbb{E}_X$ is taken with respect to the distribution $p(X)$, namely the predictive density of X after marginalizing out other parameters and state variables. In the portfolio problem, X is the vector of future returns for the investment opportunity set. The predictive distribution marginalizes over parameters and state variables such as the means and variances of portfolio returns.

Our approach will also solve functional decision problems where the optimal decision $d^{\star}(Y)$ also depends on a set of known conditioning variables Y. Here one needs the conditional predictive distribution $p(X|Y)$ where Y is a set of latent variables such as the predicted future volatilities or current state variables. We will show how to deal with the common situations where distributions are of the form $p(X|Y) = \int p(X|V)p(V|Y)dV$ with V being a set of latent variables.

MCMC methods have been tremendously useful in simulating from the distribution of the source of uncertainty, $p(X)$. However, comparatively little research has been done on their potential contribution to the usual optimization problems encountered in economics and finance. An approach similar to that presented here has been used with success for the maximization of complex likelihood

[1] See also Mueller (2000), Brockwell and Kadane (2003), Moreno-Diaz et al (2003), and Mueller et al. (2004) for related Bayesian decision-making problems.

functions by Jacquier, Johannes, Polson (2007) and Boivin and Giannoni (2006), but it has not been adapted to expected utility. Our algorithm combines MCMC simulation with the insights of simulated annealing and evolutionary Monte Carlo to produce the optimal decision d^* without having to compute a simulated expected utility and its derivatives or implement a gradient method on that simulated function.

The rest of the paper is as follows. Section 2 first describes the basic simulation-based MCMC-MEU algorithm. It then extends the framework to functional optimization. Section 3 discusses specific issues and their solution, such as shows how to address the minimization of loss functions, and exploit conjugate utilities and latent variables for efficient simulation, and use slice sampling to incorporate constraints into the portfolio optimization. Section 4 illustrate the approach with an asset allocation example. Finally, Section 5 concludes with directions for future research.

2 Maximum Expected Utility by MCMC

2.1 Algorithm

To find $d^\star = \text{argmax}_d \mathbb{E}_X \left[U(X,d) \right]$ in a standard Maximum Expected Utility (MEU) problem, we define a joint distribution, denoted by $\pi_J(\widetilde{X}^J, d)$, on both the decision rule d, and J replications of the source of uncertainty X where $\widetilde{X}^J = (X_1, \ldots, X_J)$. This pseudo-distribution effectively randomizes the decision rule d, and will be shown to collapse on the optimal decision rule $d^\star$. This is purely a computational tool, consistent with decision theory for non-randomized decision rules, see Berger (1985), Chamberlain (2000), and Zellner (2002) for work on Bayesian decision theory. The variability in the draws of the decision variable d in the algorithm is purely of a computational nature, and represents in no way an econometric or economic uncertainty for the agent. Specifically, the joint density of the decision and states is defined as

$$\pi_J(\widetilde{X}^J, d) \propto \prod_{j=1}^{J} U(X^j, d) p(X^j) \mu(d), \tag{1}$$

with respect to a measure $\mu(d)$, typically uniform, a computational device to enforce the necessary regularity conditions.

The algorithm will simultaneously perform the expected utility calculation and find the optimal decision. Pure simulated annealing *powers up* a function to be maximized in order to find its optimum, see e.g., Pincus (1968), Van Laarhoven and Aarts (1974), and Kirkpatrick et al. (1983). This however requires the function to be analytically available. We therefore incorporate further insights from evolutionary Monte Carlo to dispense with this requirement since the expected utility $EU(d) = \int U\left(d, X\right) p\left(X\right) dX$ is most always not analytically available in portfolio problems. Specifically, we rely on the following key result from evolutionary Monte Carlo: the marginal distribution $\pi_J\left(d\right) = \int \pi_J \left(d, \widetilde{X}^J\right) d\widetilde{X}^J$ collapses on $d^\star$ as $J \to \infty$, see Liang and Wong (2001) for evolutionary Monte Carlo

This distributional framework allows for restrictions on the portfolio weights d and the flexible modeling of the predictive density of returns X. The main regularity condition is that the joint density $\pi_J(\widetilde{X}^J, d)$ needs to be well-defined, this requires that $U(X, d) \geq 0$ and the function $U(X, d)p(X)$ to be integrable in both argument with respect to μ. Many utility functions in financial economics, e.g., power and exponential, take values on the whole real line and hence are not strictly positive. We show how to address this issue in the next section.

An important distribution is the marginal density on the decision variable

$$\pi_J(d) = \left\{ \int \prod_{j=1}^{J} U(X^j, d) p(X^j) d\widetilde{X}^J \right\} \mu(d),$$

obtained by integrating out $\widetilde{X}^J$ from (**??**). Now note that $U(d)^J = \int \prod_{j=1}^J U(X^j, d)p(X^j)d\widetilde{X}^J$. This leads to the following distribution

$$\pi_J(d) \propto e^{J \ln \mathbb{E}(U(\theta, d))} \mu(d).$$

We can then let $J \to \infty$ and observe that $\pi_J(d) \to \delta_{d^\star}$, a Dirac measure on the optimum decision $d^* = \arg\max E_X\left[U(X, d)\right]$. Heuristically, this is the usual Laplace approximation-style argument, see Tierney et al (1989). Notice also that the measure $\mu(d)$ is eliminated in this asymptotic analysis. We derive below an asymptotically normal limiting result in J under extra suitable regularity conditions that provides a diagnostic for selecting a large enough J in practice. Relatively low values of J will suffice due to the exponentiation of the objective function. In our extension to the case of functional optimization, where one seeks the optimal function $d^*(Y)$ for a state variable Y of interest we will see that we can easily obtain the functional estimate for practical values of J.

We use MCMC methods to find draws $\{\widetilde{X}^{J,(g)}, d^{(g)}\}_{g=1}^{G}$ from the joint distribution $\pi_J(\widetilde{X}^J, d)$, see Johannes and Polson (2010) for a survey. This also provides draws $d^{(g)}$ of the marginal distribution $\pi_J(d)$, and in turn the optimal decision $d^\star$ from the simple MC estimate $\frac{1}{G}\sum_{g=1}^{G} d^{(g)}$. The joint distribution $\pi_J(\widetilde{X}^J, d)$ is characterized by the set of $J+1$ complete conditionals:

$$\begin{aligned} X^j|d &\sim U\left(X^j, d\right) p\left(X^j\right) \text{ for } \quad j = 1, \ldots, J & (2) \\ d|X^1, \ldots, X^J &\sim \prod_{j=1}^{J} U\left(X^j, d\right) p\left(X^j\right). & (3) \end{aligned}$$

An important feature of this algorithm is that the draws from X^j are tilted away from the unconditional predictive density $p(X^j)$, toward $p(X^j)U(X^j, d)$. While the marginal draws of d converge to d^*, those of X^j concentrate on the regions of the domain of X^j with a higher utility. The algorithm effectively concentrates on values of X^j more *"useful"* to discover d, while the draws of $d^{(g)}$ tighten around d^* as the algorithm converges. Sampling the X^j's in this utility-tilted way helps converge quickly to the relevant region of the decision space using $d|\widetilde{X}^J$ and increases the computational efficiency over traditional methods.[2]

It is instructive to see how this differs from the *standard* expectation-optimization algorithms for two reasons. First we draw efficiently from $p(X|d)$ as just discussed. In contrast, expectation-optimization algorithms, at every step of d, draw G samples from $X^{(g)} \sim p\left(X\right)$, the predictive density of X. Second, they approximate the expected utility $E_X\left[U(X, d)\right]$ by $\frac{1}{G}\sum_{g=1}^{G} U\left(X^{(g)}, d\right)$, as well as all required derivatives similarly. An optimization step over d is performed typically via a gradient-based method. The process is repeated until convergence. For functional optimization problems this can be computationally intractable.

2.2 Functional optimization

We now show that the algorithm shown above readily extends to the case of an agent who wants to study the optimal functional decision $d^*(Y)$ for a wide range of values of a state variable Y of

[2] Also note that MCMC methods in high dimensional problems can be polynomial time convergent here in J and this efficient sampling reduces the curse of dimensionality. Computationally, as the model is set up as $X_j|d$ and $d \sim \mu(d)$, finding $d^\star$ is equivalent to estimating a hyper-parameter in a Bayesian hierarchical model. Fast convergence results are known for many of these hierarchical model specifications, Rosenthal (1996), Roberts and Polson (1994), Polson (1996).

interest. For example, the agent may want to understand the sensitivity of the portfolio to potential variations in volatility or updates of expected returns.

In this functional optimization problem, the uncertainty about the state variable X is described by the conditional predictive distribution $p(X|Y)$. This distribution results from marginalizing out other state variables and parameters that are directly relevant to the agent. Specifically, the agent wants to solve

$$\max_{d(Y)} E_{X|Y} \left[U \left(X, d \right) \right] .$$

To solve this problem, define the augmented joint distribution

$$\pi_J(\widetilde{X}^J, d, Y) \propto \prod_{j=1}^{J} U(X^j, d) p(X^j|Y) \mu(d, Y) \tag{4}$$

for a measure $\mu(d, Y)$ now defined jointly over the decision and state variable. Again μ is chosen to ensure that the regularity conditions, namely positivity and integrability, hold true.

Consider the following MCMC algorithm that simulates from $\pi_J(\widetilde{X}^J, d, Y)$:

$$X^j|d, Y \quad \propto \quad U\left(X^j, d\right) p\left(X^j|Y\right) \text{ for } j = 1, ..., J \ , \tag{5}$$

$$d|\widetilde{X}^J, Y \quad \propto \quad \prod_{j=1}^{J} U\left(X^j, d\right) p\left(X^j|Y\right) \ , \tag{6}$$

$$Y|\widetilde{X}^J, d \quad \propto \quad p(\widetilde{X}^J|Y) = \prod_{j=1}^{J} p(X^j|Y) \ . \tag{7}$$

Upon integrating out $\widetilde{X}^J$, the above set of conditionals reduces to those given in (**??**-**??**). Hence, the results presented below specialize to the simple global maximum expected utility problem, using (**??**), (**??**), instead of (**??**-**??**).

For the purpose of functional optimization, one could conceivably use an algorithm based upon iterating (**??**-**??**), for a selected value of Y, repeating the procedure for a discrete grid of values of Y. This brute force procedure is not efficient as possibly low probability, values of Y may be selected. In contrast, the complete algorithm in (**??**), (**??**), (**??**) has two advantages. First, the global optimum d^* is a by-product. Second, by drawing Y values more frequently with higher expected utility, as per the conditional in (**??**), we obtain efficiency gains. One may worry that, as J gets very large, the algorithm collapses, as it should, on d^*, hence around some Y^* as can be seen from (**??**). It would becomes then impractical as a means to describe the function $d^*(Y)$ on a range of Y. This is unlikely in practice as even with $J = 200$, (**??**) is akin to a draw from of X given 200 observations, and the conditional $Y|\widetilde{X}^J, d$ would require a much higher value of J to fully collapse.

Practically, the algorithm produces joint draws (d, Y) that can be plotted. A smoothed estimate of the functional $d^*(Y)$ then follows using one of the many known kernel-smoothing techniques and the optimal decision d^* can be found by averaging the draws of d as in any MCMC algorithm.

We now show that the marginal of $\pi_J(d|Y)$ collapses on the functional relationship $d^*(Y)$ as J gets large. First, it follows from (**??**) that:

$$\pi_J\left(d|Y\right) \propto e^{J\log\mathbb{E}_{X|Y}[U(X,d)]}\mu(d) \ .$$

In turn, as $J \to \infty$, we have that $\pi_J\left(d|Y\right) \longrightarrow d^*\left(y\right) = \underset{d(y)}{\arg\max} E_{X|y}\left[U\left(X,d\right)\right]$. The issue then is to find an efficient MCMC algorithm to sample this joint distribution over the space (d, X, Y). The Markov Chain produces draws $\left\{d^{(g)}, \widetilde{X}^{(j,g)}, Y^{(g)}\right\}_{g=1}^{G}$.

In single period optimization problems, we know that maximizing $U\left(d\right)$ is equivalent to simulating from the sequence of densities $\pi_J\left(d\right) \propto \exp\left(J\log U\left(d\right)\right)$. To avoid convergence issues, this suggests that the Markov Chain should operate in the higher dimensional space of $\left(d, \widetilde{X}^J\right)$. MCMC is then the natural methodology and this is why we draw iteratively from $p\left(d|\widetilde{X}^J\right)$ and $p\left(\widetilde{X}^J|d\right)$.

Recall that standard simulation draws X from $p(X)$. In contrast, this approach samples the J random variables, $X^j|d \propto U(d, X^j)p(X^j)$. This is why the approach works well for large dimensions, complicated distribution and potentially non-smooth utility. For example, in the case where the maximizing decision depends critically on the tail behavior, one generates more samples from the high-utility portions of the state space.

Another advantage of this joint optimization and integration approach is that it delivers Monte Carlo error bounds in high dimensions. This is due to the fact that using MCMC sampling can result in fast convergence such as geometric convergence in nearly all cases and polynomial time in some cases. Contrast this approach with even sophisticated Monte Carlo strategies such as importance sampling that generates the standard central limit theorem type $\sqrt{G}$ convergence[3].

[3] Aldous (1987) discusses why this is insufficient for high dimensional problems. Random polynomial time convergence

2.3 Asymptotics in J

We now derive the asymptotic properties of the sample of draws $(d_J^{(G)}, \widetilde{X}^{J,(G)})$ as a function of J and G. By the Ergodic theorem, as $G \to \infty$, the draws $(d_J^{(G)} \widetilde{X}^{J(G)})$ converge to draws from $\pi_J(d, \widetilde{R}^J)$. Now define $\sigma_U(d) = \nabla^2 \log U(d)$. Then, we have the following asymptotic normality, in the limit in J for the draws $d_J^{(G)}$. The proof is in the Appendix.

Theorem 1 *Let d_J^G be a random draw from the marginal distribution $\pi_J(d)$ then*

$$\sqrt{J}\left(d_J^{(G)} - d^*\right) \Rightarrow N\left(0, \sigma_U(d^*)^2\right) \tag{8}$$

The main requirement is for $\pi_J(d, X)$ to be integrable and positive. Jacquier, Johannes and Polson (2007) apply this methodology to likelihood maximization, where the optimal decision (d) is the MLE of a parameter vector and the state variable (R here) is a latent variable such as a vector of stochastic volatilities. In cases where singularities in the likelihood function may occur, they introduce a Lebesgue measure μ over the parameter space . In the decision theoretic framework studied here, R is described by its proper predictive density, but the utility may not meet the integrability requirement. We discuss below how a simple transformation and the measure applied to d, will help.

2.4 Comparison with Standard MC methods

The typical approach to optimization proceeds with two steps. First, expected utility is computed by averaging over a number of draws from naive simulation methods such as direct Monte Carlo or importance sampling. Second, given this approximate expected utility, one typically applies brute force optimization approaches such as hill-climbing gradient based methods. For example, consider the problem of choosing a control, d, to maximize expected utility, $U(d) = E_X[U(d, X)]$ where X is a random variable. The standard approach draws $\theta^{(g)}$ from $p(X)$, uses a Monte Carlo approximation to expected utility, $\widehat{U}(d) = G^{-1} \sum_{g=1}^{G} U\left(d, X^{(g)}\right)$, and then optimizes with standard gradient based approaches, see Rubinstein (1983). This approach, known as stochastic optimization, while perfectly understandable is not tailored to deal with high dimension of uncertainty (X), a complicated distribution of uncertainty, $p(X)$, high dimensional choice (d), and non-smooth utility (U).

allows rigorous Monte Carlo bounds that break the curse of dimensionality, see Polson (1996)

One problem with this two-step approach is that the optimal d is found given a set of simulated X's. If the utility $U(d, X)$ is high in places of low $p(X)$ - probability, such a search is in fact inefficient. For example, if an investor is particularly sensitive to outlying observations, the choice of d is very sensitive to how well the tails of the distribution are sampled. The naive maximization then requires a very large G and this breaks down when X is high dimension or $p(X)$ is difficult to sample.

3 Extensions and Specific Issues

3.1 Expected Loss Minimization

The utilities most always used in financial economics are of the power and exponential families. Both are negative. In the asset allocation section, we show how to remedy this problem by simply shifting and bounding the utility. The general algorithm can also be used to minimize functions if needed. Suppose that we wish to minimize expected loss, $\min_d E[L(X,d)]$. Then, if $d \in \mathcal{D}$ is compact and if $L \leq 0$, we consider $\max_d E[U(X,d)]$ where $U = -L \geq 0$. We now have a well defined π_J. Hence the joint distribution looks like $\pi_J \sim \prod_{j=1}^{J} L(X^j, d)$. As $\mathcal{D}$ is compact, finding $\min \pi_J(d)$ is equivalent to finding $d^\star$. Again we can simulate draws $d^{(g)}$ from $\pi_J(d)$ and find its minimum. We then use the draws of d to obtain a Monte-Carlo histogram of $\pi_J(d)$.

3.2 Exploiting Latent Variables

When the distribution on the source of uncertainty, $p(X)$, has a mixture or latent variable structure then we can use marginalisation to improve the efficiency of our algorithm. Consider a generic latent variable model where

$$p(X) = \int p(X|V)\, p(V)\, dV, \quad \text{or} \quad p(X|Y) = \int p(X|V,Y)\, p(V|Y)\, dV,$$

depending on whether there is conditional information Y or not. Here V denotes a vector of latent mixing variables. For example, in an option replication problem V might be the future average stochastic volatlity. In general our algorithm proceeds with the augmented joint distribution

$$\pi_J(d, \widetilde{V}^J, \widetilde{X}^J) \propto \prod_{j=1}^{J} U\left(d, X^j\right) p(X^j|V^j) p(V^j) \mu(d).$$

We draw samples $\{d^{(g)}, (\widetilde{X}^J)^{(g)}, (\widetilde{V}^J)^{(g)}\}$ by MCMC. The marginal $\pi_J(d)$ again concentrate on the optimal decision rule d^* as $J \to \infty$.

3.3 Exploiting Conjugate Utility

Lindley (1976) discusses the notion of a *conjugate* utility function where the *marginal* utility $U(d, V) = \int U(d, X) p(X|V) dX$ is available in closed form. One can then simplify the analysis by collapsing out the state vector X, only using the joint distribution

$$\pi_J(d, \widetilde{V}^J) \propto \prod_{j=1}^{J} U\left(d, V^j\right) p(V^j).$$

Here, the concentration on $U(d^*)$ is achieved by making J copies of the mixing state variables V without the need for the corresponding J copies of the state variable X. A common application is for non-normal distributions, such as scale mixture of normals, see Andrews and Mallows (1974) and Carlin and Polson (1992).

Specifically, suppose that the distribution $p(X)$ falls into the class of mixtures of exponential distributions where

$$p(X) = e^{X\alpha} H(X)^{\beta} K(\alpha, \beta), \quad \text{with } K(\alpha, \beta)^{-1} = \int e^{X\alpha} H(X)^{\beta} dX,$$

where $V = (\alpha, \beta)$ are possibly random hyper-parameters. The class of conjugate utility functions is then defined by

$$U(d, X) = e^{XG(d)} H(X)^{\beta(d)} C(d)$$

for pre-specified functions C, G and H. Then, given $V = (\alpha, \beta)$, we can calculate the marginal utility $U(d, V)$ in closed form as

$$U(d, V) = C(d) \frac{K(\alpha, \beta)}{K(\alpha + G(d), \beta + \beta(d))}.$$

Recall from (**??**) that the algorithm requires simulating from a conditional of the form $\pi_J(X^j|d) \propto U(X^j, d)\, p(X^j)$. In this conjugate utility framework this is given by an exponential family model of the form

$$\begin{aligned}
\pi_J(X^j|d) &\propto e^{X(\alpha+G(d))} H(X)^{\beta+\beta(d)} \\
&= e^{X(\alpha+G(d))} H(X)^{\beta+\beta(d)} K(\alpha+G(d), \beta+\beta(d))
\end{aligned}$$

This is computationally valuable as we require J draws from a distribution $\pi_J(X_j|d)$ of known form. This distribution is of the same functional form as $p(X)$ but with hyper-parameters given by the change $(\alpha, \beta) \to (\alpha + G(d), \beta + \beta(d))$. The distribution on the decision variable d can also be simplified using the Besag identity:

$$\pi_J(d) = \frac{\pi_J(d, \widetilde{X}^J)}{\pi_J(\widetilde{X}^J|d)}.$$

Hence, we can evaluate a density ratio at any two values leading naturally to a Metropolis algorithm for updating the variables jointly.

If (α, β) are known, the original expected utility $U(d)$ then obtains in closed-form, in terms of the normalization constant K. However, one still might use our algorithm to find the optimum, as K is usually defined in terms of non-standard functions (e.g. modified Bessel functions in our portfolio allocation) and numerical derivatives are harder to find than simple function evaluations. One of the advantages of MCMC is that it leads to a derivative free optimization algorithm.

3.4 Using the Slice Sampler to Enforce Constraints

In our simulation approach, the function to be maximized must be positive so as to represent a distribution kernel. Hence we need to shift the utility function to ensure its positivity. Consider a joint distribution where, in addition to copying over the state variable X^j, we introduce a vector of latent uniform auxiliary *slice* variables ω^j given by

$$p(\widetilde{X}^J, \widetilde{u}^J, d) \propto \prod_{j=1}^{J} p(X^j) 1_{0 \le \omega^j \le U(X^j, d)}$$

where 1_A denotes the indicator of A.

Slice sampling will be efficient here because the utility function is monotone increasing. Therefore, we will have two well-defined inverse functions. More specifically, the inequality

$$0 \leq \omega^j \leq U(X^j, d),$$

implies $X^j \geq U_X^{-1}(\omega^j, d)$. For the decision variable inequalities to hold we get a set $d \in U_d^{-1}(\omega^j, X^j)\ \forall j$. The MCMC algorithm based upon slice sampling is as follows. Given a current draw $\{x^{(g)}, u^{(g)}, d^{(g)}\}_{g=1}^G$.

1. Draw $x = (X_1, \ldots, X_J)$ using $p(X_j|u_j, d) \sim p(X_j)1_{X_j \geq U_x^{-1}(u_j, d)}$, a truncated version of $p(X_j)$ on the region $X_j \geq U_x^{-1}(u_j, d)$
2. Draw slice variables $u = (u_1, \ldots, u_J)$ using $p(u_j|X_j, d) \sim \text{Uni}\,(0, U(X_j, d))$, a uniform.
3. Draw the decision variable $p(d|x^J, u^J) \sim 1_{d \in U_d^{-1}(u_j, X_j)\ \forall j}$, a uniform on a constrained set.

Hence in this setting there is no need for a Metropolis algorithm.

4 Application: Asset Allocation

We now illustrate the methodology in the classic asset allocation framework. Consider the standard power utility specification

$$U(W, d) = \frac{W(d, R)^{1-\gamma}}{1-\gamma},$$

where $\gamma \geq 1$ is the relative risk aversion, $W(d, R)$ is the future wealth resulting from the investment of \$1, with weights d in a risky asset with return R and $(1-d)$ in the risk-free asset with return R_f. The wealth at the end of the period is $W = dR + (1-d)R_f$. It is also the portfolio return.

This power utility is most always combined with the log-normality assumption for returns where the risky asset is distributed i.i.d $\ln R \sim N(\mu, \sigma^2)$. Merton (1969) gives us the classic solution to the problem for i.i.d returns, known parameters and continuous re-balancing:

$$d^* = \frac{\mu - R_f}{\gamma\sigma^2}$$

The continuous re-balancing assumption is crucial to obtain a simple closed form solution. First, in Merton's world, the agent observes prices in continuous time and therefore knows the current value of σ. Second, it guarantees that wealth also follows a log-normal distribution. Due to estimation risk, in discrete time, it does not follow that X is log-normal and MEU becomes quickly intractable. The problem becomes computationally difficult in discrete time as the log-normal is not stable in the portfolio. A related problem is that, while $d \geq 1$ does not pose any problem when continuous re-balancing is permitted, it can not be allowed in discrete time as it can imply infinite disutility, see Brandt (2010) for a discussion. Therefore, following Kandel and Stambaugh (1996), we will bound d in [0,1].

Parameter uncertainty introduces a posterior distribution on the parameters (μ, σ^2). The resulting predictive distribution $p(R)$ is a log-normal mixture. In terms of the algorithm presented here, it does not make any practical difference for the global MEU problem, provided that one can make direct draws from the predictive density $p(R)$.

One needs to be careful that expected power utility remains well-defined especially when σ is estimated. In particular, Kandel and Stambaugh (1996) and Geweke (2001) point out the issue of parameter uncertainty and power utility. Jobert et al (2005) and Weitzman (2007) enforce finiteness of expected utility with priors of the form

$$p(\mu, \sigma^2) \propto \phi(\mu, \sigma^2) N(\mu_0, \sigma^2 A) IG(\nu_0, s_0^2),$$

where the usual conjugate normal-inverted gamma distribution is multiplied by a secularizing function ϕ. Jobert et al. (2005) uses a function $\phi(\mu, \sigma^2) = e^{-c/\sigma^4}$ while Weitzman uses $\phi(\mu, \sigma^2) = 1_{\sigma^2 > 0}$ to ensure finiteness of expected utility.

Zellner and Chetty (1965), Klein and Bawa (1976), Brown (1978) and Bawa et al. (1979) are among the first to formally introduce parameter uncertainty in the problem. Kandel and Stambaugh (1996) and Barberis (2000) concentrate on the effect of parameter learning on the optimal allocation. Jacquier, Kane and Marcus (2005) show how estimation uncertainty on the mean, even without learning, affects the optimal asset allocation, especially in the long-term. See Brandt (2010) and Jacquier and Polson (2010) for recent discussions of portfolio choice.

4.1 Implementation

We now implement the algorithm in (**??**) and (**??**) without assuming continuous re-balancing. The random utility of wealth is $\left(dR+(1-d)R_f\right)^{1-\gamma}/(1-\gamma)$, which is negative. We shift the utility to enforce positivity. Namely, we compute a value L of the utility for a particularly adverse scenario where $d=1$ and a return $R=R_L$, far in the left tail. Thereafter we consider the shifted utility $U_L(d,R)=U(d,R)-L$. The ranking of risky prospects is not affected by a shift in the utility. Then, the measure $\mu(d,R)$ is used to enforce positivity of the shifted utility. In an MCMC algorithm, this is done by rejecting the draws leading to a negative shifted utility. In practice, we chose R_L low enough so that such draws are rarely, if ever, encountered within a simulation cycle GJ.[4]

Following (**??**) and (**??**), the algorithm involves draws from the $J+1$ conditionals:

$$
\begin{aligned}
d|\widetilde{R}^J &\propto \prod_{j=1}^{J}\left(\frac{(dR^j+(1-d)R_f)^{1-\gamma}}{1-\gamma}-L\right), \text{ for } j=1,\ldots,J && (9)\\
R^j|d &\propto \left(\frac{(dR^j+(1-d)R_f)^{1-\gamma}}{1-\gamma}-L\right) p(R^j). && (10)
\end{aligned}
$$

We now show how to draw from these two kernels. Consider $(\mu=0.075,\sigma=0.2)$ and $\ln R_f=0.025$ in annual terms. We will consider annual and monthly returns. For relative risk aversions of 8 and 4, the Merton solutions are 0.22 and 0.44 respectively. We use $\mu-4\sigma$ and $d=1$ to compute a lower bound L on utility used throughout the simulations.

4.1.1 Drawing from $p(d|\widetilde{R}^J)$

The first point which we will illustrate is that the density concentrates on the optimum as J gets large, for practical values of J. The density $p(d|\widetilde{R}^J)$ in (**??**) does not correspond to any known density. However, as d is bounded, its integration constant is easily computed by numerical integration. We do this over a grid of 100 values of d.

Figure **??** uses the following distributions. We draw J copies $\widetilde{R}^J=(R^1,\ldots,R^J)$ of the return from $p(R)$. It plots the normalized $p(d|\widetilde{R}^J)$ for annual data with $\gamma=4$, for $J=2,20,100,1000$. It shows that the distribution concentrates quickly around the Merton value. Note that, while we expect

[4] This choice must be made ahead of time, so it is based upon $p(R^j)$. However, the algorithm draws from $p(R^j|d)$. This is not a problem as $p(R^j|d)$ is shifted to the right of $p(R^j)$, as $U(d,R)$ is increasing in R for $d\geq 0$.

the optimal solution to be close to Merton's, there is no reason for it to be exactly the same. This is because the algorithm does not assume continuous trading.

In our algorithm, draws of $\widetilde{R}^J$ come from $p(R^j|d)$ which is more efficient than drawing from $p(R)$. So Figure **??** may slightly understate the rate at which the densities concentrate on the d^* as J increases. We then compute the kernel of $p(d|\widetilde{R}^J)$ and its integration constant by numerical integration. For very low values of J, such as 2 or 10, different $\widetilde{R}^J$ can produce very different densities. Here we want to show the *typical* density for a given J. So for each value of J, we average the densities over $G = 100$ simulations. This figure with $4GJ = 448800$ normalized density computations takes a fraction of a second of laptop CPU time.

Figure **??** indeed shows that the typical conditional concentrates quickly around the Merton value as J increases. Note for $J = 2$ that the density is close to a uniform on $[0, 1]$. For values of J between 100 and 1000, the draws concentrate very precisely on the optimum.

Figure **??** shows a case of higher risk aversion where the bound on d will necessarily affect the shape of the density. While the density remains asymmetric due to the proximity of the bound, it also concentrates very fast on the optimum which seems extremely close to the Merton value. This demonstrates that the algorithm works for practical values of J.

We now discuss our choice of Metropolis blanket for $p(d|\widetilde{R}^J)$ as we will not be able to make direct draws from that density. We need to propose a blanket which can be implemented for any conditional density. Figure **??** shows four individual densities for various scenarios of risk aversion and J. The densities can take very different shapes. The top left plot shows a mode at the corner as the high risk aversion combines with a low J; the density bunches towards the bound. The top right plot shows another shape typical of a low J. In the bottom left plot, a larger J starts being effective. Even though the optimum is very close to 1, an interior mode appears. For an even larger J, the mode is clearly differentiated from the bound in the bottom right plot.

Given the bounded domain and the easily performed numerical integration, we can use an accept-reject rather than a Metropolis step to draw from $p(d|\widetilde{R}^J)$. The Beta distribution is often mentioned as the candidate of choice for drawing bounded variables. Figure **??** however shows that it would not be a good choice. This is because the Beta density is zero at the boundaries of the domain for most practical values of its parameters. It would therefore be a very inefficient blanket for the many cases where the distribution of the decision exhibits some bunching at the bounds.

In fact, the truncated normal should fit this density very well. This is because $p(d|\widetilde{R}^J)$ is a product of J independent kernels that should converge very quickly to a normal distribution by a central limit theorem. This property of d is quite general and robust to parameter uncertainty. To fit the normal, we selected 4 points from $p(d|\widetilde{R}^J)$, namely [0.05,0.5,0.95], plus the mode. These are the red dots on Figure **??**. The fitted normal also appears on the figure in blue dashes. It is quasi indistinguishable from the true conditional distribution. While we have in theory an accept-reject step, it is very close to a direct draw in practice and there will be rejections to implement the truncation on $d \in [0, 1]$.

4.1.2 Drawing from $p(R^j|d)$

The density $p(R^j|d)$ is the product of the predictive density, whatever it may be, by one shifted utility kernel. $d^{(g)}$, the latest draw of d in the MCMC chain is a parameter for that density. We consider two very different values of d; namely 0.2 and 0.9, and a risk aversion of 2.

It is convenient to consider the logarithm of returns, rather than returns as the random variable here. So, Figure **??** plots the relevant normalized densities, where $logR$ rather than R is the variable. The vertical scale is in logarithms to show the tail behavior. The red dashed lines are the normal predictive densities of $\log R$. The green dotted lines are the utility kernels. Note that they are slightly increasing in R^j, the monthly return. The product of these two is $p(R^j|d)$, the black continuous line, with a mean slightly shifted toward the positive returns. This property of the utility kernel, increasing in R, is again quite general and robust. Its primary effect is the mean shift on $p(R)$.

Further, the examination of the tail behavior of $U(R^j, d)$ shows that it does not modify the tail behavior of $p(R^j)$. So we will be able to implement an accept-reject step on $p(R^j|d)$ with a distribution similar to $p(R^j)$. We use a normal blanket for $\ln(R^j)|d$ that incorporates the mean shift. To do this, we compute the mode of $p(logR^j|d)$ and use a normal distribution with this mode as its mean. The resulting normal is the blue dashed line on the figure.

This blanket is extremely close to $p(\ln R^j|d)$, which will result in a very efficient accept-reject algorithm. Higher values of γ and lower values of d led to similar situations, with in fact even flatter kernels $U_L(R^j, d)$, and blankets closer to the target.

4.2 Slice Sampling Implementation

Alternatively, we can implement a slice sampling algorithm as indicated in section 3.4 to ensure the positivity of the function which expectation is maximized. Namely, we need $0 \leq d \leq d_U$ for the fraction invested in the risky asset. Recall that the portfolio return is $X_d = 1 + d(e^X - e^{r_f})$. Denote $d_L = 1 - d_U$. Then power utility is

$$U(X,d) = \frac{X_d^{1-\gamma} - 1}{1-\gamma},$$

where $\gamma > 1$. To be able to implement our algorithm we need utility to be the kernel of a distribution. So we need to transform utility into a positive function. We use the transformed utility:

$$U(X,d) = d_L^{1-\gamma} - X_d^{1-\gamma}.$$

This leads to the following constraints to implement the algorithm. First U_X^{-1} is defined by

$$x_j \geq \ln\left(d^{-1}\left(\left(d_L^{1-\gamma} - u_j\right)^{\frac{1}{1-\gamma}} - 1\right) + e^{r_f}\right)$$

Secondly, the constraint set for the decision variable is defined by the series of inequalities given by

$$d(e^{x_j} - e^{r_f}) \leq \left(d_L^{1-\gamma} - u_j\right)^{\frac{1}{1-\gamma}} - 1$$

Whether $e^{x_j} - e^r$ is positive or or not we get different inequality bounds on d. Eventually as J gets large this shrinks to the optimum.

5 Conclusion

We provide a new simulation based methodology for maximizing expected utility using MCMC. This approach has a number of advantages. First, it simultaneously performs integration and optimization. Specifically, it does not resort to a Monte Carlo approximation of expected utility and then a derivative, gradient-based method for finding the optimum. This has been successful in finding maximum likelihood estimates in complex models with latent variables, see Jacquier, Johannes, Polson (2007) and Boivin and Giannoni (2006). Second, it applies to a wide class of problems with mild positivity and integrability assumptions. For example, if the uncertainty in the state variable density is itself a mixture model, one can use data augmentation as in known MCMC algorithms. Finally, the methodology extends to functional optimization, where one seeks to find the optimal decision rule $d^*(Y)$ as a function of the realization of the uncertain state variable Y.

We describe the method in details for a simple asset allocation framework. The MCMC algorithm is in practice a simple Gibbs sampler with embedded accept-reject steps. The conditional $p(d|\widetilde{X}^J)$ is a product of independent utility kernels. As such, it converges very quickly to a normal distribution. Indeed, for any value of J above 10, a truncated normal can fit the conditional distribution of the decision nearly perfectly. This accept-reject step amounts to a direct draw by any practical standards. The other conditional $p(X^j|d)$ can also be fitted very closely, with a (normal) distribution, which will lead to a very efficient accept-reject step. This is because the utility kernel shifts $p(X^j)$ without affecting its shape much. These features are robust to the introduction of parameter uncertainty and will carry over to frameworks with latent variables such as with stochastic volatility.

References

Aldous, D., (1987) On the Markov Chain Simulation Method for Uniform Combinatorial Distributions and Simulated Annealing, *Probability in the Engineering and Informational Sciences* 1:33-46.

Andrews, D.F., and Mallows C.L., 1974, Scale Mixtures of Normal Distributions, Journal of the Royal Statistical Society, Series B, 36(1): 99-102.

Barberis, N., 2000, Investing for the long run when returns are predictable, *Journal of Finance 55*, 225-264.

Bawa, V., Brown, S. and R. Klein, 1979, *Estimation Risk and Optimal Portfolio Choice*, New York, North-Holland.

Berger, J., 1985, *Statistical Decision Theory and Bayesian Analysis, Springer-Verlag*, New York, 1985.

Boivin, J, and Giannoni, M.P., 2006. DSGE Models in a data-rich environment. NBER Working paper 12772, December.

Brandt, M., Goyal, A., Santa-Clara, P. and Stroud, J., 2005, A Simulation approach to dynamic portfolio choice with an application to learning about return predictability. *Review of Financial Studies*, 18, 831-873.

Brandt, M. 2010, Portfolio Choice Problems, *Handbook of Financial Econometrics*, (eds Ait-Sahalia and L. Hansen), 1, 269-336.

Brockwell, A. and J.B. Kadane, 2003, A gridding method for sequential Bayesian decision problems.*Journal of Computational and Graphical Statistics*, 12(3), 566-584.

Brown, S. J. (1978), The portfolio choice problem: Comparison of certainty equivalent and optimal Bayes portfolios, *Communications in Statistics: Simulation and Computation B7*, 321-334.

Carlin, B. P. and Polson, N. G. (1992), Monte Carlo Bayesian Methods for Discrete Regression Models and Categorical Time Series, In *Valencia 5* (eds Bernardo et al), 577-586.

Chamberlain, Gary, (2000), Econometrics and Decision Theory, *Journal of Econometrics*, 99, 255-283.

Geweke, J. (2001), A Note on Some Limitations of CRRA Utility, *Economics Letters*, 71, 341-346.

Jobert, A., Platania, A. and L.C.G. Rogers (2005). A Bayesian solution to the equity premium puzzle. *Working paper, Cambridge University.*

Jacquier, E., Johannes, M., and N.G. Polson, 2007, MCMC Maximum Likelihood for Latent State Models, *Journal of Econometrics*, 137, 615-640.

Jacquier, E., Marcus, A., and A. Kane, 2005, Optimal Forecasts of Long-Term Returns and Asset Allocation: Arithmetic, Geometric or Other Means, Journal of Financial Econometrics 3, 37-55.

Jacquier, E., and Polson, N.G., 2011, Bayesian Econometrics in Finance. *Handbook of Bayesian Econometrics*, eds J. Geweke, G. Koop, and H. Van Dijk, Oxford University Press, London.

Johannes, M and Polson, N.G., 2010, MCMC methods for Financial Econometrics. *Handbook of Financial Econometrics* (eds Y. Ait-Sahalia and L. Hansen), 1-72.

Kandel, S., and R.F. Stambaugh., 1996, On the predictability of stock returns: An asset allocation perspective, *Journal of Finance*, 51(2), 385-424.

Kirkpatrick, S., C.D. Gelatt and M.P. Vecchi, 1983, Optimization by simulated annealing, *Science 220*, 671-680.

Klein, R.W., and V.S.Bawa, 1976, The effect of estimation risk on optimal portfolio choice. *Journal of Financial Economics*, 3, 215-231.

Liang, F., and W.H. Wong, 2001, Real Parameters Evolutionary Monte Carlo with Applications to Bayesian Mixture Models, *Journal of the American Statistical Association* 96, 653-666.

Lindley, D.V., 1976, A class of utility functions, *Annals of Statistics*, 4, 1-10.

Merton, Robert C., 1969, Lifetime Portfolio Selection under Uncertainty: The Continuous-Time Case. *The Review of Economics and Statistics* 51: 247-257.

Moreno-Diaz, A., M.A. Virto and D. Rios Insua, 2003, Approximate solutions to semi Markov Decision Processes using MCMC, In: *Computer Aided Systems*, Springer, 151-162.

Mueller, P., 2000, Simulation based optimal design, *Bayesian Statistics 6* (eds Bernardo, et. al.) Oxford, 459-474.

Mueller, P., Sanso, B. and M. DeIorio, 2004, Optimal Bayesian design by inhomogeneous Markov Chain Simulation. *Journal of the American Statistical Association*,

Pastor, L., 2000, Portfolio selection and asset pricing models, *Journal of Finance*, 55, 179-223.

Pastor, L., and Stambaugh, R. F., 2001, Comparing asset pricing models: An investment perspective, *Journal of Financial Economics*, 56, 335-381.

Pincus, M. (1968). A Closed Form Solution of Certain Dynamic Programming Problems. *Operations Research*, 18, 1225-1228.

Polson, N. (1996). Convergence of MCMC algorithms, *Bayesian Statistics 5* (eds Bernardo, et. al.) Oxford, 297-321

Roberts, G. and Polson, N. (1994). On the Geometric Convergence of the Gibbs sampler. *J. R. Statistical Soc, B*, 377-384.

Rosenthal, J. (1996). Rates of Convergence for Gibbs sampling of variance component models. *Annals of Statistics*, 73, 740-61.

Rubinstein, R.Y., (1983), Smoothed Functionals in Stochastic Optimization, Mathematics of Operations Research, Vol.8, No.1, pp.26–33, 1983

Tierney, L., R. Kass and J.B. Kadane (1989). Approximate Marginal Densities of Nonlinear Functions. *Biometrika*, 76(3), 425-433.

Van Laarhoven, P.J., and E.H.L Aarts, 1987, *Simulated Annealing: Theory and Applications*, CWI Tract 51, Reidel, Amsterdam.

Weitzman, M. (2007). Subjective Expectations and Asset-Return Puzzles. *American Economic Review*, 97(4): 1102-1130.

Zellner, A., 2002, Information Processing and Bayesian Analysis, *Journal of Econometrics*, 107, 41-50.

Zellner, A. 1971, *An Introduction to Bayesian Inference in Econometrics*, John Wiley and Sons, New York.

Zellner, A. and Chetty, V.K., 1965, Prediction and decision problem in regression models from the Bayesian point of view, *Journal of the American Statistical Association*, 60, 608-616.

Appendix:

First, define the neighborhood $N_{d^*}^{(a,b)}(J) = \left(d^* + \frac{a\sigma_U(d^*)}{\sqrt{J}}, d^* + \frac{b\sigma_U(d^*)}{\sqrt{J}} \right)$. We make the following assumptions.

(A1) The utility function $U(d, \theta)$ is continuous and positive at d^*, and $U(d)$ is almost surely twice differentiable in some neighborhood of d^*;

(A2) For any (a, b) there exists a J such that for $d \in N_{d^*}^{(a,b)}(J)$ there exists a ϵ_J where $\epsilon_J \to 0$ as $J \to \infty$ such that $\sup_{N_{d_n^*}^{(a,b)}(J)} |R(a)| < \epsilon_J < 1$ where $R(d) = \sigma_U(d)^{-1} \log \left(\frac{U(d^*)}{U(d)} \right)$. Now the marginal density over the decision space can be written as

$$\pi_J(d) = \pi_J(d^*) \exp \left(J \log \left(\frac{U(d)}{U(d^*)} \right) \right).$$

By Taylor's theorem we can write

$$\log U(d) = \log U(d^*) + \frac{1}{2}(d - d^*)^2 \nabla^2 \log U(d^{**}),$$

where $d^{**} = d + \gamma(d^* - d)$ and $0 < \gamma < 1$. By definition, $R(d) = \sigma_U(d)^2 \log \left(\frac{U(d)}{U(d^*)} \right)$. Then

$$\pi_J(d) = \pi_J(d^*) e^{-\frac{J}{2\sigma_U^2}(d-d^*)^2(1-R(d))}.$$

We now need to show that for any $a < b$ we have that

$$\lim_{J \to \infty} P(a < \sqrt{J}\ \sigma_U(d_J^G - d^*) < b) = \Phi(b) - \Phi(a)$$

where $\Phi(\cdot)$ is the standard normal cdf. Now

$$P(a < \sqrt{J}\ \sigma_U(d_J^G - d^*) < b) = \int_{N_{d^*}^{(a,b)}(J)} \frac{e^{J\log U(d)}}{m_J} dd$$

Now, for any $\epsilon > 0$, by continuity of $U(d)$ (assumption (A1)) at d^* we can find J so that

$$I = \frac{\pi_J(d_n^*)}{m_J} \int_{N_{d^*}^{(a,b)}(J)} e^{-\frac{J}{2\sigma_U^2}(d-d^*)^2(1-R(d))} dd$$

By assumption (A2) we have $\sup_{N_{d^*}^{(a,b)}(J)} |R(d)| < \epsilon_J < 1$. Therefore, taking the limit as $J \to \infty$ and noting that $\epsilon_J \to 0$ we have the desired result, as

$$\int_{N_{d^*}^{(a,b)}(J)} e^{-\frac{J}{2\sigma_U^2}(d-d^*)^2(1+\epsilon_J)} dd = \sqrt{2\pi}\sigma_U(1+\epsilon_J)^{-\frac{1}{2}} \left[\Phi(\sqrt{J}\sigma_U^{-1}\left(\frac{a\sigma_U}{\sqrt{J}}\right) - \Phi(\sqrt{J}\sigma_U^{-1}\left(\frac{b\sigma_U}{\sqrt{J}}\right)\right]$$

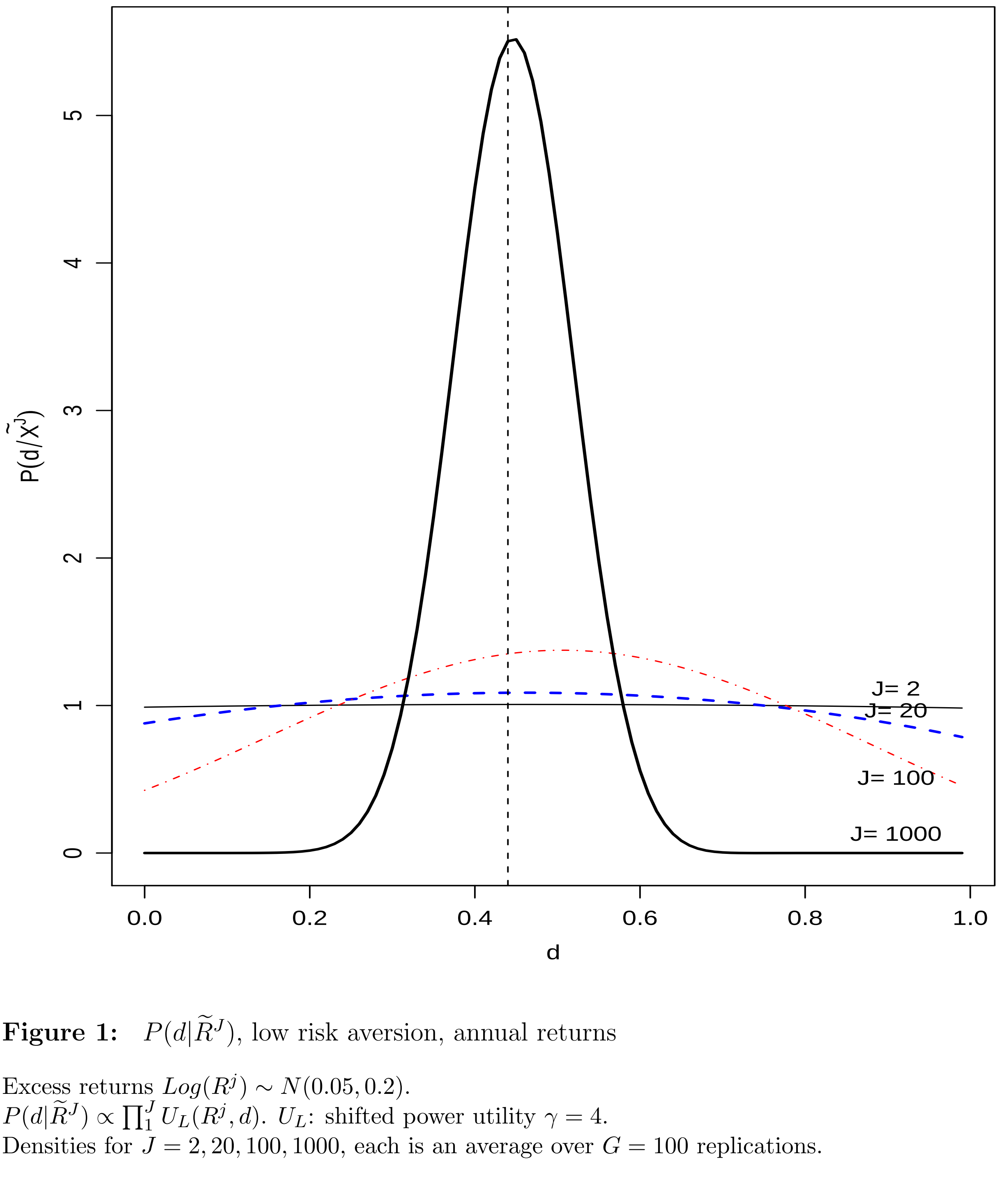

Figure 1: $P(d|\widetilde{R}^J)$, low risk aversion, annual returns

Excess returns $Log(R^j) \sim N(0.05, 0.2)$.
$P(d|\widetilde{R}^J) \propto \prod_1^J U_L(R^j, d)$. U_L: shifted power utility $\gamma = 4$.
Densities for $J = 2, 20, 100, 1000$, each is an average over $G = 100$ replications.

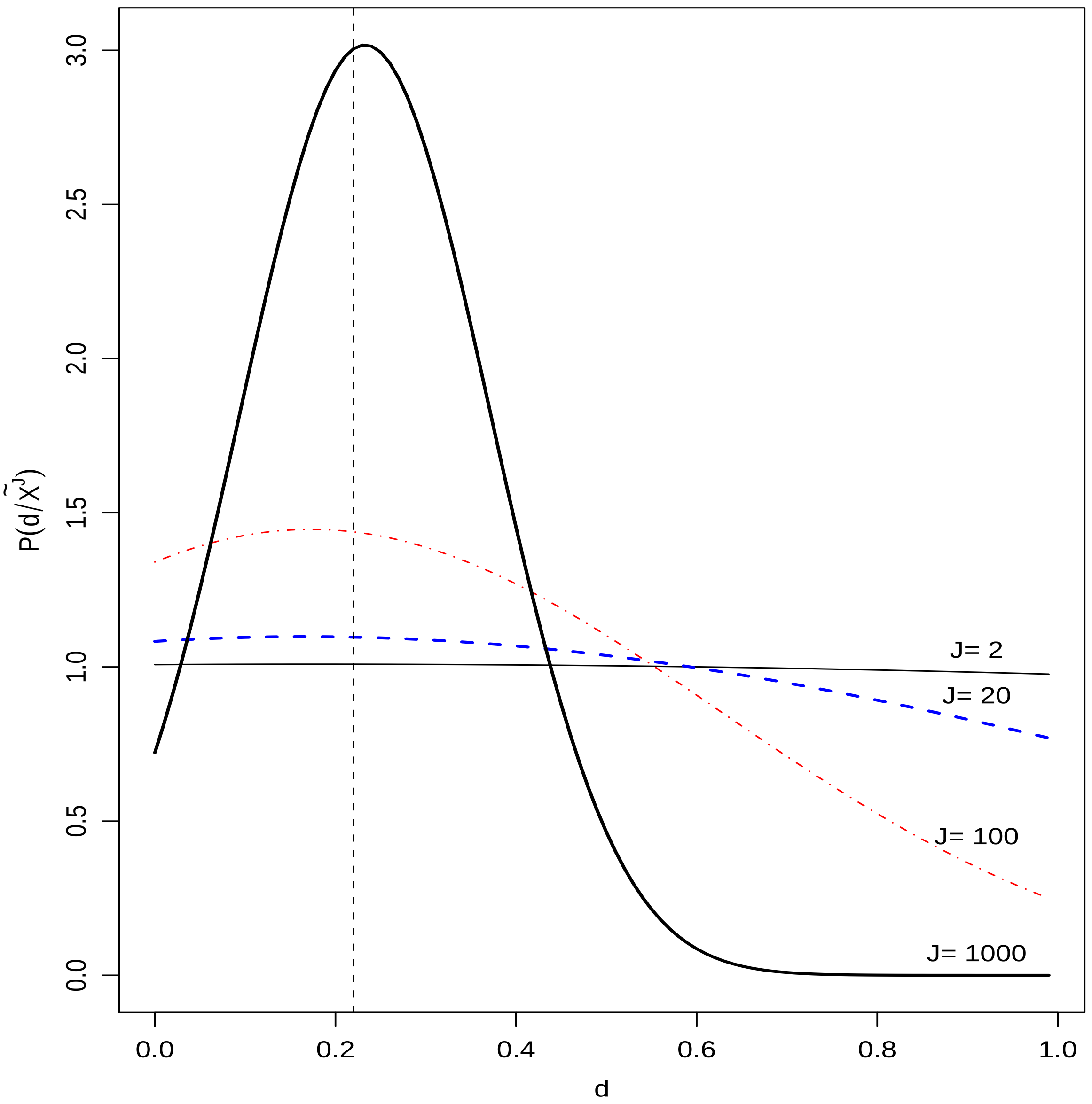

Figure 2: $P(d|\widetilde{R}^J)$, high risk aversion, monthly returns

Monthly excess returns $Log(R^j) \sim N(0.05, 0.2)$ per year.
$P(d|\widetilde{R}^J) \propto \prod_1^J U_L(R^j, d)$. U_L: shifted power utility, $\gamma = 8$.
Densities for $J = 2, 20, 100, 1000$, each density is an average over $G = 100$ replications.

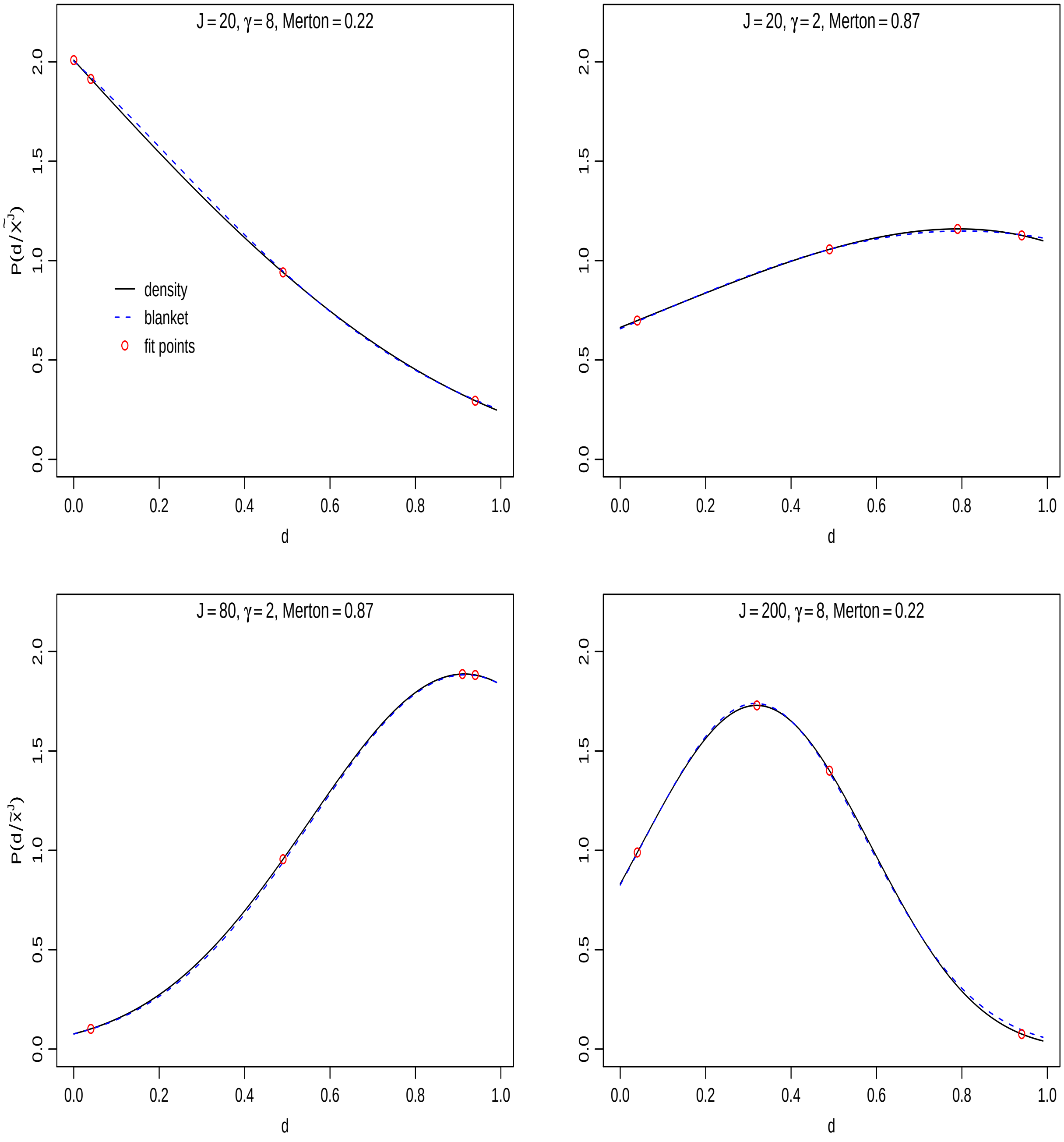

Figure 3: Truncated normal blanket for $P(d|\widetilde{R}^{j,(g)})$, monthly returns

Monthly excess returns $Log(R^j) \sim N(0.05, 0.2)$ per year.
$P(d|\widetilde{R}^J) \propto \prod_1^J U_L(R^j, d)$. U_L: shifted power utility.
Each plot shows a density conditional on one draw $\widetilde{R}^{j,(g)}$, of $\widetilde{R}^J$.

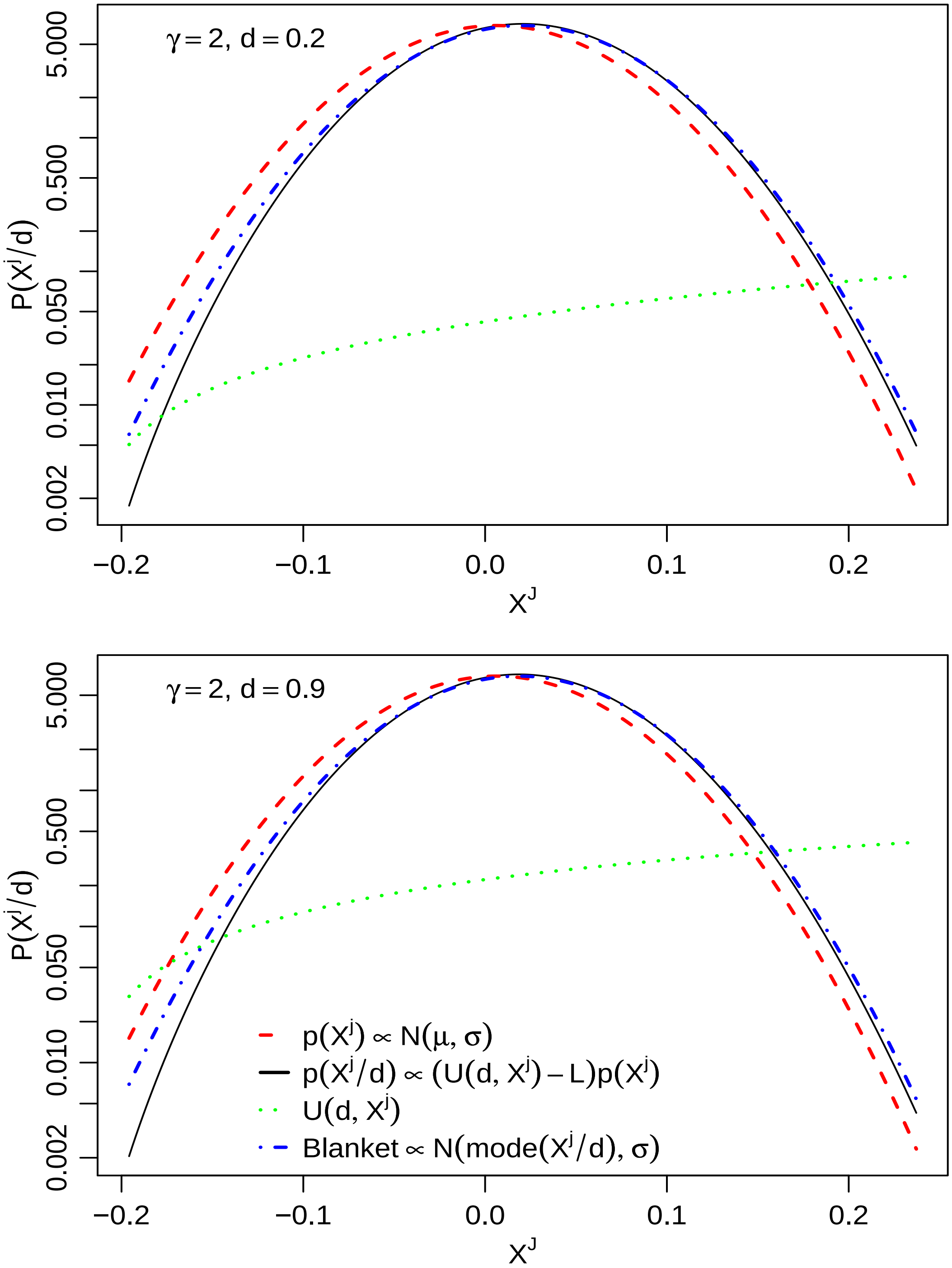

Figure 4: Normal blanket for $P(R^j|d^{(g)})$

Monthly excess returns $log(R^j) \sim N(0.05, 0.2)$ per year.
$P(R^j|d) \propto U_L(R^j, d)p(R^j)$. U_L: shifted power utility.
Each plot shows a density conditional on one draw $d^{(g)}$, of d, with $log(R^j)$ on the horizontal axis.