Markov chain Monte Carlo methods for Decision Analysis

Concha Bielza[1], Peter Müller[2], David Ríos Insua[1]
[1]Decision Analysis Group, Madrid Technical University, Spain
[2]Institute of Statistics and Decision Sciences, Duke University

Abstract

We propose a simulation based strategy for approximating optimal decisions in a Decision Analysis by defining an apropriate augmented probability model. We show that the optimal alternative is the mode of the marginal on the alternatives of the distribution associated to that introduced probability model. After drawing a sample from this distribution, we may use exploratory data analysis tools to approximately identify the optimal alternative. We specify our method for some important types of influence diagrams.

KEYWORDS: Decision Analysis, Simulation, Markov chain Monte Carlo, Influence Diagrams.

1 Introduction

We propose a simulation based approach for solving complex Decision Analysis problems involving realistic statistical models and complicated utility functions. The proposed methods are based on recent developments in posterior simulation. For that, we introduce several Markov chain Monte Carlo (MCMC) algorithms, see Smith and Roberts (1993) for a review of MCMC.

The method starts by considering an artificial distribution on the space of alternatives and states. The density of this distribution is proportional to the product of utility and probability, exploiting the duality between both concepts, as suggested in Girón (1979). The distribution is defined in such a way that its marginal on the space of alternatives is proportional to the expected utility of the alternative and, consequently, the optimal alternative coincides with the mode of the marginal. The proposed simulation based strategy follows these steps: (i) draw a sample from the artificial distribution; (ii) marginalise it to the space of alternatives; and, (iii) find the mode of the sample as a way of approximating the optimal alternative.

Section 2 describes the method. Sections 3 and 4 are of more technical nature and provide generic methods to sample approximately from the artificial distribution and identify the mode of the sample. Section 5 illustrates the algorithms with an example from water resources decision making.

2 The Approach

Here we outline the method. Given the enormous interest in influence diagrams (ID) as a tool for structuring and solving decision problems, see e.g. Matzkevich and Abramson (1995), we concentrate on such structures. For notational purposes and algorithmic reasons, we differentiate five types of nodes: decision nodes d, the value node v representing the utility function u, and three types of chance nodes: random variables x observed prior to making the decision; unobservable random variables θ, i.e. unknown parameters in the probability model; and not yet observed random variables y, i.e. data which will be observed after making the decisions. Figure 1 provides a simple generic influence diagram for our scheme, see Shachter (1986) for a complete description and an algorithm to solve IDs.

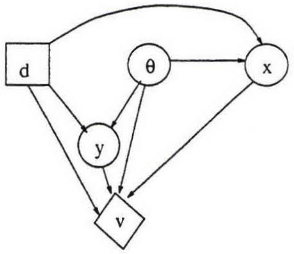


Figure 1: A generic influence diagram for our scheme.

The method we propose here is applicable to IDs with non-sequential structure, i.e. decision nodes must not have any chance nodes as predecessors which have distributions depending in turn on other decision nodes. Except for some technical conditions there will be no further requirements. In particular, the involved probability distributions can be continuous and they do not need to be in any conjugate framework. Also the decision spaces can be continuous and the utility function arbitrary. See Shachter and Kenley (1989) for comments on difficulties concerning the solution of arbitrary continuous IDs.

If a distribution depends on some index a which is not necessarily defined as a random variable, i.e. no probability model is defined for a, then we write $p_a(x)$ and $p_a(x|y)$. We will write $u(d,x,\theta,y)$ for the utility function defined in the value node of the ID. We assume that the utility function is integrable and positive. These assumptions are reasonable in most practical decision problems, for example, if the utility function and the decision space are bounded.

The ID defines the conditional distributions $p(x|\theta)$, $p(\theta)$ and $p_d(y|\theta)$, a joint distribution on the random variables (x,θ,y) via $p_d(x,\theta,y) = p(\theta)p(x|\theta)p_d(y|\theta)$, and a conditional distribution

$$p_d(\theta, y|x) \propto p(\theta)p(x|\theta)p_d(y|\theta), \tag{1}$$

for (θ, y) given the observed nodes x. Typically, x and y are independent given θ allowing the above factorization, and $p(\theta)$ does not depend on d. If a particular problem does not fit this setup, modifications of the proposed algorithm are straightforward.

Solving the ID amounts to solving

$$\max_d \left[V(d) = \int u(d,x,\theta,y) dp_d(\theta,y|x) \right]. \tag{2}$$

When problem (2) is structurally complicated, say a heavily asymmetric and dense large influence diagram with continuous non-Gaussian random variables, non quadratic utility functions and/or continuous sets of alternatives at decision nodes, finding the exact solution may be intractable analytically, and we may need an approximate solution method. We shall provide such an approximation based on simulation. We artificially augment the probability measure given in (1) to a probability model for (θ, y, d) by defining a joint p.d.f.

$$h(\theta, y, d) \propto u(d,x,\theta,y)p_d(\theta,y|x).$$

The mode of the marginal distribution $h(d) \propto \int\int u(d,x,\theta,y)p_d(\theta,y|x)d\theta dy = V(d)$ corresponds to the optimal decision d^*. As a consequence, we can solve problem (2) approximately with the following simulation based approach:

(a) Draw a random sample from the distribution $h(\theta,y,d)$.

(b) Convert it to a random sample from the marginal $h(d)$.

(c) Find the mode of this sample.

The key steps are (a) and (c). For (c) we rely mainly on tools from exploratory analysis, as we describe in Section 4. For (a), we shall introduce several Markov chain simulation methods. Their underlying idea is simple. We wish to generate a sample from a distribution over a certain space, but cannot do this directly. Suppose, however, that we can construct a Markov chain with the same state space, i.e. in (θ,y,d), which is straightforward to simulate from and defined to have $h(\theta,y,d)$ as asymptotic distribution. Practical convergence of the algorithms may be judged with a number of criteria, see e.g. Tanner (1994). If we simulate sufficiently many iterations, after dropping an initial transient phase, for big enough t the simulated values (θ^t,y^t,d^t) provide, approximately, a Monte Carlo sample from $h(\theta,y,d)$. We shall provide several algorithms for constructing chains with the desired equilibrium distribution, in our case the artificial distribution h, in Section 3. By considering the marginal distribution of d^t in this Monte Carlo sample, we can infer the optimal decision, possibly with methods discussed in Section 4.

3 Sampling from the artificial distribution

The key issue is the definition of a Markov chain with the desired limiting distribution $h(.)$. We describe first a generic method, and then two modifications that may further exploit specific structural properties of our generic ID. We state only general versions of the algorithms. Bielza et al (1996) discuss implementation issues and proofs of convergence to the desired distribution $h(\cdot)$. General discussions and results for convergence of MCMC methods are available in Tierney (1994) and Roberts and Smith (1994), from which we draw upon.

3.1 A Metropolis chain

We provide a general algorithm valid for all IDs satisfying the structural conditions specified above and some minor technical conditions discussed below. The algorithm is of the Metropolis type: we generate a new candidate for the states from a 'probing' distribution, and then move to that new state or stay at the old one according to certain 'accept-reject' probabilities. We do this transition in three steps, for d, θ and y.

We only require to be able to evaluate the utility function $u(d,x,\theta,y)$ and the probability distributions $p_d(y|\theta)$, $p(\theta)$, $p(x|\theta)$, for any relevant d,x,θ,y. This will typically be possible, since the definition of the ID includes explicit specification of these distributions, i.e. the modeler is likely to specify well-known distributions.

The scheme requires specification of probing distributions g_1, g_2 and g_3. Their choice is arbitrary, with the only constraint that the resulting Markov chain be irreducible and aperiodic. For technical reasons, whenever possible, we assume the probing distributions to be symmetric in their arguments, i.e., $g(a|b) = g(b|a)$. If $\theta \in R^p$ is a continuous parameter, we propose to use a normal kernel $g(\tilde{\theta}|\theta) = N(\theta,\Sigma)$ with appropriately chosen covariance matrix Σ. Good values for the step size can be found by trial and error with a few values. In a particular setup, Gelman, Roberts and Gilks

(1994) show that the optimal choice of step size should result in average acceptance probabilities around 25%. If θ is discrete, a simple choice for $g(\tilde{\theta}|\theta)$ could generate $\theta-1$ and $\theta+1$ with probability 0.5. Of course, many other problem specific choices are possible.

Should g be asymmetric, we would add a factor $g(\theta|\tilde{\theta})/g(\tilde{\theta}|\theta)$ in the expressions for the acceptance probabilities a. This would correspond to Metropolis-Hastings steps, rather than Metropolis steps.

The outline of our **Algorithm 1** is:

1. Start at values (d^0, θ^0, y^0) for decisions, parameters and outcomes. Set $i = 1$.

   Until convergence is practically judged

2. Evaluate $u^1 = u(d^{i-1}, x, \theta^{i-1}, y^{i-1})$.

3. *Update d*

   (a) Generate a "candidate" $\tilde{d} \sim g_1(\tilde{d}|d^{i-1})$.

   (b) Evaluate $\tilde{u}^1 = u(\tilde{d}, x, \theta^{i-1}, y^{i-1})$.

   (c) Compute

$$a_1(d^{i-1}, \tilde{d}) = \min\left[1, \frac{h(\tilde{d}, \theta^{i-1}, y^{i-1})}{h(d^{i-1}, \theta^{i-1}, y^{i-1})}\right] = \min\left[1, \frac{\tilde{u}^1}{u^1} \cdot \frac{p_{\tilde{d}}(y^{i-1}|\theta^{i-1})}{p_{d^{i-1}}(y^{i-1}|\theta^{i-1})}\right].$$

   (d) Set

$$d^i = \begin{cases} \tilde{d} & \text{with prob } a_1(d^{i-1}, \tilde{d}), \\ d^{i-1} & \text{else.} \end{cases}$$

   (e) Let $u^2 = u(d^i, x, \theta^{i-1}, y^{i-1})$.

4. *Update θ*

5. *Update y*

Steps 4 and 5 will generate candidates $\tilde{\theta} \sim g_2(\tilde{\theta}|\theta^{i-1})$ and $\tilde{y} \sim g_3(\tilde{y}|y^{i-1})$, obtaining similar expressions for $a_2(\theta^{i-1}, \tilde{\theta})$ and $a_3(y^{i-1}, \tilde{y})$, respectively. Then, we let $i = i + 1$. This algorithm defines a Markov chain, with $h(\theta, y, d)$ as stationary distribution, see Bielza et al (1996) for further details.

The generality of this algorithm comes at the price, of possibly slow convergence. Depending on the application, long simulation runs might be required to achieve practical convergence. However, this fully general algorithm is rarely required. Common applications will allow problem specific modifications by replacing the general Metropolis steps by alternative Markov chain Monte Carlo algorithms. If practicable, we strongly recommend to use such generally faster mixing Markov chain Monte Carlo schemes. Implementation details should depend on particular features of the problem studied. We will discuss here some typical situations.

Their main difference will be related to the distribution $p(\theta|x)$. In general, this distribution will not be explicitly specified in the ID, but needs to be computed through repeated applications of Bayes formula, or several arc reversals in the language of IDs. Simulating from $p(\theta|x)$ amounts to solving the statistical inference problem of generating from the posterior distribution on θ given the data x. Hence, we could appeal to versions of posterior simulation schemes appropriate for a variety of important inference problems recently discussed in the Bayesian literature.

3.2 Hybrid methods

We consider first simplifications derived from x missing in the ID, i.e. if no data is given. In that case, the expected utility is

$$\max_d \left[V(d) = \int u(d, \theta, y) dp_d(\theta, y) \right],$$

and we have a simpler algorithm. We only require a probing distribution g, the utility function for evaluation and algorithms to generate from $p(\theta)$ and $p_d(y|\theta)$. We state then our **Algorithm 2**, a **hybrid Metropolis/independence chain** one, in which only updating d remains as a Metropolis step. The updating of (d, θ, y) would be

$$(\tilde{d}, \tilde{\theta}, \tilde{y}) \sim g(\tilde{d}|d) p_{\tilde{d}}(\tilde{\theta}, \tilde{y}) = g(\tilde{d}|d) p(\tilde{\theta}) p_{\tilde{d}}(\tilde{y}|\tilde{\theta}), \tag{3}$$

implementing an independence chain using $p_d(\theta, y)$ as probing distribution. Sampling from $p_d(\theta, y) = p(\theta) p_d(y|\theta)$ will be feasible in general, since these distributions are defined explicitly in the ID.

In the general case, when the ID includes observed data x, we can still use a similar idea, with (3) replaced by $(\tilde{d}, \tilde{\theta}, \tilde{y}) \sim g(\tilde{d}|d) p_{\tilde{d}}(\tilde{\theta}, \tilde{y}|x) = g(\tilde{d}|d) p(\tilde{\theta}|x) p_{\tilde{d}}(\tilde{y}|\tilde{\theta})$. We then have the third algorithm, a **hybrid Metropolis/Gibbs** one. While simulating from $p_d(y|\theta)$ is typically straightforward, simulating from $p(\theta|x)$ is not. As mentioned above, recent literature has developed many Markov chain Monte Carlo algorithms for posterior simulation in a variety of important models. Before starting this algorithm, we could generate a sufficiently large Monte Carlo sample from $p(\theta|x)$ by whatever appropriate simulation method.

4 Finding the optimal solution

Using the previous computer simulation we obtain an approximate simulated sample $\{(d^1, \theta^1, y^1), ..., (d^N, \theta^N, y^N)\}$ from $h(d, \theta, y)$. From it we deduce the approximate sample $\{d^1, ..., d^N\}$ from the marginal $h(d)$. There remains the problem of using this MC sample to find the mode of $h(d)$. The inference of a mode from a sample may be dealt with several statistical tools, see Good and Gaskins (1980) and Scott (1992).

When the decision space is discrete, the problem is simple since we only have to count the number of times each element has appeared, and choose the one with the highest frequency. It may be worthwhile retaining not one but several of the most frequent decisions, and study them in further detail, as a way of conducting sensitivity analysis.

In the case of continuous alternatives, we propose to use graphical *exploratory data analysis tools*, especially with low dimensional decision vectors. When the dimension is one or two, we may just produce the histogram (or a smooth version) and inspect it to identify modes. For higher dimensional problems, we may still produce projected two dimensional histograms. For example, in four dimensions, we may display two dimensional arrays of two dimensional histograms, inspect them and identify solutions that would be worthwhile studying. Next example illustrates the approach.

Finally, we propose to consider the problem as one of cluster analysis. Modes of $h(d)$ correspond to d's with higher density, which suggest looking for regions with higher concentration of sampled d's. This suggests computing a hierarchical cluster tree for the simulated points d^t. Once we have a classification tree, we cut at a certain height and obtain the corresponding clusters. The location of the largest cluster indicates the area of the best decision. Again, as before, it may be useful to keep several larger clusters and explore the corresponding regions. The result of course would depend on the cutting height, but by exploring several heights we may be able to identify several decisions of interest. We illustrate it in the next section.

Similar ideas may be pursued in solving traditional statistical optimal design problems. From a formal point of view, a statistical optimal design problem can be described as a stochastic optimisation problem. This is explored in Clyde, Müller and Parmigiani (1995) for the special case of Algorithm 2 with continuous sample spaces and non-sequential setups.

5 Example: a water reservoir management problem

In Ríos Insua et al (1996), we describe a complex multiperiod decision analysis problem concerning the management of two reservoirs: Lake Kariba (K) and Cahora Bassa (C). Here we solve a simplified version with Algorithm 2. We illustrate methods to find out the optimal alternative, when the dimension of the decision space is big.

We want to find, for a given month, optimal values to be announced for release from K and C through turbines and spillgates, $u_1^k, u_2^k, u_1^c, u_2^c$, respectively. The actual amounts of water released depend on the water available, which is uncertain, since there is uncertainty about the inflows i^k and i^c to the reservoirs. There is a forecasting model for both i^k and i^c, the latter being dependent on the water released from K and the incremental inflows, which, in turn, depend on a parameter β. The preference model combines utilities for both K and C. Those for K depend on the energy deficit (def), the final storage (stor$_k$) and the amount of water spilled (spi). Those for C depend on the energy produced (ene) and the final storage (stor$_c$). Initial storages s^k and s^c have influence as well over actual releases. Figure 2 shows the ID representing the problem. Nodes with double border are either known values or deterministic functions of their predecessors. They are required to compute the value node v, but will not show up in the probability model.

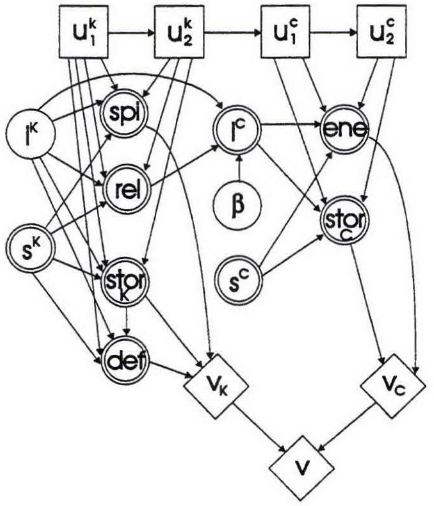
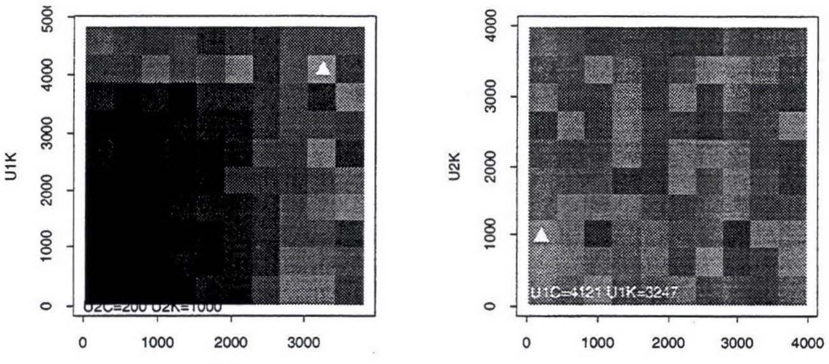


Figure 2: Influence diagram for the reservoir problem.

In terms of our notation, the problem includes four decision nodes $d = (u_1^k, u_2^k, u_1^c, u_2^c)$ and two chance nodes i^k and β .

Figure 3 shows some profiles of the histogram of the simulated $d^t \sim h(d)$, generated by Algorithm 2. The decision parameter is four dimensional. Hence we used a four dimensional grid (with

10 × 10 × 10 × 10 cells) to record a four dimensional histogram of the simulated states.

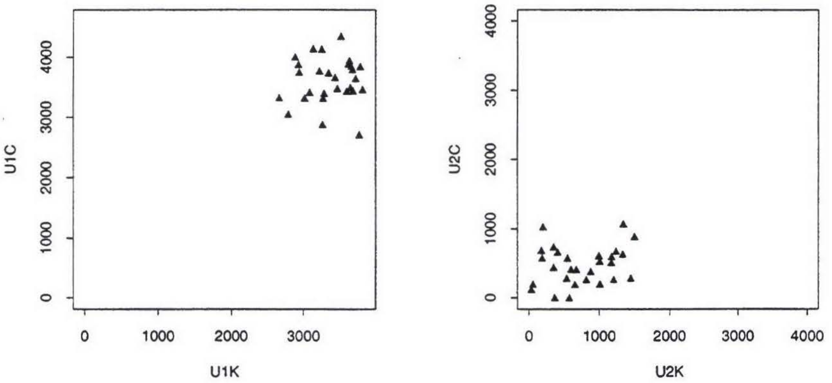


(a) $f(u_1^k, u_1^c) = h(u_1^k, u_2^{k*}, u_1^c, u_2^{c*})$ (b) $f(u_2^k, u_2^c) = h(u_1^{k*}, u_2^k, u_1^{c*}, u_2^c)$

Figure 3: Expected utility as a function of release for energy (u_1^k, u_1^c) with spill fixed at the optimal levels (a); and as a function of spill (u_2^k, u_2^c) with release through turbines fixed at the optima (b).

Simple inspection of the empirical distribution allows to read off the optimal release at $d^* = \{u_1^{k*} = 3247, u_2^{k*} = 1000, u_1^{c*} = 4121, u_2^{c*} = 200\}$. The solution is based on 100,000 simulated values from the Markov chain Monte Carlo scheme. The figure illustrates also another feature of our method which is a simple sensitivity analysis procedure at no extra cost. Darkness of Figure 3 b) suggests that expected utility is rather flat when releases through turbines are fixed at their optimal values, hence suggesting insensitivity with respect to changes in spill. On the other hand, Figure 3 a), with just one dark area where the estimated optimum is, suggests that expected utility is fairly peaked in release through turbines, and hence very sensitive to changes in energy releases.

Alternatively, as discussed in Section 4, we consider a hierarchical cluster tree of the simulation output. Figure 4 shows the solution based on cutting a hierarchical cluster tree of 1000 simulated values $d \sim h(d)$ at height 2000 and finding the cluster with the most members. The optimum is found at $d^* = \{u_1^{k*} = 3353, u_2^{k*} = 742, u_1^{c*} = 3616, u_2^{c*} = 476\}$. Comparing with Figure 3 this comes reasonably close to the optimum estimated earlier.



(a) (u_1^k, u_1^c) (b) (u_2^k, u_2^c)

Figure 4: Representation of optimal cluster for the reservoir problem.

References

Bielza, C., Müller, P., and Ríos Insua, D. (1996) Monte Carlo methods for Decision Analysis with applications to influence diagrams, DP 96-07, Duke University, *ISDS Paper.*

Clyde, M., Müller, P., Parmigiani, G. (1995) Exploring expected utility surfaces by Markov changes, *DP 95-39*, ISDS Duke University.

Gelman, A., Roberts, G. and Gilks, W. (1994) Efficient Metropolis jumping rules, *Tech. Rep., Univ. California.*

Girón, J. (1979) Probabilidad y utilidad: conceptos duales de la Teoría de la Decisión, *Rev. R. Acad. Ciencias*, 73, 225-230.

Good, I. and Gaskins, R. (1980) Density estimation and bump-hunting by the penalized likelihood method exemplified by the scattering and meteorite data (with discussion), *Jour. Amer. Stat. Assoc.*, 75, 42-73.

Matzkevich, I., and Abramson, B. (1995) Decision analytic networks in Artificial Intelligence, *Management Science*, 41, 1-22.

Ríos Insua, D., Salewicz, K., Müller, P., Bielza, C. (1996) Bayesian methods in reservoir operations: the Zambezi river case, in French, Smith (eds) *The Art of Bayesian Analysis*, Arnold.

Roberts, G. and A. Smith (1994) Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms, *Stochastic Processes and their Applications*, 49, 207-216.

Scott, D. (1992) *Multivariate Density Estimation*, Wiley.

Shachter, R. (1986) Evaluating influence diagrams. *Operations Research*, 34, 871-882.

Shachter, R. and Kenley, R. (1989) Gaussian Influence Diagrams, *Mgt. Science*, 527-550.

Smith, A., and Roberts, G. (1993) Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods, *Jour Roy. Stat. Soc., B*, 55, 3-24.

Tanner, M. (1994) *Tools for Statistical Inference*, Springer.

Tierney, L. (1994) Markov chains for exploring posterior distributions (with discussion). *Ann. Statist.* 22, 1701-1762.