f-divergences and Monte Carlo methods

Ph.D. candidate OLARIU Emanuel Florentin

Advisor Professor LUCHIAN Henri

Contents

Chapter 1

Introduction

The problem addressed by this thesis broadly concerns the use of f-divergences mainly for variance reduction in Monte Carlo (MC) integration. A f-divergence is a particular type of measure for two probability distributions. MC is a classical randomized method for solving various types of problem for which we do not know analytical solutions; it is based on generating samples from particular distributions. Hence the problem of comparing distributions comes up naturally.

By the name of one of the first who studied them, they are also called Csiszár divergences, and are generated by convex functions. More general a divergence measure is a function of two probability density (or distribution) functions, which has nonnegative values and becames zero only when the two arguments (distributions) are the same. Often, a divergence is not a symmetric function but can be easily symmetrized.

There are many techniques for reducing the variance of the MC estimator and one of these is Importance Sampling (IS). Monte Carlo, f-divergences and various directions of variance minimization for IS and MC estimators are described in more detail in Chapter 2.

We use MC method in two ways: for pricing financial derivatives known as options, and for estimation of rare-events probabilities. Both these applications are again linked with the use of f-divergences.

In Chapter 3 we develop some techniques for pricing two option styles. Spread European options are valued using IS and minimizing various divergences; this approach is compared with the least squares method for direct variance minimization. Bermudan options are priced using a modified version of MRAS algorithm, involving sampling importance resampling following the reference distributions from the standard algorithm.

The problem of estimation of rare events probabilities appears frequently in the analysis of performance of communications systems (e.g.

the probability of failure of a network system). The IS problem for this estimation consists in the increase of the frequence of rare-events by changing (to a more important) distribution. We introduce a new algorithm for such an estimation based on Rényi divergence instead of Kullback-Leibler divergence (the cross-entropy method). This algorithm with its stochastic counterpart and a version for solving continuous optimization problems are presented in Chapter 4.

The last chapter concerns the means for measuring the similarity of time series. Time series are common in various fields of science: medicine, mmultimedia, computational finance etc, and synthetic datasets are used in prediction and computer simulations.

Numerical experiments show that we can measure with great accuracy by using simple instruments like mean similarity and symmetrized divergences; these tools are easier to compute than the usual features which include common statistics, extremal points, slope and filtered data.

Chapter 2

Monte Carlo and Importance Sampling

Monte Carlo method (see [21], [27], [46]), although used in the begining for stochastic simulation only, covers today a wide range of problems which could benefit from randomness and adjacent properties. Generally speaking any technique which approaches a problem using a "large" number of random samples for various computations will take the famous name. This method is intended to solve problems for which deterministic/analytic approaches are not available, or give poor results.

This method heavily relies on computer generated numbers which are not so random being generated by deterministic mechanisms. Many techniques was developped in order to obtain unrelated (pseudo random) or low discrepancy (quasi random) sequence of numbers (see [19], [34]).

Applications of Monte Carlo method range from physical sciences, biology, medicine to weather forecasting, risk management and pricing financial derivative instruments (see [12], [17], [23]). We emphasize here a commun use in mathematics, namely Monte Carlo integration, which is, perhaps, the first technique to properly receive this name.

2.1 Monte Carlo Integration

Many problems which arise in a variety of applications can be described as the evaluation of the expected value for a given random variable. Let $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n), \mu)$ be the Lebesgue measure space. Suppose that $\mathbf{X} : \mathbb{R}^n \to \mathbb{R}$ is a random n-dimensional variable (random vector) and f is a probability density function (pdf), i.e., has the following properties:

$$f : \mathbb{R}^n \to \mathbb{R}_+, \quad \int_{\mathbb{R}^n} f(\mathrm{s})\, d\mu(\mathrm{s}) = 1. \tag{2.1}$$

Let $\mathcal{H} : \mathbb{R}^n \to \mathbb{R}_+$ be (at least) a Lebesgue measurable[1] (or, integrable) function. We want to calculate the expectation of $H(X)$:

$$m = \mathbb{E}_f\left[\mathcal{H}(\mathbf{X})\right] = \int_{\mathbb{R}^n} \mathcal{H}(\mathrm{s}) f(\mathrm{s})\, d\mu(\mathrm{s}), \tag{2.2}$$

If $\mathbb{X} = \left(X^i\right)_{1 \leqslant i \leqslant N}$ are independent and identical distributed (i.i.d.) samples from f, based on the Law of Large Numbers,

$$m_N(\mathbb{X}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{H}(X^i) \tag{2.3}$$

is an unbiased estimator of m from (2.2): $\mathbb{E}\left[m_N(\mathbf{X})\right] = m$. The variance of this estimator is

$$Var\left[m_N(\mathbb{X})\right] = \frac{1}{N} \int_{\mathbb{R}^n} \left[\mathcal{H}(\mathrm{s}) - m\right]^2 \, d\mu(\mathrm{s}) = \frac{1}{N} Var_f\left[H(\mathbf{X})\right]. \tag{2.4}$$

The above variance is a measure of Monte Carlo efficiency and becomes one of the limitations of this method; in order to halve the standard deviation of $m_N(\mathbf{X})$, you have to quadruple the number of samples. A number of techniques are used to reduce the variance, between them are Antithetic Variables, Control Variates and Importance Sampling. All these methods aim to reduce the variance without increasing the number of samples. A big part of our work concerns the Importance Sampling method for variance reduction.

[1] μ is the Lebesgue measure on $\mathbb{R}^n$.

2.2 Importance Sampling

Importance sampling is a technique for estimating a parameter of a distribution while sampling from a different distribution. This means to choose a (possibly better known, or better to simulate) distribution from which to simulate one's random variables. Let f be the original pdf and g be another pdf such as $supp(\mathcal{H} \cdot f) \subseteq supp(g)$. We can rewrite (2.2):

$$\mathbb{E}_f[\mathcal{H}(\mathbf{X})] = \int_{\mathbb{R}^n} \left[\mathcal{H}(s) \cdot \frac{f(\mathbf{s})}{g(\mathbf{s})}\right] g(\mathbf{s})\, d\mu(\mathbf{s}) = \mathbb{E}_g \left[\mathcal{H}(\mathbf{X}) \cdot \frac{f(\mathbf{X})}{g(\mathbf{X})}\right]. \tag{2.5}$$

Using (2.5) we get another unbiased estimator of m:

$$m_N(g; \mathbb{X}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{H}(X^i) \frac{f(X^i)}{g(X^i)}, \tag{2.6}$$

where $\mathbb{X} = \left(X^i\right)_{1 \leqslant i \leqslant N}$ are i.i.d. samples from g which is known as the importance sampling (IS) distribution. The idea behind Importance Sampling is to draw from another distribution (g), and then modify the result to correct the bias introduced in this way.

The variance of this estimator is

$$Var_g\left[m_N(g; \mathbb{X})\right] = \int_{\mathbb{R}^n} \left[\mathcal{H}^2(\mathbf{s}) \frac{f(\mathbf{s})}{g(\mathbf{s})} - m\right]^2 g(\mathbf{s})\, d\mu(\mathbf{s}).$$

Minimizing this variance is equivalent with

$$\min_{g \in \mathcal{G}} \mathbb{E}_f \left[\mathcal{H}^2(\mathbf{X}) \frac{f(\mathbf{X})}{g(\mathbf{X})}\right], \tag{2.7}$$

and its stochastic counterpart is

$$\min_{g \in \mathcal{G}} \sum_{j=1}^{M} \mathcal{H}^2(Y^j) \frac{f(Y^j)}{g(Y^j)}, \tag{2.8}$$

where $Y^1, Y^2, \ldots, Y^M$ are i.i.d. samples from f.

The main reason for changing the pdf is to reduce the variance by an appropriate choice of g – samples from g could be more "important" for the estimation of our integral. As long as $\mathcal{H}(t) > 0, \forall t \in \mathbb{R}^n$ the variance of this estimator is minimized when g is proportional with $\mathcal{H} \cdot f$:

$$g^*(\mathbf{s}) = \frac{\mathcal{H}(\mathbf{s})f(\mathbf{s})}{m} \tag{2.9}$$

is the zero-variance IS distribution.

This pdf is hard to determine as it depends on the desired value m; from another point of view [31] the variance of this estimator is

$$\begin{aligned} Var\left[m_N(g;\mathbb{X})\right] &= \frac{1}{N} Var_g\left[\mathcal{H}(\mathbf{X})\cdot\frac{f(\mathbf{X})}{g(\mathbf{X})}\right] = \\ &= \frac{1}{N}\mathbb{E}_g\left[\left(\mathcal{H}(\mathbf{X})\cdot\frac{f(\mathbf{X})}{g(\mathbf{X})}\right)^2\right] - \frac{1}{N}\left(\mathbb{E}_g\left[\mathcal{H}(\mathbf{X})\cdot\frac{f(\mathbf{X})}{g(\mathbf{X})}\right]\right)^2 = \\ &= \frac{1}{N}\mathbb{E}_f\left[\mathcal{H}^2(\mathbf{X})\cdot\frac{f(\mathbf{X})}{g(\mathbf{X})}\right] - \frac{m^2}{N}, \end{aligned} \tag{2.10}$$

therefore, minimizing the variance is equivalent with solving the following problem

$$\min_{g\in\mathcal{G}} \mathbb{E}_f\left[\mathcal{H}^2(\mathbf{X})\cdot\frac{f(\mathbf{X})}{g(\mathbf{X})}\right] \tag{2.11}$$

over the entire set of pdf's $\mathcal{G}$.

A more practical approach is to search for an IS distribution from a parametric family $(g_\theta)_\theta$ which has to minimize a certain divergence with respect to the zero-variance pdf g^* (see [4] and section 2.3). In many situations this search is refined and we look from distributions of the form (see [4], [22], [31] and [44]):

$$g(\mathbf{s}) = g_1(s_1)\cdot g_2(s)\cdot\ldots\cdot g_n(s_n), \mathbf{s} = (s_1, s_2, \ldots, s_n), \tag{2.12}$$

i.e., the multi-dimensional distributions having independent components. The (mean) parametrized distributions usually make part of the so-called natural exponential families ([13], [33]) which has a pdf of the form

$$f_\theta(\mathbf{s}) = \varphi(\mathbf{s})\exp\left(\langle\theta,\mathbf{s}\rangle - \psi(\theta)\right), \tag{2.13}$$

where $\mathbf{s}\in\mathbb{R}^n$, $\theta\in\mathbb{R}^n$ is the parameter of the family, and φ and ψ are known functions. We can include here the following distributions: normal with known covariance, Poisson, gamma with known shape parameter.

2.3 f-divergences and IS

Let $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n), \mu)$ the Lebesgue measure space (μ is a σ-finite measure), μ_1, μ_2 two probability measures on $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$ absolutely continuous with respect to μ ($\mu(M) = 0$ implies $\mu_i(M) = 0$). Denote by $p_i = \dfrac{d\mu_i}{d\mu}$ ($i = \overline{1,2}$), the corresponding Radon-Nikodym derivatives with respect to μ. If $f : \mathbb{R}_+ \to \mathbb{R}$ is a convex function, continuous in 0, the f-divergence of μ_1 and μ_2 (or p_1 and p_2) (cf. [2], [11] and [16]) is

$$\mathcal{D}_f(\mu_1, \mu_2) = \mathcal{D}_f(p_1, p_2) = \int_{\mathbb{R}^n} p_2(\mathrm{s}) f\left[\frac{p_1(\mathrm{s})}{p_2(\mathrm{s})}\right] \, d\mu(\mathrm{s}). \tag{2.14}$$

A few examples of particular interest:

- $f : \mathbb{R}_+ \to \mathbb{R}$, $f(x) = (x-1)^2$ gives the Karl Pearson's χ^2-divergence:
$$\mathcal{D}_{\chi^2}(\mu_1, \mu_2) = \int_{\mathbb{R}^n} \frac{[p_1(\mathrm{s}) - p_2(\mathrm{s})]^2}{p_2(\mathrm{s})} \, d\mu(\mathrm{s})$$
- $f : \mathbb{R}_+ \to \mathbb{R}$, $f(x) = |x-1|$ gives the total variance distance:
$$V(\mu_1, \mu_2) = \int_{\mathbb{R}^n} |p_1(\mathrm{s}) - p_2(\mathrm{s})| \, d\mu(\mathrm{s}) \left(= \sup_{M \in \mathcal{B}(\mathbb{R}^n)} [\mu_1(M) - \mu_2(M)]\right)$$
- $f : \mathbb{R}_+^* \to \mathbb{R}$, $f(x) = x \log x$ gives the Kullback-Leibler divergence:
$$\mathcal{D}_{KL}(\mu_1, \mu_2) = \int_{\mathbb{R}^n} p_1(\mathrm{s}) \log\left[\frac{p_1(\mathrm{s})}{p_2(\mathrm{s})}\right] \, d\mu(\mathrm{s})$$
- $f : \mathbb{R}_+^* \to \mathbb{R}$, $f(x) = x^\alpha$ gives the α-order Rényi divergence (or Rényi entropy):
$$\mathcal{D}_\alpha(\mu_1, \mu_2) = \int_{\mathbb{R}^n} p_1^\alpha(\mathrm{s}) p_2^{1-\alpha}(\mathrm{s}) \, d\mu(\mathrm{s})$$

For all these divergences, $I_f(\mu_1, \mu_2) \geqslant 0$, and $I_f(\mu_1, \mu_2) = 0$ if and only if $\mu_1 = \mu_2$, i.e., $p_1 = p_2$ almost everywhere (a.e.).

As we already mention on section 2.2, instead of solving the problem (2.11) we can try to minimize a given f-divergence with respect to the zero-variance pdf g^*, that is

$$\min_{g \in \mathcal{G}} \mathcal{D}_f(g^*, g) \tag{2.15}$$

Although sometimes we can change the order of the pdf's - remember that these divergences are not all symmetric.

2.3.1 Kullback-Leibler divergence

For Kullback-Leibler divergence we get a problem (see [4], [26] and [45]) which gives the well known method named cross-entropy:

$$\min_{g \in \mathcal{G}} \mathcal{D}_{KL}(g^*, g) = \min_{g \in \mathcal{G}} \int_{\mathbb{R}^n} g^*(\mathbf{s}) \log \left[\frac{g^*(\mathbf{s})}{g(\mathbf{s})} \right] d\mu(\mathbf{s}), \tag{2.16}$$

or

$$\min_{g \in \mathcal{G}} \int_{\mathbb{R}^n} \left[g^*(\mathbf{s}) \log g^*(\mathbf{s}) - g^*(\mathbf{s}) \log g(\mathbf{s}) \right] d\mu(\mathbf{s}),$$

which is equivalent with

$$\min_{g \in \mathcal{G}} \int_{\mathbb{R}^n} \mathcal{H}(\mathbf{s}) f(\mathbf{s}) \log g(\mathbf{s}) \, d\mu(\mathbf{s}).$$

An IS distribution is a solution to the following optimization problem

$$\arg\max_{g \in \mathcal{G}} \mathbb{E}_f \left[\mathcal{H}(\mathbf{X}) \log g(\mathbf{X}) \right] \tag{2.17}$$

2.3.2 Pearson divergence

The (reversed-) Pearson divergence can be calculated like this

$$\mathcal{D}_{\chi^2}(\mu_1, \mu_2) = \int_{\mathbb{R}^n} \left(\frac{p_1^2(\mathbf{s})}{p_2(\mathbf{s})} - 2p_1(\mathbf{s}) + p_2(\mathbf{s}) \right) d\mu(\mathbf{s}) = \int_{\mathbb{R}^n} \frac{p_1^2(\mathbf{s})}{p_2(\mathbf{s})} \, d\mu(\mathbf{s}) - 1 = \mathbb{E}_{p_1} \left[\frac{p_1(\mathbf{X})}{p_2(\mathbf{X})} \right] - 1$$

In order to find an IS distribution function with respect to this divergence we have to solve the following problem:

$$\arg\min_{g \in \mathcal{G}} \mathcal{D}_{\chi^2}(g^*, g) = \arg\min_{g \in \mathcal{G}} \mathbb{E}_f \left[\frac{\mathcal{H}^2(\mathbf{X}) f(\mathbf{X})}{g(\mathbf{X})} \right] \tag{2.18}$$

2.3.3 Rényi divergence

Relative information of order α or α-order Rényi divergence is:

$$\mathcal{D}_\alpha(p_1, p_2) = \frac{1}{\alpha - 1} \ln \left[\int_{\mathbb{R}^n} p_1^\alpha(\mathbf{s}) p_2^{1-\alpha}(\mathbf{s}) \, \mu(d\mathbf{s}) \right]$$

Therefore, for $\alpha > 1$, and reversing the order of pdf's, we have the following optimization problem

$$\arg\min_{g \in \mathcal{G}} \mathcal{D}_\alpha(g, g^*) = \arg\min_{g \in \mathcal{G}} \left[\int_{\mathbb{R}^n} g^\alpha(\mathbf{s}) \mathcal{H}^{1-\alpha}(\mathbf{s}) f^{1-\alpha}(\mathbf{s}) \, \mu(d\mathbf{s}) \right],$$

or

$$\arg\min_{g \in \mathcal{G}} \mathbb{E}_f \left[\frac{g^\alpha(\mathbf{X})}{f^\alpha(\mathbf{X})} \mathcal{H}^{1-\alpha}(\mathbf{X}) \right] \tag{2.19}$$

2.4 Sampling Importance Resampling

The sampling importance resampling (SIR) method (see [36]) draws a random sample from a distribution with pdf h in two steps. First an independent random sample $\mathbb{X} = (X_1, X_2, \ldots . X_n)$ is drawn from a distribution with pdf g, then a (usually, smaller) sample $\mathbb{Y} = (Y_1, Y_2, \ldots, Y_m)$ is drawn (often with replacement) from $\mathbb{X}$ with sample probabilities $w(X_i)$, proportional with $h(X_i)/g(X_i)$. In practice

$$w(X_i) = \frac{h(X_i)/g(X_i)}{\sum_{j=1}^{n} h(X_j)/g(X_j)} \tag{2.20}$$

We generate the new samples using a multinomial distribution with these weights. That is, from $\mathbb{X}$, we give more importance to h. We shall see the value of this technique in the following chapters.

Chapter 3

Option pricing. Importance sampling and f-divergences

In this chapter we develop some techniques used mainly for pricing financial instruments, but useful in a more general framewok. We involve here the importance sampling techniques and minimizing f-divergences.

There are many financial instruments for which closed form formulae cannot be derived from the existing mathematical models. One example of such model is the classical result of Black and Scholes which cover only a small part of the entire spectra of derivatives, especially for multivariate contingent claims.

In the following sections we study two types of option pricing: spread and a variety of American option known as Bermudan option. An option is a derivative instrument - because do not depend directly on the price of an asset (commodities, stocks, currencies or financial indexes).

An option is a contract between two parts (a seller and a buyer) in which, one - the buyer - buys the right to engage in a transaction concerning the asset (at a future date), from the second - the seller. The buyer has the right, but not the obligation to fulfill the above transaction, while the seller has the obligation to engage the transaction if the first party agrees with that. Therefore an option contract can be exercised or not at the convened moment(s) in time.

Depending on the transaction involved, there are to types of options: if the transaction gives the right to buy the asset(s) is called call option, while, if the transaction gives the right to sell the asset(s) is called put option.

Depending on the moment when the transaction can be exercised, there are two main styles of options: European - the option can be exercised only at the expiration (maturity) date and American - the contract can be

exercised any time between the writing and the expiration date. In between those two reference types exist many others, like: Bermudan option - the buyer has the right to exercise at a designated number of times, Canary option - the buyer has the right to exercise at a designated number of times but not before a time period.

3.1 Spread options

One of these derivatives is the spread option, which are very widespread in the financial markets (equity, commodities, foreign exchange and energy markets) despite the fact that the corresponding pricing and hedging methods are not so well developed.

In this section we shall use the divergences from section 2.3 to approximate the price for call european spread options. The payoff function $\mathcal{H}(\mathbf{Z})$ is given by formula (A.5), and we have to approximate $V = \mathbb{E}[H(\mathbb{Z})]$.

We consider here the restriction of $\mathcal{G}$ at the family $\mathcal{N}_2$ of bivariate normal distributions with unit variances, constant correlation, parameterized by the means $\mathbf{v} = (v_1, v_2)$. The pdf's from this family are:

$$\varphi_{\mathbf{v}}(\mathbf{s}) = \varphi_{(v_1,v_2)}(s_1, s_2) = \varphi_0(s_1 - v_1, s_2 - v_2),$$

where

$$\varphi_0(s_1, s_2) = \frac{1}{2\pi\sqrt{1-\rho_0^2}} \exp\left[-\frac{1}{2(1-\rho_0^2)}\left(s_1^2 - 2\rho_0 s_1 s_2 + s_2^2\right)\right].$$

3.1.1 Kullback-Leibler divergence

Supposing that, $f = \varphi_u$, the problem (2.17) becomes

$$\arg\max_{\mathbf{v}} \mathbb{E}_{\varphi_u}[\mathcal{H}(\mathbf{Z}) \ln \varphi_{\mathbf{v}}(\mathbf{Z})] \tag{3.1}$$

While its stochastic version is

$$\arg\max_{\mathbf{v}} \sum_{i=1}^{N} \mathcal{H}(\mathbf{Z}^i) \ln \varphi_{\mathbf{v}}(\mathbf{Z}^i), \tag{3.2}$$

where $\mathbf{Z}^1, \mathbf{Z}^2, \ldots, \mathbf{Z}^N$ a i.i.d. samples from $\varphi_{\mathbf{u}}$. After some algebra, problem (3.2) has the following form

$$\arg\min_{\mathbf{v}} \sum_{i=1}^{N} \mathcal{H}(\mathbf{Z}^i) \left[(Z_1^i - v_1)^2 - 2\rho_0(Z_1^i - v_1)(Z_2^i - v_2) + (Z_2^i - v_2)\right], \tag{3.3}$$

As the function to optimize in (3.3) is a quadratic polynomial in v_1 and v_2, the problem could be solved directly. Alternatively, for this simulation, one can use again the importance sampling, and the optimal solution can be estimated by using an adaptive procedure similar to that indicated in the cross-entropy method (see [4]).

3.1.2 Pearson divergence

Problem (2.18) is transformed in

$$\arg\min_{\mathbf{v}} \mathbb{E}_{\varphi_{\mathbf{u}}}\left[\frac{\mathcal{H}^2(\mathbf{Z})\varphi_{\mathbf{u}}(\mathbf{Z})}{\varphi_{\mathbf{v}}(\mathbf{Z})}\right] \tag{3.4}$$

Using a Monte Carlo estimator for the above expectation, we get the stochastic counterpart of (3.4):

$$\arg\min_{\mathbf{v}} \sum_{i=1}^{N} \frac{\mathcal{H}^2(\mathbf{Z}_i)\varphi_{\mathbf{u}}(\mathbf{Z}_i)}{\varphi_{\mathbf{v}}(\mathbf{Z}_i)}, \tag{3.5}$$

where $\mathbf{Z}^1, \mathbf{Z}^2, \ldots, \mathbf{Z}^N$ is a random sample from φ_u.

3.1.3 Relative information

The problem of minimizing the relative information of order α from (2.19):

$$\arg\min_{\mathbf{v}} \mathbb{E}_{\varphi_{\mathbf{u}}}\left[\frac{\mathcal{H}^{1-\alpha}(\mathbf{X})\varphi^{\alpha}_{\mathbf{v}}(\mathbf{X})}{\varphi^{\alpha}_{\mathbf{u}}(\mathbf{X})}\right], \tag{3.6}$$

with its stochastic version:

$$\arg\min_{\mathbf{v}} \sum_{i=1}^{N} \frac{\mathcal{H}^{1-\alpha}(\mathbf{Z}^i)\varphi^{\alpha}_{\mathbf{v}}(\mathbf{Z}^i)}{\varphi^{\alpha}_{\mathbf{u}}(\mathbf{Z}^i)}, \tag{3.7}$$

where $\mathbf{Z}^1, \mathbf{Z}^2, \ldots, \mathbf{Z}^N$ is a random sample from φ_u.

3.1.4 Least squares problem and IS

We recall the nonlinear least squares problem framework ([30], [35]). Let us suppose that we have a function $p : \mathbb{R}^n \to \mathbb{R}^m$, where $m \geqslant n$, and we have to solve the problem

$$\arg\min_{\mathbf{x}} P(\mathbf{x}), \text{ where } P(\mathbf{x}) = ||p(\mathbf{x})||^2 = \sum_{i=1}^{m} p_i^2(\mathbf{x}), \tag{3.8}$$

where $p(\mathbf{x}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_m(\mathbf{x}))^T$. If all p_j are linear functions we have the *linear least squares problem*, otherwise we have a *nonlinear least squares problem*.

This is a frequently approach to approximate solutions to overdetemined systems of equations: instead of solving such a system we try to minimize the sum of squares of the errors.

Applications of this method are data fitting, regression analysis and statistics. There are some very efficient methods which address this problem; we used Levenberg-Marquardt method and Powell's Dog Leg method which was potentially more efficient but gave worse results. These methods (which are iterative) are useful each time the function to minimize is a sum of non-negative, and smooth enough (say $\mathcal{C}^2$-class) terms.

Let us recall the problem (2.8), the stochastic version for variance minimization, in our framework ($\mathcal{G} = \mathcal{N}_2$, $Y^1, Y^2, \ldots, Y^M$ i.i.d. samples from $\varphi_{\mathbf{u}}$):

$$\arg\min_{\mathbf{v}} \sum_{j=1}^{M} \mathcal{H}^2(Y^j) \frac{\varphi_{\mathbf{u}}(Y^j)}{\varphi_{\mathbf{v}}(Y^j)} =$$

$$\arg\min_{\mathbf{v}} \frac{1}{2} \sum_{j=1}^{M} \left(\mathcal{H}(Y^j) \sqrt{\frac{\varphi_{\mathbf{u}}(Y^j)}{\varphi_{\mathbf{v}}(Y^j)}} \right)^2 . \tag{3.9}$$

Since $\varphi_{\mathbf{v}}(\mathbf{s}) \geqslant 0$, $\forall\, \mathbf{s} \in \mathbb{R}^n$ and $\forall\, \mathbf{v} \in \mathbb{R}^2$, the problem (3.9) is a least squares problem of form (3.8). Therefore we can solve it using one of the already mentioned iterative methods.

In conclusion, the least squares method can be used in two different ways for solving stochastic problems we already formulated. First, to minimize directly the variance and find an IS optimal distribution. On the other hand, we can use the divergences to find the IS distribution. Kulbak-Leibler divergence can be used directly, but, since $\mathcal{H}(\mathbf{s}) \geqslant 0$, $\forall\, \mathbf{s} \in \mathbb{R}^n$ the problems (3.5) and (3.7) have also the form of a least squares problem (3.8), hence, for Pearson divergence and relative information we can involve again the least squares method.

It is worth noting here that Kullback-Leibler divergence has a big advantage over many other divergences, at least when using distributions from a natural exponential family. That is because the logarithm involved in its definion cancels the exponential function of the corresponding pdfs. In this respect using Kullback-Leibler divergence conducts to exact methods, while for other divergences, as we already seen, we have to solve one more (stochastic) optimization problem.

3.1.5 Numerical results

Our numerical experiments (see [37]) address all four importance sampling variants (3.3) - KL-IS, (3.5) - χ^2-IS, (3.7) - RI-IS, and (3.9) - LSq-IS.

The tests was conducted for an European call spread option with the following characteristics: spot prices $S_1^0 = 105$ and $S_2^0 = 110$, dividend yields $q_1 = 2\%$ and $q_2 = 3\%$, volatilities $\sigma_1 = 15\%$ and $\sigma_2 = 10\%$, and interest rate $r = 5\%$.

The parameter of the distributions which is changed is the pair (μ_1, μ_2), i.e., the means of the two joint gaussian variables (Z_1, Z_2), ρ-correlated. We used as an estimate of the efficiency the variance ratio (which is a ratio between the estimated variances over $1,000,000$ simulated paths):

$$\frac{\Sigma^2_{MC}}{\Sigma^2_{IS}}$$

Table 6.4 shows the results for various levels of correlation coefficient ρ and strike K (the larger the results the bettter the method). Every value is obtained as an average of 30 samples; in paranthesis are the standard errors of the mean.

Least Squares method (LSq-IS) and Kulback-Leibler (KL-IS) are the best methods. It is worth noting that, although, LSq-IS works better most of the time, KL-IS performs better for deep in the money spread options.

KL-IS has always at least five times better (than crude Monte Carlo) variance, making this method a more reliable one. LSq-IS has its worst results for deeply correlated assets, while KL-IS is less dependent on the correlation level.

Overall, χ^2-IS is the worst of the first three methods, but still has better performance than LSq-IS for deep in the money and positively correlated options. The RI-IS method needs further investigations concerning the fine tuning of parameter α - which is 1.5 in these experiments. This method gives the best results on the levels of low correlation.

3.2 American options

This section is motivated by the problem of pricing American option - a very challenging financial instrument from mathematical point of view. American options can be exercised before (but not after) maturity time - the problem of pricing such options is more delicate since, in addition to estimate its value, one has to find first the optimal exercise time. We

restrict our study to the Bermudan options - a style of American options which can be exercised only at a finite number of times.

We improve (see [38], [39]) a method successfully used for solving optimization problem named Model Reference Adaptive Search (MRAS [22]) which is an approach similar to the cross-entropy method (see [4], [45]). Such stochastic optimization methods involve two iterative stages:

- generate data samples using a specific random procedure, most likely a distribution with known parameters.
- update the parameters for the random procedure using the data from previous step.

The calculus of parameters in the second step often involves random variable expectation which are estimated (as we already seen), by Monte Carlo simulation. Although MRAS itself is a form of importance samplng, we use here importance sampling in the form of sampling importance resampling.

That is, from first step generated samples we resample (with replacement) using a multinomial distribution having probabilities proportional with their importance ratios. In this way we give more importance to samples which shift towards another distribution; the main distributions we chose for resampling are those of reference in the original algorithm.

3.2.1 Pricing Bermudan options

Recall, from Appendix B, the value of a put Bermudan option with early exercise thresholds $\mathbf{S} = (S_i^*)_{1 \leqslant i \leqslant n}$

$$\mathcal{H}(\mathbf{S}) = \max_{\mathbf{S}} \mathbb{E}\left[\mathcal{V}_{p,\mathbf{S}}\right], \tag{3.10}$$

where $\mathcal{V}_{p,\mathbf{S}}$ is given by (B.3).

The vector of threshold prices must be generated from a multivariate gaussian distribution truncated on one of the above polytopes (B.7); the pdf for such a distribution (with certain mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$) is

$$\varphi_{\mathbb{X}}\left(\mathbf{s}; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) = \frac{\mathbb{1}_{[\mathbb{X}]}(\mathbf{s})}{\left(\sqrt{2\pi}\right)^n \cdot \sqrt{|\boldsymbol{\Sigma}|}} \exp\left[-\frac{1}{2}(\mathbf{s}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{s}-\boldsymbol{\mu})\right].$$

As the threshold prices are the optimized arguments in the algorithm, a fast and quality sampling procedure is crucial for the accuracy of our results. Therefore we avoid the accept-reject method for this truncated distribution and we used a Gibbs sampler combined with a Metropolis Chain (see [14], [24], [50]).

After all these prices are determined, the values of the desired option can be calculated by estimating the expectation of the value function (B.3) (this is done by a forward simulation - knowing the equation which models the price dynamics).

3.2.2 The algorithm

In this section we analyze only the *put options*; the problem is to find

$$\mathbf{S}^* = \arg\max_{\mathbf{S}} \mathcal{H}(\mathbf{S})$$

MRAS is an iterative procedure which, in our case, generates candidate solutions following the normal distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and updates the parameters of the distribution using:

$$\boldsymbol{\mu}_{t+1} = \frac{\mathbb{E}_t\left[\frac{s\left[\mathcal{H}(\mathbf{S})\right]^t}{\varphi_{\mathbb{X}_p}(\mathbf{S};\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)}\mathbb{1}_{\left[\mathcal{H}(\mathbf{S})\geqslant\overline{\gamma}_{t+1}\right]}\cdot\mathbf{S}\right]}{\mathbb{E}_t\left[\frac{s\left[\mathcal{H}(\mathbf{S})\right]^t}{\varphi_{\mathbb{X}_p}(\mathbf{S};\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)}\mathbb{1}_{\left[\mathcal{H}(\mathbf{S})\geqslant\overline{\gamma}_{t+1}\right]}\right]} \tag{3.11}$$

$$\boldsymbol{\Sigma}_{t+1} = \frac{\mathbb{E}_t\left[\frac{s\left[\mathcal{H}(\mathbf{S})\right]^t}{\varphi_{\mathbb{X}_p}(\mathbf{S};\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)}\mathbb{1}_{\left[\mathcal{H}(\mathbf{S})\geqslant\overline{\gamma}_{t+1}\right]}\mathcal{M}\left(\mathbf{S},\boldsymbol{\mu}_{t+1}\right)\right]}{\mathbb{E}_t\left[\frac{s\left[\mathcal{H}(\mathbf{S})\right]^t}{\varphi_{\mathbb{X}_p}(\mathbf{S};\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)}\mathbb{1}_{\left[\mathcal{H}(\mathbf{S})\geqslant\overline{\gamma}_{t+1}\right]}\right]} \tag{3.12}$$

where s is a continuous strictly increasing positive function, the expectation $\mathbb{E}_t[\cdot]$ is taken under the truncated distribution $N_{\mathbb{X}_p}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, and

$$\mathcal{M}\left(\mathbf{S},\boldsymbol{\mu}_{t+1}\right) = \left(\mathbf{S}-\boldsymbol{\mu}_{t+1}\right)\cdot\left(\mathbf{S}-\boldsymbol{\mu}_{t+1}\right)^T$$

The above expectations are estimated by Monte Carlo simulation; we sample a sequence of i.i.d. gaussian vectors $\mathbb{S} = \left(\mathbf{S}^{(t,j)}\right)_{1\leqslant j\leqslant N_t} \subseteq \mathbb{X}_p$, and, for the sake of simplicity, define the following weights:

$$w_j\left(\mathbb{S};\boldsymbol{\mu},\boldsymbol{\Sigma},\gamma\right)=\frac{\frac{s\left[\mathcal{H}\left(\mathbf{S}^{(t,j)}\right)\right]^t}{\varphi_{\mathbb{X}_p}(\mathbf{S}^{(t,j)};\boldsymbol{\mu},\boldsymbol{\Sigma})}\mathbb{1}_{\left[\mathcal{H}\left(\mathbf{S}^{(t,j)}\right)\geqslant\gamma\right]}}{\sum_{j=1}^{N_t}\frac{s\left[\mathcal{H}\left(\mathbf{S}^{(t,j)}\right)\right]^t}{\varphi_{\mathbb{X}_p}(\mathbf{S}^{(t,j)};\boldsymbol{\mu},\boldsymbol{\Sigma})}\mathbb{1}_{\left[\mathcal{H}\left(\mathbf{S}^{(t,j)}\right)\geqslant\gamma\right]}},$$

where $\boldsymbol{\mu}\in\mathbb{R}^n$, $\boldsymbol{\Sigma}\in\mathbb{R}^{n\times n}$, $\gamma\in\mathbb{R}$. The parameters for the truncated multivariate gaussian distribution are updated using the following formulas:

$$\boldsymbol{\mu}_{t+1}=\sum_{j=1}^{N_t}w_j\left(\mathbb{S};\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t,\overline{\gamma}_{t+1}\right)\cdot\mathbf{S}^{(t,j)} \tag{3.13}$$

$$\boldsymbol{\Sigma}_{t+1}=\sum_{j=1}^{N_t}w_j\left(\mathbb{S};\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t,\overline{\gamma}_{t+1}\right)\cdot\mathcal{M}\left(\mathbf{S}^{(t,j)},\boldsymbol{\mu}_{t+1}\right) \tag{3.14}$$

A smoothing coefficient is defined ($\alpha\in(0,1)$, $p\in\mathbb{N}^*$):

$$\alpha_t=\alpha-\alpha\left(1-\frac{1}{t}\right)^p.$$

And the parameters are accordingly modified:

$$\begin{aligned}\widehat{\boldsymbol{\mu}}_t&=\alpha_t\boldsymbol{\mu}_t+(1-\alpha_t)\widehat{\boldsymbol{\mu}}_{t-1}\\ \widehat{\boldsymbol{\Sigma}}_t&=\alpha_t\boldsymbol{\Sigma}_t+(1-\alpha_t)\widehat{\boldsymbol{\Sigma}}_{t-1}\end{aligned} \tag{3.15}$$

We describe a modified version of MRAS algorithm by including a sampling importance resampling phase; before calculate the sample $(1-\rho)$-quantile, we replace the actual samples using sampling importance resampling. First we can resample based on the natural weights: if $\mathbf{S}=\left(\mathbf{S}^j\right)_{1\leqslant j\leqslant N}$ is the currently generated samples we have to determine the following weights:

$$u_j=\frac{\mathcal{H}(\mathbf{S}^j)}{\sum_{i=1}^{N}\mathcal{H}(\mathbf{S}^i)} \tag{3.16}$$

and generate new samples using a multinomial distribution with these weights. That is, we give more importance to those samples which have greater payoffs.

In the following let us remember a few theoretical considerations concerning MRAS (exact version) global convergence. The parameters to update in each iteration are $(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) = \theta_t$; this merging parameter is chosen in such a way that $\varphi_{\theta_{t+1}}$, the next pdf, is "closer" to the corresponding distribution from sequence of so-called reference distribution $(g_k)_{k \geqslant 1}$, where:

$$g_{t+1}(x) = \frac{s[\mathcal{H}(x)] \cdot \mathbb{1}_{\left[\mathcal{H}(x) \geqslant \overline{\gamma}_{t+1}\right]} \cdot g_t(x)}{\mathbb{E}_{g_t}\left[s[\mathcal{H}(\mathbf{X})] \cdot \mathbb{1}_{\left[\mathcal{H}(X) \geqslant \overline{\gamma}_{t+1}\right]}\right]}, \forall\, t \geqslant 1 \tag{3.17}$$

$$g_1(x) = \frac{\mathbb{1}_{[\mathcal{H}(x) \geqslant \overline{\gamma}_1]}}{\mathbb{E}_{\theta_0}\left[\mathbb{1}_{[\mathcal{H}(X) \geqslant \overline{\gamma}_1]}\right]}. \tag{3.18}$$

Hence, for step t, a more natural resampling would be using the following weights:

$$v_j = \frac{\left[s(\mathcal{H}(\mathbf{S}^j))\right]^{t-1} \cdot \mathbb{1}_{\left[\mathcal{H}(\mathbf{S}^j) \geqslant \overline{\gamma}_t\right]}}{\sum_{i=1}^{N} \left[s(\mathcal{H}(\mathbf{S}^i))\right]^{t-1} \cdot \mathbb{1}_{\left[\mathcal{H}(\mathbf{S}^i) \geqslant \overline{\gamma}_t\right]}} \tag{3.19}$$

The algorithm follows:

Step 1. initialization: quantile level ρ_0, sample size N_0, $\boldsymbol{\mu}_0$, $\boldsymbol{\Sigma}$, sample size level λ, a limit parameter $\varepsilon > 0$, a weight $\beta \in (0, 1)$, a continuous increasing positive function $s(\cdot)$, and $t = 0$;

Step 2. the general iteration of the algorithm:

- generate i.i.d. samples

  $$\mathbb{S}^t = \mathbf{S}^{(t,1)},\ \mathbf{S}^{(t,2)},\ \ldots,\ \mathbf{S}^{(t,N_t)}$$

  from density $\widehat{\varphi}_t = \beta\varphi_t + (1-\beta)\varphi_0$, ($\varphi_t$ is the density of $N_{\mathcal{S}}\left(\widehat{\boldsymbol{\mu}}_t, \widehat{\boldsymbol{\Sigma}}_t\right)$); then, resample using (3.16) or (3.19) and get another samples:

  $$\mathbf{S}^{(t,(1))},\ \mathbf{S}^{(t,(2))},\ \ldots,\ \mathbf{S}^{(t,(N_t))}$$

- calculate the $(1-\rho_t)$ quantile, $\gamma_{t+1}(\rho_t, N_t)$, of the samples

  $$\mathcal{H}\left(\mathbf{S}^{(t,(1))}\right),\ \mathcal{H}\left(\mathbf{S}^{(t,(2))}\right),\ \ldots,\ \mathcal{H}\left(\mathbf{S}^{(t,(N_t))}\right)$$

- if ($t = 0$ or $\gamma_{t+1}(\rho_t, N_t) \geqslant \gamma_t + \varepsilon$) then

  $$\overline{\gamma}_{t+1} \leftarrow \gamma_{t+1}(\rho_t, N_t),\ \rho_{t+1} \leftarrow \rho_t,\ N_{t+1} \leftarrow N_t$$

- else, if exists

$$\overline{\rho} = \max\{\rho' \ : \ \gamma_{t+1}(\rho', N_t) \geqslant \overline{\gamma}_t + \varepsilon, \ 0 \leqslant \rho' \leqslant \rho_t\},$$

  then

$$\overline{\gamma}_{t+1} \leftarrow \gamma_{t+1}(\overline{\rho}, N_t), \ \rho_{t+1} \leftarrow \overline{\rho}, \ N_{t+1} \leftarrow N_t$$

- else

$$\overline{\gamma}_{t+1} \leftarrow \overline{\gamma}_t, \ \rho_{t+1} \leftarrow \rho_t, \ N_{t+1} \leftarrow \lambda N_t$$

- update and smooth $\boldsymbol{\mu}_{t+1}$ and $\boldsymbol{\Sigma}_{t+1}$, using (3.13), (3.14) and, respectively, (3.15)
- $t++$;

3.2.3 Models used

The techniques described above are tested on pricing bermudan options under three different models for stock price dynamics: the geometric Brownian, the normal jump diffusion, and a relatively new framework - an asymmetric double-exponentially jump diffusion model.

Geometric Brownian motion model. We say that the underlying stock price, $S(t)$, follows a geometric Brownian motion if

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t), \tag{3.20}$$

where $(W(t))_{t\geqslant 0}$ is a standard Wiener process (or Brownian motion), i.e. $W_0 \equiv 0$, $t \mapsto W(t)$ is continuous almost surely, and its increments are mutually independent and stationary ($W(t+s) - W(s) \sim N(0,t), \forall\, s > 0$).

For this model $\mu = r - \delta$, where δ is the dividend yield, and σ is the volatility – all supposed constants. Under these conditions, using Ito's lemma, equation (3.20) has the following solution:

$$S(t) = S(0) \cdot \exp\left[\left(\mu - \frac{\sigma^2}{2}\right) t + \sigma W(t)\right]$$

from where the discrete counterpart used for simulation is

$$\frac{S_{\tau+\Delta\tau}}{S_\tau} = \exp\left[\left(\mu - \frac{\sigma^2}{2}\right)\Delta\tau + \sigma\sqrt{\Delta\tau}\cdot Z\right],$$

where Z is a normal standard distributed random variable.

Merton normal jump diffusion model. From a practical point of view we know that geometric Brownian motion does not always accurately simulate the stock price behaviour. Therefore other models which allow jumps have been introduced – namely jump-diffusion models ([10], [47]). Merton ([32]) proposed the following dynamic to model the underlying stock price:

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t) + S(t)dX(t), \tag{3.21}$$

$$X(t) = \sum_{i=1}^{N(t)} Y_i,$$

where $W(t)$ is a standard Wiener process, $X(t)$ is a compound Poisson process: $N(t)$ – the number of allowed jumps – is a Poisson process with parameter λ, and $Y_1, Y_2, \ldots$ is a sequence of independent and identical log-normal distributed, $LN(-\gamma^2/2, \gamma^2)$, random variables; here λ is the frequency and γ is the volatility of the jumps.

The discrete form of equation (3.21) is:

$$\frac{S_{\tau+\Delta\tau}}{S_\tau} = \exp\left[\left(\mu - \frac{\sigma^2}{2}\right)\cdot\Delta\tau + \sigma\sqrt{\Delta\tau}\cdot Z_0 + \sum_{i=1}^{N(\Delta\tau)}\left(\gamma Z_i - \frac{\gamma^2}{2}\right)\right],$$

where $Z_0, Z_1, \ldots$ are i.i.d normal standard random variables, and $N(\Delta\tau)$ is Poisson distributed with parameter $\lambda\Delta\tau$.

A double exponentially jump diffusion model Kou proposed (see [25]) another jump diffusion model for the asset price, which basically differs from the above model by the distribution of jump sizes which are double exponentially:

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t) + S(t)dV(t), \tag{3.22}$$

$$V(t) = \sum_{i=1}^{N(t)} (V_i - 1),$$

where $W(t)$ is a standard Wiener process, $V(t)$ is a compound Poisson process: $N(t)$ is a Poisson process with rate λ, and $V_1, V_2, \ldots$ are independent and identical log-asymmetric double exponential distributed random variables, i. e. $Y = \log(V_i)$ has density:

$$f(x) = \begin{cases} p\cdot\alpha_1\cdot e^{-\alpha_1 x}, & x \geqslant 0 \\ (1-p)\cdot\alpha_2\cdot e^{\alpha_2 x}, & x < 0 \end{cases},$$

p and $(1-p)$ being the probabilities of up and down jumps. The parameters for this distribution are

$$\mathbb{E}[Y] = \frac{p}{\alpha_1} - \frac{1-p}{\alpha_2},$$

$$\mathbb{V}ar[Y] = p(1-p)\left(\frac{1}{\alpha_1} + \frac{1}{\alpha_2}\right)^2 + \frac{p}{\alpha_1^2} + \frac{1-p}{\alpha_2^2}.$$

The solution to the equation (3.22) is

$$S(t) = S(0) \cdot \exp\left[\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W(t)\right] \cdot \prod_{i=1}^{N(t)} V_i. \tag{3.23}$$

The discrete form of equation (3.23) is:

$$\frac{S_{\tau+\Delta\tau}}{S_\tau} = \exp\left[\left(\mu - \frac{\sigma^2}{2}\right)\cdot\Delta\tau + \sigma\sqrt{\Delta\tau}\cdot Z_0 + \sum_{i=1}^{N(\Delta\tau)} Y_i\right],$$

where Z_0 is a normal standard distributed random variable, Y_1, Y_1, ... are i.i.d asymmetric double exponentially distributed random variables (with density (3.23)), and $N(\Delta\tau)$ is Poisson distributed with parameter $\lambda\Delta\tau$.

3.2.4 Numerical experiments and conclusions

In our implementation the stopping criteria includes, besides the maximum number of steps ($N_t \geqslant N_{max}$), a minimum number of valid samples which will be used in the updating phase.

The numerical results (see also [38], [39]) are obtained in the following conditions: initial sample size is $N_0 = 200$, initial quantile level $\rho_0 = 0.5$, smoothing parameter $\alpha = 0.8$, sample size level correction $\alpha = 2$, weight parameter $\beta = 0.3$, $\varepsilon = 0.001$. The increasing positive function used is $s(x) = \exp(0.1x)$, initial mean is a vector having all components 10, and initial covariance matrix has diagonal elements 225. Option prices are obtained by using $50,000$ simulations, after the threshold prices are estimated.

Tables 6.1 to 6.3 shows the results of our simulations: the prices, the standard errors and the average number of iterations. All models are tested for various early exercise dates and different first threshold price values.

The two sampling importance versions are: the uniform (u-SIR) and the reference distributions sampling importance resampling (rd-SIR). Both perform less steps, while rd-SIR performs at most half steps, than standard algorithm.

The prices we get are slightly smaller for importance resampling, although remaining very close to those obtained with the standard MRAS procedure. In almost all cases the standard error of the mean is similar for all three algorithms; an exception is for Kou and Merton models on 6 early exercices where standard error almost doubles (remaining under 0.05).

Our algorithm performs almost twice as fast as the standard algorithm having same standard errors - this means that our method is a reliable and faster method.

Chapter 4

Rare-events probabilities and optimization

Estimation of probabilities of rare-events are very important for the performance guarantee of various systems – usually stochastic networks (e.g. in telecommunications). There are a number of randomized methods for this estimation: genetic algorithms, simulated annealing etc; among them cross-entropy method (see [4] and [44]) is one of the most successful.

This type of approach can be used not only for estimating probabilities but also for solving various combinatorial optimization problems. Some applications of this cross-entropy show the its utility for solving even hard combinatorial problems.

The general procedure we describe (see [40],[41]) does not involve any specific family of distributions, the only restriction is that the search space consists of product-form probability density functions. We discuss an algorithm for estimation of probability of rare events and a version for continuous optimization. The results of numerical experimentation with these algorithms carried in the last section support their performances.

4.1 Minimizing the Rényi divergence

Many problems which arise in a variety of applications of operations research can be described as the evaluation of the expected value for a given random variable. Areas of our interest which use such an evaluation are rare event simulation or global optimization.

Let us recall the problem (2.15) of minimizing a f-divergence with respect to the zero-variance pdf, g^*; this is a subproblem of (2.11). If we use the Lullback-Leibler divergence we get the problem (2.17).

In this section we present an alternative approach to the problem of estimating probabilities of rare events using the class of Rényi divergences of order α. As we later see this approach can be also used for solving global optimization problems.

In the remaining of this chapter we suppose that $\alpha > 1$. The restriction we impose is that the pdf's to have a product form, i.e., the distributions with independent components:

$$\mathcal{G} = \left\{ g : \mathbb{R}^n \to \mathbb{R}_+ \ : \ \int_{\mathbb{R}^n} g(\mathrm{s})\, d\mu(\mathrm{s}) = 1, g(\mathrm{s}) = \prod_{i=1}^{n} g_i(s_i), \forall \mathrm{s} \in \mathbb{R}^n \right\}$$

This constraint will allow us to study the problem of divergence minimization in more detail.

As mentioned earlier we propose to choose as IS distribution that which minimize the Rényi divergence:

$$\min_{g \in \mathcal{G}} \left(\int_{\mathbb{R}^n} \left[g^{*\alpha}(\mathrm{s}) g^{1-\alpha}(\mathrm{s}) \right] \, d\mu(\mathrm{s}) \right) =$$

$$\min_{g \in \mathcal{G}} \left(\int_{\mathbb{R}^n} \left[\mathcal{H}^{\alpha}(\mathrm{s}) f^{\alpha}(\mathrm{s}) g^{1-\alpha}(\mathrm{s}) \right] \, d\mu(\mathrm{s}) \right) \tag{4.1}$$

For a given $\varepsilon_0 \in (0,1)$, say $\varepsilon_0 = 1/2$, we define

$$U = \left\{ h : \mathbb{R}^n \to \mathbb{R}_+ \ : \ h \in L^1(\mathbb{R}) \right\},$$

$$U_0 = \left\{ h \in U \ : \ \left| \int_{\mathbb{R}} h(t)\, d\mu(t) - 1 \right| < \varepsilon_0 \right\};$$

obviously, U_0 and U_0^n are convex subsets of the Banach spaces $L^1(\mathbb{R})$ and $\left(L^1(\mathbb{R})\right)^n$, respectively. Where

$$L^1(\mathbb{R}) = \left\{ \varphi : \mathbb{R} \to \mathbb{R} \ : \ \int_{\mathbb{R}} |\varphi(s)|\, d\mu(s) < \infty \right\}$$

is the space of absolutely integrable[1] functions. This is the most relaxed framework we can use, although it is possible to restrain our study to the square integrable functions $L^2(\mathbb{R})$.

The latter space has the advantage of being reflexive, therefore its unit sphere is relative compact in the weak convergence. As our search set $\mathcal{G}$ is a convex subset of the unit set in $L^1(\mathbb{R}^n)$, we can direct the analysis towards a Weierstrass type optimization of a continuous function on a compact set.

As our functional is not proven to be continuous we focus on a different approach which is based on the convexity and critical points of the Lagrangian.

Problem (4.1) can be viewed as a functional minimization problem:

$$\min_{g\in U_0^n}\left(\int_{\mathbb{R}^n}\left[\mathcal{H}^\alpha(\mathrm{s})f^\alpha(\mathrm{s})\prod_{i=1}^n g_i^{1-\alpha}(s_i)\right]\,d\mu(\mathrm{s})\right), \tag{4.2}$$

$$\text{subject to } \int_{\mathbb{R}} g_i(s_i)\,d\mu(s_i)=1, \forall i=\overline{1,n}.$$

For the sake of simplicity, we make the following notations:

$$\Phi: U_0^n\to\mathbb{R},\ \Psi:\left(L^1(\mathbb{R})\right)^n\to\mathbb{R}^n,$$

$$\Phi(g)=\int_{\mathbb{R}^n}\left[\mathcal{H}^\alpha(\mathrm{s})f^\alpha(\mathrm{s})\prod_{i=1}^n g_i^{1-\alpha}(s_i)\right]\,d\mu(\mathrm{s}),$$

$$\Psi(g)=\left(\int_{\mathbb{R}} g_1(s_1)\,d\mu(s_1)-1,\ldots,\int_{\mathbb{R}} g_n(s_n)\,d\mu(s_n)-1\right)$$

4.1.1 Convex optimization problem.

Thus, problem (4.2) becomes

$$\min_{g\in U_0^n}\Phi(g),\ \Psi(g)=0. \tag{4.3}$$

This a convex optimization problem with constraints. We first prove the convexity of the objective function.

LEMMA 1 *For $\alpha>1$, Φ is a convex functional on U_0^n.*

[1] We use the same notations μ, for Lebesgue measures on different σ-algebras, $\mathcal{B}(\mathbb{R}^p)$.

PROOF: It will suffice to show that the function $\varphi : \mathbb{R}^n \to \mathbb{R}$, $\varphi(\mathbf{x}) = (x_1 \cdot \ldots \cdot x_n)^{1-\alpha}$ is a convex one.

To this end, let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and $t \in (0,1)$; first, from the concavity of $\ln(\cdot)$, one has:

$$\ln\left(\varphi\left[t\mathbf{x} + (1-t)\mathbf{y}\right]\right) = \ln\left(\prod_{i=1}^{n}\left[tx_i + (1-t)y_i\right]^{1-\alpha}\right) =$$

$$= (1-\alpha)\sum_{i=1}^{n}\ln\left[tx_i + (1-t)y_i\right] \leqslant (1-\alpha)\sum_{i=1}^{n}\left[t\ln x_i + (1-t)\ln y_i\right] =$$

$$= t\ln\left(\varphi(\mathbf{x})\right) + (1-t)\ln\left(\varphi(\mathbf{y})\right).$$

Therefore $\varphi(\cdot)$ is log-convex; now, we observe that

$$\varphi\left[t\mathbf{x} + (1-t)\mathbf{y}\right] \leqslant t\varphi(\mathbf{x}) + (1-t)\varphi(\mathbf{y})$$

is equivalent with

$$\ln\left(\varphi\left[t\mathbf{x} + (1-t)\mathbf{y}\right]\right) \leqslant \ln\left[t\varphi(\mathbf{x}) + (1-t)\varphi(\mathbf{y})\right]. \tag{4.4}$$

Since $\varphi(\cdot)$ is log-convex, we have

$$\ln\left(\varphi\left[t\mathbf{x} + (1-t)\mathbf{y}\right]\right) \leqslant t\ln\left(\varphi(\mathbf{x})\right) + (1-t)\ln\left(\varphi(\mathbf{y})\right) \tag{4.5}$$

and, by concavity of $\ln(\cdot)$,

$$t\ln\left(\varphi(\mathbf{x})\right) + (1-t)\ln\left(\varphi(\mathbf{y})\right) \leqslant \ln\left(t\varphi(\mathbf{x}) + (1-t)\varphi(\mathbf{y})\right). \tag{4.6}$$

Inequalities (4.5) and (4.6) combined give (4.4), hence, $\varphi(\cdot)$ is a convex function. □

The Lagrange function of (4.3) is $L(g,\lambda) = \Phi(g) + \langle\lambda, \Psi(g)\rangle$, and is a convex functional.

COROLLARY 1 *For every $\lambda \in \mathbb{R}^n$, $L(\cdot,\lambda)$ is a convex function.*

PROOF: Use Lemma 1 and the fact that $\Psi(\cdot)$ is affine. □

It easily seen that, for $0 < \alpha < 1$, with similar arguments one can prove that Φ is a concave functional. Thus, in this case, problem (4.3) can be written as

$$\min_{g \in U_0^n} -\Phi(g),\ -\Psi(g) = 0. \tag{4.7}$$

As we already said in the beginning of this section, we choose to study only the case $\alpha > 1$, the two cases being so similar. Therefore, in the following, we shall assume that $\alpha > 1$ if not otherwise mentioned.

4.1.2 Lagrangian analysis

For every $h \in U^n$ and $g \in U_0^n$, it exists a $t_0 > 0$, such that $(g + t_0 \cdot h) \in U_0^n$; Φ being convex, the function

$$t \to \frac{\Phi(g + t \cdot h) - \Phi(g)}{t}$$

is monotone, therefore the directional derivatives of the above functionals can be calculated using the Lebesgue monotone convergence theorem:

$$D\Phi(g)(h) = \lim_{t \to 0} \frac{\Phi(g + t \cdot h) - \Phi(g)}{t} =$$

$$= (1 - \alpha) \sum_{i=1}^{n} \int_{\mathbb{R}^n} \left[\mathcal{H}^{\alpha}(\mathrm{s}) f^{\alpha}(\mathrm{s}) g^{1-\alpha}(\mathrm{s}) \frac{h_i(s_i)}{g_i(s_i)} \right] d\mu(\mathrm{s}),$$

$$D\Psi(g)(h) = \lim_{t \to 0} \frac{\Psi(g + t \cdot h) - \Psi(g)}{t} =$$

$$= \left(\int_{\mathbb{R}} h_1(s_1)\, d\mu(s_1), \ldots, \int_{\mathbb{R}} h_n(s_n)\, d\mu(s_n) \right),$$

$$DL(g, \overline{\lambda})(h) = D\Phi(g)(h) + \langle \overline{\lambda}, D\Psi(g)(h) \rangle.$$

Using Theorem 3.4 from [3], $\overline{g}$ is solution to (4.3) if and only if it exists a $\overline{\lambda} \in \mathbb{R}^n$ (Lagrange multipliers) such that

$$\overline{g} \in \arg\min_{g \in U_0^n} L(g, \overline{\lambda}) \text{ and } \Psi(\overline{g}) = 0_{\mathbb{R}^n}. \tag{4.8}$$

For a given $\overline{\lambda} \in \mathbb{R}^n$, first condition in (4.8) is equivalent with $0 \in \partial L(\overline{g}, \overline{\lambda})$ - the subdifferential of $L(\cdot, \overline{\lambda})$. As $L(\cdot, \overline{\lambda})$ admits directional derivatives (at least) on every direction $h \in U^n$, a natural way to solve (4.8) is to find a solution to $DL(g, \overline{\lambda})(h) = 0$, $\forall\, h \in U^n$ or, equivalently,

$$(\alpha - 1) \sum_{i=1}^{n} \int_{\mathbb{R}^n} \left[\mathcal{H}^{\alpha}(\mathbf{s}) f^{\alpha}(\mathbf{s}) g^{1-\alpha}(\mathbf{s}) \frac{h_i(s_i)}{g_i(s_i)} \right] d\mu(\mathbf{s}) = \tag{4.9}$$

$$= \sum_{i=1}^{n} \overline{\lambda}_i \int_{\mathbb{R}} h_i(s_i)\, d\mu(s_i), \forall\, h \in U^n.$$

For an index $1 \leqslant i \leqslant n$, and a vector $\mathbf{s} = (s_1, s_2, \ldots, s_n) \in \mathbb{R}^n$, let us denote $\mathbf{s}^i = (s_1, \ldots s_{i-1}, s_{i+1}, \ldots, s_n)$.

LEMMA 2 *If $\overline{g}$ is a solution to (4.9), then, for every $1 \leqslant i \leqslant n$, we have*

$$\overline{g}_i(s_i) = \frac{\int_{\mathbb{R}^{n-1}} \left[\mathcal{H}^{\alpha}(\mathbf{s}) f^{\alpha}(\mathbf{s}) \overline{g}^{1-\alpha}(\mathbf{s}) \right] d\mu(\mathbf{s}^i)}{\int_{\mathbb{R}^n} \left[\mathcal{H}^{\alpha}(\mathbf{s}) f^{\alpha}(\mathbf{s}) \overline{g}^{1-\alpha}(\mathbf{s}) \right] d\mu(\mathbf{s})} \quad a.\ e.,\ and \tag{4.10}$$

if X_i is a random variable having density $\overline{g}_i$, and $b : \mathbb{R} \to \mathbb{R}$ is a continuous function, then

$$\mathbb{E}_{\overline{g}_i}\left[b(X_i) \right] = \frac{\int_{\mathbb{R}^n} \left[\mathcal{H}^{\alpha}(\mathbf{s}) f^{\alpha}(\mathbf{s}) \overline{g}^{1-\alpha}(\mathbf{s}) b(s_i) \right] d\mu(\mathbf{s})}{\int_{\mathbb{R}^n} \left[\mathcal{H}^{\alpha}(\mathbf{s}) f^{\alpha}(\mathbf{s}) \overline{g}^{1-\alpha}(\mathbf{s}) \right] d\mu(\mathbf{s})}. \tag{4.11}$$

PROOF: equation (4.9) has the form:

$$\sum_{i=1}^{n} \int_{\mathbb{R}^n} A(\mathbf{s}) h_i(s_i)\, d\mu(\mathbf{s}) = \sum_{i=1}^{n} \int_{\mathbb{R}} \overline{\lambda}_i h_i(s_i)\, d\mu(s_i), \forall\, h \in U^n. \tag{4.12}$$

We will use in this proof, the simple fact that, for a given measurable function $\psi : \mathbb{R} \to \mathbb{R}_+$, $\int_{\mathbb{R}} \psi(s)\, d\mu(s) = 0$ if and only if $\psi(s) = 0$ almost everywhere (i.e., except, perhaps, for s lying in a negligible set).

For a given i, let us choose $h_j \equiv 0$, for all $j \neq i$, and, denote

$$\Lambda_+^i = \left\{ s_i \,:\, \int_{\mathbb{R}^{n-1}} A(\mathbf{s})\, d\mu(\mathbf{s}^i) > \overline{\lambda}_i \right\},\ \Lambda_-^i = \mathbb{R} \setminus \Lambda_+^i.$$

We choose for (4.12), first

$$h_i(s_i) = \left[\int_{\mathbb{R}^{n-1}} A(\mathrm{s})\, d\mu(\mathrm{s}^i) - \overline{\lambda}_i\right]^+,\ i = \overline{1,n},$$

and secondly,

$$h_i(s_i) = \left[\int_{\mathbb{R}^{n-1}} A(\mathrm{s})\, d\mu(\mathrm{s}^i) - \overline{\lambda}_i\right]^-,\ i = \overline{1,n},$$

and we obtain

$$\int_{\Lambda_+^i} \left[\int_{\mathbb{R}^{n-1}} A(\mathrm{s})\, d\mu(\mathrm{s}^i) - \overline{\lambda}_i\right]^2 d\mu(\mathrm{s}_i) = 0$$

and, respectively,

$$\int_{\Lambda_-^i} \left[\int_{\mathbb{R}^{n-1}} A(\mathrm{s})\, d\mu(\mathrm{s}^i) - \overline{\lambda}_i\right]^2 d\mu(\mathrm{s}_i) = 0,$$

or, equivalently,

$$\int_{\mathbb{R}^{n-1}} A(s)\, d\mu(s^i) = \overline{\lambda}_i,\ a.e.$$

From (4.12) we get

$$g_i(s_i) = \frac{\alpha - 1}{\overline{\lambda}_i} \int_{\mathbb{R}^{n-1}} \left[\mathcal{H}^\alpha(s) f^\alpha(s) \overline{g}^{1-\alpha}(s)\right]\, d\mu(s^i).$$

Moreover, if, for any given i, we choose $h_i = \overline{g}_i$, and $h_j \equiv 0$ for all $j \neq i$, we get:

$$\overline{\lambda}_i = (\alpha - 1) \int_{\mathbb{R}^n} \left[\mathcal{H}^\alpha(s) f^\alpha(s) \overline{g}^{1-\alpha}(s)\right]\, d\mu(s).$$

From here (4.10) follows easily; on the other hand (4.11) is an easy problem of calculation. □

4.2 Estimation of rare-events probabilities

In this section we suppose that $\mathcal{H}(\mathrm{s}) = \mathbb{1}_{[\mathcal{F}(\mathrm{s}) \geqslant a]}$, where $\mathcal{F}$ is a Lebesgue measurable function, $a \in R$, and $[\mathcal{F}(\mathrm{s}) \geqslant a]$ is a small probability event (say, at most 10^{-5}).

We follow here the framework of a multistage procedure for the estimation of $\overline{g}$ - this type of algorithm appear often in the literature (e.g. [4], [31], [45]).

The line of the algorithm consists of building a sequence of thresholds $\left(\gamma^{(k)}\right)_{k\geqslant 0}$ which converges to a for $k \to \infty$, and a sequence of densities $\left(g^{(k)}\right)_{k\geqslant 0}$ convergent to $\overline{g}$. The first sketch of the algorithm follows the equations (4.10):

Algorithm 1 : Deterministic version

Step 1 $g^{(0)} \leftarrow f$; $k \leftarrow 1$;

Step 2 do {

for $(i = \overline{1,n})$

(1) $g_i^{(k)}(s_i) = \dfrac{\displaystyle\int_{\mathbb{R}^{n-1}} \left[\mathcal{H}^{\alpha}(\mathrm{s}) f^{\alpha}(\mathrm{s}) \left(g^{(k-1)}(\mathrm{s})\right)^{1-\alpha}\right] d\mu(\mathrm{s}^i)}{\displaystyle\int_{\mathbb{R}^n} \left[\mathcal{H}^{\alpha}(\mathrm{s}) f^{\alpha}(\mathrm{s}) \left(g^{(k-1)}(\mathrm{s})\right)^{1-\alpha}\right] d\mu(\mathrm{s})}$;

$k++$;

}

until(some stopping conditions)

It is worthnoting that above algorithm builds a sequence of pdf's which has to converge towards a fixed point of a certain operator on $\left(L^1(\mathbb{R})\right)^n$ (this operator becomes apparent in Lemma 2).

In order to present the stochastic version of the above algorithm we make some notations (see [22] and [31]). For a given level $\rho \in (0,1)$ we denote by $\gamma(g,\rho)$ a $(1-\rho)$-quantile of $\mathcal{F}(X)$ (X having density g):

$$\mathcal{P}_g\left[\mathcal{F}(\mathbf{X}) \geqslant \gamma(g,\rho)\right] \geqslant \rho \text{ and } \mathcal{P}_g\left[\mathcal{F}(\mathbf{X}) \leqslant \gamma(g,\rho)\right] \geqslant 1-\rho$$

If $\mathbb{X} = (X^1, X^2, \ldots, X^N)$ is a i. i. d. sample from g, then $\gamma(g,\rho)$ can be approximated by a $(1-\rho)$-sample quantile of

$$\mathcal{F}(X^1), \mathcal{F}(X^2), \ldots, \mathcal{F}(X^N),$$

denoted by $\gamma_N(\mathbb{X},\rho)$.

For continuous distributions clearly we can not implement this version of the algorithm, but, in most cases, using (4.11) we can completely determine the distribution $g^{(k)}$ using some of its moments (those who are sufficient for the calculation of expectation, variance etc).

Distributions that can be completely determined by some of their moments are, e.g., those who make part of natural exponential families (see

[33]) - normal, Poisson, Weibull or some types of gamma (such as exponential) distributions. In such cases we change (1) by:

$$\text{for } (r = \overline{1,p})$$
$$\mu_i^{(k),r} = \frac{\int_{\mathbb{R}^n} \left[\mathcal{H}^\alpha(\mathrm{s}) f^\alpha(\mathrm{s}) \left(g^{(k-1)}(\mathrm{s})\right)^{1-\alpha} s_i^r\right] d\mu(\mathrm{s})}{\int_{\mathbb{R}^n} \left[\mathcal{H}^\alpha(\mathrm{s}) f^\alpha(\mathrm{s}) \left(g^{(k-1)}(\mathrm{s})\right)^{1-\alpha}\right] d\mu(\mathrm{s})};$$
$$\text{and update } g^{(k)} \text{ using } \left(\mu_i^{(k),r}\right)_{\substack{i=\overline{1,n,}\\ r=\overline{1,p}}};$$

In our algorithm we make use of a well known method ([31]) which increases the sample size when the number of samples (used for parameter updating) becomes too small. We say that a given ρ satisfies the condition $C(\rho)$ if $\gamma_N(\mathbb{X}, \rho)$ is greater than or equal to $\min\{a, \gamma^{(k-1)} + \varepsilon\}$. Depending on $C(\rho_{k-1})$ we either resample with the same size, choose another, more appropriate, quantile level, or resample after an increase in the sample size. This method was first introduced in [31] and can enhance the quality of the algorithm in the presence of distributions which give poor estimates to the small probability we search for - by increasing the sample size.

With final distribution $g^{(k)} \cong \overline{g}$ we estimate m using (2.2):

$$m_N(g^{(k)}) = \frac{1}{N_0} \sum_{j=1}^{N_0} \mathbb{1}_{[\mathcal{F}(X^j) \geqslant \gamma]} \frac{f(X^j)}{g(X^j)}.$$

4.2.1 Numerical results

In this subsection we present the numerical results of our algorithm rare events probabilities; for our experiments we use a common example from literature (see [4], [5] or [45]).

We test our algorithm using the example from [4] - a weighted bridge with random independent weights X_i, exponential distributed: $X_i \sim Exp(\lambda_i)$, $i = \overline{1,5}$ (this distribution is parameterized by the mean $\mu_i = 1/\lambda_i$, hence is completely determined by its first moment). All intermediate distributions will make part of this family of exponential distributions. Consider the case $(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5) = (1, 1, 3, 2, 10)$ and estimate the probability that the shortest path from u to v is at least $a \in \{7, 8, 9\}$.

Figure 4.1: A bridge graph used as example

Algorithm 2 : Stochastic version

Step 1 initialization:

$g^{(0)} = f$; $k = 1$; N an initial sample size; ρ_0 an initial quantile level;

generate i.i.d. samples $\mathbb{X} = (X^1, X^2, \ldots, X^N)$ from $g^{(k-1)}$;

$\gamma^{(k)} = \min\{a, \gamma_N(\mathbb{X}, \rho_{k-1})\}$;

Step 2 repeat until $(\gamma^{(k)} = a)$

- for $(i = \overline{1,n},\ r = \overline{1,p})$ do

$$(2)\ \mu_i^{(k),r} = \frac{\sum_{j=1}^{N} \left[\mathbb{1}_{\left[\mathcal{F}(X^j) \geqslant \gamma^{(k)}\right]} f^{\alpha}(X^j) \left(g^{(k-1)}(X^j)\right)^{1-\alpha} \left(X_i^j\right)^r\right]}{\sum_{j=1}^{N} \left[\mathbb{1}_{\left[\mathcal{F}(X^j) \geqslant \gamma^{(k)}\right]} f^{\alpha}(X^j) \left(g^{(k-1)}(X^j)\right)^{1-\alpha}\right]};$$

- update $g^{(k)}$ using $\left(\mu_i^{(k),r}\right)_{\substack{i=\overline{1,n},\\ r=\overline{1,p}}}$;
- if $C(\rho_{k-1})$ then $\rho_k = \rho_{k-1}$; $k++$; generate i.i.d. samples $\mathbb{X} = (X^1, X^2, \ldots, X^N)$ from $g^{(k-1)}$ and update $\gamma^{(k)} \leftarrow \min\{a, \gamma_N(\mathbb{X}, \rho_{k-1})\}$;
- else, if $C(\rho')$ for some $\rho' < \rho_{k-1}$, then let ρ_k be the largest of such ρ' and $\gamma^{(k)} = \min\{a, \gamma_N(\mathbb{X}, \rho_{k-1})\}$;
- if $C(\rho')$ is not satisfied for any $\rho' \leqslant \rho_{k-1}$, then $N \leftarrow \nu \cdot N$; generate i.i.d. samples $\mathbb{X} = (X^1, X^2, \ldots, X^N)$ from $g^{(k-1)}$ and update $\gamma^{(k)} = \min\{a, \gamma_N(\mathbb{X}, \rho_{k-1})\}$;

In all our tests the initial data are: sample size is $N = 1000$, quantile level $\rho_0 = 0.01$, sample size increasing factor is $\nu = 2$, $\varepsilon = 0.001$, and $\alpha \in \{0.5, 1.5\}$. The results are listed in table 6.5 with 10^8 samples for basic Monte Carlo; our algorithm estimates the probabilities using (for the final step) $M = 1 \cdot 10^5$ and $2 \cdot 10^5$ simulations replications. The algorithm is run for $n = 30$ different seeds, and table 6.5 presents the estimates, the relative errors and the execution times on average.

It can be seen that the time simulation effort is reduced by roughly a factor of 350. Our estimates are more accurate for smaller probabilities (i. e., for $a = 9$) and have smaller relative errors.

Results from Table 6.5 allow us to compare our method involving Rényi divergence minimization with cross-entropy method. We see that our algorithm is at least as good in terms of estimates, relative error and execution time; the estimates are in general better although the differences remain small. We can conclude that our method is a good alternative for estimating small probabilities, and probably a procedure for tuning the parameter α will improve these results.

Table 6.6 shows the evolution of the means (first order moments) and the thresholds for different values of a. We see that the sequence of thresholds converges very fast; in almost all of the cases the desired value, a, is reached in the first three steps of the algorithm.

4.3 Optimization

An adaptation of the algorithm can be used for optimization if in the initialization step we add $g^{(0)}, g^{(-1)} \leftarrow h_0$ (h_0 being an initial distribution), and change (2) by

$$(3)\ \mu_i^{(k),r} = \frac{\sum_{j=1}^{N} \mathbb{1}_{\left[\mathcal{F}(X^j) \geqslant \gamma^{(k)}\right]} \left(g^{(k-2)}(X^j)\right)^{\alpha} \left(g^{(k-1)}(X^j)\right)^{1-\alpha} \left(X_i^j\right)^r}{\sum_{j=1}^{N} \mathbb{1}_{\left[\mathcal{F}(X^j) \geqslant \gamma^{(k)}\right]} \left(g^{(k-2)}(X^j)\right)^{\alpha} \left(g^{(k-1)}(X^j)\right)^{1-\alpha}};$$

This form of the algorithm follows the structure of equations (4.10) which is a recursive formula for the optimum distribution (solution for problem(4.3)). Hence, for rare event probabilities estimation we used equations (4.10) having a given distribution f and we approximate a fix point for a certain operator (readily seen in cited equations).

For optimization purposes we search for a distribution which (say) minimizes an objective function, therefore we can use any distribution as an initial candidate, and improve it along some iterations. A variant for this

algorithm could be to iterate (for a fixed $g^{(k-1)}$) until a good candidate for $g^{(k)}$ is find. As our focus was primarily on the rare-event probabilities estimation, we choose to present here a more simplified form (with quite good results) of this possible algorithm.

4.3.1 Numerical results for optimization

In the following experiments we use the family of multivariate normal distributions with independent components, that is, with probability density functions of product form. Normal distribution is completely determined by its first and second moments, hence the updates of intermediate distributions can be easily done.

Initial mean vector has components randomly chosen from $[-50, 50]$ and covariance matrix is a diagonal one with variances 625. Other initial values are: sample size $N = 2000$, quantile level $\rho_0 = 0.01$ and $\alpha = 1.5$. Many of the problems from below are with constraints - in these cases we use the accept/reject method for generate appropriate samples. We consider global minimization problems having the form:

$$\min_{\mathbf{x}\in\chi} \mathcal{F}(\mathbf{x})$$

The problems analyzed are enumerated below (for each problem we give the known optimum value and the corresponding optimum solution, $\mathbf{x}^*$):

- a trigonometric function ($n = 10$) which has frequent local minima:

  $$\mathcal{F}_1(\mathbf{x}) = \sum_{i=1}^{n} 8\sin^2\left(7(x_i - 0.9)^2\right) + 6\sin^2\left(14(x_i - 0.9)^2\right) + (x_i - 0.9)^2,$$

  with $-10 \leqslant x_i \leqslant 10$, $\mathbf{x}^* = (0.9, 0.9, \ldots, 0.9)^t$, and $\mathcal{F}_1(\mathbf{x}^*) = 1$.

- Griewangk's function ($n = 10$); it has many widespread local minima regularly distributed:

  $$\mathcal{F}_2(\mathbf{x}) = 0.00025 * \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$

  with $-600 \leqslant x_i \leqslant 600$, $\mathbf{x}^* = (0, 0, \ldots, 0)^t$, and $\mathcal{F}_2(\mathbf{x}^*) = 0$.

- Pinter's function ($n = 10$) which has many local minima:

  $$\mathcal{F}_3(\mathbf{x}) = \sum_{i=1}^{n} i x_i^2 + 20\sum_{i=1}^{n} i\sin^2\left(x_{i-1}\sin x_i - x_i + \sin x_{i+1}\right) + \\ + \sum_{i=1}^{n} i\log_{10}\left[1 + i(x_{i-1}^2 - 2x_i + 3x_{i+1} - \cos x_i + 1)^2\right],$$

where $x_0 = x_n$, $x_{n+1} = x_1$, $\mathbf{x}^* = (0, 0, \ldots, 0)^t$, and $\mathcal{F}_3(\mathbf{x}^*) = 0$.

- Rosenbrock's valley ($n = 2, 10$); global minimum lays inside a flat valley:

$$\mathcal{F}_4(\mathbf{x}) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right],$$

where $-2.048 \leqslant x_i \leqslant 2.048$, $\mathbf{x}^* = (1, 1, \ldots, 1)^t$, and $\mathcal{F}_4(\mathbf{x}^*) = 0$.

- Goldstein-Price's function ($n = 2$):

$$\begin{aligned}\mathcal{F}_5(\mathbf{x}) = \Big[1 + (x_1 + x_2 + 1)^2 \cdot \big(19 - 14x_1 + 3x_1^2 - 14x_2 + \\ + 6x_1x_2 + 3x_2^2\big)\Big] \cdot \Big[30 + (2x_1 - 3x_2)^2 \cdot \big(18 - 32x_1 + 12x_1^2 + \\ + 48x_2 - 36x_1x_2 + 27x_2^2\big)\Big].\end{aligned}$$

where $-2.0 \leqslant x_i \leqslant 2.0$, $\mathbf{x}^* = (0, -1)$, and $\mathcal{F}_5(\mathbf{x}^*) = 3$.

Our algorithm estimates the optimum value using the final mean (i.e., first order moments: $\mu^{(k),1}$); the algorithm is run for $n = 30$ independent replications, and Table 6.7 presents the functions values for the final means on average, corresponding standard errors (se), and the final $\gamma^{(k)}$ values.

If we compare these results with those from [22] obtained with the MRAS algorithm, we can see that they are quite similar; even in the estimate for $\mathcal{F}_4$ which is missed by both methods for large dimension.

Chapter 5

Measure the similarity using divergences

Univariate time series are sequences of numbers representing values of an attribute like weather data, medical data, financial information (option or stock prices, exchange). Time series are very common in various fields such as in medicine, multimedia and financial applications. Synthetic time series are of great help in prediction and various computer simulations; the resulted output must have the similar characteristics as the original time series in terms of statistical parameters and shape.

The relevance of synthetic time series relies first on the data generator used, and, secondly, on the accuracy of the similarity measures involved in the process of comparing the original and the synthetic time series. The critical issue is to choose an appropriate "distance" to represent the similarity between two time series.

This chapter aims to investigate some known and some new similarity measures on template and synthetic data sets. Real world time series being very complex their theoretical distributions are very difficult to formulate and very often hard to predict. Such complex datasets come from economic and scientific fields such as medicine, financial markets, multimedia.

Classic examples are financial data (stocks, bonds or option prices) which have unknown and hard to simulate distributions. The generation of clones of real (historical) datasets are required for the validation of efficient trading algorithms. Financial data are interesting because they exhibit specific patterns, which cannot be simulate using simple random generators.

The simplest way to measure the similarity between two time series is to use the Euclidean norm, Pearson product-moment correlation coefficient (see [9], [28]), or other metric-based similarities (Dynamic Time Warping, Edit Distance with Real Penalty). Other distances based on a matched

threshold, such as Edit Distance on Real Subsequence ([48]), rely on dynamic programming and require a big computational effort.

We propose to use as similarity measure some known symmetrised divergences (see [15]). Our experiments prove that such measures combined with mean similarity give more accurate results in terms of trajectory quality.

5.1 The data-generator

The aim of an efficient dataset generator is to offer many datasets which has to be similar but not identical with the original data.

In such cases an usual tool could be a data-generator which uses the original data and extract, for simulating purposes, some useful features. The generator produces random datasets which behave much like the original data and, based on the extracted features, keeps the most "similar" ones. This approach could be used in order to classify time series: the classification is driven by relevant features and using some similarities. The data-generator used here is similar with that from [42] and consists of

- a probabilistic context free grammar built from the original data;
- a random number generator which will serve a Ziggurat type normal deviates generator (as consecutive increments in original data are considered normally distributed);
- a low pass filter which smooths the original and the synthetic data (these filtered data will be tested for similarity as the first pair);
- a feature extractor for some statistics on increments of the datasets: mean, standard deviation, kurtosis, skewness;

5.1.1 Syntactic pattern recognition

The structure of the time dependency is captured using a probabilistic context-free grammar (see [18], [42]). This approach is more flexible as a context-sensitive grammar seems to be much more difficult to obtain; in such a case the number of production rules will be dictated by the original trajectory and could not be restricted by the user without drastically disturb the resulted synthetic data.

A probabilistic grammar is a quadruple $\Gamma = (T, N, \Pi, P, s_0)$: T is the terminal symbols set, N is the set of non-terminal symbol, s_0 is a starting symbol (usually from N), Π is the set of production rules, and P is a probability on the space of production rules.

The rules are of form: $x \to_p w$, where w is a word from $(N \cup T)^*$. If all the rules starting with x are

$$x \to_{p_1} w_1, x \to_{p_2} w_2, \ldots, x \to_{p_k} w_k$$

then

$$\sum_{i=1}^{k} p_i = 1.$$

The probabilistic context-free grammar is built like this:

- the range of increments[1] in the original data is divided into t equal sized intervals; this number remains to be carefully tuned by user.
- each interval will be represented by a symbol: we will have t symbols, say $x_1, x_2, \ldots, x_t$ plus a starting symbol s_0.
- we calculate the frequencies f_{ij} from the dependencies between symbols and, then, we determine the production rules and the corresponding probabilities: $x_i \rightarrow_{p_{ij}} x_j$ means x_i is followed by x_j with probability p_{ij}.

5.1.2 Generating data

The sequence of the differences could be viewed as a sequence of symbols. A new sequence of symbols is generated by using the production rules and the probabilities. This new sequence of symbols is transformed into a sequence of numbers (differences) by generating random numbers in the intervals associated with the symbols. Starting from the first original value and adding these new differences we obtain the synthetic data.

Formally, having already built the probabilistic grammar we proceed to generate a new trajectory:

- we start with s_0 (this symbol corresponds to the first value in the original data) and use the production rules;
- if the current symbol is x_i we generate an uniform variate from the assigned interval[2] and add it to the previous value;
- then we generate an uniform variate and choose the next production rule accordingly, say $x_i \rightarrow_{p_{ij}} x_j$; x_j will be the next symbol in line.

5.2 Measuring similarities

After the generator is up and produces synthetic time series, the following task is to choose one among all candidates. We implement this stage like a classifying procedure, i.e., we retain the time series which is most "similar" with the original one. There are many technique for measuring

[1] Which are considered normally distributed.

[2] We transform the symbol into a real value.

Figure 5.1: S&P 500 starting from January 2004 and a low pass filter.

similarities between two time series; our investigation concerns to types of similarity: one based on the feature extraction versus the similarity using some symmetrised divergence measures.

5.2.1 Feature extraction

Part of the template features could be preserved by the synthetic sequence. Usually some statistical characteristics of the original data are calculated and compared with the synthetic counterpart using mean similarity (see below). Feature extraction goes:

- the original and the generated data are smoothed using a low-pass filter - this emphasizes the trend of the series drastically reducing the noise;
- we estimate the mean, skewness and kurtosis for the distribution of the increments (which are supposed to be gaussian);

Some approaches could use more features including: slope, extremal points and indexing values. On the other hand these statistical features could be extracted after filtering the data. We choose here to calculate the statistics before filtering and to compare the filtered trajectories apart.

5.2.2 Similarity measures

For two time series $(U_i)_{1\leqslant i\leqslant n}$ and $(V_i)_{1\leqslant i\leqslant n}$ we denote

$$U_i' = \frac{U_i}{\sum_{i=1}^{n} U_i} \text{ and } V_i' = \frac{V_i}{\sum_{i=1}^{n} V_i},$$

(U_i, V_i have to be positive). We use the following similarities (slightly modified on a zero-to-one scale):

1. Mean similarity:

$$S(U,V) = \frac{1}{n}\sum_{i=1}^{n} s(U_i, V_i)$$

where

$$s(u,v) = 1 - \frac{|u-v|}{|u|+|v|}.$$

2. Pearson correlation coefficient $\rho(U,V)$:

$$\frac{n\sum_{i=1}^{n} U_i' V_i'}{\sqrt{\left(n\sum_{i=1}^{n} U_i'^2 - 1\right)}\sqrt{\left(n\sum_{i=1}^{n} V_i'^2 - 1\right)}}$$

3. Symmetric measure based on Kullback-Leibler divergence:

$$J(U,V) = \sum_{i=1}^{n} U_i' \log\left(U_i'/V_i'\right) - V_i' \log\left(U_i'/V_i'\right).$$

4. Symmetric measure based on χ^2-divergence:

$$\Psi(U,V) = \sum_{i=1}^{n}\left(\frac{U_i'^2}{V_i'} + \frac{V_i'^2}{U_i'}\right) - 2.$$

5. Symmetric measure based on AG-divergence:

$$T(U,V) = \frac{1}{4}\sum_{i=1}^{n}(U_i' + V_i')\left[\log\left(U_i' + V_i'\right) - V_i' \log\left(U_i' \cdot V_i'\right)\right] - \log 2.$$

6. Symmetric measure based on JS-divergence:

$$I(U,V) = \frac{1}{2}\sum_{i=1}^{n}\left[U_i \log\left(U_i'\right) + V_i \log\left(V_i'\right) - V_i' \log\left(U_i' + V_i'\right)\right] + \log 2.$$

The similarity between the original and the synthetic trajectories was checked using some of the following measures:

- the mean similarity on the above statistics - mean, kurtosis, skewness (δ_1);

- the mean similarity calculated on the raw and (δ_2) filtered data (δ_3);

Figure 5.2: S&P 500 simulated (21.323s) with $w_1 = 0.6$, $w_2 = 0.1$, $w_3 = 0.2$, $w_4 = 0.1$

- the Pearson product-moment correlation coefficient (δ_4) for raw (unfiltered) data;
- the symmetrised versions of: Kullback-Leibler divergence (δ_5), χ^2-divergence (δ_6), AG-divergence (δ_7), and JS-divergence (δ_8) on the raw data.

These similarities are combined into a weighted measure of similarity and give the algorithm 3.

Algorithm 3

Step 1 let $U = (U_i)_{1 \leqslant i \leqslant n}$ be the original data;

Step 2 repeat {

$V = (V_i)_{1 \leqslant i \leqslant n} \leftarrow \text{generate}();$// generate a synthetic trajectory.

$X = (\text{mean}(U), \text{stdev}(U), \text{skew}(U));$

$Y = (\text{mean}(V), \text{stdev}(V), \text{skew}(V));$

$\delta_1 \leftarrow S(X, Y)$, $\delta_2 \leftarrow S(U, V)$;

$\delta_3 \leftarrow S(\text{filter}(U), \text{filter}(V));$

$\delta_4 \leftarrow \rho(U, V);$

$\delta_5 \leftarrow 1 - |J(U, V)|$; $\delta_6 \leftarrow 1 - |\Psi(U, V)|$;

$\delta_7 \leftarrow 1 - |T(U, V)|$; $\delta_8 \leftarrow 1 - |I(U, V)|$; }

until $(\sum w_i \delta_i < \text{threshold})$

return V;

5.3 Numerical results

Financial data: we have used stock prices from the Standard and Poor's 500 index (daily closing) starting from January 2004 to December 2007 (figure 5.1).

The probabilistic grammar has between 15 and 20 symbols; the template data was split in segments of length 225 (except the last segment which

can be smaller), and the simulation was conducted separately on all these segments. Some preliminary experiments showed that an optimum length for a segment must be somewhere between 200 and 250.

We run the above algorithm using a maximum number of 5000 iterations in the repeat-until loop with a threshold $\theta = 0.995$). The low pass filtered and the original data are depicted in figure 5.1. The rest of the graphics illustrate synthetic trajectories obtained using combination of weights defined above.

The first two tests combine the statistical features and the low pass filter trajectory: $w_1 = 0.6, w_2 = 0.1, w_3 = 0.2, w_4 = 0.1$ is a mix which gives a bigger weight on mean similarity (figure 5.2) and $w_1 = 0.3, w_2 = 0.2, w_3 = 0.4, w_4 = 0.1$ which is a more equilibrated mix (figure ??). Both these tests avoid the (symmetrised) divergences, while the following use only combinations of mean similarity and a symmetrised divergence: $w_1 = w_j = 0.5$ with $j = \overline{5, 8}$ (see figures ?? - ??).

As we can see from the above figures we can obtain the same or a better quality trajectory combining only mean similarity and a divergence.

Moreover, the experiments show that we can lower the threshold and the algorithm will run faster without affecting the quality of the trajectory.

Chapter 6

Conclusions

We review in this chapter the original contributions presented in our thesis. The published articles which support these results are listed in the reference chapter; these results are also cited in the above chapters.

In Chapter 3, section 3.1 we investigate some Importance Sampling strategies and we apply them for the first time to the pricing of the European spread options. For valuing this type of options, we compare two methods, one based on the so-called f-divergence (Kullback-Leibler, χ^2, Rényi) and one based on the least squares method in order to choose the importance sampling functions.

Our numerical results reveals that the use of the divergences is frequently less computationally and time costly. As an example Kullback-Leibler divergence has always at least five times better (smaller) variance than crude Monte Carlo method, making this method a more reliable one.

We described a new version of Model Reference Adaptive Search algorithm which uses a sampling importance resampling phase (Chapter 3, section 3.2). This method has in background a sequence of reference distributions introduced mainly for theoretical purposes - at each step is considered the closest distribution to the corresponding reference distribution.

Our main contribution was to resample using this references, i.e., these distributions are involved in the calculus of the weights for the resampling stage. As an alternative, resampling with natural weights are also compared with the standard algorithm. The two sampling importance resampling (SIR) versions perform less step, while reference distributions SIR performs at most half steps than, than standard algorithm.

We apply these techniques to the pricing of Bermudan options. The prices we get are slightly smaller for importance resampling, although remaining very close to those obtained with the standard MRAS procedure. In almost all cases the standard error of the mean is similar for all three

algorithms, i.e., remains very small.

The proosed algorithm performs almost twice as fast as the standard algorithm having same standard errors - this means that our method is more reliable and faster method.

In Chapter 4 we introduced a new algorithm for estimation of rare-events probabilities (section 4.2) which can be used also for global optimization (section 4.3). Our method is not an adaptive one and is based on a different class of divergences than already known methods such cross-entropy method; these are Rényi divergences of order α. In this research we have studied mostly the case $\alpha > 1$, but is clearly that the other case ($\alpha \in (0, 1)$) is a similar one The optimal IS density is searched as a product-form probability density function - this is the only restriction we imposed.

We have also described an implementable version of the algorithm and tested it on some common example from literature. Numerical output is quite accurate and shows that the proposed algorithm gives good results. Global optimization algorithm version was tested on a number of known optimization problems and the results are encouraging for function with moderate dimension.

As future work we are interested in using some techniques which can reduce the danger of degeneracy of the likelihood ratio (like the screening method). Other issue of our algorithm concerns the exact convergence of the algorithm which is more probably to get on a reflexive space like $(L^2(\mathbb{R}))^n$ in the weak convergence topology.

We also proposed an improved method for generating high quality synthetic time series which follow a pattern - e.g. historic data (Chapter 5). Our method generates random trajectories and, using a classifying-like mechanism, preserves the most "similar" one.

Our main contribution is the introducing of symmetrised divergences as measures of similarity between time series. The similarities tested emphasize the value of the so-called divergences, which are usually used to compare distributions. While other approaches use feature extraction techniques and low pass filter for measuring similarity, we proved by experiments that, with a smaller computational effort - using divergences, we get a better quality for the synthetic data.

Numerical experiments show that in order to generate good trajectories we can avoid the feature extraction step (such as mean, standard deviation, skewness and kurtosis, extremal points, slope) and the filtering of data. Any of the tested symmetrised divergences corrected with the mean similarity are sufficient and offer a faster and more reliable technique.

Tables

Table 6.1: Geometric Brownian model - prices for a 3-years Bermudan put option with: $r = 0.05$, $\sigma = 0.25$, $\delta = 0$, $S_0 = K = 110$.

Number of exercices		S_0^* = 40	60	80	100	
$n = 2$		10.898 0.009/23.58	10.901 0.009/23.02	10.889 0.009/23.30	10.904 0.011/24.04	mras
		10.903 0.011/16.78	10.880 0.010/16.64	10.895 0.010/17.18	10.894 0.008/16.96	u-SIR
		10.893 0.010/13.2	10.941 0.033/12.6	10.914 0.008/12.46	10.905 0.009/12.60	rd-SIR
$n = 4$		11.529 0.016/9.44	11.545 0.019/9.50	11.508 0.019/9.62	11.5488 0.015/10.58	mras
		11.379 0.017/9.28	11.403 0.016/9.32	11.396 0.015/8.76	11.388 0.018/8.76	u-SIR
		11.418 0.020/6.94	11.416 0.021/6.44	11.431 0.025/7.18	11.417 0.022/6.94	rd-SIR
$n = 6$		12.508 0.017/11.16	12.529 0.012/11.46	12.536 0.014/11.02	12.517 0.019/11.22	mras
		12.465 0.019/8.88	12.448 0.018/9.20	12.458 0.019/9.20	12.453 0.026/9.16	u-SIR
		12.471 0.022/6.62	12.455 0.024/6.40	12.450 0.017/6.84	12.434 0.023/6.14	rd-SIR

Table 6.2: Jump diffusion model (Merton) - prices for an 1-year Bermudan put option having: $r = 0.05$, $\sigma = 0.1$, $\delta = 0.02$, $S_0 = K = 110$, $\lambda = 2$, $\gamma = 0.2$.

Number of exercices		S_0^* = 30	S_0^* = 50	S_0^* = 70	S_0^* = 90	
$n = 2$		10.410 0.009/24.66	10.401 0.008/25.08	10.383 0.008/24.2	10.372 0.010/24.34	mras
		10.402 0.010/16.80	10.404 0.009/18.20	10.375 0.009/16.52	10.379 0.008/17.76	u-SIR
		10.412 0.008/11.92	10.408 0.009/12.70	10.390 0.001/11.66	10.375 0.009/12.06	rd-SIR
$n = 4$		10.479 0.010/21.86	10.474 0.009/22.42	10.480 0.009/23.32	10.493 0.009/22.24	mras
		10.483 0.010/16.14	10.489 0.009/15.54	10.473 0.009/17.04	10.491 0.010/17.34	u-SIR
		10.481 0.009/11.74	10.481 0.010/10.58	10.488 0.007/11.42	10.467 0.009/10.78	rd-SIR
$n = 6$		10.643 0.008/23.44	10.656 0.008/23.18	10.645 0.008/23.46	10.645 0.009/21.94	mras
		10.652 0.011/17.18	10.664 0.011/16.18	10.652 0.009/16.18	10.651 0.007/16.68	u-SIR
		10.598 0.021/9.38	10.637 0.010/10.72	10.630 0.012/9.44	10.66 0.010/10.58	rd-SIR

Table 6.3: Double exponential jump diffusion model (Kou) - prices for an 1-year Bermudan put option with the following characteristics: $r = 0.05$, $\sigma = 0.1$, $\delta = 0.02$, $S_0 = K = 110$, $\lambda = 3.0$, $p = 0.3$, $\alpha_1 = 40$, $\alpha_2 = 12$.

Number of exercices		S_0^* = 30	S_0^* = 50	S_0^* = 70	S_0^* = 90	
$n = 2$		13.451 0.009/24.38	13.472 0.008/23.96	13.472 0.008/22.90	13.460 0.009/23.88	mras
		13.453 0.008/17.22	13.467 0.009/17.50	13.444 0.009/17.36	13.475 0.008/17.04	u-SIR
		13.439 0.008/12.62	13.454 0.009/11.84	13.470 0.010/12.64	13.460 0.007/12.86	rd-SIR
$n = 4$		13.440 0.010/23.78	13.425 0.009/24.24	13.418 0.009/23.56	13.423 0.006/24.06	mras
		13.430 0.010/17.46	13.419 0.007/17.04	13.409 0.009/17.36	13.424 0.008/16.50	u-SIR
		13.426 0.009/11.64	13.403 0.008/11.58	13.428 0.007/12.40	13.407 0.009/11.74	rd-SIR
$n = 6$		13.026 0.045/13.04	13.051 0.022/12.38	13.082 0.017/12.38	13.053 0.017/12.72	mras
		12.907 0.038/11.94	12.970 0.027/11.26	12.984 0.016/11.1	12.951 0.038/11.52	u-SIR
		12.9305 0.033/7.82	12.846 0.048/7.52	12.860 0.055/8.30	12.882 0.053/8.86	rd-SIR

Table 6.4: Pricing European Call spread option ($r = 5\%$, $S_1^0 = 105$, $S_2^0 = 110$, $q_1 = 2\%$, $q_2 = 3\%$, $\sigma_1 = 15\%$, $\sigma_2 = 10\%$) for various values of correlation and of strike.

ρ	K	LSq-IS	KL-IS	χ^2-IS	RI-IS
	-20	4.06(0.04)	5.87(0.06)	1.41(0.01)	0.77(0.06)
	-10	5.39(0.04)	6.12(0.06)	1.45(0.01)	0.64(0.07)
-0.9	0	6.72(0.04)	6.64(0.09)	2.52(0.34)	0.84(0.06)
	10	8.54(0.02)	8.30(0.09)	5.14(0.37)	0.91(0.05)
	20	11.28(0.05)	10.48(0.19)	8.97(0.44)	0.91(0.05)
	-20	3.80(0.04)	5.79(0.09)	4.58(0.24)	0.98(0.03)
	-10	5.22(0.04)	5.97(0.08)	4.80(0.18)	0.97(0.03)
-0.6	0	6.70(0.03)	6.89(0.05)	6.14(0.18)	1.03(0.01)
	10	8.58(0.02)	8.26(0.09)	7.30(0.23)	0.93(0.02)
	20	11.82(0.04)	11.00(0.15)	10.33(0.31)	0.94(0.04)
	-20	3.45(0.05)	5.76(0.08)	3.81(0.27)	2.63(0.11)
	-10	4.95(0.04)	5.91(0.07)	4.14(0.28)	2.92(0.09)
-0.3	0	6.55(0.03)	6.65(0.07)	5.77(0.23)	3.23(0.14)
	10	8.68(0.03)	8.27(0.11)	6.37(0.39)	3.79(0.18)
	20	12.65(0.03)	12.06(0.13)	10.37(0.5)	4.65(0.24)
	-20	3.16(0.03)	5.80(0.06)	3.76(0.26)	5.73(0.07)
	-10	4.58(0.03)	5.77(0.05)	4.05(0.24)	5.81(0.06)
0.0	0	6.46(0.03)	6.61(0.07)	4.67(0.29)	6.20(0.09)
	10	8.84(0.03)	8.42(0.09)	7.32(0.26)	6.36(0.11)
	20	13.75(0.05)	12.72(0.18)	10.86(0.51)	8.12(0.27)
	-20	2.58(0.03)	5.79(0.09)	3.53(0.27)	3.35(0.13)
	-10	4.19(0.04)	5.57(0.09)	4.16(0.26)	4.45(0.12)
0.3	0	6.17(0.03)	6.24(0.07)	4.88(0.25)	6.50(0.05)
	10	9.16(0.02)	8.93(0.08)	7.80(0.29)	8.02(0.13)
	20	16.10(0.07)	14.58(0.25)	12.99(0.57)	13.50(0.35)
	-20	1.95(0.02)	5.55(0.14)	4.35(0.27)	1.80(0.09)
	-10	3.30(0.03)	5.32(0.06)	4.00(0.22)	2.41(0.09)
0.6	0	5.83(0.03)	6.05(0.07)	5.06(0.20)	4.69(0.11)
	10	9.63(0.03)	9.13(0.13)	8.35(0.26)	8.06(0.07)
	20	22.82(0.11)	20.69(0.48)	19.66(0.67)	19.59(0.31)
	-20	1.17(0.02)	5.79(0.23)	3.03(0.15)	0.30(0.02)
	-10	1.20(0.03)	5.14(0.07)	3.31(0.19)	0.31(0.02)
0.9	0	4.80(0.02)	5.37(0.05)	4.93(0.07)	0.76(0.03)
	10	11.97(0.06)	11.23(0.20)	7.95(0.20)	0.89(0.11)
	20	131.62(5.90)	107.68(7.50)	84.67(6.05)	33.73(2.59)

Table 6.5: Results for the estimation of various small probabilities obtained using Rényi divergence minimization and cross-entropy.

M	$a = 7$	$a = 8$	$a = 9$	
$1 \cdot 10^5$	$5.33 \cdot 10^{-6}$ 0.060 (0.825s)	$6.77 \cdot 10^{-7}$ 0.020 (0.828s)	$8.88 \cdot 10^{-8}$ 0.008 (0.829s)	Rényi $\alpha = 1.5$
	$4.68 \cdot 10^{-6}$ 0.055 (0.830s)	$6.16 \cdot 10^{-7}$ 0.024 (0.837s)	$8.19 \cdot 10^{-8}$ 0.009 (0.828s)	Rényi $\alpha = 0.5$
	$5.80 \cdot 10^{-6}$ 0.068 (0.826s)	$7.17 \cdot 10^{-7}$ 0.029 (0.831s)	$8.79 \cdot 10^{-8}$ 0.009 (0.818s)	CE
$2 \cdot 10^5$	$5.70 \cdot 10^{-6}$ 0.083 (1.611s)	$6.53 \cdot 10^{-7}$ 0.026 (1.612s)	$8.61 \cdot 10^{-8}$ 0.011 (1.607s)	Rényi $\alpha = 1.5$
	$5.09 \cdot 10^{-6}$ 0.075 (1.635s)	$6.14 \cdot 10^{-7}$ 0.029 (1.634s)	$7.99 \cdot 10^{-8}$ 0.010 (1.625s)	Rényi $\alpha = 0.5$
	$4.78 \cdot 10^{-6}$ 0.067 (1.624s)	$6.78 \cdot 10^{-7}$ 0.031 (1.633s)	$8.73 \cdot 10^{-8}$ 0.011 (1.616s)	CE
	$1.6 \cdot 10^{-7}$/0.999 (815s)	$1.2 \cdot 10^{-7}$/0.999 (754s)	$3 \cdot 10^{-8}$/0.999 (761s)	MC

Table 6.6: Typical evolutions of the parameters.

	k	$\gamma^{(k)}$	$\mu_1^{(k),1}$	$\mu_2^{(k),1}$	$\mu_3^{(k),1}$	$\mu_4^{(k),1}$	$\mu_5^{(k),1}$
$a = 7$	1	2.379	1.0	1.0	0.333	0.500	0.100
	2	5.895	2.518	2.742	0.086	0.199	0.012
	3	7.000	6.105	6.199	0.028	0.167	0.003
$a = 8$	1	2.547	1.0	1.0	0.333	0.500	0.100
	2	6.311	2.920	2.880	0.260	0.376	0.107
	3	8.000	7.134	8.022	0.308	0.186	0.076
$a = 9$	1	2.427	1.0	1.0	0.333	0.500	0.100
	2	6.311	2.497	2.550	0.160	0.269	0.075
	3	9.000	7.670	7.513	0.094	0.171	0.056

Table 6.7: Results for different continuous multi-extremal optimization problems.

	optimum	estimate	se	final $\gamma^{(k)}$
$\mathcal{F}_1$	1.0	1.0001	$6.9 \cdot 10^{-10}$	1.005
$\mathcal{F}_2$	0.0	0.0043	$6.3 \cdot 10^{-6}$	0.003
$\mathcal{F}_3$	0.0	0.043	$7.1 \cdot 10^{-5}$	0.0276
$\mathcal{F}_4$ ($n = 2$)	0.0	0.0007	$4.9 \cdot 10^{-7}$	0.0003
$\mathcal{F}_4$ ($n = 10$)	0.0	7.1541	$4.5 \cdot 10^{-3}$	7.0905
$\mathcal{F}_5$	3.0	3.0000	$4.7 \cdot 10^{-10}$	3.0000

Appendix A

Spread options

Consider two assets whose values we denote $S_1(t)$ and $S_2(t)$ – at time t, a spread option is an option on the difference of the two underlying assets. It becomes a new instrument

$$S(t) = S_2(t) - S_1(t), 0 \leqslant t \leqslant T,$$

buying this spread means to buy S_2 and sell S_1, and selling means to buy S_1 and sell S_2.

The European type spread option gives the buyer the right to be paid, at the maturity time T, the spread: $S(T) = S_2(T) - S_1(T)$. In order to exercise the option, the owner must pay at maturity a fixed price K known as the exercise price, or the strike. The payoff of a spread option is

$$\max\{S_2(T) - S_1(T) - K, 0\} = (S_2(T) - S_1(T) - K)^+ \tag{A.1}$$

We assume the classical framework([7], [8]): constant interest rate r, constant volatilities σ_1, σ_2, continuous dividend yields q_1, q_2. The risk-neutral price dynamics of these two assets are given by two stochastic differential equations

$$\begin{aligned} dS_1(t) &= S_1(t)[(r - q_1)dt + \sigma_1 dW_1(t)] \\ dS_2(t) &= S_2(t)[(r - q_2)dt + \sigma_2 dW_2(t)] \end{aligned} \tag{A.2}$$

where W_1 and W_2 are two Brownian motions with correlation ρ. The price of our spread option will be:

$$V = e^{-rT}\mathbb{E}\left[(S_2(T) - S_1(T) - K)^+\right] \tag{A.3}$$

Because the above equations can be solved separately, the price can be expressed as

$$V = e^{-rT}\mathbb{E}\Big[\Big(S_2^0 \exp\left(r - q_2 - \sigma_2^2/2 + \sigma_2 Z_2\right) - S_1^0 \exp\left(r - q_1 - \sigma_1^2/2 + \sigma_1 Z_1\right) - K\Big)^+\Big] \tag{A.4}$$

where S_1^0 and S_2^0 are the spot prices (i.e. $S_1^0 = S_1(0), S_2^0 = S_2(0)$), and (Z_1, Z_2) is the Gaussian joint distribution of $W_1(T)$ and $W_2(T)$. Except for the case of an exchange option (when the strike equals zero), the price of the spread option cannot be given by a closed form formula. Our aim is to estimate this price in the Monte Carlo framework, and to reduce the variance using importance sampling technique.

The payoff function will be

$$\mathcal{H}(\mathbf{Z}) = \Big[S_2^0 \exp\left((r - q_2 - \sigma_2^2/2) + \sigma_2 Z_2\right) - S_1^0 \exp\left((r - q_1 - \sigma_1^2/2) + \sigma_1 Z_1\right) - K\Big]^+, \tag{A.5}$$

where $\mathbf{Z} = (Z_1, Z_2)$ has a normal bivariate distribution.

Appendix B

American options

We consider here the American option with a finite number of early exercise opportunities, also called Bermudan options (somewhere between, but closer to, American and European options). Let us suppose that: K is the strike price, r is the interest rate, T is the maturity, n the number of early exercise opportunities (without maturity), the exercise dates being

$$0 < t_1 < \ldots < t_{n+1} = T,$$

S_i^* are the early exercise thresholds, and S_i the stock prices.

For such a derivative the value of a *call option* is:

$$\max_{\mathbf{S}} \mathbb{E}\left[\mathcal{V}_{c,\mathbf{S}}\right], \tag{B.1}$$

where $\mathbf{S} = (S_1^*,\ S_2^*,\ \ldots,\ S_n^*)$, and (see [51])

$$\mathcal{V}_{c,\mathbf{S}}(\mathcal{S}) = \sum_{i=1}^{n} (S_i - K)^+ \cdot e^{-rt_i} \cdot \mathbb{1}_{\left[S_1 < S_1^*,\ \ldots,\ S_{i-1} < S_{i-1}^*,\ S_i > S_i^*\right]} + \tag{B.2}$$

$$+ (S_{n+1} - K)^+ \cdot e^{-rT} \cdot \mathbb{1}_{\left[S_1 < S_1^*,\ \ldots,\ S_{n-1} < S_{n-1}^*,\ S_n < S_n^*\right]}$$

where $\mathcal{S} = (S_1, S_2, \ldots, S_n) \in \mathbb{R}^n$ is the vector of stock prices.

In a similar way the payoff for a *put option* is:

$$\mathcal{V}_{p,\mathbf{S}}(\mathcal{S}) = \sum_{i=1}^{n} (K - S_i)^+ \cdot e^{-rt_i} \cdot \mathbb{1}_{\left[S_1 > S_1^*,\ \ldots,\ S_{i-1} > S_{i-1}^*,\ S_i < S_i^*\right]} + \tag{B.3}$$

$$+ (K - S_{n+1})^+ \cdot e^{-rT} \cdot \mathbb{1}_{\left[S_1 > S_1^*,\ \ldots,\ S_{n-1} > S_{n-1}^*,\ S_n > S_n^*\right]}$$

In both equations (B.2) and (B.3) the second term on the right side is the payoff at the maturity time. For a *call option* the threshold prices form a decreasing sequence of non-negative numbers with the final threshold being the strike price:

$$S_1^* \geqslant S_2^* \geqslant \ldots \geqslant S_n^* \geqslant S_{n+1}^* = K \tag{B.4}$$

For a *put option* the threshold prices form an increasing sequence of non-negative numbers with the final threshold being the strike price:

$$S_1^* \leqslant S_2^* \leqslant \ldots \leqslant S_n^* \leqslant S_{n+1}^* = K \tag{B.5}$$

$\mathbf{S} = (S_1^*, S_2^*, \ldots, S_n^*)$ follow a multivariate gaussian distribution $N(\mu, \Sigma)$. In the case of the *call option* the restrictions on these prices are:

$$S_i^* \geqslant 0,\ i = \overline{1,n};\ \text{and}\ S_1^* \geqslant S_2^* \geqslant \ldots, \geqslant S_n^* \geqslant S_{n+1}^* = K.$$

defining a particular polytope

$$\mathbb{X}_c = \{\mathbf{x} \in \mathbb{R}_+^n\ :\ x_1 \geqslant x_2 \geqslant \ldots \geqslant x_n \geqslant K\}. \tag{B.6}$$

Similarly, the thresholds for a *put option* lie in the following polytope

$$\mathbb{X}_p = \{\mathbf{x} \in \mathbb{R}_+^n\ :\ x_1 \leqslant x_2 \leqslant \ldots \leqslant x_n \leqslant K\}. \tag{B.7}$$

References

[1] Agrawal, R., C. Faloutsos, and A. Swami, *Efficient similarity search in sequence databases*, in Proceedings of the 4th Inter- national Conference on Foundations of Data Organization and Algorithms (FODO '93), vol. 730, LNCS, pp. 69-84, 1993.

[2] Anwar, M., S. Hussain, J. Pecarić, *Some inequalities for Csiszár-divergence measures*, Int. Journal of Math. Analysis, vol 3., 26, pp. 1295-1304, 2009.

[3] Barbu, V., T. Precupanu, *Convexity and Optimization in Banach Spaces*, 4th edition, Springer Verlag, 2012.

[4] de Boer, P. T., D. P. Kroese, S. Mannor, R. Y. Rubinstein, *A tutorial on the cross-entropy method*, Annals of Operations Research, 134, pp. 19-67, 2005.

[5] Botev, Z. I., D. P. Kroese, *An efficient algorithm for rare-event probability estimation, combinatorial, and counting*, Methodology and Computing in Applied Probability, 10(4), pp. 471-505, 2008.

[6] Boudraa, A.-O., J.-C. Cexus, M. Groussat, and P.Brunagel, *An Energy-Based Similarity Measure for Time Series*, Journal of Advances in Signal Processing, ID, 135892, vol. 2008.

[7] Carmona, R., V. Durrleman, *Pricing and Hedging Spread Options*, SIAM Review, Vol. 45, No. 4, pp. 627-685, 2003.

[8] Carmona, R., V. Durrleman, *Generalizing the Black-Scholes formula to multivariate contingent claims*, Journal of Computational Finance, Vol. 9, No. 2, pp. 43-67, 2005.

[9] Chu, S., E. Keogh, D. Hart, and M. Pezzani, *Iterative deepening dynamic time warping for time series*, Proceedings of the 2nd SIAM International Conference on Data Mining, 2002.

[10] Cont, R., P. Tankov, *Financial Modelling with Jump Processes*, Chapman & Hall/CRC, 2004.

[11] Csiszár, I., *Information measures, A critical survey*, Trans. 7th Prague Conf. on Information Theory, vol. B, Academia Prague, pp. 73-86, 1978.

[12] Dagpunar, J. S., *Simulation and Monte Carlo - with applications in finance and MCMC*, First Edition, John Wiley & Sons, 2007.

[13] Davidson, A. C., *Statistical Models*, Cambridge University Press, 2003.

[14] Dobigeon, N., J.-Y. Tourneret, *Efficient sampling according to a multivariate Gaussian ditribution*, Technical Report, 2007.

[15] Dragomir, S. S., J. Sunde, C. Buşe, *Some new inequalities for Jeffreys divergence measure in information theory*, RGMIA Res. Rep. Coll., 3 (2), Article 5, 2000.

[16] van Erven, T., P. Harremoes, *Rényi divergence and majorization*, IEEE International Symposium on Information Theory Proceedings (ISIT), vol. 305, pp. 1335-1339, 2010.

[17] Fishman, G., *Monte Carlo: Concepts, Algorithms and Applications*, Springer Verlag, New York, 1997.

[18] Fu, K. S., *Syntactic pattern recognition and applications*,Englewood C liffs, Prentice Hall, 1982.

[19] Gentle, J. E., *Random number generation and Monte Carlo methods*, Springer Verlag, 2003.

[20] Geweke, J., *Efficient simulation from the multivariate normal and student t-tistributions subject to linear constraints*, Computer Sciences and Statistics Proceedings of the 23d Symposium on the Interface, pp. 571-578, 1991.

[21] Geweke, J., *Monte Carlo Simulation and Numerical Integration*, Federal Reserve Bank of Minneapolis, Research Department Staff Report 192, 1995.

[22] Hu, J., M. C. Fu, S. I. Marcus, *A Model Reference Adaptive Search Method for Global Optimization*, INFORMS Operations Research, vol. 55, no. 3, pp. 549-568, 2007.

[23] Jäeckel, P., *Monte Carlo methods in finance*, John Wiley & Sons, 2002.

[24] Kannan, R., G. Li, *Sampling according to the multivariate normal density*, 1999.

[25] Kou, S., G., *A Jump-Diffusion Model for Option Pricing*, Management Science, INFORMS, vol. 48, no. 8, pp. 1086-1101, 2002.

[26] Kroese, D. P., S. Porotsky, R, Y. Rubinstein, *The cross-entropy method for continuous multi-extremal optimization*, Methodology and Computing in Applied Probability, 8, pp. 383-407, 2006.

[27] Kroese, D. P., T. Taimre, Z. I. Botev, *Handbook of Monte Carlo methods*, John Wiley & Sons, 2011.

[28] Last, M., Y. Klein, and A. Kandel, *Knowledge discovery in time series databases*, IEEE Trans. on Systems, Man, and Cybernetics, 31B(1), 2001.

[29] Leva, J. L., *A fast normal random number generator*, ACM Trans. Math. Softw. 18, 4, pp. 449-453, 1992.

[30] Madsen, K., H.B. Nielsen and O. Tingleff, *Methods for Nonlinear Least Squares Problems*, 2nd Electronic Edition, IMM, DTU, 2004.

[31] Homem-de-Mello, T., *A Study on the Cross-Entropy Method for Rare-Event Probability Estimation*, INFORMS Journal of Computing, vol. 19, no. 3, pp. 381-394, 2007.

[32] Merton, R. C., *Option pricing when underlying stock returns are discontinuous*, Journal of Financial Economics, 3, pp. 125-144, 1976.

[33] Morris, C. N., *Natural exponential families with quadratic variance functions*, The Annals of Statistics, vol. 10, 1, pp. 65-80, 1982.

[34] Niederreiter, H., *Random number generation and quasi-Monte Carlo methods*, SIAM CBMS-NSF, vol. 63, Regional Conference Series in Applied Mathematics, 1992.

[35] Nocedal, J., S.J. Wright, *Numerical Optimization*, Springer Verlag, 1999.

[36] Øivind, S., E. Bølviken. L. Holden, *Improved sampling-importance Resampling and reduced bias importance sampling*, Scandinavian Journal of Statistics, vol. 30, 4, pp. 719-737, 2003.

[37] Olariu, E. F., *Monte Carlo variance reduction. Importance sampling techniques*, 11th Interantional Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2009.

[38] Olariu, E. F., *A model reference type algorithm using importance resampling*, 14th Interantional Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2012.

[39] Olariu, E. F., *Importance resampling with MRAS algorithm for Bermudan option pricing*, Technical Report, TR12-01, `http://profs.info.uaic.ro/~tr/tr12-01.pdf`, 2012.

[40] Olariu, E. F., *Importance sampling using Rényi divergence*, Proccedings of the 20th International Conference on Computational Statistics (COMPSTAT), 2012.

[41] Olariu, E. F., *Importance sampling using Rényi divergence*, Technical Report, TR12-02, `http://profs.info.uaic.ro/~tr/tr12-02.pdf`, 2012.

[42] Ramos, J. R., V. Rego, *Feature-based generators for time series data*, Proceedings of the 2005 Winter Simulation Conference.

[43] Robert, C. P., *Simulation of truncated normal variables*, Statistics and Computing, 5, pp. 121-125, 1995.

[44] Rubinstein, R. Y., *Optimization of computer simulation models with rare events*, Methodology and Computing in Applied Probability, 1, pp. 127-190, 1997.

[45] Rubinstein, R. Y., *The cross-entropy method for combinatorial and continuous optimization*, Methodology and Computing in Applied Probability, 1, pp. 127-190, 1999.

[46] Rubinstein, R. Y., D. P. Kroese *Simulation and the Monte Carlo method*, John Wiley & Sons, 2007.

[47] Tankov, P., E. Voltchkova, *Jump-diffusion models: a practitioner's guide*, Banque et Marchés, No. 99, March-April 2009.

[48] Vlachos, M., G. Kollios, and D. Gunopulos, *Discovering Similar Multidimensional Trajectories*, ICDE, 673-684, 2002.

[49] Wang, Q., V. Megalooikonomou, *A dimensionality reduction technique for efficient time series similarity analysis*, Inf. Syst. 33 (1), pp. 115-132, 2008.

[50] Walsh, B., *Markov chain Monte Carlo and Gibbs sampling*, Lecture notes for EEB 581, 2004.

[51] Zhang, H., M. C. Fu, *Applying model reference adaptive search to american style option pricing*, Proceedings of the Winter Simulation Conference, 2006.