On Variance Reduction of Mean-CVaR Monte Carlo Estimators

Václav Kozmík

Faculty of Mathematics and Physics
Charles University in Prague

April 3, 2014

Outline

- Risk-Averse optimization
  - Mean-risk objectives with CVaR are often used
  - To solve complex models, we need to use approximations
    - Monte-Carlo methods
- Standard estimators are not convenient for mean-CVaR operators
  - They lead to high variance, due to the properties of CVaR
  - We propose a sampling scheme based on importance sampling
    - Analytically evaluated under the assumption of normal distribution
    - For general setup, algorithm is given to find a suitable sampling scheme
  - We validate our results with a numerical example, which uses Stochastic Dual Dynamic Programming algorithm

Basic model

- CVaR formula:

$$\mathrm{CVaR}_\alpha[Z] = \min_u \left(u + \frac{1}{\alpha}\mathbb{E}[Z-u]_+ \right)$$

- Consider following mean-risk functional:

$$\mathcal{Q}_\alpha[Z] = (1-\lambda)\,\mathbb{E}[Z] + \lambda\,\mathrm{CVaR}_\alpha[Z]$$

  - Z represents random losses
  - convex sum: $\lambda \in [0,1]$
  - suppose that Z follows a pdf f
- Such functionals are present in many types of models, static cases, multistage cases
  - Wide range of applications of our sampling scheme

Standard Monte Carlo

- Standard Monte Carlo approach is not convenient for estimation of CVaR

Example

Consider the following estimator of $\mathrm{CVaR}_\alpha[Z]$, *where* $Z^1, Z^2, \ldots, Z^M$ *are independent and identically distributed (i.i.d.) from the distribution of* Z:

$$\min_u \left(u + \frac{1}{\alpha M} \sum_{j=1}^{M} \left[Z^j - u \right]_+ \right).$$

If $\alpha = 0.05$ *only about 5% of the samples contribute nonzero values to this estimator of* CVaR.

Importance sampling

- Aims to solve the issues mentioned in previous example
- Suppose we want to compute $\mathbb{E}\left[\mathcal{Q}(\mathbf{x}, Z)\right]$ with respect to the pdf f of the random variable Z
- Therefore: $\mathbb{E}_f\left[\mathcal{Q}(\mathbf{x}, Z)\right] = \int_{-\infty}^{\infty} \mathcal{Q}(\mathbf{x}, z) f(z) \mathrm{d}z$
- Choose another pdf g of some random variable and compute:

$$\int_{-\infty}^{\infty} \mathcal{Q}(\mathbf{x}, z) f(z) \mathrm{d}z = \int_{-\infty}^{\infty} \mathcal{Q}(\mathbf{x}, z) \frac{f(z)}{g(z)} g(z) \mathrm{d}z = \mathbb{E}_g\left[\mathcal{Q}(\mathbf{x}, Z) \frac{f(Z)}{g(Z)}\right]$$

- Therefore

$$\mathbb{E}_f\left[\mathcal{Q}(\mathbf{x}, Z)\right] = \mathbb{E}_g\left[\mathcal{Q}(\mathbf{x}, Z) \frac{f(Z)}{g(Z)}\right]$$

Importance sampling

- In the context of Monte Carlo, $\mathbb{E}_f\left[\mathcal{Q}(\mathbf{x}, Z)\right]$ is replaced with:
  - Sample $Z^1, Z^2, \ldots, Z^M$ from distribution with pdf f
  - Compute

$$\frac{1}{M}\sum_{j=1}^{M}\mathcal{Q}(\mathbf{x}, Z^j)$$

- The importance sampling scheme is as follows:
  - Sample $Z^1, Z^2, \ldots, Z^M$ from distribution with pdf g
  - Compute

$$\frac{1}{M}\sum_{j=1}^{M}\mathcal{Q}(\mathbf{x}, Z^j)\frac{f(Z^j)}{g(Z^j)}$$

- Function g should be chosen such that the variance of the sum above is minimal

Further variance reduction

- The term $w^j = \frac{f(Z^j)}{g(Z^j)}$ could be considered as a weight:

$$\frac{1}{M}\sum_{j=1}^{M} \mathcal{Q}(\mathbf{x}, Z^j) w^j$$

- In expectation, we have $\mathbb{E}\left[w^j\right] = 1$, but the term itself is random and has nonzero variance
- Replace the $M = \mathbb{E}\left[\sum_{j=1}^{M} w^j\right]$ with the actual value:

$$\frac{1}{\sum_{j=1}^{M} w^j}\sum_{j=1}^{M} \mathcal{Q}(\mathbf{x}, Z^j) w^j$$

Further variance reduction

- We no longer have the expectation equality:

$$\mathbb{E}_g\left[\frac{1}{\sum_{j=1}^M w^j}\sum_{j=1}^M \mathcal{Q}(\mathbf{x}, Z^j)w^j\right] \neq \mathbb{E}_f\left[\frac{1}{M}\sum_{j=1}^M \mathcal{Q}(\mathbf{x}, Z^j)\right]$$

- But we can show consistency:

$$\mathbb{E}_g\left[\frac{1}{\sum_{j=1}^M w^j}\sum_{j=1}^M \mathcal{Q}(\mathbf{x}, Z^j)w^j\right] \to \mathbb{E}_f\left[\mathcal{Q}(\mathbf{x}, Z)\right], \ \ w.p.\ 1,$$

  as $M \to \infty$.
- The benefit is usually significant variance reduction over the standard importance sampling scheme

Mean-CVaR estimation

- What is a suitable importance sampling scheme for mean-CVaR?

$$\mathcal{Q}_\alpha[Z] = (1-\lambda)\,\mathbb{E}[Z] + \lambda\,\mathrm{CVaR}_\alpha[Z]$$

  - The functional clearly depends on all outcomes of Z
  - We have observed that CVaR is hard to estimate with standard Monte Carlo approach
  - We will divide the support of the distribution into two atoms:
    - "CVaR" atom
    - "non-CVaR" atom
  - We can select the same weight for both atoms, but is it a reasonable choice?

Mean-CVaR estimation

- Since $\mathrm{CVaR}_\alpha[Z] = \mathbb{E}[Z | Z > \mathrm{VaR}_\alpha[Z]]$, we can easily define the "CVaR" atom
- Using the pdf f, we compute the value at risk $u_Z = \mathrm{VaR}_\alpha[Z]$
  - the threshold can be also estimated using sampling
- The proposed importance sampling pdf is, with $\beta \in (0, 1)$:

$$g(z) = \begin{cases} \dfrac{\beta}{\alpha} f(z), & \text{if } z \geq u_Z \\ \dfrac{1-\beta}{1-\alpha} f(z), & \text{if } z < u_Z \end{cases}$$

- We are more likely to draw sample observations above $\mathrm{VaR}_\alpha[Z]$
- Suitable choice of β should be tailored to the values of α and λ

Variance reduction

- We define:

$$Q^s = (1-\lambda)\,Z + \lambda\left(u_Z + \frac{1}{\alpha}\,[Z-u_Z]_+\right)$$
$$Q^i = \frac{f}{g}\left((1-\lambda)\,Z + \lambda\left(u_Z + \frac{1}{\alpha}\,[Z-u_Z]_+\right)\right)$$

- It clearly holds $\mathcal{Q} = \mathbb{E}_g\left[Q^i\right] = \mathbb{E}_f\left[Q^s\right]$
- Our aim is to minimize variance, e.g. finding suitable parameter β, so that $\mathrm{var}_g\left[Q^i\right] < \mathrm{var}_f\left[Q^s\right]$
- With another random variable, we will write Q^s_X, Q^i_X, etc.

Basic properties

- The variance of our importance sampling estimator is invariant to addition of a constant and scales well with transformations

Proposition

Let X, Y be random variables, $Y = X + \mu$, $\mu \in \mathbb{R}$, f_X and f_Y the corresponding pdfs. Suppose that their importance sampling versions g_X and g_Y are defined using the same value of parameter β. Then $\operatorname{var}_{g_Y}\left[Q_Y^i\right] = \operatorname{var}_{g_X}\left[Q_X^i\right]$.

Proposition

Let X, Y be random variables, $Y = \sigma X$, $\sigma > 0$, f_X and f_Y the corresponding pdfs. Suppose that their importance sampling versions g_X and g_Y are defined using the same value of parameter β. Then $\operatorname{var}_{g_Y}\left[Q_Y^i\right] = \sigma^2 \operatorname{var}_{g_X}\left[Q_X^i\right]$.

Normal distribution

- We will now suppose that the losses follow normal distribution, with $\phi(x)$ as its pdf and $\Phi(x)$ its distribution function

Proposition

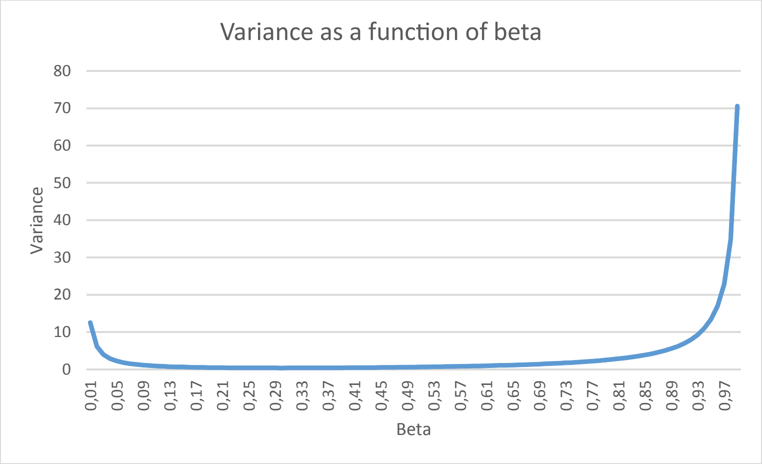
Let $Z \sim \mathcal{N}(\mu, \sigma^2)$ be a random variable. In order to minimize the variance $\mathrm{var}_g\left[Q_Z^i\right]$ the optimal value of the importance sampling parameter β can be obtained by solving the quadratic equation:

$$\frac{\partial}{\partial \beta}\left(\mathrm{var}_g\left[Q_Z^i\right]\right) = 0$$

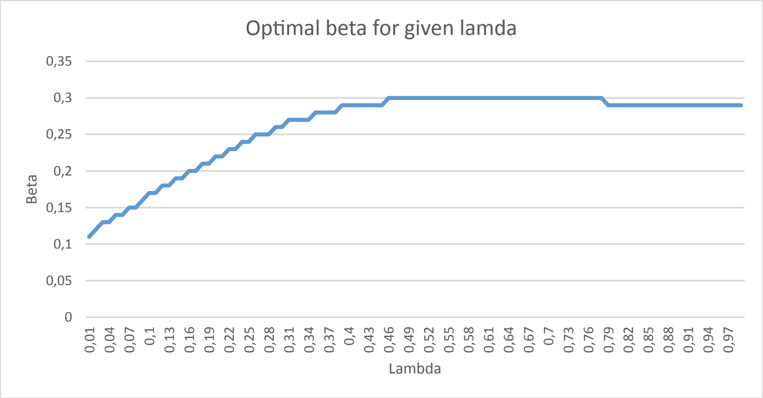
Normal distribution

$$
\begin{aligned}
\frac{\partial}{\partial\beta}(\ldots) =& \frac{1-\alpha}{(1-\beta)^2}(1-\lambda)^2(1-\alpha-u_Z\phi(u_Z)) \\
&- \frac{\alpha}{\beta^2}(1-\lambda)^2(\alpha+u_Z\phi(u_Z)) \\
&- \frac{\lambda^2}{\alpha\beta^2}\left(\alpha-u_Z\phi(u_Z)+u_Z^2\alpha\right) \\
&+ \lambda^2 u_Z^2\left(\frac{(1-\alpha)^2}{(1-\beta)^2}-\frac{\alpha^2}{\beta^2}\right) \\
&- 2\frac{\lambda(1-\lambda)\alpha}{\beta^2} + 2\lambda u_Z(1-\lambda)\phi(u_Z)\left(-\frac{\alpha}{\beta^2}-\frac{1-\alpha}{(1-\beta)^2}\right) \\
&+ 2\frac{\lambda^2}{\beta}u_Z(\phi(u_Z)-\alpha u_Z)
\end{aligned}
$$

Example - normal distribution with $\lambda = 0.5$



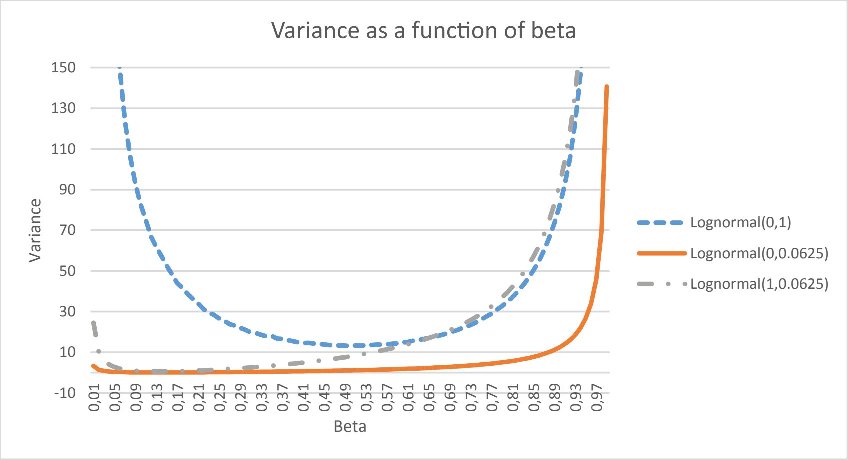
Example - normal distribution



Other distributions

- For other distributions, the same analysis can be performed and the derivative computed
- If this is not possible due to the complexity of the evaluations, we can estimate the suitable β by sampling
  - We choose a mesh of possible values, e.g. $\mathcal{B} = \{0.01, 0.02, \ldots, 0.99\}$
  - For each of them, we sample prescribed number of scenarios, Z^j
  - We compute the mean and variance of the values Q^j given by Z^j
  - The lowest variance is selected as a suitable choice of β
- In general, the solutions depend on the distribution parameters

Example - lognormal distribution



Risk-averse multistage model

- Inspired by Ruszczynski and Shapiro
- Given risk coefficients λ_t and random loss variable Z we define:

$$\rho_{t,\boldsymbol{\xi}_{[t-1]}}[Z] = (1-\lambda_t)\,\mathbb{E}\left[Z|\boldsymbol{\xi}_{[t-1]}\right] + \lambda_t\,\mathrm{CVaR}_{\alpha_t}\left[Z|\boldsymbol{\xi}_{[t-1]}\right]$$

- Nested model can be written:

$$\min_{\mathbf{A}_1\mathbf{x}_1=\mathbf{b}_1,\mathbf{x}_1\geq 0} \mathbf{c}_1^\top \mathbf{x}_1 + \rho_{2,\boldsymbol{\xi}_{[1]}}\Bigg[\min_{\mathbf{B}_2\mathbf{x}_1+\mathbf{A}_2\mathbf{x}_2=\mathbf{b}_2,\mathbf{x}_2\geq 0} \mathbf{c}_2^\top \mathbf{x}_2 + \cdots$$
$$\cdots + \rho_{T,\boldsymbol{\xi}_{[T-1]}}\Bigg[\min_{\mathbf{B}_T\mathbf{x}_{T-1}+\mathbf{A}_T\mathbf{x}_T=\mathbf{b}_T,\mathbf{x}_T\geq 0} \mathbf{c}_T^\top \mathbf{x}_T\Bigg]\Bigg]$$

- Convex optimization problem
- We assume feasibility, relatively complete recourse and finite optimal value

Model properties

- Allows to develop dynamic programming equations, using:

$$\mathrm{CVaR}_\alpha\left[Z\right] = \min_u \left[u + \frac{1}{\alpha}\mathbb{E}\left[Z - u\right]_+\right]$$

- Denote $Q_t(\mathbf{x}_{t-1}, \boldsymbol{\xi}_t)$, $t = 2, \ldots, T$ as the optimal value of:

$$\begin{aligned}
\min_{\mathbf{x}_t, u_t}\ & \mathbf{c}_t^\top \mathbf{x}_t + \lambda_{t+1} u_t + \mathcal{Q}_{t+1}(\mathbf{x}_t, u_t) \\
\text{s.t. } & \mathbf{B}_t \mathbf{x}_{t-1} + \mathbf{A}_t \mathbf{x}_t = \mathbf{b}_t \\
& \mathbf{x}_t \geq 0
\end{aligned}$$

- Recourse function is given by ($\mathcal{Q}_{T+1}(\cdot) \equiv 0$):

$$\begin{aligned}
\mathcal{Q}_{t+1}(\mathbf{x}_t, u_t) =& \mathbb{E}\Big[\left(1 - \lambda_{t+1}\right) Q_{t+1}(\mathbf{x}_t, \boldsymbol{\xi}_{t+1}) + \\
& + \frac{\lambda_{t+1}}{\alpha_{t+1}}\left[Q_{t+1}(\mathbf{x}_t, \boldsymbol{\xi}_{t+1}) - u_t\right]_+\Big]
\end{aligned}$$

Asset allocation model

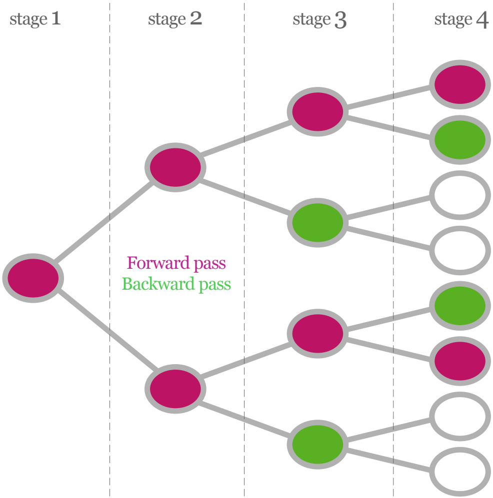
- At stage t we observe the price ratio between the new price and the old price $\mathbf{p}_t$
- $\mathbf{x}_t$ contains the optimal allocation (in USD, say)
- The total portfolio value is tracked as a multiple of the initial value
- Dynamic programming equations are very simple:

$$\begin{aligned} \min_{\mathbf{x}_t, u_t} \; & -\mathbf{1}^\top \mathbf{x}_t + \lambda_{t+1} u_t + \mathcal{Q}_{t+1}(\mathbf{x}_t, u_t) \\ \text{s.t. } & \mathbf{p}_t^\top \mathbf{x}_{t-1} - \mathbf{1}^\top \mathbf{x}_t = 0 \\ & \mathbf{x}_t \geq 0 \end{aligned}$$

SDDP algorithm properties

- First designed to solve hydro-scheduling problems
- Relies on the stage-independence assumption
- Each iteration runs with linear complexity
- Provides approximate solution using Benders' cuts
  - Cuts provide polyhedral approximation of the recourse function
  - LP duality - subgradient computed from the dual variables
  - Lower bound
- Policy evaluation procedure
  - Upper bound
- Upper bound requires estimation
  - Precise calculation fails to scale with T
  - Algorithm stops if lower bound is close enough to confidence interval for the upper bound
    - rarely done in a statistically rigorous manner

SDDP scheme



SDDP algorithm outline

- Because of the stage independence, cuts collected at any node from the stage t are valid for all nodes from that stage
- Algorithm consists of forward and backward iterations
- Forward iteration
  - Samples $\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^J$ sample paths
  - Policy is evaluated using all the cuts collected so far
  - Value of the policy gives the upper bound
- Backward iteration
  - Subset of the scenarios from the forward iteration is chosen
  - For every chosen node the Benders' cut is calculated
    - Using all of its immediate descendants (not just scenarios from the forward pass)
  - Optimal value of the root problem gives the lower bound
- The bounds are compared and the process is repeated

Inter-stage independence

- In order to use SDDP some form of independence is required
  - Efficient algorithms usually rely on an inter-stage independence assumption
  - Otherwise, memory issues arise even for modest number of stages
- This assumption can be weakened
  - One extension is to incorporate an additive dependence model
    - See Infanger & Morton [1996]
  - Another approach to bring dependence into the model is the use of a Markov chain in the model
    - See Philpott & Matos [2012]
  - Yet another approach couples a "small" scenario tree with general dependence structure with a second tree that SDDP can handle
    - See Rebennack et al. [2012]

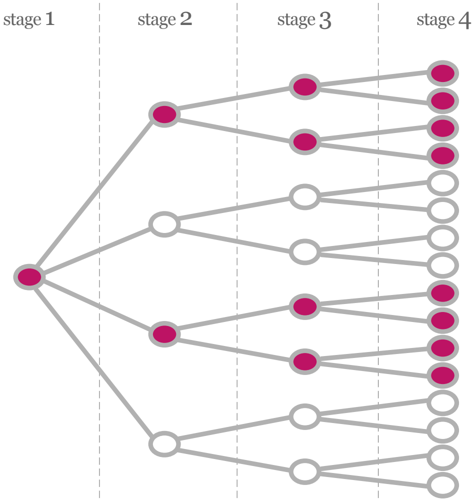
Upper bound overview

- Risk-neutral problems
  - The value of the current optimal policy can be estimated easily
  - Expectation at each node can be estimated by single chosen descendant
- Risk-averse problems
  - To estimate the CVaR value we need more descendants in practice
  - Leads to intractable estimators with exponential computational complexity
- Current solution (to our knowledge)
  - Run the risk-neutral version of the same problem and determine the number of iterations needed to stop the algorithm, then run the same number of iterations on the risk-averse problem
  - Inner approximation scheme proposed by Philpott et al. [2013]
    - Works with different policy than the outer approximation
    - Probably the best alternative so far

Our SDDP implementation

- Using own software developed in C++
- CPLEX and COIN-OR used as solvers for the LPs
- Stock assets allocation problem used as the example
- SDDP applied to a sampled tree from the continuous problem
- The algorithm can be implemented for parallel processing
  - We have not done so
- Testing data from US stock indices
- Log-normal distribution of returns is assumed
- Risk aversion coefficients set to $\lambda_t = \frac{t-1}{T}$
- Tail probability for CVaR set to 5% for all stages

Exponential estimator scheme



Exponential estimator

- Described by Shapiro
- For stages $t = 2, \ldots, T$, we form:

$$\hat{\mathfrak{v}}_t(\boldsymbol{\xi}^i_{t-1}) = \frac{1}{M_t} \sum_{j=1}^{M_t} \Bigg[(1-\lambda_t) \left((\mathbf{c}^j_t)^\top \mathbf{x}^j_t + \hat{\mathfrak{v}}_{t+1}(\boldsymbol{\xi}^j_t) \right) + \\ + \lambda_t u^i_{t-1} + \frac{\lambda_t}{\alpha_t} \left[(\mathbf{c}^j_t)^\top \mathbf{x}^j_t + \hat{\mathfrak{v}}_{t+1}(\boldsymbol{\xi}^j_t) - u^i_{t-1} \right]_+ \Bigg]$$

- $\hat{\mathfrak{v}}_{T+1}(\boldsymbol{\xi}^i_T) \equiv 0$
- The final cost is estimated by:

$$U^{\mathbf{e}} = (\mathbf{c}_1)^\top \mathbf{x}_1 + \hat{\mathfrak{v}}_2$$

Exponential estimator results

- Results for the exponential estimator:
  - $\sim 1,000$ LPs solved to obtain the estimator ($\sim 20,000$ for $T = 10$)
  - As number of stages grows so does bias (and variance)
  - $\underline{z}$ denotes the lower bound

T	desc./node	M_t	$\underline{z}$	$U^{\mathbf{e}}$ (s.d.)
2	50,000	1,000	-0.9518	-0.9518 (0.0019)
3	1,000	32	-1.8674	-1.8013 (0.0302)
4	100	11	-2.7811	-2.6027 (0.0883)
5	50	6	-3.6794	-2.9031 (0.5207)
10	50	3	-7.6394	1.5×10^7 (1.3×10^6)

Upper bound enhancements

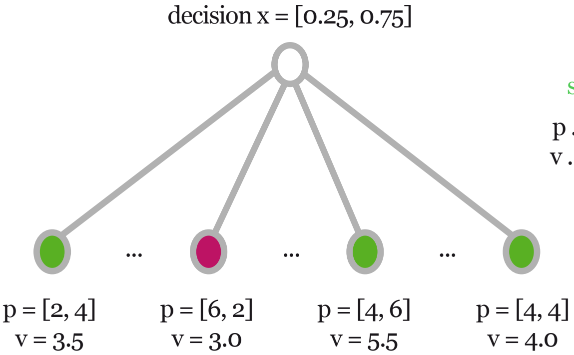
- We would like an estimator with linear complexity
- Ideally it should be unbiased, or in practice, have small bias
- We will incorporate two ideas:
  - Linear estimator from the risk-neutral case
  - Importance sampling, with an additional assumption needed

Assumption

Let $h_t(\mathbf{x}_{t-1}, \boldsymbol{\xi}_t)$ approximate the recourse value of our decisions $\mathbf{x}_{t-1}$ after the random parameters $\boldsymbol{\xi}_t$ have been observed, and let $h_t(\mathbf{x}_{t-1}, \boldsymbol{\xi}_t)$ be cheap to evaluate.

- For example in our portfolio model: $h_t(\mathbf{x}_{t-1}, \boldsymbol{\xi}_t) = -\boldsymbol{\xi}_t^\top \mathbf{x}_{t-1} = -\mathbf{p}_t^\top \mathbf{x}_{t-1}$

Importance sampling example



CVaR node
standard node

p .. price scenario
v .. portfolio value

Importance sampling

- We start with standard pmf, all probabilities equal for D_t scenarios:

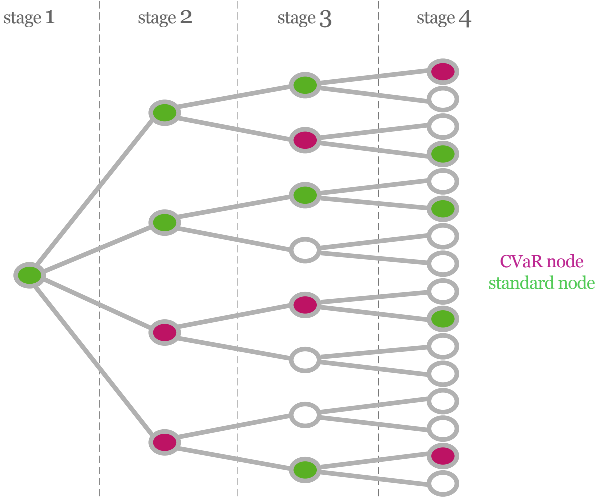
$$f_t(\boldsymbol{\xi}_t) = \frac{1}{D_t}\mathbb{I}\left[\boldsymbol{\xi}_t \in \left\{\boldsymbol{\xi}_t^1, \ldots, \boldsymbol{\xi}_t^{D_t}\right\}\right]$$

- We change the measure to put more weight to the CVaR nodes:

$$g_t(\boldsymbol{\xi}_t|\mathbf{x}_{t-1}) = \begin{cases} \dfrac{\beta_t}{\alpha_t} f_t, & \text{if } h_t \geq \mathrm{VaR}_{\alpha_t}\left[h_t(\mathbf{x}_{t-1}, \boldsymbol{\xi}_t)\right] \\ \dfrac{1-\beta_t}{1-\alpha_t} f_t, & \text{if } h_t < \mathrm{VaR}_{\alpha_t}\left[h_t(\mathbf{x}_{t-1}, \boldsymbol{\xi}_t)\right] \end{cases}$$

  - We select forward nodes according to this measure
  - $\mathbb{E}_{f_t}\left[Z\right] = \mathbb{E}_{g_t}\left[Z\,\frac{f_t}{g_t}\right]$
  - $w(\boldsymbol{\xi}^j) = \prod_{t=2}^{T} \frac{f_t(\boldsymbol{\xi}_t)}{g_t(\boldsymbol{\xi}_t|\mathbf{x}_{t-1})}$

Linear estimator scheme



Linear estimators

- The nodes can be selected randomly from the standard i.i.d. measure or from the importance sampling measure
- For stages $t = 2, \ldots, T$ is given by:

$$\hat{\mathfrak{v}}_t(\xi_{t-1}^{j_{t-1}}) = (1-\lambda_t)\left((\mathbf{c}_t^{j_t})^\top \mathbf{x}_t^{j_t} + \hat{\mathfrak{v}}_{t+1}(\xi_t^{j_t})\right) + \\ + \lambda_t u_{t-1}^{j_{t-1}} + \frac{\lambda_t}{\alpha_t}\left[(\mathbf{c}_t^{j_t})^\top \mathbf{x}_t^{j_t} + \hat{\mathfrak{v}}_{t+1}(\xi_t^{j_t}) - u_{t-1}^{j_{t-1}}\right]_+$$

- $\hat{\mathfrak{v}}_{T+1}(\xi_T^{j_T}) \equiv 0$
- Along a single path for scenario j the cost is estimated by:

$$\hat{\mathfrak{v}}(\xi^j) = \mathbf{c}_1^\top \mathbf{x}_1 + \hat{\mathfrak{v}}_2$$

Linear estimators

- For scenarios selected via the original pmf we have the naive estimator

$$U^{\mathbf{n}} = \frac{1}{M} \sum_{j=1}^{M} \hat{\mathfrak{v}}(\xi^j)$$

- With weights again defined via

$$w(\xi^j) = \prod_{t=2}^{T} \frac{f_t(\xi_t)}{g_t(\xi_t | \mathbf{x}_{t-1})}$$

- For scenarios selected via the IS pmf we have the IS estimator

$$U^{\mathbf{i}} = \frac{1}{\sum_{j=1}^{M} w(\xi^j)} \sum_{j=1}^{M} w(\xi^j) \hat{\mathfrak{v}}(\xi^j)$$

Linear estimator results

- Results for both linear estimators—with and without importance sampling ($\beta = 0.5$)
  - $\sim 1,000$ LPs solved to obtain the estimator, $\sim 10,000$ for $T = 10$
  - Still fails for bigger setups - for 10 stages the bias grows large

T	$\underline{z}$	$U^{\mathbf{n}}$ (s.d.)	$U^{\mathbf{i}}$ (s.d.)
2	-0.9518	-0.9515 (0.0020)	-0.9517 (0.0012)
3	-1.8674	-1.8300 (0.0145)	-1.8285 (0.0108)
4	-2.7811	-2.4041 (0.1472)	-2.3931 (0.1128)
5	-3.6794	-3.4608 (0.1031)	-3.4963 (0.1008)
10	-7.6394	9.3×10^4 (1.4×10^4)	9.0×10^4 (8.7×10^4)

Upper bound enhancements

- The reason for the bias of the estimator comes from poor estimates of CVaR
  - Once the cost estimate for stage t exceeds u_{t-1} the difference is multiplied by α_t^{-1}
  - When estimating stage $t-1$ costs in the nested model we sum stage $t-1$ costs and stage t estimate which means that we usually end up with costs greater than u_{t-2} so another multiplication occurs
  - This brings both bias and large variance

Assumption

For every stage $t = 2, \ldots, T$ and decision $\mathbf{x}_{t-1}$ the approximation function h_t satisfies:

$$Q_t \geq \mathrm{VaR}_{\alpha_t}[Q_t] \text{ if and only if } h_t \geq \mathrm{VaR}_{\alpha_t}[h_t].$$

Improved estimator

- Provided that the equivalence assumption holds we can reduce the bias of the estimator
  - The positive part operator in the equation is used only in the case of CVaR node
- For stages $t = 2, \ldots, T$ we have

$$\hat{\mathfrak{v}}^{\mathbf{h}}_t(\xi^{j_{t-1}}_{t-1}) = (1-\lambda_t)\left((\mathbf{c}^{j_t}_t)^\top \mathbf{x}^{j_t}_t + \hat{\mathfrak{v}}^{\mathbf{h}}_{t+1}(\xi^{j_t}_t)\right) + \lambda_t u^{j_{t-1}}_{t-1} +$$
$$+ \mathbb{I}[h_t > \mathrm{VaR}_{\alpha_t}[h_t]] \frac{\lambda_t}{\alpha_t}\left[(\mathbf{c}^{j_t}_t)^\top \mathbf{x}^{j_t}_t + \hat{\mathfrak{v}}^{\mathbf{h}}_{t+1}(\xi^{j_t}_t) - u^{j_{t-1}}_{t-1}\right]_+$$

- $\hat{\mathfrak{v}}^{\mathbf{h}}_{T+1}(\xi^{j_T}_T) \equiv 0$
- $$U^{\mathbf{h}} = \frac{1}{\sum_{j=1}^M w(\xi^j)} \sum_{j=1}^M w(\xi^j)\hat{\mathfrak{v}}^{\mathbf{h}}(\xi^j)$$

Improved estimator results

- Results compared to exponential estimator

T	$\underline{z}$	$U^{\mathbf{e}}$ (s.d.)	$U^{\mathbf{h}}$ (s.d.)
2	-0.9518	-0.9518 (0.0019)	-0.9517 (0.0011)
3	-1.8674	-1.8013 (0.0302)	-1.8656 (0.0060)
4	-2.7811	-2.6027 (0.0883)	-2.7764 (0.0126)
5	-3.6794	-2.9031 (0.5207)	-3.6731 (0.0303)
10	-7.6394	NA	-7.5465 (0.2562)
15	-11.5188	NA	-11.0148 (0.6658)

  - For problems with up to 5 stages $\sim 1,000$ LPs solved
  - For 10 stages $10,000$ LPs, for 15 stages $50,000$ LPs
  - We test challenging instances in terms of risk coefficients λ_t

Improved estimator results

T	$\underline{z}$	$U^{\mathbf{n}}$ (s.d.)	$U^{\mathbf{i}}$ (s.d.)	$U^{\mathbf{h}}$ (s.d.)	$U^{\mathbf{e}}$ (s.d.)
2	-0.9518	-0.9515 (0.0020)	-0.9517 (0.0012)	-0.9517 (0.0011)	-0.9518 (0.0019)
3	-1.8674	-1.8300 (0.0145)	-1.8285 (0.0108)	-1.8656 (0.0060)	-1.8013 (0.0302)
4	-2.7811	-2.4041 (0.1472)	-2.3931 (0.1128)	-2.7764 (0.0126)	-2.6027 (0.0883)
5	-3.6794	-3.4608 (0.1031)	-3.4963 (0.1008)	-3.6731 (0.0303)	-2.9031 (0.5207)
10	-7.6394	9.3×10^4 (1.4×10^4)	9.0×10^4 (8.7×10^4)	-7.5465 (0.2562)	1.5×10^7 (1.3×10^6)
15	-11.5188	NA	NA	-11.0148 (0.6658)	NA

- For $T = 2, \ldots, 5$ variance reduction of $U^{\mathbf{h}}$ relative to $U^{\mathbf{e}}$: 3 to 25 to 50 to 300.
- Computation time for $U^{\mathbf{n}}$ for $T = 5, 10, 15$: 8.7 sec. to 31.6 sec. to 67.4 sec.
- Computation time for $U^{\mathbf{h}}$ for $T = 5, 10, 15$: 6.8 sec. to 30.0 sec. to 66.5 sec.

Computational setup for variance reduction

- Risk aversion coefficients set to $\lambda_t = \frac{1}{2}$
- Tail probability for CVaR set to 5% for all stages
- We formed 100 i.i.d. replicates of the estimators with approx. $10,000$ LPs solved for each of them
- All 100 replicates used the same single run of SDDP
- Large-scale problems, $\mathrm{T} = 5$; 10 and 15
- 50 descendant scenarios per node

Suitable β

- Our random inputs are supposed to have log-normal distribution
- The portfolio value is a sum of log-normal distributions
  - We don't have exact analytical form of the resulting distribution
  - It's sometimes approximated with log-normal distribution
- But, what does the convex combination of expectation and CVaR do with the distribution?
- Nested structure of the model brings additional complex transformations
- Different values of β should be selected for every stage, as the parameters of the distributions also vary
- For small ratios of standard deviation over the mean, log-normal distribution can be approximated by normal distribution, see Hald [1952]
- We have used $\beta = 0.3$ which came out from our normal-distribution analysis for $\lambda = 0.5$

Results

- Standard Monte Carlo setup $\hat{\mathcal{Q}}^s$ ($\beta_t = \alpha_t = 0.05$)
- Improved estimator $\hat{\mathcal{Q}}^i$ with $\beta_t = 0.3$
- Lower bound $\underline{z}$

T	total scenarios	$\underline{z}$	$\hat{\mathcal{Q}}^s$ (s.d.)	$\hat{\mathcal{Q}}^i$ (s.d.)
5	$6,250,000$	-3.5212	-3.5166 (0.0168)	-3.5158 (0.0042)
10	$\approx 10^{14}$	-7.3885	-7.2833 (0.2120)	-7.2741 (0.0315)
15	$\approx 10^{25}$	-10.4060	-10.1482 (0.8184)	-10.1246 (0,1266)

- Variance reduction by a factor between 4 and 7
- Negligible effect on computation times

Conclusion

- We propose a new approach to estimate functionals that incorporate risk via CVaR
  - Allows to tweak existing procedures which rely on sampling in estimation of mean-risk objectives
  - Significantly smaller variance than a standard Monte Carlo estimator
  - Negligible effect on computation times in optimization problems
- Future research
  - Applications such as hydroelectric scheduling under inflow uncertainty
  - Other risk measures, different importance sampling pdfs

References

- HALD, A. (1952): *Statistical Theory with Engineering Applications*, John Wiley & Sons, New York
- INFANGER, G. and MORTON, D. P. (1996): *Cut sharing for multistage stochastic linear programs with interstage dependency*, Mathematical Programming 75 pp. 241-256.
- KOZMÍK, V. and MORTON, D. (2013): *Risk-averse Stochastic Dual Dynamic Programming*, Optimization Online
- PEREIRA, M. V. F. and PINTO, L. M. V. G. (1991): *Multi-stage stochastic optimization applied to energy planning*, Mathematical Programming 52, pp. 359–375. pp. 63-72.

References

- PHILPOTT, A. B., DE MATOS, V. L.: *Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion*. Eur. J. of Oper. Res. 218, pp. 470–483 (2012)
- REBENNACK, S., FLACH, B., PEREIRA, M. V. F., PARDALOS, P. M.: *Stochastic hydro-thermal scheduling under* CO_2 *emissions constraints*. IEEE Transactions on Power Systems 27, pp. 58–68 (2012)
- RUSZCZYNSKI, A. and SHAPIRO, A. (2006): *Conditional risk mappings*, Mathematics of Operations Research 31, pp. 544–561
- SHAPIRO, A. (2011): *Analysis of stochastic dual dynamic programming method*, European Journal of Operational Research 209, pp. 63-72.

Conclusion

Thank you for your attention!

Václav Kozmík
vaclav@kozmik.cz