Stochastic Scenario Generation

Sebastiano Vitali

Faculty of Mathematics and Physics
Charles University in Prague

Faculty of Management, Economics and Quantitative Methods
University of Bergamo
Suppose that we face an optimization problem in which some elements of the constraints, in this case $T$ and $h$, depend on the stochastic variable $\xi \in \Xi \subseteq \mathbb{R}^n$.

\begin{align*}
\min_x & \quad c^\top x \\
\text{s.t.} & \quad Ax = b \\
& \quad T(\xi)x = h(\xi) \\
& \quad x \geq 0
\end{align*}

$x$ is the decision variable and we must decide $x$ before knowing the realization $\tilde{\xi}$ of the stochastic variable $\xi$. 
To deal with this kind of problems we define a method to insert a penalty in the objective function with respect to the realization of the stochastic variable taking into account the error on the constraint.

Let’s define $W \in \mathbb{R}^{r \times m}$ called (fixed) recourse matrix and $y(\tilde{\xi}) \in \mathbb{R}^m$ called recourse vector.

Now let’s define the compensation vector:

$$Wy(\tilde{\xi}) = h(\tilde{\xi}) - T(\tilde{\xi})x$$

And the recourse function:

$$Q(x, \tilde{\xi}) = \min_y \left\{ q(\tilde{\xi})^T y : Wy = h(\tilde{\xi}) - T(\tilde{\xi})x; y \geq 0 \right\}$$

Where $q(\tilde{\xi})$ is the cost of the introduction of the recourse components.
To deal with this kind of problems we define a method to insert a penalty in the objective function with respect to the realization of the stochastic variable taking into account the error on the constraint.

Let’s define $W \in \mathbb{R}^{r \times m}$ called \textbf{(fixed) recourse matrix} and $y(\tilde{\xi}) \in \mathbb{R}^{m}$ called \textbf{recourse vector}.

Now let’s define the compensation vector:

$$Wy(\tilde{\xi}) = h(\tilde{\xi}) - T(\tilde{\xi})x$$

And the recourse function:

$$Q(x, \tilde{\xi}) = \min_y \left\{ q(\tilde{\xi})^\top y : Wy = h(\tilde{\xi}) - T(\tilde{\xi})x; y \geq 0 \right\}$$

Where $q(\tilde{\xi})$ is the cost of the introduction of the recourse components.
Now we can add this penalty function (in term of expected value) directly in the objective function and the problem becomes:

\[
\begin{align*}
\min_x & \quad c^\top x + E_\xi Q(x, \xi) \\
\text{s.t.} & \quad Ax = b \\
Q(x, \tilde{\xi}) = & \quad \min_y \left\{ q(\tilde{\xi})^\top y : Wy = h(\tilde{\xi}) - T(\tilde{\xi})x ; y \geq 0 \right\} \\
x & \geq 0
\end{align*}
\]

This formulation is called \textbf{Implicit representation} of the stochastic linear program. See Birge and Louveaux (1997) or Shapiro and Philpott (2007) or Shapiro, Dentcheva and Ruszczyński (2009) or Kall and Mayer (2005).
If we discretize the randomness using scenarios we can reformulate the problem in the following way:

\[
\min_x \quad \mathbf{c}^\top x + \sum_{k=1}^{K} p^k \mathbf{q}^k \mathbf{y}^k
\]

\[
s.t. \quad A\mathbf{x} = \mathbf{b}
\]

\[
T^1 \mathbf{x} + W\mathbf{y}^1 = \mathbf{h}^1
\]

\[
T^2 \mathbf{x} + W\mathbf{y}^2 = \mathbf{h}^2 \quad x \geq 0, y^k \geq 0, k = 1, ..., K
\]

\[
: \quad T^K \mathbf{x} + W\mathbf{y}^K = \mathbf{h}^K
\]

where \( \mathbf{q}^k, T^k, \mathbf{h}^k \) are coefficients depending by scenario \( \zeta^k \) and \( \sum_k p^k = 1 \).

This formulation is called **Scenario-Based representation** of the stochastic linear program.

See Dupačová, Hurt and Štěpán (2002)
Scenario generation can be used in all the fields that need to manage uncertainty:

- **Investment Strategy** evaluation
  (see Ziemba and Ziemba (2007) or Mulvey (1996))
- Bond portfolio **immunization** (see Hiller and Schaack (1990)):
  - Cash-flows matching (Dedication Technique)
  - Duration matching
- Immunization evaluation using **Tracking System**
  (see Dembo (1991))
- **Energy** production (Greenhall (2013))
- **Logistics problems** (Powell and Topaloglu (2003))
- **Not Plain Vanilla Option** pricing
- **Insurance Products** pricing
- Bond (with relevant **default risk**) pricing
Mainly we have three sources of data or combinations thereof:

- Historical Data
- Simulation
- Expert Opinion
  - Theoretical model
A good scenario generation method should:

- **Influence** the solution the least possible
- Let the solution **converge** to the true optima as the number of scenario increases
- Be as possible good for a given number of scenarios
Conditional Sampling

At each step we can sample several values from the stochastic process \( \{\xi_t\} \) in two ways:

- By an explicit formula that describes the process evolution:
  \[ \tilde{\xi}_{t+1} = z(\tilde{\xi}_t, \tilde{e}) \], having a sample \( \tilde{e} \)
- By sampling directly from the distribution of \( \xi_t \)

The problems of this methods are:

- We must know the real distribution of \( \xi_t \)
- If we need to sample a random vector we could just sample separately each component so they result independent
At each step we can **sample** several values from the stochastic process \( \{\xi_t\} \) in two ways:

- By an explicit formula that describes the process evolution:
  \[
  \tilde{\xi}_{t+1} = z(\tilde{\xi}_t, \tilde{e}), \text{ having a sample } \tilde{e}
  \]
- By sampling directly from the distribution of \( \xi_t \)

The problems of this methods are:

- We must know the **real distribution** of \( \xi_t \)
- If we need to sample a random vector we could just sample separately each component so they result **independent**
This is an **intuitive evolution** of the previous method.

- It takes into account the **correlations** between the different random variables.
- Clearly it needs the specification of each **marginal distribution** and of a **correlation matrix**.

This method is described in Cario and Nelson (1997), Deler and Nelson (2001) and Lurie and Goldberg (1998).
The **Sample Average Approximation** is a particular case of Conditional Sampling. It is very well discussed by several authors such as Kleywegt, Shapiro and Homem-de-Mello (2001).

This method assumes that each sample $\xi_j$ with $j = 1, \ldots, N$ has the **same probability** $p_j = \frac{1}{N}$. This hypothesis is not necessarily true, we must look to the real distribution.
Conditional Sampling - SAA
Suppose we have an original unsolvable problem:

\[
\min_{x \in X} F(x, \xi)
\]

We replace it with the approximate problem:

\[
\min_{x \in X} F(x, \eta)
\]

If we generate \(k\) samples \(\tilde{\eta}_k\), Pflug (2001) suggests to measure the error between the original problem and each approximated problem with:

\[
e_f(\xi, \tilde{\eta}_k) = F(\arg\min_x F(x, \tilde{\eta}_k); \xi) - F(\arg\min_x F(x, \xi); \xi)
\]

and proofs that under certain Lipschitz conditions

\[
e_f(\xi, \tilde{\eta}_k) \leq 2 \sup_x |F(x, \tilde{\eta}_k) - F(x, \xi)| \leq 2Ld(\tilde{\eta}_k, \xi)
\]

where \(d(\tilde{\eta}_k, \xi)\) is the Wasserstein-distance between the distributions \(\tilde{\eta}_k\) and \(\xi\).
So if we find a distribution $\tilde{\eta}_k^*$ that minimize the distance $d(\tilde{\eta}_k^*, \xi)$ we automatically minimize the error.

Using this method we know that:

- Whole trajectory is generated at once.
- The scenario set is optimal in a specified sense.
- It works only for univariate processes.
Handling Multi-Period Scenario

There are several ways to deal with **Multi-Period Scenario**.

The correct one depends by:

- the **structure** of the problem
- the **distribution assumptions**
  - the **dependence** or **independence** hypothesis between the different periods
Indepence

- The **distribution does not change** period by period
- The distribution variation is **time independent**.

Dependence

- The dependence is modeled using a stochastic process as ARMA, ARCH, GARCH, mean reverting, etc.
- We can add Poisson jumps component, stochastic variance, etc.
Handling Multi-Period Tree

**Independence**
- The distribution does not change period by period
- The distribution variation is time independent.

**Dependence**
- The dependence is modeled using a stochastic process as ARMA, ARCH, GARCH, mean reverting, etc.
- We can add Poisson jumps component, stochastic variance, etc.
Path-based methods

We start generating complete paths by evolving the stochastic process.

- We don’t obtain a tree but a set of paths (called *fan*).
- We need to cluster together the scenarios to obtain a tree and/or to reduce the dimension of the scenario set.
- The evolution process can be modeled in several ways.

This method is described in Dupačová, Consigli and Wallace (2000).
Path-based methods

The idea of a non tree structure paths set is well known in finance. We present an overview of financial scenario generation using the so called MonteCarlo technique.

- In this case we don’t need a stage approach.
- Each path is independent by the other.
- Simply each path start at a certain point in the space and in the time and lives until the maturity we have fixed.
- It’s very important to specify the distribution of the returns of the asset we want to model.
- The model can be defined simply by a distribution or can be built adding several contributions:
  - trend
  - seasonality
  - Poisson jump process
  - a mix of all, etc.
Monte Carlo

Geometric Brownian Motion
Monte Carlo
Geometric Brownian Motion plus Seasonality
MonteCarlo
Geometric Brownian Motion plus Poisson Jumps
Monte Carlo

- If we just want to verify the **performance** of a given portfolio we are quite free to use the risk model we prefer.

- In the case that we are **pricing** an asset we must be more careful: for example we could need that the process is a martingale.

- If we would like to simulate scenarios for two or more assets jointly we must take into account the **correlation structure**.

- If the risk model is Gaussian the Cholesky decomposition is enough to generate a correlated simulation, otherwise we need a **copula approach**.

- If we need to use the path set in a stochastic programming problem we need to **cluster** paths to obtain a tree structure.
Monte Carlo

- If we just want to verify the performance of a given portfolio we are quite free to use the risk model we prefer.
- In the case that we are pricing an asset we must be more careful: for example we could need that the process is a martingale.
- If we would like to simulate scenarios for two or more assets jointly we must take into account the correlation structure.
- If the risk model is Gaussian the Cholesky decomposition is enough to generate a correlated simulation, otherwise we need a copula approach.
- If we need to use the path set in a stochastic programming problem we need to cluster paths to obtain a tree structure.
Monte Carlo

- If we just want to verify the performance of a given portfolio we are quite free to use the risk model we prefer.

- In the case that we are pricing an asset we must be more careful: for example we could need that the process is a martingale.

- If we would like to simulate scenarios for two or more assets jointly we must take into account the correlation structure.

- If the risk model is Gaussian the Cholesky decomposition is enough to generate a correlated simulation, otherwise we need a copula approach.

- If we need to use the path set in a stochastic programming problem we need to cluster paths to obtain a tree structure.
In case we know just some properties, Kaut and Wallace (2003) proposed the Moment Matching method.

The method consists in building the trajectories in such a way that a set of properties of the distribution at each period is matched.

- We don’t need to know the entire distribution.
- We just need to estimate the requested statistics.
- Increasing the number of scenarios we don’t converge to the true distribution.
- In case we know the distribution using this method we lose information.
- Two distributions can be absolutely different but share the same moments.
Moment Matching

(Pflug, Lecture notes, PhD Winterschool 2013)
Once we have composed the scenario set we could want to reduce its cardinality. A scenario reduction method is proposed in Dupačová et al. (2003) and in Heitsch and Römisch (2003).

- The method could be **forward** or **backward**.
- Can be applied to **single scenarios** or to **scenario subsets**.
- Suggests which scenario to delete and how to redistribute its **probability** among the other scenarios.
Scenario Reduction
Scenario Reduction
Suppose we have an original unsolvable problem:

\[
\min_{x \in X} F(x, \xi)
\]

We replace it with the approximate problem:

\[
\min_{x \in X} F(x, \tilde{\eta})
\]

If we generate \( k \) sample \( \tilde{\eta}_k \), suppose each leads to solution:

\[
x^*_k = \arg\min_{x \in X} F(x, \tilde{\eta}_k)
\]

Pflug (2001) suggests to measure the error between the original problem and each approximated problem with:

\[
e_f(\xi, \tilde{\eta}_k) = F(\arg\min_{x \in X} F(x, \tilde{\eta}_k); \xi) - F(\arg\min_{x \in X} F(x, \xi); \xi)
\]

\[
= F(x^*_k, \xi) - \min_{x} F(x, \xi) \geq 0
\]
Stability

In-sample-stability

We compare the behavior of the optimal solution of the original problem according to different scenario samples:

\[
\min_x F(x, \tilde{\eta}_k) \approx \min_x F(x, \tilde{\eta}_l)
\]

Out-sample-stability

We compare the behavior of the optimal solutions of the approximated problems with the true distribution:

\[
F(x_k^*, \xi) \approx F(x_l^*, \xi)
\]
In-sample-stability

We compare the behavior of the optimal solution of the original problem according to different scenario samples:

\[
\min_x F(x, \tilde{\eta}_k) \approx \min_x F(x, \tilde{\eta}_l)
\]

Out-sample-stability

We compare the behavior of the optimal solutions of the approximated problems with the true distribution:

\[
F(x^*_k, \xi) \approx F(x^*_l, \xi)
\]
To price a **Plain Vanilla Option** when we assume that the underlying returns are **Gaussian** distributed we can use Black and Scholes formula.

In other cases we need different approaches:

- **Lattice structure approach**
  For example for American Option
- **MonteCarlo approach**
  For example to price options with jumping underlying
To price a **Plain Vanilla Option** when we assume that the underlying returns are **Gaussian** distributed we can use Black and Scholes formula.

In other cases we need different approaches:

- **Lattice structure approach**
  For example for American Option
- **MonteCarlo approach**
  For example to price options with jumping underlying
For example, if we have an American put option we can use a **recombining binomial lattice** to approximate the future value of the underlying.

Methodology steps:

1. **estimate parameters** according to the returns distribution assumption
2. **estimate probability** according to risk neutral hypothesis for a given risk free rate
3. **generate** the lattice
4. **price** the option with a backward procedure
Binomial lattice approach

We assume that the distribution is Gaussian and by a MLE we obtain these annual estimations:

$$\mu = 0.328$$

$$\sigma = 0.1789$$

In a risk neutral world for a trimestral horizon we obtain:

$$u = e^{\sigma/\sqrt{4}} = 1.094$$

$$d = e^{-\sigma/\sqrt{4}} = 0.914$$

(see Cox, Ross and Rubinstein (1979))

We assume an annual risk free rate $r_f = 0.03$ and then the up probability is given by:

$$p_u = \frac{\exp(r_f/4) - d}{u - d} = 0.519$$
Last we suppose that:

- the put is issued at the money
- the underlying stock price in the root is 100
- we have 5 periods with branching 2 and then in each period
  the underlying can:
  - increase with rate 9.35% with probability 0.5197,
  - decrease with rate 8.56% with probability 0.4803.
Binomial lattice approach

\[ P_T = \max(K - S_T, 0) \]

\[ P_t = \max((P_{t+1}^u \cdot p^u + P_{t+1}^d \cdot p^d) \cdot (1 + r)^{-1}, K - S_t) \]
Binomial lattice approach

\[ P_T = \max(K - S_T, 0) \]

\[ P_t = \max \left( \left( P_{t+1}^u \cdot p^u + P_{t+1}^d \cdot p^d \right) \cdot (1 + r)^{-1}, K - S_t \right) \]
MonteCarlo approach

- In case we want to price an option and the returns of the underling are **not Gaussian distributed** we can use a MonteCarlo approach.

- Methodology steps:
  1. **estimate parameters** according to the returns distribution assumption
  2. **generate** the trajectories with MonteCarlo in a risk neutral world
  3. **price** the option
MonteCarlo approach

For example we can assume that the underlying process contains also a jump Poisson process.

\[ S_t = S_{t-1}e^{(r_f - \frac{\sigma^2}{2})dt + \sigma \sqrt{dt} W_t} (1 + P_t) \]

The risk free rate is fixed at \( r_f = 0.035 \) and by a MLE we obtain these annual estimations:
- \( \mu = 0.174 \) expected return
- \( \sigma = 0.359 \) standard deviation
- \( \lambda = 30.05 \) average number of jumps
- \( \theta = 0.015 \) average of the lognormal jump size
- \( \nu = 0.053 \) standard deviation of the lognormal jump size

And now we generate the scenarios...
Bibliography


Bibliography


Bibliography

Bibliography


