Stochastic Optimization

Introduction and Stochastic Programming

Edoardo Fadda
Dip. di Scienze Matematiche – Politecnico di Torino
e-mail: edoardo.fadda@polito.it

Version: October 14, 2024

Introduction

- Stochastic Programming
- From Dynamic Programming to Reinforcement Learning

The goal of the course is practical.

Exam

The exam:

- 8 points
- Project on the topic of the course.
- You will be evaluated on the quality of the code, and on the analysis of the results that you get.
- You will submit the code and a small report, and we will have a short discussion.
- Group of max 4 people [More on assignment selection later].
- Create a github account (at least one per group).

Stochastic Optimization

Let us consider a function $F(\mathbf{x}, \omega)$, and consider it to be a profit (possibly depending by a demand).

Our goal is to compute

$$\max_{\mathbf{x} \in \mathcal{X}} \mathcal{R}[F(\mathbf{x}, \omega)]$$

where

- $\mathbf{x}$ is the vector of decision variables (e.g., the quantity to produce),
- $\mathcal{X}$ is called **feasibility space**,
- $F(\mathbf{x}, \omega)$ is a random variable defining, e.g., the profit,
- $\mathcal{R}$ is an operator mapping the random variable $F(\mathbf{x}, \omega)$ in $\mathbb{R}$.

Most of the time we will consider $\mathcal{R}[\cdot] = \mathbb{E}[\cdot]$.

Notice that $\max_{x \in \mathcal{X}} \mathcal{R}[F(x, \omega)] = -\min_{x \in \mathcal{X}} -\mathcal{R}[F(x, \omega)]$.

News-vendor problem

A newsvendor could purchase a given number of newspapers x each one with:

- A cost c.
- The selling price of each copy is p.

Demand $d = D(\omega)$ is a random variable with probability mass function $f(d)$.

The profit is

$$P(x, \omega) = p \min[D(\omega), x] - cx$$

Example: $x = 4, D(\omega) = 5 \rightarrow \ldots$

Maximizing $P(x, \omega)$ does not make sense.

We decide to maximize its expected value, i.e.,

$$\max_x \mathbb{E}[P(x, \omega)] = \max_x \mathbb{E}[p \min[D(\omega), x] - cx].$$

Example: if $p = 10$, $c = 1$, and:

$$\mathbb{P}[D = d] = \begin{cases} 0 \text{ with } \pi = 0.4 \\ 1 \text{ with } \pi = 0.3 \\ 2 \text{ with } \pi = 0.2 \\ 3 \text{ with } \pi = 0.1 \end{cases}$$

Note that $\mathbb{E}[D] = 1$. Let us solve this problem...

N.B.: The optimal solution is not the expected value of newspapers. Why?

Notice that we were lucky on the distribution of D. For example, if $D \sim \mathcal{B}in(10^6, 0.01)$ computing

$$\mathbb{E}[p \min[D(\omega), x] - cx]$$

is not easy. The problem is to compute $\mathbb{E}[p \min[D(\omega), x]$.

The goal of Monte Carlo methods is to approximate

$$\mathbb{E}[h(X)]$$

If X has dimension 1 we can approximate it using standard methods integrating on a grid. Same holds for dimension 2. Nevertheless, when the dimension of X grows, we have the *course of dimensionality*.

If the dimension of X prevent using standard integration, and if we are able to sample from X, we can use the following algorithm:

Algorithm 1 Monte Carlo Integration

1: generate $x_1, x_2, \ldots, x_N \sim X$
2: compute $h(x_1), h(x_2), \ldots, h(x_N)$
3: estimate $\mathbb{E}[h(X)]$ by

$$\bar{h}_N = \frac{1}{N} \sum_{i=1}^{N} h(x_i)$$

It works because of the *law of large numbers* and the *central limit theorem*.

For the *law of large numbers* we know that if $N \to +\infty$, we expect that $x_1, x_2, \ldots, x_N$ becomes representative of f. In other words, the empirical c.d.f.

$$\widehat{F}_N(t) \xrightarrow{\text{a.s.}} F(t)$$

As a result, $\bar{h}_N$ which is based on our sample will also converge to $\mathbb{E}[h(X)]$ as $N \to +\infty$. In formula,

$$\frac{1}{N}\sum_{i=1}^{N} h(x_i) \to \mathbb{E}[h(X)] \qquad \text{as} \qquad N \to +\infty.$$

Moreover, we have that $\bar{h}_N$ is an unbiased estimator of $\mathbb{E}[h(X)]$ since:

$$\mathbb{E}[\bar{h}_N] = \mathbb{E}[\frac{1}{N}\sum_{i=1}^{N} h(X_i)] = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}[h(X_i)] = \mathbb{E}[h(X)].$$

The variance of the estimator is:

$$\mathrm{Var}[\bar{h}_N] = \mathrm{Var}[\frac{1}{N}\sum_{i=1}^{N} h(X_i)] = \frac{1}{N^2}\mathrm{Var}[\sum_{i=1}^{N} h(X_i)] = \frac{1}{N^2}\sum_{i=1}^{N} \mathrm{Var}[h(X_i)] = \frac{1}{N}\mathrm{Var}[h(X)]$$

If we do not know Var$[h(X)]$, we can use again MC as:

$$\text{Var}[h(X)] = \mathbb{E}[(h(X) - \mathbb{E}[h(X)])^2] \approx \frac{1}{N}\sum_{i=1}^{N}(h(x_i) - \mathbb{E}[h(X)])^2 \approx \frac{1}{N}\sum_{i=1}^{N}(h(x_i) - \bar{h}_n)^2$$

Therefore, we get

$$\text{Var}[\bar{h}_N] \approx v_N \doteq \frac{1}{N}\left(\frac{1}{N}\sum_{i=1}^{N}(h(x_i) - \bar{h}_N)^2\right)$$

Because $\bar{h}_N$ is a sample mean of $h(X)$, we can use the CLT to approximate its sampling distribution.

The CLT states that if $\bar{y}$ is a sample mean, its sampling distribution can be approximated with a normal distribution with mean μ and variance $\frac{1}{n}s^2$, where $\mu = \mathbb{E}[Y]$ and $s^2 = \text{Var}[Y]$. Therefore:

$$\bar{h}_N \sim \mathcal{N}(\mathbb{E}[h(X)], \frac{1}{n}\text{Var}[h(X)])$$

Knowing that $\bar{h}_N$ is normally distribute allows us to create confidence bounds. In formula, we have the α-percentage confidence that the true value is in the interval:

$$\left[\bar{h}_N - z_{1-\frac{\alpha}{2}}\sqrt{\frac{v_N}{N}}, \bar{h}_N + z_{1-\frac{\alpha}{2}}\sqrt{\frac{v_N}{N}}\right],$$

where $z_{1-\frac{\alpha}{2}}$ is the $1-\frac{\alpha}{2}$ quantile of the standard normal distribution (to be precise...).

N.B. there is nothing probabilistic about the single confidence interval after the data have been obtained and the interval's endpoints have been given numerical values. The correct interpretation to give to the confidence interval is as follows : If one constructs a very large number of independent $100(1-\alpha)$ percent confidence intervals, each based on n observations, where n is sufficiently large, the proportion of these confidence intervals that contain (cover) μ should be $1-\alpha$.

How many random values do we need to sample to achieve a certain level of accuracy in our estimate?

The width of the interval is:

$$2z_{1-\frac{\alpha}{2}}\sqrt{\frac{v_N}{N}} = w \iff N = \frac{v_N 4z^2}{w^2}$$

To answer this, we first need to know the variance of our estimate (v_N).

Let us see it in practice...

What if we are not able to sample according to $f(x)$ or it is inefficient? There are several techniques (e.g., acceptance rejection). We will focus on *Importance Sampling*.

The idea of importance sampling is to sample values from another distribution $g(x)$, more friendly, such that the support of $f(x)$ has to be within the support of $g(x)$ (i.e., $g(x) > 0$ for all the x in the support of $f(x)$).

Then, we can do the following:

$$\mathbb{E}_f[h(X)] = \int_{\mathcal{X}} h(x)f(x)dx = \int_{\mathcal{X}} h(x)f(x)\frac{g(x)}{g(x)}dx = \int_{\mathcal{X}} h(x)f\frac{f(x)}{g(x)}g(x)dx = \mathbb{E}_g[h(X)\frac{f(X)}{g(X)}]$$

Algorithm 2 Importance Sampling

1: generate $x_1, x_2, \ldots, x_N \sim X$
2: compute $h(x_1)\frac{f(x_1)}{g(x_1)}, h(x_2)\frac{f(x_2)}{g(x_2)}, \ldots, h(x_N)\frac{f(x_N)}{g(x_N)}$
3: estimate $\mathbb{E}[h(X)]$ using

$$\frac{1}{N}\sum_{i=1}^{N} h(x_i)\frac{f(x_i)}{g(x_i)}$$

You can interpret $\frac{f(x_i)}{g(x_i)}$ as a weight of the sample collected.

Example:

We want to estimate the probability that Z (a standard Normal distribution) is over 5 (the real answer is $1 - F(5)$).

If we want to approximate this using Monte Carlo integration, we would sample many values from the normal distribution and see how many are larger than 5.

Because $\mathbb{P}[Z > 5]$ is so rare, it happens only 2 times in a million. This is not a very good estimate of the true probability.

We can improve our accuracy by sampling many more values, or we can improve our performance using importance sampling.

We can consider generating values from $g(x)$, an exponential that begins at 5 ($\mathcal{X} = [5, +\infty)$), i.e., $g(x) = e^{-(x-4.5)}$ The weights are then:

$$w(x_i) = \frac{f(x_i)}{g(x_i)} = \frac{\frac{1}{\sqrt{2\pi}} e^{-x_i^2/2}}{e^{-(x_i - 4.5)}}$$

Random Number Generation

Generating good random numbers is very important. Luckily nice algorithms are available, but is important to use them correctly.

N.B.: The central key is to being able to sample from a $[0, 1]$ uniform distribution.

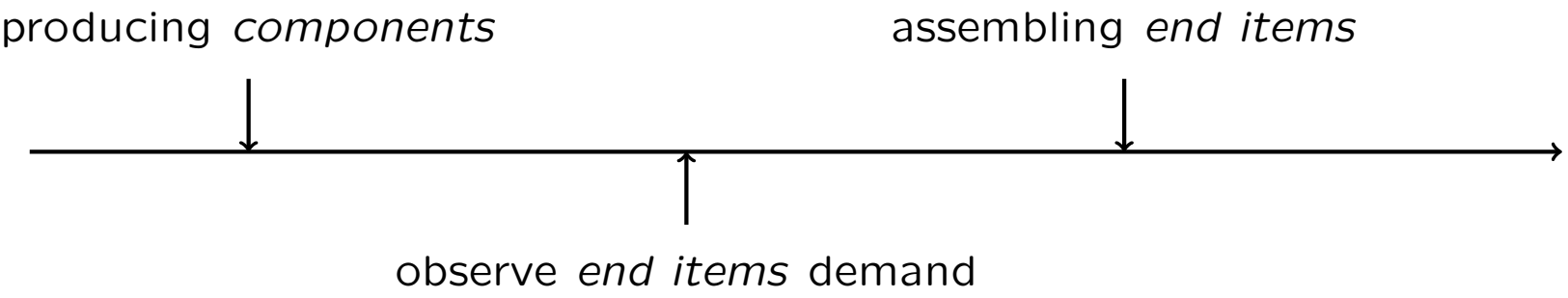
The simplest algorithm to generate random numbers is the *Linear Congruential Generators*:

$$Z_i = (aZ_{i-1} + c)\%m.$$

Given the first number (called Z_0) all the sequence of numbers is fixed. Z_0 is called *seed*.

Assemble To Order

- Demand uncertainty is one of the main difficulties of production planning.
- One possibility to hedge against demand uncertainty is to postpone product differentiation.
- *Assemble-to-order* (ATO): the manufacturer stocks *components* and assembles them into the final *item* when a customers place orders.



- ATO is used in the production of pizzas, cars, fashion, etc.

Note that while the news-vendor cannot adapt to uncertainty, in ATO we can.

- $\mathcal{I} = \{1, \ldots, I\}$ set of components;

- $\mathcal{J} = \{1, \ldots, J\}$ set of items;
- $\mathcal{M} = \{1, \ldots, M\}$ set of machines.
- C_i cost of component $i \in \mathcal{I}$;
- P_j selling price of the item $j \in \mathcal{J}$;
- L_m time availability of machine $m \in \mathcal{M}$;
- T_{im} time required to produce $i \in \mathcal{I}$ in $m \in \mathcal{M}$;
- G_{ij} amount of components $i \in \mathcal{I}$ required for assembling item $j \in \mathcal{J}$, commonly known as *gozinto factor*;
- $d_j(\omega)$ demand for item $j \in \mathcal{J}$.

Decision variables:

- $x_i \in \mathbb{R}^I$ the amount of component $i \in \mathcal{I}$ produced;
- $y_j(\omega) \in \mathbb{R}^J$ the amount of items $j \in \mathcal{J}$. It is a random variable, why?

Model:

$$\begin{aligned}
\max_{x\in\mathbb{R}^I, y(\omega)\in\mathbb{R}^J} \quad & -\sum_{i\in\mathcal{I}} C_i x_i + \mathbb{E}\big[\sum_{j\in\mathcal{J}} P_j y_j(\omega)\big] & \\
\text{s.t.} \quad & \sum_{i\in\mathcal{I}} T_{im} x_i \le L_m & \forall m \in \mathcal{M} \\
& y_j(\omega) \le d_j(\omega) & \forall j \in \mathcal{J}\ \forall \omega \in \Omega \\
& \sum_{j\in\mathcal{J}} G_{ij} y_j(\omega) \le x_i & \forall i \in \mathcal{I}\ \forall \omega \in \Omega \\
& y_j(\omega), x_i \ge 0 & \forall i \in \mathcal{I}\ \forall j \in \mathcal{J}\ \forall \omega \in \Omega
\end{aligned}$$

Example

Two types of pizza: Margherita, 4 seasons.

Three ingredients: dough, tomato source, vegetables

One pizza maker:

$$C = [1, 1, 3]€\,,\ P = [6, 8.5]€\,,\ T = [0.5, 0.25, 0.25]h,\ L = 6h$$

$$G = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Different scenarios:

$$\begin{cases} d_1 = 100 \; d_2 = 400 & \pi = 0.3 \\ d_1 = 90 \; d_2 = 100 & \pi = 0.1 \\ d_1 = 80 \; d_2 = 250 & \pi = 0.6 \end{cases}$$

$$\begin{aligned} \max \quad & -x_1 - x_2 - 3x_3 + 0.3(6y_1^1 + 8.5y_2^1) + \dots \\ \text{s.t.} \quad & \sum_{i \in \mathcal{I}} 0.5x_1 + 0.25x_2 + 0.25x_3 \leq 6 \\ & y_1^1 \leq 100 \\ & \dots \\ & y_2^3 \leq 250 \\ & y_1^1 + y_2^1 \leq x_1 \\ & \dots \end{aligned}$$

N.B. In the ATO problem we are just interested in the x variables, the *here-and-now* solution. We do not care about the y, the *recourse actions*..

If the integral is not easy to solve, we can approximate by sampling. Therefore, we define a set of *scenarios* $\mathcal{S} = 1, \dots, S$ each one with its own probability π_s (intuitively $\frac{1}{S}$). Then we can write the Sample Average Approximation of the model as:

- $\mathcal{S} = \{1, \dots, S\}$ set of possible outcomes or *scenarios*.
- d_j^s: demand for item $j \in \mathcal{J}$ in scenario $s \in \mathcal{S}$;
- π^s: probability of scenario s.

$$
\begin{aligned}
\max_{x\in\mathbb{R}^I, y\in\mathbb{R}^J} \quad & -\sum_{i\in\mathcal{I}} C_i x_i + \sum_{s\in\mathcal{S}} \pi^s \sum_{j\in\mathcal{J}} P_j y_j^s & \\
\text{s.t.} \quad & \sum_{i\in\mathcal{I}} T_{im} x_i \leq L_m & \forall m \in \mathcal{M} \\
& y_j^s \leq d_j^s & \forall j \in \mathcal{J}\ \forall s \in \mathcal{S} \\
& \sum_{j\in\mathcal{J}} G_{ij} y_j^s \leq x_i & \forall i \in \mathcal{I}\ \forall s \in \mathcal{S} \\
& y_j^s, x_i \geq 0 & \forall i \in \mathcal{I}\ \forall j \in \mathcal{J}\ \forall s \in \mathcal{S}
\end{aligned}
$$

N.B.: we are able to solve this problem because we trade a complex integral for a big model.

Exercise: Implement this model.

Two stage linear stochastic programming problems

The classical two-stage linear stochastic programming problems can be formulated as

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & c^T x + E_\xi[Q(x,\xi)] \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array} \tag{1}$$

where $Q(x,\xi)$ is the optimal value of the second-stage problem

$$\begin{array}{ll} \min\limits_{y \in \mathbb{R}^m} & q(\xi)^T y \\ \text{subject to} & T(\xi)x + W(\xi)y = h(\xi) \\ & y \geq 0 \end{array} \tag{2}$$

In such formulation $x \in \mathbb{R}^n$ is the first-stage decision variable vector $y \in \mathbb{R}^m$ is the second-stage decision variable vector.

Using the SAA, we get:

$$\begin{array}{lll} \min\limits_{x \in \mathbb{R}^n, y \in \mathbb{R}^m} & c^T x + \sum_{s=1}^{S} \pi_s q_s^T y_s & \\ \text{subject to} & Ax = b & \\ & T_s x + W_s y_s = h_s & \forall s \in \mathcal{S} \\ & x \geq 0 & \\ & y_s \geq 0 & \forall s \in \mathcal{S} \end{array} \tag{3}$$

How many scenarios I need to have a reliable solution for the problem?

Two main concepts:

- In-Sample stability
- Out-of-Sample stability

In-Sample stability: given two sets of scenarios $\mathcal{S}$ and $\mathcal{S}'$, with **the same cardinality**, we want that:

$$\min_{x \in \mathcal{X}} f(x|\mathcal{S}) \approx \min_{x \in \mathcal{X}} f(x|\mathcal{S}')$$

To achieve *In-Sample* stability we increase the number of scenarios until the objective function (computed using two different scenario trees of the same size) is almost the same.

Algorithm 3 Compute In-Sample stability

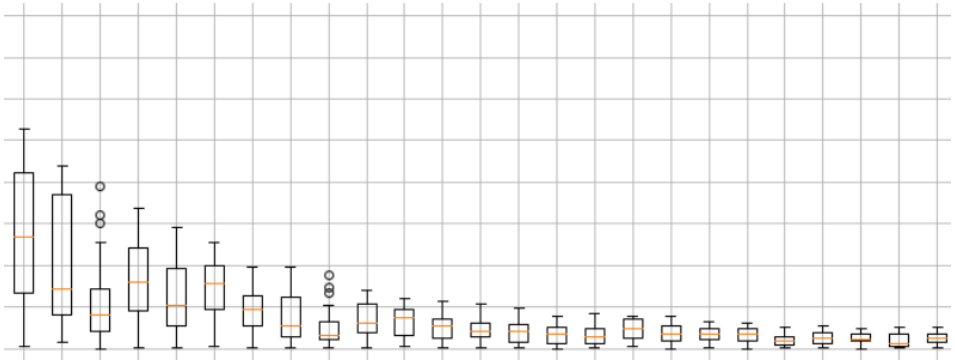
1: initialize `n_scenario`
2: generate $\mathcal{S}$ and $\mathcal{S}'$ both with cardinality `n_scenario`
3: **while** stopping condition **do**
4: increase `n_scenario`
5: generate $\mathcal{S}$ and $\mathcal{S}'$
6: **end while**

What about the stopping condition?

Since $\mathcal{S}$ and $\mathcal{S}'$ are independent, then also $\min_{x\in\mathcal{X}} f(x|\mathcal{S})$ and $\min_{x\in\mathcal{X}} f(x|\mathcal{S}')$ are independent. Therefore, we can compute:

$$\phi_i \doteq \min_{x\in\mathcal{X}} f(x|\mathcal{S}_i) - \min_{x\in\mathcal{X}} f(x|\mathcal{S}'_i),$$

for $i = 1, \ldots, N$. Then, $\frac{1}{N}\sum_{i=1}^{N} \phi_i$ satisfy the CLT conditions and we can stop when $0 \in [\mu - z_\alpha \frac{\sigma}{\sqrt{N}}, \mu + z_\alpha \frac{\sigma}{\sqrt{N}}]$ for a given α.



Out-of-sample stability: given a scenario tree used to compute the solution (call it $\mathcal{S}$) and a far greater scenario tree $\mathcal{E}$, we want that, by calling $x^* = \arg\min_x f(x|\mathcal{S})$:

$$f(x^*|\mathcal{S}) \approx f(x^*|\mathcal{E})$$

Algorithm 4 Compute Out-of-Sample stability

1: initialize `n_scenario`
2: generate $\mathcal{S}$ with `n_scenario` and $\mathcal{E}$ with `big_n_scenario`
3: **while** stopping condition **do**
4: increase `n_scenario`
5: generate $\mathcal{S}$
6: **end while**

Let us do it for the newsvendor problem...

Exercise: Compute in-sample and out-of-sample stability for the ATO problem.

Indicators

Two stage problems may require a lot of computational effort. Therefore different indicators have been developed to help in the analysis of the model.

To introduce these models, let us consider the general formulation of a two stage recourse problem:

$$\min_{x\in X} \mathbb{E}_\xi[z(\mathbf{x},\xi)] = \min_{\mathbf{x}\in X} f_1(\mathbf{x}) + \mathbb{E}_\xi[f_2(\mathbf{x},\xi)]$$

with $f_2(\mathbf{x},\xi) = \min_{y\in Y(\mathbf{x},\xi)} f_2(y|\mathbf{x},\xi)$. Its optimal value is $RP = \mathbb{E}_\xi[z(\mathbf{x}^*,\xi)]$

We call the *wait-and-see*:

$$WS = \mathbb{E}_\xi[z(\bar{\mathbf{x}}(\xi),\xi)]$$

Notice that $\bar{\mathbf{x}}(\xi)$ depend on ξ.

Example: consider the newsvendor:

$$\max_x \mathbb{E}[p\min[D(\omega),x] - cx] \approx \max_x \sum_{s=1}^{S} \pi_s[p\min[D_s,x] - cx]$$

the wait and see is:

$$\max_{x_s} \sum_{s=1}^{S} \pi_s[p\min[D_s,x_s] - cx_s] = \sum_{s=1}^{S} \pi_s \max_{x_s}[p\min[D_s,x_s] - cx_s]$$

with the obvious solution $x_s = D_s\ \forall s$.

We call *Expected Value of the Perfect Information* (EVPI):

$$RP - EVPI(\text{for min}) \qquad EVPI - RP(\text{for max})$$

We call *expected value solution:*

$$\bar{\mathbf{x}} \in \arg\min_{\mathbf{x}\in X} f_1(\mathbf{x}) + f_2(\mathbf{x}, \mathbb{E}_\xi[\xi]).$$

We call the *expected value of the expected value solution*:

$$EEV = \mathbb{E}_\xi[z(\bar{\mathbf{x}}, \xi)].$$

Example: consider the newsvendor:

$$\max_x \mathbb{E}[p\min[D(\omega), x] - cx] \approx \max_x \sum_{s=1}^{S} [p\min[D_s, x] - cx]$$

the expected value solution is:

$$\max_x [p\min[\sum_{s=1}^{S} \pi_s D_s, x_s] - cx_s]$$

with the obvious solution $x = \sum_{s=1}^{S} \pi_s D_s$.

We call Value of Stochastic Solution (VSS):

$$RP - EEV(\text{for min}) \qquad EEV - RP(\text{for max}).$$

Let us compute them for the newsvendor problem...

- Rough Monte Carlo
- Moment matching
- Minimizing Wasserstain distance

Moment Matching: Our goal is to have a distribution that mimic the moments (mean, variance, skewness, and kurtosis) of a given distribution. For example, given μ and σ^2, we can solve:

$$\min \quad |\sum_{i=1}^{S} \pi_s x_s - \mu| + \lambda |\sum_{i=1}^{S} \pi_s (x_s - \mu)^2 - \sigma^2| \tag{4}$$

$$\text{s.t.} \quad \sum_{i=1}^{S} \pi_s = 1 \tag{5}$$

$$x_s \in \mathcal{X} \qquad \forall\ s \tag{6}$$

$$\pi_s \in [0, 1] \qquad \forall\ s. \tag{7}$$

Minimizing Wasserstain distance Given two probability distributions μ and ν on a metric space X, the Wasserstein distance quantifies how much "work" is required to move the probability mass in μ to match ν. The work is computed based on a

cost function that measures the effort required to transport probability mass between points in X.

In the discrete case, the Wasserstein distance can be formulated as an optimization problem. Let us consider the one dimensional case and let μ and ν be two discrete probability distributions defined on finite sets $\{x_1, x_2, \ldots, x_m\}$ and $\{y_1, y_2, \ldots, y_n\}$, respectively.

The cost of transporting probability mass from x_i to y_j is denoted as c_{ij}, where c_{ij} is a known non-negative cost (typically, the Euclidean distance or another metric between x_i and y_j).

Let γ_{ij} represent the amount of probability mass transported from x_i to y_j. The goal is to find a transport plan γ that minimizes the total transport cost while ensuring the mass constraints are respected.

$$\min \quad \sum_{i=1}^{m}\sum_{j=1}^{n} d(x_i, y_j)\gamma_{ij} \tag{8}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} \gamma_{ij} = \mu_i \qquad \forall i = 1, \ldots, m \tag{9}$$

$$\sum_{i=1}^{m} \gamma_{ij} = \nu_j \qquad \forall\ j = 1, \ldots, n \tag{10}$$

$$\gamma_{ij} \geq 0 \in [0, 1] \quad \forall\ i = 1, \ldots, m, j = 1, \ldots, n. \tag{11}$$

It is possible to use this concept in several heuristic ways to reduce the dimension of a scenario tree. For example:

- Start from $S_1 = \{x_1, x_2, \ldots, x_m\}$, remove some of the observations with some policy and get $S_2 \subseteq S_1$;

- Solve the model:

$$\min \quad \sum_{i=1}^{m}\sum_{j=1}^{n} d(x_i, y_j)\gamma_{ij} \tag{12}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} \gamma_{ij} = \mu_i \qquad \forall i = 1, \ldots, m \tag{13}$$

$$\sum_{i=1}^{m} \gamma_{ij} = \nu_j \qquad \forall\ j = 1, \ldots, n \tag{14}$$

$$\sum_{i=1}^{m} \nu_j = 1 \tag{15}$$

$$\gamma_{ij} \geq 0 \in [0, 1] \qquad \forall\ i = 1, \ldots, m, j = 1, \ldots, n. \tag{16}$$