

Dynamic Generation of Scenario Trees

Georg Ch. Pflug* and Alois Pichler†

August 5, 2014

Abstract

We present new algorithms for the dynamic generation of scenario trees for multistage stochastic optimization. The different methods described are based on random vectors, which are drawn from conditional distributions given the past and on sample trajectories.

The structure of the tree is not determined beforehand, but dynamically adapted to meet a distance criterion, which insures the quality of the approximation. The criterion is built on transportation theory, which is extended to stochastic processes.

Keywords: Decision trees, stochastic optimization, optimal transportation

Classification: 90C15, 60B05, 62P05

1 Introduction

Scenario trees are the basic data structure for multistage stochastic optimization problems. They are discretizations of stochastic processes and therefore an approximation to real phenomena. In this paper we describe general algorithms, which approximate the underlying stochastic process with an arbitrary, prescribed precision.

The traditional way *from data to tree models* is as follows:

- (i) Historical time series data are collected,
- (ii) a parametric model is specified for the probability law which governs the data process,
- (iii) the parameters are estimated on the basis of the observations (and possibly some additional information),
- (iv) future scenarios are generated according to the identified probability laws, and finally
- (v) these scenarios are concentrated in a tree, typically by stepwise reduction.

In the last step a concept of closeness of scenarios and similarity between the simulated paths and the tree has to be used. Some authors use as a criterion for similarity the coincidence of moments (cf. Wallace et al., [12, 14]), others use distance concepts such as the squared norm and a filtration distance (cf. Heitsch and Römisch and others, [5, 7, 8, 10, 9, 11]).

*University of Vienna. Department of Statistics and Operations Research.
International Institute for Applied Systems Analysis (IIASA), Laxenburg, Austria.

†Norwegian University of Science and Technology, NTNU. The author gratefully acknowledges support of the Research Council of Norway (grant 207690/E20).

It has been shown in Pflug and Pichler [20] that an appropriate distance concept for stochastic processes and trees is given by the nested distance (see Definition 18 below). The relevant theorem for multistage stochastic optimization (cited as Theorem 19 below) is extended and simplified for the particular case of pushforward measures (Theorem 14). Based on transportation theory this paper presents in addition theorems which are particularly designed to extract scenario trees by employing techniques, which are adopted from stochastic approximation.

A related issue is the choice of the tree topology, or the branching structure of the scenario tree: how bushy and how big should the approximating tree be in order not to exceed a given, maximal discretization error? The cited papers do not address this problem. The scenario reduction methods of Heitsch and Römisch are inspired by their work on squared norm and filtration distances, but do not give explicit error bounds.

We propose here a new way *from data to tree models* as follows:

- (i) as above, historical time series data are collected,
- (ii) a simulator has to be provided, which allows sampling trajectories or from all conditional distributions of the estimated data process,
- (iii) a threshold for the maximum discretization error has to be specified, then
- (iv) our algorithms generate a tree with automatically chosen topology and maximal chosen discretization error.

The algorithms apply to stochastic processes with higher dimensional state space too, for which an appropriate distance has to be chosen. This is of relevance and importance in many economic applications.

Outline of the paper. The next section (Section 2) recalls the notion of transportation distances on probability spaces. Section 3 provides the mathematical basis for algorithms to approximate probability measures. These algorithms are based on stochastic approximation. Section 4 generalizes the results to stochastic processes and gives the related theorems for stochastic programming based on transportation distance. Section 5 introduces the nested distance and generalizes the results from transportation distances to the nested distance. Further, this section explains how scenario trees can be extracted from a set of trajectories of the underlying process. A series of examples demonstrates that it is possible to extract useful scenario trees even from a sample, which is smaller than the nodes of the scenario tree. In Section 6 we discuss the relevance of the algorithms and conclude.

2 Approximation of probability measures

It has been elaborated in a sequence of publications that the nested distance is an appropriate concept to provide a distance for stochastic processes, the basic theorem is provided below (Theorem 14). The nested distance is built on the transportation distance (sometimes also Wasserstein, or Katorovich distance), which is a distance for probability measures.

Definition 1 (Transportation distance for probability measures). Assume that P (\tilde{P} , resp.) are probability measures on probability spaces Ξ ($\tilde{\Xi}$, resp.), such that for $\xi \in \Xi$ and $\tilde{\xi} \in \tilde{\Xi}$ a *distance*

$d(\xi, \tilde{\xi})$ is defined. To the metric d one may associate the pertaining *Wasserstein distance of order* $r \geq 1$ (also called *Kantorovich distance*) of probability measures by

$$d_r(P, \tilde{P}) := \inf \left\{ \left(\iint_{\Xi \times \tilde{\Xi}} d(\xi, \tilde{\xi})^r \pi(d\xi, d\tilde{\xi}) \right)^{1/r} \mid \begin{array}{l} \pi \text{ is a probability measure on } \Xi \times \tilde{\Xi} \\ \text{with marginal distributions } P \text{ and } \tilde{P} \end{array} \right\}. \quad (1)$$

The methods we develop in what follows consider a special probability measure \tilde{P} , which is associated with the initial measure P by a transport map.

2.1 Transport maps

A particular situation arises if the second probability measure \tilde{P} is a pushforward measure (or image measure) of P for some *transport map*¹ $T : \Xi \rightarrow \tilde{\Xi}$ linking the spaces, $\tilde{P} = P^T := P \circ T^{-1}$. Then an upper bound for the Wasserstein distance is given by

$$d_r(P, P^T)^r \leq \int_{\Xi} d(\xi, T(\xi))^r P(d\xi), \quad (2)$$

because the bivariate measure

$$\pi_T(A \times B) := P(A \cap T^{-1}(B)) \quad (3)$$

associated with T has the marginals required in (1).

The situation $\tilde{P} = P \circ T^{-1}$ naturally arises in approximations, where the outcome ξ is approximated by $T(\xi)$. Notice that if $T(\xi)$ is a close approximation of ξ , then $d(\xi, T(\xi))$ is small and the integral in (2) is small as well, which makes P^T an approximation of interest for P .

The upper bound (2) is useful in many respects. First, the measure π_T is computationally much easier to handle than a solution of (1), because the integral in (2) is just over Ξ , and not over the product $\Xi \times \tilde{\Xi}$ as in (1). Further, for $r = 2$, Brenier's polar factorization theorem [2, 3] implies that the optimal transport plan π solving (1) has the general form (3) for some measure preserving map T , such that involving a transport map is not restrictive. Finally, the transport map allows a generalization to stochastic processes which we address in Section 4.2.

2.2 Single-period Wasserstein distance minimization.

Assume that P and \tilde{P} are probabilities on $\Xi = \mathbb{R}^m$, which is endowed with the distance

$$d(\xi, \tilde{\xi}).$$

To the distance d one may associate the pertaining Wasserstein-distance according to (1) in Definition 1. Our goal is to approximate P by the "best" discrete multivariate distribution \tilde{P} sitting on s points $z^{(1)}, \dots, z^{(s)}$ in the sense that the transportation distance $d_r(P, \tilde{P})$ is minimized.

Given a collection of points $Z = (z^{(1)}, \dots, z^{(s)})$, which can be seen as a $m \times s$ matrix, introduce the *Voronoi partition* $\mathcal{V}_Z = \{V_Z^{(i)} : i = 1, \dots, s\}$ of \mathbb{R}^m , where

$$V_Z^{(i)} = \left\{ \xi \in \mathbb{R}^m \mid \begin{array}{l} d(\xi, z^{(i)}) = \min_j d(\xi, z^{(j)}) \text{ and} \\ d(\xi, z^{(k)}) > \min_j d(\xi, z^{(j)}) \text{ for } k < i \end{array} \right\}$$

¹Often also transport plan.

such that²

$$\bigsqcup_{i \in \{1, \dots, s\}} V_Z^{(i)} = \mathbb{R}^m.$$

For a given probability P on \mathbb{R}^m we use the notation P_Z for the discrete distribution sitting on the points of the set Z with masses $P(V_Z^{(i)})$, i.e.,

$$P_Z = \sum_{i=1}^s P(V_Z^{(i)}) \cdot \delta_{z^{(i)}}.$$

Remark 2. Notice that the measure P_Z is induced by the plan T , $P_Z = P^T$, where $T : \Xi \rightarrow Z \subset \Xi$ is the transport map

$$T(\xi) := z^{(i)}, \text{ if } \xi \in V_Z^{(i)}.$$

For a fixed P let

$$D(Z) := \int_{\Xi} \min_{i=1, \dots, s} d(\xi, z^{(i)})^r P(d\xi) = \sum_{i=1}^s \int_{V_Z^{(i)}} d(\xi, z^{(i)})^r P(d\xi). \quad (4)$$

Then

$$\begin{aligned} D(Z)^{1/r} &= \min \{d_r(P, \bar{P}) : \bar{P}(Z) = 1\} \\ &= \min \{d_r(P, \bar{P}) : \bar{P} \text{ sits on the points of the set } Z\} = d_r(P, P_Z), \end{aligned} \quad (5)$$

such that $D(Z)$ measures the quality of the approximation of P , which can be achieved by probability measures with supporting points Z (cf. [6, Lemma 3.4]).

Facility location. The approximation problem is thus reduced to finding the best point set Z (the facility location problem). We discuss in the next section three algorithms for solving this minimization problem:

- (i) A deterministic iteration procedure, which is applicable, if the necessary integrations with respect to P can be carried out numerically.
- (ii) A *stochastic approximation* procedure, which is based on a sample from P and which converges to a local minimum of D .
- (iii) A branch-and-bound procedure, which is also based on a sample from P and which converges to a global minimum of D .

3 Algorithms to approximate probability measures

Before introducing the algorithms we mention the differentiability properties of the mapping $Z \mapsto D(Z)$. This is useful as the first order conditions of optimality for (5) require the derivatives to vanish.

²The disjoint union $\bigsqcup_i V_i$ symbolizes that the sets are pairwise disjoint, $V_i \cap V_j = \emptyset$, whenever $i \neq j$.

Let $\nabla D(Z)$ be the $m \times s$ matrix with column vector $\nabla_{z^{(i)}} D(Z)$ given by the formal derivative

$$\int_{V_Z^{(i)}} r \mathbf{d}(\xi, z^{(i)})^{r-1} \cdot \nabla_{\xi} \mathbf{d}(\xi, z^{(i)}) P(d\xi), \quad i = 1, \dots, s \quad (6)$$

of (4).

Proposition 3. *If P has a Lebesgue density g , then $Z \mapsto D(Z)$ is differentiable and the derivative is $\nabla D(Z)$.*

If there is a monotonically decreasing, integrable function k such that $g(\xi) \leq k(\|\xi\|_1)$, then $Z \mapsto \nabla D(Z)$ is Lipschitz.

Proof. Notice first that by convexity of the distance \mathbf{d} the gradient $\nabla_{\xi} \mathbf{d}(\xi, z^{(i)})$ in the integral (6) is uniquely determined except on a countable set. As P has a Lebesgue density the exception set has measure zero and the integral is well defined.

That (6) is indeed the derivative, as well as the second assertion follow by standard means or from Pflug [16, Corollary 3.52, page 184]. \square

3.1 The deterministic iteration

We start with a well known cluster algorithm for partitioning a larger set of points in \mathbb{R}^m into s clusters. Algorithm 1 is a typical example of an algorithm, which clusters a given set of points into subsets of small intermediate distance. While some authors use such type of algorithms to generate representative scenarios, we use it only to find good starting configurations for the following optimization algorithm.

Algorithm 1

A typical hierarchical cluster algorithm (complete linkage)

- (i) **Sampling.** Suppose that n points $\{z^{(1)}, \dots, z^{(n)}\}$ in \mathbb{R}^m endowed with metric \mathbf{d} is given. The set $Z = \{z^{(i)} : i = 1, \dots, n\}$ is iteratively partitioned into disjoint clusters, such that their number decreases from step to step. At the beginning, each point is a cluster of itself.
- (ii) **Iteration.** Suppose that the current partition of the set is $\mathbb{R}^m = \bigsqcup_j C_j$. Find the pair of clusters (C_j, C_k) for which

$$\sup \{\mathbf{d}(z, z') : z \in C_j, z' \in C_k\}$$

is minimal. Create a new cluster by merging C_j and C_k .

- (iii) **Stopping criterion.** If the number of clusters has decreased to the desired number s , then stop. Otherwise goto (ii).
-

The subsequent single-period algorithm (Algorithm 2) requires integration with respect to P , as well as nonlinear optimizations to be carried out numerically. Since this is a difficult task, especially for higher dimensions, we present an alternative algorithm based on stochastic algorithm later.

Algorithm 2

Optimal discretization of probability P by a discrete probability sitting on s points: a deterministic, but numerically difficult algorithm

- (i) **Initialization.** Set $k = 0$ and start with an arbitrary point set $Z(0) = \{z^{(i)} : i = 1, \dots, s\}$. It is advisable to choose the initial point set according to a cluster algorithm, e.g., to use Algorithm 1 to find clusters and then start with the cluster medians.
- (ii) **Voronoi partition.** Find the Voronoi sets $V_{Z(k)}^{(i)}$ for $1 \leq i \leq s$.
- (iii) **Optimization step.** For all i compute the *center of order r* , i.e., let

$$z^{(i)}(k+1) \in \operatorname{argmin}_y \left\{ \int_{V_{Z(k)}^{(i)}} d(\xi, y)^r P(d\xi) \right\} \quad (7)$$

and form the new set $Z(k+1) = \{z^{(i)}(k+1) : i = 1, \dots, s\}$.

- (iv) **Integration step.** Calculate $D(Z(k+1))$. Stop, if $D(Z(k+1)) \geq D(Z(k))$; otherwise set $k := k+1$ and goto (ii).
-

Remark 4. To compute the argmin in the optimization step (iii) of Algorithm 2 is in general difficult. However, there are two important cases.

- (i) Whenever \mathbb{R}^m is endowed with the weighted Euclidean metric $d(\xi, \tilde{\xi})^2 = \sum_{j=1}^m w_j |\xi_j - \tilde{\xi}_j|^2$ and the order of the Wasserstein distance is $r = 2$, then the argmin is known to be the conditional barycenter, i.e., the conditional expected value

$$z^{(i)}(k+1) = \frac{1}{P(V_{Z(k)}^{(i)})} \int_{V_{Z(k)}^{(i)}} \xi P(d\xi). \quad (8)$$

This is an explicit formula, which is easily available in many situations of practical relevance. Computing (8) instead of (7) may significantly accelerate the algorithm.

- (ii) Whenever \mathbb{R}^m is endowed with the weighted ℓ^1 -metric $d(\xi, \tilde{\xi}) = \sum_{j=1}^m w_j |\xi_j - \tilde{\xi}_j|$, then $z^{(i)}(k+1)$ in (7) is the componentwise median of the probability P restricted to $V_{Z(k)}^{(i)}$. In general and in contrast to the Euclidean metric, no closed form is available here.

Remark 5 (Initialization). Whenever the probability measure is a measure on \mathbb{R}^1 with cumulative distribution function (cdf) G , then the quantiles

$$z^{(i)}(0) := G^{-1} \left(\frac{i-1/2}{s} \right) \quad (i = 1, 2, \dots, s)$$

can be chosen as initial points for the Wasserstein distance in (i) of Algorithm 2. These points are optimal for the Kolmogorov distance, that is, they minimize $\sup_{z \in \mathbb{R}} |P((-\infty, z]) - \hat{P}_n((-\infty, z])|$ for the measure P and the empirical measure $\hat{P}_n = \frac{1}{n} \sum_{i=1}^n \delta_{z^{(i)}}$.

For the Wasserstein distance of order $r \geq 1$ even better choices are

$$z^{(i)}(0) = G_r^{-1} \left(\frac{i - 1/2}{s} \right) \quad (i = 1, 2, \dots, s),$$

where G_r is the cdf with density $g_r \sim g^{1/1+r}$ (provided that a density g is available). This is derived in Graf and Luschgy [6, Theorem 7.5]. Their result is even more general and states that the optimal points $z^{(i)}$ have asymptotically density

$$g_r = \frac{g^{m/(m+r)}}{\int g^{m/(m+r)},}$$

whenever the initial probability measure P on \mathbb{R}^m has density g .

The following proposition addresses the convergence of the deterministic iteration algorithm.

Proposition 6. *If $Z(k)$ is the sequence of point sets generated by the deterministic iteration algorithm (Algorithm 2), then*

$$D(Z(k+1)) \leq D(Z(k)).$$

If $D(Z(k^+1)) = D(Z(k^*))$ for some k^* , then $D(Z(k)) = D(Z(k^*))$ for all $k \geq k^*$ and*

$$\nabla_{z^{(i)}} D(Z(k^*)) = 0$$

for all i .

Proof. Notice that

$$\begin{aligned} D(Z(k)) &= \int_{\Xi} \min_j d(\xi, z^{(j)})^r P(d\xi) = \sum_{i=1}^s \int_{V_Z^{(i)}} d(\xi, z^{(i)}(k))^r P(d\xi) \\ &\geq \sum_{i=1}^s \int_{V_Z^{(i)}} d(\xi, z^{(i)}(k+1))^r P(d\xi) = \int_{\Xi} \min_j d(\xi, z^{(j)}(k+1))^r P(d\xi) \\ &= D(Z(k+1)). \end{aligned}$$

If $D(Z(k^*+1)) = D(Z(k^*))$, then necessarily, for all i ,

$$z_j^{(i)}(k) \in \operatorname{argmin}_y \left\{ \int_{V_Z^{(i)}} d(\xi, y)^r P(d\xi) \right\},$$

which is equivalent to

$$\int_{V_Z^{(i)}} r d(\xi, z^{(i)})^{r-1} \cdot \nabla_{\xi} d(\xi, z^{(i)}) P(d\xi) = 0 \text{ for all } i$$

by Proposition 3. Hence $\nabla_Z D(Z(k^*)) = 0$ and evidently, the iteration has reached a fixed point. \square

We remark here that the method outlined in Algorithm 2 is related to the k-means method of cluster analysis (see, e.g., McQueen [15]).

3.2 Stochastic approximation

Now we describe how one can avoid the optimization and integration steps of Algorithm 2 by employing stochastic approximation to compute the centers of order r . The stochastic approximation algorithm (Algorithm 3) requires that we can sample an independent, identically distributed (i.i.d.) sequence

$$\xi(1), \dots, \xi(n),$$

of vectors of arbitrary length n , each distributed according to P .³

Proposition 7. *Suppose that $\mathfrak{F} = (\mathcal{F}_1, \mathcal{F}_2 \dots)$ is a filtration and (Y_k) is a sequence of random variables, which are uniformly bounded from below and adapted to \mathfrak{F} . In addition, let (A_k) and (B_k) be sequences of nonnegative random variables also adapted to \mathfrak{F} . If $\sum_k B_k < \infty$ a.s. and the recursion*

$$\mathbb{E}[Y_{k+1} | \mathcal{F}_k] \leq Y_k - A_k + B_k \quad (9)$$

is satisfied, then Y_k converges and $\sum_k A_k < \infty$ almost surely.

Proof. Let $S_k := \sum_{\ell=1}^k B_\ell$ and $T_k := \sum_{\ell=1}^k A_\ell$. Then (9) implies that

$$\mathbb{E}[Y_{k+1} - S_k | \mathcal{F}_k] = \mathbb{E}[Y_{k+1} - S_{k-1} | \mathcal{F}_k] - B_k \leq Y_k - S_{k-1} - A_k \leq Y_k - S_{k-1}.$$

Hence $Y_{k+1} - S_k$ is a supermartingale, which is bounded from below and which converges a.s. by the supermartingale convergence theorem (cf. Williams [26, Chapter 11]). Since S_k converges by assumption, it follows that Y_k converges almost surely. Notice finally that (9) is equivalent to

$$\mathbb{E}[Y_{k+1} - S_k + T_k | \mathcal{F}_k] \leq Y_k - S_{k-1} + T_{k-1},$$

and by the same reasoning as above it follows that $Y_{k+1} - S_k + T_k$ converges a.s., which implies that $T_k = \sum_{\ell=1}^k A_\ell$ converges a.s. \square

Proposition 8. *Let $F(\cdot)$ be a real function defined on \mathbb{R}^m , which has a Lipschitz-continuous derivative $f(\cdot)$. Consider a recursion of the form*

$$X_{k+1} = X_k - a_k f(X_k) + a_k R_{k+1} \quad (10)$$

with some starting point X_0 , where $\mathbb{E}[R_{k+1} | R_1, \dots, R_k] = 0$. If $a_k \geq 0$, $\sum_k a_k = \infty$ and $\sum_k a_k^2 \|R_{k+1}\|^2 < \infty$ a.s., then $F(X_k)$ converges. If further $\sum_k a_k R_{k+1}$ converges a.s., then $f(X_k)$ converges to zero a.s.

Proof. Let $Y_k := F(X_k)$ and let K be the Lipschitz constant of f . Using the recursion (10) and the mean value theorem, there is a $\theta \in [0, 1]$ such that

$$\begin{aligned} F(X_{k+1}) &= F(X_k) + f(X_k + \theta(-a_k f(X_k) + a_k R_{k+1}))^\top \cdot (-a_k f(X_k) + a_k R_{k+1}) \\ &\leq F(X_k) + f(X_k)^\top \cdot (-a_k f(X_k) + a_k R_{k+1}) + K \cdot \|-a_k f(X_k) + a_k R_{k+1}\|^2 \\ &\leq F(X_k) - a_k \|f(X_k)\|^2 + a_k f(X_k)^\top R_{k+1} + 2K a_k^2 \|f(X_k)\|^2 + 2K a_k^2 \|R_{k+1}\|^2. \end{aligned}$$

³Generating random vectors can be accomplished by rejection sampling in \mathbb{R}^m , e.g., or by a standard procedure as addressed in the Appendix.

Taking the conditional expectation with respect to R_1, \dots, R_k one gets

$$\begin{aligned} \mathbb{E}[F(X_{k+1})|R_1, \dots, R_k] &\leq F(X_k) - a_k \|f(X_k)\|^2 + 2Ka_k^2 \|f(X_k)\|^2 + 2Ka_k^2 \|R_{k+1}\|^2 \\ &\leq F(X_k) - \frac{a_k}{2} \|f(X_k)\|^2 + 2Ka_k^2 \|R_{k+1}\|^2 \end{aligned}$$

for k large enough. Proposition 7, applied for $Y_k = F(X_k)$, $A_k = \frac{a_k}{2} \|f(X_k)\|^2$ and $B_k = 2Ka_k^2 \|R_{k+1}\|^2$, implies now that $F(X_k)$ converges and

$$\sum_k a_k \|f(X_k)\|^2 < \infty \text{ a.s.} \quad (11)$$

It remains to be shown that $f(X_k) \rightarrow 0$ a.s. Since $\sum_k a_k = \infty$, it follows from (11) that $\liminf_k \|f(X_k)\| = 0$ a.s. We argue now pointwise on the set of probability 1, where $\sum_k a_k \|f(X_k)\|^2 < \infty$, $\liminf_k \|f(X_k)\| = 0$ and $\sum_k a_k R_k$ converges. Suppose that $\limsup_k \|f(X_k)\|^2 > 2\epsilon$. Let $m_\ell < n_\ell < m_{\ell+1}$ be chosen such that

$$\begin{aligned} \|f(X_k)\|^2 &> \epsilon \text{ for } m_\ell < k \leq n_\ell \text{ and} \\ \|f(X_k)\|^2 &\leq \epsilon \text{ for } n_\ell < k \leq m_{\ell+1}. \end{aligned} \quad (12)$$

Let ℓ_0 be such large that

$$\sum_{k=m_{\ell_0}}^{\infty} a_k \|f(X_k)\|^2 \leq \frac{\epsilon^2}{2K} \quad \text{and} \quad \left\| \sum_{k=s}^t a_k R_{k+1} \right\| < \frac{\epsilon}{2} \quad \text{for all } s, t \geq m_{\ell_0}.$$

Then, for $\ell \geq \ell_0$ and $m_\ell \leq k \leq n_\ell$, by the recursion (10) and (12), as well as the Lipschitz property of f ,

$$\begin{aligned} \|f(X_{i+1}) - f(X_{m_\ell})\| &\leq K \|X_{i+1} - X_{m_\ell}\| = K \left\| \sum_{k=m_\ell}^i a_k f(X_k) + a_k R_{k+1} \right\| \\ &\leq K \sum_{k=m_\ell}^i a_k \|f(X_k)\| + K \left\| \sum_{k=m_\ell}^i a_k R_{k+1} \right\| \\ &\leq \frac{K}{\epsilon} \sum_{k=m_\ell}^i a_k \|f(X_k)\|^2 + \frac{\epsilon}{2} < \epsilon. \end{aligned}$$

Since $\|f(X_{m_\ell})\| \leq \epsilon$ it follows that $\limsup_k \|f(X_k)\| \leq 2\epsilon$ for every $\epsilon > 0$ and this contradiction establishes the result. \square

The following result ensures convergence of an algorithm of stochastic approximation type, which is given in Algorithm 3 to compute useful approximating measures.

Algorithm 3

A *stochastic approximation* algorithm: Optimal discretization of a probability measure P by a discrete probability sitting on s points

- (i) **Initialization.** Sample n random variates from the distribution P . Use a cluster algorithm (e.g., Algorithm 1) to find s clusters. Set $k = 0$ and let $Z(0) = (z^{(1)}(0), \dots, z^{(s)}(0))$ be the cluster medians. Moreover, choose a nonnegative and nonincreasing sequence a_k such that

$$\sum_{k=1}^{\infty} a_k^2 < \infty \text{ and } \sum_{k=1}^{\infty} a_k = \infty.$$

- (ii) **Iteration.** Use a new independent sample $\xi(k)$. Find the index i such that

$$d(\xi(k), z^{(i)}(k)) = \min_{\ell} d(\xi(k), z^{(\ell)}(k)),$$

set

$$z^{(i)}(k+1) := z^{(i)}(k) - a_k \cdot r d(\xi(k), z^{(i)}(k))^{r-1} \cdot \nabla_{z^{(i)}} d(\xi(k), z^{(i)}(k))$$

and leave all other points unchanged to form the new point set $Z(k+1)$.

- (iii) **Stopping criterion.** Stop, if either the predetermined number of iterations are performed or if the relative change of the point set Z is below some threshold ϵ . If not, then set $k = k+1$ and goto (ii).
- (iv) **Determination of the probabilities.** After having fixed the final point set Z , generate another sample $(\xi(1), \dots, \xi(n))$ and find the probabilities

$$p_i = \frac{1}{n} \# \left\{ \ell : d(\xi(\ell), z^{(i)}) = \min_k d(\xi(\ell), z^{(k)}) \right\}.$$

The final, approximate distribution is $\tilde{P} = \sum_{i=1}^s p_i \cdot \delta_{z^{(i)}}$, and the distance is

$$d_r(P, \tilde{P})^r \simeq \frac{1}{n} \sum_{\ell=1}^n \min_k d(\xi(\ell), z^{(k)})^r.$$

Theorem 9. Suppose that the step lengths a_k in Algorithm 3 satisfy

$$a_k \geq 0, \sum_k a_k = \infty \text{ and } \sum_k a_k^2 < \infty.$$

Suppose further that the assumptions of Proposition 3 are fulfilled. If $Z(k)$ is the sequence of point sets generated by the stochastic approximation algorithm (Algorithm 3), then $D(Z(k))$ converges a.s. and

$$\nabla_Z D(Z(k)) \rightarrow 0 \quad \text{a.s.}$$

as $k \rightarrow \infty$. In particular, if $D(Z)$ has a unique minimizer Z^* , then

$$Z(k) \rightarrow Z^* \quad \text{a.s.}$$

Proof. The matrices $Z(k)$ satisfy the recursion

$$Z(k+1) = Z(k) - a_k \nabla_Z D(Z(k)) - a_k W(k)$$

with

$$W(k) = \sum_{i=1}^s \mathbb{1}_{V_{Z(k)}^{(i)}}(\xi(k)) \cdot r \mathbf{d}(\xi(k), z^{(i)}(k))^{r-1} \cdot \nabla_{\xi} \mathbf{d}(\xi(k), z^{(i)}(k)) \\ - \int_{V_{Z(k)}^{(i)}} r \mathbf{d}(\xi(k), z^{(i)}(k))^{r-1} \cdot \nabla_{\xi} \mathbf{d}(\xi(k), z^{(j)}(k)) P(d\xi).$$

Notice that the vectors $W(k)$ are independent and bounded, $\mathbb{E}[W(k)] = 0$ and $\sum_i a_i W(i)$ converges a.s. Proposition 8 applied for $X_k = Z(k)$, $F(\cdot) = D(\cdot)$, $f(\cdot) = \nabla_Z D(\cdot)$ and $R_k = W(k)$ leads to the assertion. \square

Remark 10. A good choice for the step sizes a_k in Algorithm 3 is

$$a_k = \frac{C}{(k+30)^{3/4}}.$$

These step sizes satisfy the requirements $\sum_k a_k = \infty$, the sequence a_k is nonincreasing and $\sum_k a_k^2 < \infty$.

Remark 11. A variant of Algorithm 3 avoids the determination of the probabilities in the separate step (iv) but counts the probabilities on the fly.

It was mentioned and it is evident that Algorithm 2 converges to a local minimum, which is possibly not a global minimum.

3.3 Global approximation

There are also algorithms which find the globally optimal discretization. However, these algorithms are such complex that only very small problems, say to find two or three optimal points in \mathbb{R}^2 or \mathbb{R}^3 , can be handled effectively. In addition, the probability measure P must have bounded support.

For the sake of completeness we mention such an algorithm which is able to provide a globally best approximating probability measure located on not more than s supporting points. Algorithm 4 produces successive refinements, which converge to a globally optimal approximation of the initial measure P .

4 Trees, and their distance to stochastic processes

In this section we give bounds for the objective value of stochastic optimization problems. By generalizing an important result from multistage stochastic optimization we provide bounds first when the law of the underlying process is approximated by a process with a pushforward measure.

The goal is to construct a valued probability tree, which represents the process $(\xi_t)_{t=0}^T$ in the best possible way. Trees are represented by a tuple consisting of the treestructure (i.e., the predecessor relations), the values of the process sitting on the nodes and the (conditional) probabilities sitting on the arcs of the tree. To be more precise, let $\mathbb{T} = (n, \text{pred}, z, Q)$ represent a tree with

Algorithm 4

Optimal discretization of probability P by a probability \tilde{P} sitting on s points: a global optimization algorithm.

- Suppose that the optimal configuration of s points in a bounded set (for simplicity the unit cube $[0, 1]^m$ in \mathbb{R}^m) is to be found. The optimal configuration is an element of $[0, 1]^{m \times s}$. At stage ℓ the unit cube is dissected into smaller cubes, say $[0, 1]^m = \bigcup C_j$. By considering all selections $C_{j_1} \times C_{j_2} \times \dots \times C_{j_s}$ a dissection of the search space is defined. The “local” problem finds a stochastic lower and a stochastic upper bound for

$$\min_{z^{(i)} \in C_{j_i}} \int_{\Xi} \min_i d(u, z^{(i)}) P(du).$$

- **Bounding step.** Configurations which have a lower bound larger than the upper bound of another configuration are excluded and not investigated further.
 - **Branching step.** The best configuration will be refined by dissecting the pertaining cubes into smaller cubes.
 - **Stopping.** If the gap between the upper bound and the lower bound is small enough, then stop.
-

- n nodes;
- a function pred mapping $\{1, 2, \dots, n\}$ to $\{0, 1, 2, \dots, n\}$. $\text{pred}(k) = \ell$ means that node ℓ is a direct predecessor of node k . The root is node 1 and its direct predecessor is formally encoded as 0;
- a valuation $z_i \in \mathbb{R}^m$ of each node $i \in \{1, 2, \dots, n\}$;
- the conditional probability $Q(i)$ of reaching node i from its direct predecessor; for the root we have $Q(1) = 1$.

It is always assumed that these parameters are consistent, i.e., that they form a tree of height T , meaning that all leaves of the tree are at the same level T . The distance of each node to the root is called the *stage* of the node. The root is at stage 0 and the leaves of the tree are at stage T .

Let $\tilde{\Omega}$ be the set of all leaf nodes, which can be seen as a probability space carrying the unconditional probabilities $P(n)$ to reach the leaf node $n \in \tilde{\Omega}$ from the root. Obviously the unconditional probability $\tilde{P}(i)$ of any node i is the product of the conditional probabilities of all its predecessors (direct and indirect).

Let $\text{pred}_t(n)$ denote the predecessor of node n at stage t . These mappings induce a filtration $\tilde{\mathfrak{F}} = (\tilde{\mathcal{F}}_0, \dots, \tilde{\mathcal{F}}_T)$, where $\tilde{\mathcal{F}}_t$ is the sigma-algebra induced by pred_t . $\tilde{\mathcal{F}}_0$ is the trivial sigma-algebra and $\tilde{\mathcal{F}}_T$ is the power set of $\tilde{\Omega}$. The process $(\tilde{\xi}_t)$ takes the values z_i for all nodes i at stage t with probability $\tilde{P}(i)$.

On the other hand, also the basic stochastic process (ξ_t) is defined on a filtered probability space $(\Omega, \mathfrak{F} = (\mathcal{F}_0, \dots, \mathcal{F}_T), P)$, where \mathcal{F}_0 is the trivial sigma-algebra. Via the two stochastic processes, the basic process (ξ_t) defined on Ω and its discretization $(\tilde{\xi}_t)$ defined on $\tilde{\Omega}$, a *distance* between $u \in \Omega$

and $v \in \tilde{\Omega}$ is defined by

$$d(u, v) = \sum_{t=1}^T d_t(\xi_t(u), \tilde{\xi}_t(v)), \quad (13)$$

where d_t are distances on \mathbb{R}^{m_t} (the state \mathbb{R}^{m_t} may depend on the stage t , but to keep the notation simple we consider only \mathbb{R}^m processes in what follows, i.e., $m_t = m$ for all t). To measure the quality of the approximation of the process (ξ_t) by the tree \mathbb{T} one may use the nested distance (see Definition 18 below) or its simpler variant, a stagewise transportation bound.

4.1 Approximation of stochastic processes

Different algorithms have been demonstrated in the previous sections to construct a probability measure $\tilde{P} = \sum_{i=1}^s p_i \delta_{z_i}$ approximating P . The approximating measures \tilde{P} presented are all induced by the transport map

$$\begin{aligned} T: \Xi &\rightarrow Z \\ \xi &\mapsto z_i, \text{ if } \xi \in V_Z^{(i)}. \end{aligned}$$

It holds moreover that $V_Z^{(i)} = \{T = z_i\}$ (and in particular $P(V_Z^{(i)}) = P(T = z_i)$), which shows that the facility location problems can be formulated by involving just transport maps (cf. Remark 2).

In what follows we generalize the concept of transport maps to stochastic processes. We generalize a central theorem in stochastic optimization, which provides a bound in terms for the pushforward measure for transport maps. We demonstrate that an adequately measurable, finitely valued transport map represents a tree. Further, we employ stochastic approximation techniques again to find a useful tree representing a process. The methods allow computing bounds for the corresponding stochastic optimization problem.

4.2 The main theorem of stochastic optimization for pushforward measures

Consider a stochastic process $\xi = (\xi_t)_{t=0}^T$, which is discrete in time. Each component $\xi_t : \Omega \rightarrow \Xi_t$ has the state space Ξ_t (which may be different for varying t 's). Further let $\Xi := \Xi_0 \times \dots \times \Xi_T$ and observe that Ξ_t is naturally embedded in Ξ .

Definition 12. We say that a process $x = (x_t)_{t=0}^T$ (with $x_t : \Xi \rightarrow \mathbb{X}_t$) is *nonanticipative* with respect to the stochastic process $\xi = (\xi_t)_{t=0}^T$, if x_t is measurable with respect to the sigma algebra $\sigma(\xi_0, \dots, \xi_t)$. We write

$$\begin{aligned} x \prec \sigma(\xi), \quad &\text{if } x_t \text{ is measurable with respect to the} \\ &\text{sigma algebra } \sigma(\xi_0, \dots, \xi_t) \text{ for every } t = 0, \dots, T. \end{aligned}$$

It follows from Shiryaev [22, Theorem II.4.3] that a process x is nonanticipative, if there is a measurable function (denoted x_t again), such that $x_t = x_t(\xi_0, \dots, \xi_t)$, i.e., $x_t(\omega) = x_t(\xi_0(\omega), \dots, \xi_t(\omega))$ for all t .

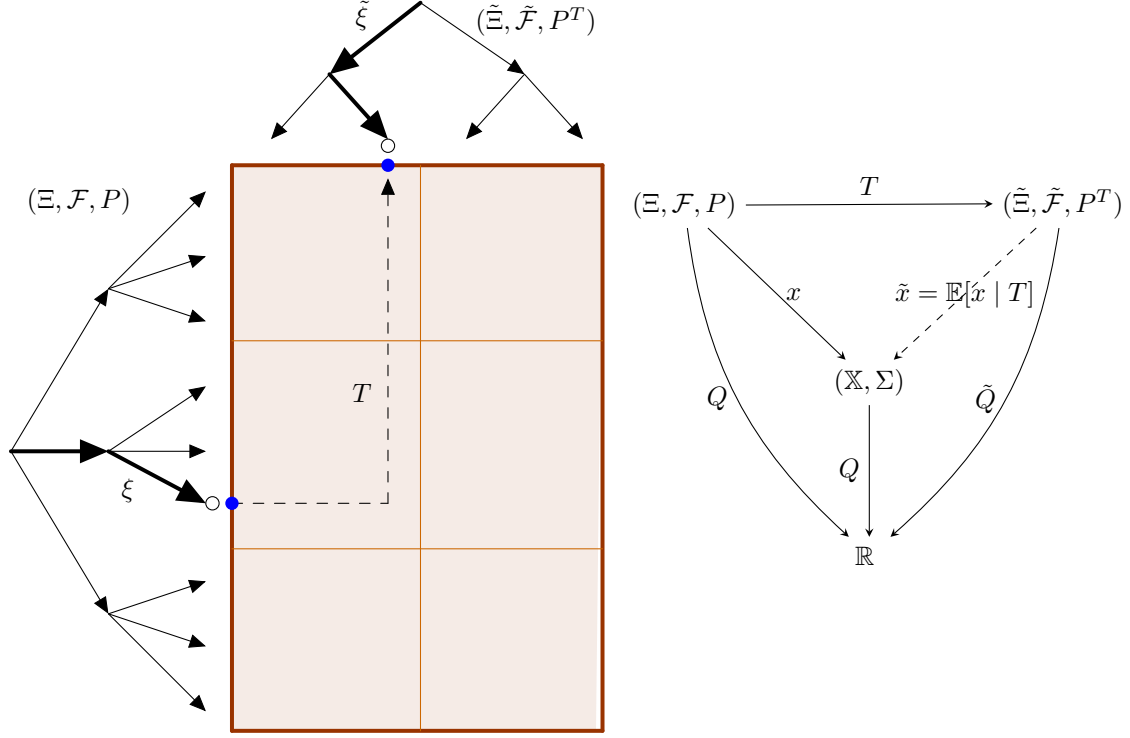


Figure 1: Left: the measurable transport map T , mapping trajectories ξ to $\tilde{\xi} = T(\xi)$. Right: the diagram displays the domain and codomain of the functions involved. The diagram is commutative on average.

Definition 13. A transport map $T : \Xi \rightarrow \tilde{\Xi}$ is *nonanticipative* if

$$T \circ \xi \triangleleft \sigma(\xi),$$

that is, each component $T(\xi)_t \in \tilde{\Xi}_t$ satisfies $T(\xi_0, \dots, \xi_T)_t = T(\xi_0, \dots, \xi_t)_t$ for all $t = 0, \dots, T$.

We consider first the stochastic optimization problem

$$\min \{ \mathbb{E}_P [Q(x, \xi)] : x \in \mathbb{X}, x \triangleleft \sigma(\xi) \}, \quad (14)$$

where the decision x is measurable with respect to the process ξ , $x \triangleleft \sigma(\xi)$.

The following theorem generalizes an important observation (cf. [20, Theorem 11]) to image measures. This outlines the central role of a nonanticipative transport map in stochastic optimization.

Theorem 14 (Stagewise transportation bound). *Let \mathbb{X} be convex and the \mathbb{R} -valued function $\tilde{Q} : \mathbb{X} \times \tilde{\Xi} \rightarrow \mathbb{R}$ be uniformly convex in x , that is,*

$$\tilde{Q}((1 - \lambda)x_0 + \lambda x_1, \tilde{\xi}) \leq (1 - \lambda)\tilde{Q}(x_0, \tilde{\xi}) + \lambda\tilde{Q}(x_1, \tilde{\xi}) \quad (\tilde{\xi} \in \tilde{\Xi}).$$

Moreover, let $Q: \mathbb{X} \times \Xi \rightarrow \mathbb{R}$ be linked with \tilde{Q} by

$$|Q(x, \xi) - \tilde{Q}(x, \tilde{\xi})| \leq c(\xi, \tilde{\xi}) \quad \text{for all } \xi \in \Xi \text{ and } \tilde{\xi} \in \tilde{\Xi}, \quad (15)$$

where $c: \Xi \times \tilde{\Xi} \rightarrow \mathbb{R}$ is a function (called cost function).

Then for every nonanticipative transport map

$$T: \Xi \rightarrow \tilde{\Xi}$$

it holds that

$$\left| \inf_{x \triangleleft \sigma(\xi)} \mathbb{E}_P Q(x(\xi), \xi) - \inf_{\tilde{x} \triangleleft \sigma(\tilde{\xi})} \mathbb{E}_{P^T} \tilde{Q}(\tilde{x}(\tilde{\xi}), \tilde{\xi}) \right| \leq \mathbb{E}_P c(\xi, T(\xi)). \quad (16)$$

Remark 15. Equation (16) relates the problem (14), the central problem of stochastic optimization, with another stochastic optimization problem on the image measure P^T . The problem on P^T may have a different objective (\tilde{Q} instead of Q), but it is easier to solve, as it is reduced to the simpler probability space with pushforward measure P^T instead of P .

The right hand side of (16), $\mathbb{E}_P c(\xi, T(\xi))$, is notably not a distance, but an expectation of the cost function c in combination with the transport map T .

Remark 16. In a typical application of Theorem 14 one has that $\tilde{\Xi} \subset \Xi$ and $\tilde{Q}(\cdot) = Q(\cdot)$. Further, $c(\xi, \tilde{\xi}) = L \cdot d(\xi, \tilde{\xi})$, where d is a distance on $\Xi \times \Xi$ and L , by means of (15), is a Lipschitz constant for the objective function Q .

Proof of Theorem 14. First, let \tilde{x} be any feasible policy with $\tilde{x} \triangleleft \sigma(\tilde{\xi})$, that is, $\tilde{x}_t = \tilde{x}_t(\tilde{\xi}_0, \dots, \tilde{\xi}_t)$ for all t . It follows from the measurability of the transport map T that the derived policy $x := \tilde{x} \circ T$ is nonanticipative, i.e., $x \triangleleft \sigma(\xi)$. By relation (15) it holds for the policy x that

$$\mathbb{E}Q(x(\xi), \xi) = \mathbb{E}Q(\tilde{x}(T(\xi)), \xi) \leq \mathbb{E}\tilde{Q}(\tilde{x}(T(\xi)), T(\xi)) + \mathbb{E}c(\xi, T(\xi)),$$

and by change of variables thus

$$\mathbb{E}Q(x(\xi), \xi) \leq \mathbb{E}_{P^T} \tilde{Q}(\tilde{x}(\tilde{\xi}), \tilde{\xi}) + \mathbb{E}c(\xi, T(\xi)).$$

One may pass to the infimum with respect to \tilde{x} , and it follows, as $x = \tilde{x} \circ T \triangleleft \sigma(\xi)$, that

$$\inf_{x \triangleleft \sigma(\xi)} \mathbb{E}Q(x(\xi), \xi) \leq \inf_{\tilde{x} \triangleleft \sigma(\tilde{\xi})} \mathbb{E}_{P^T} \tilde{Q}(\tilde{x}(\tilde{\xi}), \tilde{\xi}) + \mathbb{E}c(\xi, T(\xi)). \quad (17)$$

For the converse inequality suppose that a policy $x \triangleleft \sigma(\xi)$ is given. Define

$$\tilde{x} := \mathbb{E}(x|T), \quad \text{i.e., } \tilde{x}_t(\tilde{\xi}) := \mathbb{E}(x_t | T_t(\cdot) = \tilde{\xi})$$

(Figure 1 visualizes the domain and codomain of this random variable) and note that $\tilde{x} \triangleleft \sigma(T(\xi))$ by construction and as T is nonanticipative.

As the function \tilde{Q} is convex it follows from Jensen's inequality, conditioned on $\{T(\cdot) = \tilde{\xi}\}$, that

$$\begin{aligned} \tilde{Q}(\tilde{x}(\tilde{\xi}), \tilde{\xi}) &= \tilde{Q}(\mathbb{E}(x|T)(\tilde{\xi}), \tilde{\xi}) \\ &= \tilde{Q}(\mathbb{E}(x(\xi) | T(\xi) = \tilde{\xi}), \tilde{\xi}) \\ &\leq \mathbb{E}\left(\tilde{Q}(x(\xi), \tilde{\xi}) \mid T(\xi) = \tilde{\xi}\right). \end{aligned}$$

By assumption (15) linking Q and \tilde{Q} it holds further that

$$\begin{aligned}
\tilde{Q}(\tilde{x}(\tilde{\xi}), \tilde{\xi}) &\leq \mathbb{E}\left(\tilde{Q}(x(\xi), \tilde{\xi}) \mid T(\xi) = \tilde{\xi}\right) \\
&= \mathbb{E}\left(\tilde{Q}(x(\xi), T(\xi)) \mid T(\xi) = \tilde{\xi}\right) \\
&\leq \mathbb{E}\left(Q(x(\xi), T(\xi)) + c(\xi, T(\xi)) \mid T(\xi) = \tilde{\xi}\right) \\
&= \mathbb{E}\left(Q(x(\xi), T(\xi)) \mid T(\xi) = \tilde{\xi}\right) + \mathbb{E}\left(c(\xi, T(\xi)) \mid T(\xi) = \tilde{\xi}\right),
\end{aligned}$$

and by taking expectations with respect to the measure P^T it follows that

$$\mathbb{E}_{P^T} \tilde{Q}(\tilde{x}(\tilde{\xi}), \tilde{\xi}) \leq \mathbb{E} Q(x(\xi), T(\xi)) + \mathbb{E} c(\xi, T(\xi)).$$

Recall that $x \triangleleft \sigma(\xi)$ was arbitrary, by taking the infimum it follows thus that

$$\inf_{\tilde{x} \triangleleft \sigma(T(\xi))} \mathbb{E}_{P^T} \tilde{Q}(\tilde{x}(\tilde{\xi}), \tilde{\xi}) \leq \inf_{x \triangleleft \sigma(\xi)} \mathbb{E} Q(x(\xi), \xi) + \mathbb{E} c(\xi, T(\xi)).$$

Together with (17) this is the assertion. \square

4.3 Approximation by means of a pushforward measure

In this section we construct a tree by establishing a transport map $T : \Xi \rightarrow \Xi$ with the properties of Definition 1. The algorithm is based on stochastic approximation and extends Algorithm 3, as well as an algorithm contained in Pflug [17]. In contrast to the existing literature we do not require more than a sample of trajectories (i.e., scenarios). The scenarios may result from observations or from simulation.

Algorithm 5 is the tree equivalent of Algorithm 3. It uses a sample of trajectories to produce a tree approximating the process ξ . The algorithm further provides the estimate $\mathbb{E} c(\xi, T(\xi))$, which describes the quality of the approximation of the tree $T \circ \xi$ in comparison with the original process ξ .

Algorithm 5

Tree generation by stochastic approximation, based on stochastic approximation of measures (Algorithm 3)

- (i) **Initialization.** Set $k = 0$, let $c_E = 0$ set the counters $c(n) = 0$ and let $Z^{(0)}(n) \in \Xi_t$ be chosen for each node n of the tree. Moreover, choose a nonnegative and nonincreasing sequence a_k such that

$$\sum_{k=1} a_k^2 < \infty \text{ and } \sum_{k=1} a_k = \infty.$$

- (ii) **Iteration.** Use a new independent trajectory

$$\xi(k) = (\xi_0(k), \dots, \xi_T(k))$$

with law P .

Find a trajectory of successive nodes n_0, n_1, \dots, n_T in the tree with $n_t = \text{pred}_t(n_{t+1})$ such that

$$n_t \in \underset{n' \in \mathcal{N}_t(n_0, \dots, n_{t-1})}{\text{argmin}} \mathbf{d}_t(\xi_t(k), Z^{(k)}(n')),$$

where $\mathcal{N}_t(n_0, \dots, n_{t-1})$ collects all nodes at stage t with predecessors n_0, \dots, n_{t-1} . Assign the new values

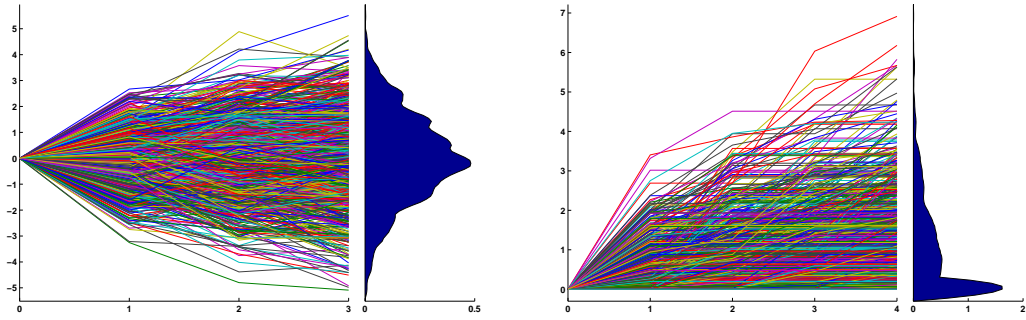
$$Z^{(k)}(n_t) := Z^{(k-1)}(n_t) - a_k \cdot r \mathbf{d}_t(\xi_t(k), Z^{(k-1)}(n_t))^{r-1} \cdot \nabla_{\xi} \mathbf{d}_t(\xi_t(k), Z^{(k-1)}(n_t)),$$

increase the counters $c(n_t) = c(n_t) + 1$ for the nodes n_0, n_1, \dots, n_t and set $c_E := c_E + \left(\sum_{t=0}^T \mathbf{d}_t(\xi_t(k), Z^{(k-1)}(n_t))\right)^r$. For the other nodes let the values unchanged, i.e., $Z^{(k)}(n) := Z^{(k-1)}(n)$ whenever $n \notin \{n_0, n_1, \dots, n_T\}$.

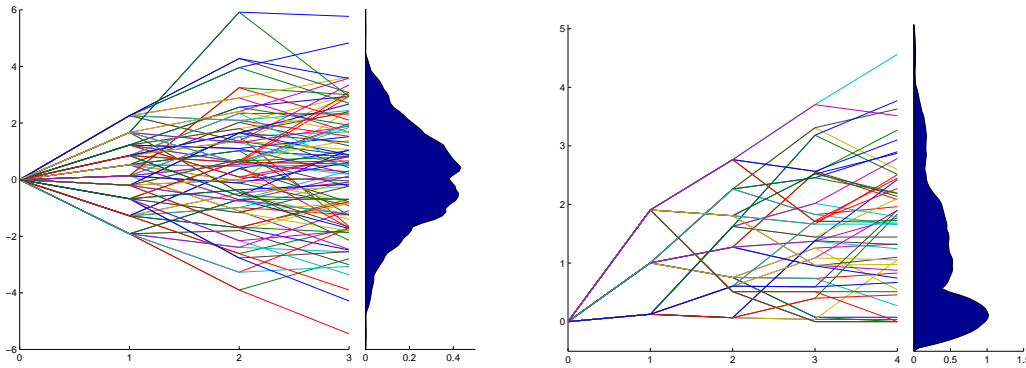
- (iii) **Stopping criterion.** Stop, if the predetermined number of iterations is performed. If not, then set $k = k + 1$ and goto (ii).
- (iv) Set the conditional probabilities $p(n) = c(n)/N$, where N is the total number of draws. The quantity $\mathbb{E} \mathbf{d}(\xi, T(\xi))^r$ is estimated by

$$\mathbb{E} \mathbf{d}(\xi, T^n(\xi))^r \simeq \frac{1}{N} c_E. \tag{18}$$

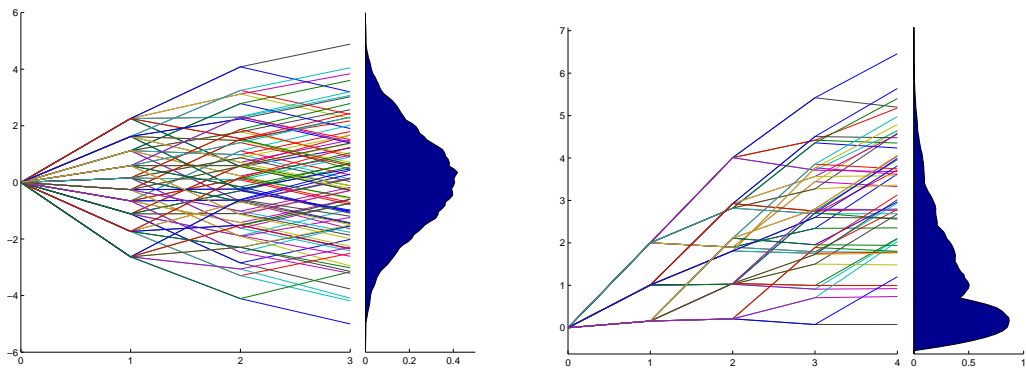
Example 17. To demonstrate Algorithm 5 we consider a Gaussian walk in 3 stages first. The tree with business $(10, 5, 2)$, found after 1000 and 100,000 samples, is displayed in Figure 2 (left plots). The probability distribution of the leaves is annotated in the plots. The final distribution of the initial process is $N(0, 3)$ (Figure 2a). The Gaussian walk and the tree are at a distance of $(\mathbb{E} \mathbf{d}(\xi, T(\xi))^2)^{1/2} \simeq 0.084$, where we have employed the usual Euclidean distance and $r = 2$.



(a) 1000 sample paths of the Gaussian random walk and the (non-Markovian) running maximum process



(b) Trees with bushiness $(10, 5, 2)$ and $(3, 3, 3, 2)$ approximating the process in Figure 2a



(c) The transportation bound to the underlying Gaussian process is 0.084, the transportation bound to the non-Markovian running maximum process is 0.13

Figure 2: Trees produced by Algorithm 5 after 1000 (Figure 2b) and 100,000 (Figure 2c) samples. Annotated is a density plot of the probability distribution at the final stage.

The process which we consider further is the running maximum

$$M_t := \max \left\{ \sum_{i=1}^t \xi_i : t' \leq t \right\} \text{ with } \xi_i \sim N(0, 1). \quad (19)$$

Note, that the running maximum is *not* a Markovian process. The results of Algorithm 5 are displayed in Figure 2 (right) for a bushiness of (3, 3, 3, 2). The running maximum process and the tree in Figure 2c have distance $(\mathbb{E} d(\xi, T(\xi))^2)^{1/2} \simeq 0.13$.

5 The nested distance

In the previous section we have proposed an algorithm to construct an approximating tree from observed sample paths by stochastic approximation. It is an essential observation that the proposed algorithm works if the measure \tilde{P} on the tree is induced by a transport map T , $\tilde{P} = P^T$. It is a significant advantage of Theorem 14 that the bound

$$\mathbb{E} c(\xi, T(\xi)) \quad (20)$$

in equation (16) is very cheap to compute (Eq. (18) in Algorithm 5, e.g., provides $\mathbb{E} c(\xi, T(\xi))$ as a byproduct). However, the algorithm works only if the second law is a pushforward measure P^T , where T is adequately measurable.

In the general situation the quantity (20) is not symmetric, that is, there does not exist a transportation map \tilde{T} , say, such that $\mathbb{E} c(\xi, T(\xi)) = \mathbb{E} c(\tilde{T}(\tilde{\xi}), \tilde{\xi})$. For this (20) does not extend to a distance of processes, as is the case for the Wasserstein distance for probability measures.

The nested distance was introduced to handle the general situation. In what follows we recall the definition and cite the result, which are essential for tree generation. Then we provide algorithms again to construct approximating trees, which are close in the nested distance.

Definition 18 (The nested distance, cf. [18]). Assume that two probability models

$$\mathbb{P} = (\Omega, (\mathcal{F}_t), P, \xi) \text{ and } \tilde{\mathbb{P}} = (\tilde{\Omega}, (\tilde{\mathcal{F}}_t), P, \tilde{\xi})$$

are given, such that for $u \in \Omega$ and $v \in \tilde{\Omega}$ a distance $d(u, v)$ is defined by (13). The nested distance of order $r \geq 1$ is the optimal value of the optimization problem

$$\begin{aligned} & \underset{(\text{in } \pi)}{\text{minimize}} \quad \left(\iint d(u, v)^r \pi(du, dv) \right)^{1/r} \\ & \text{subject to } \pi(A \times \tilde{\Omega} \mid \mathcal{F}_t \otimes \tilde{\mathcal{F}}_t) = P(A \mid \mathcal{F}_t) \quad (A \in \mathcal{F}_t, t = 0, \dots, T) \text{ and} \\ & \quad \pi(\Omega \times B \mid \mathcal{F}_t \otimes \tilde{\mathcal{F}}_t) = \tilde{P}(B \mid \tilde{\mathcal{F}}_t), \quad (B \in \tilde{\mathcal{F}}_t, t = 0, \dots, T), \end{aligned} \quad (21)$$

where the infimum in (21) is among all bivariate probability measures $\pi \in \mathcal{P}(\Omega \times \tilde{\Omega})$ which are measures for $\mathcal{F}_T \otimes \tilde{\mathcal{F}}_T$. Its optimal value – the nested, or multistage distance – is denoted by

$$dl_r(\mathbb{P}, \tilde{\mathbb{P}}).$$

By (13) the distance depends on the image measures induced by $\xi_t: \Omega \rightarrow \mathbb{R}^m$ and $\tilde{\xi}: \tilde{\Omega} \rightarrow \mathbb{R}^m$ on \mathbb{R}^m .

The following theorem is the counterpart of Theorem 14 for general measures and proved in [20]. No transport map is available for the general situation. However, the nested distance can be applied to reveal the same type of result.

Theorem 19. *Let $\mathbb{P} = (\Omega, (\mathcal{F}_t), P, \xi)$ and $\tilde{\mathbb{P}} = (\tilde{\Omega}, (\tilde{\mathcal{F}}_t), P, \tilde{\xi})$ be two probability models. Assume that \mathbb{X} is convex and the cost function Q is convex in x for any fixed ξ . Moreover let Q be uniformly Hölder continuous in ξ with constant L_β and exponent β , that is*

$$|Q(x, \xi) - Q(x, \tilde{\xi})| \leq L_\beta \left(\sum_{t=1}^T d_t(\xi_t, \tilde{\xi}_t) \right)^\beta$$

for all $x \in \mathbb{X}$. Then the optimal value function inherits the Hölder constants with respect to the nested distance,

$$\left| \begin{array}{l} \min \{ \mathbb{E}_P [Q(x, \xi)]: x \in \mathbb{X}, x \triangleleft \mathfrak{F}, \mathbb{P} = (\Omega, \mathfrak{F}, P, \xi) \} \\ - \min \{ \mathbb{E}_{\tilde{P}} [Q(x, \xi)]: x \in \mathbb{X}, x \triangleleft \tilde{\mathfrak{F}}, \tilde{\mathbb{P}} = (\tilde{\Omega}, \tilde{\mathfrak{F}}, \tilde{P}, \tilde{\xi}) \} \end{array} \right| \leq \mathbf{dl}_r(\mathbb{P}, \tilde{\mathbb{P}})^\beta \quad (22)$$

for any $r \geq 1$. This bound cannot be improved.

The relation between the nested distance and the single period Wasserstein distance.

The nested distance $\mathbf{dl}_r(\mathbb{P}, \tilde{\mathbb{P}})$ can be bounded by the Wasserstein distances of the conditional probabilities as is described in Theorem 21 below. It uses the notion of the K -Lipschitz property.

Definition 20 (K -Lipschitz property). Let P be a probability on \mathbb{R}^{mT} , dissected into transition probabilities P_1, \dots, P_T on \mathbb{R}^m . We say that P has the K -Lipschitz property for $K = (K_1, \dots, K_{T-1})$, if the transitional probability measures satisfy

$$\mathbf{d}_r(P_{t+1}(\cdot|u^t), P_{t+1}(\cdot|v^t)) \leq K_t \mathbf{d}(u^t, v^t) \quad (23)$$

for all $u^t, v^t \in \mathbb{R}^{m(t-1)}$ and $t = 1, \dots, T-1$.

Theorem 21 (Stagewise transportation distance). *Suppose that the probability measure P on \mathbb{R}^{mT} fulfills a (K_1, \dots, K_{T-1}) -Lipschitz property and that the conditional distributions of P and \tilde{P} satisfy*

$$\mathbf{d}_r(P_1, \tilde{P}_1) \leq \epsilon_1$$

and

$$\mathbf{d}_r(P_{t+1}(\cdot|v^t), \tilde{P}_{t+1}(\cdot|v^t)) \leq \epsilon_{t+1} \quad \text{for all } v^t \text{ and } t = 0, \dots, T-1. \quad (24)$$

Then the nested distance is bounded by

$$\mathbf{dl}_r(\mathbb{P}, \tilde{\mathbb{P}}) \leq \sum_{t=1}^{T-1} \epsilon_t \cdot \prod_{s=t}^T (1 + K_s). \quad (25)$$

Proof. The proof is contained in [19]. □

The previous sections address approximations of a probability measure, where the quality of the approximation is measured by the Wasserstein distance. The algorithms described in Section 3 can be employed at every node separately to build the tree. To apply the general result (25)

Algorithm 6

Tree generation with fixed bushiness

Parameters. Let T be the desired height of the tree and let (b_1, \dots, b_T) be the given bushiness parameters per stage.

- **Determining the Root.** The value of the process at the root is ξ_0 . Its stage is 0. Set the root as the current open node.
 - **Successor generation.** Enumerate the tree stagewise from the root to the leaves.
 - (i) Let k be the node to be considered next and let $t < T$ be its stage. Let $\xi_0, \xi_1, \dots, \xi_t$ be the already fixed values at node k and all its predecessors. Call the stochastic approximation algorithm (Algorithm 3) to generate b_t points $z^{(1)}, \dots, z^{(b_t)}$ out of the probability distribution

$$F_{t+1}(\cdot | \xi_t, \xi_{t-1}, \dots, \xi_0) \tag{26}$$
 and find the corresponding conditional probabilities $p(\tilde{z}^{(i)})$.
 - (ii) Store the b_t successor nodes, say with node numbers (n_1, \dots, n_{b_t}) of node k and assign to them the values $\xi(n_1) = z^{(1)}, \dots, \xi(n_{b_t}) = z^{(b_t)}$ as well as their conditional probabilities $q(n_i) = p(\tilde{z}^{(i)})$.
 - **Stopping Criterion.** If all nodes at stage $T - 1$ have been considered as parent nodes, the generation of the tree is finished. One may then calculate the unconditional probabilities out of the conditional probabilities.
-

it is necessary to condition on the values of the previous nodes (cf. (24)), which have already been fixed in earlier steps of the algorithm. To apply the algorithms described in Section 3 in connection with Theorem 21 it is thus necessary that the conditional probability measure is available to compute (24), that is, samples from

$$F_{t+1}(\cdot | \xi_t, \xi_{t-1}, \dots, \xi_0)$$

can be drawn.

5.1 Fixed branching structure

Algorithm 6 elaborates on generating scenario trees in further detail. The trees are constructed to have b_t successor nodes at each node at level t , the vector (b_1, \dots, b_T) is the *bushiness* of the tree.

The algorithm based on Theorem 21 described in this and the following section have been implemented in order to demonstrate their behavior by using the following example.

Example 22. Consider the Markovian process

$$\xi_0 = 2, \quad \xi_{t+1} = \xi_t^{1-\beta_t} \cdot e^{\eta_t}, \tag{27}$$

where $\beta_0 = 0.62$, $\beta_1 = 0.69$, $\beta_2 = 0.73$, $\beta_3 = 0.75$ and $\beta_4 = 0.77$, and

$$\eta_0 \sim \eta_2 \sim \eta_4 \sim \mathcal{N}(0, 0.5\beta_t^2) \quad \text{and} \quad \eta_1 \sim \eta_3 \sim \mathcal{N}(0, 0.2\beta_t^2).$$

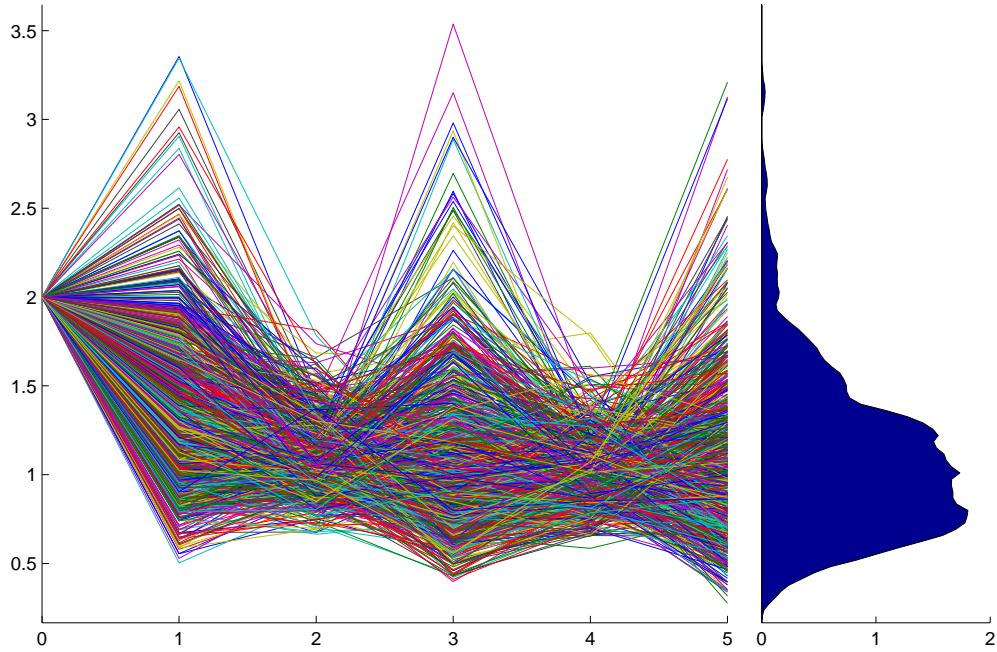


Figure 3: 1000 trajectories of the process (27) in Example 22

Notice that the distribution, given the past, is explicitly available by (27), although the conditional variance depends heavily on t .

Figure 3 displays some trajectories of the process (27). A binary tree, approximating the process (27), is constructed by use of Algorithm 6. Figure 4 displays the tree structure, as well as the approximating binary tree.

5.2 Tree, meeting a prescribed approximation precision

Assume that the law of the process to be approximated satisfies the K -Lipschitz property introduced in Definition 20. Theorem 21 then can be used to provide an approximation of the initial process in terms of the nested distance up to a prescribed precision. Algorithm 7 outlines this approach. Again, as in the preceding algorithm, it is necessary to have samples of the distribution $F_{t+1}(\cdot|\xi_t, \xi_{t-1}, \dots, \xi_0)$ available, given that the past is revealed up to the present stage t .

The algorithm is demonstrated again for the process (27) in Example 22. Results are displayed in Figure 5.

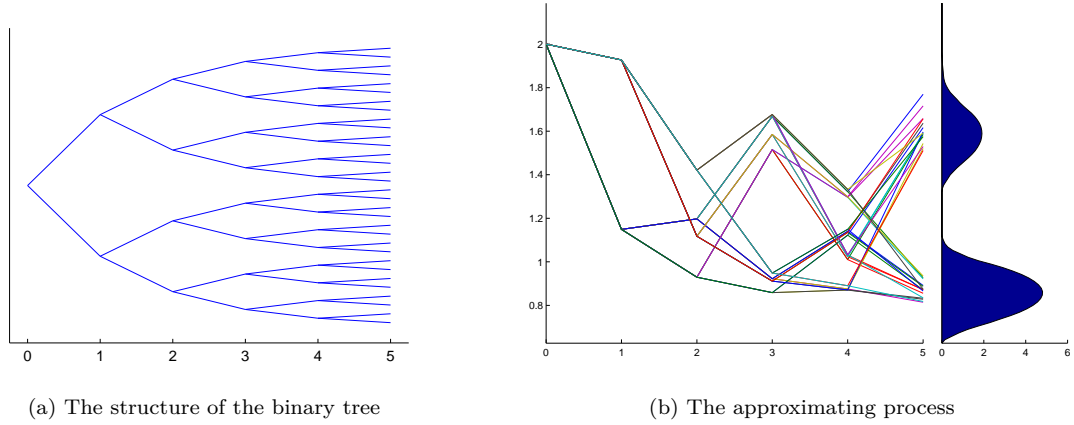


Figure 4: A binary tree generated by Algorithm 6 approximating process (27), Example 22

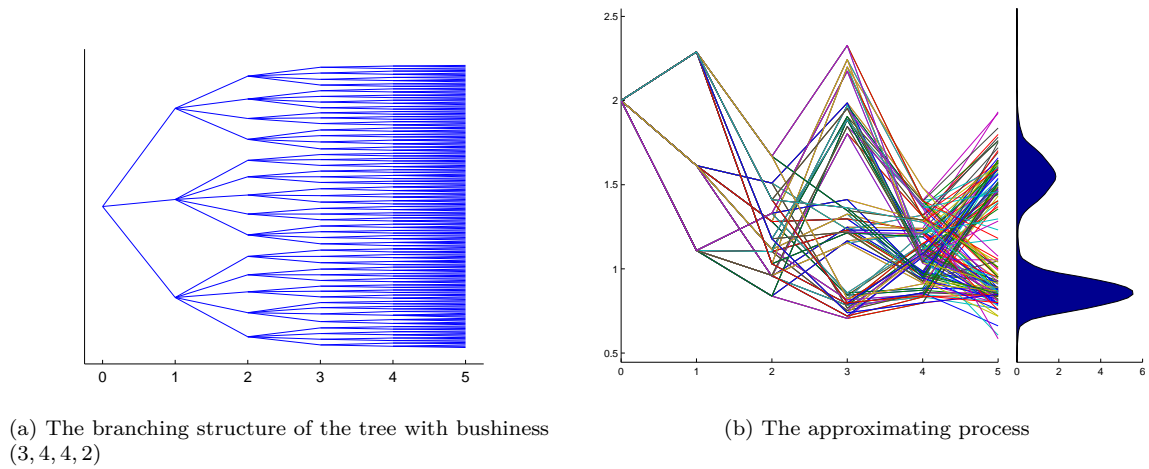


Figure 5: A tree, dynamically generated by Algorithm 7, approximating the process (27). The maximal stagewise distances were 0.30, 0.15, 0.30, 0.30 and 0.40. The approximating tree process has 390 nodes and 224 leaves.

Algorithm 7Dynamic tree generation with flexible bushiness

- **Parameters.** Let T be the desired height of the tree, let (b_1, \dots, b_T) be the minimal bushiness values and (d_1, \dots, d_T) the maximal stagewise transportation distances. These two vectors are fixed in advance.
 - **Determining the Root.** The value of the process at the root is ξ_0 , its stage is 0. Set the root as the current open node.
 - **While** there are open nodes **do**
 - (i) Let k be the next open node and let $t < T$ be its stage. Let $\xi_0, \dots, \xi_{t-1}, \xi_t$ be the already fixed values at node k and at its predecessors. Set the initial number of successors of k to $s = b_{t+1}$.
 - (ii) Call the stochastic approximation algorithm (Algorithm 3) to generate s points z_1^*, \dots, z_s^* out of the distribution
$$F_{t+1}(\cdot | \xi_t, \xi_{t-1}, \dots, \xi_0) \tag{28}$$
and compute the distance $d = d(F_{t+1}(\cdot | \xi_t, \xi_{t-1}, \dots, \xi_0), \sum_{i=1}^s p_i \cdot \delta_{z^{(i)}})$.
 - (iii) If the distance d is larger than d_{t+1} , then increase b by one and return to (ii).
 - (iv) Store the b successor nodes of node k using the values z_k^* as well as their conditional probabilities p_k^* and mark them as open.
 - **Stopping Criterion.** If all nodes at stage $T - 1$ have been considered as parent nodes, the generation of the tree is finished.
-

5.3 Estimating trees based on observed trajectories

The methods presented in the Sections 5.1 and 5.2 employ a conditional distribution in (26) and in (28). In case that this distribution is not available explicitly, then samples can be drawn from proper approximations.

In order to estimate the distribution conditional on (ξ_0, \dots, ξ_t) one may apply multivariate kernel estimation. It is demonstrated in Villani [25, Problem 10, pp. 327] and Bolley et al. [1] that kernel estimation is consistent with the Wasserstein (or Kantorovich) distance. To approximate the distribution based on samples, as well as to sample new trajectories one may apply the Nadaraya–Watson estimator (cf. Tsybakov [24]), which estimates the density of the conditional probability as

$$\hat{f}(z_{t+1} | \xi_1, \dots, \xi_t) = \frac{1}{h_n^{m_{t+1}}} \sum_{i=1}^n \pi_i \cdot k \left(\frac{\|z_{t+1} - Z_{i,t+1}\|}{h_n} \right), \tag{29}$$

Algorithm 8

Tree generation (based on Algorithm 6) with fixed bushiness from a sample of paths

Parameters. Let T be the desired height of the tree and let (b_1, \dots, b_T) be the given bushiness parameters per stage.

- **Determining the root.** The value of the process at the root is ξ_0 . Its stage is 0. Set the root as the current open node.
 - **Successor generation.** Enumerate the tree stagewise from the root to the leaves.
 - (i) Let k be the node to be considered next and let $t < T$ be its stage. Let $\xi_0, \xi_1, \dots, \xi_t$ be the already fixed values at node k and all its predecessors. Find an approximation of the form $\sum_{i=1}^{b_t} p_i \delta_{z^{(i)}}$, which is close in the Wasserstein distance to the distribution with density

$$f(z_{t+1} | \xi_1, \dots, \xi_t) \sim \hat{f}_n(z_{t+1} | \xi_1, \dots, \xi_t)$$
 (cf. (29)) by employing (for example) Algorithm 2.
 - (ii) Store the b_t successor nodes, say with node numbers (n_1, \dots, n_{b_t}) of node k and assign to them the values $\xi(n_1) = z^{(1)}, \dots, \xi(n_{b_t}) = z^{(b_t)}$ as well as their conditional probabilities $q(n_i) = p(z^{(i)})$ in the new tree.
 - **Stopping Criterion.** If all nodes at stage $T - 1$ have been considered as parent nodes, the generation of the tree is finished. One may then calculate the unconditional probabilities out of the conditional probabilities.
-

where the weights are⁴

$$\pi_i = \frac{k\left(\frac{\|\xi_1 - Z_{i,1}\|}{h_n}\right) \cdot \dots \cdot k\left(\frac{\|\xi_t - Z_{i,t}\|}{h_n}\right)}{\sum_{i'=1}^n k\left(\frac{\|\xi_1 - Z_{i',1}\|}{h_n}\right) \cdot \dots \cdot k\left(\frac{\|\xi_t - Z_{i',t}\|}{h_n}\right)} \quad (30)$$

and $(Z_{i,1:T})_{i=1}^n$ are the sample paths (trajectories) observed. It is shown in Hyndman et al. [13] that the choice $h_n \sim n^{-1/(4+m_1+\dots+m_{t+1})}$ minimizes the integrated, mean square error asymptotically. Algorithm 8 employs the kernel density (29) in order to construct a tree by using a set of samples directly.

Sampling from the kernel estimator. It should be noted that samples from (29) can be obtained in a very fast way by sampling an index i^* from the probabilities π_i first, and subsequently ξ_{t+1} from $\frac{1}{h_n^{m_{t+1}}} k\left(\frac{\|\cdot - Z_{i^*,t+1}\|}{h_n}\right)$.⁵

⁴The weights $\pi_i = k\left(\frac{\|\xi_{1:t} - Z_{i,1:t}\|}{h_n}\right) / \sum_{i'=1}^n k\left(\frac{\|\xi_{1:t} - Z_{i',1:t}\|}{h_n}\right)$ may be used alternatively, but the form as a product in (30) reflects the stagewise evolution of the observations.

⁵The method is occasionally called *composition method*.

Example 23. Figure 6a displays 1000 sample paths from a Gaussian walk in 12 stages. A binary tree with 4095 nodes was extracted (cf. Figure 6b) by employing Algorithm 8. Note that the extracted tree has $2^{11} = 2048$ leaves, which is more than twice the size of the original sample ($n = 1000$). Nevertheless, the approximating tree is apparently a useful approximation of the Gaussian process.

The Figures 6c and 6d display the results of Algorithm 8 for the (non Markovian) running maximum process defined in (19).

Choice of the parameters. For kernel estimation, the Epanechnikov kernel $k(z) = \frac{3}{4} (1 - z^2)_+$ is often proposed, as its shape is most efficient (for a specific criterion, cf. Tsybakov [24] for details). In the present situation the Epanechnikov kernel is not the best choice, as a division by zero has to be avoided in (29). This can be achieved by employing, e.g., the logistic kernel

$$k(z) = \frac{1}{e^z + 2 + e^{-z}} = \frac{1}{4} \frac{1}{\left(\cosh \frac{z}{2}\right)^2},$$

whis is strictly positive for all $z \in \mathbb{R}$.

As for the optimal bandwidth we recall from Caillerie et al. [4] that

$$\mathbb{E} d_2 \left(P, \hat{P}_n \right)^2 \leq \frac{C}{n^{2/m+4}}, \tag{31}$$

where \hat{P}_n is the measure with density $\frac{1}{n h_n^m} \sum_{i=1}^n k \left(\frac{z - Z_i}{h_n} \right)$. The rate (31) is the same rate as obtained by Silverman’s rule of thumb (cf. Silverman [23]) or Scott’s rule, which suggests to use

$$h_n \simeq \text{std}(X) \cdot \left(\frac{4}{n(m+2)} \right)^{1/m+4} \simeq \text{std}(X) \cdot n^{-1/m+4}.$$

For the most important case, $m = 1$, this is

$$h_n \simeq \text{std}(X) \cdot \left(\frac{4}{3n} \right)^{1/5} \simeq \text{std}(X) \cdot n^{-1/5}.$$

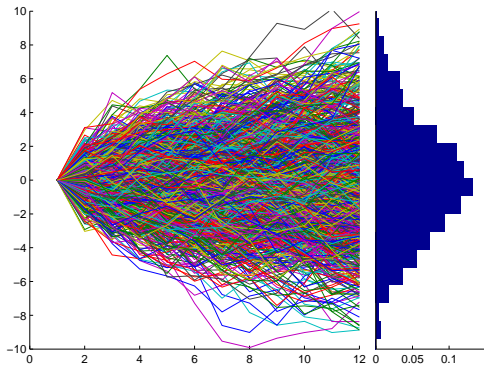
Remark 24. It should be noted that the estimate (31) does not require that the measure P has a density. Slight improvements of the rate of convergence are known in the case that a density is available—cf. Rachev [21] for a discussion.

6 Summary

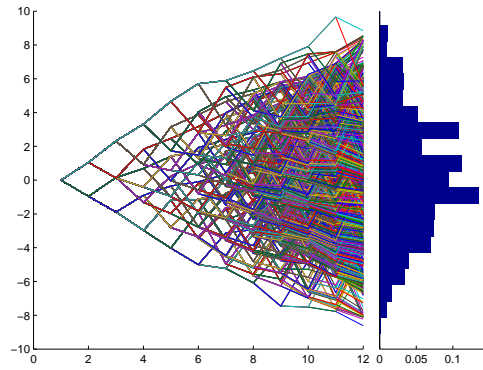
This paper addresses scenario tree generation, which is of interest in many economic and managerial situations, in particular for stochastic optimization.

It is demonstrated that techniques, which are used to approximate probability measures, can be extended to extract approximating trees, which model stochastic process in finite stages and states.

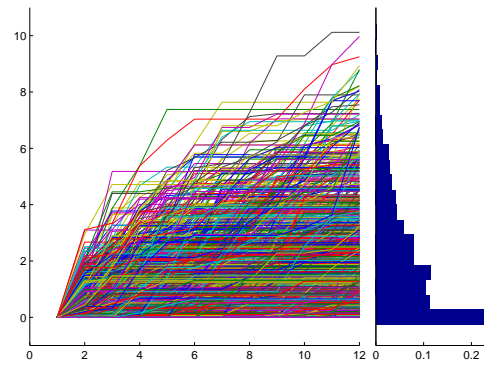
Various algorithms are shown first to approximate probability measures by discrete measures. These algorithms are combined then at a higher level to provide tree approximations of stochastic



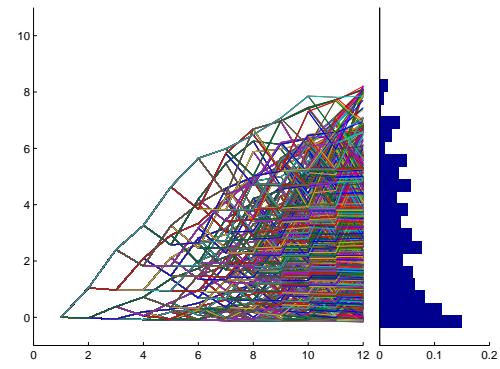
(a) 1000 sample paths from a (modified) Gaussian random walk



(b) Binary tree of height 12 with 4095 nodes, approximating the random walk from Figure 6a



(c) The running maximum process from Figure 6a



(d) Binary tree, extracted from the running maximum process in Figure 6c

Figure 6: Sample paths (left) and extracted trees (right) of a Markovian (above) and non-Markovian (below) process based on Algorithm 8

processes. The trees meet a quality criterion, which is formulated in terms of the nested distance. The nested distance is the essential distance to compare stochastic processes for stochastic optimization.

The algorithms presented require sampling according to probability distributions based on the past evolutions of the process. In case that these distributions are not available explicitly, then bootstrapping methods are employed which are based on kernel density estimators.

Several examples and charts demonstrated the quality of the solutions.

References

- [1] F. Bolley, A. Guillin, and C. Villani. Quantitative concentration inequalities for empirical measures on non-compact spaces. *Probability Theory and Related Fields*, 137(3-4):541–593, 2007. [24](#)
- [2] Y. Brenier. Décomposition polaire et réarrangement monotone des champs de vecteurs. *Comptes Rendus de l'Académie des Sciences Paris Sér. I Math*, 305(19):805–808, 1987. [3](#)
- [3] Y. Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4):375–417, 1991. [3](#)
- [4] C. Caillerie, F. Chazal, J. Dedecker, and B. Michel. Deconvolution for the Wasserstein metric and geometric inference. *Electronic Journal of Statistics*, 5:1394–1423, 2011. [26](#)
- [5] J. Dupačová, N. Gröwe-Kuska, and W. Römisch. Scenario reduction in stochastic programming. *Mathematical Programming, Ser. A*, 95(3):493–511, 2003. [1](#)
- [6] S. Graf and H. Luschgy. *Foundations of Quantization for Probability Distributions*, volume 1730 of *Lecture Notes in Mathematics*. Springer-Verlag Berlin Heidelberg, 2000. [4](#), [7](#)
- [7] H. Heitsch and W. Römisch. Scenario reduction algorithms in stochastic programming. *Comput. Optim. Appl. Stochastic Programming*, 24(2-3):187–206, 2003. [1](#)
- [8] H. Heitsch and W. Römisch. A note on scenario reduction for two-stage stochastic programs. *Operations Research Letters*, 6:731–738, 2007. [1](#)
- [9] H. Heitsch and W. Römisch. Scenario tree modeling for multistage stochastic programs. *Math. Program. Ser. A*, 118:371–406, 2009. [1](#)
- [10] H. Heitsch and W. Römisch. Scenario tree reduction for multistage stochastic programs. *Computational Management Science*, 2:117–133, 2009. [1](#)
- [11] H. Heitsch and W. Römisch. *Stochastic Optimization Methods in Finance and Energy*, chapter 14: Scenario Tree Generation for Multistage Stochastic Programs, pages 313–341. Springer Verlag, 2011. [1](#)
- [12] K. Høyland and S. Wallace. Generating scenario trees for multistage decision problems. *Management Science*, 47 (2):295–307, 2001. [1](#)
- [13] R. J. Hyndman, D. M. Bashtannyk, and G. K. Grunwald. Estimating and visualizing conditional densities. *Journal of Computational and Graphical Statistics*, 5(4):pp. 315–336, 1996. [25](#)

- [14] M. Kaut and S. W. Wallace. Shape-based scenario generation using copulas. *Computational Management Science*, 8(1–2):181–199, 2011. [1](#)
- [15] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, California, 1967. University of California Press. [7](#)
- [16] G. Ch. Pflug. *Optimization of Stochastic Models*, volume 373 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, 1996. [5](#)
- [17] G. Ch. Pflug. Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programming*, 89:251–271, 2001. [16](#)
- [18] G. Ch. Pflug. Version-independence and nested distribution in multistage stochastic optimization. *SIAM Journal on Optimization*, 20:1406–1420, 2009. [19](#)
- [19] G. Ch. Pflug and A. Pichler. *Approximations for Probability Distributions and Stochastic Optimization Problems*, volume 163 of *International Series in Operations Research & Management Science*, chapter 15, pages 343–387. Springer, New York, 2011. [20](#)
- [20] G. Ch. Pflug and A. Pichler. A distance for multistage stochastic optimization models. *SIAM Journal on Optimization*, 22(1):1–23, 2012. [2](#), [14](#), [20](#)
- [21] S. T. Rachev. *Probability metrics and the stability of stochastic models*. John Wiley and Sons Ltd., West Sussex PO19, 1UD, England, 1991. [26](#)
- [22] A. N. Shiryaev. *Probability*. Springer, New York, 1996. [13](#)
- [23] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall/CRC, 1998. [26](#)
- [24] A. B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2008. [24](#), [26](#)
- [25] C. Villani. *Topics in Optimal Transportation*, volume 58 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2003. [24](#)
- [26] D. Williams. *Probability with Martingales*. Cambridge University Press, Cambridge, 1991. [8](#)

A Conditional method for multivariate distributions

To generate instants from a multivariate distribution (random vectors) with cdf $F(z_1, \dots, z_m)$ one may employ *rejection sampling*, one may generate the random variables by employing the *ratio of uniforms* method, or generate the vector component by component by proceeding as follows:

- (i) Generate Z_1 from $F_1(z)$, where $F_1(z_1) = F(z_1, \infty, \dots, \infty)$ is the first marginal. This can be accomplished by solving

$$U_1 = F_1(Z_1)$$

(the probability integral transform, where U_1 is a uniformly distributed random variable) or by rejection sampling;

- (ii) Given the random vector up to dimension $i-1$ one may generate Z_i conditionally on (Z_1, \dots, Z_{i-1}) by solving

$$U_i = F_i(z_i | Z_1, \dots, Z_{i-1}) = \frac{F(Z_1, \dots, Z_{i-1}, z_i, \infty, \dots, \infty)}{F(Z_1, \dots, Z_{i-1}, \infty, \dots, \infty)},$$

where U_i is uniformly distributed, and independent from U_1, \dots, U_{i-1} .