Distributions and Bootstrap for Data-based Stochastic Programming

Xiaotie Chen - Department of Mathematics
David L. Woodruff*- Graduate School of Management

UC Davis, Davis CA, USA

January 18, 2024

Abstract

In the context of optimization under uncertainty, we consider various combinations of distribution estimation and resampling (bootstrap and bagging) for obtaining samples used to estimate a confidence interval for an optimality gap. This paper makes three experimental contributions to on-going research in data driven stochastic programming: a) most of the combinations of distribution estimation and resampling result in algorithms that have not been published before, b) within the algorithms, we describe innovations that improve performance, and c) we provide open-source software implementations of the algorithms. Among others, three important conclusions can be drawn: using a smoothed point estimate for the optimality gap for the center of the confidence interval is preferable to a purely empirical estimate, bagging generally performs better than bootstrap, and smoothed bagging sometimes performs better than bagging based directly on the data.

Keywords: Optimization under Uncertainty, smoothed bootstrap, bagging, stochastic programming, optimality gap

When presented with an optimization problem under uncertainty, sampled data may inform the inputs to a solution procedure. A solution obtained in this way is almost surely not the optimal solution to the problem due to the inherent stochastics, so a confidence interval (CI) for the objective function optimality gap should be computed. We consider various combinations of distribution estimation and resampling (bootstrap and bagging) for obtaining samples used to estimate the optimality gap a confidence interval for it.

Our paper concerns optimization for a population with unknown distribution F. We state the problem in abstract form using

$$\min_x E_{\xi\sim F} h(x,\xi) \tag{1}$$

We will consider examples with explicit constraints, but for now we use notation where they are implicit. The decision vector is x and the vector of uncertain data is ξ. Data known with certainty, along with constraints, are captured in the specification of the function h. Our interest is when there is a sample $\mathbb{Z}_N = \{\mathfrak{z}_i,\ i = 1,\ldots,N\}$ that can be used to estimate a confidence interval for the optimality gap associated with a given $\hat{x}$. The subscript i indicates the i^{th} vector; we never reference vector elements in this paper.

The optimality gap $\mathcal{G}_{\hat{x}}$ for a decision $\hat{x}$ with respect to a distribution $\mathbb{Q}$ is defined as the deviation of the object function value from the optimal:

$$\mathcal{G}_{\hat{x}}(\mathbb{Q}) = E_{\xi\sim\mathbb{Q}} h(\hat{x},\xi) - \min_x E_{\xi\sim\mathbb{Q}} h(x,\xi).$$

For ease of notation, when $\mathbb{Q}$ is the empirical distribution derived from some sample set $S_N = \{s_i\}_{i=1}^N$, we allow $\mathcal{G}_{\hat{x}}$ to directly accept S_N as input, so that

$$\mathcal{G}_{\hat{x}}(S_N) = \frac{1}{N}\sum_{i=1}^N h(\hat{x},s_i) - \min_x \frac{1}{N}\sum_{i=1}^N h(x,s_i).$$

*DLWoodruff@UCDavis.edu; ORCID 0000-0002-5902-8329

We drop the subscript $\hat{x}$ when it does not cause confusion, and write the optimality gap function as $\mathcal{G}(\cdot)$.

After describing algorithms, we report on simulation experiments to measure the effect of various parameter settings and algorithm design decisions. The relative running times of the algorithms and the quality of their estimates are considered. Out-of-sample tests for the quality of an estimated $\hat{x}$ is fairly straightforward, however, evaluation of α-level confidence interval estimates of the optimality gap is more complicated because their quality is a two-dimensional object. The dimensions are sometimes called *skill*, which refers to the degree to which $1-\alpha$ of subsequently observed data is within the interval, and *sharpness*, which refers to the size of the interval. The time required to compute the estimates adds a third dimension.

This paper makes three contributions to on-going research in data driven stochastic programming: a) most of the combinations of distribution estimation and resampling result in algorithms that have not been published before, b) within the algorithms, we describe innovations that improve performance, and c) we provide open-source software implementations of the algorithms. The end result is significant improvement in the ability to use data to estimate a confidence interval around the objective function value for a candidate solution to an optimization problem under uncertainty.

The paper proceeds as follows. The next section reviews relevant literature, followed by Section 2 that describes the main algorithms. Section 3 describes experiments and conclusions drawn from them. The paper closes with conclusions and directions for future research.

1 Literature Review

Stochastic programming involves modeling optimization problems under uncertainty. There has been a relatively rich literature on how to formulate and solve a stochastic program [King and Wallace, 2012, Birge and Louveaux, 2011, Ruszczyński and Shapiro, 2003, Prékopa, 2013]. The literature mostly discusses the situation where some problem parameters are random variables that follow given distributions, and many use sampling methods to computationally address the problems when it is technically unsolvable. There has been a line of work that discusses uncertainty quantification for stochastic programs when the parameter distributions are known. For example, Mak et al. [1999], De Matos et al. [2017], Linderoth et al. [2006] constructed confidence interval of the optimality gap for a given candidate solution; Higle and Sen [1991], Bayraksan and Morton [2011], Bayraksan and Pierre-Louis [2012] developed sequential sampling methods that produces a series of candidate solutions and estimated the corresponding solution quality. The discussion of asymptotic properties of the solutions can be found in for example, Shapiro [1991, 2003], Eichhorn and Römisch [2007].

Bootstrap has been widely used for statistical inference ever since it was first proposed in Efron [1981], see Efron [1982], Shao and Tu [2012], Davison and Hinkley [1997] for a comprehensive introduction. It can be used to construct confidence intervals when the underlying true distribution remains unknown.

In the area of stochastic programming, an early work of Higle and Sen [1991] used bootstrap to develop stopping rules for the Stochastic Decomposition algorithm, Eichhorn and Römisch [2007], Lam and Qian [2018a] proposed the use of bootstrap and related resampling methods to derive confidence intervals for the optimal function value. Anitescu and Petra [2011] discussed some of the theoretical properties of bootstrap confidence interval for stochastic programming.

The classical bootstrap estimates a statistic of interest, say $\alpha(F)$, by its empirical version $\alpha(F_n)$. There is a literature that discusses the properties of smoothed bootstrap, where the discrete distribution F_n is replaced with a smoothed distribution for estimation, see for example, Efron [1982], Silverman and Young [1987], Hall et al. [1989], De Angelis and Young [1992] for theoretical discussions, and Li and Wang [2008], Fuentes and Lillo-Bañuls [2015] for example applications. The application of smoothed bootstrap in the area of optimization remains limited.

The bagging method, also known as the bootstrap aggregating method, was proposed in Breiman [1996]. It was widely used in machine learning community to produce an accurate and robust prediction by aggregating the predictions of multiple models, where each model uses bootstrap samples from the original data set; see Bühlmann and Yu [2002] and the literature therein for theoretical analysis. Lee and Cho [2001], Raviv and Intrator [1996] proposed smoothed bagging for classification and regression problems, with added noise in the resampled data. More recently Lam and Qian [2018a] proposed use of the bagging method to construct confidence interval for a candidate solution in stochastic programming, and Chen and Woodruff [2023] developed a software package [BOOT-SP, 2023] that implemented the bootstrap and the bagging method for the confidence intervals for stochastic programming. That paper describes software that implements the methods described by Eichhorn and Römisch [2007] and Lam and Qian [2018a]. This is in contrast to the present paper that proposes smoothed bootstrap and bagging algorithms for confidence intervals on the optimality gap and provides empirical contributions.

Despite the lack of the application of smoothed bootstrap in the area of optimization, there are some works on combining probability density estimation with stochastic optimization problems. For example, Huh et al. [2011] used the KM estimator to construct an empirical cdf for censored data for newsvendor problem to solve for the optimal; Parpas et al. [2015] used Markov chain Monte Carlo methods with kernel density estimation algorithms to build a nonparametric importance sampling distribution for recourse function.

2 Algorithms

Below we discuss the bootstrap and bagging procedures for estimating confidence intervals on optimality gaps for general stochastic programming problems, it is important to note that both procedures allow for variations in how the center of the confidence interval is computed and the distribution employed for resampling the data. In cases where the empirical distribution is utilized for bootstrap and bagging, we refer readers to our software boot-sp [Chen and Woodruff, 2023, BOOT-SP, 2023]. This software provides comprehensive tools and methodologies for effectively implementing and leveraging the benefits of these procedures.

2.1 Classical bootstrap for Stochastic Programming

For completeness we include Algorithm 1 that describes the procedure for finding an approximate confidence interval by using the classical bootstrap procedure[Efron, 1981], where the CI is centered at $\mathcal{G}(\mathbb{Z}_N)$, the gap associated with the set $\mathbb{Z}_N$, and the quantile of bootstrap sampled gaps is used to derive the limits.

Algorithm 1: Classical Bootstrap

---

input : A sample $\mathbb{Z}_N = \{\mathfrak{z}_i\}_{i=1}^N$, number of batches B, and a candidate solution $\widehat{x}$

Compute the optimality gap associated with the set $\mathbb{Z}_N$

$$\mathcal{G}_{\widehat{x}}(\mathbb{Z}_N) = \frac{1}{N}\sum_{i=1}^{N} h(\widehat{x}, \mathfrak{z}_i) - \min_x \frac{1}{N}\sum_{i=1}^{N} h(x, \mathfrak{z}_i)$$

for $b \leftarrow 1$ **to** B **do**

Resample from $\mathbb{Z}_N$ to get the bootstrap set $\widetilde{\mathbb{Z}}_N^b = \{\widetilde{\mathfrak{z}}_1^b, \ldots, \widetilde{\mathfrak{z}}_N^b\}$;

Compute the associated gap

$$\mathcal{G}(\widetilde{\mathbb{Z}}_N^b) = \frac{1}{N}\sum_{i=1}^{N} h(\widehat{x}, \widetilde{\mathfrak{z}}_i^b) - \min_x \frac{1}{N}\sum_{i=1}^{N} h(x, \widetilde{\mathfrak{z}}_i^b),$$

end

Compute the upper $1-\alpha$-quantile $\varrho_{1-\alpha}$ and lower α-quantile ϱ_α for $\{\mathcal{G}(\widetilde{\mathbb{Z}}_N^b) - \mathcal{G}(\mathbb{Z}_N)\}$;

Return $[\mathcal{G}(\mathbb{Z}_N) - \varrho_{1-\alpha}, \mathcal{G}(\mathbb{Z}_N) - \varrho_\alpha]$ as the $(1-2\alpha)$ CI for the optimality gap $\mathcal{G}(F)$;

Let $\boldsymbol{\mathbb{Z}}_N$ and $\mathbb{Z}_N$ represent the random set and its realization, respectively. Each $\mathfrak{z}_i$ in $\mathbb{Z}_N$ is a realization from the distribution F. And let $\widetilde{\boldsymbol{\mathbb{Z}}}_N^b$ be the random set whose elements obey the empirical distribution of the set $\mathbb{Z}_N$. The classical bootstrap method is based on the theoretical validation of the asymptotic similarity (Shao and Tu [2012]) between the two distributions, $\mathcal{G}(\boldsymbol{\mathbb{Z}}_N) - \mathcal{G}(F)$ and $\mathcal{G}(\widetilde{\boldsymbol{\mathbb{Z}}}_N^b) - \mathcal{G}(\mathbb{Z}_N)$, with the latter one conditioned on one realization of the random variable $\boldsymbol{\mathbb{Z}}_N$, which is the random sample $\mathbb{Z}_N$. Note that there are a few variations on the classical bootstrap method, in that different metrics can be employed to derive the confidence interval from the resampled gaps $\mathcal{G}(\widetilde{\mathbb{Z}}_N^b)$. For example, instead of using the quantiles of $\mathcal{G}(\widetilde{\mathbb{Z}}_N^b)$ in one way or another, one can also use

$\mathcal{G}(\widetilde{\mathbb{Z}}_N^b)$ to fit a standard normal confidence interval. As in Efron [1981], one can compute the variance of $\mathcal{G}(\widetilde{\mathbb{Z}}_N^b)$, denoted as s^2, and return $[\mathcal{G}(\mathbb{Z}_N) - z_{1-\alpha}s, \mathcal{G}(\mathbb{Z}_N) + z_{1-\alpha}s]$ as the CI, with $z_{1-\alpha}$ being the quantile for the standard Gaussian variable.

2.2 Smoothed Point Estimator

In the classical bootstrap method, the confidence interval is constructed around the point estimator, $\mathcal{G}(\mathbb{Z}_N)$, of the optimality gap under the set $\mathbb{Z}_N$. This is based on the idea that as if one does not have access to the entire population F, and instead all that is available is a sample $\mathbb{Z}_N$, then the natural choice for an estimation of $\mathcal{G}(F)$ is the optimality gap associated with $\mathbb{Z}_N$.

However for many important applications, using only the optimality gap associated with the empirical distribution of $\mathbb{Z}_N$ might not be enough, especially when the sample size is small. In this case, one may seek to use density estimation or probability distribution fitting tools to get a better estimation of the optimality gap. Kernel density estimation and epi-spline fitting Royset and Wets [2015] are two of the possible tools. The idea of using a smoothed estimator is also proposed in Huh et al. [2011], Parpas et al. [2015] in a different setting.

We describe a general procedure to find a point estimator, $\bar{G}$, for $\mathcal{G}(F)$ in Algorithm 2. Instead of $\mathcal{G}(\mathbb{Z}_N)$, one may use $\bar{G}$ from Algorithm 2 as the point estimator for the optimality gap. For the form of fitted distribution, we assume that when given enough data points, the fitted distribution should be able to recover the true distribution.

Algorithm 2: Distribution-based Point Estimator

input : A sample $\mathbb{Z}_N$, replication R, sample size n_c, form of distribution $\breve{F}$.
Fit a distribution function $\breve{F}_N$ using the set $\mathbb{Z}_N$;
for $b \leftarrow 1$ **to** R **do**
  Sample from the distribution $\breve{F}_N$ to get n_c samples $\{\tilde{\mathfrak{z}}_1^b, \ldots, \tilde{\mathfrak{z}}_{n_c}^b\}$;
  Compute $G_j = \frac{1}{n_c}\sum_{i=1}^{n_c} h(\hat{x}, \tilde{\mathfrak{z}}_i^b) - \min_x \frac{1}{n_c}\sum_{i=1}^{n_c} h(x, \tilde{\mathfrak{z}}_i^b)$,
end
Return $\bar{G} = \frac{1}{j}\sum_{j=1}^{R} G_j$ as a point estimator for the optimality gap $\mathcal{G}(F)$;

The above algorithm to some extent provides a unified framework for estimating the optimality gap for stochastic programming problems. We highlight two special cases that link Algorithm 2 back to the point estimators that have been used in the literature.

- Classical Bootstrap: If we do not incorporate smoothness into our form of distribution, but instead use the empirical distribution of $\mathbb{Z}_N$ as the fitted distribution, which assigns an atom of probability with mass $1/N$ to each observation $\mathfrak{z}_i$, then sampling from the distribution $\breve{F}_N$ is equivalent to resampling from the data set $\mathbb{Z}_N$. In this case, with one replication $R = 1$ and $n_c = N$ as the sample size, Algorithm 2 returns $\mathcal{G}(\mathbb{Z}_N)$ as the point estimator, which is the center of the confidence interval in the classical basic bootstrap.

- Classical Bagging: If we use the empirical distribution of $\mathbb{Z}_N$ as $\breve{F}_N$, but run multiple replications $(R > 1)$ with possibly smaller resample size n_c, then the returned $\bar{G}$ is a bagging estimator of the optimality gap. The bagging estimator is used as the center of the confidence interval in Lam and Qian [2018a,b], where they construct a confidence interval around the optimality gap in stochastic programming via bagging.

In the following sections we discuss the situations where we use a smooth function as the fitted distribution function $\breve{F}_N$, and how the existing methods can be adapted to construct a confidence interval around the smoothed point estimator for the optimality gap.

2.3 Smoothed Bootstrap

Instead of directly computing a point estimator $\mathcal{G}(\mathbb{Z}_N)$ using the dataset $\mathbb{Z}_N$ as in classical bootstrap, an alternative approach is to employ a smoothed density estimate, such as kernel density estimation, as depicted in Algorithm 2. This algorithm fits a smoothed distribution $\breve{F}_N$ based on $\mathbb{Z}_N$. Subsequently, a large batch of samples is drawn from $\breve{F}_N$ to obtain an estimation. In this case, the estimated center $\bar{G}$ serves as a reliable approximation for $\mathcal{G}(\breve{F}_N)$, which, in turn, provides an estimation for $\mathcal{G}(F)$.

Now of course the next question becomes how to construct a confidence interval around $\bar{G}$. The most obvious way would be to directly apply the classical Algorithm 1 . That is, we use $\bar{G}$ in place of $\mathcal{G}(\mathbb{Z}_N)$ in Algorithm 1, and the corresponding return should be a CI for the optimality gap $\mathcal{G}(F)$.

However, if we use a smoothed function for fitting the distribution function $\breve{F}_N$, then it is natural to introduce some smoothness to the bootstrap procedure as well, as in the practice of the standard smoothed bootstrap. That is, instead of resampling from the empirical distribution of $\mathbb{Z}_N$ in Algorithm 1, we instead resample from $\breve{F}_N$, a smoothed version of the empirical c.d.f., for bootstrap samples. We use the same smoothed distribution $\breve{F}_N$ for finding the center and for estimating the confidence interval.

We describe in Algorithm 3 the procedure to find a CI for the optimality gap in conjunction with the smoothed point estimator returned by Algorithm 2. We use standard normal confidence interval in our algorithm and estimate the width of the confidence interval by estimating the variance of the limit distribution, but the percentile bootstrap interval or the bias-corrected and accelerated bootstrap interval (Diciccio and Romano [1988]) could also be used here.

Algorithm 3: Smoothed Bootstrap

input : A sample $\mathbb{Z}_N$, number of batches B, form of distribution $\breve{F}$, and a candidate solution $\hat{x}$

Fit a smoothed distribution function $\breve{F}_N$ using the set $\mathbb{Z}_N$;

Run Algorithm 2 with the same $\breve{F}_N$, $R = 1$ and a sufficiently large resample size n_c to find the point estimator $\bar{G}$.

for $b \leftarrow 1$ **to** B **do**

  Sample from distribution $\breve{F}_N$ to get a new set $\widetilde{\mathbb{Z}}_N^b = \{\tilde{\mathfrak{z}}_1^b, \ldots, \tilde{\mathfrak{z}}_N^b\}$;

  Compute $\mathcal{G}(\widetilde{\mathbb{Z}}_N^b) = \frac{1}{N}\sum_{i=1}^N h(\hat{x}, \tilde{\mathfrak{z}}_i^b) - \min_x \frac{1}{N}\sum_{i=1}^N h(x, \tilde{\mathfrak{z}}_i^b)$,

end

Compute the sample variance s^2 for $\mathcal{G}(\widetilde{\mathbb{Z}}_N^b)$;

Return $[\bar{G} - z_{1-\alpha}s, \bar{G} + z_{1-\alpha}s]$ as the CI.

Algorithm 3 is a small generalization of the classical smoothed bootstrap method, in that we allow different options for the point estimator that serves as the center of the confidence interval. When $\bar{G}$ equals $\mathcal{G}(\mathbb{Z}_N)$, i.e. when the empirical distribution is used in Algorithm 2 for providing a point estimator, Algorithm 3 is identical to the classical smoothed bootstrap [Efron, 1982].

Since $\bar{G}$ can essentially be regarded as a smoothed version of $\mathcal{G}(\mathbb{Z}_N)$, the same theory that is used to support the classical smoothed bootstrap method can be used here to justify the asymptotic consistency of the output of Algorithm 1. That is, conditioned on $\mathbf{Z}_N = \mathbb{Z}_N$, let $\widetilde{\mathbf{Z}}_N^b$ be the random set whose elements obey distribution $\breve{F}_N$, the distribution of $\mathcal{G}(\widetilde{\mathbf{Z}}_N^b) - \bar{G}$ is asymptotically similar to the distribution of $\bar{G} - \mathcal{G}(F)$(De Angelis and Young [1992]). Notice that $\bar{G}$ depends on $\breve{F}_N$ and hence in turn is correlated to $\mathbf{Z}_N$.

2.4 Smoothed Bagging

The bagging method for estimating the confidence interval for the optimality gap was proposed by Lam and Qian [2018a] and is implemented in our software tool boot-sp [2023]. Compared with bootstrap method, the bagging method is known to reduce the variance in general applications. The center of the bagging confidence interval is constructed alongside the bagging procedure. For completeness we include the original, non-smoothed algorithm as Algorithm 4.

The first few steps of the bagging algorithms conforms with Algorithm 2 for finding the center of the confidence interval as outlined in Section 2.2. The rest steps aims to construct an empirical version of the infinitesimal jackknife estimator of the variance, which in turn guides the construction of the confidence interval.

Following the same argument as in the smoothed bootstrap, one may wish to introduce some smoothness into the bagging estimator, especially when the sample size is small. So instead of resam-

Algorithm 4: Bagging-based sampling

input : A sample $\mathbb{Z}_N$, number of bags B, bag sample size k, significance level α, and a candidate solution $\hat{x}$

for $b \leftarrow 1$ **to** B **do**

  Resample from $\mathbb{Z}_N$ to get a bagging set of size k, $\widetilde{\mathbb{Z}}_N^b = \{\tilde{\mathfrak{z}}_1^b, \ldots, \tilde{\mathfrak{z}}_k^b\}$;

  Compute $\mathcal{G}(\widetilde{\mathbb{Z}}_N^b) = \frac{1}{k}\sum_{i=1}^k h(\hat{x}, \tilde{\mathfrak{z}}_i^b) - \min_x \frac{1}{k}\sum_{i=1}^k h(x, \tilde{\mathfrak{z}}_i^b)$, ,

end

Compute the mean of $\mathcal{G}(\widetilde{\mathbb{Z}}_N^b)$ as the center of the confidence interval, so

$$G = \frac{1}{B}\sum_{b=1}^{B} \mathcal{G}(\widetilde{\mathbb{Z}}_N^b)$$

Compute the error term

$$\tilde{\sigma}^2 = \begin{cases} \sum_{i=1}^n \widehat{cov}_i^2 & \text{if with replacement} \\ \frac{n^2}{(n-k)^2}\sum_{i=1}^n \widehat{cov}_i^2 & \text{if without replacement} \end{cases},$$

where

$$\widehat{cov}_i = \frac{1}{B}\sum_{b=1}^{B}(N_i^b - k/n)(\mathcal{G}_{gap}(\widetilde{\mathbb{Z}}_N^b) - G),$$

and N_i^b = number of times the i^{th} element of $\mathbb{Z}_N$ appears in $\widetilde{\mathbb{Z}}_N^b$;

Return $[G - z_{1-\alpha/2}\tilde{\sigma}, G + z_{1-\alpha/2}\tilde{\sigma}]$ as the $(1-\alpha)$ CI for the optimality gap $\mathcal{G}(F)$, with $z_{1-\alpha/2}$ being the $(1-\alpha/2)$ quantile for a standard normal variable;

pling from the set $\mathbb{Z}_N$ to get the bagging sets, we resample from a fitted distribution $\check{F}_N$ to get the samples and compute the gaps, and take the average of the gaps to be our point estimator.

As the infinitesimal jackknife estimator of the variance for the bagging estimator in Algorithm 4 does not directly apply for a smoothed bagging estimator, we seek an alternative approach to estimate the variance using the results from Mentch and Hooker [2016]. In particular, consider $\mathcal{G}_k$ as a kernel function $\mathcal{G}_k(\mathfrak{z}_1, \ldots, \mathfrak{z}_k)$ that returns the optimality gap associated with scenarios $\{\mathfrak{z}_1, \ldots, \mathfrak{z}_k\}$, with

$$\mathcal{G}_k(\mathfrak{z}_1, \ldots, \mathfrak{z}_k) = \frac{1}{k}\sum_{i=1}^k h(\hat{x}, \mathfrak{z}_i) - \min_x \frac{1}{k}\sum_{i=1}^k h(x, \mathfrak{z}_i)$$

and let

$$\varsigma_{c,k} = var\left(E[\mathcal{G}_k(\boldsymbol{\mathfrak{z}}_1, \ldots, \boldsymbol{\mathfrak{z}}_k)|\boldsymbol{\mathfrak{z}}_1 = \mathfrak{z}_1, \ldots, \boldsymbol{\mathfrak{z}}_c = \mathfrak{z}_c]\right),$$

which is actually the covariance between two instances of the function h with c shared arguments.

By [Mentch and Hooker, 2016, Theorem 1], the variance of the bagging estimator can be estimated with

$$\sigma^2 = \frac{1}{B}\left(\frac{k^2 B}{n}\varsigma_{1,k} + \varsigma_{k,k}\right),$$

with B being the number of bagging sets used to construct the point estimator.

The two variances $\varsigma_{1,k}$ and $\varsigma_{k,k}$ can be estimated using similar Monte Carlo methods as depicted in [Mentch and Hooker, 2016, Section 3]. To estimate $\varsigma_{1,k}$, we start by randomly select one scenario $\tilde{\mathfrak{z}}^{(1)}$ and fixing $\boldsymbol{\mathfrak{z}}_1 = \tilde{\mathfrak{z}}^{(1)}$, then choose B_{MC} bagging sets $\{\widetilde{\mathbb{Z}}_N^{b'}\}_{b'=1}^{B_{MC}}$, each of size k and contains the fixed seed scenario $\tilde{\mathfrak{z}}(1)$. Because each bagging set corresponds to an associated optimality gap, the average of the B_{MC} gaps serves as a Monte Carlo approximation to the mean

$$E[\mathcal{G}_k(\boldsymbol{\mathfrak{z}}_1, \ldots, \boldsymbol{\mathfrak{z}}_k)|\boldsymbol{\mathfrak{z}}_1 = \tilde{\mathfrak{z}}^{(1)}] \approx \bar{G}^{(1)} \stackrel{def}{=} \frac{1}{B_{MC}}\sum_{b'=1}^{B_{MC}} \mathcal{G}_k(\widetilde{\mathbb{Z}}_N^{b'}).$$

We then repeat the above steps for B_I times, each time with a independently selected fixed, seed scenario $\tilde{\mathfrak{z}}^{(i)}$. The sample variance among the B_I averaged gaps $\bar{\mathcal{G}}^{(i)}$ can then be used as the estimator for $\varsigma_{1,k}$,

$$\varsigma_{1,k} \approx var(\bar{G}^{(1)}, \ldots, \bar{G}^{(B_I)})$$

The variance $\varsigma_{k,k}$ can be estimated using similar approach, by independently sampling B bagging sets, and computing the variance of the corresponding bagging gaps. But instead of computing the point estimator G and the variances $\varsigma_{1,k}$ and $\varsigma_{k,k}$ in three separate runs, we can incorporate the three procedures into one by utilizing the $B_I * B_{MC}$ gaps (each fixed initial seed scenario yields B_{MC} gaps, and we have in total B_I initial fixed seed scenarios) that are used to estimate $\varsigma_{1,k}$. That is, after generating $B = B_I * B_{MC}$ bagging sets and finding an estimate for the variance $\varsigma_{1,k}$, we take the average of those B gaps

$$G = \frac{1}{B_I * B_{MC}} \sum \mathcal{G}_k(\widetilde{\mathbb{Z}}_N^b) = \frac{1}{B_I} \sum_{b=1}^{B_I} \bar{G}^{(b)}$$

to be our bagging estimator G, and use the sample variance of the B gaps as an estimation for $\varsigma_{k,k}$. Algorithm 5 outlines the procedure for constructing a confidence interval around the smoothed bagging estimator. The sample variance s_1 is used to estimate $\varsigma_{1,k}$, and s_2 is for $\varsigma_{k,k}$.

Algorithm 5: Smoothed Bagging with Variance estimation

---

input : A sample $\mathbb{Z}_N$, number of initial seed points B_I, number of Monte Carlo simulations for each initial points B_{MC},subsample size k, significance level α, and a candidate solution $\hat{x}$

Fit a smoothed distribution function for $\breve{F}_N$ using the set $\mathbb{Z}_N$;

for $b \leftarrow 1$ **to** B_I **do**

  Select initial seed point $\tilde{\mathfrak{z}}^{(b)}$, by sampling from the fitted distribution;

  for $b' \leftarrow 1$ **to** B_{MC} **do**

    Resample from $\breve{F}_N$ to get bagging set of size k, $\widetilde{\mathbb{Z}}_k^{b,b'} = \{\tilde{\mathfrak{z}}_1^{b,b'}, \ldots, \tilde{\mathfrak{z}}_k^{b,b'}\}$, that includes the initial seed point $\tilde{\mathfrak{z}}^{(b)}$;

    Compute $\mathcal{G}(\widetilde{\mathbb{Z}}_k^{b,b'}) = \frac{1}{k}\sum_{i=1}^k h(\hat{x}, \tilde{\mathfrak{z}}_i^{b,b'}) - \min_x \frac{1}{k}\sum_{i=1}^k h(x, \tilde{\mathfrak{z}}_i^{b,b'})$;

  end

  Compute the average of the B_{MC} gaps, denoted as $\bar{G}^{(b)}$;

end

Compute the mean of $\bar{G}^{(b)}$ as the center of the confidence interval, so

$$G = \frac{1}{B_I} \sum_{b=1}^{B_I} \bar{G}^{(b)} = \frac{1}{B_I * B_{MC}} \sum \mathcal{G}(\widetilde{\mathbb{Z}}_k^{b,b'})$$

Compute the variance s_1^2 for the B_I averages $\bar{G}^{(b)}$;

Compute the variance s_2^2 for all $\mathcal{G}(\widetilde{\mathbb{Z}}_k^{b,b'})$;

Return $[G - z_{1-\alpha/2}s, G + z_{1-\alpha/2}s]$ as the $(1-\alpha)$ CI for the optimality gap $\mathcal{G}(F)$ with

$$s^2 = \frac{1}{B_I \cdot B_{MC}} \left(\frac{k^2 \cdot B_I \cdot B_{MC}}{N} s_1^2 + s_2^2 \right),$$

and $z_{1-\alpha/2}$ being the $(1-\alpha/2)$ quantile for a standard normal variable;

---

3 Experimental Results

3.1 Problem Examples

We conducted experiments over three different example problems. For each of the problems, we report on a series of experiments and compare the coverage rate, the width of the interval, and the computational time to generate confidence interval. The algorithms discussed in this paper, along with the examples used in our experiments, are implemented in boot-sp [BOOT-SP, 2023]. Tables from additional experiments can be found at https://github.com/boot-sp/boot-sp/doc/pdfs/smoothed_bagging_tables.pdf.

3.1.1 CVaR

A one-stage CVaR problem as in Lam and Qian [2018a]:

$$\min_x \left\{ x + \frac{1}{a} E\left[(\xi - x)_+\right] \right\}$$

where $(\cdot)_+$ is defined as $\max \cdot, 0$, $a = 0.1$ and ξ is a drawn from a standard normal distribution.

3.1.2 Scalable Farmer

This example is derived from the well-established farmer example from Birge and Louveaux [2011]. It has been adapted for stress-testing various pieces of software such as Knueven et al. [2020]. To enable scalability, instance configuration parameters *cropsmult* and N are added. The original problem has three crops and three base scenarios. By incorporating these parameters, the scalable instances are created with *cropsmult* sets of the original crops. All scenarios are grouped in threes, with a uniformly distributed pseudo-random number added to the yield values of the original three scenarios. These retain the characteristics of the initial problem, while yields vary according to the specific scenario.

We further introduced a new feature, denoted as "yield-cv," which represents the coefficient of variation of the crop yields. This inclusion offers the flexibility to introduce variability in problem settings, and is universally applicable to all crops. In cases where it is not explicitly specified, the distribution of the farmer example adheres to the original model with uniform distributions.

3.1.3 Multi Knapsack

This problem is derived from the stochastic programming problem in Vaagen and Wallace [2008] (also see [King and Wallace, 2012, Chapter 6] for discussion). The problem can be viewed as a multidimensional newsvendor problem with substitution. The main source of uncertainty in this problem stems from the unpredictable popularity and demand for fashion products, and a simple two-stage stochastic program is formulated with the goal of maximizing profit. In the first stage, production decisions are made, while in the second stage (after demand is observed), the program optimally allocates direct and substitution sales. In our experiments, we consider the sale of six products with a universal substitution rate at 0.1.

3.2 Experimental Results

To simplify and enhance the clarity of our experimental outcomes, we employ abbreviations for denoting our suggested algorithms. Specifically, we use "BT" instead of "bootstrap algorithms," and "BG" instead of "bagging algorithms." We utilize the prefix "S" to signify the application of the smoothed fitted distribution, while the suffix "K/E" will indicate the choice between kernel density estimation and epi-spline fitting. Furthermore, we append the suffix "Q" to "BT" when employing the quantile method for creating bootstrap confidence intervals. Since there was no substantial difference in the outcomes between bagging with replacement and bagging without replacement, the subsequent figures and tables will exclusively display results obtained using bagging with replacement, which is indicated by the method label "BG". Times are given in seconds.

Table 1 summarizes the distributions that are used in our algorithms and whether or not the point estimators are obtained via aggregation in each method. The notation F_N is used to represent the empirical distribution derived from the sample, $F_N = \frac{1}{N}\sum \delta_{\mathfrak{z}_i}$, while $\check{F}_N$ is employed to denote the fitted smoothed distribution function, which can be obtained through methods such as kernel density estimation or epi-spline fitting.

method	Algorithm	Distribution Used	Aggregated Point Estimator
BT	1	F_N	No
S-BT	3	$\breve{F}_N$	No
BG	4	F_N	Yes
S-BG	5	$\breve{F}_N$	Yes

Table 1: Method summary

3.2.1 Summary of Point Estimator Comparisons

We present box plots that illustrate the differences among various point estimators employed to approximate the optimality gaps using Algorithm 2 and the true ground truth optimality gap.

While in the CVaR example in Figure 1, all of the tested point estimators exhibited a relatively significant error when compared to the actual scale of the true optimality gap, in the farmer example in Figure 2 and the multi-knapsack problem in Figure 3, the height of the box plots, which represents the error in the point estimators, decreased as we introduce smoothness into the point estimator. This reduction in variance suggests that these point estimators may be more effective in approximating the optimality gap in these scenarios.

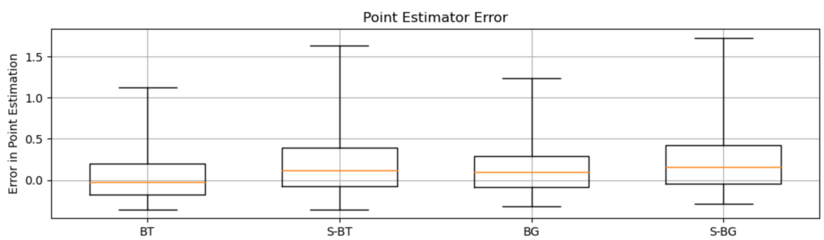

Figure 1: Results for CVaR problem based on 500 replications. The box-plot displays the distribution of errors in the point estimators with respect to the actual optimality gap 0.36. The size of the set $\mathbb{Z}_N$ is fixed at $N = 40$. The BT point estimator is derived by directly computing the gap associated with the set $\mathbb{Z}_N$. The S-BT estimator is estimated by resampling $n_c = 8N$ data points from the fitted distribution. For BG and S-BG, we uses $R = 400$ replications and the resample size is $n_c = 30$.

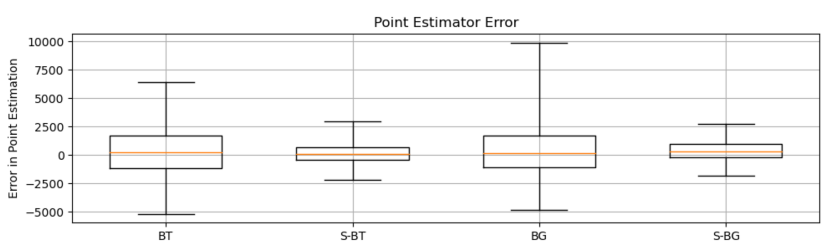

Figure 2: Results for farmer problem based on 500 replications. The box-plot displays the distribution of errors in the point estimators with respect to the actual optimality gap 7648.32. The size of the set $\mathbb{Z}_N$ is fixed at $N = 40$. The BT point estimator is derived by directly computing the gap associated with the set $\mathbb{Z}_N$. The S-BT estimator is estimated by resampling $n_c = 8N$ data points from the fitted distribution. For BG and S-BG, we uses $R = 400$ replications and the resample size is $n_c = 30$.

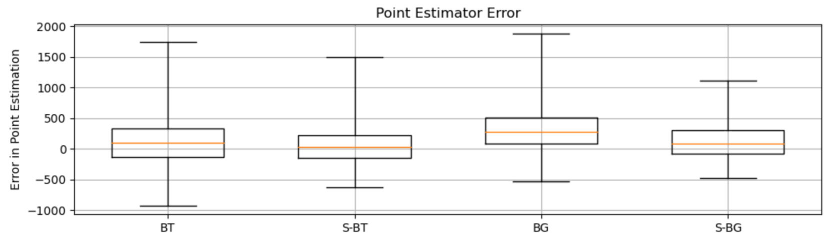

Figure 3: Results for multi-knapsack problem based on 500 replications. The box-plot displays the distribution of errors in the point estimators with respect to the actual optimality gap 2301.95. The size of the set $\mathbb{Z}_N$ is fixed at $N = 40$. The BT point estimator is derived by directly computing the gap associated with the set $\mathbb{Z}_N$. The *S-BT* estimator is estimated by resampling $n_c = 8N$ data points from the fitted distribution. For *BG* and *S-BG*, we uses $R = 400$ replications and the resample size is $n_c = 30$.

3.2.2 Summary of Method Comparisons

We provide plots that compares the coverage rates and the length of the generated confidence intervals for different methods.

Based on our initial set of experiments, we have observed that, at least for the three examples we have tested, the bagging methods has a higher coverage rate for the confidence intervals. Within the distinct categories of bootstrap and bagging, we have noticed that when we introduce a ”smoothing” effect by incorporating kernel density estimation instead of the empirical distribution for resampling, the coverage rate increases, often without sacrificing the length of the confidence interval. See Figures 4, 5, 6.

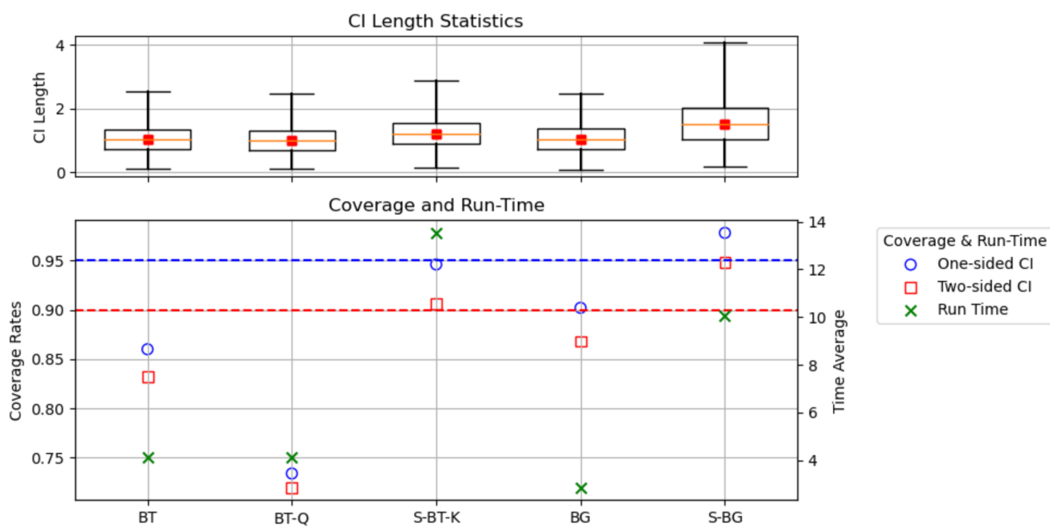

Figure 4: Results for CVaR based on 500 replications. Two-sided CI reports the coverage rate for the two-sided 90% interval, and One-sided CI reports the coverage rate for the one-sided 95% interval. The center for the smoothed bootstrap method is estimated by resampling $8N$ data points from the fitted distribution. The size of the set $\mathbb{Z}_N$ is fixed at $N = 40$ and the number of bootstrap replications is fixed at $B = 400$. The sub-sample size for the bagging and the smoothed bagging methods is $k = 30$. The smoothed bagging methods used $B_I = 10$ and $B_{MC} = 40$, so that the total number of batches matches the one in the original bagging procedure.

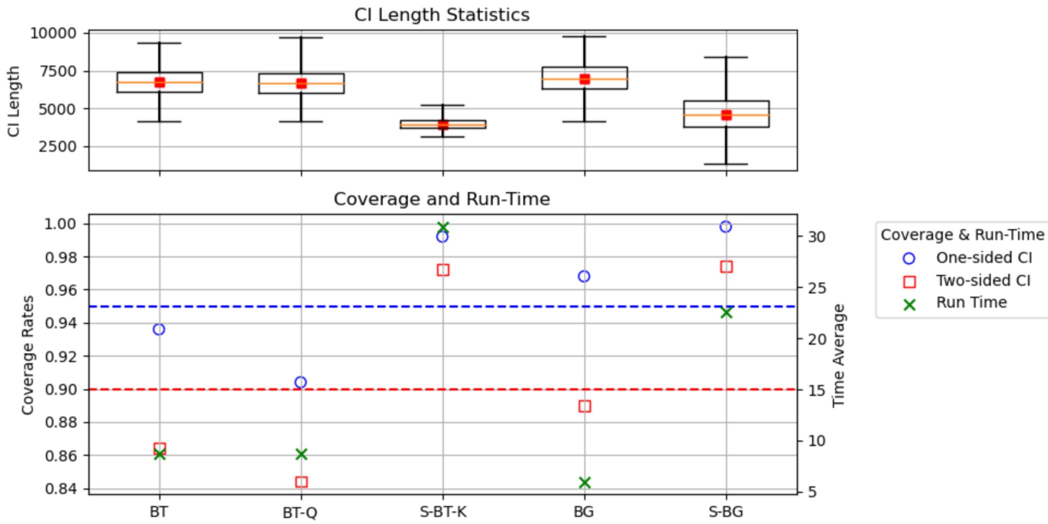

Figure 5: Results for farmer based on 500 replications. Two-sided CI reports the coverage rate for the two-sided 90% interval, and One-sided CI reports the coverage rate for the one-sided 95% interval. The center for the smoothed bootstrap method is estimated by resampling $8N$ data points from the fitted distribution. The size of the set $\mathbb{Z}_N$ is fixed at $N = 40$ and the number of bootstrap replications is fixed at $B = 400$. The sub-sample size for the bagging and the smoothed bagging methods is $k = 30$. The smoothed bagging methods used $B_I = 10$ and $B_{MC} = 40$, so that the total number of batches matches the one in the original bagging procedure.

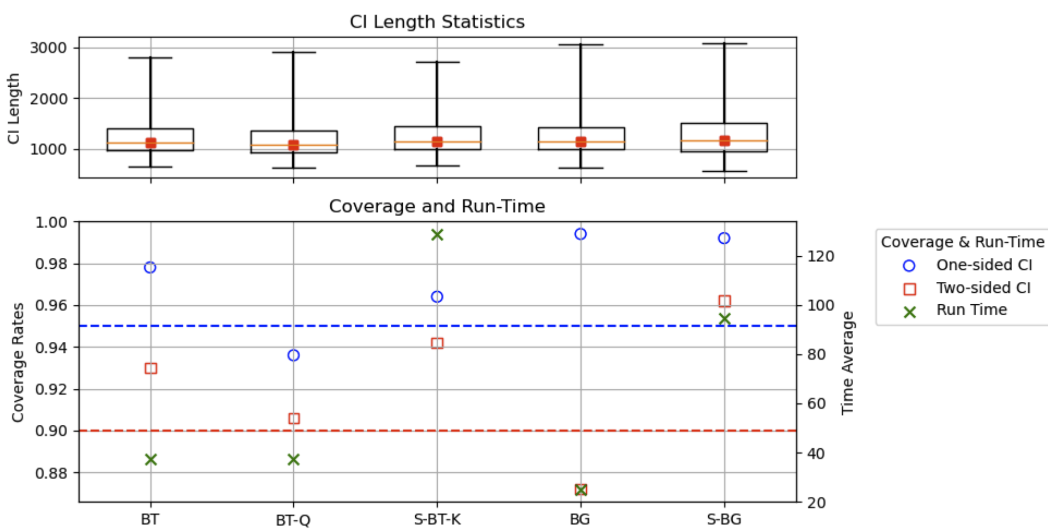

Figure 6: Results for multi-knapsack problem based on 500 replications. Two-sided CI reports the coverage rate for the two-sided 90% interval, and One-sided CI reports the coverage rate for the one-sided 95% interval. The center for the smoothed bootstrap method is estimated by resampling $8N$ data points from the fitted distribution. The size of the set $\mathbb{Z}_N$ is fixed at $N = 40$ and the number of bootstrap replications is fixed at $B = 400$. The sub-sample size for the bagging and the smoothed bagging methods is $k = 30$. The smoothed bagging methods used $B_I = 10$ and $B_{MC} = 40$, so that the total number of batches matches the one in the original bagging procedure.

3.2.3 Additional Experimental Results for Bootstrap

In our experiments concerning the parameters of Algorithm 3, we resample n_c points from the fitted smoothed distribution, and use the optimality gap associated with the n_c points as our point estimator

$\bar{G}$. It can be seen from Tables 2 and 3 that enhanced performance is achieved by resampling more datapoints from the fitted distribution, as elevating the resample size n_c for the center estimator corresponds to an increase in the coverage rates of the two-sided confidence interval as well.

method	n_c	coverage-2	coverage-1
S-BT-K	320	0.906	0.946
S-BT-K	200	0.880	0.958
S-BT-K	120	0.863	0.965

Table 2: Results for CVaR with α=0.05 based on 500 replications. The size of the set $\mathbb{Z}_N$ is fixed at $N = 40$, and the number of batches for bootstrap replications is fixed at $B = 400$. Center estimated using varying n_c data points. coverage-2 reports coverage rates for two-sided 90% confidence interval, and coverage-1 reports coverage rates for one-sided 95% confidence interval. The length of the confidence interval remains the same.

method	n_c	coverage-2	coverage-1
S-BT-K	320	0.972	0.992
S-BT-K	200	0.958	0.988
S-BT-K	120	0.932	0.975

Table 3: Results for farmer with α=0.05 based on 500 replications. The size of the set $\mathbb{Z}_N$ is fixed at $N = 40$, and the number of batches for bootstrap replications is fixed at $B = 400$. Center estimated using varying n_c data points. coverage-2 reports coverage rates for two-sided 90% confidence interval, and coverage-1 reports coverage rates for one-sided 95% confidence interval. The length of the confidence interval remains the same.

For reference we included the results for the smoothed bootstrap algorithm as in Algorithm 3, but use the empirical point estimator $\mathcal{G}(\mathbb{Z}_N)$ as the point estimator in constructing the confidence interval. As seen in Table 4, the confidence interval utilizing the empirical point estimator $\mathcal{G}(\mathbb{Z}_N)$ as the center yields sub-optimal coverage rates and should be avoided in practice.

problem	method	coverage-2	coverage-1
CVaR	S-BT-K	0.703	0.713
farmer	S-BT-K	0.523	0.738

Table 4: Results with α=0.05 based on 500 replications. The size of the set $\mathbb{Z}_N$ is fixed at $N = 40$, and the number of batches for bootstrap replications is fixed at $B = 400$.

Using the kernel density estimation works well for getting good results with the smoothed bootstrap method, as shown in Table 5. It seems to be just as good as or even better than using epi-spline fitting. We noticed that when we switch from epi-spline fitting to kernel density estimation, the coverage rate increases with the same settings. By default, the bandwidth utilized in the kernel density estimation within our code is determined by Scott's Rule [Scott, 2015]. However, before finalizing the bandwidth selection, we visually inspect the results of the kernel density estimation to ensure that the resulting curve strikes a balance between smoothness and avoiding over smoothing. We defer the exploration of the impacts of varying bandwidths to future research endeavors.

problem	method	len-avg	coverage-2	coverage-1
CVaR	S-BT-E	1.02	0.845	0.880
CVaR	S-BT-K	1.23	0.906	0.946
farmer	S-BT-E	3803.31	0.963	0.985
farmer	S-BT-K	3978.63	0.972	0.992

Table 5: Results with α=0.05 based on 500 replications. The size of the set $\mathbb{Z}_N$ is fixed at $N = 40$, and the number of batches for bootstrap replications is fixed at $B = 400$. The center for the smoothed bootstrap method is estimated by resampling $8N$ data points from the fitted distribution.

3.2.4 Additional Experimental Results for Bagging

The bagging methods work well in constructing a confidence interval, even when we only have a really small data set at hand. For example, in Table 6, with a sample size at $N = 20$, one is still able to apply the bagging methods to construct a confidence interval with high coverage rate. The introduced smoothness introduces more randomness into the problem, with increase lengths for the confidence intervals and increased coverage rates with small dataset, as can be seen from Table 6.

problem	method	B	len-avg	len-std	coverage-2	coverage-1
CVaR	BG	200	1.51	0.83	0.902	0.927
CVaR	S-BG	10/40	1.86	1.08	0.860	0.973
CVaR	S-BG	20/80	1.87	0.91	0.939	0.985
farmer	BG	200	9564.85	1973.87	0.885	0.959
farmer	S-BG	10/40	9907.56	1452.74	1.000	1.000
farmer	S-BG	20/80	9826.73	897.56	1.000	1.000
knapsack	BG	200	1920.37	616.58	0.743	0.998
knapsack	S-BG	10/40	2366.64	697.35	0.887	0.995
knapsack	S-BG	20/80	2451.05	495.75	0.950	0.998

Table 6: Results based on 800 replications. The size of the set $\mathbb{Z}_N$ is fixed at $N = 20$, and the subsample size $k = N/2$. Coverage-2 reports the coverage rate for the two-sided 90% interval, and coverage-1 reports the coverage rate for the one-sided 95% interval. For the smoothed bagging method, the first number in the B column represents B_I, and the second one is B_{MC}.

Certainly, a larger dataset $\mathbb{Z}_N$ results in an improved estimation of the confidence interval. In Table 7, when we increase the sample size from $N = 20$ to $N = 40$, we observe reductions in both the average length and the standard deviation of the interval without sacrificing the superior coverage rate.

method	N	B	len-avg	len-std	coverage-2	coverage-1
BG	20	200	1.51	0.83	0.902	0.927
S-BG	20	20/80	1.87	0.91	0.939	0.985
BG	40	200	1.10	0.47	0.900	0.926
S-BG	40	20/80	1.29	0.52	0.916	0.981

Table 7: Results for CVaR based on 800 replications. The subsample size k is fixed at $k = N/2$ .Coverage-2 reports the coverage rate for the two-sided 90% interval, and coverage-1 reports the coverage rate for the one-sided 95% interval.

For smoothed bagging algorithm, both a sufficiently large B_I and a sufficiently large B_{MC} is required to obtain a good coverage without excessive long length in the confidence interval, but as indicated in Mentch and Hooker [2016], it is more critical to use a large B_{MC} to obtain an accurate estimation for the variance, see Table 8.

method	B-I	B-MC	len-avg	len-std	coverage-2	coverage-1
S-BG	10	20	1.69	0.75	0.968	0.994
S-BG	10	100	1.22	0.62	0.879	0.968
S-BG	20	20	1.77	0.67	0.981	0.995
S-BG	20	100	1.30	0.54	0.934	0.980
S-BG	30	20	1.78	0.63	0.983	0.995
S-BG	30	100	1.33	0.51	0.949	0.984

Table 8: Results for CVaR based on 800 replications. The size of the set $\mathbb{Z}_N$ is fixed at $N = 40$, and the subsample size $k = N/2$. Coverage-2 reports the coverage rate for the two-sided 90% interval, and coverage-1 reports the coverage rate for the one-sided 95% interval. For the smoothed bagging method, the first number in the B column represents B_I, and the second one is B_{MC}.

4 Conclusion and Future Directions

In this paper, we introduced various combinations of distribution estimation and resampling techniques for data-driven stochastic programming problems. Specifically, we adapted the smoothed bootstrap method and developed the smoothed bagging method in the context of stochastic optimization. These algorithms are designed for acquiring solutions and computing confidence intervals for the optimality gap. Our experiments demonstrated their effectiveness constructing confidence intervals for small datasets, albeit with a longer computational time compared with empirical bootstrap and bagging algorithms.

Among others, three important conclusions stand out. First, using a smoothed point estimate for the optimality gap at the center of the confidence interval is favored over relying on a purely empirical estimate. The introduced smoothness tends to yield more consistent estimates. Second, our results show that bagging methods often outperform bootstrap methods, providing better coverage rates and tighter confidence intervals across various scenarios. This superior performance may be attributed to bagging's inherent ability to reduce variance and improve the stability of predictions. Third, the smoothed bagging procedure proposed in this paper introduces an additional layer of smoothing to the resampling process. It can, at times, improve results compared to bagging based directly on the data. This approach effectively balances bias and variance, offering a compelling alternative for constructing more robust and accurate confidence intervals.

Despite the promising results, there are several questions that remain for further investigation in future research. One of the key questions is to understand the types of problems that benefit most from the smoothness effect introduced by our proposed algorithms. As our methods offer a trade-off between achieving a high coverage rate and reducing running time, it becomes interesting to explore how many bootstrap samples are sufficient to produce accurate estimations. Additionally, there is potential to investigate whether it is possible to construct more robust confidence intervals using a limited number of bootstrap samples. One could also delve into alternative methods for computing the variance of the estimates, potentially leading to a more efficient and accurate way for constructing the confidence interval.

Also, in our experiments, we primarily used kernel density estimation with bandwidth determined by the Scott's Rule for non-parametric estimations. Subsequent research could be directed towards deriving an optimal bandwidth tailored to specific problem domains. While our experiments revealed that epi-spline fitting did not outperform kernel density estimation, further exploration is warranted. This might include refining hyperparameter tuning, adding constraints to epi-spline fitting, and assessing their influence on algorithm performance, particularly when guided by prior knowledge. In other domains smoothed bootstrap has been proposed for dealing with time series data (e.g. Gregory et al. [2018] for quantile regression). The use of smoothed bootstrap for such data in the context of stochastic programming remains as a future research topic. A final area for future research is the use of smoothed bootstrap and bagging for obtaining an incumbent solution, $\widehat{x}$.

Competing Interests

There are no competing interests for this work.

References

M. Anitescu and C. Petra. Higher-order confidence intervals for stochastic programming using bootstrapping. Technical report, Citeseer, 2011.

G. Bayraksan and D. P. Morton. A sequential sampling procedure for stochastic programming. *Operations Research*, 59(4):898–913, 2011.

G. Bayraksan and P. Pierre-Louis. Fixed-width sequential stopping rules for a class of stochastic programs. *SIAM Journal on Optimization*, 22(4):1518–1548, 2012.

J. R. Birge and F. Louveaux. *Introduction to stochastic programming.* Springer Science & Business Media, 2011.

boot-sp. Boot-sp software. https://github.com/boot-sp/boot-sp, 2023.

L. Breiman. Bagging predictors. *Machine learning*, 24:123–140, 1996.

P. Bühlmann and B. Yu. Analyzing bagging. *The annals of Statistics*, 30(4):927–961, 2002.

X. Chen and D. L. Woodruff. Software for data-based stochastic programming using bootstrap estimation. *to appear in INFORMS Journal on Computing*, 2023.

A. C. Davison and D. V. Hinkley. *Bootstrap methods and their application*. Number 1. Cambridge university press, 1997.

D. De Angelis and G. A. Young. Smoothing the bootstrap. *International Statistical Review/Revue Internationale de Statistique*, pages 45–56, 1992.

V. L. De Matos, D. P. Morton, and E. C. Finardi. Assessing policy quality in a multistage stochastic program for long-term hydrothermal scheduling. *Annals of Operations Research*, 253:713–731, 2017.

T. J. Diciccio and J. P. Romano. A review of bootstrap confidence intervals. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 50(3):338–354, 1988.

B. Efron. Nonparametric estimates of standard error: the jackknife, the bootstrap and other methods. *Biometrika*, 68(3):589–599, 1981.

B. Efron. *The jackknife, the bootstrap and other resampling plans*. SIAM, 1982.

A. Eichhorn and W. Römisch. Stochastic integer programming: Limit theorems and confidence intervals. *Mathematics of Operations Research*, 32(1):118–135, 2007.

R. Fuentes and A. Lillo-Bañuls. Smoothed bootstrap malmquist index based on dea model to compute productivity of tax offices. *Expert systems with applications*, 42(5):2442–2450, 2015.

K. B. Gregory, S. N. Lahiri, and D. J. Nordman. A smooth block bootstrap for quantile regression with time series. *The Annals of Statistics*, 46(3):1138–1166, 2018. ISSN 00905364, 21688966. URL https://www.jstor.org/stable/26542820.

P. Hall, T. J. DiCiccio, and J. P. Romano. On smoothing and the bootstrap. *The Annals of Statistics*, pages 692–704, 1989.

J. L. Higle and S. Sen. Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Mathematics of operations research*, 16(3):650–669, 1991.

W. T. Huh, R. Levi, P. Rusmevichientong, and J. B. Orlin. Adaptive data-driven inventory control with censored demand based on kaplan-meier estimator. *Operations Research*, 59(4):929–941, 2011.

A. J. King and S. W. Wallace. *Modeling with stochastic programming*. Springer Science & Business Media, 2012.

B. Knueven, D. Mildebrath, C. Muir, J. Siirola, D. Woodruff, and J.-P. Watson. mpi-sppy: Optimization under uncertainty for pyomo. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2020.

H. Lam and H. Qian. Assessing solution quality in stochastic optimization via bootstrap aggregating. In *2018 Winter Simulation Conference (WSC)*, pages 2061–2071. IEEE, 2018a.

H. Lam and H. Qian. Bounding optimality gap in stochastic optimization via bagging: Statistical efficiency and stability. *arXiv preprint arXiv:1810.02905*, 2018b.

S. Lee and S. Cho. Smoothed bagging with kernel bandwidth selectors. *Neural processing letters*, 14: 157–168, 2001.

Y. Li and Y.-G. Wang. Smooth bootstrap methods for analysis of longitudinal data. *Statistics in medicine*, 27(7):937–953, 2008.

J. Linderoth, A. Shapiro, and S. Wright. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research*, 142(1):215–241, 2006.

W.-K. Mak, D. P. Morton, and R. K. Wood. Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations research letters*, 24(1-2):47–56, 1999.

L. Mentch and G. Hooker. Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *The Journal of Machine Learning Research*, 17(1):841–881, 2016.

P. Parpas, B. Ustun, M. Webster, and Q. K. Tran. Importance sampling in stochastic programming: A markov chain monte carlo approach. *INFORMS Journal on Computing*, 27(2):358–377, 2015.

A. Prékopa. *Stochastic programming*, volume 324. Springer Science & Business Media, 2013.

Y. Raviv and N. Intrator. Bootstrapping with noise: An effective regularization technique. *Connection Science*, 8(3-4):355–372, 1996.

J. O. Royset and R. J.-B. Wets. Fusion of hard and soft information in nonparametric density estimation. *European Journal of Operational Research*, 247(2):532–547, 2015. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2015.06.034. URL https://www.sciencedirect.com/science/article/pii/S0377221715005482.

A. Ruszczyński and A. Shapiro. Stochastic programming models. *Handbooks in operations research and management science*, 10:1–64, 2003.

D. W. Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

J. Shao and D. Tu. *The jackknife and bootstrap*. Springer Science & Business Media, 2012.

A. Shapiro. Asymptotic analysis of stochastic programs. *Annals of Operations Research*, 30:169–186, 1991.

A. Shapiro. Statistical inference of multistage stochastic programming problems. *Math. Methods of Oper. Res*, 58:57–68, 2003.

B. Silverman and G. Young. The bootstrap: to smooth or not to smooth? *Biometrika*, 74(3):469–479, 1987.

H. Vaagen and S. W. Wallace. Product variety arising from hedging in the fashion supply chains. *International Journal of Production Economics*, 114(2):431–455, 2008.