An Adaptive, Distribution-Free Algorithm for the Newsvendor Problem with Censored Demands, with Applications to Inventory and Distribution

Gregory A. Godfrey
Warren B. Powell

Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544

December 15, 2000

Abstract

We consider the problem of optimizing inventories for problems where the demand distribution is unknown, and where it does not necessarily follow a standard form such as the normal. We address problems where the process of deciding the inventory, and then realizing the demand, occurs repeatedly. The only information we use is the amount of inventory left over. Rather than attempting to estimate the demand distribution, we directly estimate the value function using a technique called the Concave, Adaptive Value Estimation (CAVE) algorithm. CAVE constructs a sequence of concave piecewise linear approximations using sample gradients of the recourse function at different points in the domain. Since it is a sampling-based method, CAVE does not require knowledge of the underlying sample distribution. The result is a nonlinear approximation that is more responsive than traditional linear stochastic quasi-gradient methods and more flexible than analytical techniques that require distribution information. In addition, we demonstrate near-optimal behavior of the CAVE approximation in experiments involving two different types of stochastic programs — the newsvendor stochastic inventory problem, and two-stage distribution problems.

We consider the newsvendor problem which is solved repeatedly. At each iteration, we have to set the inventory which is then sold up to the available demand. We do not know the distribution of the demand and nor are we able to see the actual demand. As is typically the case, we observe only the amount of product that is sold.

The standard treatment of this version of the newsvendor problem involves first estimating the distribution of demand, and then solving for the optimal order point based on the well-known critical ratio (see Lee and Nahmias [11], Porteus [12] and Khouja [9] for thorough reviews of the inventory literature). A major practical limitation of this theory is the challenge of estimating actual demands when we often have access only to sales data. This is the well-known problem of handling censored demands. Braden and Freimer [3] present methods for estimating the parameters of a known distribution based on censored data. Lau and Lan [10] propose a method for estimating the demand distribution which combines estimates of the fractiles of the demand distribution with subjective estimates of the distribution. These estimates are then used to fit a very general demand distribution. Bell [1] suggests a method for adjusting the moments of the observed demand distribution to produce an improved estimate.

It is also possible to approach this problem from the perspective of the two-stage stochastic programming literature (see Sen and Higle [16] and Birge [2]). Classical techniques (see Kall and Wallace [8], Rockafellar and Wets [15], and Van Slyke and Wets [18]) require having at least a sample of the full outcome space, a requirement that would not be satisfied here. But, stochastic gradient techniques would be applicable, since they only require an estimate of the gradient after each observation (Ermoliev [4], Gupal and Bazhenov [7]). These methods tend to be unstable and encounter a scaling problem, since the units of the gradient are different than the units of the amount of product being ordered.

In this paper, we provide a near-optimal algorithm to the newsvendor problem with censored demands which is fast and stable. We do not make an effort to estimate the demand distribution directly, and as a result we do not need to make any assumptions about the shape of the demand curve. We assume that we are solving the problem iteratively, and we require only the information on the remaining inventory. Our technique, called the

CAVE algorithm (Concave, Adaptive Value Estimation), uses sample gradient information to adaptively estimate a (piecewise-linear) concave function which represents the expected profits as a function of the amount that was ordered. At each iteration, we simply find the maximum of this function and make that the order quantity for the next iteration. We do not enccounter the scaling problems that arise in stochastic gradient methods.

A side benefit of our approach, which is useful in some specialized applications, is that our estimation routine can be adjusted to produce piecewise-linear functions defined over discrete order quantities. In this case, the algorithm always returns integer solutions.

We also demonstrate the technique on a distribution problem that would arise when a plant is distributing product to various warehouses which serve nonoverlapping regions. In this case, the value of product at one warehouse is independent of the amount of product shipped to another warehouse (implying a separable value function). We demonstrate the algorithm for this special case in this paper only because we can also solve this problem to optimality, allowing us to estimate the quality of the solution.

The technique is not limited to these special cases. Godfrey and Powell [6] apply the method to a multistage distribution problem that arises in the management of homogeneous fleets of vehicles over space and time. In this setting, we are using the CAVE algorithm to produce separable approximations of a nonseparable value function. Topaloglu and Powell [17] apply the technique to stochastic multicommodity flow problems, which would also arise in multiproduct problems with product substitution. For both of these problem classes it is not possible to find even good bounds for stochastic versions of the problem. However, these papers show that the algorithm provides near-optimal solutions when applied to deterministic instances (which can also be solved using a commercial linear programming package), and significantly outperform classical rolling horizon procedures which depend on deterministic estimates of the future.

The outline of this paper is as follows. Section 1 describes the CAVE algorithm and gives a proof that the updating operations preserve concavity. Section 2 demonstrates the method in the context of the newsvendor problem, and compares the solution quality to those that

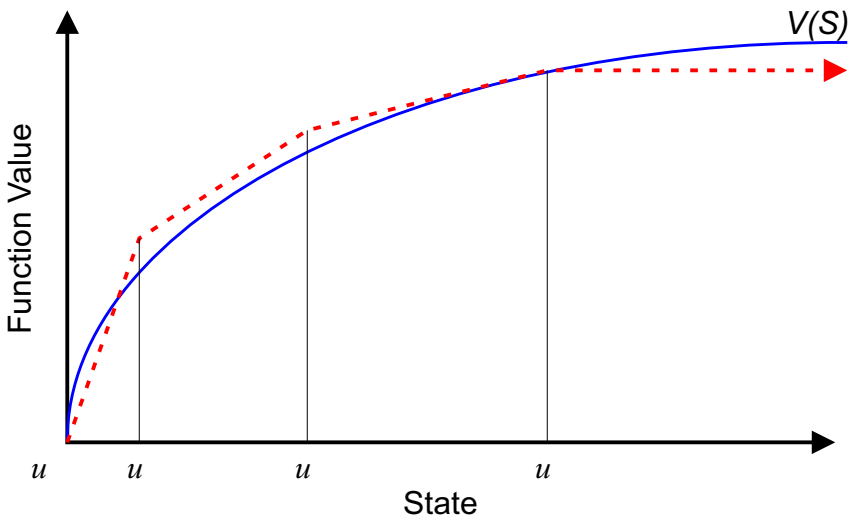

Figure 1: Piecewise linear approximation $\hat{V}(s)$ defined by the collection $\{(\nu^k, u^k) : k \in \mathcal{K}\}$ where $\mathcal{K} = \{0, 1, 2, \ldots, k_{\max}\}$

would have been obtained if we did in fact know the demand distribution. Finally, sections 3 and 4 describe the two-stage stochastic distribution program with network recourse and present experimental results, respectively. We compare the CAVE approximation of the second-stage recourse function against the exact expected recourse function obtained using the nodal recourse method of Powell and Frantzeskakis [5] and against a traditional stochastic linearization method.

1 CAVE Algorithm

The Concave, Adaptive Value Estimation (CAVE) algorithm was created to estimate recourse functions, but we present it in its most general form of simple functional estimation. Consider a continuous, concave function $V(s)$ defined on $\mathbb{R}_+$ with $V(0) \equiv 0$. We present an original method for constructing a concave, piecewise-linear approximation of $V(s)$ using repeated stochastic gradient samples of the function at different domain values. In addition, we prove that the rules used to update the approximation preserve concavity.

Figure 1 serves as a visual guide to the notation. The piecewise-linear approximation of $V(s)$, which we denote $\hat{V}(s)$, is defined by a finite set of ordered breakpoints $\{(\nu^k, u^k) : k \in \mathcal{K}\}$ where $\mathcal{K} = \{0, 1, 2, \ldots, k_{\max}\}$. Each breakpoint defines a linear segment with ν^k as the

slope of the segment projected from u^k where $u^0 \equiv 0$. The breakpoints are ordered such that $0 \equiv u^0 < u^1 < u^2 < \ldots < u^{k_{\max}} < \infty$ and $\nu^0 > \nu^1 > \ldots > \nu^{k_{\max}}$. Also, we define $\nu(s)$ to be the slope of $\hat{V}$ at s. It follows from the monotonically decreasing slopes that $\hat{V}(s)$ is concave.

We define notation describing the gradient estimates used to update the approximation. Let $\pi^-(s_n)$ and $\pi^+(s_n)$ be random variables representing the left and right gradients at $V(s_n)$, respectively. Individual sample gradients are denoted $\pi^-(s_n, \omega)$ and $\pi^+(s_n, \omega)$. We assume that the gradients have finite variance and proper expectations, that is,

$$E\left[\pi^-(s_n)\right] = \left.\frac{\partial V(s)}{\partial s^-}\right|_{s=s_n} \quad (1)$$

$$E\left[\pi^+(s_n)\right] = \left.\frac{\partial V(s)}{\partial s^+}\right|_{s=s_n} \quad (2)$$

CAVE requires only finite variance. However, in order to have desirable limiting properties, the proper expectations are necessary, just as with other sampling-based techniques.

In addition, we assume that the probability measure of the outcome space Ω is such that $P\{\pi^-(s,\omega) \geq \pi^+(s,\omega)\} = 1$ for all $s \geq 0$. That is, the left sample gradient is greater than or equal to the right sample gradient, almost surely. To reduce notational clutter, we drop ω and refer directly to the random variables $\pi^-(s)$ and $\pi^+(s)$ where practical.

These gradient properties are not unduly restrictive. If at some state s the left and right derivatives of $V(s)$ are equal, then so are the left and right gradients $\pi^-(s)$ and $\pi^+(s)$, with their common value depending on the random sample. On the other hand, if the left and right derivatives are not equal, such as with $V(s) = \{s \text{ for } 0 \leq s \leq 1; 1 \text{ otherwise}\}$ at $s = 1$, then concavity requires that the left derivative be strictly greater than the right derivative. Accordingly, the gradients must satisfy $\pi^-(s) \geq \pi^+(s)$. In the experiments that follow, we establish only that $\pi^-(s) \geq \pi^+(s)$ for all sample gradients (and that they are finite), but not that the gradients have proper expectations, since that is not a necessary property for CAVE.

The steps of the CAVE algorithm for updating the piecewise-linear approximation $\hat{V}(s)$ appear in Figure 2. The basic premise of the update is that $\pi^-(s)$ is smoothed into the approximation slopes to the left of s to some minimal extent, say to the slopes from $s-1$ to

STEP 1 Initialization:

- Let $\mathcal{K} = \{0\}$, where $\nu^0 = 0, u^0 = 0$.
- Initialize parameters $\varepsilon^-, \varepsilon^+, \alpha$.

STEP 2 Collect Gradient Information:

- Given a state $s \geq 0$, sample the gradients $\pi^-(s, \psi)$ and $\pi^+(s, \psi)$ with random outcome $\psi \in \Psi$.

STEP 3 Define Smoothing Interval:

- Let $k^- = \min\{k \in \mathcal{K} : \nu^k \leq (1-\alpha)\nu^{k+1} + \alpha\pi^-(s)\}$ and $k^+ = \max\{k \in \mathcal{K} : (1-\alpha)\nu^{k-1} + \alpha\pi^+(s) \leq \nu^k\}$.
- Define the smoothing interval $\mathcal{Q} = \left[\min\{s - \varepsilon^-, u^{k^-}\}, \max\{s + \varepsilon^+, u^{k^+ + 1}\}\right)$. If $u^{k^+ + 1}$ is undefined, then set $u^{k^+ + 1} = \infty$.
- Create new breakpoints at s and the endpoints of $\mathcal{Q}$ as needed. Since a new breakpoint always divides an existing segment, the segment slopes on both sides of the new breakpoint are the same initially.

STEP 4 Perform Smoothing:

- For each segment in the interval $\mathcal{Q}$, update the slope according to $\nu^k_{new} = \alpha\pi + (1-\alpha)\nu^k_{old}$ where $\pi = \pi^-(s)$ if $u^k < s$ and $\pi = \pi^+(s)$ otherwise.
- Adjust $\varepsilon^-, \varepsilon^+, \alpha$ according to step size rules.
- Return to Step 2.

Figure 2: The Concave, Adaptive Value Estimation (CAVE) Algorithm

s. If the new approximation is not concave, then continue smoothing $\pi^-(s)$ into the slopes further to the left until concavity is restored, say from $s-3$ to s. The same procedure holds for smoothing $\pi^+(s)$ into the slopes to the right of s.

Three parameters control the amount of change in $\hat{V}(s)$ at each iteration. The parameters $\varepsilon^-, \varepsilon^+$ are defined such that $[s - \varepsilon^-, s + \varepsilon^+)$ is the smallest interval over which the slopes are updated. Let α be the smoothing parameter applied to the segment slopes. At each iteration, we can apply declining step size rules to $\varepsilon^-, \varepsilon^+$, and α for stability.

We illustrate the algorithm with two examples. For simplicity, we assume that $\alpha = 1$ in both examples. In Figure 3a, there are four segments and $\nu(s) > \pi^-(s) > \pi^+(s)$. Here, $\pi^-(s)$ is greater than or equal to both ν^2 and ν^3, so $k^- = 2$. Since $\pi^+(s)$ is greater than only ν^3, then $k^+ = 3$. This means that we stretch the interval $[s-\varepsilon^-, s+\varepsilon^+)$ to include $u^{k^+} = u^3$ on the right side to preserve concavity, leading to the smoothing interval $\mathcal{Q} = [s - \varepsilon^-, u^3)$. New breakpoints are inserted at $s-\varepsilon^-$ and s (both between u^1 and u^2) with slopes initialized

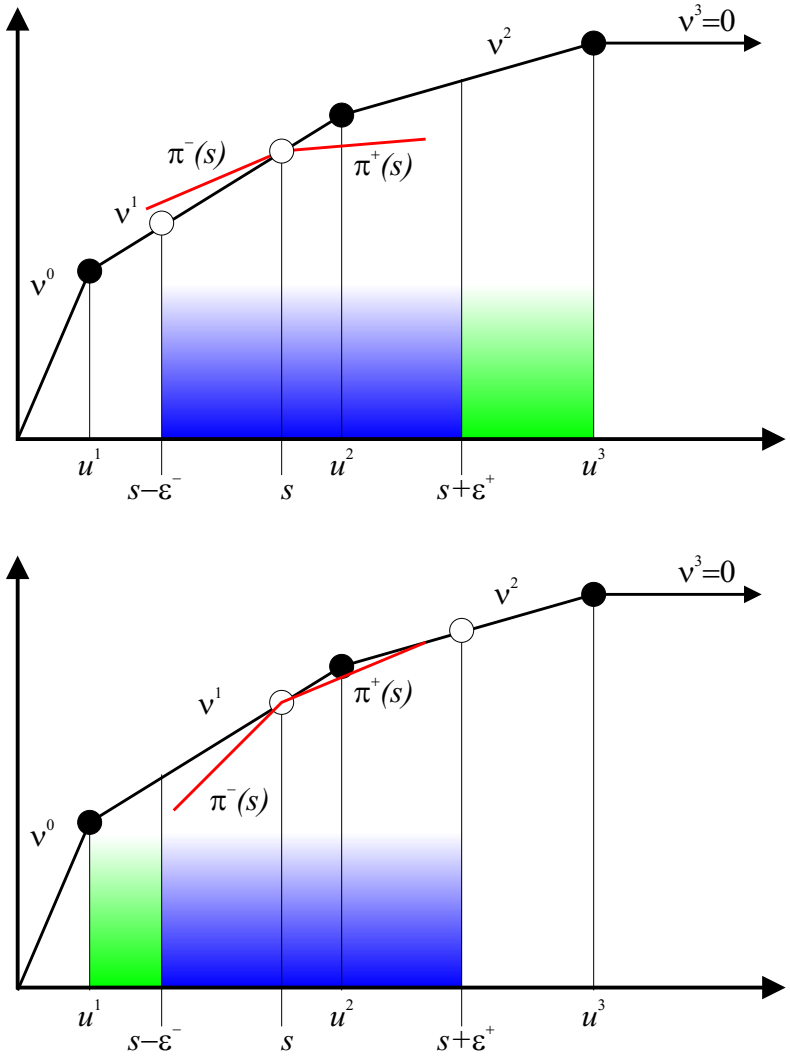

Figure 3: Two Examples of Smoothing Intervals for the CAVE Update (a and b)

to ν^1. In this example, smoothing decreases all of the slopes in $\mathcal{Q}$.

The example in Figure 3b is similar, except that $\pi^-(s) > \nu(s) > \pi^+(s)$. This time, $k^- = 1$ and $k^+ = 2$. Stretching the interval $[s - \varepsilon^-, s + \varepsilon^+)$ to include $u^{k^-} = u^1$ on the left side leads to the smoothing interval $\mathcal{Q} = [u^1, s + \varepsilon^+)$. New breakpoints are inserted at s and $s + \varepsilon^+$ with initial slopes ν^1 and ν^2, respectively. In this example, smoothing increases the slopes over $[u^1, s) \cup [u^2, s + \varepsilon^+)$ and decreases the slopes over $[s, u^2)$.

An important property of the CAVE algorithm is that concavity is preserved under the updating operations (Theorem 1.1). The proof has two parts—the first part proves concavity over the smoothing interval and the second part proves concavity at the endpoints of the

interval.

Theorem 1.1 *Consider a concave function defined by the breakpoints* $\{(\nu^k, u^k) : k \in \mathcal{K}\}$. *Using the updating algorithm described in Figure 2, we may rewrite the smoothing interval* $\mathcal{Q} = \left[\min\{s - \varepsilon^-, u^{k^-}\}, \max\{s + \varepsilon^+, u^{k^+ + 1}\}\right)$ *as* $[u^m, u^n)$ *where* $m < n$ *(otherwise,* $\mathcal{Q}$ *is empty). Concavity is preserved under the smoothing operation performed in Step 4 of the algorithm.*

Proof: The proof has two parts. First, we show concavity is preserved within the smoothing interval $\mathcal{Q}$. Second, we show that concavity is preserved at the two ends where the smoothed interval connects to the original function.

To show concavity is preserved within $\mathcal{Q}$, we apply the smoothing equation $\nu^k_{new} = \alpha\pi + (1-\alpha)\nu^k_{old}$ where $\pi = \pi^-(s)$ for $k = m, m+1, \ldots, q$ and $\pi = \pi^+(s)$ for $k = q+1, \ldots, n-1$ for an appropriate q. We then show that the new slope values decrease monotonically in k.

$$\begin{aligned}
\nu^m &> \ldots > \nu^q > \nu^{q+1} > \ldots > \nu^{n-1} && (3)\\
(1-\alpha)\nu^m &> \ldots > (1-\alpha)\nu^q > (1-\alpha)\nu^{q+1} > \ldots > (1-\alpha)\nu^{n-1} && (4)\\
\alpha\pi^+(s) + (1-\alpha)\nu^m &> \ldots > \alpha\pi^+(s) + (1-\alpha)\nu^q > \\
&\quad \alpha\pi^+(s) + (1-\alpha)\nu^{q+1} > \ldots > \alpha\pi^+(s) + (1-\alpha)\nu^{n-1} && (5)\\
\alpha\pi^-(s) + (1-\alpha)\nu^m &> \ldots > \alpha\pi^-(s) + (1-\alpha)\nu^q > \\
&\quad \alpha\pi^+(s) + (1-\alpha)\nu^{q+1} > \ldots > \alpha\pi^+(s) + (1-\alpha)\nu^{n-1} && (6)\\
\nu^m_{new} &> \ldots > \nu^q_{new} > \nu^{q+1}_{new} > \ldots > \nu^{n-1}_{new} && (7)
\end{aligned}$$

The first set of inequalities holds by concavity of the original function. The second set holds by multiplication by a positive constant $(1-\alpha)$. The third set holds by addition of the constant $\alpha\pi^+(s)$. In the fourth set, we use the fact that $\pi^-(s) \geq \pi^+(s)$ to add a positive constant $\pi^-(s) - \pi^+(s)$ to components m through q. The last set shows that the new slope values have the desired monotonicity.

In the second part of the proof, we need to show that concavity is preserved at the endpoints. That is, we need to show that

$$\nu^{m-1} > \nu^m_{new} = (1-\alpha)\nu^m + \alpha\pi^-(s) \qquad (8)$$

$$(1-\alpha)\nu^{n-1} + \alpha\pi^+(s) = \nu^{n-1}_{new} > \nu^n \tag{9}$$

To prove the first part (the left endpoint), given the definition $k^- = \min\{k \in \mathcal{K} : \nu^k \leq (1-\alpha)\nu^{k+1} + \alpha\pi^-(s)\}$, then

$$\nu^k > (1-\alpha)\nu^{k+1} + \alpha\pi^-(s) \text{ for all } k < k^-.$$

Adjusting the indices, we have

$$\nu^{k-1} > (1-\alpha)\nu^k + \alpha\pi^-(s) \text{ for all } k \leq k^-. \tag{10}$$

Since m was defined such that $u^m = \min\{s - \varepsilon^-, u^{k^-}\}$, then $m \leq k^-$. Applying this result to equation (10) proves equation (8).

The proof for the right endpoint is similar. Given the definition $k^+ = \max\{k \in \mathcal{K} : (1-\alpha)\nu^{k-1} + \alpha\pi^+(s) \leq \nu^k\}$, then

$$(1-\alpha)\nu^{k-1} + \alpha\pi^+(s) > \nu^k \text{ for all } k > k^+. \tag{11}$$

Since n was defined such that $u^n = \max\{s + \varepsilon^+, u^{k^+ + 1}\}$, then $n \geq k^+ + 1 > k^+$. Applying this result to equation (11) proves equation (9).

With the CAVE algorithm in place, we test its effectiveness on three problems, in order of increasing complexity. The first problem is to approximate the function $\ln(1+s)$ under random perturbations of the exact gradient values at random points on the domain. The second problem uses CAVE to estimate the optimal order quantity for a simple newsvendor stochastic inventory problem by approximating the expected profit function. The final problem uses CAVE to approximate the second-stage expected recourse function in a two-stage stochastic program with network recourse.

2 CAVE Applied to the Newsvendor Model

The newsvendor model provides a simple instance in which the CAVE algorithm may be used to estimate a function that is itself embedded in a larger optimization problem. We

begin with a brief description of the newsvendor model. Further details and citations may be found in Lee and Nahmias [11] and Porteus [12].

The newsvendor model applies to single-period, stochastic demand problems where perishable goods such as newspaper and fresh produce are ordered. The difficulty is that the vendor may place only a single order for a quantity Q (at unit cost c) of the good, subject to a random (continuous) demand D during the next period. The goods are sold at a unit price $p > c$. If at the end of the period any inventory remains, then it is dumped at a salvage unit price of $h < c$. We assume that the penalty cost per unit on unsatisfied demand is simply the lost sales price p.

For a particular demand sample, $D(\omega)$ with $\omega \in \Omega$, and order quantity Q, we may write the total profit as

$$V(Q, D(\omega)) = p \min\{Q, D(\omega)\} + h \max\{Q - D(\omega), 0\} - cQ. \tag{12}$$

Or, in terms of only Q and the truncated demand $\min\{Q, D(\omega)\}$,

$$V(Q, D(\omega)) = (p - h) \min\{Q, D(\omega)\} - (c - h)Q. \tag{13}$$

Given the cumulative demand distribution function $F(t)$, we may compute the expected profit for a particular Q as $\bar{V}(Q) = E_\omega V(Q, D(\omega))$. Leaving the details, the function $\bar{V}(Q)$ is concave in Q and its maximizing value of Q, call it Q^*, may be found by setting the derivative to zero and solving to get

$$F(Q^*) = P[D \leq Q^*] = \frac{p - c}{(p - c) + (c - h)} = \frac{p - c}{p - h}. \tag{14}$$

This quantity $F(Q^*)$ is also known as the *critical ratio*. Thus, an optimal order quantity Q^* exists, but in practice, there are two problems computing it accurately. First, it requires knowing the demand distribution in advance. Second, estimating proper parameters for D is difficult because we observe only the truncated demand $\min\{Q, D(\omega)\}$, not the actual demand $D(\omega)$. However, we show how the adaptive qualities of the CAVE algorithm estimate Q^* without *a priori* knowledge of the demand distribution and instead using only the observed demands.

The main idea is to use CAVE to construct a piecewise-linear approximation of $\bar{V}(Q)$, call it $\hat{V}(Q)$. The estimate of Q^*, call it $\hat{Q}^*$, is simply the order quantity that maximizes $\hat{V}(Q)$. After placing an order for $\hat{Q}^*$, we observe the truncated demand $\min\{\hat{Q}^*, D(\omega)\}$ for a particular sample demand, compute left and right gradients and update the CAVE approximation $\hat{V}(Q)$.

In order to update the approximation, consider the total profit when $\hat{Q}^* \leq D(\omega)$

$$V(\hat{Q}^*, D(\omega)) = (p-h)\hat{Q}^* - (c-h)\hat{Q}^* = (p-c)\hat{Q}^*. \tag{15}$$

In this case, we use $\hat{Q}^*$ for the state and $\pi^-(\hat{Q}^*) = \pi^+(\hat{Q}^*) = p-c$ for the gradient estimates. This assumes that if one additional unit had been ordered, then it would have been sold (which we do not know for sure). On the other hand, if $\hat{Q}^* > D(\omega)$, then the total profit is

$$V(\hat{Q}^*, D(\omega)) = (p-h)D(\omega) - (c-h)\hat{Q}^*. \tag{16}$$

In this case, it is much more effective if we measure the gradient assuming that the order quantity had been $D(\omega)$ rather than the larger $\hat{Q}^*$. As a result, we use $D(\omega)$ for the state and $\pi^-(D(\omega)) = (p-h) - (c-h) = p-c$ and $\pi^+(D(\omega)) = h-c$ for the gradient estimates. Note that in either case, the gradients are finite and that $\pi^- \geq \pi^+$. After updating $\hat{V}(Q)$, we determine the new order quantity $\hat{Q}^*$ and continue to the next iteration.

In the experiments that follow, we use three different distributions (Gaussian, Poisson and Uniform) and two different critical ratios ($F(Q^*) = 0.333, 0.667$). The relevant distribution parameters, costs and optimal order quantities appear in Table 1.

Distribution	Gaussian		Poisson		Uniform	
Parameters	$\mu = 20, \sigma = 5$		$\lambda = 20$		$[10, 30]$	
Unit cost c	\$150	\$100	\$150	\$100	\$150	\$100
Sale price p	\$200	\$200	\$200	\$200	\$200	\$200
Salvage price h	\$50	\$50	\$50	\$50	\$50	\$50
$F(Q^*) = \frac{p-c}{p-h}$	0.333	0.667	0.333	0.667	0.333	0.667
Q^*	17.9	22.1	18	22	16.7	23.3

Table 1: Summary of Demand and Cost Parameters for Newsvendor Experiments

For each of the experiments, we use an initial approximation $\hat{V}(Q) = (h-c)Q$, which has a negative slope. The epsilon values $\varepsilon^- = \varepsilon^+$ start at 4, and then are cut in half after

ten and twenty iterations. We set the smoothing parameter to be $\alpha_n = 5/(5+n)$ at the n^{th} iteration for all segments.

The experimental procedure is as follows. Before sampling the random demand $D(\omega)$, we choose the order quantity to be the value Q that maximizes the current CAVE profit approximation $\hat{V}(Q)$. Using the observed demand $\min\{Q, D(\omega)\}$, we compute the total profit given that sample, including any salvage value of excess stock. In addition, we estimate gradients, update the CAVE approximation, and continue the process for a new order quantity and observed demand sample.

As an "ideal" comparison, we use the *actual* demand $D(\omega)$ at each sample to estimate the parameters $\hat{\mu}$ and $\hat{\sigma}$ of a Gaussian distribution. Given the critical ratio $F(Q^*)$ and the current estimates of $\hat{\mu}$ and $\hat{\sigma}$, we can derive an order quantity to compare against each demand sample. We call this Gaussian order quantity "ideal" because it relies on perfect information. That is, the Gaussian order quantity relies on knowing the actual demand $D(\omega)$ instead of the more realistic observed demand $\min\{Q, D(\omega)\}$ used by the CAVE approximation.

Our metric for comparing these two approaches is to track the total profit given the same sequence of demand samples. To get a sense of the relative quality of the CAVE and Gaussian profits, we also compute the total profits obtained using the optimal order quantity Q^* and its neighboring values $Q^* - 1$ and $Q^* + 1$ for all demand samples. The profit at the neighboring values reveals the sensitivity of the profits to the order quantity.

To allow the CAVE and Gaussian approximations to "warm up", we process the first 50 demand samples without tracking the total profit. We then apply all five order quantity strategies to demand samples 51-1000 and track the total profit. The total profit for each strategy (averaged over ten different simulations and expressed as a deviation from the optimal Q^* profit) appears in Table 2.

In all cases, the profits using the CAVE and Gaussian order quantities are higher than using $Q^* \pm 1$. Not surprisingly, the Gaussian perfect information strategy excels when the demands are Gaussian or Poisson (large λ). However, even with perfect information, the Gaussian approach struggles when the underlying demand is uniform. In the uniform case,

the CAVE approach (which is distribution-independent) has the higher profit despite observing the truncated demand process.

In conclusion, the CAVE approach compares quite well against idealized benchmarks even while constrained by realistic assumptions of no demand distribution information and observation of the truncated (not actual) demands.

	Gaussian		Poisson		Uniform	
Critical Ratio $F(Q^*)$	0.333	0.667	0.333	0.667	0.333	0.667
$Q^* - 1$	0.77%	0.31%	0.75%	0.24%	0.74%	0.25%
Q^*	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
$Q^* + 1$	0.73%	0.34%	0.88%	0.41%	0.37%	0.20%
CAVE approximation	0.28%	0.09%	0.40%	0.21%	0.25%	0.15%
Gaussian/Perfect Infomation	0.08%	0.02%	0.19%	0.07%	0.35%	0.19%

Table 2: Summary of Total Profits of Newsvendor experiments (expressed as the deviation from the optimal Q^* order quantity profits)

3 CAVE Applied to a Two-Stage Stochastic Program

Having applied the CAVE technique to approximate a simple concave function, we proceed to the more complex two-stage stochastic program with recourse. In this two-stage model, decisions made in the deterministic first stage (time 0) have downstream consequences that are captured in the uncertain second stage (time 1). The goal is to optimize the first-stage costs plus the expected second-stage costs.

For the motivating example, we formulate a dynamic network with random arc capacities as a two-stage stochastic program with network recourse. We choose this network recourse problem only because it can be solved optimally (see Powell and Cheung [14]), thus providing a benchmark for evaluating the CAVE approximation results. The CAVE approach can be applied to a more general class of problems, but it becomes more difficult to find suitable benchmarks.

Figure 4 illustrates the dynamic network with random arc capacities. Each first-stage

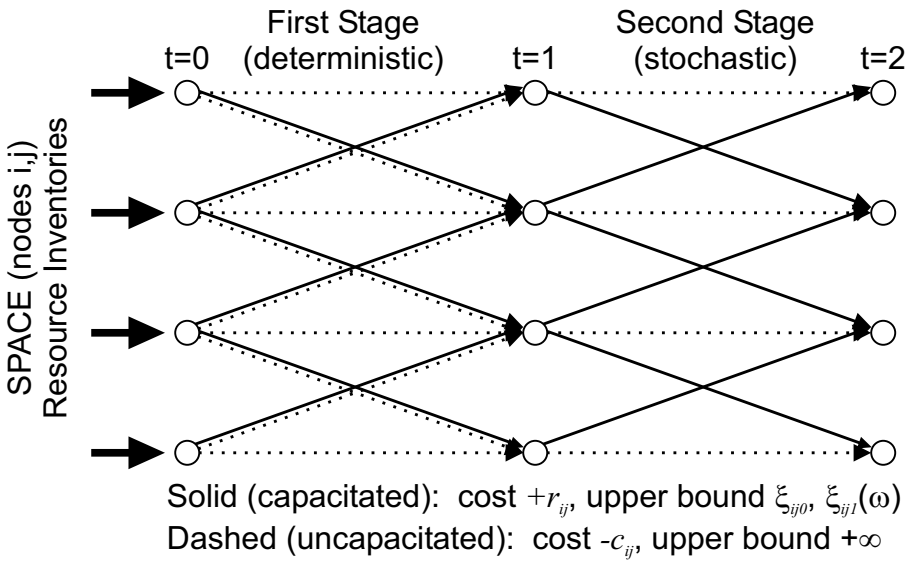

Figure 4: Illustration of two-stage dynamic network with random arc capacities

node $i \in \mathcal{I}$ has an initial supply S_{i0}. Each supply unit at node $i \in \mathcal{I}$ flows to a second-stage node $j \in \mathcal{I}$ in one of two ways. The first option is to flow over a capacitated (revenue) arc with cost r_{ij} and (deterministic) upper bound ξ_{ij0}. The second option is to flow over an uncapacitated arc incurring cost $-c_{ij}$ (with $c_{ii} \equiv 0$). We denote the total flow between nodes i and j via these two options by x_{ij0} and y_{ij0}, respectively.

The second-stage problem is similar, except that the upper bounds on the revenue arcs ξ_{ij1} are random variables defined on the probability space $(\Omega, \mathcal{H}, P)$ with elementary outcomes $\omega \in \Omega$. We also assume that the probability measure P is discrete and known. The realization of these random upper bounds $\xi_{ij1}(\omega)$ come *after* the first-stage decisions are made. Given a second-stage node $i \in \mathcal{I}$ with supply S_{i1}, each supply unit flows through either a revenue arc with cost r_{ij} or an uncapacitated arc with zero cost. We represent these two flows, for a particular sample $\omega \in \Omega$, by $x_{ij1}(\omega)$ and $y_{ij1}(\omega)$, respectively.

The first-stage decisions may be written as

$$\max_{x_0, y_0, S_1} \quad rx_0 - cy_0 + EV_1(S_1, \xi_1) \tag{17}$$

$$\text{subject to:} \quad \sum_{j \in \mathcal{I}} (x_{ij0} + y_{ij0}) = S_{i0} \quad \forall i \in \mathcal{I} \tag{18}$$

$$\sum_{i \in \mathcal{I}} (x_{ij0} + y_{ij0}) = S_{j1} \quad \forall j \in \mathcal{I} \tag{19}$$

$$0 \leq x_{ij0} \leq \xi_{ij0} \quad \forall i,j \in \mathcal{I} \tag{20}$$

$$y_{ij0} \geq 0 \quad \forall i,j \in \mathcal{I} \tag{21}$$

where the first stage upper bounds ξ_{ij0} are deterministic. The last part of the objective function (17) is the expected future cost, called the expected recourse function. The first-stage decisions are the two flow types $\mathbf{x}_0, \mathbf{y}_0$ and the second-stage supplies $\mathbf{S}_1$. Given the second-stage supplies $\mathbf{S}_1$ and a realization $\xi_{\mathbf{1}}(\omega)$, we may solve the second stage problem:

$$V_1(\mathbf{S}_1, \xi_1(\omega)) = \max_{x_1(\omega), y_1(\omega)} \quad rx_1(\omega) - cy_1(\omega) \tag{22}$$

$$\text{subject to:} \quad \sum_{j \in \mathcal{I}} (x_{ij1}(\omega) + y_{ij1}(\omega)) = S_{i1} \quad \forall i \in \mathcal{I} \tag{23}$$

$$0 \leq x_{ij1}(\omega) \leq \xi_{ij1}(\omega) \quad \forall i,j \in \mathcal{I} \tag{24}$$

$$y_{ij1}(\omega) \geq 0 \quad \forall i,j \in \mathcal{I} \tag{25}$$

where the decisions $x_1(\omega), y_1(\omega)$ depend on the particular realization $\omega \in \Omega$. Since each value of V_1 requires solving a second-stage optimization problem, the expected recourse function of a general two-stage stochastic program is computationally tractable for only small problems (Wallace [19]). However, for the special case where the second-stage problem has a tree structure, as this problem has, the expected recourse function can be calculated exactly given the probability measure P.

Before proceeding, we point out a few properties of the expected recourse function. First, the tree structure in the second stage leads to a decomposition of the recourse problem by node, making the recourse function separable. That is,

$$V_1(S_1, \xi_1(\omega)) = \sum_{i \in \mathcal{I}} V_{i1}(S_{i1}, \xi_{\mathbf{1}}(\omega)). \tag{26}$$

Second, each of these node recourse functions V_{i1} is a concave (because we are maximizing) function of S_{i1} (see Powell and Frantzeskakis [5], Section 2.3, for one such proof). Consequently, V_1 is also concave. Finally, we define the expected recourse function

$$\bar{V}_1(S_1) \equiv EV_1(S_1, \xi_1) \tag{27}$$

In the experiments, we estimate $\bar{V}_1(S_1)$ in two ways—directly using an exact approach and iteratively using CAVE.

Nodal Recourse Approach

For the class of problems in which the second-stage problem has a tree structure, exact methods exist to compute the recourse function. In particular, the nodal recourse method of Powell [13] and Powell and Frantzeskakis [5], which later evolved into the general tree recourse method of Powell and Cheung [14], can find the exact expected recourse function. However, the nodal recourse approach requires full distribution information, while CAVE does not.

The nodal recourse method works one node at a time. Starting at a second-stage location i, we rank each of N possible options for a supply unit at i by profit $\bar{c}_n$ (highest to lowest). In our example, supply may flow on revenue arcs to each destination node $j \in \mathcal{I}, j \neq i$ with profit r_{ij} or "do nothing" at zero cost.

Next, we define the routing probabilities, $\phi(s,n)$:

$$\phi(s,n) = P\{\text{the } s^{th} \text{ unit of flow takes the } n^{th} \text{ option}\} \tag{28}$$

Given the routing probability, we can easily compute the expected marginal profit for the s^{th} unit of flow, $G(s)$:

$$G(s) = \sum_{n=1}^{N} \bar{c}_n \cdot \phi(s,n) \tag{29}$$

The sum of the expected marginal profits gives us the expected recourse function $\bar{V}_{i1}(S_{i1})$:

$$\bar{V}_{i1}(S_{i1}) = \sum_{s=1}^{S_{i1}} G(s) = \sum_{s=1}^{S_{i1}} \sum_{n=1}^{N} \bar{c}_n \cdot \phi(s,n) \tag{30}$$

Thus, the routing probabilities are the key to computing the exact expected recourse function. The details on how to compute these routing probabilities appear independently in Powell [13], Powell and Frantzeskakis [5], and Powell and Cheung [14]. However, nodal recourse requires the second-stage random upper bounds to be independent, but CAVE does not.

CAVE Approach

The CAVE algorithm can be used to approximate the expected recourse function at each second-stage terminal. Instead of computing the function directly, we solve a series

of two-stage problems, each with a different sample of the second-stage capacities to build and smooth the CAVE approximation. Although it is inexact, the sampling-based CAVE approach is effective on a broader class of problems than approaches based on functional expectations. In particular, the CAVE approach can be applied to problems in which the underlying demand distribution is unknown or in which capacities depend upon decisions in prior stages.

There are three values needed to construct the CAVE approximation. For each second-stage node i, we need the supply S_{i1}, the left dual estimate π_{i1}^-, and the right dual estimate π_{i1}^+. For a particular outcome $\omega \in \Omega$, the second-stage supply S_{i1} follows from solving the first-stage problem using the current approximation of the recourse function $\sum_{i \in \mathcal{I}} \hat{V}_{i1}(S_{i1})$ in place of the expectation.

The dual values are easy to calculate. Given the second-stage realization of the upper bounds, solve the second-stage problem in equations (22)-(25) by sorting the outbound arcs from highest profit to lowest and assigning flow accordingly. The dual values follow directly from the sort. The right dual value $\pi_{i1}^+(\omega)$ is the value of having one additional unit of supply at i. The left dual value $\pi_{i1}^-(\omega)$ is the value of taking one unit of supply away from i. If there is no supply at i, then $\pi_{i1}^-(\omega)$ is undefined, but that is not a problem since CAVE does not smooth to the left when the flow is zero. If the demand revenues and repositioning costs are finite, then so are the dual values. Furthermore, sorting the outbound arcs by profit guarantees that $\pi_{i1}^-(\omega) \geq \pi_{i1}^+(\omega)$ almost surely.

Stochastic Quasi-Gradient Approach

Although the CAVE approach may be effective over a broader class of problems than the exact methods, it does not show that CAVE is more effective (or efficient) than other sampling-based techniques such as classical stochastic quasi-gradient methods. For this reason, we describe a third approach for solving this problem that uses stochastic linearization.

The approach is similar to CAVE except a strictly linear approximation of the recourse function is constructed using the right dual estimate π_{i1}^+. Unfortunately, the linear approximation is potentially unstable with respect to the first-stage decisions. To add stability,

we smooth the first-stage decisions by taking a weighted average of the first-stage flow in the current iteration and the smoothed flow from the previous iteration. In iteration n, the weight on the previous smoothed flow is $(n-1)/n$ and the weight on the current flow is $1/n$.

Since the problem is discrete, these smoothed first-stage flows must be rounded before execution. However, this rounding must be done carefully. For example, consider a first-stage node with a supply of two resources and arcs to nodes A, B and C. The first iteration decision assigns one unit each to A and B. The second iteration decision assigns one unit each to B and C. After averaging, the adjusted assignments are 1/2 units to A, 1 unit to B, and 1/2 units to C. How should we assign the two units of supply? Suppose we break the rounding tie between A and C somehow. The third iteration decision assigns one unit each to A and C. The smoothed assignments are now 2/3 units to all three destinations, yet any reasonable rounding rule would send a full unit to all three locations.

After running the experiments repeatedly using different rounding schemes, we found a "reasonable" rule that outperformed the other rounding rules while keeping the rounded flow equal to the supply. Additional tweaking did not change the results significantly. This rounding rule assigns flow unit-by-unit. Start by assigning available supply to the integer portion of each smoothed flow. For example, if the smoothed flow vector was $(2.8, 1.6, 2.6)$ with total supply of seven, then immediately assign $(2, 1, 2)$. The other two units are allocated using the highest unsatisfied flow as first tie-breaker and highest arc revenue as the second tie-breaker. If the arc revenue vector is $(\$50, -\$10, \$20)$, then the final rounded flow would be $(3, 1, 3)$.

Having described how to apply each of these procedures to solve the two-stage stochastic program, we conduct a series of experiments to compare their results in the next section.

4 Experimental Results for Two-Stage Problem

To compare the different approaches, we construct a problem with 40 nodes scattered uniformly across a geographic space with a distance d_{ij} between each pair of nodes. On average,

the cumulative upper bound on revenue arcs at each stage is 250. The revenue on demand arcs connecting nodes i and $j \neq i$ is $\$0.70 \cdot d_{ij}$, and the cost on the uncapacitated arcs is $\$1.00 \cdot d_{ij}$.

We assume that the upper bound distribution between pairs of terminals $i \neq j$ is Poisson with parameter λ_{ij}. The weights are such that λ_{ij} is proportional to $\beta_i(1-\beta_j)$ where β_i is a uniform random variable between zero and one associated with terminal i. This means that the expected capacity into a node is negatively correlated with the expected capacity out of a node. This emphasizes the importance of robust first-stage decisions since the first-stage capacities would tend to focus flow into second-stage nodes with limited outbound capacity. The 200 initial supply units $\mathbf{S}_0$ are also distributed according to the inbound weights.

From the set of outcomes Ω, we draw one first-stage sample to represent known information and 400 independent, random second-stage samples. All three algorithms are tested on the same sequence of 400 samples to reduce the effect of sample-to-sample error.

At each second-stage node i, the exact expected recourse function $\bar{V}_{i1}(S_{i1})$ is computed using the nodal recourse method based on the sorted revenues and Poisson parameters λ_{ij}. The expected objective function of the two-stage problem given the initial state $\mathbf{S}_0$ is \$162,042. All costs used in this analysis are scaled by this value in order to normalize the results.

In the first set of experiments, we use the CAVE and stochastic quasi-gradient algorithms to approximate the recourse function. In each of the 4,000 iterations, we solve the first-stage problem using the approximation, draw a random sample from our set of 400 second-stage outcomes, and then update the approximations. For all of the CAVE runs, we set $\varepsilon^- = \varepsilon^+ = 1.0$ for all iterations and $\alpha_n = 2/(2+n)$ where n is the iteration number starting with $n = 0$. The initial approximation of the recourse function at each node is $\nu^0 = 0, u^0 = 0$. The stochastic linearization uses the same initial approximation, but the scalar slope is smoothed using $\alpha_n = 1/(1+n)$.

Figure 5 compares the objective function using the CAVE and stochastic quasi-gradient approximations with the exact recourse function on each of the 400 samples. Each data point is the average deviation of the previous 50 samples. Within 100 iterations, the CAVE

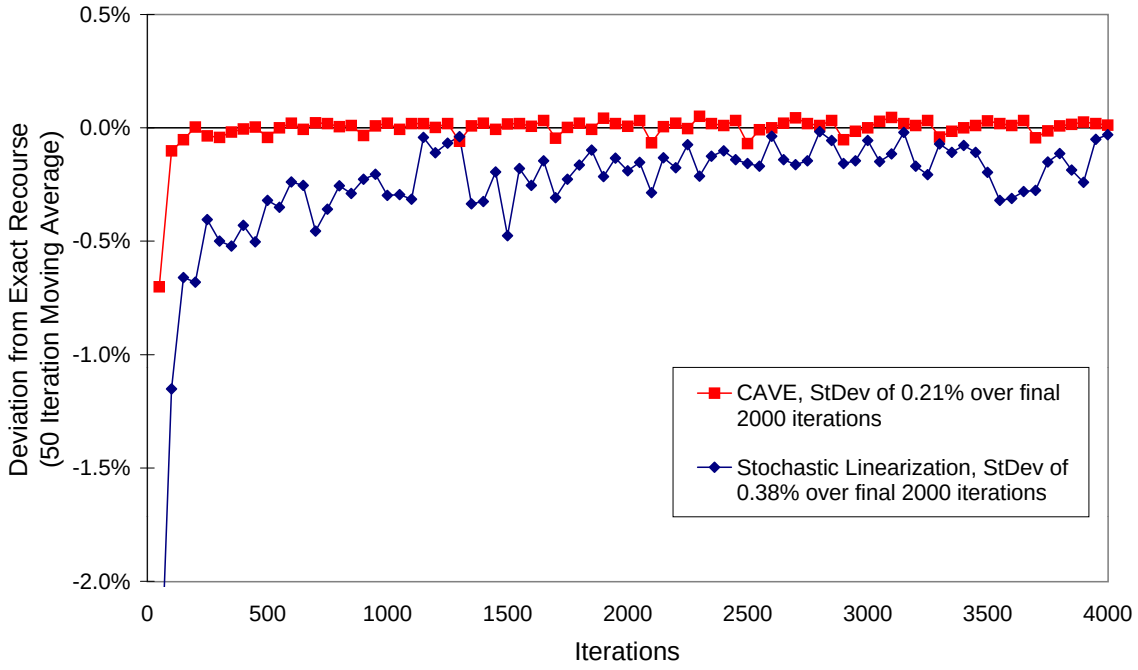

Figure 5: Piecewise-linear CAVE approximation versus linear stochastic quasi-gradient approximation using 400 random samples (each point is the average of the previous 50 iterations to smooth the results)

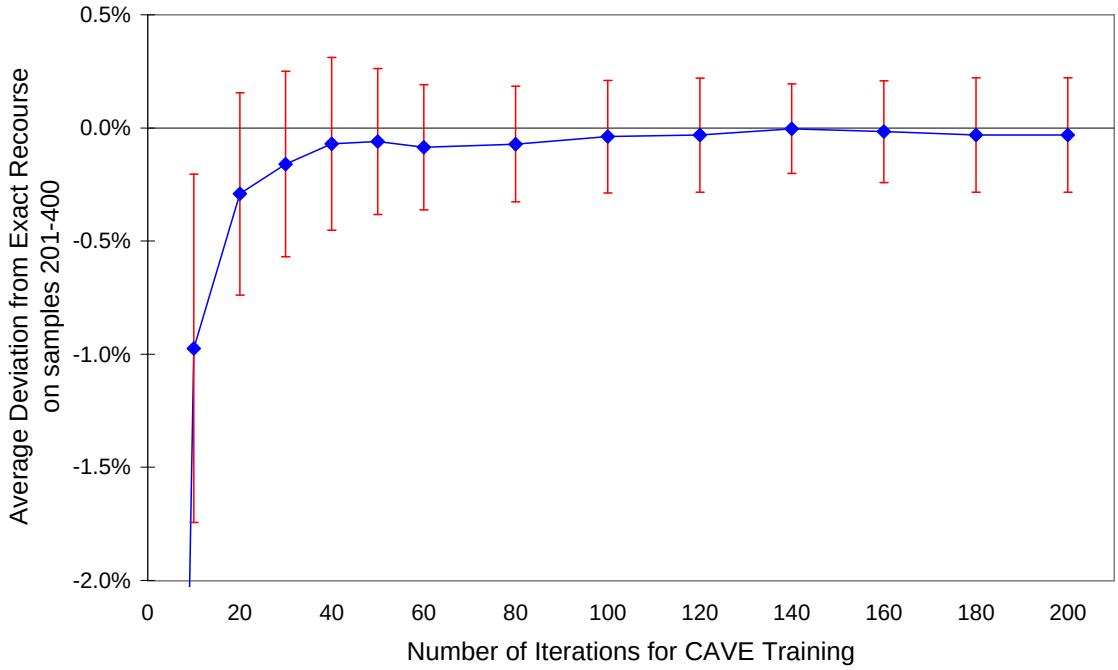

Figure 6: CAVE performance against Exact Expected Recourse function on samples 201-400 with CAVE approximation fixed after a specific number of training iterations

approximation provides results that are consistently as good as those produced by the exact recourse function. The stochastic linearization results, on the other hand, are within 0.5% in the first 200 iterations, but then wander significantly and converge much more slowly. Clearly, the piecewise-linear shape of the CAVE approximation provides better stability and allows the use of the left and right gradient information at different states to accelerate the convergence.

This does not mean that the CAVE approximation converges to the exact recourse function at each node. The reason is that the CAVE domain is restricted to the visited states based on the first-stage decisions. For this reason, we calculate the objective function value using the CAVE approximation by solving two separate network problems, one to get the first-stage profits and the other to get the correct second-stage profits. The same is done for the stochastic linearization. The key point is that the first-stage decisions based on the CAVE approximation appear to converge to the decisions based on the exact recourse function.

In the second set of experiments, we train the CAVE approximation on a specific number of samples, fix the approximation, and then compare its results with exact recourse on samples 201-400 without further updates. We did not include the stochastic quasi-gradient method due to the its slow convergence in the earlier experiment. Figure 6 features the results based on 10 to 200 training iterations. The error bars represent the standard deviation of the differences across the last 200 samples. After about 40 training iterations, the CAVE approximation produces nearly identical results to exact recourse. Since the exact expected recourse function produces the "best" results possible, the CAVE solutions appear to converge to the optimal solution.

5 Conclusion

In closing, we have introduced a new method for constructing concave, piecewise-linear approximations of concave, continuous functions using only repeated stochastic sample gradients at different points in the domain. The method (CAVE) exploits the recourse function

structure that exists in many stochastic programs with recourse to accelerate the slow converence normally exhibited by sample gradient methods. In addition, we have proven that the updating operations of the CAVE algorithm preserve concavity.

We have tested the quality of the CAVE approximations in three different stochastic environments, and in each case, the results have suggested convergent behavior. We plan to extend the CAVE research in later papers by demonstrating its effectiveness in solving multi-stage stochastic programs.

References

[1] Bell, P.C., "Forecasting demand variation when there are stockouts," *J. Operational Research Society*, 51, 3 (2000), 358–363.

[2] Birge, J.R. "Stochastic programming computation and applications," *INFORMS Journal on Computing*, 9, 2 (1997), 111–133.

[3] Braden, D.B. and M. Freimer, "Informational dynamics of censored observations," *Management Science*, 37, 11 (1991), 1390–1404.

[4] Ermoliev, Y., "Stochastic quasigradient methods," In Y. Ermoliev and R. Wets, editors, *Numerical Techniques for Stochastic Optimization*. Springer-Verlag, Berlin, (1988).

[5] Frantzeskakis, L. and W. B. Powell, "A successive linear approximation procedure for stochastic dynamic vehicle allocation problems," *Transportation Science*, 24, 1 (1990), 40–57.

[6] Godfrey,G. and W. B. Powell, "An adaptive, dynamic programming algorithm for stochastic resource allocation problems I: Single period travel times," Technical report, Department of Operations Research and Financial Engineering, Princeton University, 2000. Cl-00-04.

[7] Gupal, A.M. and L. G. Bazhenov, "A stochastic method of linearization," *Cybernetics* (1972), 482–484.

[8] Kall, P. and S. Wallace, *Stochastic Programming*. John Wiley and Sons, New York, 1994.

[9] Khouja, M., "The single-period (news-vendor)problem: literature review and suggestions for future research," *OMEGA-International Journal of Management Science*, 27, 5 (1999), 537–553.

[10] Lau, H.S. and A. H. L. Lan, "Estimating the demand distributions of single-period items having frequent Stockouts," *European Journal of Operational Research*, 92, 2 (1996), 254–265,

[11] Lee, H. L. and S. Nahmias, "Single-product, single-location models," In S. C. Graves, A. H. G. R. Kan, and P. H. Zipkin, editors, Handbook in Operations Research and Management Science, *Volume on* Logistics of Production and Inventory, North Holland (1993), 3–58.

[12] Porteus, E.L., "Stochastic Inventory Theory," *Handbook sin Operations Research and Management Science: Stochastic Models*, North Holland (1990), 605–652.

[13] Powell, W.B., "A comparative review of alternative algorithms for the dynamic vehicle allocation problem," In B. Golden and A. Assad, editors, *Vehicle Routing: Methods and Studies*, North Holland, New York (1988) 249–292.

[14] Powell, W.B. and R. K.-L. Cheung, "Stochastic programs over trees with random arc capacities," *Networks*, 24 (1994) 161–175.

[15] Rockafellar, R. and R. Wets, "Scenarios and policy aggregation in optimization under uncertainty," *Mathematics of Operations Research*, 16, 1 (1991) 119–147.

[16] Sen, S. and J. Higle, "An introductory tutorial on stochastic linear programming models," *Interfaces*, 29, 2 (1999), 33–61.

[17] Topaloglu, H. and W. B. Powell, "Dynamic programming approximations for time-staged integer multi-commodity flow problems," Technical Report CL-00-02, Department of Operations Research and Financial Engineering, Princeton University, 2000.

[18] Van Slyke, R. and R. Wets, "L-shaped linear programs with applications to optimal control and stochastic programming," *SIAM Journal of Applied Mathematics*, 17, 4 (1969), 638–663.

[19] Wallace, S., "Solving stochastic programs with network recourse," *Networks*, 16 (1986), 295–317.