Nonparametric Density Estimation for Stochastic Optimization with an Observable State Variable

Lauren A. Hannah
Duke University
Durham, NC 27701
lh140@duke.edu

Warren B. Powell
Princeton University
Princeton, NJ 08544
powell@princeton.edu

David M. Blei
Princeton University
Princeton, NJ 08544
blei@cs.princeton.edu

Abstract

In this paper we study convex stochastic optimization problems where a noisy objective function value is observed after a decision is made. There are many stochastic optimization problems whose behavior depends on an exogenous state variable which affects the shape of the objective function. Currently, there is no general purpose algorithm to solve this class of problems. We use nonparametric density estimation to take observations from the joint state-outcome distribution and use them to infer the optimal decision for a given query state s. We propose two solution methods that depend on the problem characteristics: function-based and gradient-based optimization. We examine two weighting schemes, kernel based weights and Dirichlet process based weights, for use with the solution methods. The weights and solution methods are tested on a synthetic multi-product newsvendor problem and the hour ahead wind commitment problem. Our results show that in some cases Dirichlet process weights offer substantial benefits over kernel based weights and more generally that nonparametric estimation methods provide good solutions to otherwise intractable problems.

1 Introduction

In stochastic optimization, a decision maker makes a decision and faces a random cost based on that decision. The goal is to choose a decision that minimizes the expected cost using information from previous observations. Stochastic optimization problems with continuous decision spaces have many viable solution methods, including function averaging and stochastic gradient descent [20]. However, in many situations conditions for the previous observations may not be the same as the current conditions; the conditions can be viewed as state variables. There are currently no general purpose solution methods for stochastic optimization problems with state variables, although they would be useful for finance, energy, dynamic pricing, inventory control and reinforcement learning applications.

We consider the newsvendor problem, a classic inventory management problem, to illustrate existing solution methods for stochastic optimization problems with state variables and their limitations. Here, newspapers can be bought in advance for cost c, and up to D of them can be sold for price p, where D is a random demand; the goal is to determine how many papers should be ordered so as to maximize the expected profit. A state variable that contains information about the random demand may also be included. For example, a rainy forecast may correlate to a lower demand while a sunny forecast may correlate to a higher. A natural solution method would be to partition the previous observations into "rainy" and "sunny" bins, and then solve the problem for each partition. This essentially models the problem as a single time period Markov Decision Process and solves the problem accordingly [16, 21]. Partitioning methods work when the state space can take a small number of discrete values.

1

Two problems arise with partitioning methods when the state space becomes larger. First, the number of states grows exponentially with the dimension of the state space. If there are 10 attributes, like weather, stock prices, days until an election, etc, and each can take 100 values, then there will be 10^{20} individual states. Second, previous observations are sparse over these states; a vast number of observations must be gathered before there are enough to make a reasonable decision for a given state. Rather than partitioning, we propose using observations from "similar" states to create a deterministic decision-expected cost function, also called an objective function, that is conditioned on a particular state.

Similar methods have been proposed in an approximate dynamic programming setting that use basis functions, such as linear and polynomial predictors, to construct approximate value functions [22, 14]. Basis functions, however, are hard to choose manually and automatic selection is an area of active research [12]. Moreover, basis functions do not guarantee that the approximate objective function is convex in the decision.

We propose using nonparametric density estimation for the joint state and outcome distribution to group observations from "similar" states with weights. These are then used to construct deterministic, convex approximations of the noisy function given the current observed information. The results are a deterministic, convex math program. These can be efficiently solved by a number of commercial solvers, even with very large decision spaces (10 to 1,000+ variables and constraints).

We give two methods to construct an approximate objective function using previous observations. The first is a function-based method. In some cases, entire random objective functions can be viewed retrospectively. For example, if the demand is known in the newsvendor problem, then the value of all decisions is also known. In these particular cases, the approximate objective function is modeled as a weighted average of the observed functions. The second method is based on stochastic gradients. In some cases, it is not possible to observe entire functions or observed functions may be too complex to manipulate. When this happens, we propose constructing a separable, piecewise linear approximate objective function. A piecewise linear, convex function is created in each decision dimension by generating a slope function from a weighted, order-restricted regression of the gradients, and then integrating that function. The result is an approximate objective function that is not necessarily the same as the original objective function, but one that has the same minima.

Both methods depend heavily on weights to capture dependence between the state and the outcome. We propose two weighting schemes: kernels weights and Dirichlet process mixture model weights. Kernels are simple to implement, but Dirichlet process mixture models have certain appealing properties. First, they act as a local bandwidth selector across the state space; second, the weights are generated by partitions rather than products of uni-dimensional weights, so the results scale better to higher-dimensional settings.

We contribute novel algorithms for stochastic optimization problems with a state variable that work with large, continuous decision spaces and propose a new use of Dirichlet process mixture models. We give empirical analysis for these methods where we show promising results on test problems. The paper is organized as follows. In Section 2, we review traditional function-based and gradient-based optimization methods and in each case present novel algorithms to accommodate an observable state variable. We present an empirical analysis of our methods for synthetic newsvendor data and the hour ahead wind commitment problem in Section 4 and a discussion in Section 5.

2 Stochastic optimization for problems with an observable state variable

Traditional stochastic optimization problems have the form

$$\min_{x \in \mathcal{X}} \mathbb{E}\left[F(x, Z)\right],\tag{1}$$

where $x \in \mathbb{R}^d$ is the decision, $Z : \Omega \to \Psi$ is a random outcome, $\mathcal{X}$ is a decision set and $F(x, Z(\omega))$ is a random objective function [20]. In the newsvendor problem, which we will use as a running example, x is the stocking level and Z is the random demand. Given x and $Z(\omega)$, F is deterministic. When a state variable is inlcuded, we first observe a random state $S \in \mathcal{S}$ that may influence F and the distribution of Z, then we make a decision x, and finally we observe the random variable Z. Eq. (1) becomes

$$\min_{x \in \mathcal{X}} \mathbb{E}\left[F(x, s, Z)|S = s\right].\tag{2}$$

2

Traditional stochastic optimization techniques require us to sample from the conditional distribution of $p(Z|S = s)$, treating each state observation independently [20]. We will use nonparametric density estimation for the joint distribution of (S, Z) to take into account that similar values of S affect Z and F in a similar way. We now describe new methods for function-based and gradient-based optimization for problems with an observable state variable.

2.1 Function-based optimization with an observable state variable

Function-based optimization is used when a single outcome ω can tell us the value of all decisions given that outcome [19]. For example, in the newsvendor problem, if the demand is known then the value of all inventory levels is known. Function-based optimization relies on sampling a set of scenarios, $\omega_1, \ldots, \omega_n$ from Ω, to approximate Eq. (1):

$$\min_{x \in \mathcal{X}} \frac{1}{n} \sum_{i=1}^{n} F(x, Z(\omega_i)). \tag{3}$$

Since Eq. (3) is deterministic given $\omega_{1:n}$, deterministic solution methods can be used. These methods are well developed and are implemented in a variety of commercial solvers.

When a state variable is introduced, we wish to solve Eq. (2) for a fixed query state $s \in \mathcal{S}$. However, scenarios are not i.i.d. from the distribution $p(Z|S = s)$, but rather from the joint distribution $(p(Z, S)$. Let $(S_i, Z(\omega_{i+1}))_{i=0}^{n-1}$ be a set of n observations. Instead of taking a naive average of the observations as in Eq. (3), we weight the observations based on the distance between the query state s and each observation S_i with weight $w_n(s, S_i)$. The weights must sum to 1, $\sum_{i=0}^{n-1} w_n(s, S_i) = 1$, and the weights may change with the number of observations, n. Set

$$\bar{F}_n(x|s) = \sum_{i=0}^{n-1} w_n(s, S_i) F(x, S_i, Z(\omega_{i+1})). \tag{4}$$

The optimization problem becomes

$$\min_{x \in \mathcal{X}} \bar{F}_n(x|s). \tag{5}$$

Note that because $F(x, S_i, Z(\omega_{i+1}))$ is convex in x for every S_i and ω_{i+1}, $\bar{F}_n(x|s)$ is convex and Eq. (5) can be solved with a commercial solver. We discuss weight functions in Section 3.

2.2 Gradient-based optimization with an observable state variable

In gradient-based optimization, we no longer observe an entire function $F(x, S, Z(\omega))$, but only a derivative taken at x,

$$\hat{\beta}(x_i, s, \omega_{i+1}) = \nabla_x F(x_i, s, Z(\omega_{i+1})). \tag{6}$$

Stochastic approximation is the most popular way to solve stochastic optimization problems using a gradient; it modifies gradient search algorithms to account for random gradients [17, 9]. The general idea is to optimize x by iterating,

$$x_{n+1} = \Gamma_{\mathcal{X}} \left(x_n - a_n \nabla_x F(x_n, Z(\omega_{n+1})) \right), \tag{7}$$

where $\Gamma_{\mathcal{X}}$ is a projection back into the constraint set $\mathcal{X}$, $\nabla_x F(x_n, Z(\omega_{n+1}))$ is a stochastic gradient at x_n and a_n is a stepsize. Other approaches to gradient-based optimization have included construction of piecewise linear, convex functions to approximate $F(x)$ in the region where x is near the optimal decision, x^* [15].

Including a state variable into gradient-based optimization is less straightforward than it is for function-based optimization. We run into difficulties because we choose x_n given S_n. When we include state S_n, the decision x_n is based on the state S_n. But x_{n-1} depends on S_{n-1}, so no iterative procedure like Eq. (7) can be used. Moreover, constructing the approximate function $\bar{F}_n(x|s)$ is not trivial because the stochastic gradients depend on *both* x_n and S_n.

Therefore, we propose modeling $F(x|s)$ with a piecewise linear, convex, separable approximation. Even if $F(x|s)$ is not itself separable, we aim to approximate it with a simpler (separable) function that has the same minimum for all fixed s. Approximating the minimum is easier than approximating

3
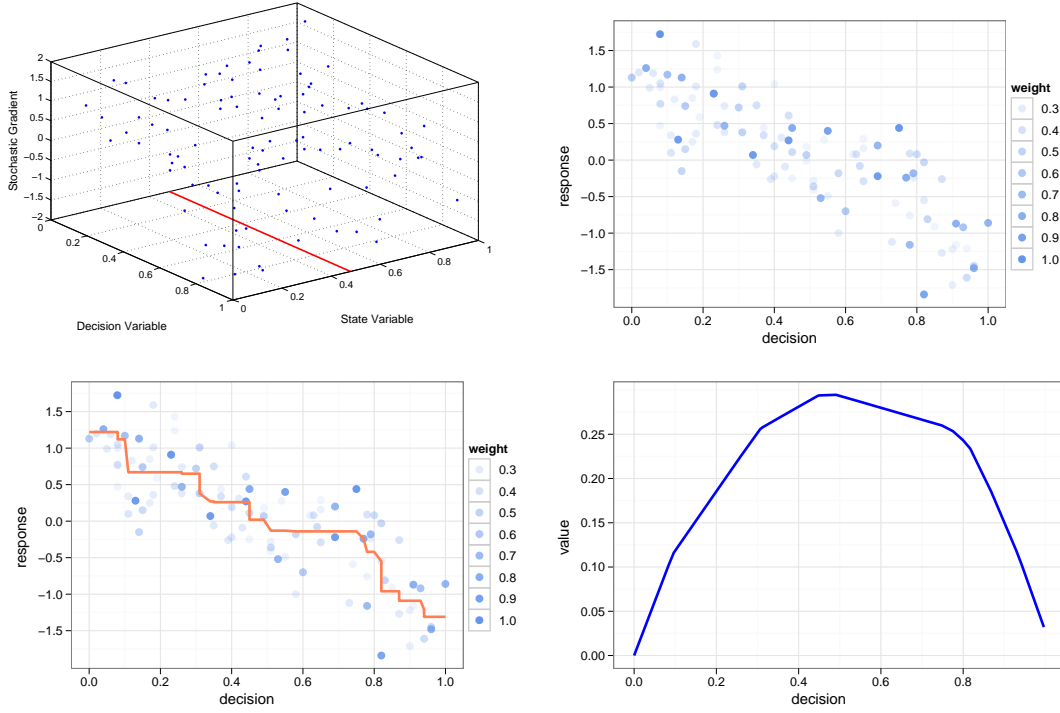
Figure 1: A graphical depiction of gradient-based method in one dimension for a maximization problem. (Top left) Observe gradients, state. (Top right) Weight observations based on state. (Bottom left) Fit isotonic regression to weighted slopes. (Bottom right) Integrate isotonic regression to form $f_n^k(x_n|S_n)$.

the entire convex function [4, 15]. Moreover, convex regression is easier in one dimension than multiple dimensions. We approximate $\mathbb{E}[F(x, s, Z)]$ by a series of separable functions,

$$\bar{F}_n(x|s) = \sum_{k=1}^{d} f_n^k(x^k|s),$$

where x^k is the kth component of x. We enforce convexity restrictions on $f_n^k(x|s)$ for every $s \in \mathcal{S}$. Unlike the function-based method, the gradient-based method is a fundamentally online algorithm: x_n is used to choose x_{n+1}. Given S_n, we choose x_n as follows,

$$x_n = \arg\min_{x \in \mathcal{X}} \sum_{k=1}^{d} f_n^k(x^k|S_n).$$

We then receive $\hat{\beta}(x_n, S_n, \omega_{n+1})$. The observations $(x_i, S_i, \hat{\beta}(x_i, S_i, \omega_{i+1}))_{i=0}^{n-1}$ are used to update $\bar{F}_n(x|s)$ sequentially. Fix $k \in \{1, \ldots, d\}$; we want to construct a piecewise linear $f_n^k(x|s)$ by constructing an increasing slope function, $v_n^k(x|S_n) = \frac{d}{dx} f_n^k(x|S_n)$ based on the stochastic gradient observations, $\hat{\beta}_{1:n}$. We use weights to group the gradients from states "similar" to S_n and a weighted isotonic (order restricted) regression to construct $v_n^k(x|S_n)$. Order the decision observations $x_{[0]}^k, \ldots, x_{[n-1]}^k$, and then solve to find slopes for the decision-ordered space,

$$v_n^k(x_{0:n-1}|S_n) = \arg\min_v \sum_{i=0}^{n-1} w_n\left(S_n, S_{[i]}\right) \left(\hat{\beta}(x_{[i]}^k, S_{[i]}, \omega_{[i+1]}) - v_{[i]}\right)^2, \qquad (8)$$

$$\text{subject to} : v_{[i-1]} \le v_{[i]}, \quad i = 1, \ldots, n-1.$$

First $v_n^k(x|S_n)$ is generated by interpolating the point estimates from Eq. (8) across the kth dimension of the decision space, and then $f^k(x|S_n)$ is created by integrating $v_n^k(x|S_n)$. The monotonicity of $v_n^k(x|S_n)$ ensures the convexity of $f_n^k(x|S_n)$. See Figure 1 for an example. The general method for constructing $\bar{F}_n(x|s)$ is as follows:

4

1. Observe S_n and constructing weights $((w_n(S_n, S_i))_{i=0}^{n-1}$,
2. Use the weights $w_n(S_n, S_i)_{i=0}^{n-1}$, previous decisions $x_{0:n-1}$ and gradients to construct slopes $v_{1:K}^k(S_n)$ with Eq. (8),
3. Reconstruct $f^k(x|S_n)$ from the slopes and construct $\bar{F}(x|S_n)$ from $(f^k(x|S_n))_{k=1}^d$, and
4. Choose x_n given $\bar{F}(x|S_n)$: $\quad x_n = \arg\min_{x \in \mathcal{X}} \bar{F}_n(x|S_n)$.

Details are given in the supplementary material. We now discuss the choice of weight functions.

3 Weight functions

Like the choice of step size in stochastic approximation, the choice of weight functions in Eqs. (4) and (8) determines whether and under which conditions function-based and gradient-based optimization produce acceptable results. Weighting functions rely on density estimation procedures to approximate the conditional density $f(z|s)$, where s is the state and z is the response. Conditional density estimation weights observations from a joint distribution to create a conditional distribution. We use this to obtain weights from two nonparametric density estimators, kernels and Dirichlet process mixture models.

3.1 Kernel weights

Kernel weights rely on kernel functions, $K(s)$, to be evaluated at each observation to approximate the conditional density. A common choice for K with continuous covariates is the Gaussian kernel, $K_h(s) = (2\pi h)^{-1/2} \exp\{-s^2/2h\}$, where the variance h is called the bandwidth. Kernel weights have the advantage of being simple and easy to implement. The simplest and most universally applicable weighting scheme is based on the Nadaraya-Watson estimator [10, 23]. If $K(s)$ is the kernel and h_n is the bandwidth after n observations, define

$$w_n(s, S_i) = K\left((s - S_i)/h_n\right) / \sum_{j=0}^{n-1} K\left((s - S_j)/h_n\right).$$

Kernel estimators require a well sampled space, are poor in higher dimensions and highly sensitive to bandwidth size [5].

3.2 Dirichlet process weights

One of the curses of dimensionality is sparseness of data: as the number of dimensions grows, the distance between observations grows exponentially. In kernel regression, this means that only a handful of observations have weights that are effectively non-zero, producing non-stable estimates. Instead, we would like to average responses for "similar" observations. We propose modeling the distribution of the state variable with a Dirichlet process mixture model, which is then decomposed into weights.

Dirichlet process mixture models. A mixture model represents a distribution, $g(s)$, as a weighted infinite sum of simpler distributions, $g(s \mid \theta_i)$, parameterized by θ_i, $g(s) = \sum_{i=1}^{\infty} p_i g(s \mid \theta_i)$. Here, p_i is the mixing proportion for component i. We can use a Dirichlet process (DP) with base measure $\mathbb{G}_0$ and concentration parameter α to place a distribution over the joint distribution of (p_i, θ_i), the mixture proportion and location of component i [6, 1]. Assume that data $S_1, \ldots, S_n$ are iid with a distribution that is modeled by a mixture over distribution $G(\theta)$,

$$P \sim DP(\alpha, \mathbb{G}_0), \qquad \theta_i | P \sim P, \qquad S_i | \theta_i \sim G(\theta_i). \qquad (9)$$

The distribution P drawn from a Dirichlet process is an almost surely discrete measure over parameters, with the mixture proportion associated with θ as the atomic weight. The hidden measure P in Eq. (9) can be integrated out to obtain a conditional distribution of $\theta_n | \theta_{1:n-1}$ [3]

$$\theta_n \mid \theta_1, \ldots, \theta_{n-1} \sim \frac{1}{\alpha + n - 1} \sum_{i=1}^{n-1} \delta_{\theta_i} + \frac{\alpha}{\alpha + n - 1} \mathbb{G}_0. \qquad (10)$$

Here, δ_θ is the Dirac measure with mass at θ. Eq. (10) is known as a Polya urn posterior; the variable θ_n has positive probability of assuming the value of one of the previously observed θ_i, but it also can take a new value drawn from $\mathbb{G}_0$ with positive probability. The parameter α controls how likely θ_n is to take a new value. We now discuss how weights can be constructed from Eq. (9).

5

Dirichlet process mixture model weights. A Dirichlet process mixture model can be used to model an unknown density, but it can simultaneously be used to produce a distribution of the partition structure of observed data [13, 8]. This is shown in the Polya urn posterior of Eq. (10); each hidden parameter has positive probability of taking the same value as another parameter. If two hidden parameters have the same value, they are in the same partition/cluster. The partition structure induces weights on the observations, proportional to 1 if they are in the same cluster, 0 if not.

Let $\mathbf{p} = \{C_1, \ldots, C_{n(\mathbf{p})}\}$ be the partition of the observations $\{1, \ldots, n\}$. Here $C_i = \{j : \theta_j = \theta_i^*\}$ is the partition set generated by $n(\mathbf{p})$ unique parameter values, denoted $\theta_1^*, \ldots, \theta_{n(\mathbf{p})}^*$. Now suppose that we know the partition $\mathbf{p}$. Given $\mathbf{p}$, we include the query state s into cluster C_i with probability

$$p_s(C_i|\mathbf{p}) = \mathbb{P}(s \in C_i \,|\, \mathbf{p}, S_{1:n}) \propto |C_i| \int g(s \,|\, \theta^*) dH_{C_i}(\theta^*),$$

where $|C_i|$ is the number of elements in C_i, and $H_{C_i}(\theta^*)$ is the posterior distribution of θ^* conditioned on $\mathbb{G}_0$ and the set of observations $\{S_j : S_j \in C_i\}$. Given $\mathbf{p}$, the weighting function is the probability that the hidden parameter for s would be θ_i, the hidden parameter for S_i,

$$w_n(s, S_i) \,|\, \mathbf{p} = \sum_{j=1}^{n(\mathbf{p})} \frac{p_s(C_j \,|\, \mathbf{p})}{|C_j|} \mathbf{1}_{\{S_i \in C_j\}}. \tag{11}$$

Eq. (11) is conditioned on a partition structure, but the Dirichlet process produces a distribution over partition structures. Let $\pi(\mathbf{p})$ be the prior distribution for partitions $\mathbf{p}$ and $\pi(\mathbf{p}|S_{0:n-1})$ the posterior. Integrating of the partition posterior, we obtain unconditional weights,

$$w_n(s, S_i) = \sum_{\mathbf{p}} \pi(\mathbf{p}|S_{1:n}) \sum_{j=1}^{n(\mathbf{p})} \frac{p_s(C_j \,|\, \mathbf{p})}{|C_j|} \mathbf{1}_{\{S_i \in C_j\}} \approx \frac{1}{M} \sum_{m=1}^{M} \sum_{j=1}^{n(\mathbf{p}^{(m)})} \frac{p_s(C_j \,|\, \mathbf{p}^{(m)})}{|C_j|} \mathbf{1}_{\{S_i \in C_j\}}. \tag{12}$$

It is infeasible to integrate over all of the partitions; therefore, we approximate Eq. (12) by performing a Monte Carlo integration with M posterior partition samples, $(\mathbf{p}^{(m)})_{m=1}^{M}$. We obtain $(\mathbf{p}^{(m)})_{m=1}^{M}$ by generating M iid samples of the hidden parameters, $\theta_{0:n-1}$, from the posterior of Eq. (9) with Gibbs sampling [11].

4 Empirical analysis

4.1 Multi-product constrained newsvendor problem

A multi-product newsvendor problem is a classic operations research inventory management problem. In the two product problem, a newsvendor is selling products A and B. She must decide how much of each product to stock in the face of random demand, D_A and D_B. A and B can be be bought for (c_A, c_B) and sold for (p_A, p_B), respectively. Any inventory not sold is lost. Let (x_A, x_B) be the stocking decisions for A and B respectively; it is subject to a budget constraint, $b_A \, x_A + b_B \, x_B \leq b$, and a storage constraint, $r_A \, x_A + r_B \, x_B \leq r$. An observable state $S = (S_1, S_2)$ contains information about D_A and D_B. The problem is,

$$\max_{x_A, x_B} \; -c_A \, x_A - c_B \, x_B + \mathbb{E}\left[p_A \, \min\left(x_A, D_A\right) + p_B \, \min\left(x_B, D_B\right) \,|\, S = s\right] \tag{13}$$

$$\text{subject to} : b_A \, x_A + b_B \, x_B \leq b, \quad r_A \, x_A + r_B \, x_B \leq r.$$

We generated data for Problem (13) in the following way. Demand and two state variables were generated in a jointly trimodal Gaussian mixture. The following methods were compared.

Function-based with kernel and **Gradient-based with kernel.** Bandwidth is selected according to the "rule of thumb" method of the *np* package for R, $h_j = 1.06\sigma_j n^{-1/(4+d)}$, where σ_j is defined as min(sd, interquartile range/1.349) [7].

Function-based with DP and **Gradient-based with DP.** We used the following hierarchical model,

$$P \sim DP(\alpha, \mathbb{G}_0), \qquad \theta_i = (\mu_{i,s}, \sigma_{i,s}^2)|P \sim P, \qquad S_{i,j}|\theta_i \sim N(\mu_{i,s,j}, \sigma_{i,s,j}^2), \; j = 1, 2.$$

6
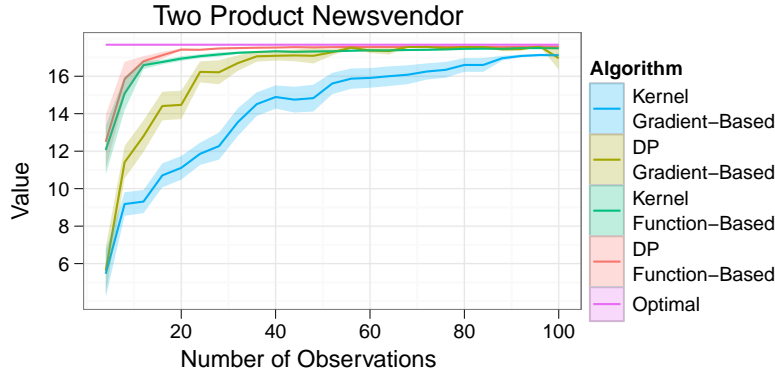
Figure 2: Gradient-based and function-based methods as a function of number of data points sampled. Results are averaged over 100 test problems with observed demand.

Posterior samples were drawn using Gibbs sampling with a fully collapsed sampler run for 500 iterations with a 200 iteration burn-in with samples taken every 5 iterations.

Optimal. These are the optimal decisions with known mixing parameters and unknown components.

Results. Decisions were made under each regime over eight sample paths; 100 test state/demand pairs were fixed and decisions were made for these problems given the observed states/decisions in the sample path for each method. Results are given in Figure 2. The kernel and Dirichlet process weights performed approximately equally for each method, but the function-based methods converged more quickly than the gradient-based methods.

4.2   Hour ahead wind commitment

In the hour ahead wind commitment problem, a wind farm manager must decide how much energy to promise a utility an hour in advance, incorporating knowledge about the current state of the world. The decision is the amount of wind energy pledged, a scalar variable. If more energy is pledged than is generated, the difference must be bought on the spot market, which is expensive with a price that is unknown when the decision is made; otherwise, the excess is lost. The goal is to maximize expected revenue. The observable state variable is the time of day, time of year, wind history from the past two hours, contract price and current spot price,

$$
\begin{array}{llll}
T_i^D &=& \text{time of day,} & T_i^Y &=& \text{time of year,} \\
P_i^S &=& \text{current spot price,} & P_i^C &=& \text{contract price,} \\
W_{i-1} &=& \text{wind speed an hour ago,} & W_i &=& \text{current wind speed,} \\
S_i &=& \text{observable state variable} & &=& (T_i^D, T_i^Y, P_i^C, P_i^S, W_i, W_{i-1}), \\
x_i &=& \text{amount of energy pledged,} & Y_{i+1}(x) &=& P_i^C x - P_{i+1}^S \max(x - W_{i+1}, 0).
\end{array}
$$

The revenue that the wind farm receives, $Y_{i+1}(x)$, depends on the variables P_{i+1}^S and W_{i+1}, which are not known until the next hour. We used wind speed data from the North American Land Data Assimilation System with hourly observations from 2002–2005 in the following locations: **Amarillo, TX.** Latitude: 35.125 N, Longitude: 101.50 W. The data have strong daily and seasonal patterns. The mean wind level is 186.29 $(m/s)^3$ with standard deviation 244.86. **Tehachapi, CA.** Latitude: 35.125 N, Longitude: 118.25 W. The data have strong seasonal patterns. The mean wind level is 89.45 $(m/s)^3$ with standard deviation 123.47.

Clean spot and contract price data for the time period were unavailable, so contract prices were generated by Gaussian random variables with a mean of 1 and variance of 0.10. Spot prices were generated by a mean-reverting (Ornstein-Uhlenbeck) process with a mean function that varies by time of day and time of year [18]. The data were analyzed separately for each location; they were divided by year, with one year used for training and the other three used for testing. The following methods were compared on this dataset:

Known wind. The wind is known, allowing maximum possible commitment, $x_i = W_{i+1}(\omega_{i+1})$. It serves as an upper bound for all of the methods.

7

METHOD/LOCATION	2002	2003	2004	2005
TEHACHAPI, CA				
KNOWN WIND	97.5	94.5	73.7	91.8
FUNCTION WITH KERNEL	78.8 *(80.8%)*	77.3 *(81.8%)*	58.9 *(79.9%)*	72.1 *(78.5%)*
FUNCTION WITH DP	**85.1** *(87.3%)*	**82.6** *(87.4%)*	**63.9** *(86.7%)*	**79.6** *(86.7%)*
IGNORE STATE	30.4 *(31.1%)*	31.1 *(32.9%)*	22.8 *(30.9%)*	29.3 *(31.9%)*
AMARILLO, TX				
KNOWN WIND	186.0	175.2	184.9	175.2
FUNCTION WITH KERNEL	155.1 *(83.4%)*	149.6 *(85.4%)*	154.7 *(83.7%)*	146.2 *(83.5%)*
FUNCTION WITH DP	**168.2** *(90.4%)*	**160.6** *(91.7%)*	**167.1** *(90.4%)*	**159.4** *(91.0%)*
IGNORE STATE	70.3 *(37.8%)*	68.7 *(39.2%)*	69.6 *(37.6%)*	66.1 *(37.7%)*

Table 1: Mean values of decisions by method, year and data set. Percentages of the upper bound, Known Wind, are given for the other methods.

Function-based with kernel. Function-based optimization where the weights are generated by a Gaussian kernel. Bandwidth is selected according to the "rule of thumb" method of the *np* package for R, $h_j = 1.06\sigma_j n^{-1/(4+d)}$, where σ_j is defined as min(sd, interquartile range/1.349) [7].

Function-based with DP. Function-based optimization with Dirichlet process based weights. We model the state distribution with the following hierarchical model,

$$P \sim DP(\alpha, \mathbb{G}_0), \qquad\qquad \theta_i|P \sim P,$$
$$T_i^D|\theta_i \sim \text{von Mises}(\mu_{i,D}, \phi_D), \qquad\qquad T_i^Y|\theta_i \sim \text{von Mises}(\mu_{i,Y}, \phi_Y),$$
$$P_i^C|\theta_i \sim N(\mu_{i,C}, \sigma_{i,C}^2), \qquad\qquad P_i^S|\theta_i \sim N(\mu_{i,S}, \sigma_{i,S}^2),$$
$$W_i|\theta_i \sim N(\mu_{i,W1}, \sigma_{i,W1}^2), \qquad\qquad W_{i-1}|\theta_i \sim N(\mu_{i,W2}, \sigma_{i,W2}^2),$$
$$\theta_i = (\mu_{i,D}, \mu_{i,Y}, \mu_{i,C}, \sigma_{i,C}^2, \mu_{i,S}, \sigma_{i,S}^2, \mu_{i,W1}, \sigma_{i,W1}^2, \mu_{i,W2}, \sigma_{i,W2}^2).$$

We modeled the time of day, T_i^D, and year, T_i^Y, with a von Mises distribution, an exponential family distribution over the unit sphere; the dispersion parameters, ϕ_D and ϕ_Y, are hyperparameters. The base measure was Normal-Inverse Gamma for P_i^C, P_i^S, W_i and W_{i-1} and uniform for the means of T_i^D and T_i^Y. 100 posterior samples were drawn using Gibbs sampling with a collapsed sampler for all conjugate dimensions after a 1,000 iteration burn-in and 10 iteration pulse between samples.

Ignore state. Sample average approximation is used, $\bar{F}_n(x|s) = \frac{1}{n}\sum_{i=0}^{n-1} Y_{i+1}(x)$.

Results. Results are presented in Table 1. We display the value of each algorithm, along with percentages of Known Wind for the other three methods. Both forms of function-based optimization outperformed the algorithm in which the state variable was ignored by a large margin ($\geq$45% of the best possible value). Dirichlet process weights outperformed kernel weights by a smaller but still significant margin (5.6–8.2% of best possible value).

5 Discussion

We presented two new methods to solve stochastic optimization problems with an observable state variable, including state variables that are too large for partitioning. Our methods make minimal assumptions. They are promising additions to areas that rely on observational data to make decisions under changing conditions (energy, finance, dynamic pricing, inventory management), and some communities that make sequential decisions under uncertainty (reinforcement learning, stochastic programming, simulation optimization). Our methods can accommodate much larger state and decision spaces than MDPs and other table lookup methods, particularly when combined with Dirichlet process mixture model weights. Unlike existing objective function approximation methods, such as basis functions, our methods provide convex objective function approximations that can be used with a variety of commercial solvers.

Acknowledgments

The research was funded in part by the Air Force Office of Scientific Research under AFOSR contract FA9550-08-1-0195, and the NSF under grant CMMI-0856153. David M. Blei is supported by ONR 175-6343, NSF CAREER 0745520, AFOSR-09NL202 and the Alfred P. Sloan foundation.

8

References

[1] Antoniak, C. E. [1974], 'Mixtures of Dirichlet processes with applications to Bayesian non-parametric problems', *The Annals of Statistics* **2**(6), 1152–1174.

[2] Bennett, K. P. and Parrado-Hernández, E. [2006], 'The interplay of optimization and machine learning research', *The Journal of Machine Learning Research* **7**, 1265–1281.

[3] Blackwell, D. and MacQueen, J. B. [1973], 'Ferguson distributions via Polya urn schemes', *The Annals Statistics* **1**(2), 353–355.

[4] Cheung, R. K. and Powell, W. B. [2000], 'SHAPE-A stochastic hybrid approximation procedure for two-stage stochastic programs', *Operations Research* **48**(1), 73–79.

[5] Fan, J. and Gijbels, I. [1996], *Local Polynomial Modelling and Its Applications*, Chapman & Hall/CRC.

[6] Ferguson, T. S. [1973], 'A Bayesian analysis of some nonparametric problems', *The Annals of Statistics* **1**(2), 209–230.

[7] Hayfield, T. and Racine, J. S. [2008], 'Nonparametric econometrics: The np package', *Journal of Statistical Software* **27**(5), 1–32.

[8] Ishwaran, H. and James, L. F. [2003], 'Generalized weighted Chinese restaurant processes for species sampling mixture models', *Statistica Sinica* **13**(4), 1211–1236.

[9] Kiefer, J. and Wolfowitz, J. [1952], 'Stochastic estimation of the maximum of a regression function', *The Annals of Mathematical Statistics* **23**(3), 462–466.

[10] Nadaraya, E. A. [1964], 'On estimating regression', *Theory of Probability and its Applications* **9**(1), 141–142.

[11] Neal, R. M. [2000], 'Markov chain sampling methods for Dirichlet process mixture models', *Journal of Computational and Graphical Statistics* **9**(2), 249–265.

[12] Parr, R., Painter-Wakefield, C., Li, L. and Littman, M. [2007], Analyzing feature generation for value-function approximation, *in* 'Proceedings of the 24th international conference on Machine learning', ACM, p. 744.

[13] Pitman, J. [1996], 'Some developments of the Blackwell-MacQueen urn scheme', *Lecture Notes-Monograph Series* **30**, 245–267.

[14] Powell, W. B. [2007], *Approximate Dynamic Programming: Solving the curses of dimensionality*, Wiley-Blackwell.

[15] Powell, W. B., Ruszczyński, A. and Topaloglu, H. [2004], 'Learning algorithms for separable approximations of discrete stochastic optimization problems', *Mathematics of Operations Research* **29**(4), 814–836.

[16] Puterman, M. L. [1994], *Markov decision processes: Discrete stochastic dynamic programming*, John Wiley & Sons, Inc. New York, NY, USA.

[17] Robbins, H. and Monro, S. [1951], 'A stochastic approximation method', *The Annals of Mathematical Statistics* **22**(3), 400–407.

[18] Schwartz, E. S. [1997], 'The stochastic behavior of commodity prices: Implications for valuation and hedging', *The Journal of Finance* **52**(3), 923–973.

[19] Shapiro, A., Homem-de Mello, T. and Kim, J. [2002], 'Conditioning of convex piecewise linear stochastic programs', *Mathematical Programming* **94**(1), 1–19.

[20] Spall, J. C. [2003], *Introduction to stochastic search and optimization: estimation, simulation, and control*, John Wiley and Sons.

[21] Sutton, R. S. and Barto, A. G. [1998], *Introduction to reinforcement learning*, MIT Press Cambridge, MA, USA.

[22] Tsitsiklis, J. N. and Van Roy, B. [2001], 'Regression methods for pricing complex American-style options', *IEEE Transactions on Neural Networks* **12**(4), 694–703.

[23] Watson, G. S. [1964], 'Smooth regression analysis', *Sankhyā: The Indian Journal of Statistics, Series A* **26**(4), 359–372.

9