Supplement for 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

This is the supplement for "Nonparametric Density Estimation for Stochastic Optimization with an Observable State Variable." It contains details on constructing the approximate function for gradient-based optimization.

1 Details for gradient-based optimization

Let $\bar{F}_n(x \mid s)$ be the approximate function. We construct it in the following manner.

Step 1: Observe S_n and generate weights $(w_n(S_n, S_i))_{i=0}^{n-1}$. This is discussed in Section 3 of the main paper. An example observation of stochastic gradients and random state is given in Figure 1. An example of those stochastic gradients weighted by the state is given in Figure 2.

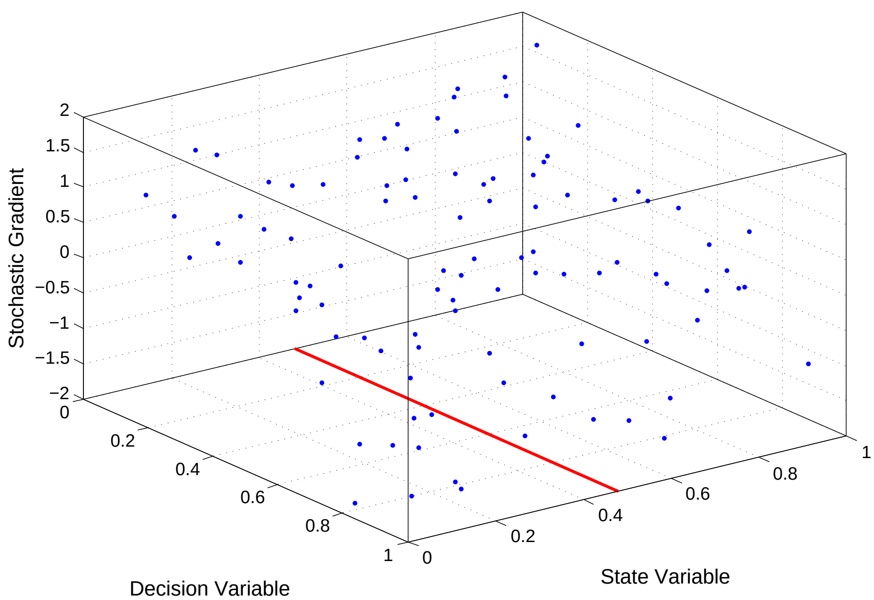

Figure 1: Observe gradients, here blue dots, and the random state, S_n, here a red line.

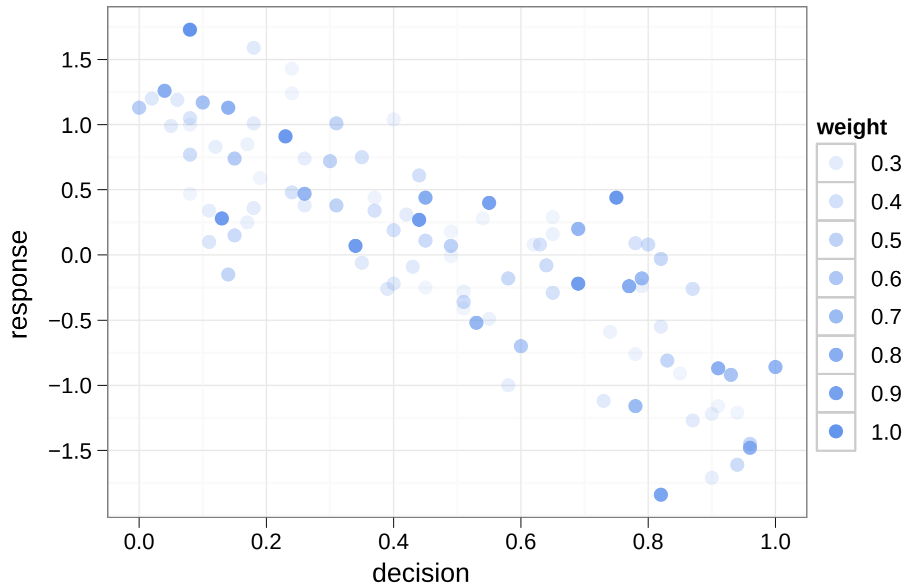

Figure 2: Weight observed gradients with $(w_n(S_n, S_i))_{i=0}^{n-1}$. Shading indicates relative weight.

Step 2: Construct slopes for $f_n^k(x|S_n)$ given gradient $\hat{\beta}_{1:n}$, decisions $x_{0:n-1}$ and weights $(w_n(S_n, S_i))_{i=0}^{n-1}$. We begin by placing the observed decisions in ascending order: $x_{[0]}^k \leq x_{[1]}^k \leq \cdots \leq x_{[n-1]}^k$, where $[0], \dots, [n-1]$ is the ordered numbering. A necessary and sufficient condition for $f_n^k(x|S_n)$ to be convex is for the slopes to be nondecreasing; that is,

$$\frac{d}{dx} f_n^k(x|S_n) \leq \frac{d}{dy} f_n^k(y|S_n)$$

for every $x \leq y$. We find a set of slopes $v_{n,[0]}^k(S_n) \leq \cdots \leq v_{n,[n-1]}^k(S_n)$ corresponding to the ordered decisions $x_{[0]}^k, \dots, x_{[n-1]}^k$ using weighted least squares minimization, which is a quadratic program,

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

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

Equation (1) is a quadratic program and easily solvable with a solver. An example of slopes found in this manner is given in Figure 3.

Step 3: Reconstruct marginal functions $f_n^k(x|S_n)$ and approximate function $\bar{F}_n(x|s)$ given slopes $v_n^k(s) = \{v_{n,[0]}^k(S_n), \dots, v_{n,[n-1]}^k(S_n)\}$. Suppose that $\mathcal{X}$ is compact; there exists a minimum value x_{min}^k and a maximum value x_{max}^k for each dimension k. Set $x_{[-1]}^k = x_{min}^k$ and $x_{[n]}^k = x_{max}^k$. Define $f_n^k(x|S_n)$ as follows,

$$f_n^k(x|S_n) = \sum_{i=0}^{\ell} v_{n,[i]}^k(S_n)\,(x_{[i]}^k - x_{[i-1]}^k) + v_{n,[\ell]}^k(S_n)\,(x - x_{[\ell]}^k), \tag{2}$$

where ℓ is the smallest index such that $x_{[\ell]}^k \leq x < x_{[\ell+1]}^k$. An example is given in Figure 4. Construct the approximate function by setting

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

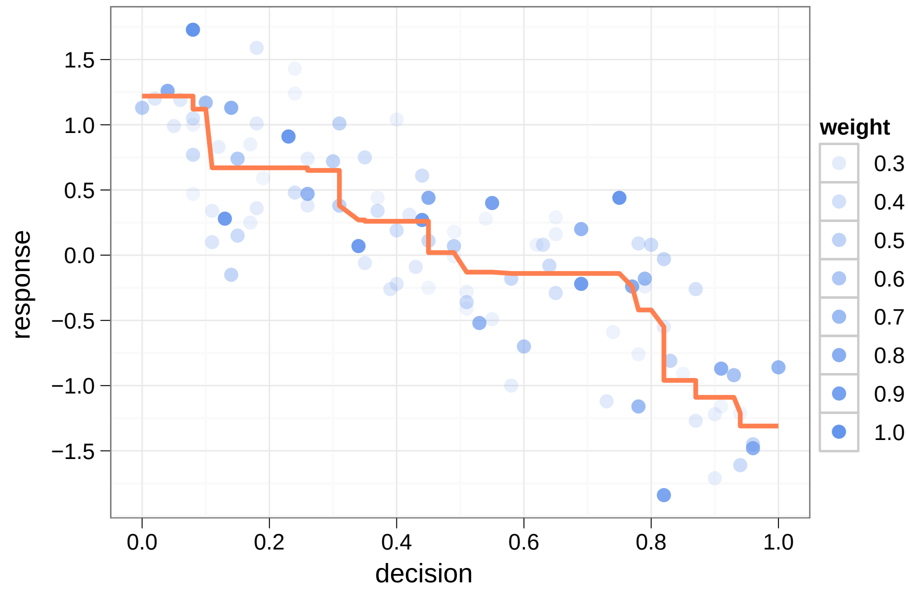

Figure 3: Weight observed gradients with $(w_n(S_n, S_i))_{i=0}^{n-1}$, with slope found via Equation (1).

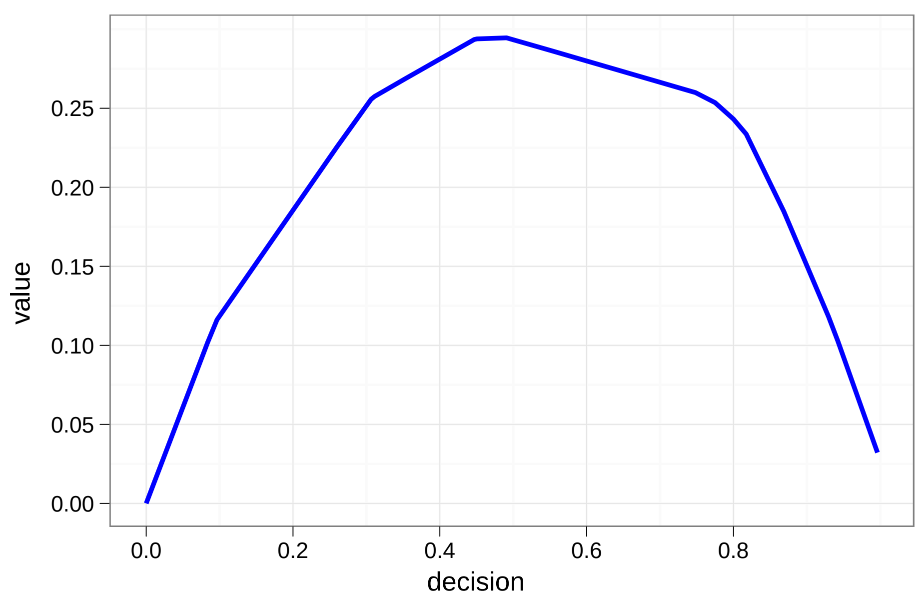

Figure 4: Reconstructed marginal function $f_n^k(x|S_n)$ from slope function in Figure 3.

Step 4: Choose x_n given $\bar{F}_n(x|S_n)$. We want to choose an x_n so that we learn as much as possible for an arbitrary s. This is done by picking x_n as follows,

$$x_n = \arg\min_{x \in \mathcal{X}} \bar{F}_n(x|S_n). \tag{3}$$

Note that $\bar{F}_n$ is a piecewise linear function; if $\mathcal{X}$ is a linear constraint set, the minimum can be found with a linear program. Note that linear programs can quickly be solved for thousands of decision variables and constraints.

The full implementation is given in Algorithm 1.

Algorithm 1: Gradient-based optimization with an observable state variable

Require: Query state s, initial slopes v_0.

1: **for** $i = 0$ to $n-1$ **do**
2: Observe random state S_i.
3: Generate weights $(w_i(S_i, S_j))_{j=0}^{i-1}$.
4: **for** $k = 1$ to d **do**
5: Place decision observations in ascending order: $x^k_{[0]} \leq x^k_{[1]} \leq \cdots \leq x^k_{[i-1]}$.
6: Compute slopes $v^k_i(S_i)$ by

$$v^k_i(S_i) = \arg\min_v \sum_{j=0}^{i-1} w_i\left(S_i, S_{[j]}\right)\left(\hat{\beta}(x^k_{[j]}, S_{[j]}, \omega_{[j+1]}) - v_{[j]}\right)^2,$$

$$\text{subject to}: v_{[j-1]} \leq v_{[j]}, \quad j = 1, \dots, i-1.$$

7: Reconstruct marginal function $f^k_i(x^k|S_i)$ using slopes $v^k_i(S_i)$ by

$$f^k_i(x|S_i) = \sum_{j=0}^{\ell} v^k_{i,[j]}(S_i)\,(x^k_{[j]} - x^k_{[j-1]}) + v^k_{i,[\ell]}(S_i)\,(x - x^k_{[\ell]}),$$

where ℓ is the smallest index such that $x^k_{[\ell]} \leq x < x^k_{[\ell+1]}$.

8: **end for**
9: Set

$$x_i = \arg\min_{x \in \mathcal{X}} \sum_{k=1}^{d} f^k_n(x^k|S_i).$$

10: Observe random gradient $\hat{\beta}(x_i, S_i, \omega_{i+1}) = \nabla_x F(x_i, S_i, Z(\omega_{i+1}))$.
11: **end for**
12: Compute $v^k_n(s)$, $k = 1, \dots, d$ as in Step 6.
13: Compute $f^k_n(x^k|s)$, $k = 1, \dots, d$ using $v^k_n(s)$ as in Step 7.
14: Set

$$x^*_n(s) = \arg\min_{x \in \mathcal{X}} \sum_{k=1}^{d} f^k_n(x^k|s).$$