Meeting Update

Linear Model Selection Using Minimum Energy Designs

Kristyn Pantoja

Department of Statistics
Texas AM University

28 February 2019

Review of “Sequential Exploration of Complex Surfaces Using Minimum Energy Designs,” Joseph et. al. 2015

“Sequential Exploration of Complex Surfaces Using Minimum Energy Designs,” Joseph et. al. 2015

The Point$^{\text{TM}}$

To explore unknown regions of the design space of particular interest to an experimenter by minimization of the total potential energy of charged particles (design points) inside a box. (Joseph 2015)
This method allows Bayesians to obtain deterministic and representative samples from a(n unnormalized) posterior distribution with fewer samples than MCMC by choosing samples that are far away enough but also also have high posterior probability.

Our Goal

We want to apply this idea to choose the design points that will allow us to discriminate between two linear models.

MED Criterion

Generalized Version of MED

Choose the design $\mathbf{D} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ given by

$$\min_{\mathbf{D}} \mathrm{GE}_k = \left\{ \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(\frac{q(\mathbf{x}_i)q(\mathbf{x}_j)}{d(\mathbf{x}_i, \mathbf{x}_j)} \right)^k \right\}^{1/k} \tag{1}$$

$k \in [1, \infty)$
((4) in Joseph 2015)

Criterion when $k \to \infty$

$$\max_{\mathbf{D}} \min_{i,j} \frac{d(\mathbf{x}_i, \mathbf{x}_j)}{q(\mathbf{x}_i)q(\mathbf{x}_j)} \tag{2}$$

((5) in Joseph 2015)

A Greedy Algorithm

Generate an N-point MED $\mathbf{D}_N$ for f:

1. Obtain *numCandidates* candidate points, $\mathbf{x}$, in $[0,1]^p$ (rescale, if necessary).
2. Initialize D_N by choosing $\mathbf{x}_i$ to be the candidate $\mathbf{x}$ with the highest value for $f(\mathbf{x})$, i.e.
$$\mathbf{x}_1 = \arg\max_{\mathbf{x}} f(\mathbf{x}) \tag{3}$$
3. Choose the next point $\mathbf{x}_{n+1}$ by:
$$\mathbf{x}_{n+1} = \arg\min_{\mathbf{x}} \sum_{i=1}^{n} \left(\frac{q(\mathbf{x}_i)q(\mathbf{x})}{d(\mathbf{x}_i, \mathbf{x})} \right)^k \tag{4}$$
   where $q(\mathbf{x}) = 1/\{f(\mathbf{x})\}^{(1/2p)}$, $d(x, y)$ is Euclidean distance, and (suggested that) $k = 4p$.

(p. 67 Joseph 2015)

Estimating f (when the time comes)

When the true response surface f is unknown or expensive to evaluate, generate an N-point MED D_N by " 'learn'[ing] about the underlying surface f sequentially and implemen[ting] MED accordingly." (p. 69 Joseph 2015):

1. Suppose we have already generated an n-point design $\mathbf{D}_n = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ and obtained the observations $\mathbf{y}^{(n)} = (y_1, ..., y_n)^T$, where $y_i = f(\mathbf{x}_i)$
2. Use some statistical techniques such as regression or kriging to estimate the underlying response surface. Denote it by $\hat{f}^{(n)}(\mathbf{x})$. Let $\hat{q}^{(n)}(\mathbf{x}) = 1/\{\hat{f}^{(n)}(\mathbf{x})\}^{(1/2p)}$.
3. Choose the next point $\mathbf{x}_{n+1}$ by:

$$\mathbf{x}_{n+1} = \arg\min_{\mathbf{x}} \sum_{i=1}^{n} \left(\frac{\hat{q}^{(n)}(\mathbf{x}_i)\hat{q}^{(n)}(\mathbf{x})}{d(\mathbf{x}_i, \mathbf{x})} \right)^k \tag{5}$$

The generated design is called Sequential Minimum Energy Design (SMED). (p. 67 Joseph 2015)

Some Computational Tricks

The problem with the above method for obtaining SMED is being unable to ensure that the starting point is good. Some ideas for this (p. 69, Joseph 2015):

1. Something like"burn-in"
2. Fractional-factorial design as initial design for physical experiments
3. Space-filling design as initial design for computer experiments
4. An MED as the initial design: Let

   $$\hat{q}^{(n)}(\mathbf{x}) = 1/\{\hat{f}^{(n)}(\mathbf{x})\}^{\gamma_n} \tag{6}$$

   where $\gamma_n = 0$ for $n = 1, \ldots, n_0$ to generate the initial design (?????) and $\gamma_n = 1/(2p)$ for $n \geq n_0$ (or allow γ_n to increase from 0 to $1/(2p)$ for $n = 1, \ldots, n_0$ for some adaptability).

The generated design is called Sequential Minimum Energy Design (SMED). (p. 67 Joseph 2015)

SMED for Linear Model Selection

SMED Applied to Model Selection for Linear Models

Something motivational here.

Recall our Goal

We want to apply this idea to choose the design points that will allow us to discriminate between two linear models, $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$.

Why?

We want design points that are spaced out enough but still important enough to both models.

How?

Instead of taking the charge function q to be $1/f^{\alpha}$, let

$$q(\mathbf{x}) = 1/\mathrm{W}^{\alpha}(\phi_{0,\mathbf{x}}, \phi_{1,\mathbf{x}}) \tag{7}$$

where $\phi_{i,\mathbf{x}}$ is the Normal distribution of y at $\mathbf{x}$ for linear model $i = 1, 2$, and W is the Wasserstein distance.
(in Joseph 2015, $\alpha = 1/2p$)

The Linear Model

$$y = f(\mathbf{x}) + \epsilon$$
$$f(x) = \mathbf{x}\beta$$

Where $\epsilon \sim N(0, \sigma_\epsilon^2)$, hence $y \sim N(\mathbf{x}\beta, \sigma_\epsilon^2)$.

- For model $f_0(\mathbf{x}) = \mathbf{x}\beta_0$, we have $y \sim N(\mathbf{x}\beta_0, \sigma_{\epsilon_0}^2) := \phi_{0,\mathbf{x}}$.
- For model $f_1(\mathbf{x}) = \mathbf{x}\beta_1$, we have $y \sim N(\mathbf{x}\beta_1, \sigma_{\epsilon_1}^2) := \phi_{1,\mathbf{x}}$.
- The Wasserstein distance $\mathrm{W}(\phi_1, \phi_2)$ between two Normals $\phi_1 = N(\mu_1, \Sigma_1)$ and $\phi_2 = N(\mu_2, \Sigma_2)$ can be obtained from:

$$\mathrm{W}^2(\phi_1, \phi_2) = ||\mu_1 - \mu_2||_2^2 + \mathrm{Tr}(\Sigma_1 + \Sigma_2 - 2(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2})^{1/2}) \tag{8}$$

- And for $\phi_{0,\mathbf{x}}$ and $\phi_{1,\mathbf{x}}$ at a given $\mathbf{x}$,

$$\mathrm{W}^2(\phi_{0,\mathbf{x}}, \phi_{1,\mathbf{x}}) = (\mathbf{x}\beta_0 - \mathbf{x}\beta_1)^2 + \sigma_{\epsilon_0}^2 + \sigma_{\epsilon_1}^2 - 2\sqrt{\sigma_{\epsilon_0}^2\sigma_{\epsilon_1}^2} \tag{9}$$

A Greedy Algorithm for Linear Model Selection

Generate an N-point MED $\mathbf{D}_N$ for f:

1. Obtain *numCandidates* candidate points, $\mathbf{x}$, in $[0,1]$.
2. Initialize D_N by choosing $\mathbf{x}_i$ to be the candidate $\mathbf{x}$ which results in the largest absolute difference between the two models, i.e.

$$\mathbf{x}_1 = \arg\max_{\mathbf{x}} |f_0(\mathbf{x}) - f_1(\mathbf{x})| \tag{10}$$

3. Choose the next point $\mathbf{x}_{n+1}$ by:

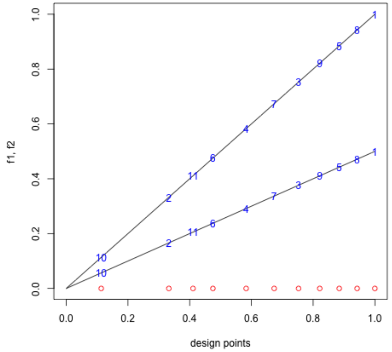
$$\mathbf{x}_{n+1} = \arg\min_{\mathbf{x}} \sum_{i=1}^{n} \left(\frac{q(\mathbf{x}_i) q(\mathbf{x})}{d(\mathbf{x}_i, \mathbf{x})} \right)^k \tag{11}$$

where $q(\mathbf{x}) = 1/\{\mathrm{W}(\phi_{0,\mathbf{x}}, \phi_{1,\mathbf{x}})\}^{(1/2p)}$, $d(x,y)$ is Euclidean distance, and (suggested that) $k=4$.

Results

The 2 Models

- $f_0(\mathbf{x}) = \mathbf{x}\beta_0$ with $\beta_0 = 1$
- $f_1(\mathbf{x}) = \mathbf{x}\beta_1$ with $\beta_1 = 1/2$
- $\sigma_{\epsilon 0} = \sigma_{\epsilon 1}$



Some Observations

- It seems that the SMED method (Joseph 2015) will iteratively choose bisection points until the points become too close, then explore farther-away points

Making it Bayesian: Putting a Prior on the Slope

Why make it Bayesian?

Recall our Goal

We want to apply this idea to choose the design points that will allow us to discriminate between two linear models, $f_0(\mathbf{x})$ and $f_1(\mathbf{x})$ so that we can obtain a design with design points that are spaced out enough but still important enough to both models.

Why Bayesian?

In most cases, the slope β is an unknown parameter, but we might have some idea of what it should be, so we put a (Normal) prior on it.

Prior on β

$$\beta \sim N(\tilde{\beta}, \sigma_\beta^2)$$

Bayesian Linear Model

$$y = f(\mathbf{x}) + \epsilon$$
$$f(x) = \mathbf{x}\beta$$

Where $\epsilon \sim N(0, \sigma_\epsilon^2)$ and $\beta \sim N(\tilde{\beta}, \sigma_\beta^2)$.
Hence $y \sim N(\mu_y, \sigma_y^2)$ where μ_y, σ_y^2 can be computed by iterated expectation and variance:

$$\mu_y = E[y] = E[E[y|\beta]] = \tilde{\beta}\mathbf{x}$$
$$\sigma_y^2 = E[y] = E[Var[y|\beta]] + Var[E[y|\beta]] = \sigma_\epsilon^2 + \mathbf{x}^2\sigma_\beta^2$$

- For model $f_0(\mathbf{x}) = \mathbf{x}\beta_0$, we have $y \sim N(\tilde{\beta}_0\mathbf{x}, \sigma_{\epsilon_0}^2 + \mathbf{x}^2\sigma_{\beta_0}^2) := \phi_{0,\mathbf{x}}$.
- For model $f_1(\mathbf{x}) = \mathbf{x}\beta_1$, we have $y \sim N(\tilde{\beta}_1\mathbf{x}, \sigma_{\epsilon_1}^2 + \mathbf{x}^2\sigma_{\beta_1}^2) := \phi_{1,\mathbf{x}}$.
- The Wasserstein distance can be computed like before.

A Greedy Algorithm for Bayesian Linear Model Selection

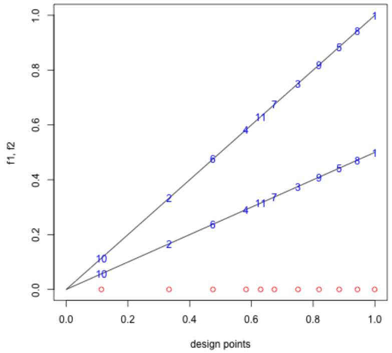
Generate an N-point MED $\mathbf{D}_N$ for f:

1. Sample β_i from the prior on each of the two models: $\beta_i \sim N(\tilde{\beta}_i, \sigma^2_{\beta_i})$
2. Obtain *numCandidates* candidate points, $\mathbf{x}$, in $[0, 1]$.
3. Initialize D_N by choosing $\mathbf{x}_i$ to be the candidate $\mathbf{x}$ which results in the largest absolute difference between the two models, i.e.
$$\mathbf{x}_1 = \arg\max_{\mathbf{x}} |f_0(\mathbf{x}) - f_1(\mathbf{x})| \tag{12}$$
4. Choose the next point $\mathbf{x}_{n+1}$ by:
$$\mathbf{x}_{n+1} = \arg\min_{\mathbf{x}} \sum_{i=1}^{n} \left(\frac{q(\mathbf{x}_i)q(\mathbf{x})}{d(\mathbf{x}_i, \mathbf{x})} \right)^k \tag{13}$$
where $q(\mathbf{x}) = 1/\{\text{Wasserstein}(\phi_{0,\mathbf{x}}, \phi_{1,\mathbf{x}})\}^{(1/2p)}$, $d(x, y)$ is Euclidean distance, and (suggested that) $k = 4$.

Results

The 2 Models

- $f_0(\mathbf{x}) = \mathbf{x}\beta_0$ with $\tilde{\beta}_0 = 1$
- $f_1(\mathbf{x}) = \mathbf{x}\beta_1$ with $\tilde{\beta}_1 = 1/2$
- $\sigma_{\epsilon_0} = \sigma_{\epsilon_1}$



Observations

- Results are similar to the non-Bayesian case, except with added variance from the β prior, and sampled β's lead to slightly different lines, if you choose to model by those, instead of the slope means, $\tilde{\beta}$.

Review of “Deterministic Sampling of Expensive Posteriors Using Minimum Energy Designs,” Joseph et. al. 2018

“Deterministic Sampling of Expensive Posteriors Using Minimum Energy Designs,” Joseph et. al. 2018

Abstract

- *Summary of (Joseph 2015):* minimum energy design (MinED) is a recently proposed deterministic sampling method, which makes use of the posterior evaluations to obtain a weighted space-filling design in the region of interest.
- *Problem:* the existing implementation of MinED is inefficient because it requires several global optimizations and thus numerous evaluations of the posterior.
- *(Joseph 2018):* proposes an efficient algorithm that can generate MinED samples with few posterior evaluations and works better in high dimensions.

Our Goal

We want this too, but for discriminating between two models!

Section 2: Review of MinED

To set up notation...

Definition

Design $\mathbf{D} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ is a minimum energy design if it minimizes the total potential energy given by

$$\sum_{i \neq j} \frac{q(\mathbf{x}_i) q(\mathbf{x}_j)}{d(\mathbf{x}_i, \mathbf{x}_j)} \tag{14}$$

which is similar to minimizing (1) (p. 5 in Joseph 2018).

As $k \to \infty$, the criterion in (1) approaches

$$E(\mathbf{D}) = \max_{i \neq j} \frac{q(\mathbf{x}_i) q(\mathbf{x}_j)}{d(\mathbf{x}_i, \mathbf{x}_j)} \tag{15}$$

which I think is supposed to be the reciprocal of (2), even though they look slightly different.
Call $\log E(\mathbf{D})$ the definition of MinED.

Section 2: Review of Minimum Energy Designs, continued...

The Objective

Letting $q = 1/f^{1/2p}$, the objective is to find the design that minimizes

$$\max_{i \neq j} \frac{1}{f^{1/2p}(\mathbf{x}_i) f^{1/2p}(\mathbf{x}_j) d(\mathbf{x}_i, \mathbf{x}_j)}$$

or, equivalently, the design that maximizes

$$\psi(\mathbf{D}) = \min_{i \neq j} f^{1/2p}(\mathbf{x}_i) f^{1/2p}(\mathbf{x}_j) d(\mathbf{x}_i, \mathbf{x}_j) \tag{16}$$

This is great for Bayesian problems, because we only need to know f up to a constant.

Section 3: A Fast Algorithm for Generating MinED

Why do we need a faster algorithm?

Recall the one-point-at-a-time greedy algorithm in Joseph 2015, where the first design point is given by (3), and the subsequent $n-1$ design points picked one at a time by (4); equivalently, after picking $\mathbf{x_1}$ and having generated $\mathbf{x_2}, \mathbf{x_3}, \ldots, \mathbf{x_{j-1}}$ sequentially, the jth sample is given by

$$\mathbf{x}_j = \arg\max_{\mathbf{x}} \min_{i=1:(j-1)} f^{1/2p}(\mathbf{x}) f^{1/2p}(\mathbf{x}_i) d(\mathbf{x}, \mathbf{x}_i) \tag{17}$$

((5) in Joseph 2018)

This algorithm has the following problems:

1. The mode is difficult to find when f expensive to evaluate.
2. Each step in the algorithm requires a global optimization and numerous evaluations of the density f (see above).
3. Even if you approximate f with a GP at each step, fitting a global GP at every step is also time-consuming.

(Then why have a deterministic sampling method?)

Section 3: The Fast Algorithm

Generate an n-point MED $\mathbf{D}$ for f:

1. Assume $\mathbf{x} \in \chi = [0,1]^p$ (rescale, if necessary), i.e. that the support of f is in the unit hypercube.
2. Consider the annealed version of f,

$$f^{\gamma}(\mathbf{x}), \gamma \in [0,1] \tag{18}$$

   Where $\gamma = 0$ gives a uniform distribution in χ, and $\gamma = 1$ gives the target distribution, f.
3. For K steps, choose $\gamma_k = (k-1)/(K-1), k = 1, ..., K$, and at each step, generate n MinED samples. Let $\mathbf{D}_k = \{\mathbf{x}_1^k, ..., \mathbf{x}_n^k\}$ be the MinED at the kth step and f_k the corresponding density evaluations.

(p. 67 Joseph 2015)

Section 3: The Fast Algorithm, continued...

How do we generate n MinED samples at each of the K steps? Idea: Construct $\mathbf{D}_{k+1}$ by generating one new sample from each of the n samples in $\mathbf{D}_k$. This ensures that (if f is estimated by a GP $\hat{f}$?) the total number of evaluations is only Kn (otherwise, Kn^2?). A new point at each step is generated as follows: For each $\mathbf{x}_j^k \in \mathbf{D}_k$,

1. Let $L_{jk} = B(\mathbf{x}_j^k, \min_{i \neq j} |\mathbf{x}_i^k - \mathbf{x}_j^k|)$.
2. Generate new point $\mathbf{x}_j^{k+1} \in \mathbf{D}_{k+1}$ by (17), or, equivalently, using the annealed f and taking the log objective:

$$\mathbf{x}_j^{k+1} = \arg\max_{\mathbf{x} \in L_{jk}} \min_{i=1:(j-1)} \gamma_k \log f(\mathbf{x}) + \gamma_k \log f(\mathbf{x}_i^{k+1}) \\ + 2p \log d(\mathbf{x}, \mathbf{x}_i^{k+1})$$

(not exactly equivalent since we're using γ_k instead of $1/2p$)

Section 3: Computation Tricks

- Using the log version ensures numerical stability.
- But it still requires an optimization (albeit in a smaller region) at each step. Unlike the global optimization needed in (Joseph 2015), only need a local approximation of f in L_{jk}; can use cheaper approximation methods such as the limit kriging predictor (Joseph, 2006) to obtain $\hat{f}$ (one day!). In that case, the new point is given by:

  $$\mathbf{x}_j^{k+1} = \arg\max_{\mathbf{x}\in L_{jk}} \min_{i=1:(j-1)} \gamma_k \log \hat{f}^{(jk)}(\mathbf{x}) + \gamma_k \log \hat{f}^{(jk)}(\mathbf{x}_i^{k+1}) + 2p \log d(\mathbf{x}, \mathbf{x}_i^{k+1})$$

  ((7) in Joseph 2018)
  Can evaluate approximation error and re-estimate $\hat{f}^{(jk)}$.
- To make the optimization less time-consuming, use a space-filling design in L_{jk} and choose the best point according to the criterion above.
- (Included linear combinations of adjacent points in $\mathbf{D}_k$)

Section 3: What is the Candidate Set, then?

For now:

- For each $j = 1, ..., n$, the space-filling design over L_{jk} gives the n candidate points $\tilde{\mathbf{D}}^{j}_{k+1}$. We obtain the corresponding $\mathbf{x}^{k+1}_{j}$ by optimizing over the set $\mathbf{C}^{j}_{k+1} = \mathbf{C}^{j}_{k} \cup \tilde{\mathbf{D}}^{j}_{k+1}$. Hence, for each of the n design points in design $\mathbf{D}_k$, n candidate points are created from the space-filling design and combined with $\mathbf{C}^{j}_{1}$ for a total of $2n$ candidate points for $\mathbf{x}^{k}_{j}$ to be picked from.
- For $k = 1$, $\mathbf{C}^{j}_{1} = \mathbf{D}_1$, where $\mathbf{D}_1$ is from a space-filling design over the support $[0, 1]^p$.
- At each step $k = 2, ..., K$, the first design point is $\mathbf{x}^{k}_{1} = \arg\max_{\mathbf{x} \in \mathbf{C}_{k+1}} \log f(\mathbf{x})$, the candidates for the next $j = 2, ..., n$ design points are generated as described above, and finally the design point $\mathbf{x}^{k+1}_{j}$ is picked to satisfy:

$$\mathbf{x}^{k+1}_{j} = \arg\max_{\mathbf{x} \in \mathbf{C}^{j}_{k+1}} \min_{i=1:(j-1)} \gamma_k \log f(\mathbf{x}) + \gamma_k \log f(\mathbf{x}^{k+1}_{i}) + 2p \log d(\mathbf{x}, \mathbf{x}^{k+1}_{i}) \quad (19)$$

Section 3: What is the Candidate Set, then? continued...

Just to further illustrate this point (because I *will* be confused again):

- For $k = 1$, $\mathbf{C}_1 = \text{Lattice}([0,1]^p, N)$ and the resulting design is $\mathbf{D}_1$.
- For $k = 2$ and $j = 1, ..., N$, $\mathbf{C}_2^j = \mathbf{C}_1 \cup \text{Lattice}(L_{j2}, N)$ $= \text{Lattice}([0,1]^p, N) \cup \text{Lattice}(L_{j2}, N)$
- For $k = 3$ and $j = 1, ..., N$, $\mathbf{C}_3^j = \mathbf{C}_2^j \cup \text{Lattice}(L_{j3}, N)$ $= \text{Lattice}([0,1]^p, N) \cup \text{Lattice}(L_{j2}, N) \cup \text{Lattice}(L_{j3}, N)$ $= \text{Lattice}([0,1]^p, N) \cup_{m=1}^{3} \text{Lattice}(L_{jm}, N)$
- ...
- Hence, for $k = 2, ..., K$ and $j = 1, ..., N$, $\mathbf{C}_k^j = \text{Lattice}([0,1]^p, N) \cup_{m=1}^{k} \text{Lattice}(L_{jm}, N)$

A Greedy Algorithm for Linear Model Selection

Generate an N-point MED $\mathbf{D}_N$ for f:

1. Sample β_i from the prior on each of the two models: $\beta_i \sim N(\tilde{\beta}_i, \sigma^2_{\beta_i})$
2. Obtain N candidate points, $\mathbf{x}$, in $[0,1]$ using a space-filling design. These points will be $\tilde{\mathbf{D}}_1$.
3. For $k = 1, ..., K$, and for $n = 1, ..., N$,
   Determine L_{jk} and obtain n candidate points, $\tilde{\mathbf{D}}_1$, in L_{jk} using a space-filling design. If $k = 1$, $\mathbf{C}_1 = \tilde{\mathbf{D}}_1$; if $k > 1$, $\mathbf{C}_k = \mathbf{C}_{k-1} \cup \tilde{\mathbf{D}}_k$
   Find the design point $\mathbf{x}_n^k$ that minimizes

$$\max_{i \neq j} \frac{1}{f^{\gamma_k}(\mathbf{x}_i) f^{\gamma_k}(\mathbf{x}_j) d(\mathbf{x}_i, \mathbf{x}_j)} \tag{20}$$

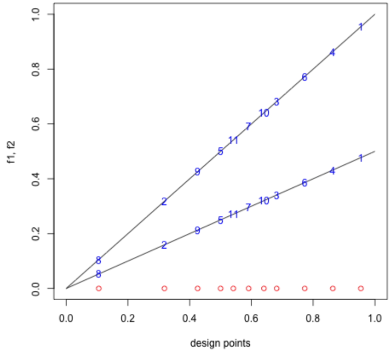
   where $\gamma_k = (k-1)/(K-1)$.
   Method similar to Greedy Algorithm:

$$\mathbf{x}_n^k = \arg\min_{\mathbf{x} \in \mathbf{C}_k} \max_{i \neq j} \frac{1}{f^{\gamma_k}(\mathbf{x}_i) f^{\gamma_k}(\mathbf{x}) d(\mathbf{x}_i, \mathbf{x})} \tag{21}$$

Results

The 2 Models

- $f_0(\mathbf{x}) = \mathbf{x}\beta_0$ with $\tilde{\beta}_0 = 1$
- $f_1(\mathbf{x}) = \mathbf{x}\beta_1$ with $\tilde{\beta}_1 = 1/2$
- $\sigma_{\epsilon_0} = \sigma_{\epsilon_1}$
- $K = 16$



Observations

- Results are comparable to those based on Joseph 2015