MED for Model Selection

Kristyn Pantoja

5 September 2019

MED Overview

Minimum Energy Design

Minimum energy design (MED) is a deterministic sampling method which makes use of evaluations of the target distribution f to obtain a weighted space-filling design.

Definition:

Design $\mathbf{D} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ is a MED if it minimizes the total potential energy, given by:

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

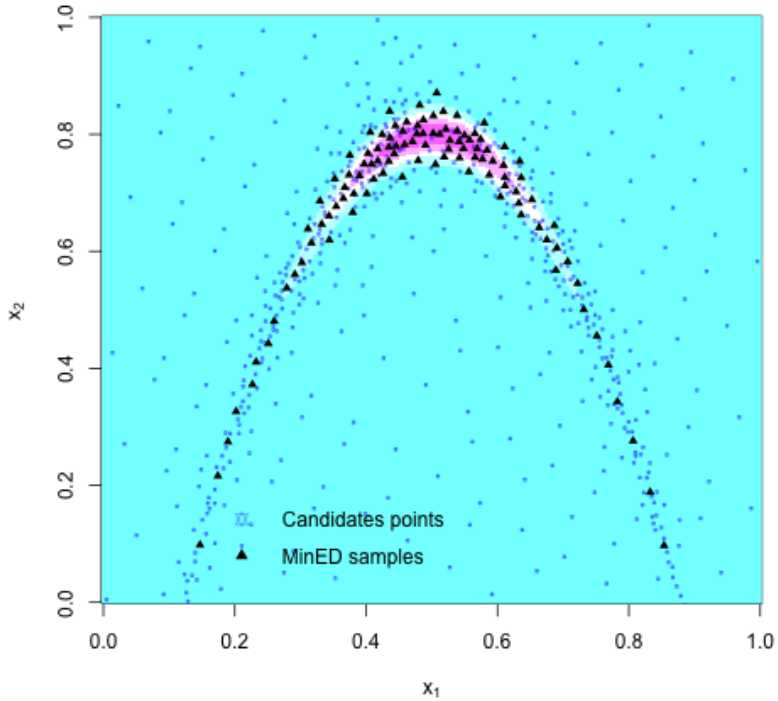
If $q = \frac{1}{f^{1/2p}}$, the **limiting distribution**[1] of the design points is target distribution, f.

[1] "Sequential Exploration of Complex Surfaces Using Minimum Energy Designs," Joseph et. al. 2015, Result 1

Advantages of MED

Sampling the "Banana" Function:
MED gen. by "fast"[2]algorithm:

- $N = 109$
- $K = 6$
- $NK = 654$ evaluations of f



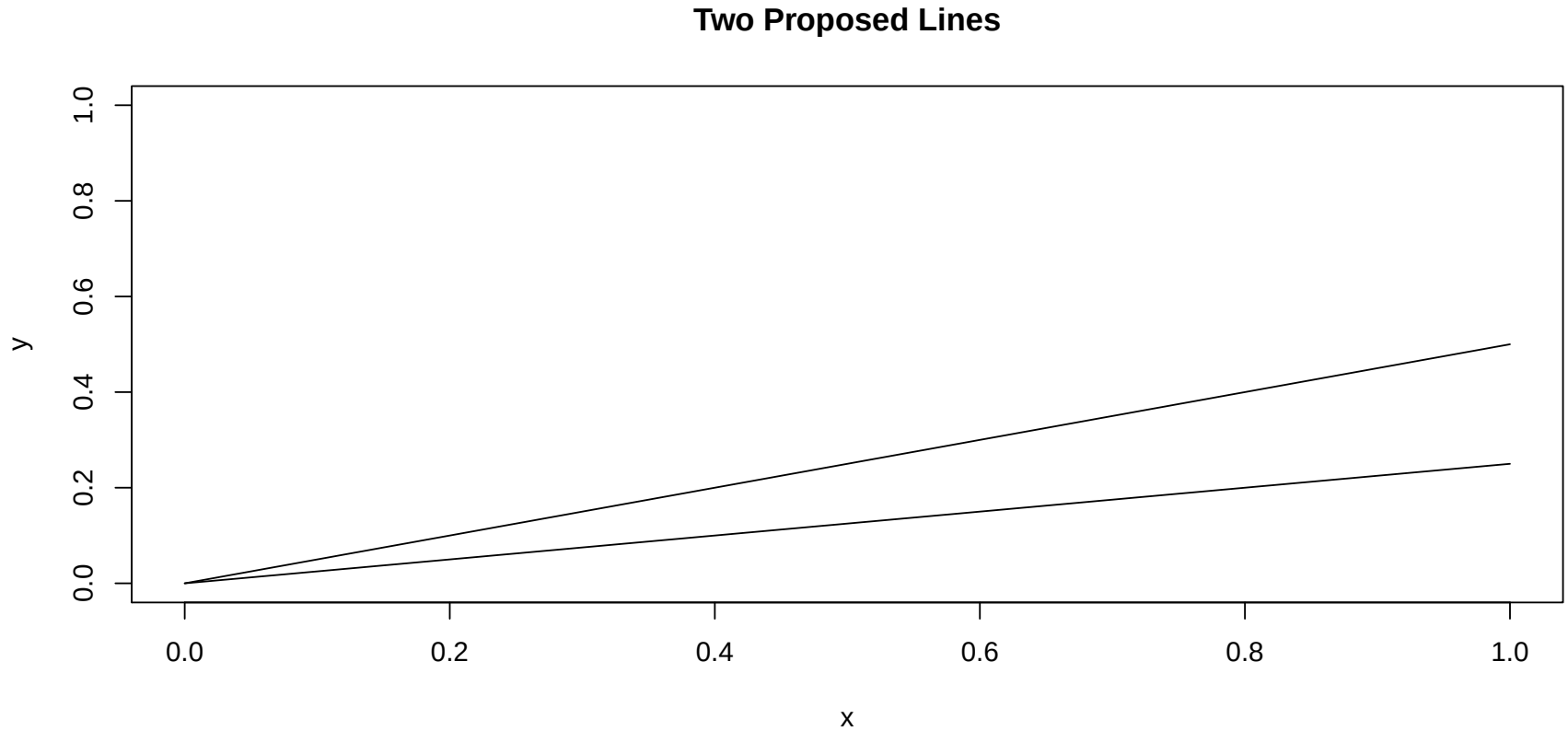
Compared to other sampling methods, MED:

- has fewer points and hence requires fewer evaluations of f (unlike MCMC)
- concentrates in high-density regions (unlike QMC)

[2] "Deterministic Sampling of Expensive Posteriors Using Minimum Energy Designs" Joseph et. al. 2018

MED for Model Selection

Design an Experiment that Estimates Slope



- Goal: A design $\mathbf{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ to gather data that will
  1. help distinguish these two slopes
  2. allow adequate estimation of β
- Modify the Minimum Energy Design: $f(x)$ as normalized Wasserstein distance between $y|H_0, x$ and $y|H_1, x$
  - Assumes a bounded design space.

MED Algorithm Criteria

We consider two algorithms[3] which both seek to minimize the Total Potential Energy of the points in MED $\mathbf{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$:

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

1. One-at-a-Time Algorithm (Joseph et. al. 2015): minimize

$$\left\{ \sum_{i \neq j} \left(\frac{q(\mathbf{x}_i) q(\mathbf{x}_j)}{d(\mathbf{x}_i, \mathbf{x}_j)} \right)^k \right\}^{1/k}$$

   which gives the Total Potential Energy when $k = 1$

2. Fast Algorithm(Joseph et. al. 2018): minimize

$$\max_{i \neq j} \frac{q^{\gamma_s}(\mathbf{x}_i) q^{\gamma_s}(\mathbf{x}_j)}{d(\mathbf{x}_i, \mathbf{x}_j)}$$

   using tempering/annealing technique on q over S stages

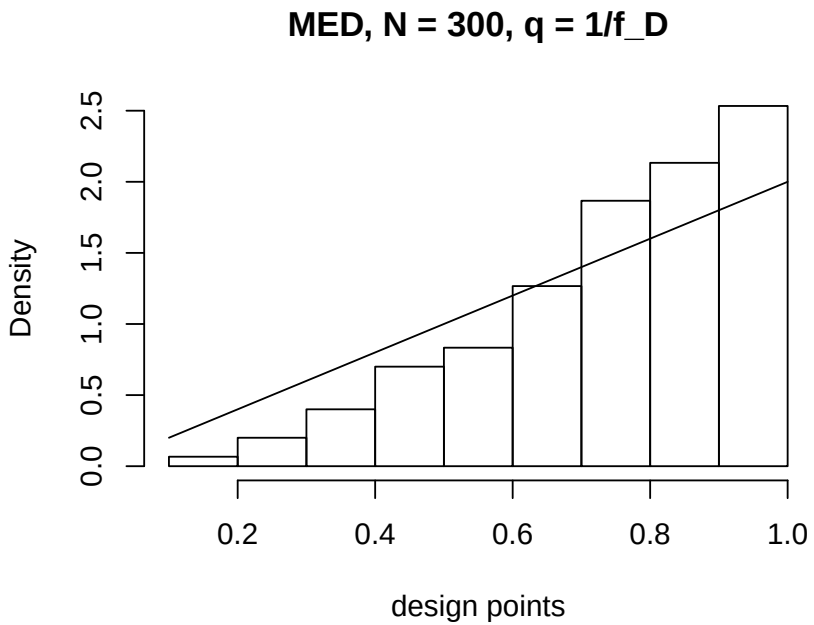
[3] See Appendix A for details about their implementation

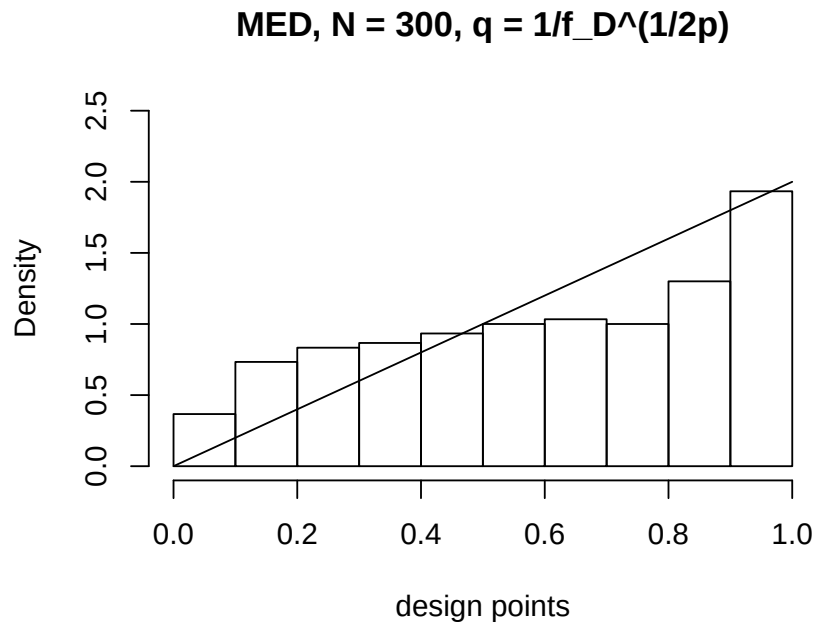
Tuning Parameter

Introducing tuning parameter α in the charge function q:

$$q = \frac{1}{f_D^{\alpha/2p}}$$

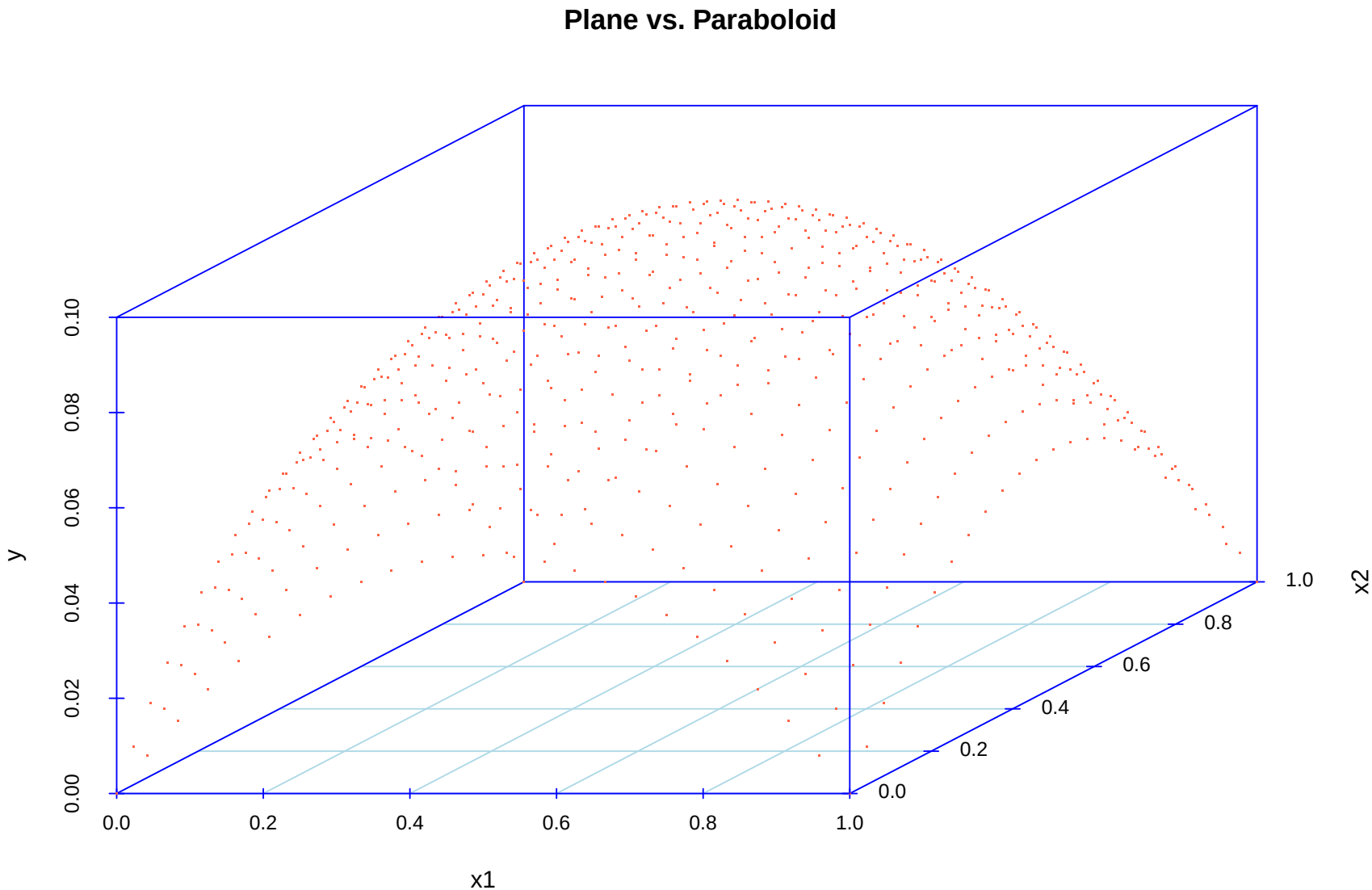
- $\alpha = 2p$
- $f_D(\mathbf{x}) = \text{Wasserstein}(\phi_{0,\mathbf{x}}, \phi_{1,\mathbf{x}})$
- Results in a design that is more sensitive to smaller values of f_D, i.e. hypothesized models that are closer together.
- This can be interpreted as changing the distance measure.





MED for Comparing Linear and Quadratic Model

Linear Model vs. Quadratic Model in 2 Dimensions



Linear Model vs. Quadratic Model in 2 Dimensions

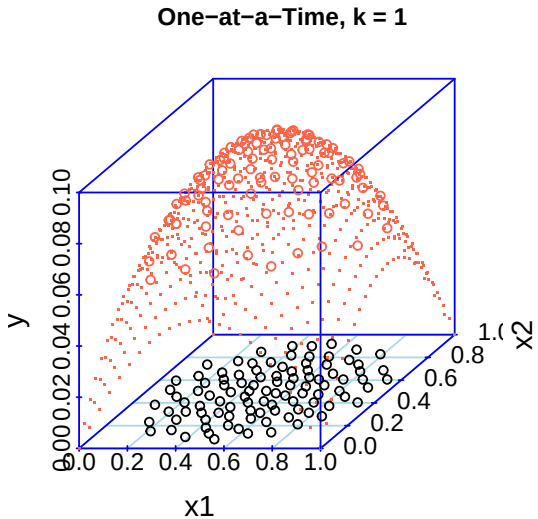
Compare the linear model with two factors $y = \beta_0 + x_1\beta_1 + x_2\beta_2 + \varepsilon_i$ with the quadratic model without cross-terms $y = \beta_0 + x_1\beta_1 + +x_1^2\beta_2 + x_2\beta_3 + +x_2^2\beta_4 + \varepsilon_i$

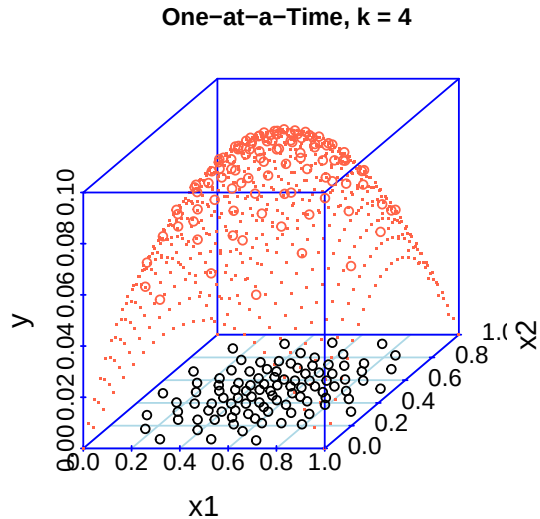
- $y|\beta, X \sim N(X\beta, \sigma^2 I)$
- $y|X, H_k \sim N(X\mu_k, \sigma_\varepsilon^2 I + XV_kX^T)$, $k = 0, 1$

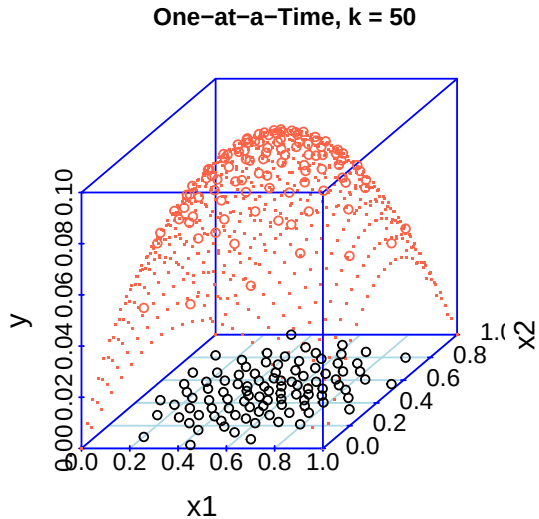
Hypotheses

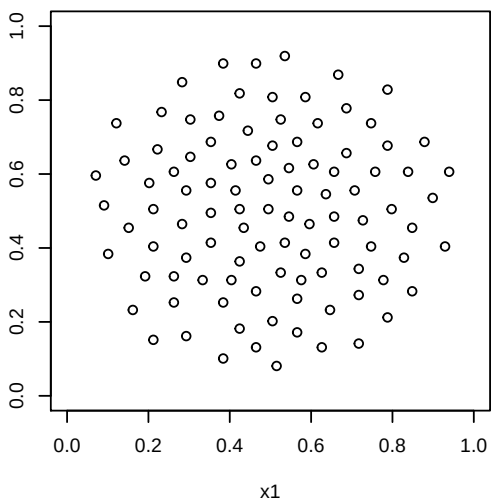
$$
\begin{aligned}
H_0 : \beta &\sim N(\mu_0, V_0), \\
\mu_0 &= (\mu_{00}, \mu_{01}, \mu_{02})^T, \\
V_0 &= \text{diag}(\nu_{00}^2, \nu_{01}^2, \nu_{02}^2) \\
H_1 : \beta &\sim N(\mu_1, V_1), \\
\mu_0 &= (\mu_{10}, \mu_{11}, \mu_{12}, \mu_{13}, \mu_{14})^T, \\
V_1 &= \text{diag}(\nu_{10}^2, \nu_{11}^2, \nu_{12}^2, \nu_{13}^2, \nu_{14}^2)
\end{aligned}
$$

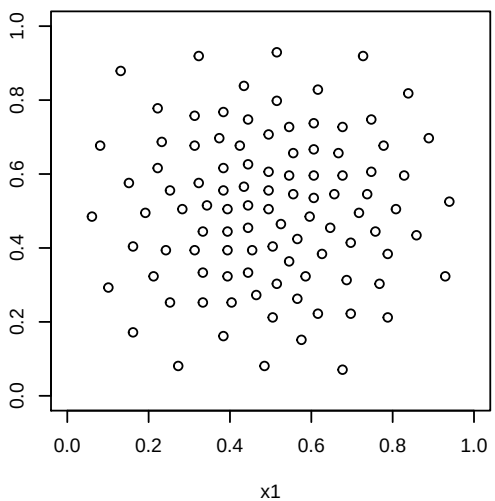
One-at-a-Time Algorithm, k = 1, 4, 50 and N = 100

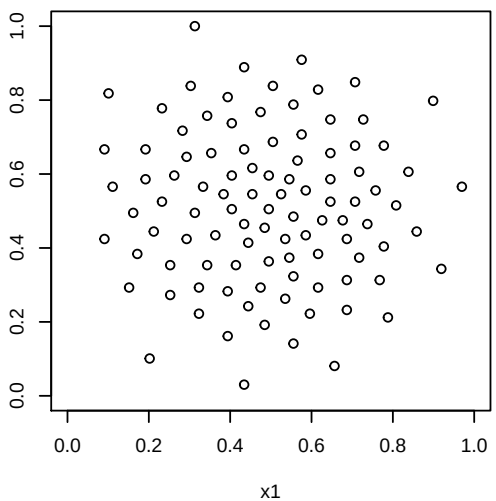




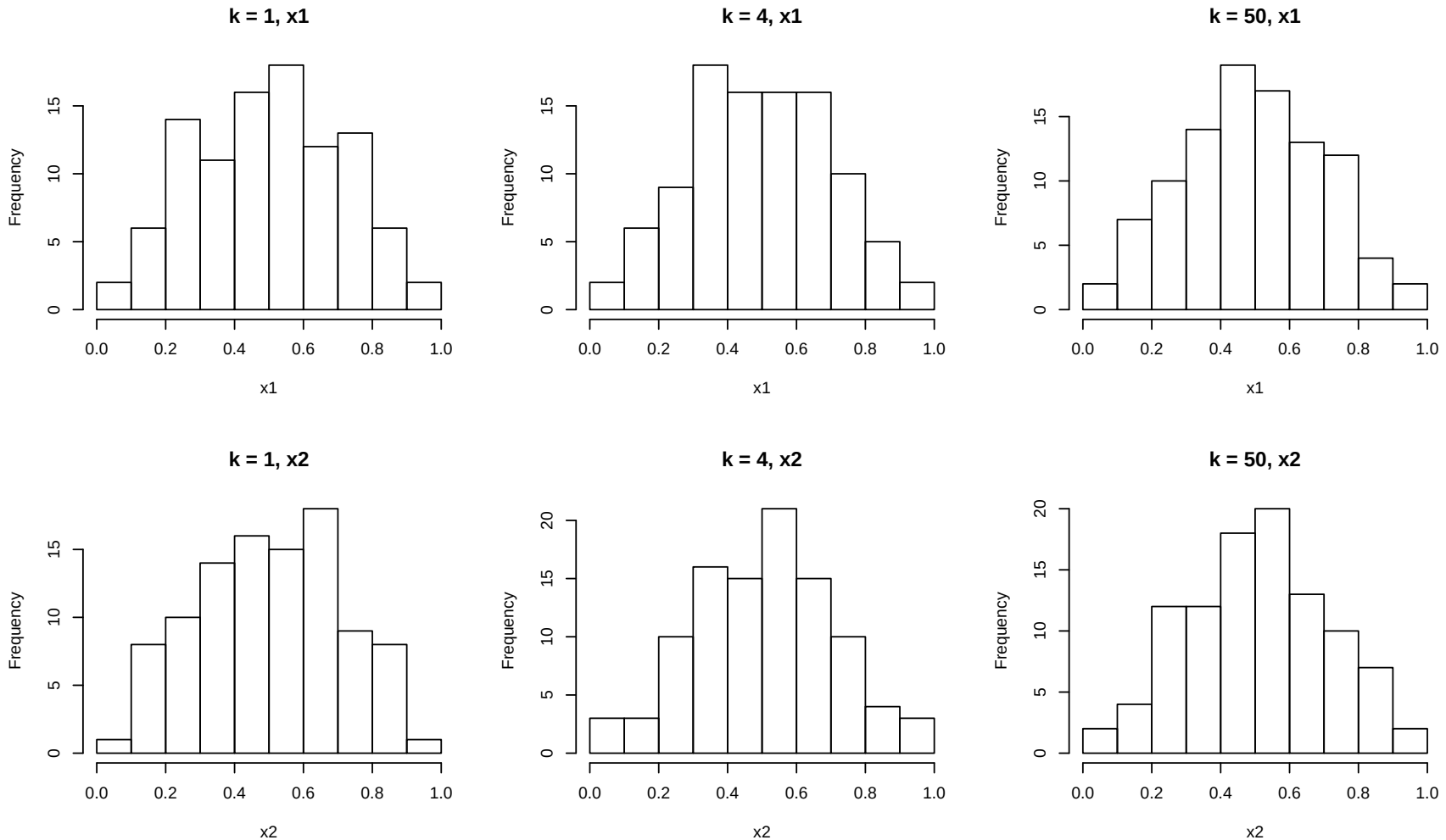




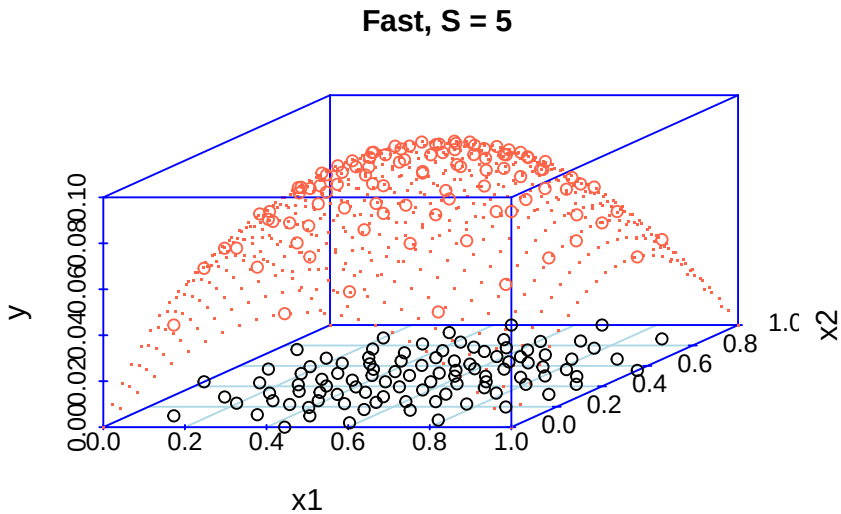


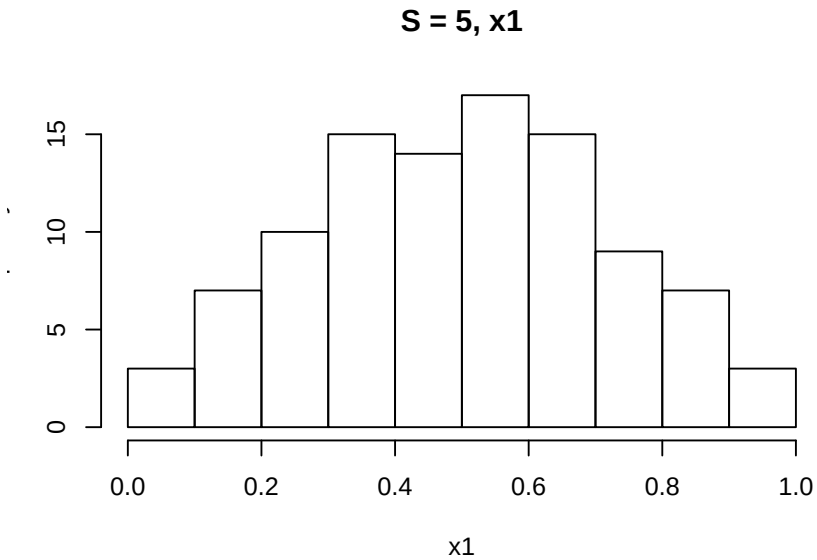


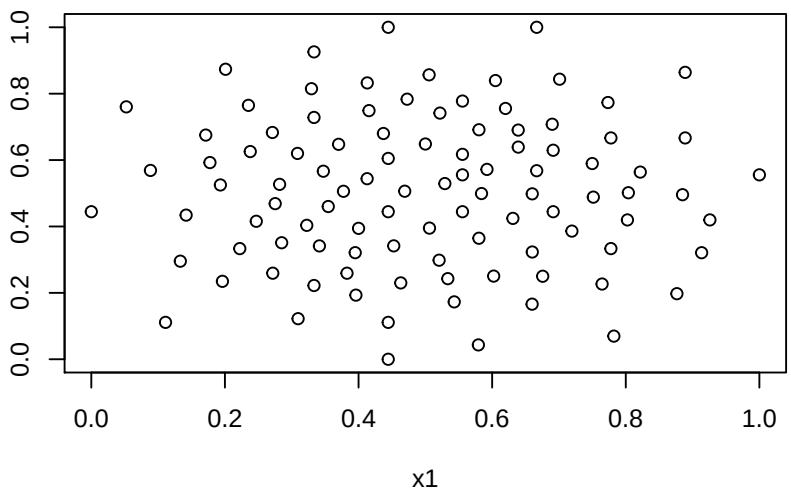
One-at-a-Time Algorithm, k = 1, 4, 50 and N = 100

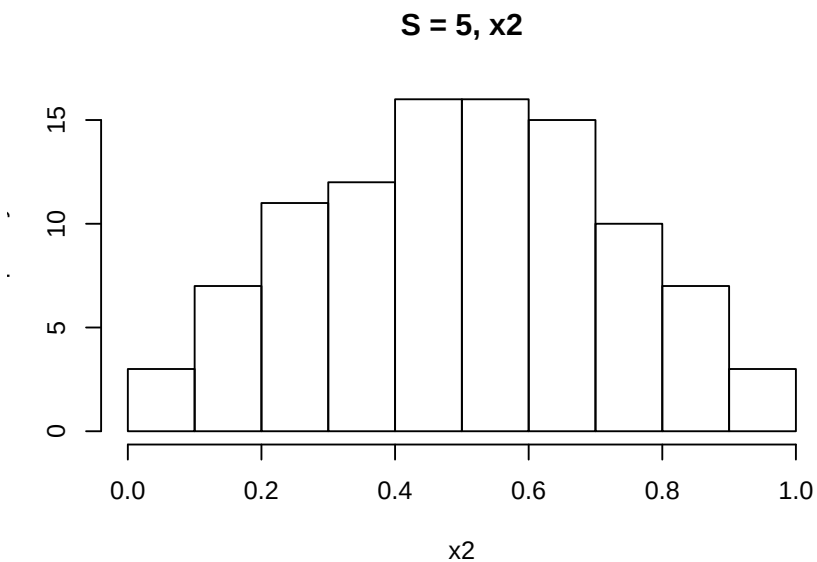


Fast Algorithm, S = 5, N = 100





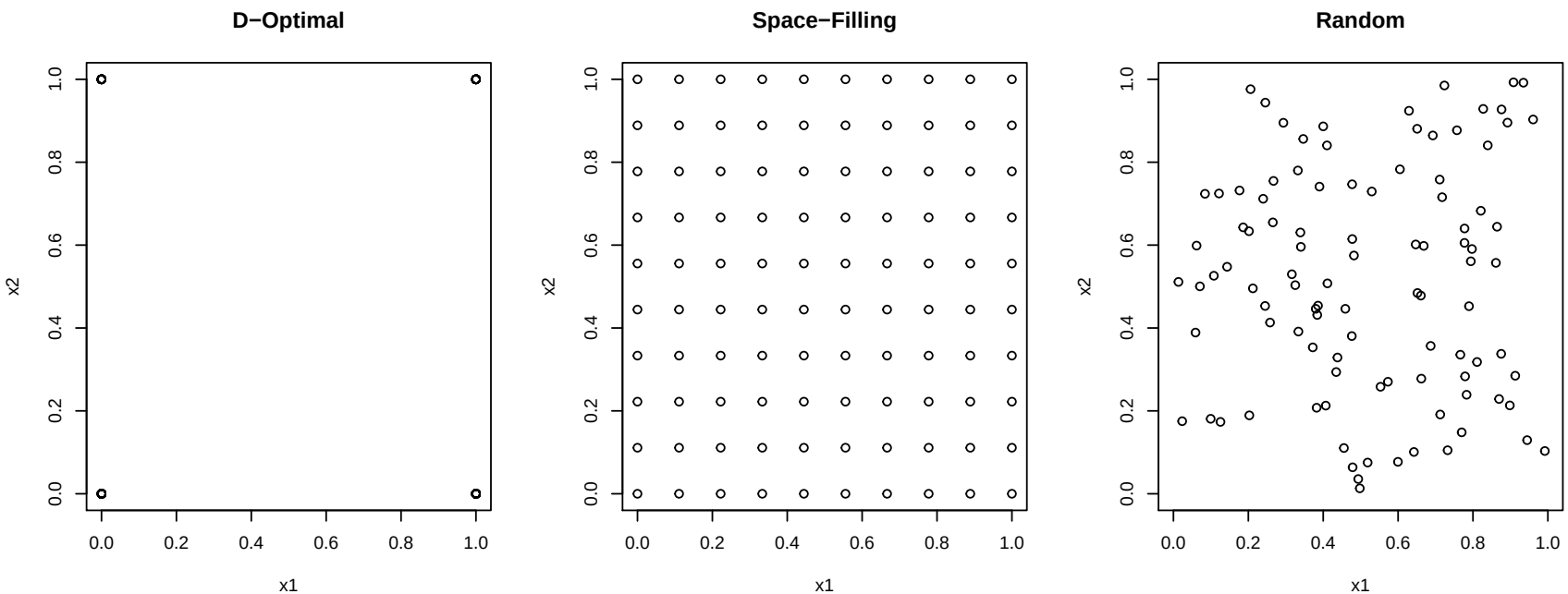




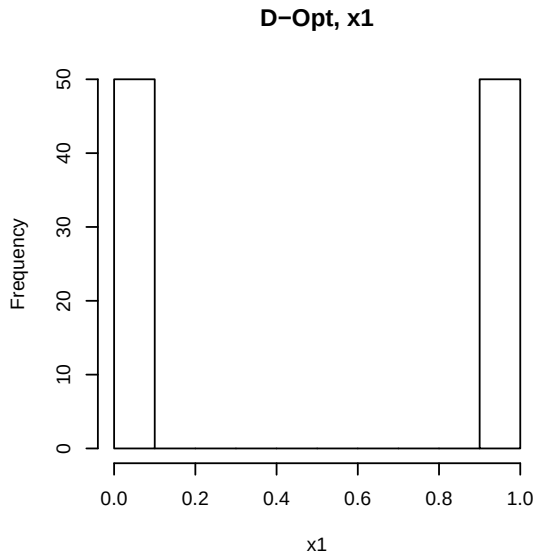
Other Designs

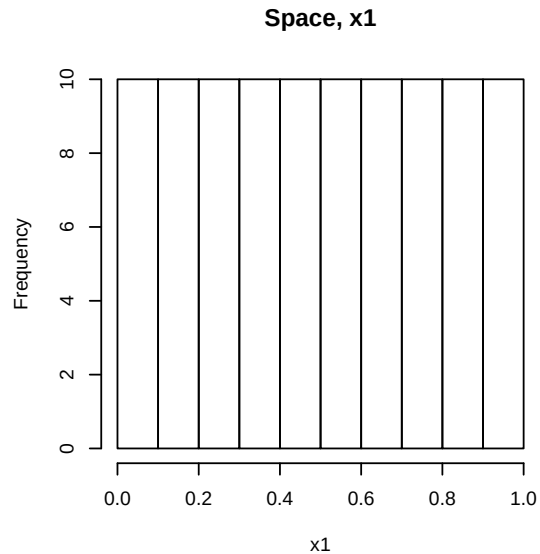
We would like to evaluate MED against other popular designs:

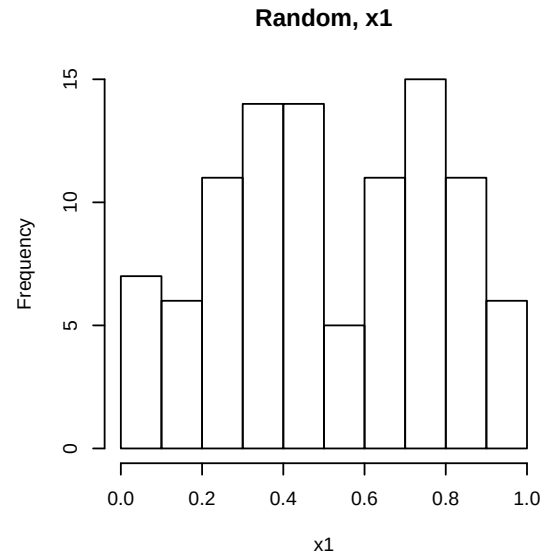
- D-Optimal Design
  - best for parameter estimation if model is correctly specified
  - not robst to model misspecification
- Space-filling Design: evenly spaced points over $[0, 1]$
  - good for response estimation
  - robust to model misspecification
- Random Design: $(\mathbf{x} \sim U([0,1]^p), \forall \mathbf{x} \in \mathbf{D})$

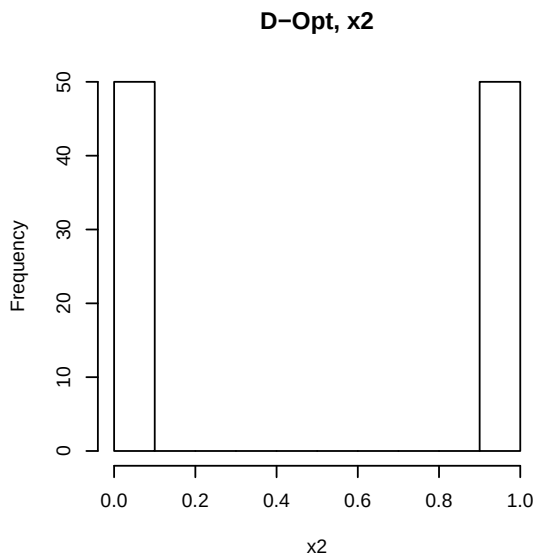


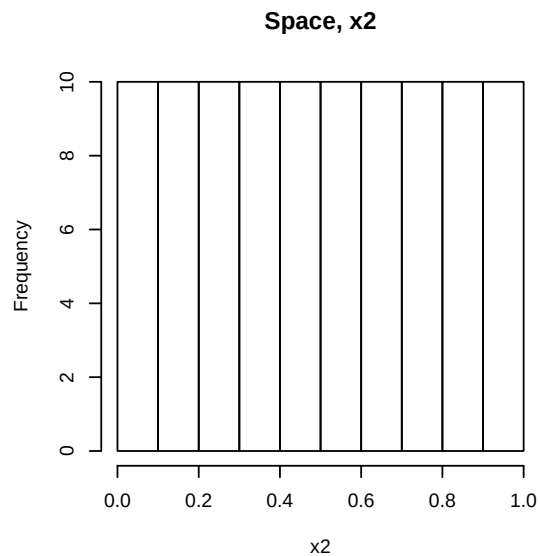
Other Designs

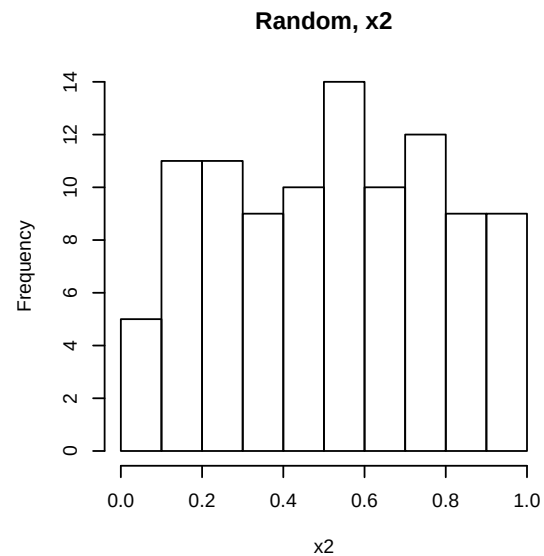












Evaluating the Designs

- Posterior Variance[4], i.e. $Var[\beta|y, X, H_k] = \sigma^2(X^T X + \sigma^2 V_k^{-1})^{-1}$
- MSE of the Posterior Mean (MC estimates)
- Expected Posterior Probabilities of Hypotheses & Bayes Factor (MC estimates)
- Design Criteria:
  - Total Potential Energy
  - Criterion for One-at-a-Time Algorithm
  - Criterion for Fast Algorithm

Interpretations

- A design that is better for estimating β may have smaller posterior variance and MSE for the posterior mean.
- A design that is better for hypothesis testing may give a larger expected posterior probability to the true model from simulated responses.

[4] See Appendix B for details on their calculations

Evaluations

- Evaluating algorithm criteria:

	1atT,k=1	1atT,k=4	1atT,k=50	Fast,S=5	D-Opt	Space	Random
TPEx10e-3	2593.9	2635.6	2633.9	2647.1	Inf	Inf	3765
Fast Crit	3099.4	2412.6	2327.8	2726.8	Inf	Inf	150520
1atT Crit (k=4)	8016.4	7548.1	7626.4	7852.7	Inf	Inf	150750

- Evaluating ability to distinguish hypotheses:

	1atT,k=1	1atT,k=4	1atT,k=50	Fast,S=5	D-Opt	Space	Random
E[P(H0|Y,D)|H0,D]	0.77	0.779	0.794	0.838	0.536	0.933	0.874
E[P(H1|Y,D)|H0,D]	0.23	0.221	0.206	0.162	0.464	0.0673	0.126
E[BF01 | H0,D]	44.5	66.8	54.9	117	1.28	3190	879
E[P(H0|Y,D)|H1,D]	0.212	0.185	0.185	0.164	0.445	0.0571	0.114
E[P(H1|Y,D)|H1,D]	0.788	0.815	0.815	0.836	0.555	0.943	0.886
E[BF01|H1,D]	0.986	0.64	0.883	0.929	0.983	0.801	0.935

Evaluations, continued

- Evaluating ability to estimate parameters $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4)^T$

	1atT,k=1	1atT,k=4	1atT,k=50	Fast,S=5	D-Opt	Space	Random
V[B0|Y,X,H1]	0.000508	0.000532	0.000542	0.000478	0.000144	0.000294	0.00042
V[B1|Y,X,H1]	0.00244	0.00246	0.00245	0.00238	0.00255	0.00194	0.00225
V[B2|Y,X,H1]	0.00239	0.00244	0.00246	0.0023	0.00255	0.00183	0.00214
V[B3|Y,X,H1]	0.00246	0.00249	0.00249	0.00235	0.00255	0.00194	0.00224
V[B4|Y,X,H1]	0.00246	0.00244	0.00238	0.00227	0.00255	0.00183	0.00207

- MSE of $E[\beta|Y, X, H_0]$ where the prior mean is $\mu = \mu_0$ and the true mean is $\beta_T = \mu_1$.

	1atT,k=1	1atT,k=4	1atT,k=50	Fast,S=5	D-Opt	Space	Random
MSE B0	0.00449	0.00447	0.00446	0.00452	0.00486	0.00471	0.00458
MSE B1	0.0426	0.0425	0.0425	0.0426	0.0425	0.0431	0.0427
MSE B2	0.0426	0.0426	0.0425	0.0427	0.0425	0.0432	0.0429
MSE B3	0.0425	0.0425	0.0425	0.0427	0.0425	0.0431	0.0428
MSE B4	0.0425	0.0426	0.0426	0.0427	0.0425	0.0432	0.0429

Other MED Applications: Gaussian Process Model Selection

Applying MED to Gaussian Process Model Selection

- When there are two Gaussian Process Models that can be used to estimate a function, e.g. Matern vs. Squared Exponential covariance functions[5]
  - Squared Exponential: infinitely differentiable, standard choice
  - Matern: more reasonable smoothness assumptions

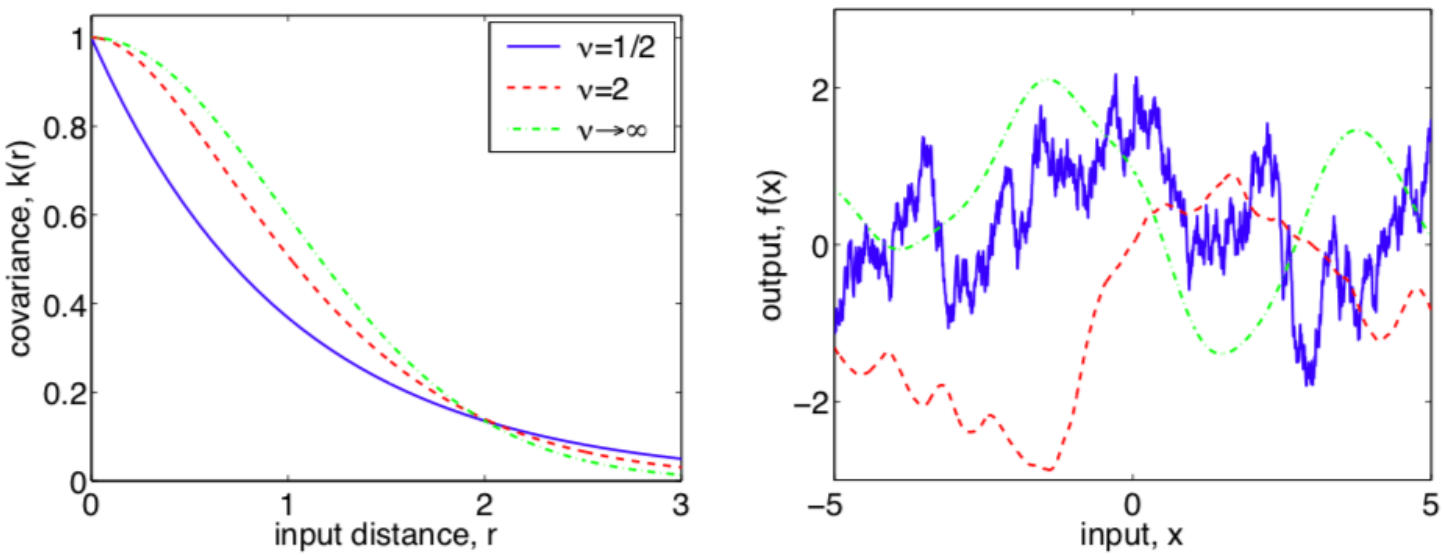


Figure 4.1: Panel (a): covariance functions, and (b): random functions drawn from Gaussian processes with Matérn covariance functions, eq. (4.14), for different values of ν, with $\ell = 1$. The sample functions on the right were obtained using a discretization of the x-axis of 2000 equally-spaced points.

[5] "Gaussian Processes for Machine Learning" Rasmussen et. al. 2005

Applying MED to Gaussian Process Model Selection

- Goal: Choose a design that will distinguish the two gaussian process models.
- Distinguishing functions vs. distributions over functions:
  - For regression models, we use $f_D(\mathbf{x}) = \text{Wasserstein}(\phi_{0,\mathbf{x}}, \phi_{1,\mathbf{x}})$. What is the distance function now? What are $\phi_{0,\mathbf{x}}, \phi_{0,\mathbf{x}}$?
  - Key Question: Do we need to consider the predictive distribution for each GP model?
    - Doing so would give us an option for $\phi_{0,\mathbf{x}}, \phi_{0,\mathbf{x}}$.
    - However, we will need data (and possibly need to choose new points one at a time).

One-at-a-Time Algorithm (2015) Review

Steps to obtain MED using One-at-a-Time algorithm:

1. Obtain *numCandidates* candidate points, $\mathbf{x}$, in $[0, 1]$ to form candidate set C.
2. Initialize D_N by choosing $\mathbf{x}_1$ to be the candidate $\mathbf{x}$ which optimizes f, where $f(\mathbf{x}) = \text{Wasserstein}(\phi_{0,\mathbf{x}}, \phi_{1,\mathbf{x}})$ and

$$\phi_{0,\mathbf{x}} = N(\mu_0 \mathbf{x}, \sigma_0^2 + \mathbf{x}^2 \nu_0^2),$$
$$\phi_{1,\mathbf{x}} = N(\mu_1 \mathbf{x}, \sigma_1^2 + \mathbf{x}^2 \nu_1^2)$$

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

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

where $q = 1/f^{(1/2p)}$, $d(x, y)$ is Euclidean distance and $k = 4p$.

One-at-a-Time Algorithm for GP?

Suppose you have training data X, Y.

1. Obtain candidate set C
2. Initialize $\mathbf{D}$ as the candidate point $\mathbf{x}_*$ that maximizes $f(\mathbf{x}) = \text{Wasserstein}(\phi_{0,\mathbf{x}}, \phi_{1,\mathbf{x}})$, where, here, $\phi_{\ell,\mathbf{x}}$ is the predictive distribution $f_*|\mathbf{x}_*, X, f \sim N(k_*^T(K + \sigma_e^2 I)^{-1} Y, k(\mathbf{x}, \mathbf{x}) - k_*^T(K + \sigma_e^2)^{-1} k_*)$, where $k_* = k(\mathbf{x}, X), K = K(X, X)$, and k and K are determined by the hypothesis ℓ.
3. For subsequent design points, choose:

$$\mathbf{x}_{j+1} = \arg\min_{\mathbf{x} \in C} \sum_{i=1}^{j} \left(\frac{q(\mathbf{x}_i) q(\mathbf{x})}{d(\mathbf{x}_i, \mathbf{x})} \right)^k + \sum_{x_i \in X} \left(\frac{q(\mathbf{x}_i) q(\mathbf{x})}{d(\mathbf{x}_i, \mathbf{x})} \right)^k$$

What is the data for previously added design points, $\{(\mathbf{x}_i) | i = 1 : j\}$?

Issues

- When a candidate point $\mathbf{x}_*$ are too close to the design points, correlation is too high, which leads to standard deviation for posterior predictive distribution $y_*|\mathbf{x}_*, X, Y$ to be too close to zero (computational issues).
  - how do we know? this is indicated by fewer invalid standard deviations when parameter l is smaller
  - a fix: add noise
- After we choose one new design point, how do we update the posterior predictive distribution when choosing another design point?
  - a fix: andomly choose a posterior predictive mean (from H_0 or H_1 each w.p. $1/2$)

Other MED Applications: Updating a Design

Updating a Design

What do we do about initializing the first point, considering the design has already been initialized?

- Check if the optimal point (that which maximizes the Wasserstein distance between f_0 and f_1) is in the set and proceed accordingly
- Don't initialize with the optimal point: skip straight to choosing the next design point based on the algorithm's criterion to approach minimizing total potential energy.

What do we do when initial design points cause TPE to be infinity?

- Leave them out
- Add a buffer to the difference in means in the Wasserstein distance calculation so that it does not evaluate to 0, which is what causes this problem since $f = W$ and $q = 1/f^{(1/2p)}$.

Appendix A: MED Algorithms

One-at-a-Time Algorithm (2015)

Steps to obtain MED using One-at-a-Time algorithm:

1. Obtain *numCandidates* candidate points, $\mathbf{x}$, in $[0, 1]$.
2. Initialize $\mathbf{D}_N$ by choosing $\mathbf{x}_1$ to be the candidate $\mathbf{x}$ which optimizes f, where $f(\mathbf{x}) = \text{Wasserstein}(\phi_{0,\mathbf{x}}, \phi_{1,\mathbf{x}})$ and

$$\phi_{0,\mathbf{x}} = N(\mu_0 \mathbf{x}, \sigma_0^2 + \mathbf{x}^2 \nu_0^2),$$
$$\phi_{1,\mathbf{x}} = N(\mu_1 \mathbf{x}, \sigma_1^2 + \mathbf{x}^2 \nu_1^2)$$

3. For $j = 1, \ldots, N$, choose the next point $\mathbf{x}_{j+1}$ by:

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

where $q = 1/f^{(1/2p)}$, $d(x, y)$ is Euclidean distance and $k = 4p$.

- This is a greedy algorithm for choosing points one at a time

Fast Algorithm (2018)

In each of S stages, create a new design to iteratively minimize

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

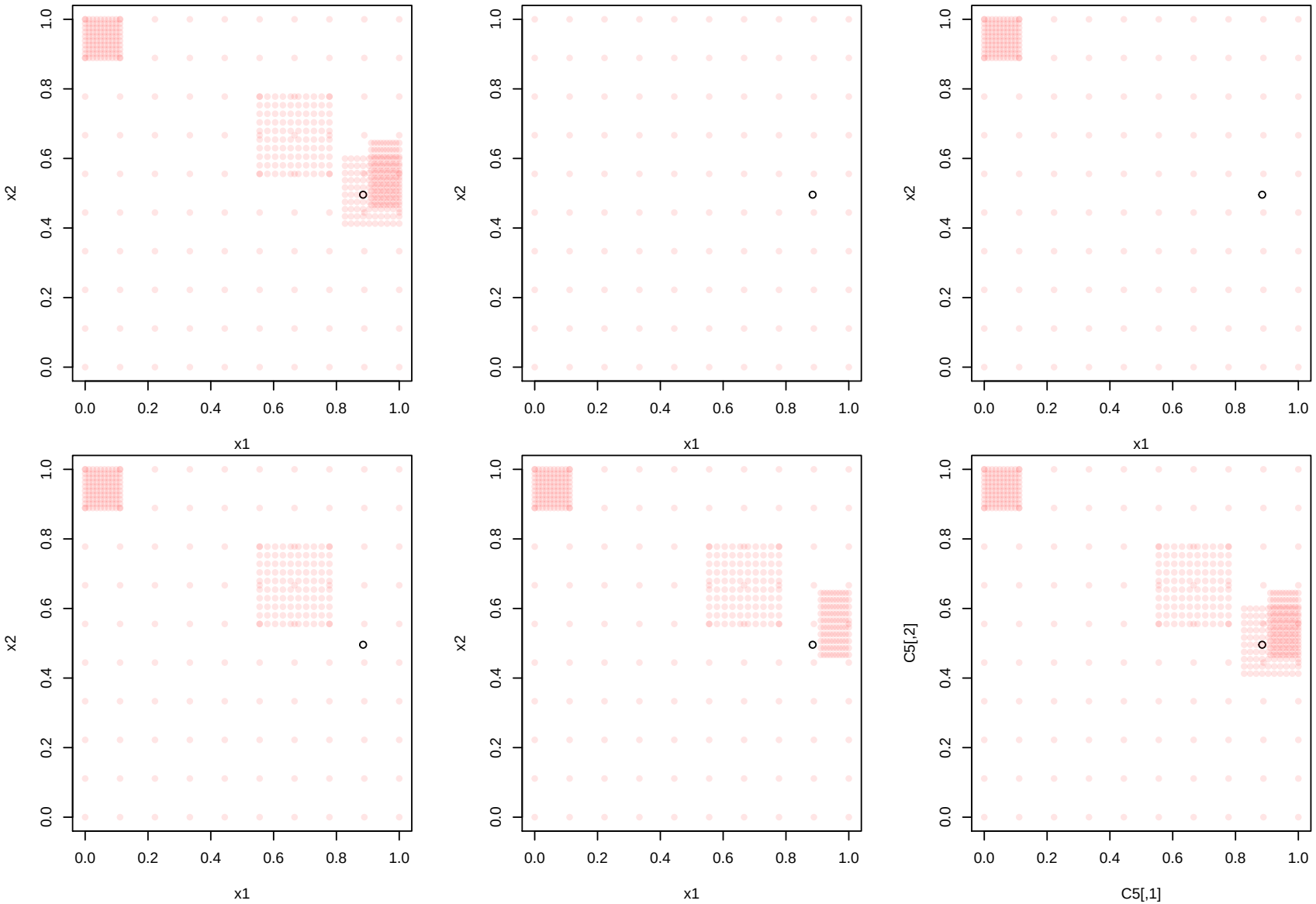
1. Initialize space-filling design $\mathbf{D}_1 = \{\mathbf{x}_1^{(1)} \dots \mathbf{x}_N^{(1)}\}$
2. For $s = 1, \dots, S-1$ stages, obtain each design point $\mathbf{x}_j^{(s+1)} \in \mathbf{D}_{s+1}$ by:

$$\begin{aligned} \mathbf{x}_j^{s+1} &= \arg\min_{\mathbf{x} \in \mathbf{C}_j^{s+1}} \max_{i=1:(j-1)} \frac{1}{f^{\gamma_s}(\mathbf{x}_i) f^{\gamma_s}(\mathbf{x}) d^{(2p)}(\mathbf{x}_i, \mathbf{x})} \\ &= \arg\min_{\mathbf{x} \in \mathbf{C}_j^{s+1}} \max_{i=1:(j-1)} \frac{q^{\gamma_s}(\mathbf{x}_i) q^{\gamma_s}(\mathbf{x})}{d(\mathbf{x}_i, \mathbf{x})} \end{aligned}$$

   where $\gamma_s = s/(S-1)$ and $\mathbf{C}_j^{s+1}$ is the candidate set for $\mathbf{x}_j^{(s+1)}$

- Points migrate to more optimal locations in each stage

Candidates for Design Point indexed at 10



Appendix B: Evaluations

Posterior Variance

In the Bayesian linear regression framework,

$$y|\beta, X \sim N(X\beta + \sigma^2 I)$$
$$\beta \sim N(\mu, V)$$

with $X \in \mathbb{R}^{N \times p}, \beta \in \mathbb{R}^p, V \in \mathbb{R}^{p \times p}$,

- $\hat{\beta} = \frac{1}{\sigma^2}\Sigma_B(X^T y + \sigma^2 V^{-1}\mu)$ with posterior distribution

$$\beta|y, X \sim N\left(m_B, \Sigma_B\right)$$

where

$$\Sigma_B = \sigma^2(X^T X + \sigma^2 V^{-1})^{-1}$$
$$m_B = \frac{1}{\sigma^2}\Sigma_B(X^T y + \sigma^2 V^{-1}\mu)$$

- Σ_B can be used to evaluate a design **D**'s ability to estimate β

Posterior Probabilities of Hypotheses

- Posterior Probability of model $H_\ell, \ell \in 1, ..., M$:

$$P(H_\ell|y, X) = \frac{\pi_\ell f(y|H_\ell, X)}{\sum_{m=1}^{M} \pi_m f(y|H_m, X)}$$

  where π_m is the prior on H_m (typically $\pi_m = \frac{1}{M}$), and $f(y|H_m, X)$ is the model evidence.
- $P(H_\ell|y, X)$ tells which hypothesis is more likely to give the correct model.
- $E[P(H_\ell|y, X)|H_r, X]$ may be estimated using MC approximation from simulated responses y under a chosen hypothesis H_r.
- $E[P(H_\ell|y, \mathbf{D})|H_r, \mathbf{D}]$ can be used to evaluate a design $\mathbf{D}$'s ability to distinguish hypotheses

Estimate Expected Posterior Probability of a Hypothesis

Estimate the expected posterior probability of hypothesis H_ℓ for J simulations of Y under H_r, given design $\mathbf{D} = \{x_1, ..., x_N\}$:

1. For $j = 1, \ldots, J$:
   - 1.1 Draw $\beta \sim N(\mu_r, \nu_r^2)$
   - 1.2 Draw $y_i^{(j)}|\mathbf{D} \sim N(\mathbf{x}_i\beta, \sigma_r^2)$, $\forall \mathbf{x}_i \in \mathbf{D}$
   - 1.3 $\forall m \in \{1, ..., M\}$, calculate model evidences $f(y^{(j)}|H_m, \mathbf{D})$
     - $f(y|H_m, \mathbf{D})$ is the marginal likelihood $N(\mathbf{D}\mu_m, \sigma^2 I + \mathbf{D}V\mathbf{D}^T)$ evaluated at y and $\mathbf{D}$.
   - 1.4 Calculate the posterior probability of H_ℓ, $P(H_\ell|y^{(j)}, \mathbf{D})$, from simulation j

$$P(H_\ell|y^{(j)}, \mathbf{D}) = \frac{\pi_\ell f(y^{(j)}|H_\ell, \mathbf{D})}{\sum_{m=1}^{M} \pi_m f(y^{(j)}|H_m, \mathbf{D})}$$

2. Average the estimated posterior probabilities of H_ℓ over $\forall j$ to obtain MC estimate of $E[P(H_\ell|y, \mathbf{D})|H_r, \mathbf{D}]$

Closed Form MSE of Posterior Mean

For notation, call $E[\beta|Y] = \beta_n$.

$$\begin{aligned} MSE(\beta_n) &= V[\beta_n] + (E[\beta_n] - \beta_T)^2 \\ &= V[\beta_n] + (E[\beta_n])^2 - 2\beta_T E[\beta_n] + \beta_T^2 \end{aligned}$$

where

$$\begin{aligned} V[\beta_n] &= V[\frac{1}{\sigma_e^2}\Sigma_B(X^T y + \sigma^2 V^{-1}\mu)] = V[\frac{1}{\sigma_e^2}\Sigma_B X^T y] \\ &= (\frac{1}{\sigma_e^2})^2 \Sigma_B X^T V[y] X \Sigma_B = (\frac{1}{\sigma_e^2})^2 \Sigma_B X^T (\sigma_e^2 I + XVX^T) X \Sigma_B \\ &= \frac{1}{\sigma_e^2}\Sigma_B X^T X \Sigma_B + (\frac{1}{\sigma_e^2})^2 \Sigma_B X^T XVX^T X \Sigma_B \\ E[\beta_n] &= E[\frac{1}{\sigma_e^2}\Sigma_B(X^T y + \sigma^2 V^{-1}\mu)] = \frac{1}{\sigma_e^2}\Sigma_B(X^T E[y] + \sigma^2 V^{-1}\mu) \\ &= \frac{1}{\sigma_e^2}\Sigma_B(X^T X\mu + \sigma^2 V^{-1}\mu) = \frac{1}{\sigma_e^2}\Sigma_B(X^T X + \sigma^2 V^{-1})\mu \end{aligned}$$