Meeting Update

Gaussian Process Covariance Function Selection Using Minimum Energy Designs, For Real This Time

Kristyn Pantoja

Department of Statistics
Texas AM University

05 April 2019

Outline

Last Time

Limiting Distribution of SMED for LM Selection

- Review Comparison between Fast & One-at-a-Time Algorithm
- Pictures for $N = 11, 67$ and $K = 4$
- When K Gets Large
- $K = 20$
- $K = 40$
- $K = 100$

Gaussian Process Model Selection

- "Fast" Algorithm Idea

Last Time

Last Meeting Recap

What happened last time

1. Fixed the candidate sets, $\mathbf{C}_{k+1}^{j}$, for $\mathbf{x}_{j}^{k+1}$.
2. Talked about how to implement SMED model selection for Gaussian Processes
3. Also talked about using the computer clusters `https://it.tamu.edu/services/academics-and-research/research/hprc/index.php`

Questions

1. What is the limiting distribution of the design? i.e. what happens when K gets large?
2. Histograms of the distances between points: total distance and distances in the sub-regions.
3. Implement SMED for GP model selection

Limiting Distribution of SMED for LM Selection

Limiting Distribution of SMED for LM Selection

Review Comparison between Fast & One-at-a-Time Algorithm

Review

1. Fast Algorithm vs. One-at-a-Time Algorithm, $N = 11, K = 4$: not exactly the same, but close. Is either of N or K not large enough?
2. Fast Algorithm vs. One-at-a-Time Algorithm, $N = 67, K = 4$: distribution of design points in the new fast algorithm is definitely closer here. But still not exactly the same.
3. Seeing Order of Design Points $N = 67, K = 4$: Still the difference in order of design points, though, - not so much in the beginning, but towards the end, where the points seem to be in ascending order. Again, is it because K is not large enough?

Revisit the old pictures...

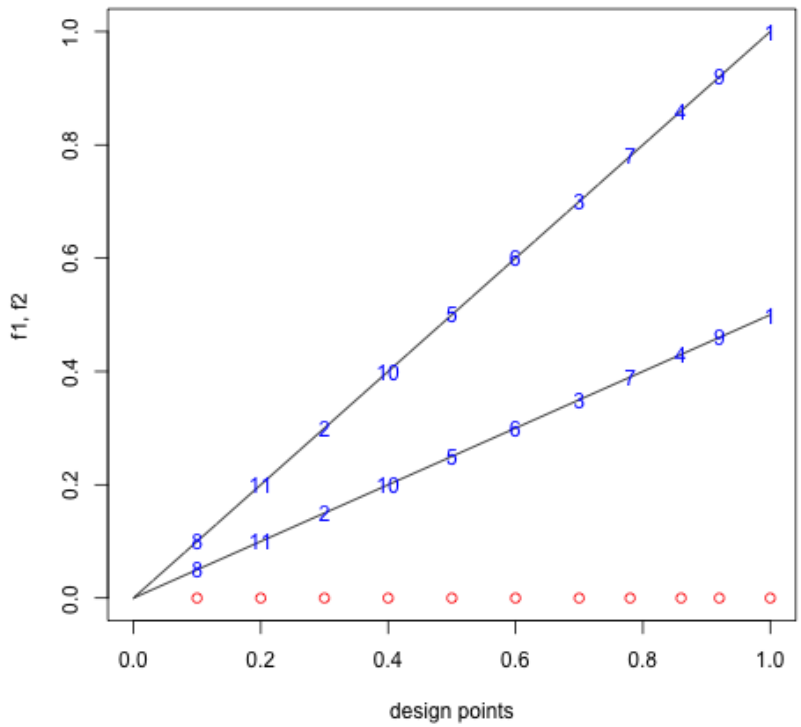
Limiting Distribution of SMED for LM Selection

Pictures for $N = 11, 67$ and $K = 4$

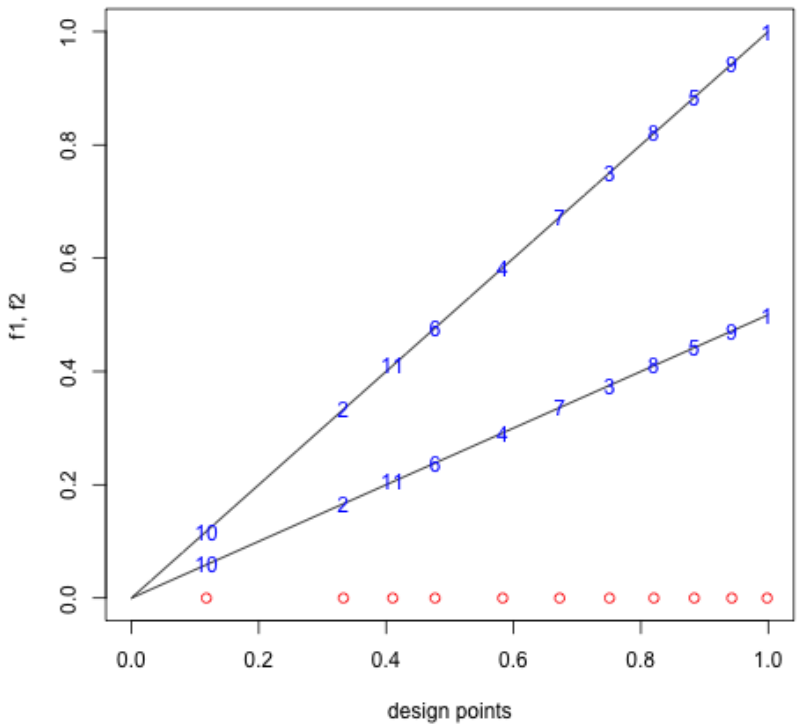
Fast vs. One-at-a-Time, $N = 11, K = 4$

Fast Algorithm seems more spread out. We will see that this is due to K not being large enough.

Fast Algorithm



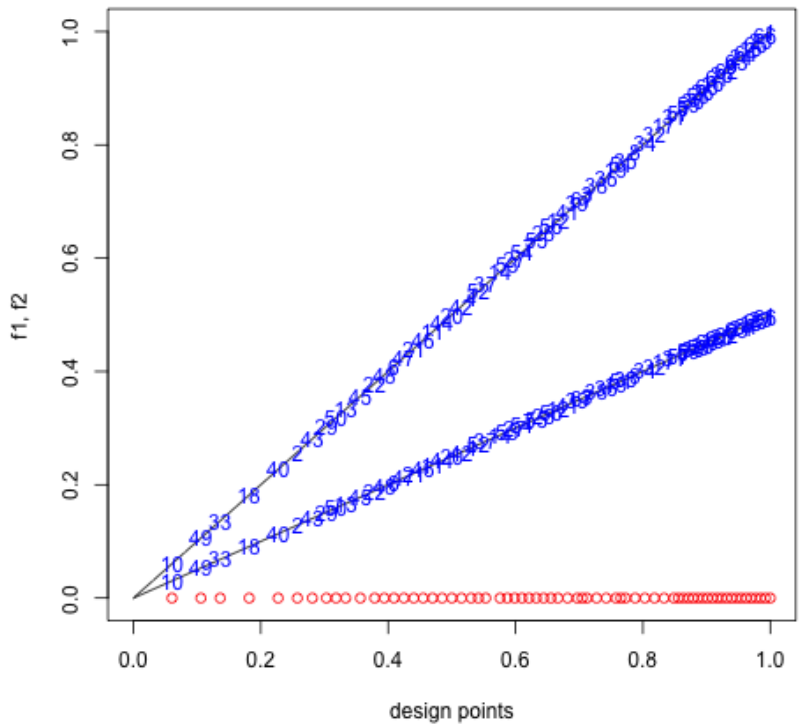
One-at-a-Time Algorithm



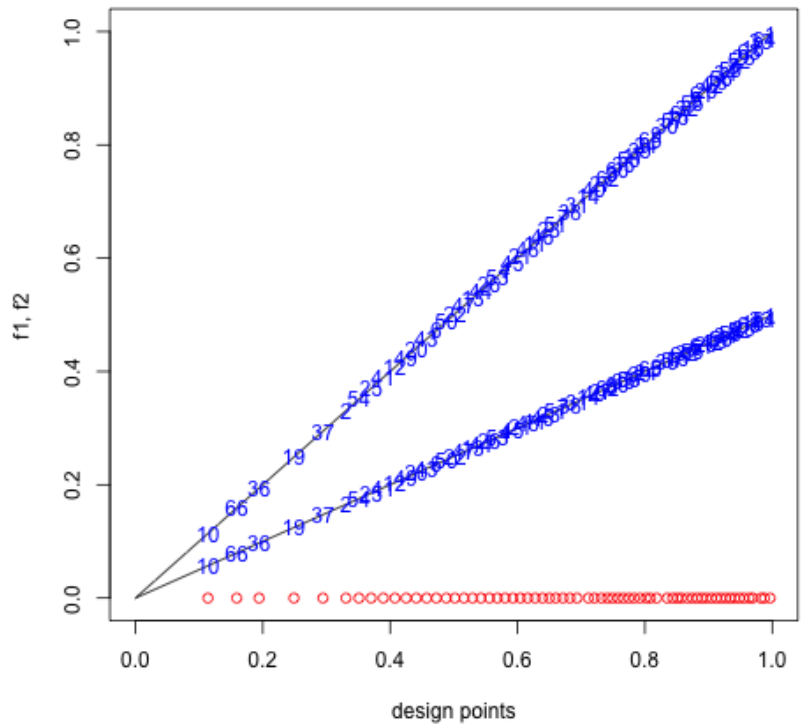
Fast vs. One-at-a-Time, $N = 67, K = 4$

Distribution of design points in the new fast algorithm is definitely closer to that of the one-at-a-time algorithm, compared to the wrong/old fast algorithm.

Fast Algorithm



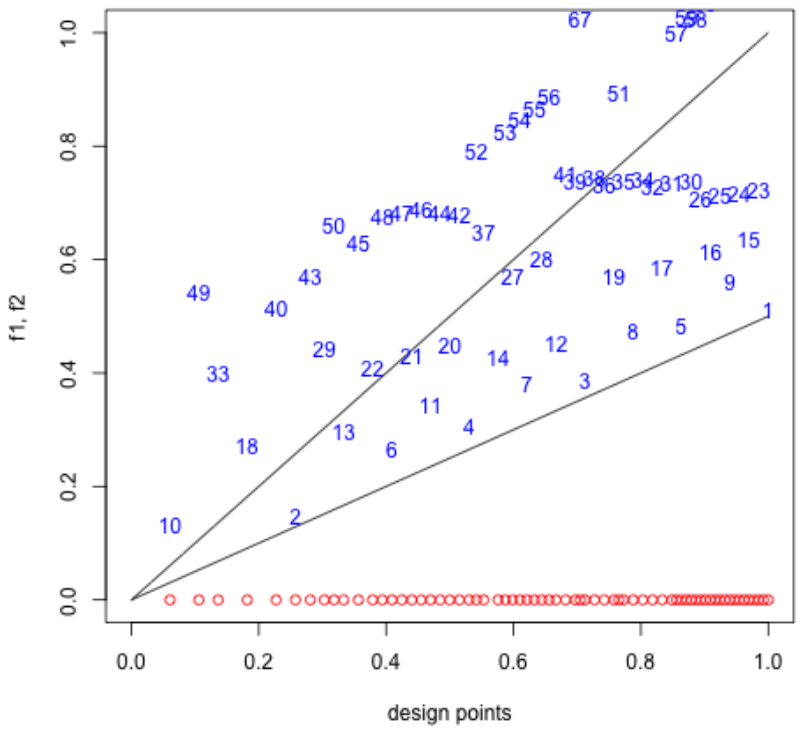
One-at-a-Time Algorithm



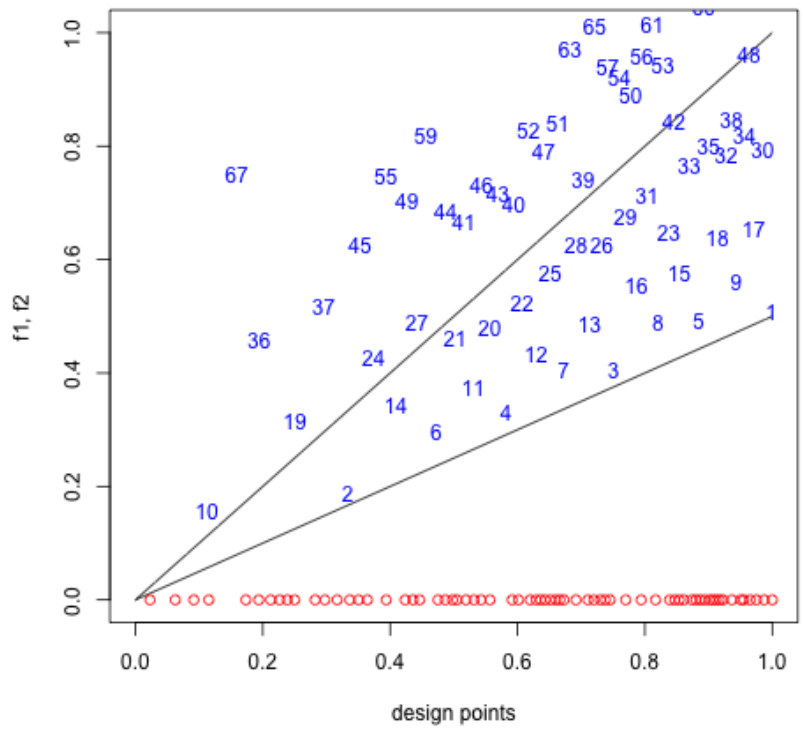
Order of Design Points $N = 67, K = 4$

It seems there is still the difference in order of design points, though, - not so much in the beginning, but towards the end. Not enough distance between later points, for some reason (even though this is not the case for earlier points!).

Fast Algorithm



One-at-a-Time Algorithm



Limiting Distribution of SMED for LM Selection

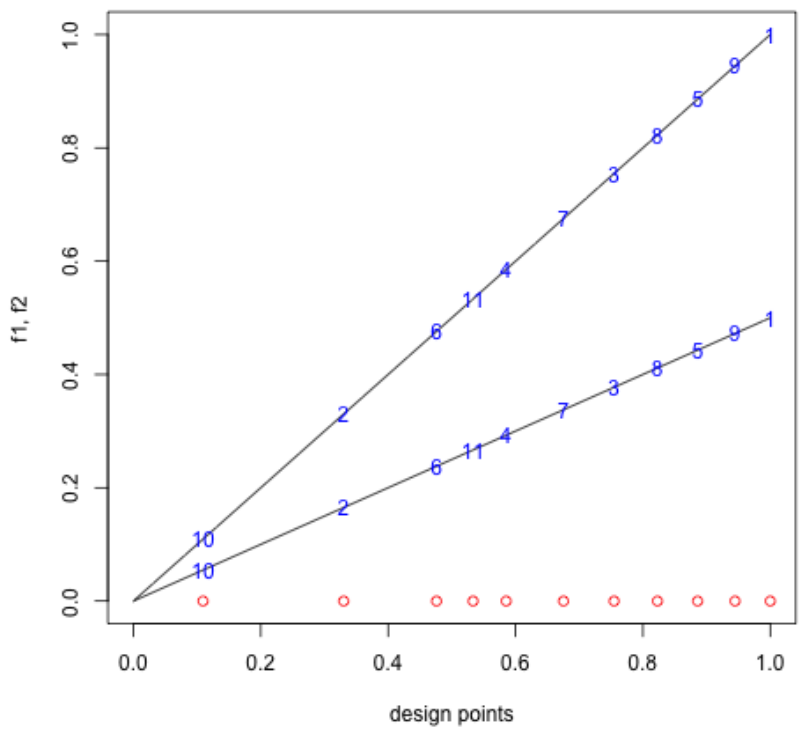
When K Gets Large

Fast vs. One-at-a-Time, $N = 11, K = 100$

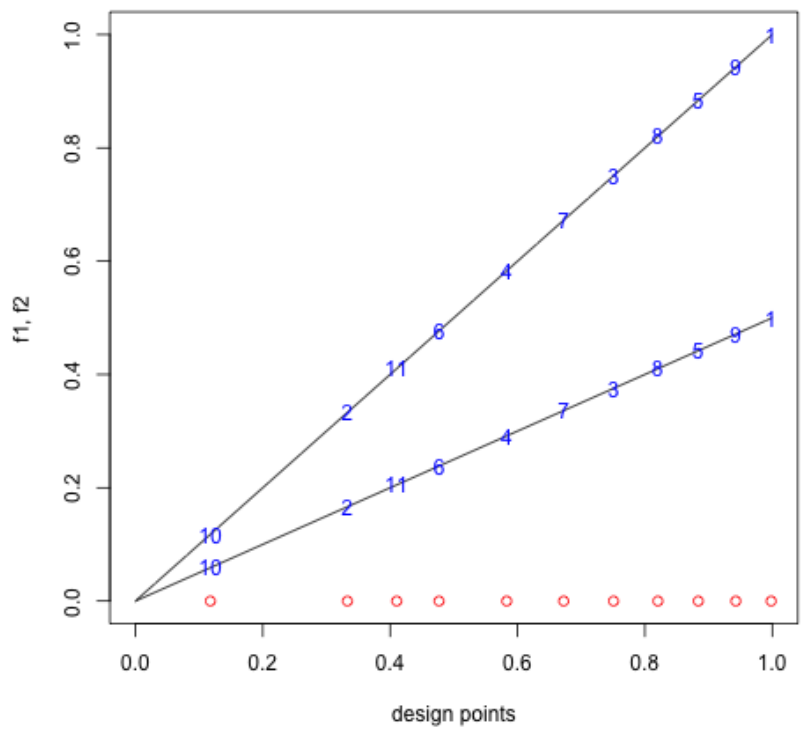
Now the two designs look very similar.

As K gets higher, even around $K = 50$, there are some slight changes, especially for the location of 11. The changes tend to be at the 0.5 mark. Below is for $K = 100$. (The design didn't change much after $K = 60$, though.)

Fast Algorithm



One-at-a-Time Algorithm



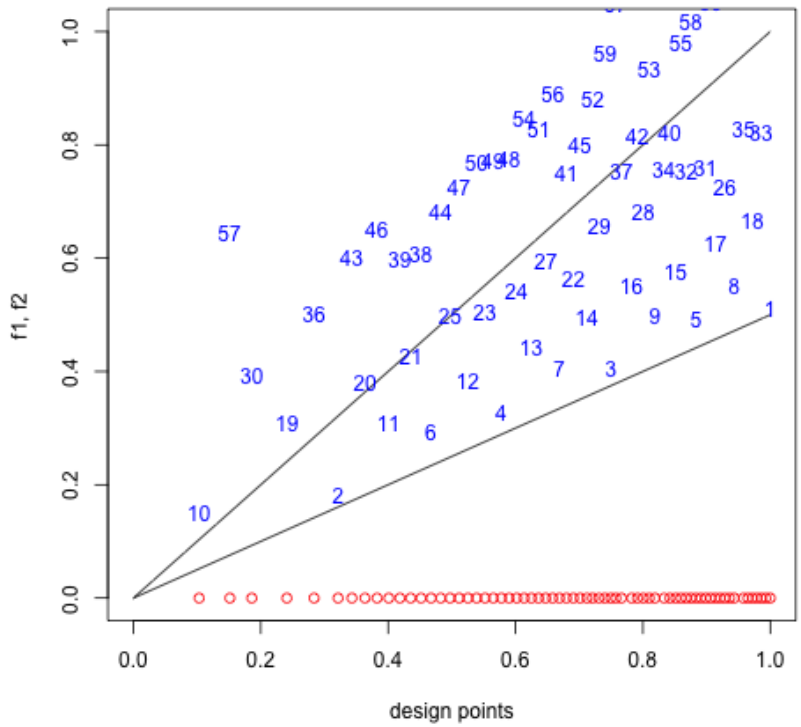
Limiting Distribution of SMED for LM Selection

$K = 20$

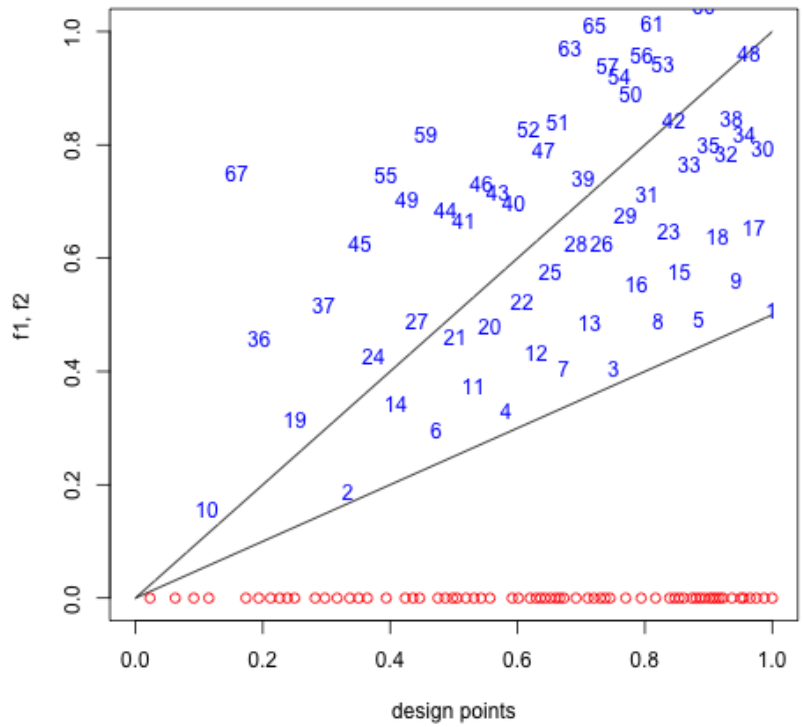
Order of Design Points $N = 67, K = 20$

Some points get switched, but otherwise their locations seem comparable.

Fast Algorithm



One-at-a-Time Algorithm

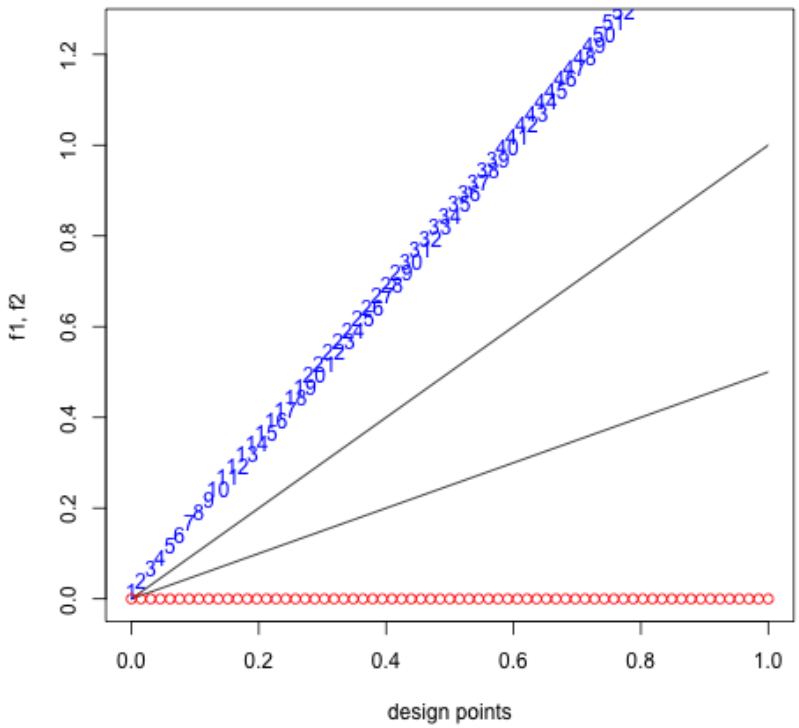


Limiting Distribution of SMED for LM Selection

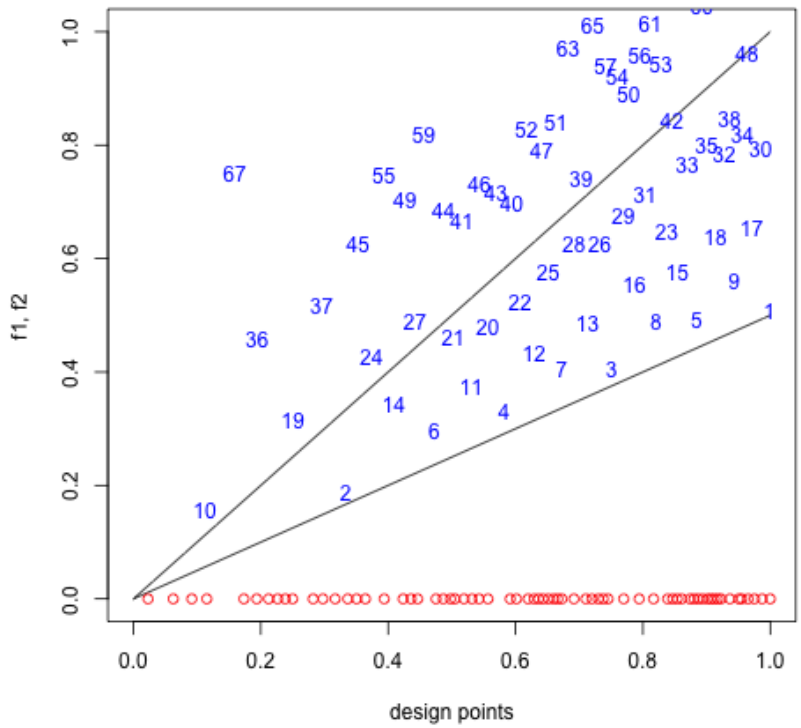
$K = 40$

$k = 1$

$K = 40$

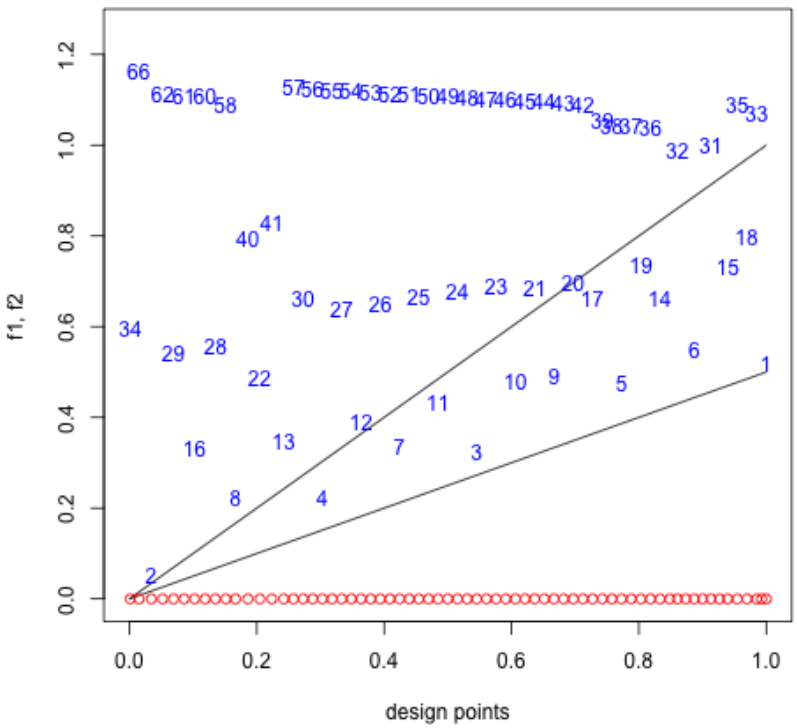


One-at-a-Time

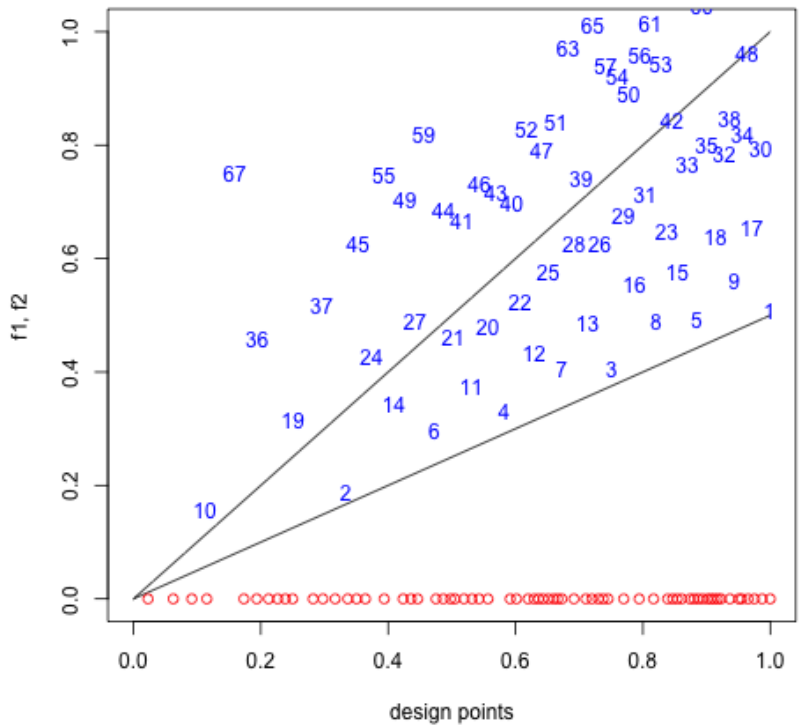


$k = 5$

$K = 40$

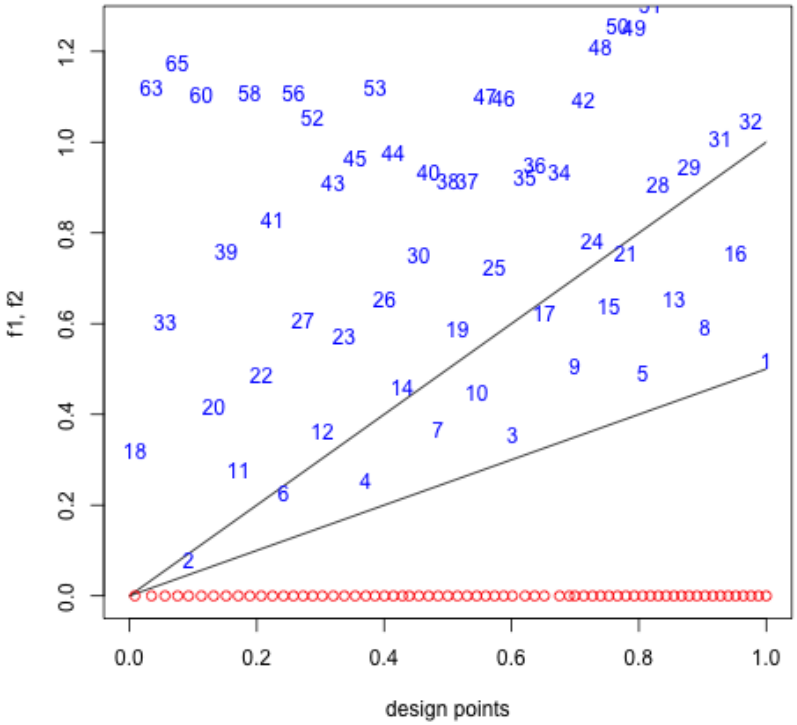


One-at-a-Time

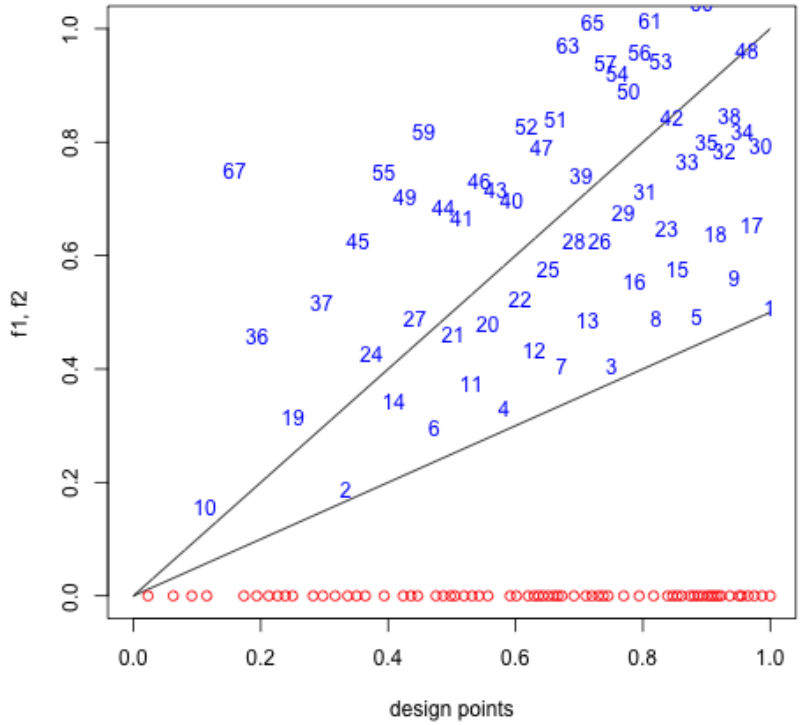


$k = 10$

$K = 40$

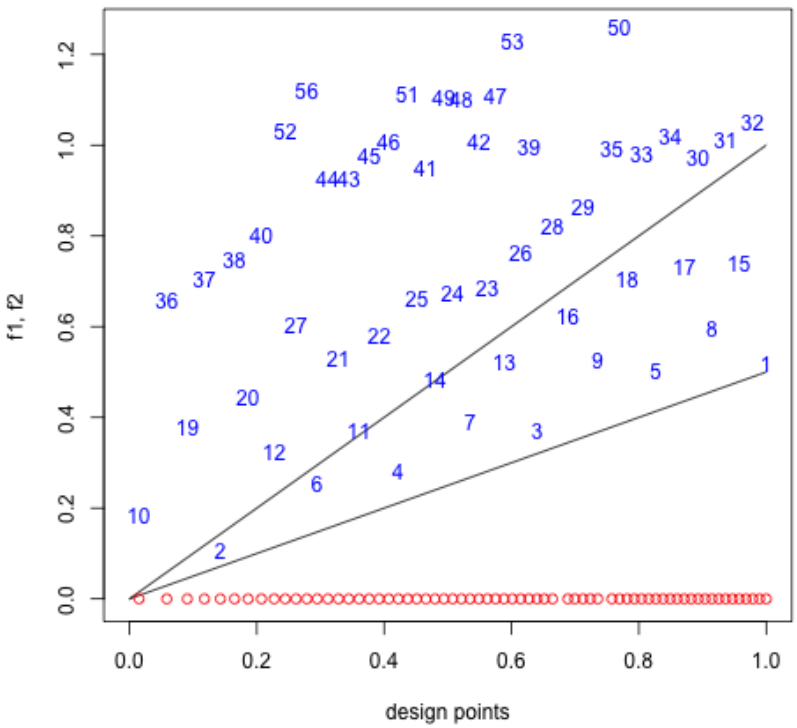


One-at-a-Time

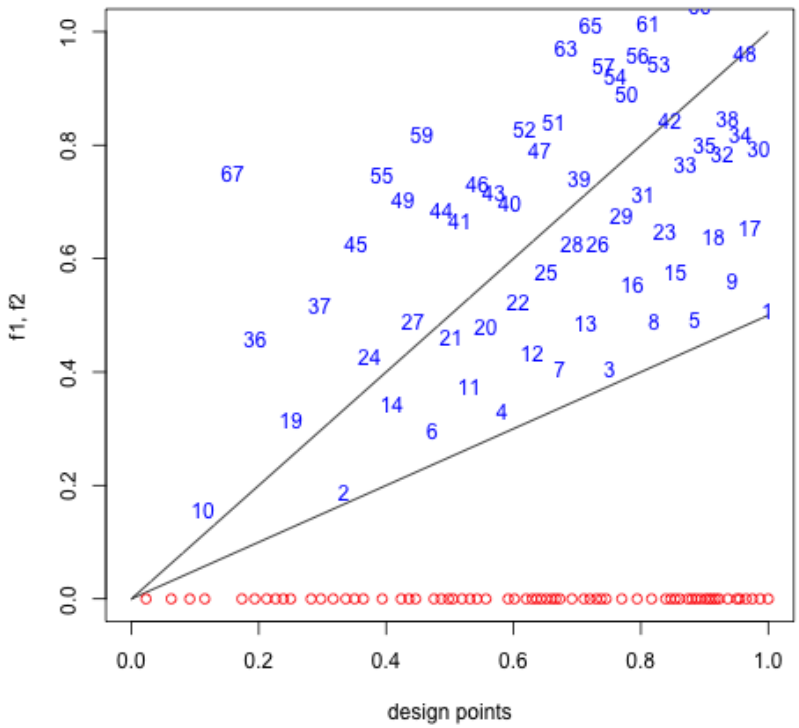


$k = 15$

$K = 40$

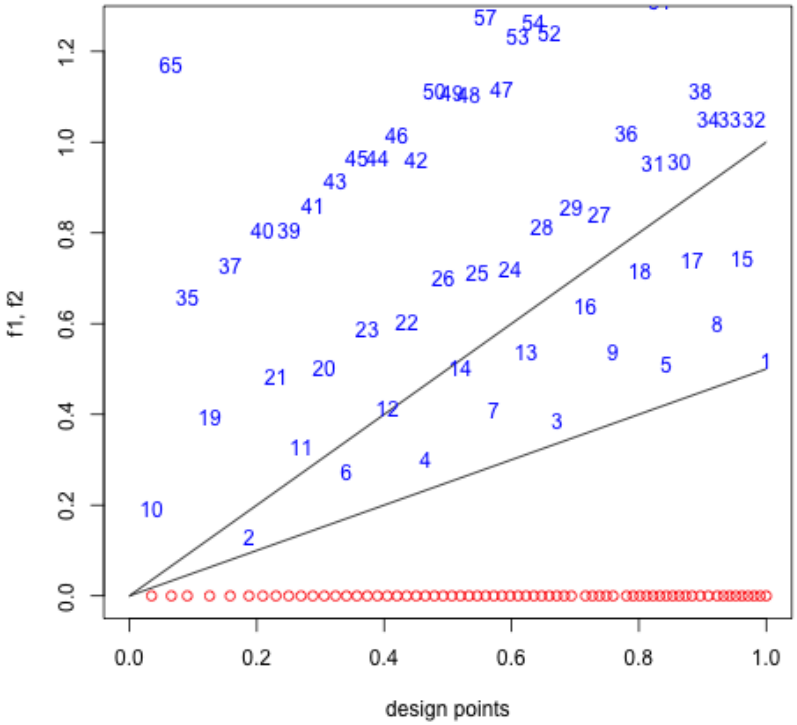


One-at-a-Time

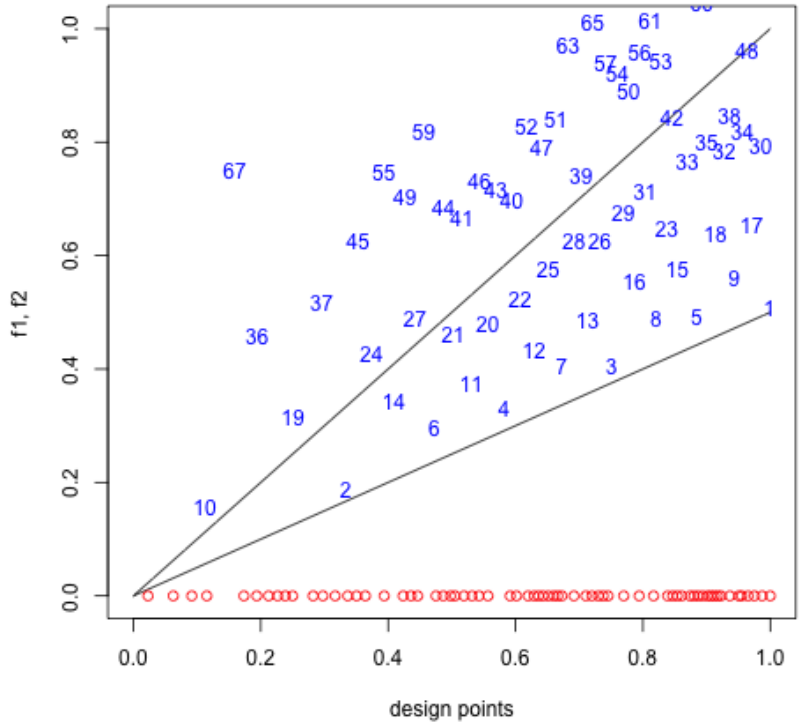


$k = 20$

$K = 40$

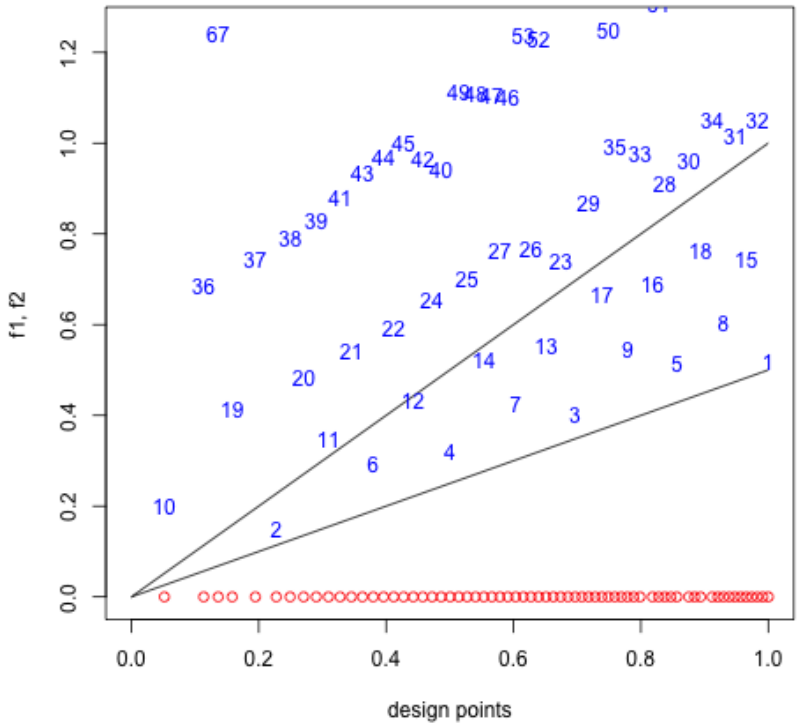


One-at-a-Time

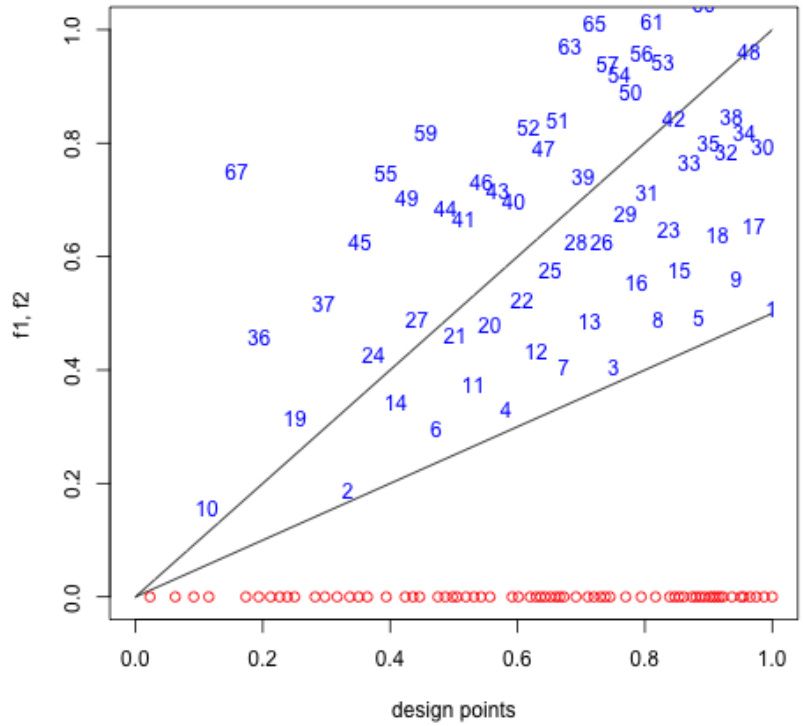


$k = 25$

$K = 40$

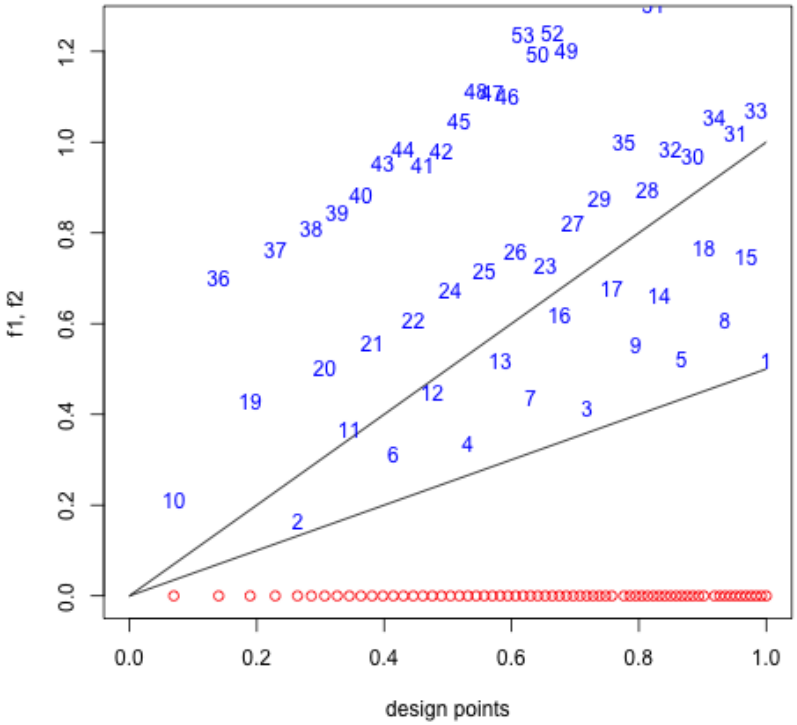


One-at-a-Time

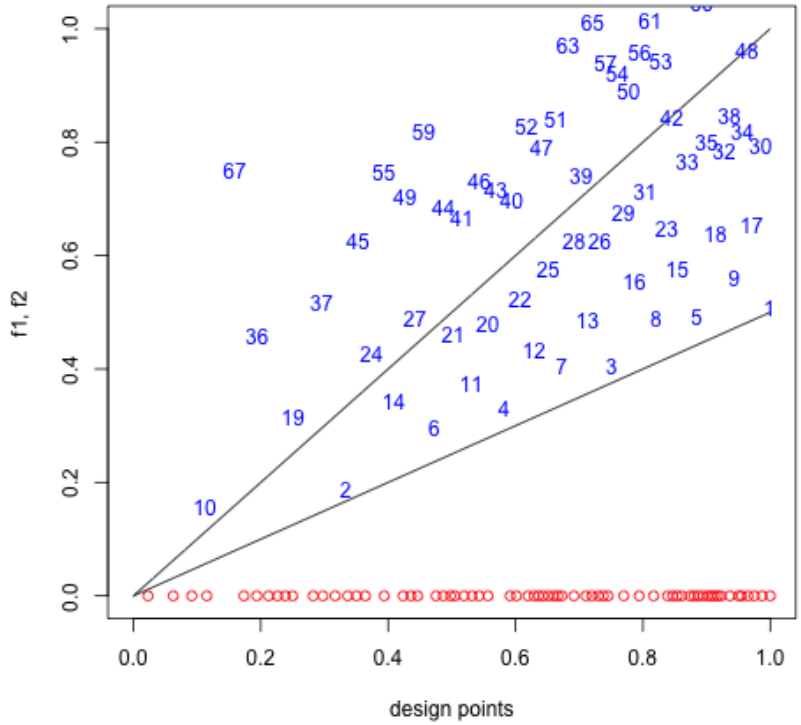


$k = 30$

$K = 40$

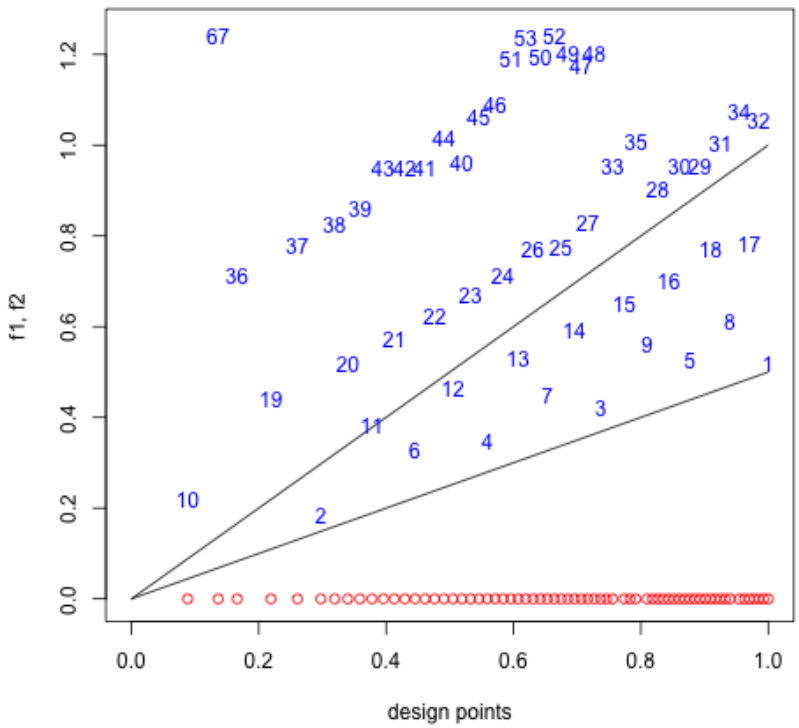


One-at-a-Time

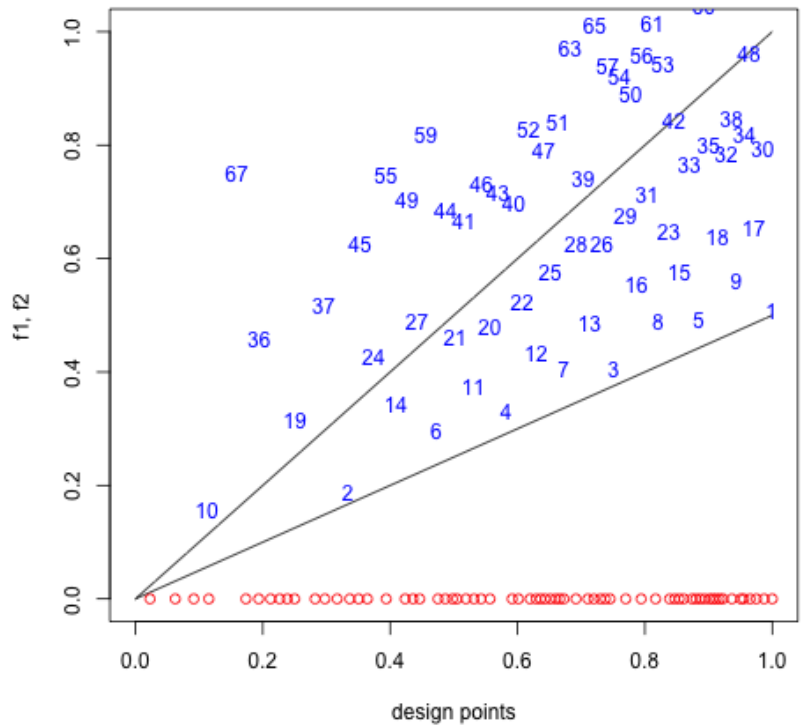


$k = 35$

$K = 40$

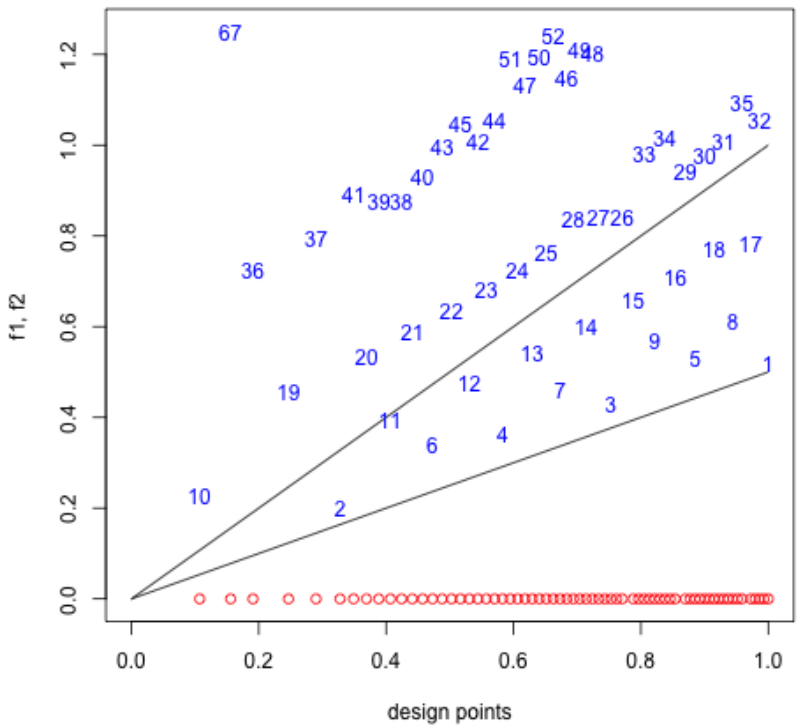


One-at-a-Time

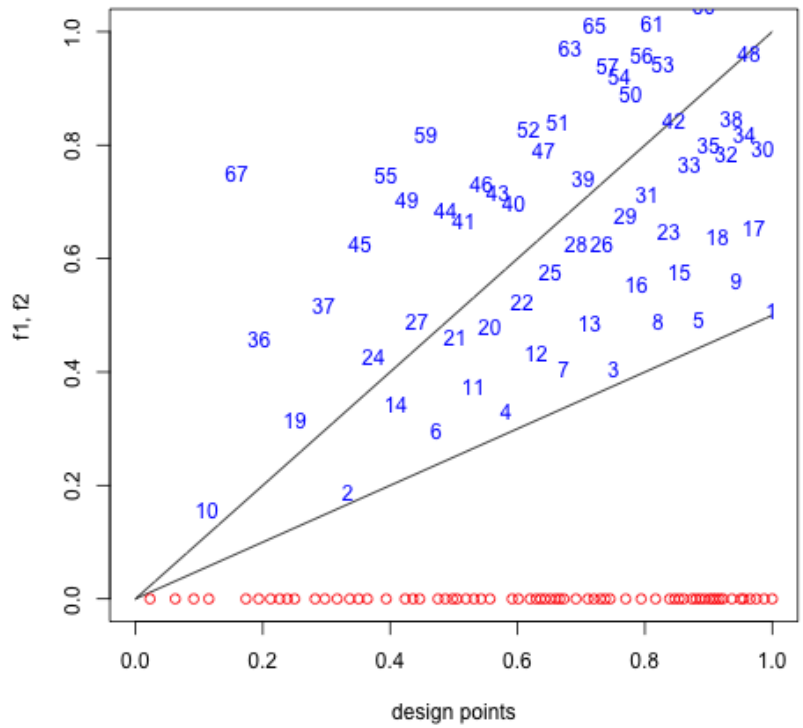


$k = 40$

$K = 40$



One-at-a-Time

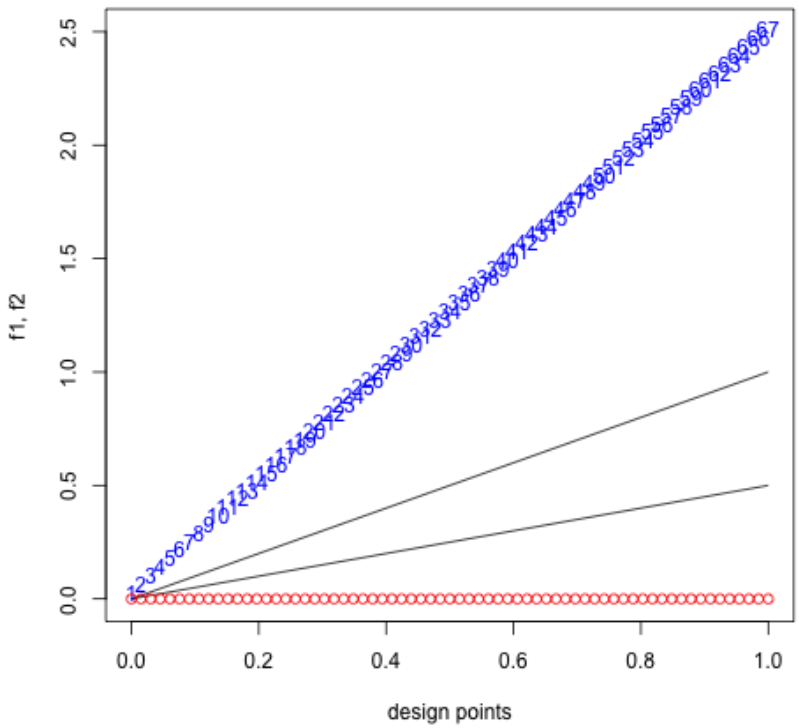


Limiting Distribution of SMED for LM Selection

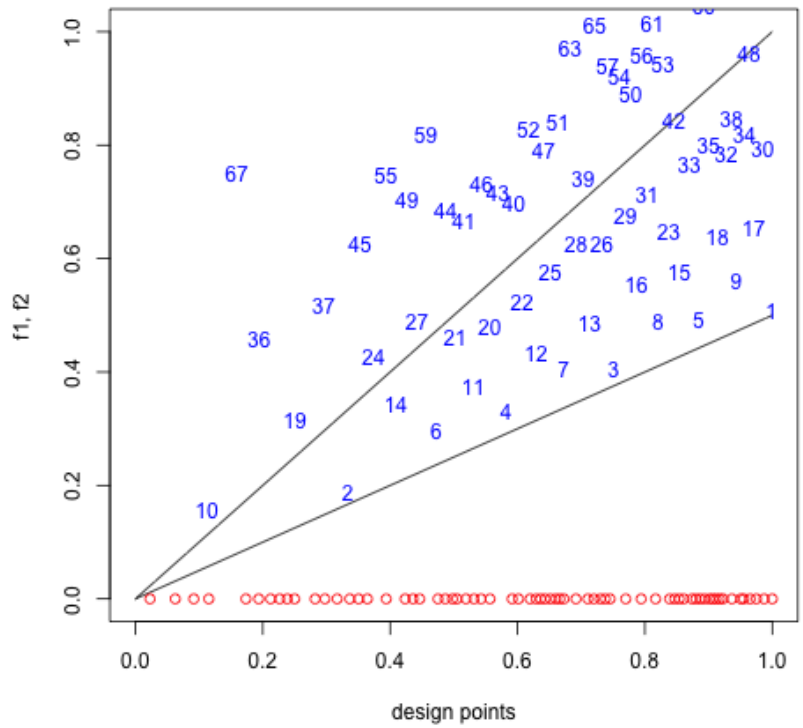
$K = 100$

$k = 1$

$K = 100$

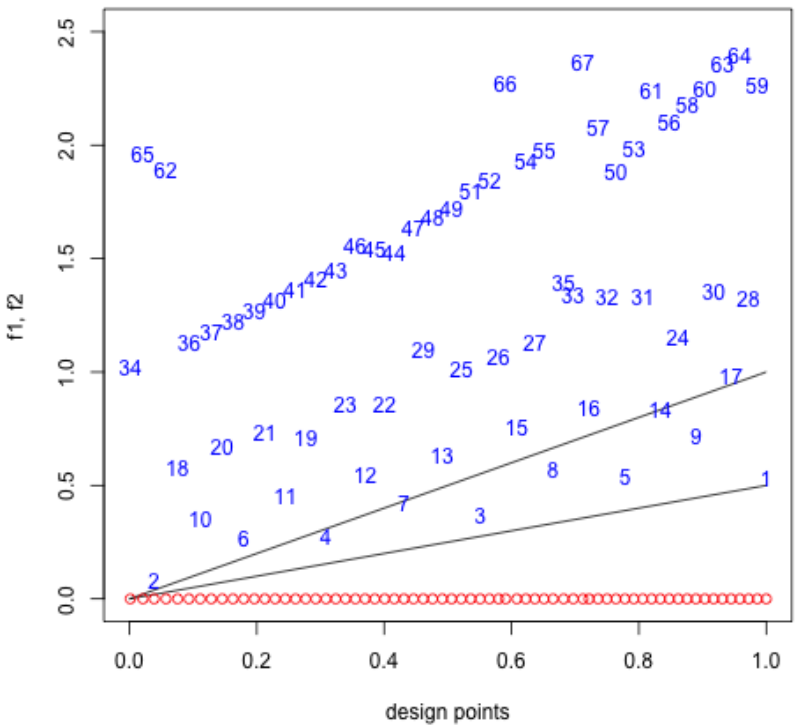


One-at-a-Time

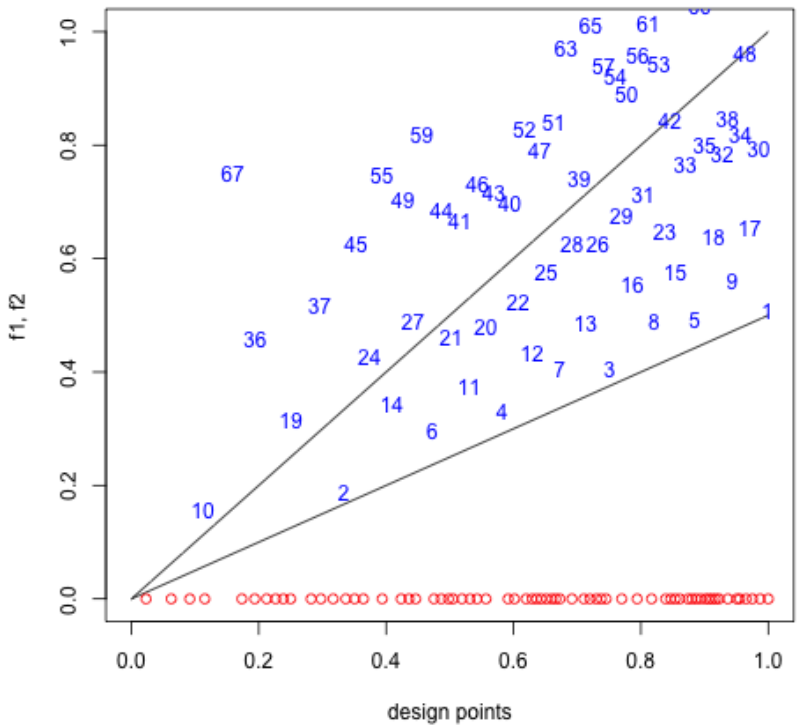


$k = 10$

$K = 100$

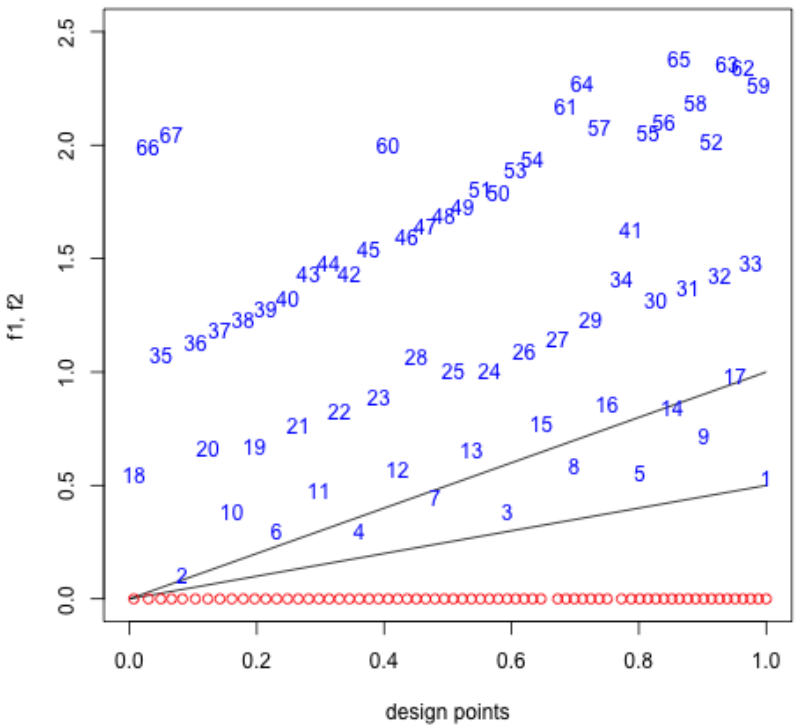


One-at-a-Time

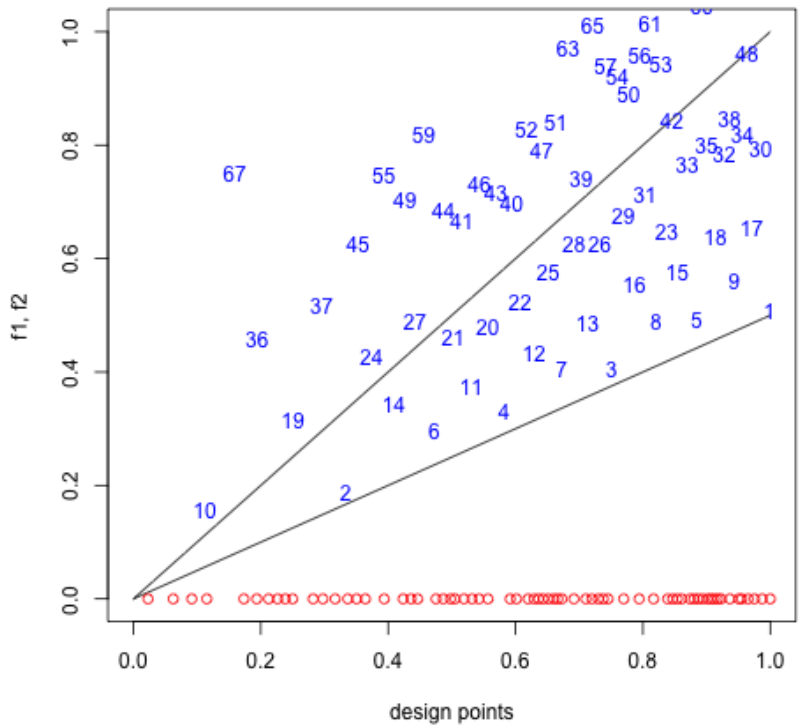


$k = 20$

$K = 100$

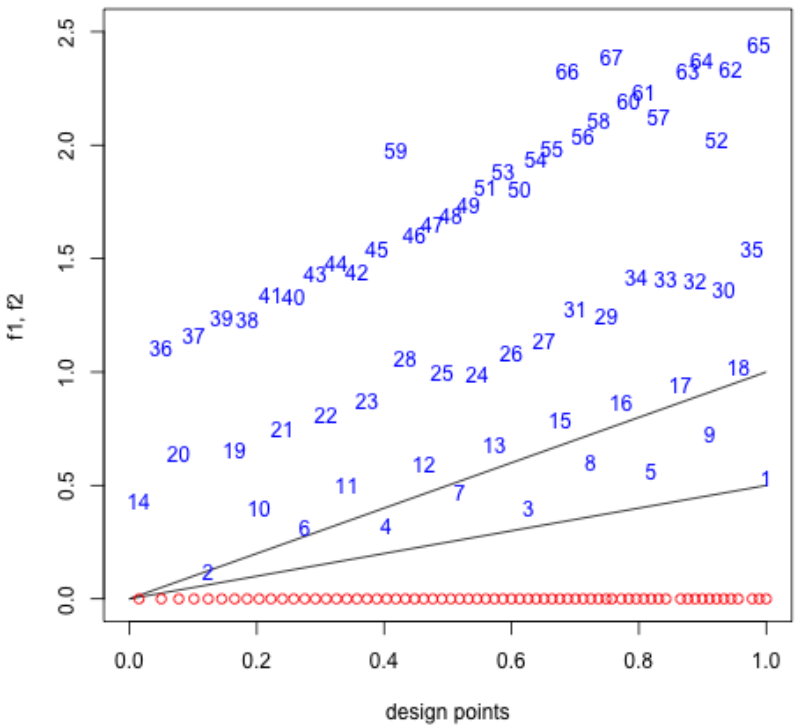


One-at-a-Time

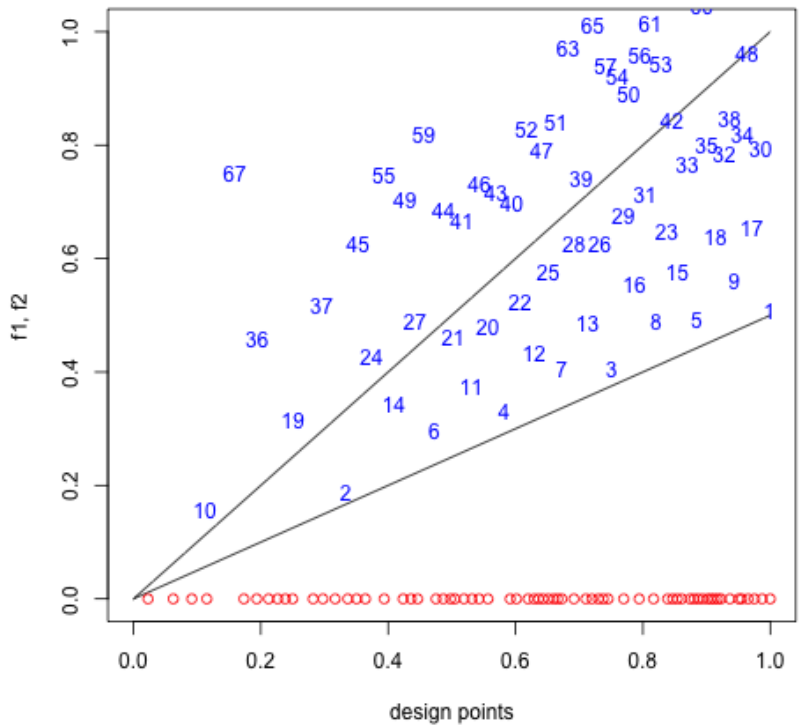


$k = 30$

$K = 100$

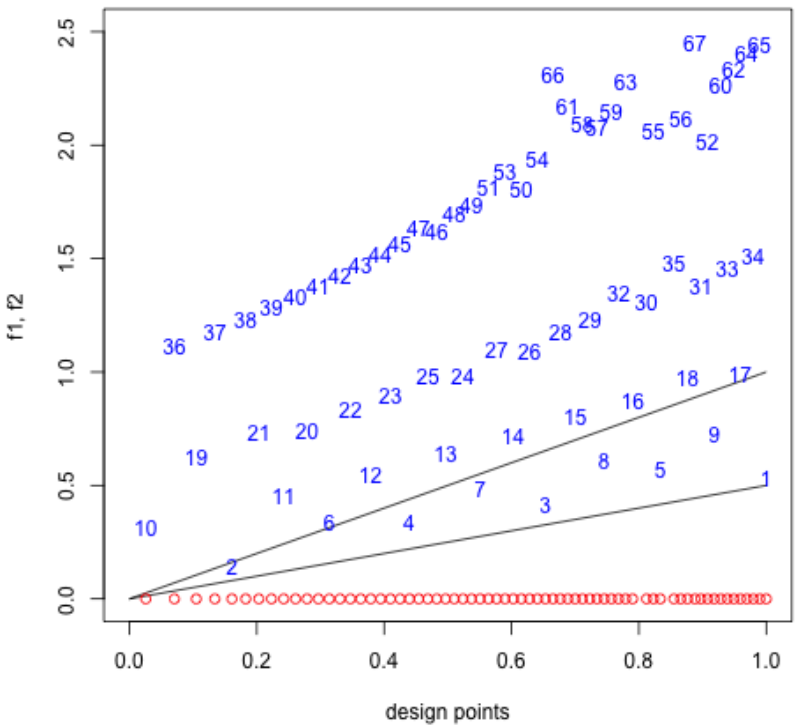


One-at-a-Time

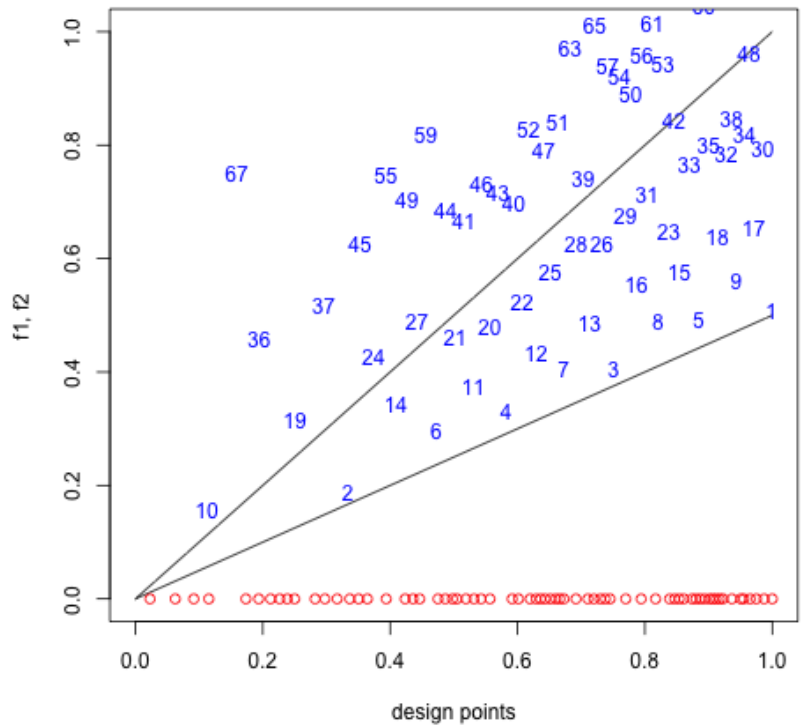


$k = 40$

$K = 100$

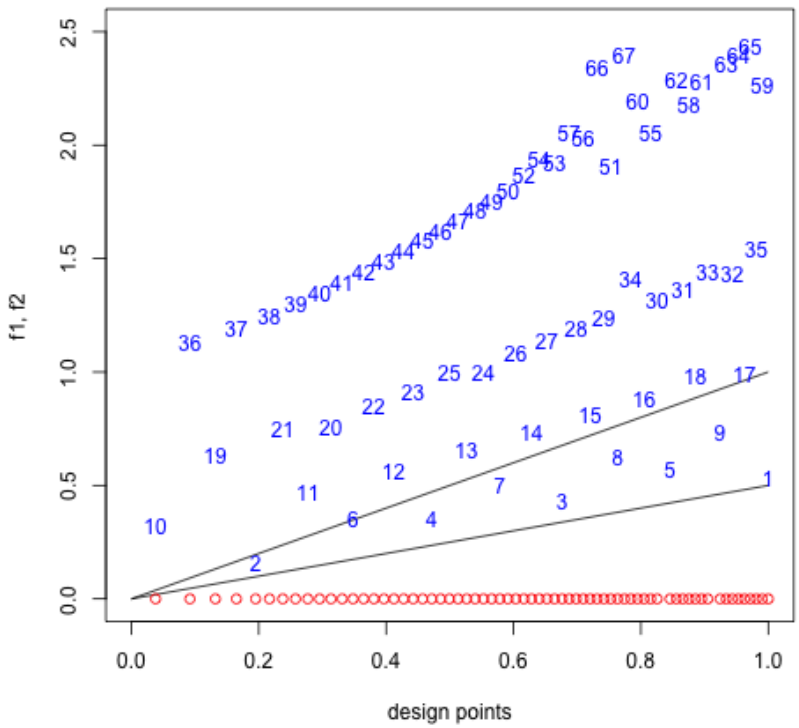


One-at-a-Time

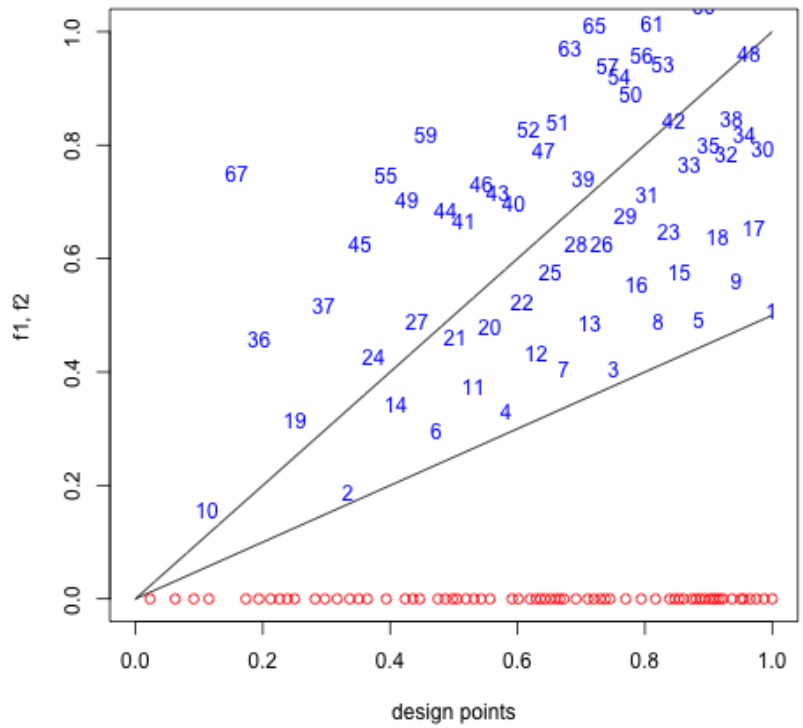


$k = 50$

$K = 100$

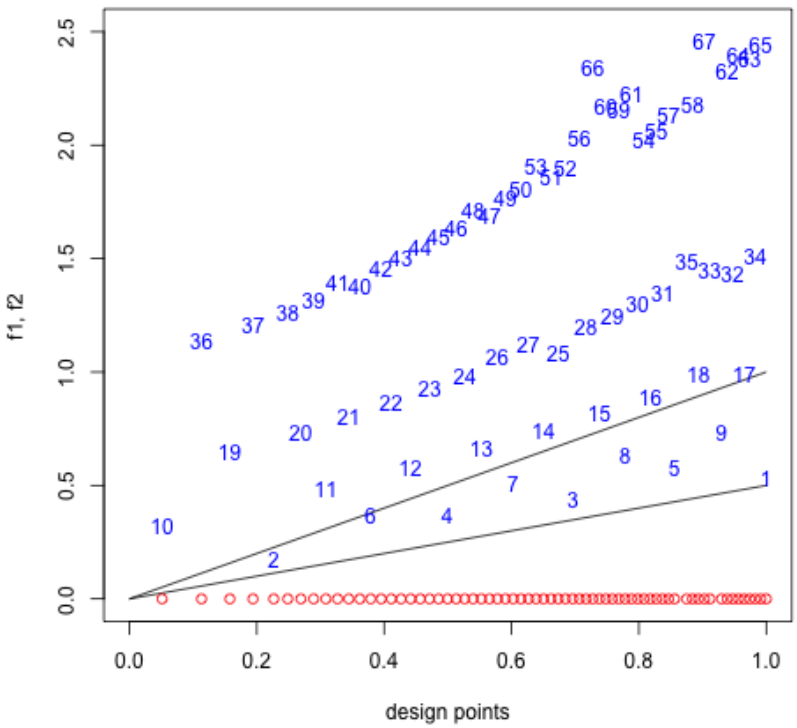


One-at-a-Time

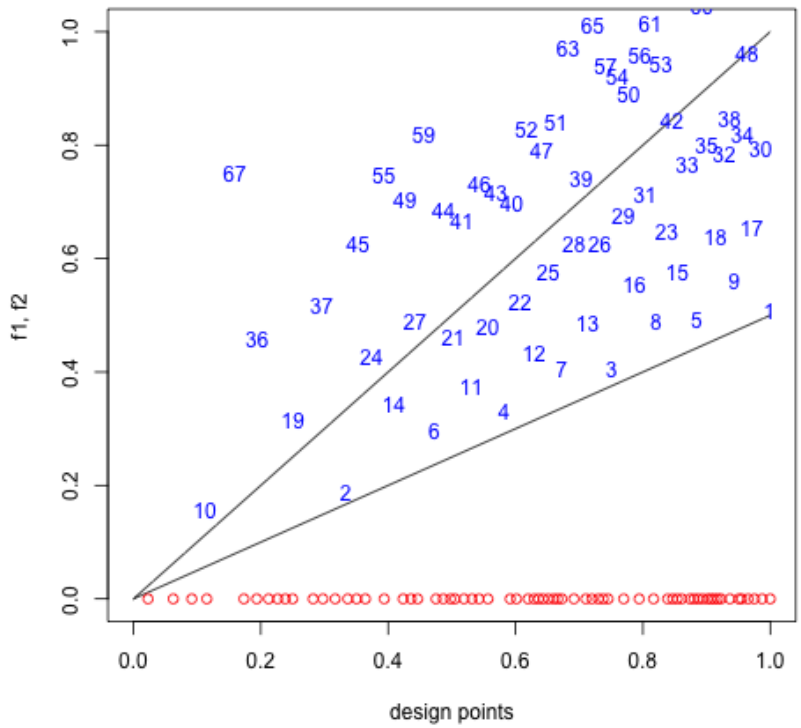


$k = 60$

$K = 100$

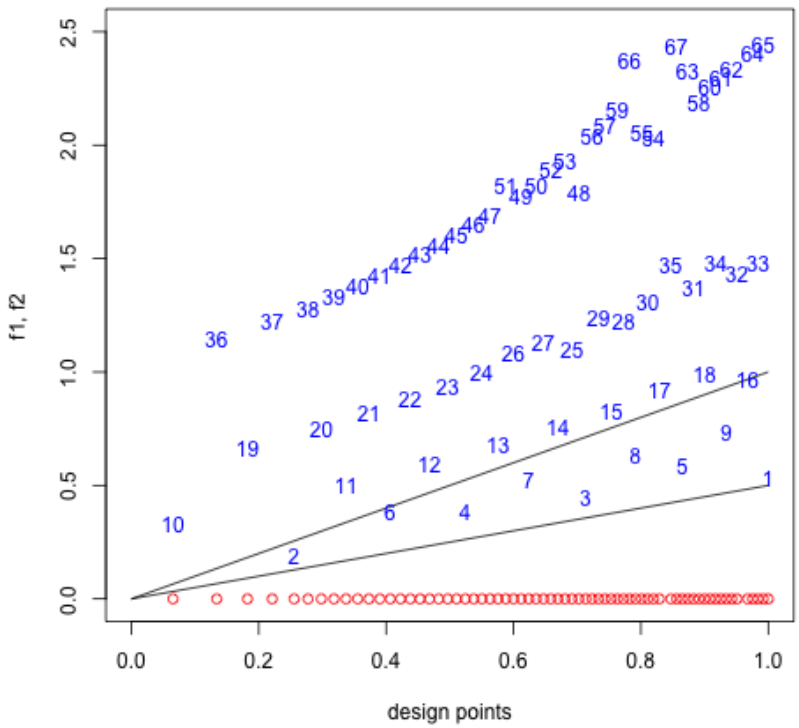


One-at-a-Time

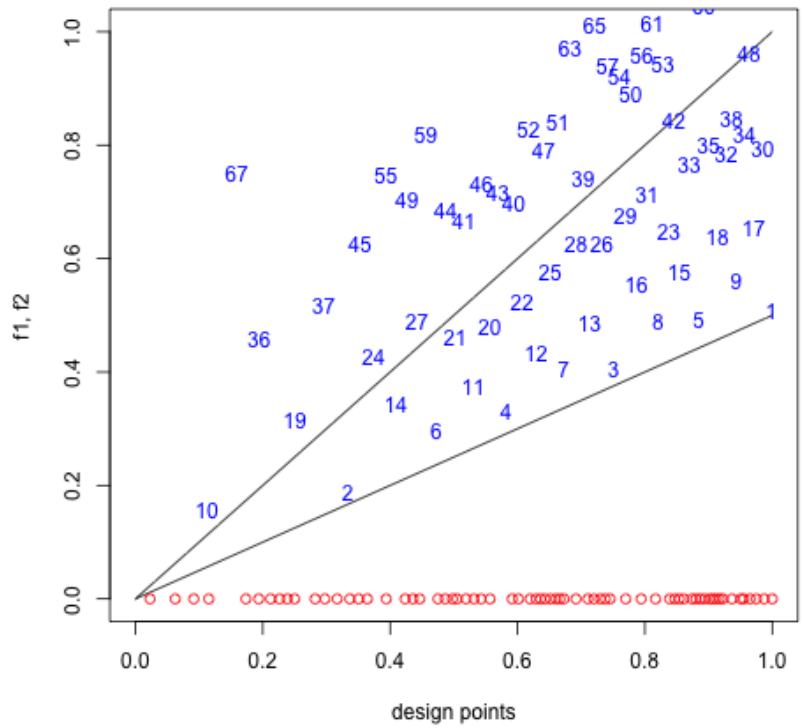


$k = 70$

$K = 100$

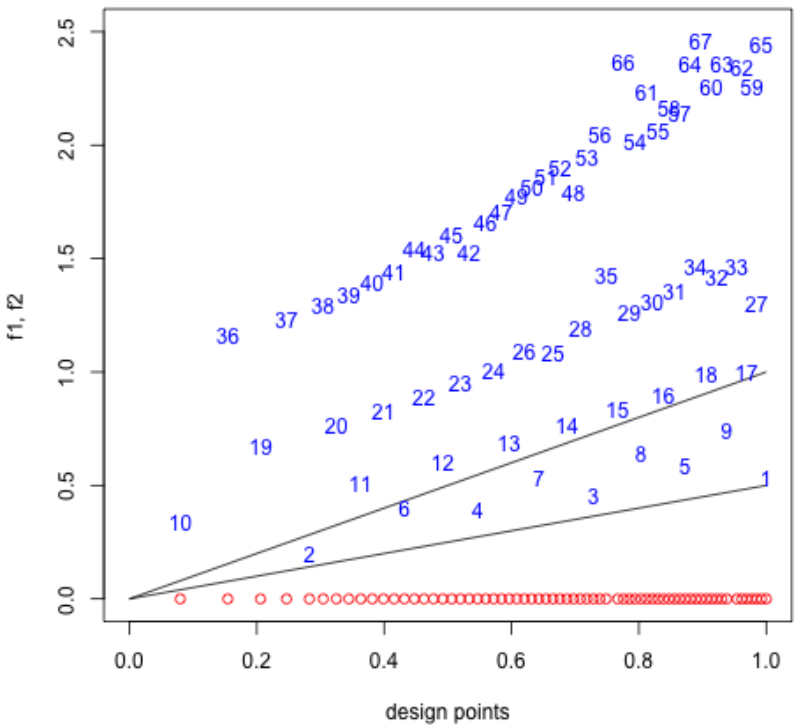


One-at-a-Time

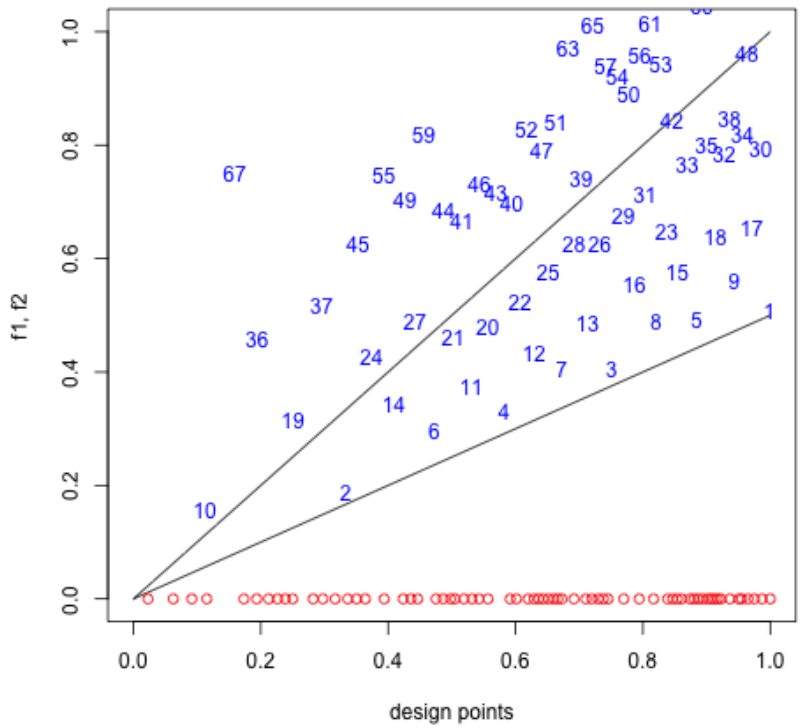


$k = 80$

$K = 100$

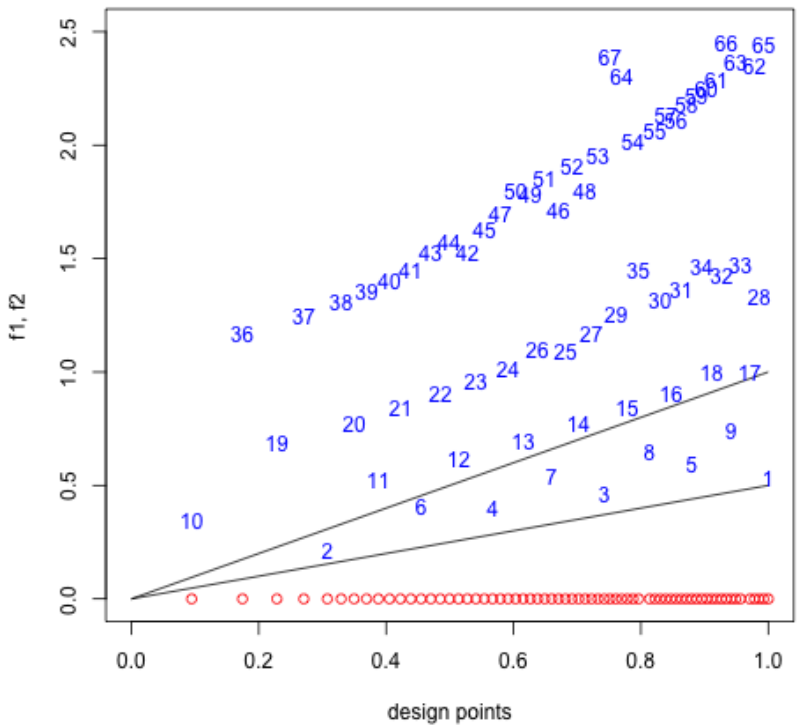


One-at-a-Time

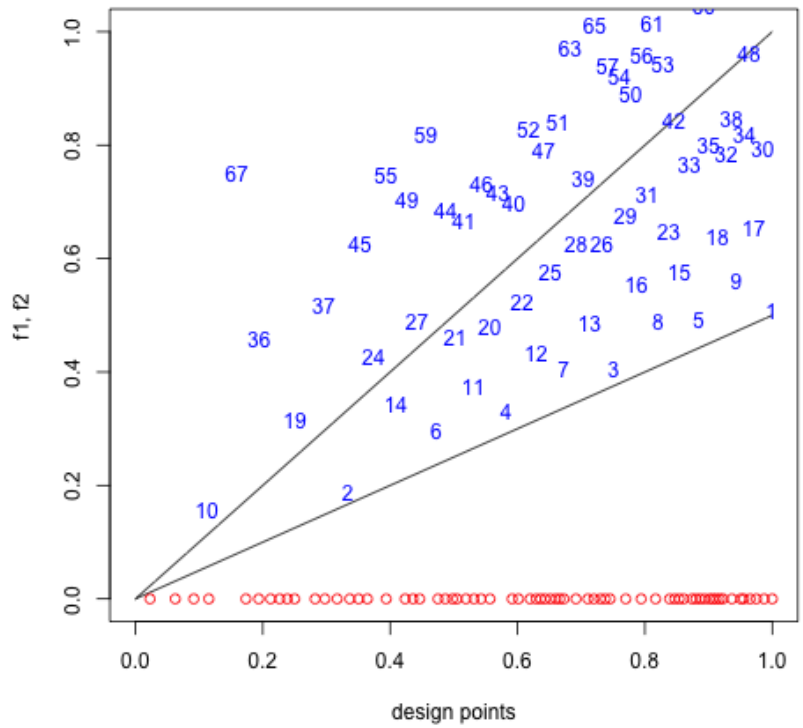


$k = 90$

$K = 100$

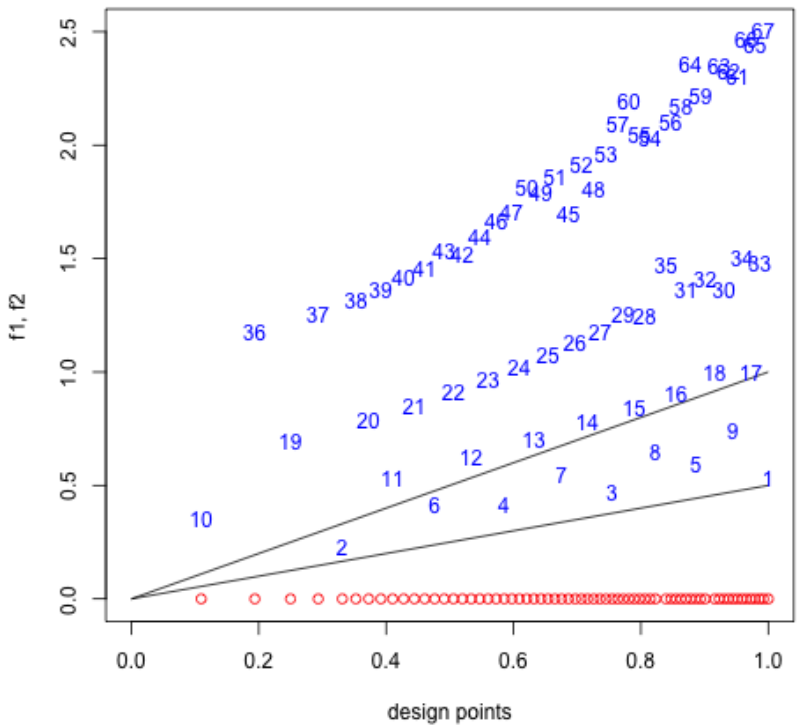


One-at-a-Time

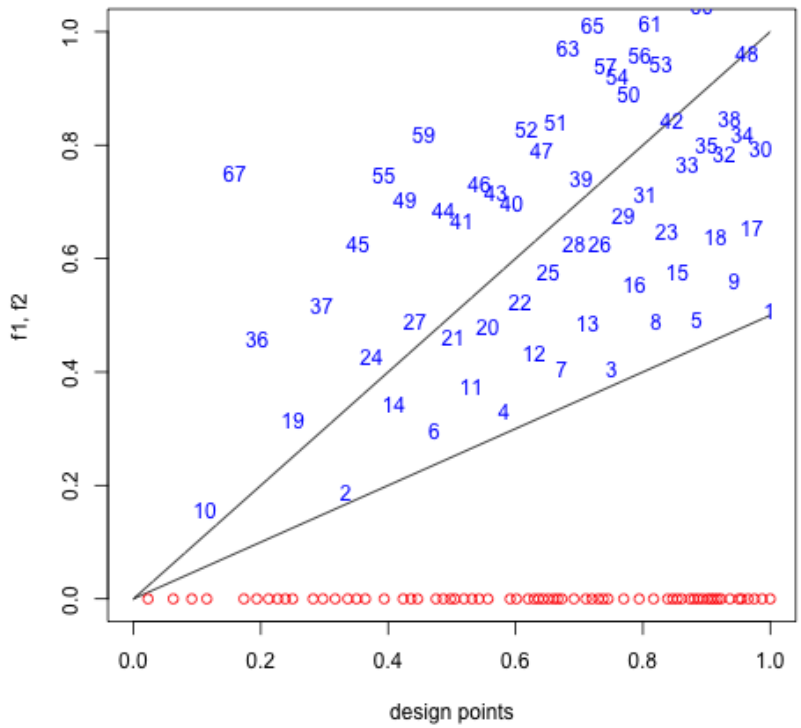


$k = 100$

$K = 100$



One-at-a-Time



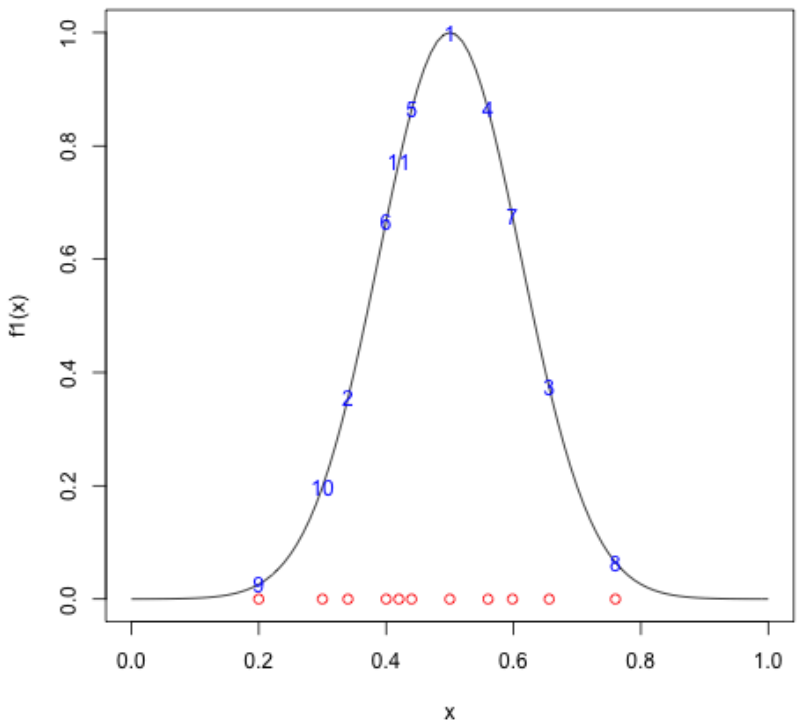
Some Observations

1. Seems like after awhile, the design becomes a monotonically increasing sequence of design points - seems more space-filling than parameter-estimating. At least it's more flexible for prediction? (...)
2. It seems like the size of K is having big affect on how quickly the design degenerates to monotonic sequence - maybe due to $\gamma_k = \frac{k-1}{K-1}$?
3. Unsure about whether limiting distribution can be studied by observing large K.
4. It seems that at some point, for large K, space-filling becomes more important than balancing the charges between particles.
5. My implementation of the fast algorithm in the original paper seems comparable for Normal and Beta distributions, so maybe not something wrong with the code?

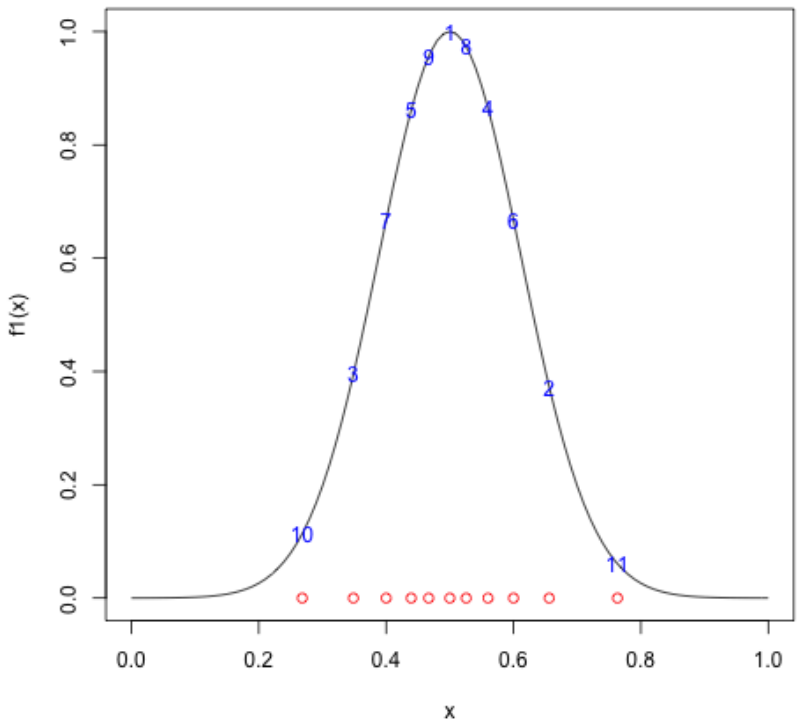
Normal Example

$f_1 = N(1/2, 1/9)$

$K = 4$



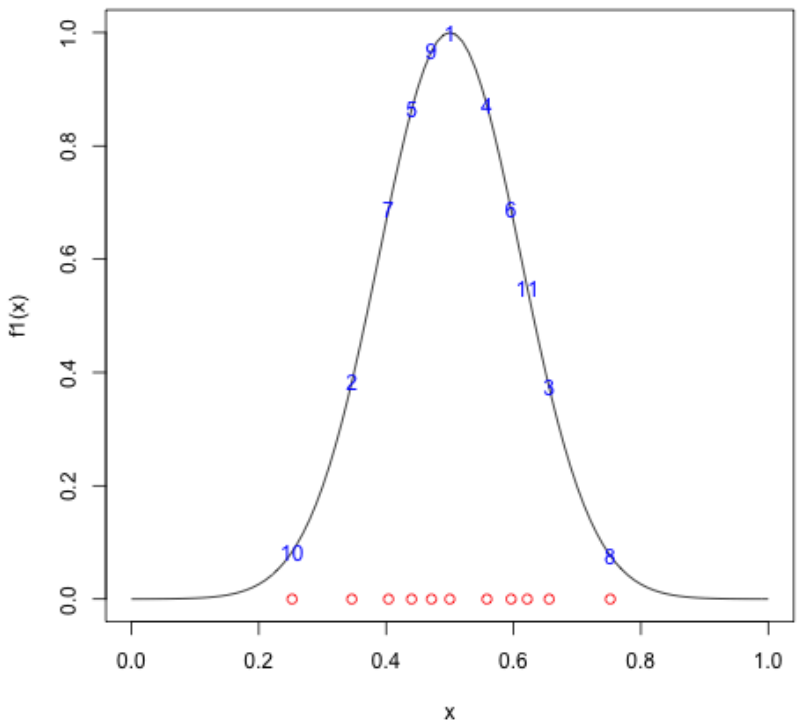
mined Package



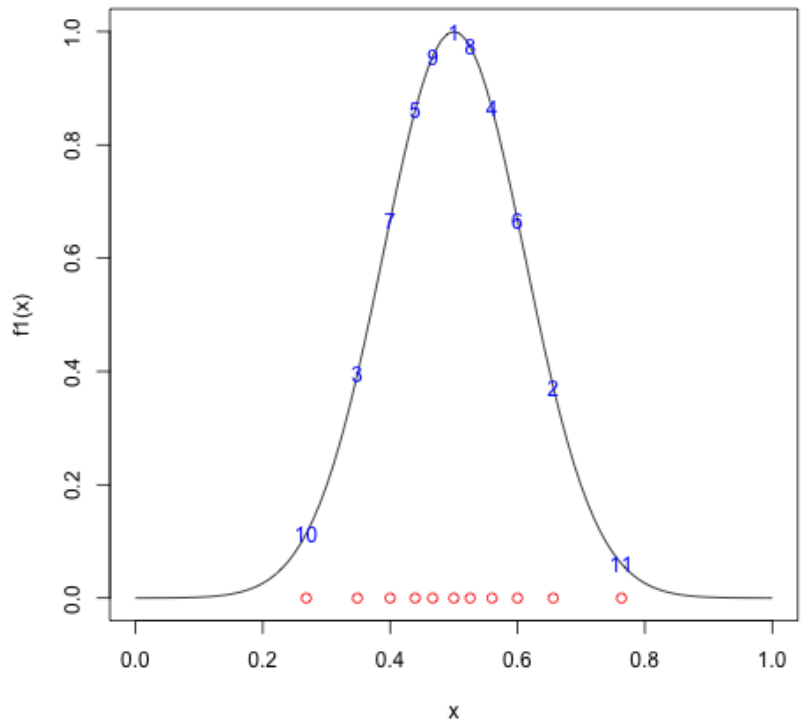
Normal Example

$$f_1 = N(1/2, 1/9)$$

$K = 10$



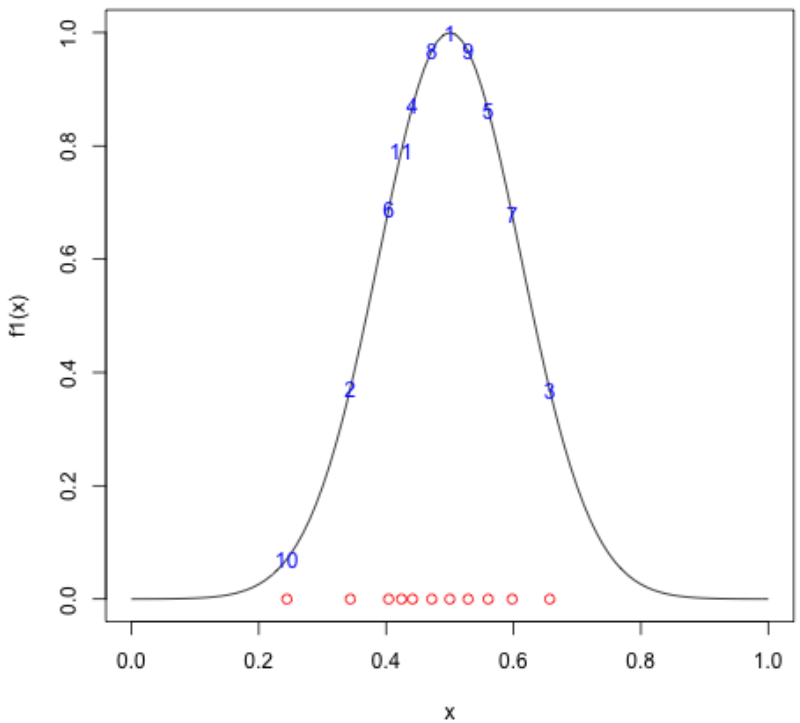
mined Package



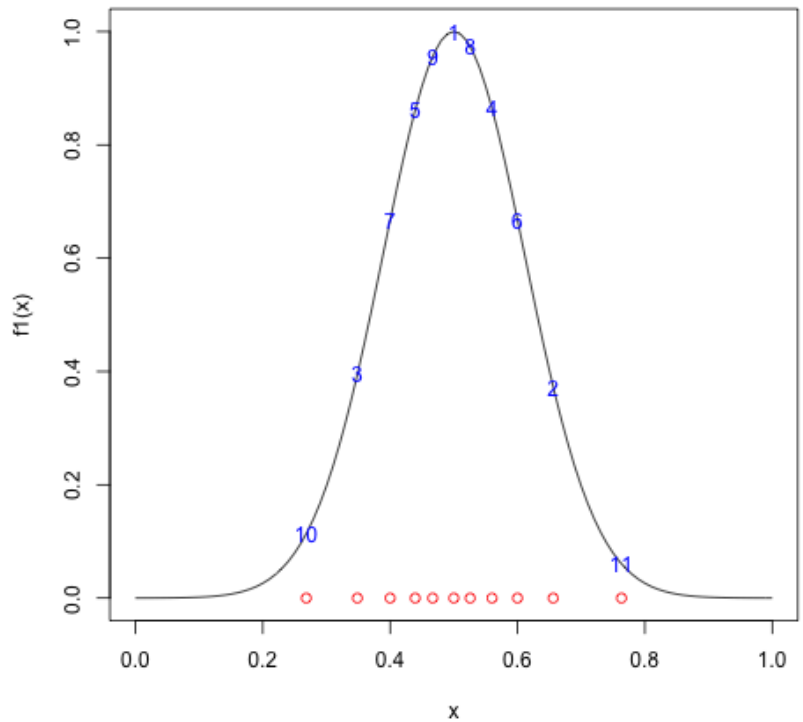
Normal Example

$$f_1 = N(1/2, 1/9)$$

$K = 20$



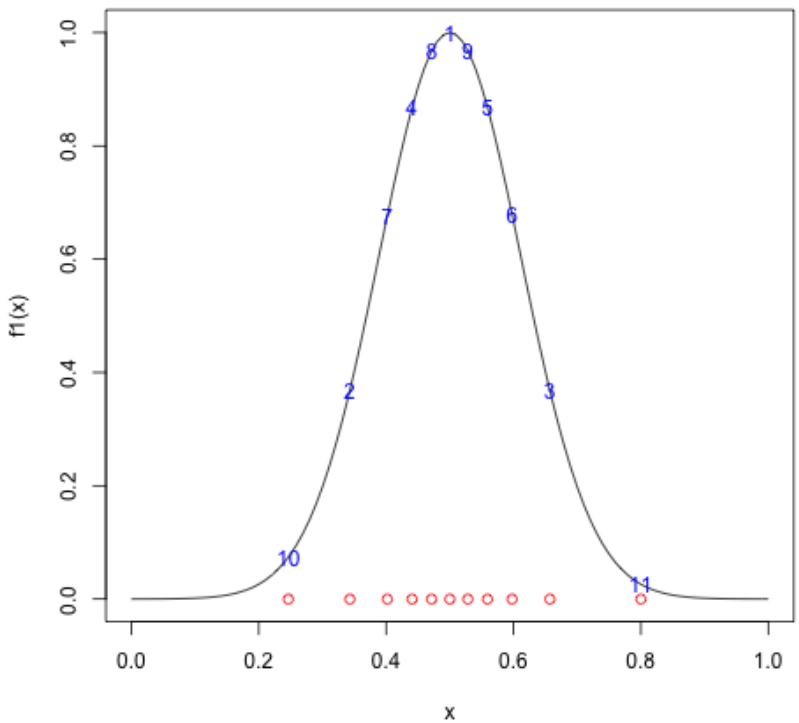
mined Package



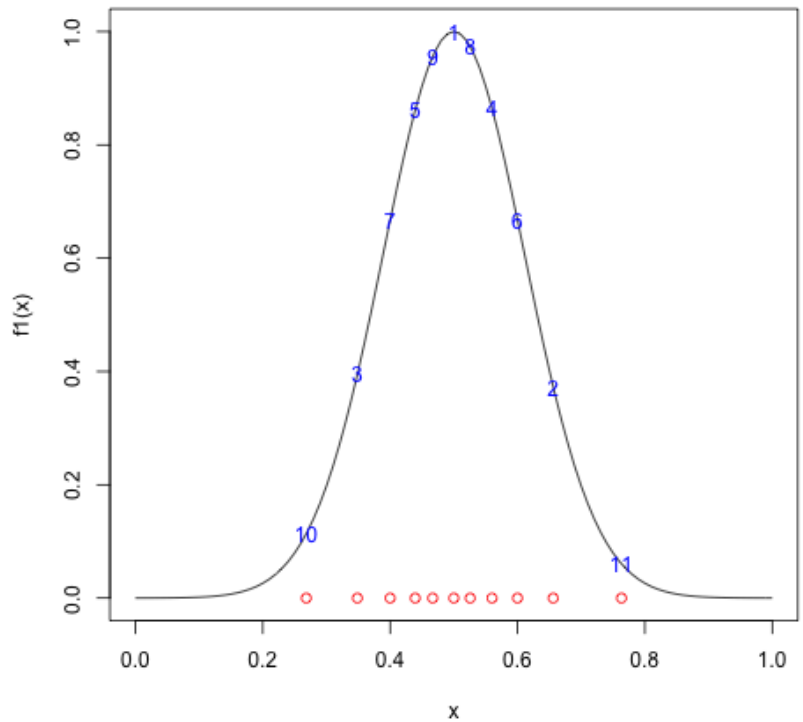
Normal Example

$f_1 = N(1/2, 1/9)$

$K = 50$



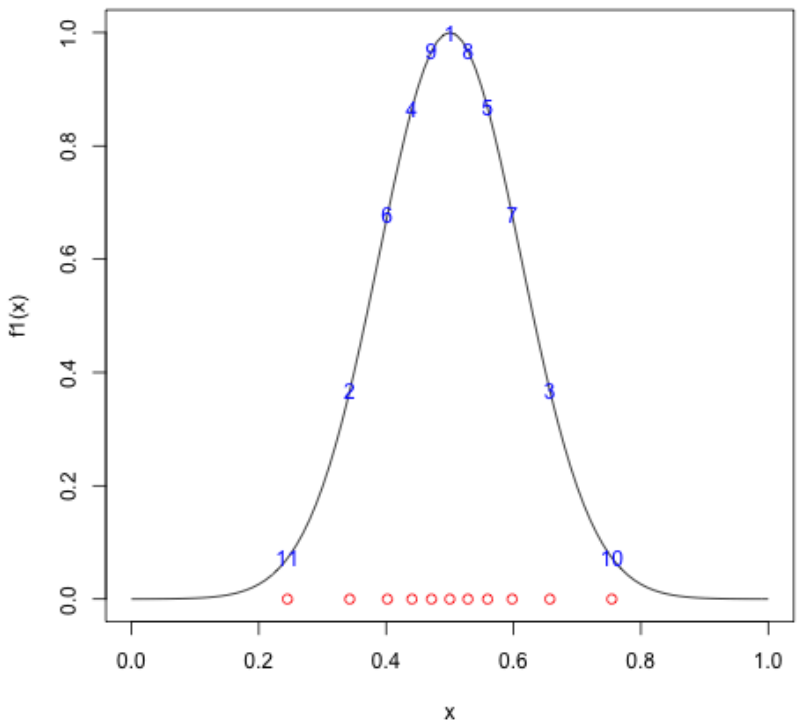
mined Package



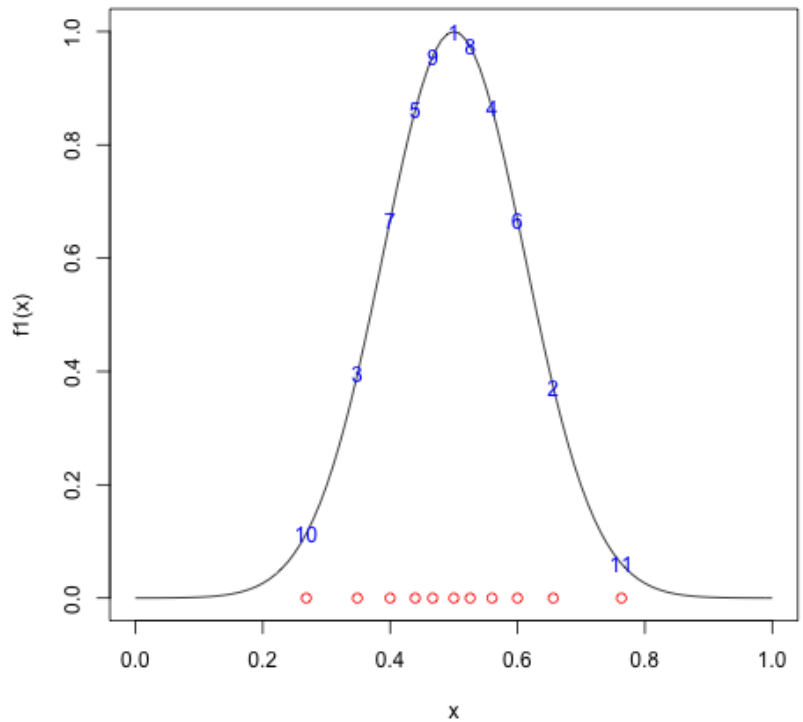
Normal Example

$$f_1 = N(1/2, 1/9)$$

$K = 100$



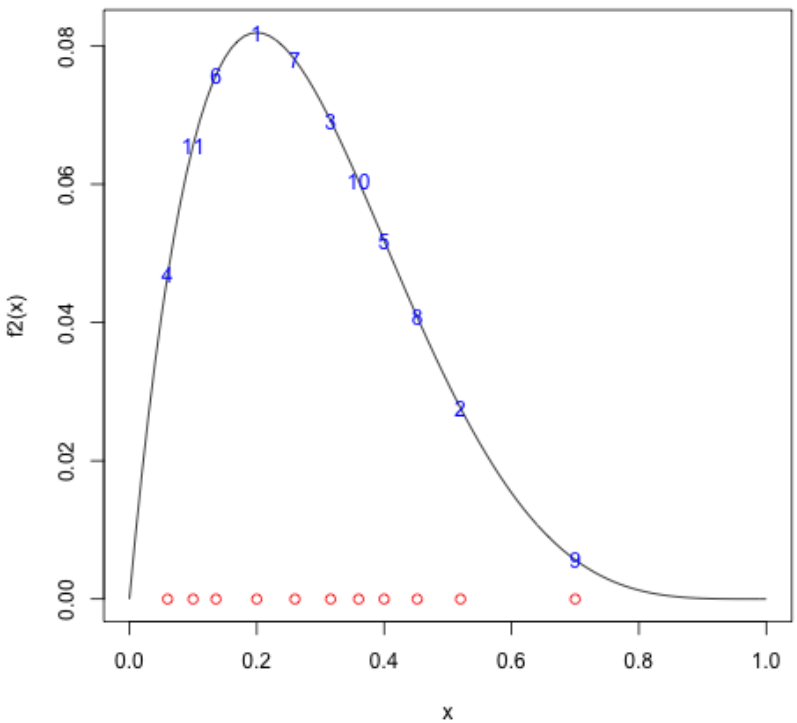
mined Package



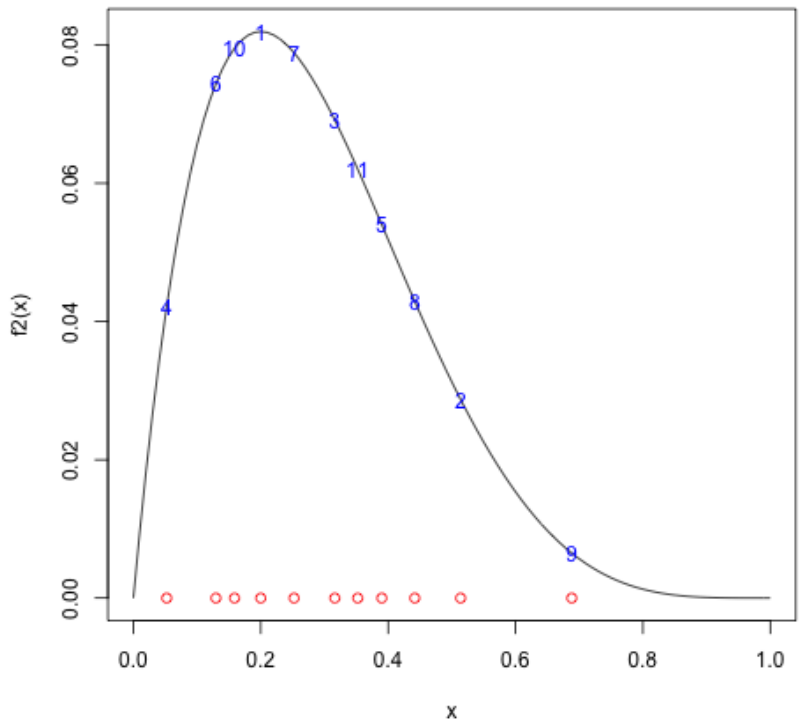
Beta Example

$f_1 = Beta(2, 5)$

$K = 4$



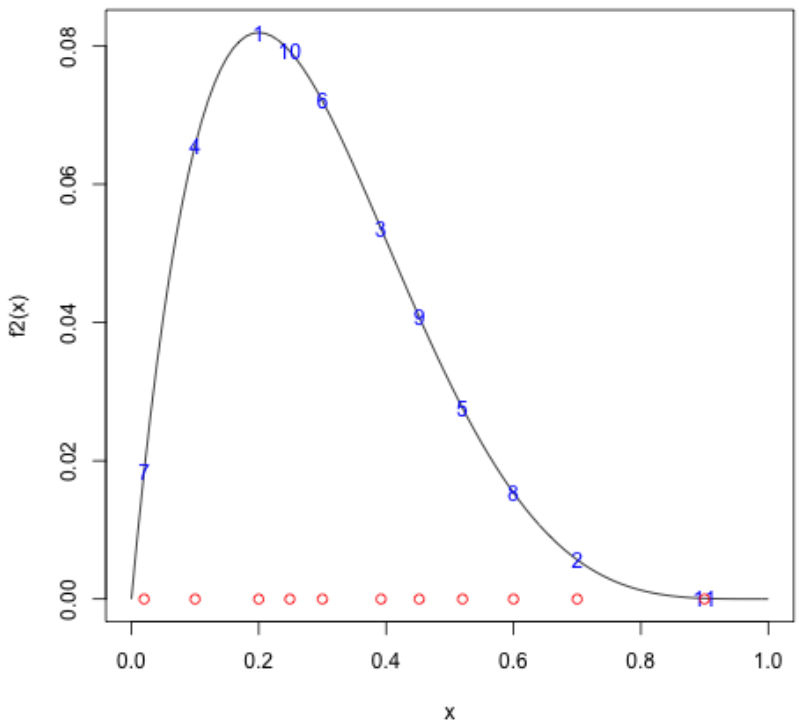
mined Package



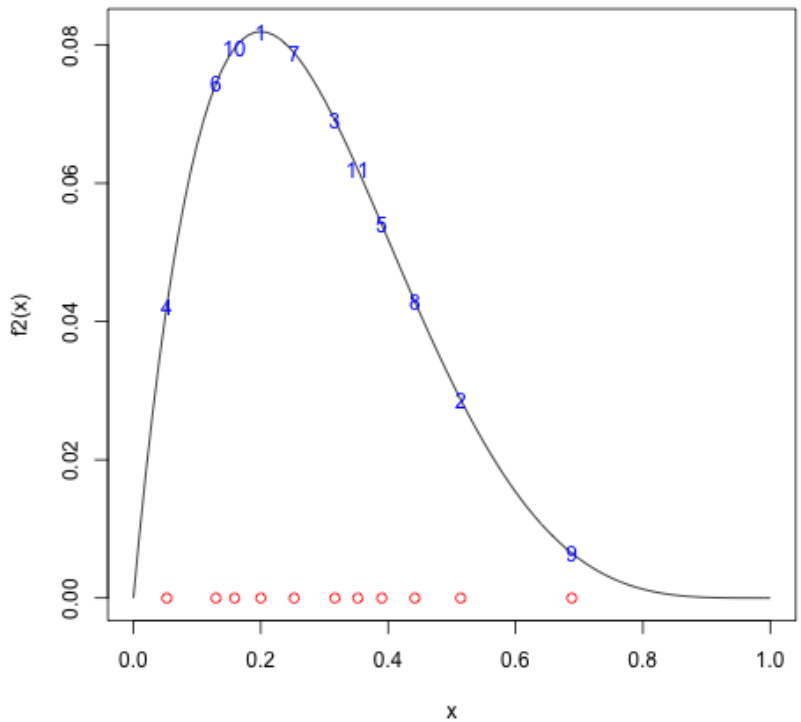
Beta Example

$f_1 = Beta(2, 5)$

$K = 10$



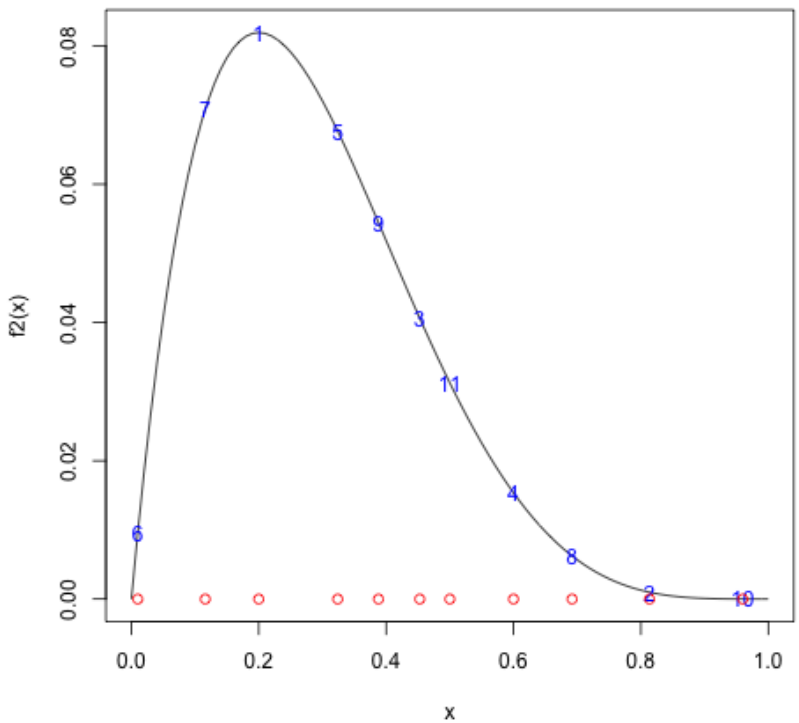
mined Package



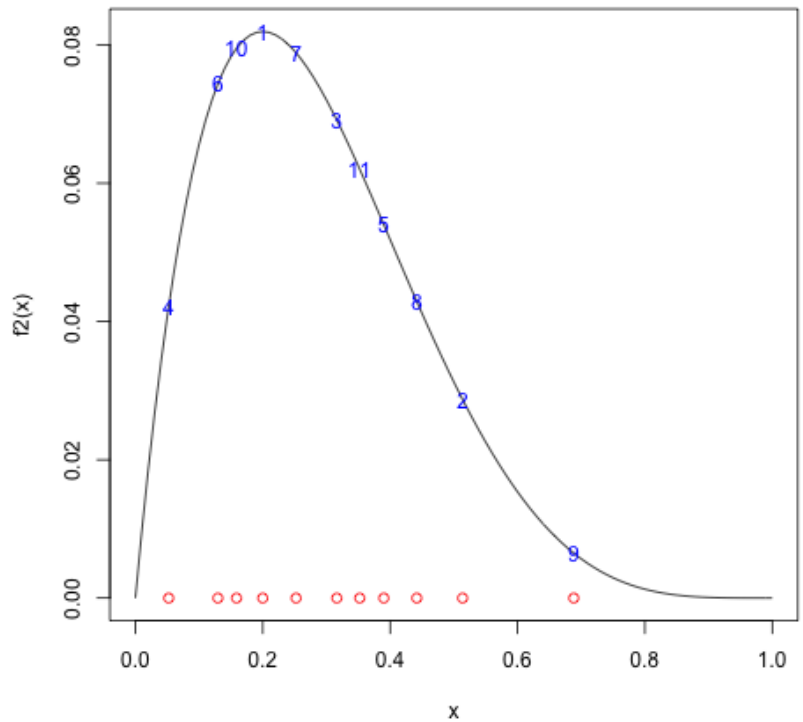
Beta Example

$f_1 = Beta(2, 5)$

$K = 20$



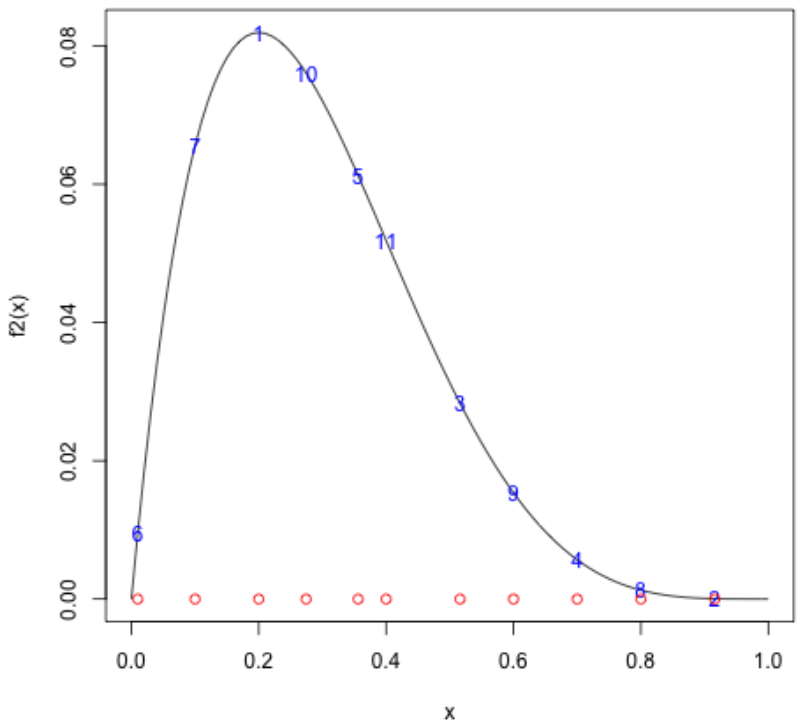
mined Package



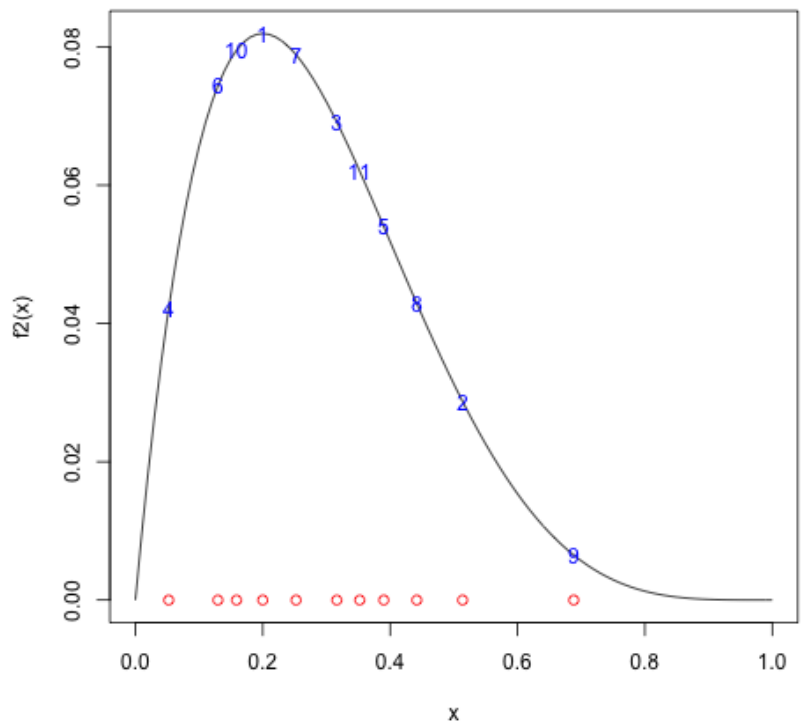
Beta Example

$f_1 = Beta(2, 5)$

$K = 50$



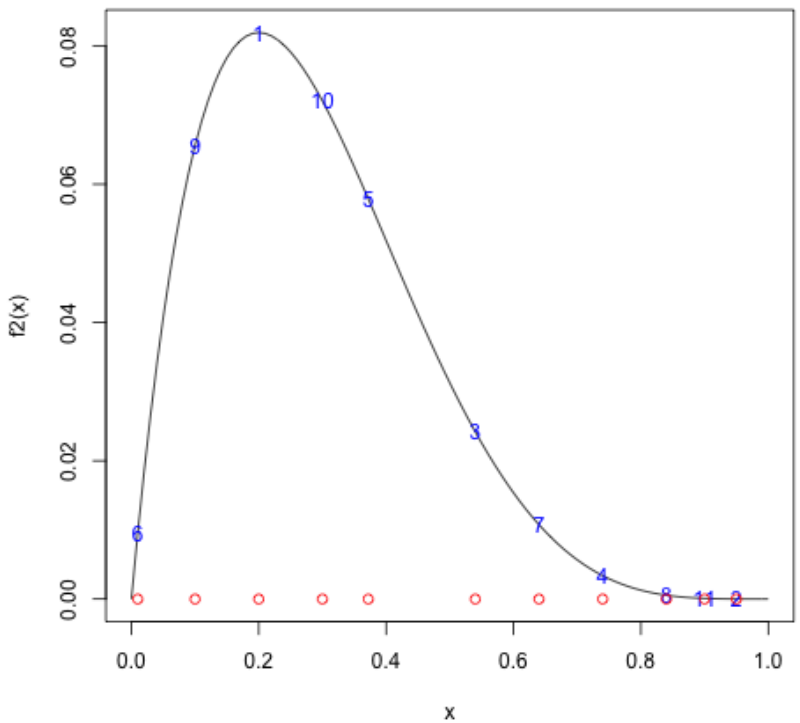
mined Package



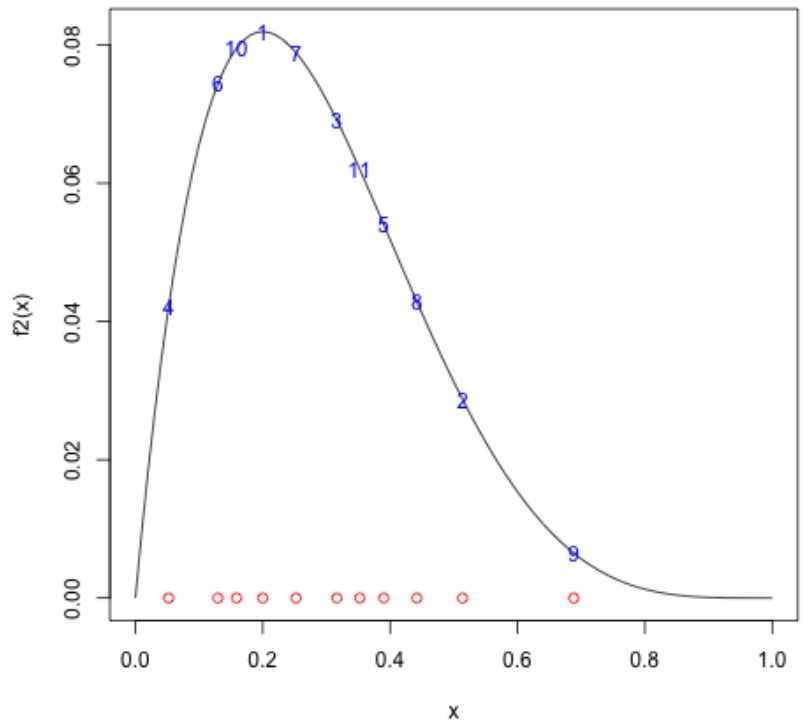
Beta Example

$f_1 = Beta(2, 5)$

$K = 100$



mined Package



Gaussian Process Model Selection

Gaussian Process Model Selection

"Fast" Algorithm Idea

"Fast" Algorithm for Gaussian Process Model Selection

Generate N-point MED $\mathbf{D}_N$ to distinguish between two Gaussian models with different covariance functions H_0 & H_1:

1. Obtain N candidate points, $\mathbf{D}_1$, in $[0,1]$ using a space-filling design.
2. For $k = 1, ..., K-1$, get next design $\mathbf{D}_{k+1}$: for $j = 1, ..., N$, Determine L_{jk} and use it to obtain N candidate points, $\tilde{\mathbf{D}}_k^j$ to get candidate set $\mathbf{C}_{k+1}^j$ for selection of design point $\mathbf{x}_j^{k+1}$, in the next design, $\mathbf{D}_{k+1}$.
   Find the design point $\mathbf{x}_j^{k+1}$ that minimizes

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

Compare GP and Linear Model Selection: Linear Case

Updating $\mathbf{x}_j^{k+1}$, jth design point of design $\mathbf{D}_{k+1}$:

Linear Model Selection:

2. To find design point $\mathbf{x}_j^{k+1}$, use a method similar to One-at-a-Time Algorithm:

$$\mathbf{x}_j^{k+1} = \arg\min_{\mathbf{x} \in \mathbf{C}_{k+1}^j} \max_{i=1:(j-1)} \frac{1}{f^{\gamma_k}(\mathbf{x}_i) f^{\gamma_k}(\mathbf{x}) d(\mathbf{x}_i, \mathbf{x})} \tag{2}$$

where $\gamma_k = k/(K-1)$, $d(\cdot,\cdot)$ is Euclidean distance, $f(\mathbf{x}) = \{\text{Wasserstein}(\phi_{0,\mathbf{x}}, \phi_{1,\mathbf{x}})\}^{(1/2p)}$, and where

$$\phi_{0,\mathbf{x}} = N(\tilde{\beta}_0 \mathbf{x}, \sigma_{\epsilon_0}^2 + \mathbf{x}^2 \sigma_{\beta_0}^2),$$
$$\phi_{1,\mathbf{x}} = N(\tilde{\beta}_1 \mathbf{x}, \sigma_{\epsilon_1}^2 + \mathbf{x}^2 \sigma_{\beta_1}^2)$$

Compare GP and Linear Model Selection: GP Case

Updating $\mathbf{x}_j^{k+1}$, jth design point of design $\mathbf{D}_{k+1}$:

Gaussian Process Model Selection:

2. To find design point $\mathbf{x}_j^{k+1}$, use a method similar to One-at-a-Time Algorithm:

$$\mathbf{x}_j^{k+1} = \underset{\mathbf{x}\in\mathbf{C}_{k+1}^j}{\arg\min} \max_{i\neq j} \frac{1}{f^{\gamma_k}(\mathbf{x}_i) f^{\gamma_k}(\mathbf{x}) d(\mathbf{x}_i, \mathbf{x})} \tag{3}$$

where $\phi_{m,\mathbf{x}}$, for $m = 0, 1$, is the posterior predictive distribution $y|y_{-j}^{(t)}, X_{-j}, \mathbf{x}$:

$$y|y_{-j}^{(t)}, X_{-j}, \mathbf{x} \sim \phi_{m,\mathbf{x}} = N(M_{m,\mathbf{x}}, \Sigma_{m,\mathbf{x}}),$$

$$M_{m,\mathbf{x}} = K_m(\mathbf{x}, X_{-j}) K_m(X_{-j}, X_{-j})^{-1} y_{-j}^{(t)}$$

$$\Sigma_{m,\mathbf{x}} = K_m(\mathbf{x}, \mathbf{x}) - K_m(\mathbf{x}, X_{-j}) K_m(X_{-j}, X_{-j})^{-1} K_m(X_{-j}, \mathbf{x})$$

...

Compare GP and Linear Model Selection: GP Case, continued

...

and so $f(\mathbf{x}) = \text{Wasserstein}(\phi_{0,\mathbf{x}}, \phi_{1,\mathbf{x}})\}^{(1/2p)}$, is estimated from T simulations of $y_{-j}^{(t)}$, for $t = 1, \ldots, T$, as follows:

i Draw $\ell \sim \text{Cat}(H_0, H_1; \pi_0, \pi_1)$, where $\pi_0 = \pi_1 = 1/2$.

ii Draw $y_{-j}^{(t)} \sim N(0, K_\ell(X_{-j}, X_{-j}))$

iii Evaluate $M_{m,\mathbf{x}}$ and $\Sigma_{m,\mathbf{x}}$ for $m = 0, 1$.

iv Calculate Wasserstein$(\phi_{0,\mathbf{x}}, \phi_{1,\mathbf{x}})^{(t)}$. Here, the Wasserstein distance is between two multivariate Gaussians.

After T simulations, calculate
$\bar{\omega} = \frac{1}{T} \sum_{t=1}^{T} \text{Wasserstein}(\phi_{0,\mathbf{x}}, \phi_{1,\mathbf{x}})^{(t)}$
Then $f(\mathbf{x}) = \{\text{Wasserstein}(\phi_{0,\mathbf{x}}, \phi_{1,\mathbf{x}})\}^{(1/2p)} \approx \{\bar{\omega}\}^{1/(2p)}$