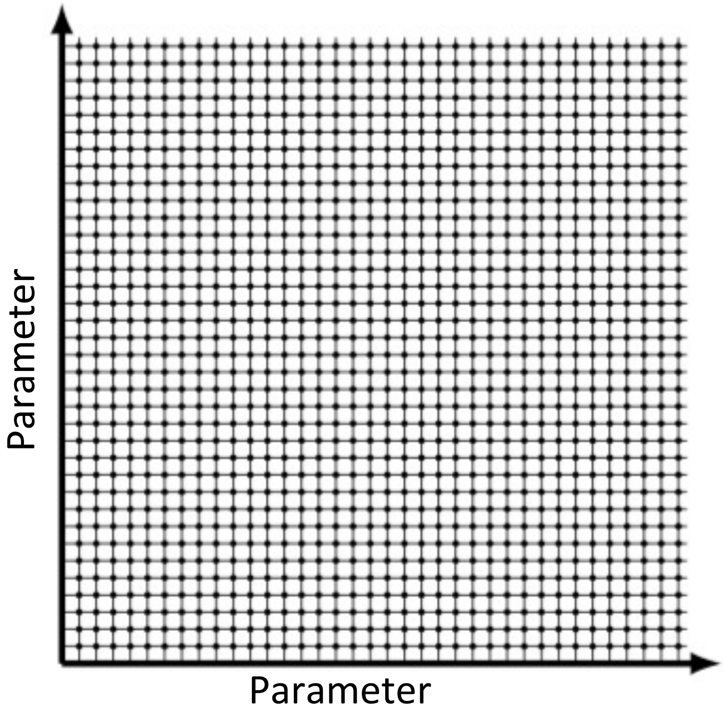
Lecture 10:
Modeling Data Surrogate Based Modeling, Hybrid Piecewise Polynomial Modeling, and Random Forest

COSC 526: Introduction to Data Mining
Spring 2020

THE UNIVERSITY OF TENNESSEE
KNOXVILLE
BIG ORANGE. BIG IDEAS.®

Exhaustive Search

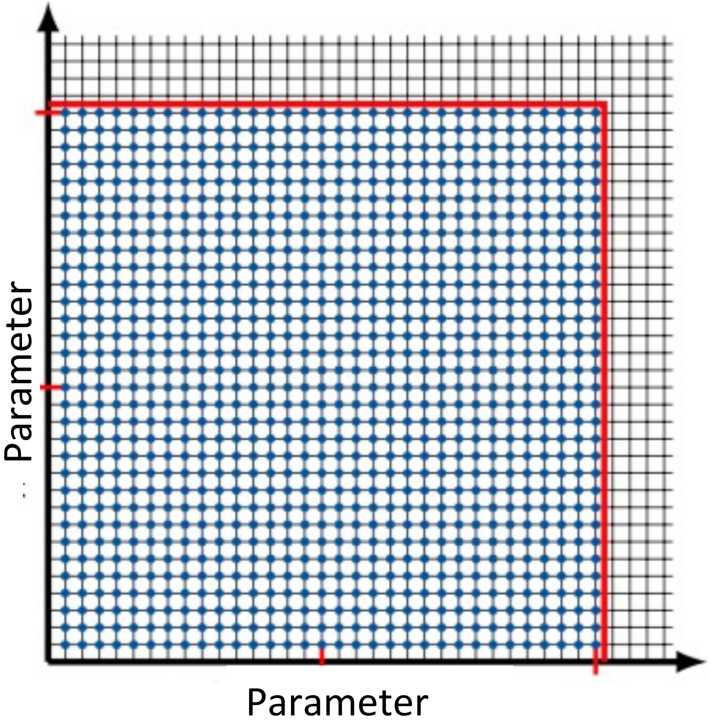
Exhaustive Search



Step 1: Reduce to finite search space

P. K. Lakkimsetti, A Framework for Automatic Optimization of MapReduce Programs Based on Job Parameter Configurations, M. Thesis, Kansas State University, 2011.

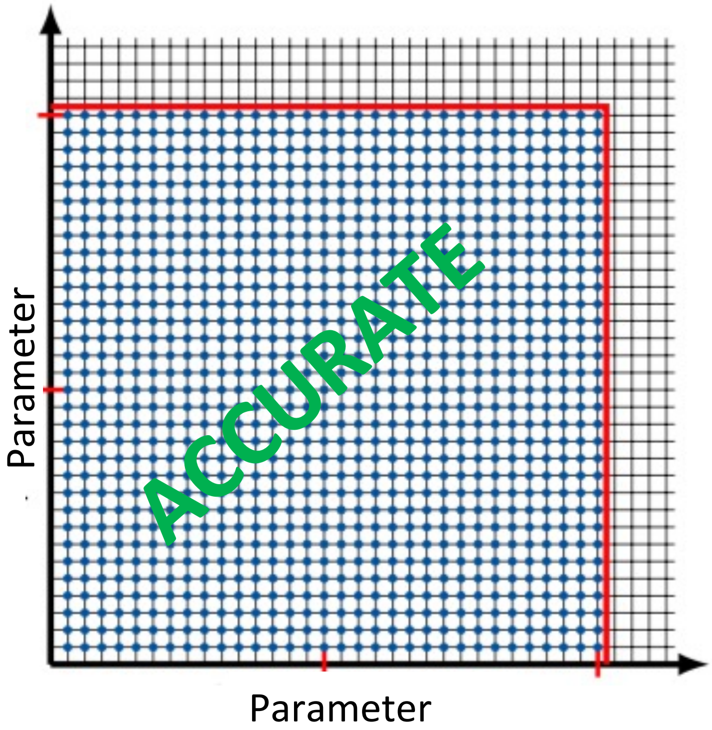
Exhaustive Search



Step 2: Sample all these points

P. K. Lakkimsetti, A Framework for Automatic Optimization of MapReduce Programs Based on Job Parameter Configurations, M. Thesis, Kansas State University, 2011.

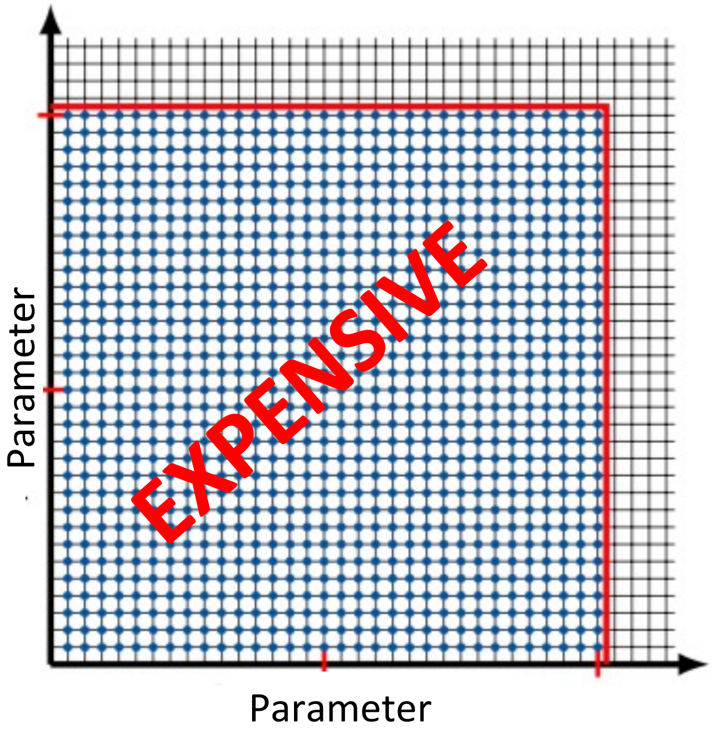
Exhaustive Search



Step 2: Sample all these points

P. K. Lakkimsetti, A Framework for Automatic Optimization of MapReduce Programs Based on Job Parameter Configurations, M. Thesis, Kansas State University, 2011.

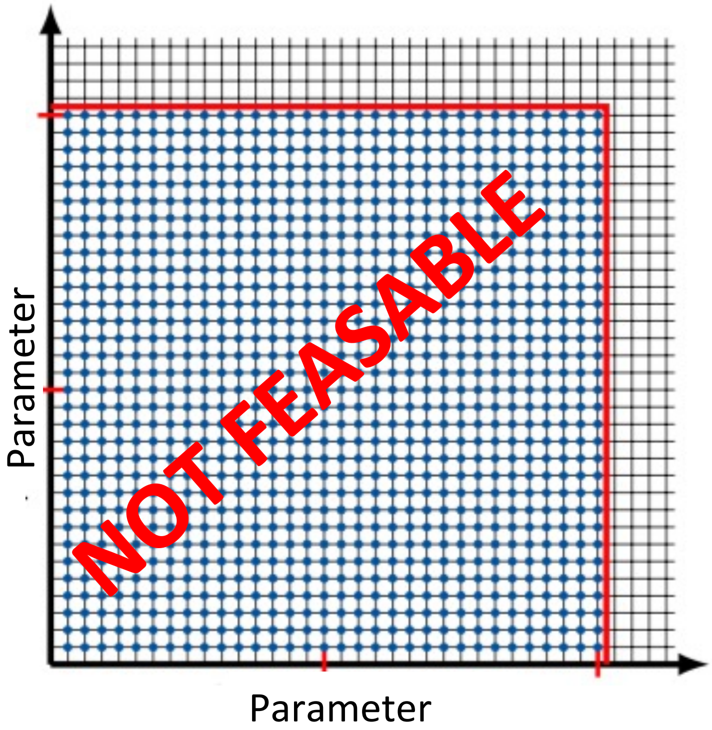
Exhaustive Search



Step 2: Sample all these points **EXPENSIVE**

- Sample the entire population of the United States to leaner what each individual eats

Exhaustive Search



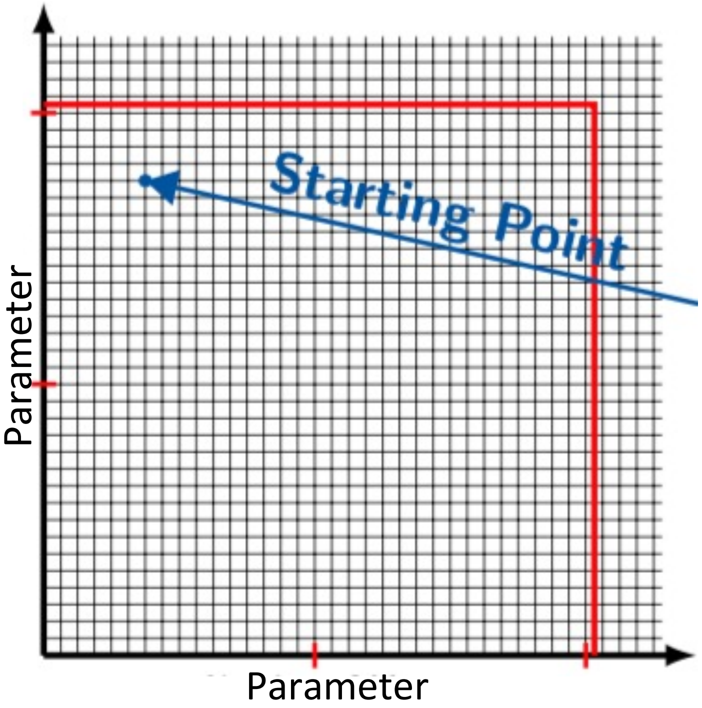
Step 2: Sample all these points **NOT FEASABLE**

- Satellites do not collect all the points across the globe

P. K. Lakkimsetti, A Framework for Automatic Optimization of MapReduce Programs Based on Job Parameter Configurations, M. Thesis, Kansas State University, 2011.

Local Search

Local Searching Algorithms



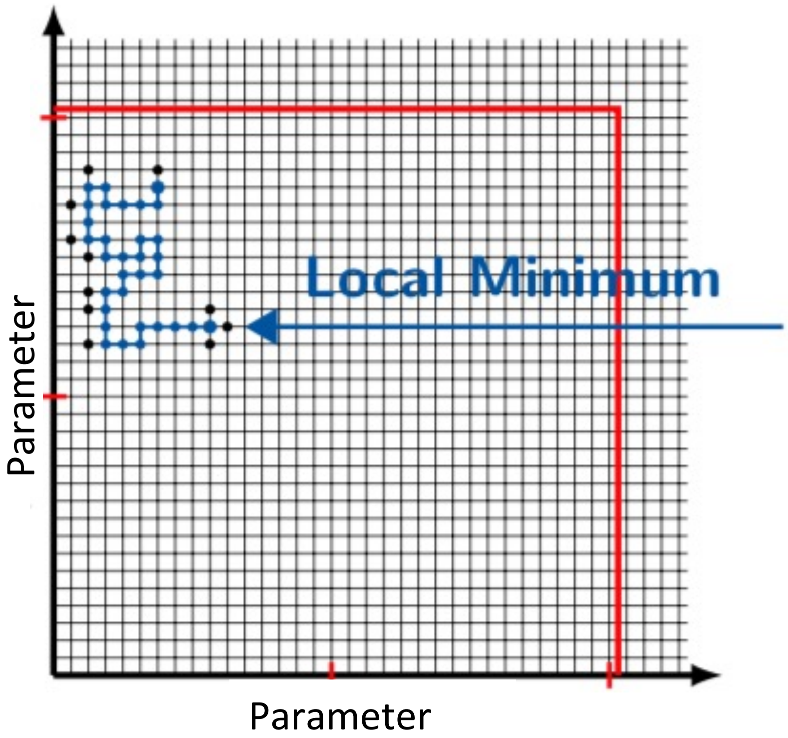
Methods: Grid Hill [1] and Simulated Annealing [2]

Step 1: Reduce to finite search space

Step 2: Randomly choose starting point and sample neighbors

[1] K. Wang, X. Lin, W. Tang, Predator-An Experience Guided Configuration Optimizer..., IEEE CloudCom 2012.
[2] D. Wu, A Profiling and Performance Analysis Based Self-tuning System for Optimization..., M. Thesis, Vanderbilt, 2013.

Local Searching Algorithms



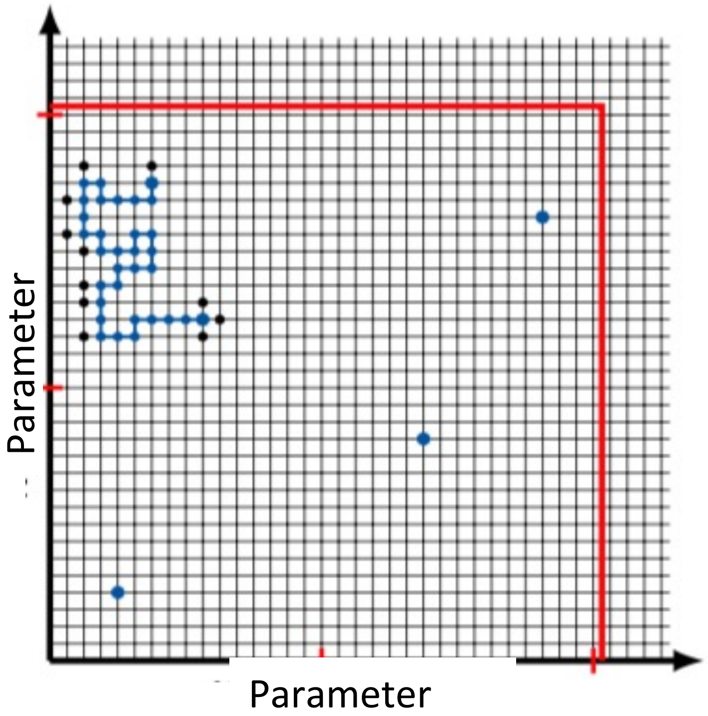
Methods: Grid Hill [1] and Simulated Annealing [2]

Step 1: Reduce to finite search space

Step 2: Randomly choose starting point and sample neighbors

[1] K. Wang, X. Lin, W. Tang, Predator-An Experience Guided Configuration Optimizer..., IEEE CloudCom 2012.
[2] D. Wu, A Profiling and Performance Analysis Based Self-tuning System for Optimization..., M. Thesis, Vanderbilt, 2013.

Local Searching Algorithms

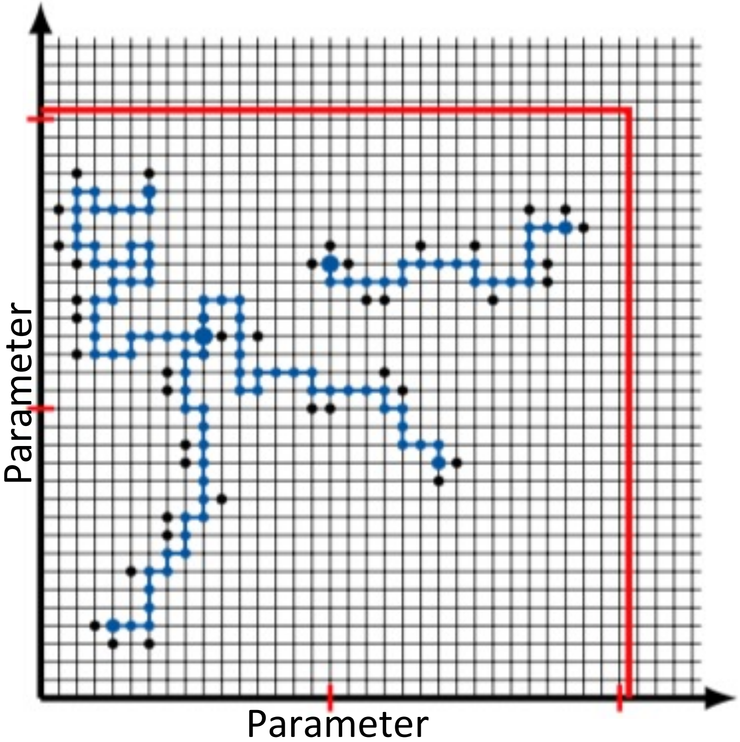


Methods: Grid Hill [1] and Simulated Annealing [2]

Step 3: Repeat Step 2 with new starting point(s)

[1] K. Wang, X. Lin, W. Tang, Predator-An Experience Guided Configuration Optimizer..., IEEE CloudCom 2012.
[2] D. Wu, A Profiling and Performance Analysis Based Self-tuning System for Optimization..., M. Thesis, Vanderbilt, 2013.

Local Searching Algorithms

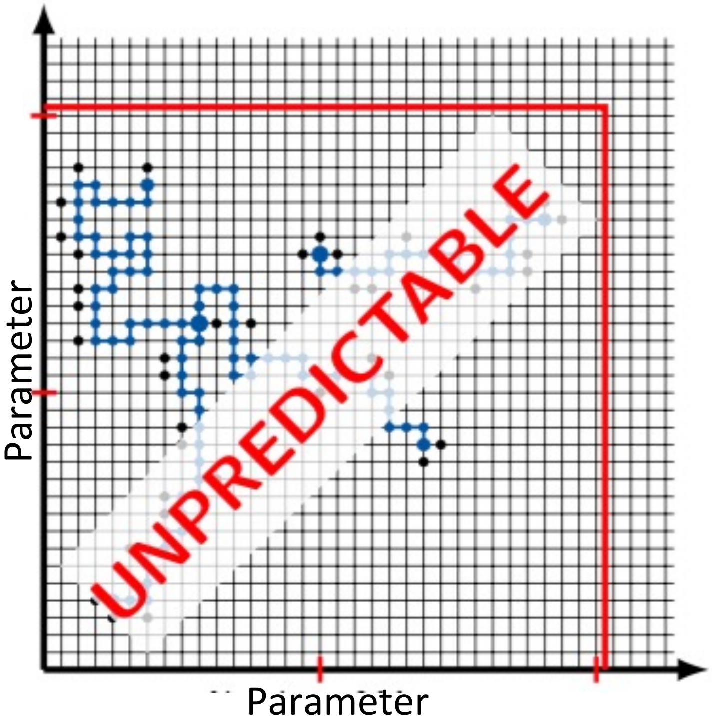


Methods: Grid Hill [1] and Simulated Annealing [2]

Step 3: Repeat Step 2 with new starting point(s)

[1] K. Wang, X. Lin, W. Tang, Predator-An Experience Guided Configuration Optimizer..., IEEE CloudCom 2012.
[2] D. Wu, A Profiling and Performance Analysis Based Self-tuning System for Optimization..., M. Thesis, Vanderbilt, 2013.

Local Searching Algorithms



Methods: Grid Hill [1] and Simulated Annealing [2]

Step 3: Repeat Step 2 with new starting point(s)

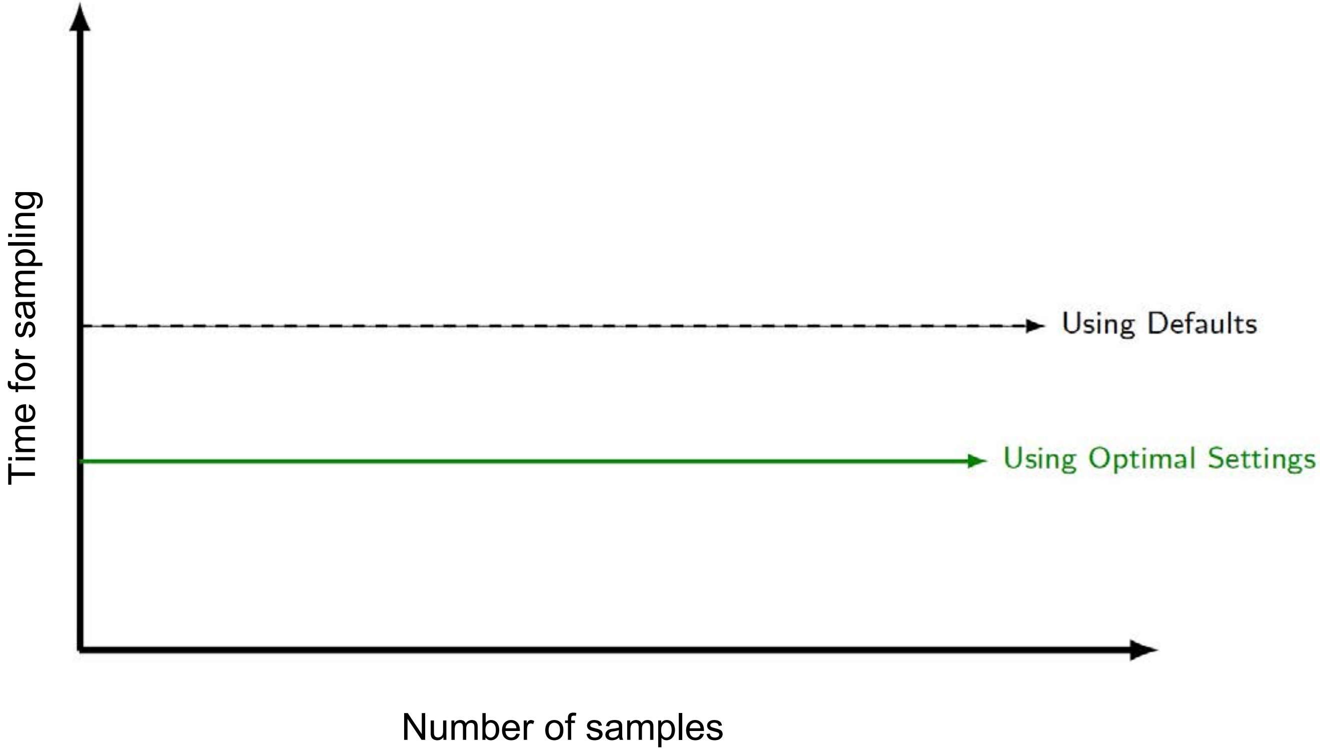
[1] K. Wang, X. Lin, W. Tang, Predator-An Experience Guided Configuration Optimizer..., IEEE CloudCom 2012.
[2] D. Wu, A Profiling and Performance Analysis Based Self-tuning System for Optimization..., M. Thesis, Vanderbilt, 2013.

Exhaustive and Local Searches

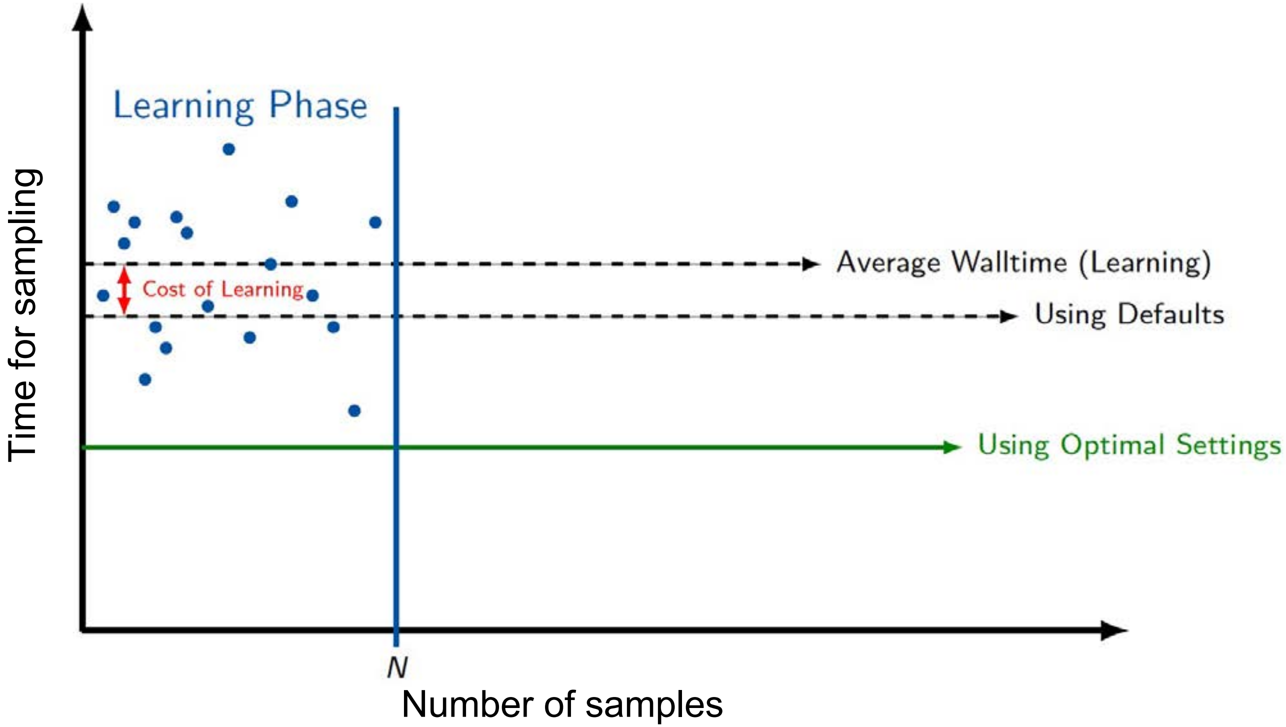
- Exhaustive sampling is too expensive
- Local search algorithms (LSAs) sample fewer points but are unpredictable

Cost of Sampling

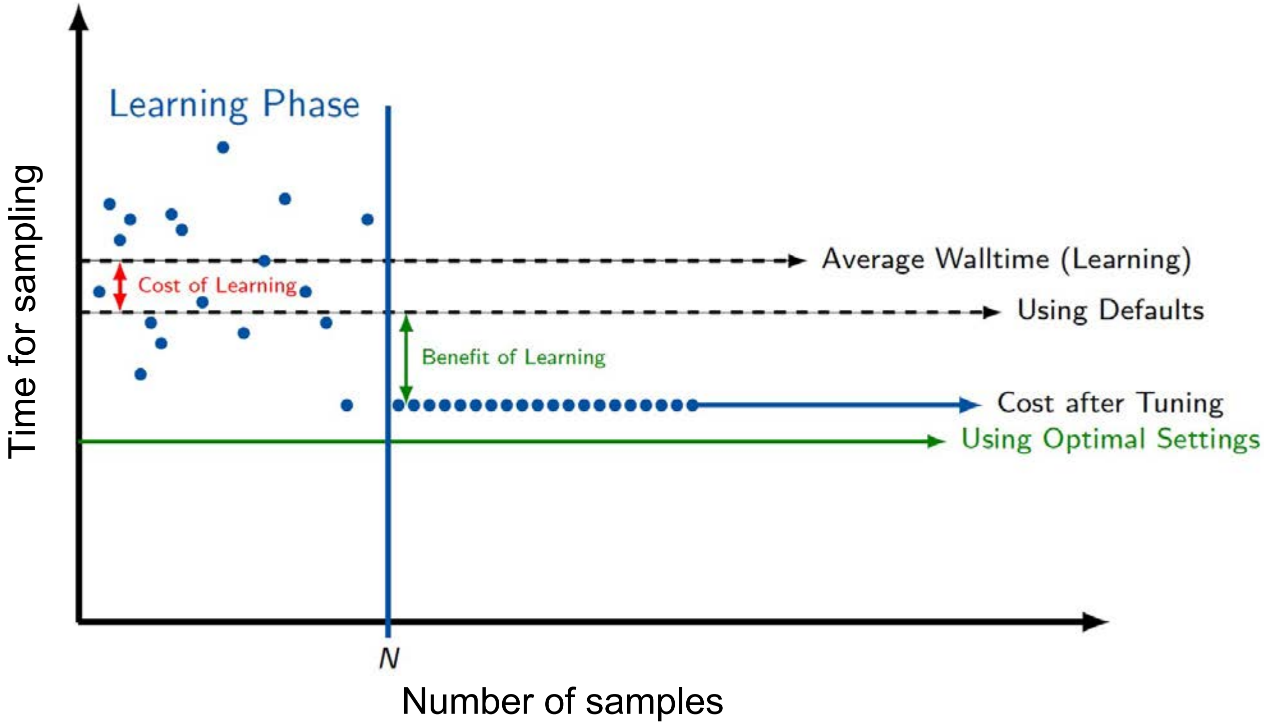
Amortizing the Cost of Sampling



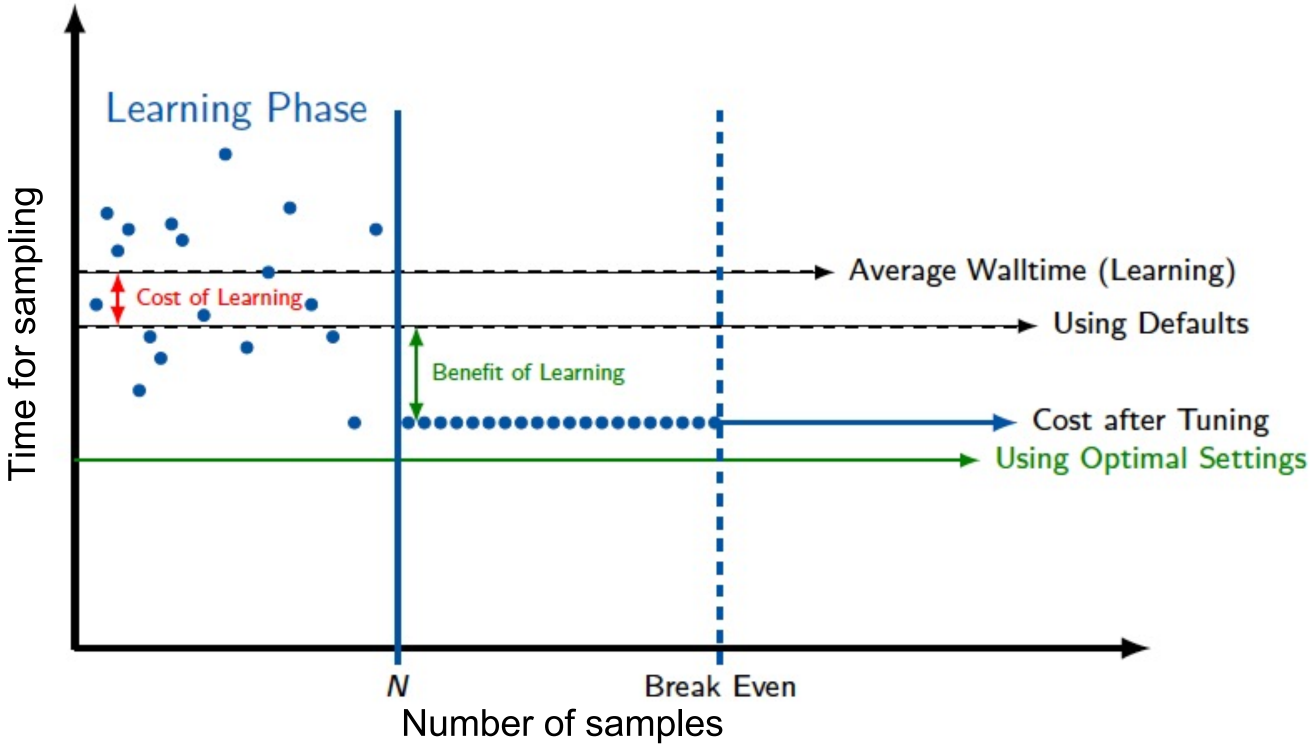
Amortizing the Cost of Sampling



Amortizing the Cost of Sampling



Amortizing the Cost of Sampling

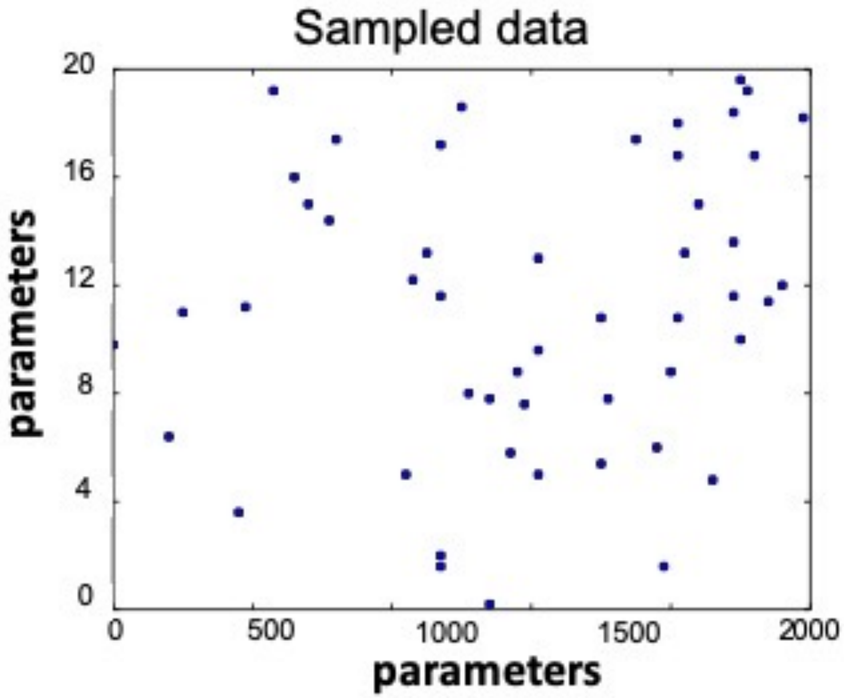


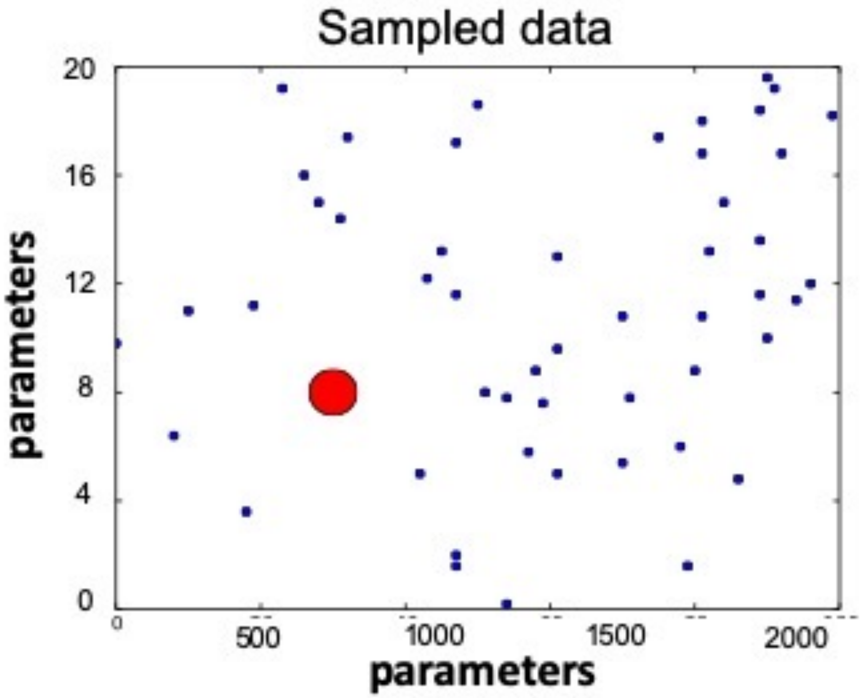
Amortizing the Cost of Sampling

- How do we focus the learning phase to maximize the return on our investment?

k Nearest Neighbors or kNN

k Nearest Neighbors Model

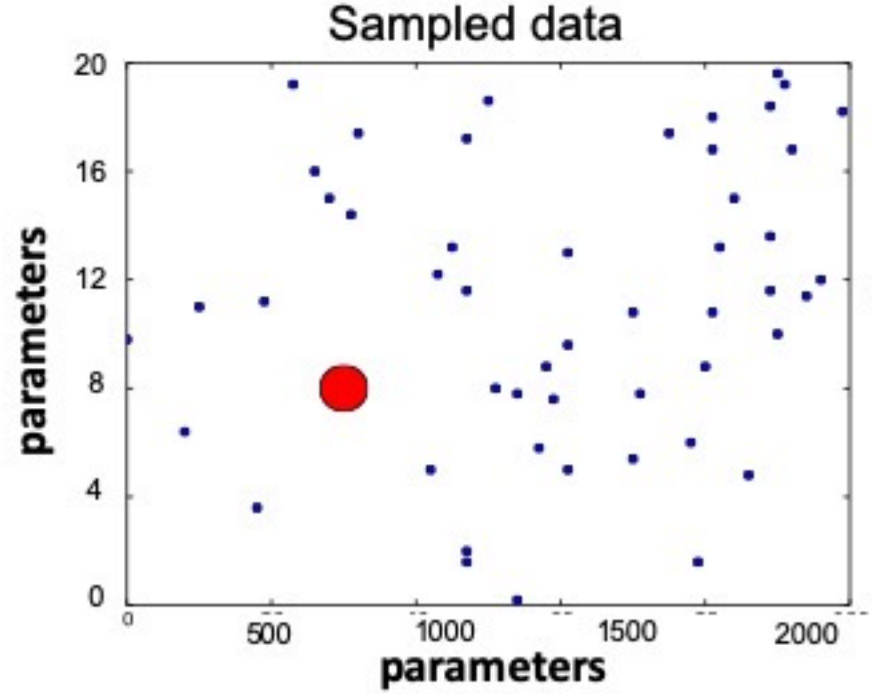


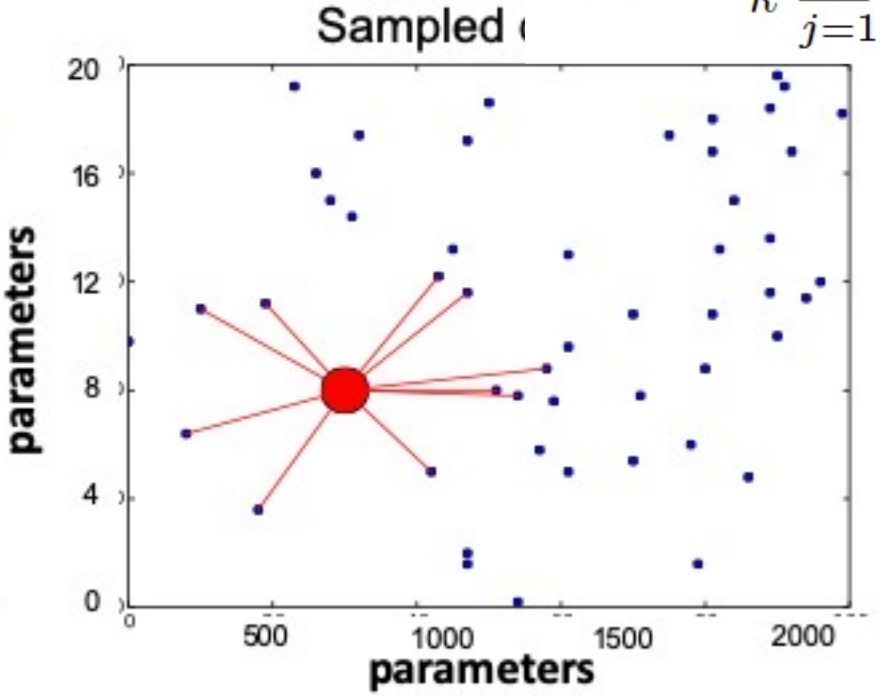


k Nearest Neighbors Model

Polynomial with degree zero

$$z(\vec{x}) := \frac{1}{k}\sum_{j=1}^{k} z_{i_j}$$





kNN

KNN:

→ Use **local data**

→ Compute k and distance kernel using cross validation automatically

→ Compute weighted means with the kernel (***many values***)

0	0	0	0

0	0	0	0	0

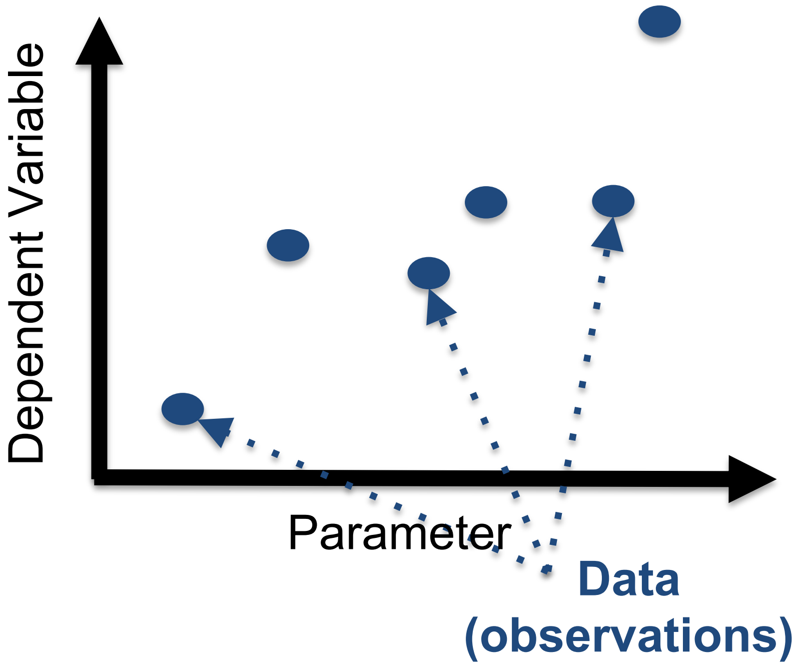
D. Rorabaugh, M. Guevara, R. Llamas, J. Kitson, R. Vargas, and **M. Taufer**. SOMOSPIE: A Modular SOil MOisture SPatial Inference Engine based on Data Driven Decisions. eScinece 2019.

Surrogate-Based Modeling

Using Surrogate-Based Modeling

- The surrogate model may predict optimal configurations that were never sampled in the learning phase
- We can explicitly determine the number of points required to build a surrogate model

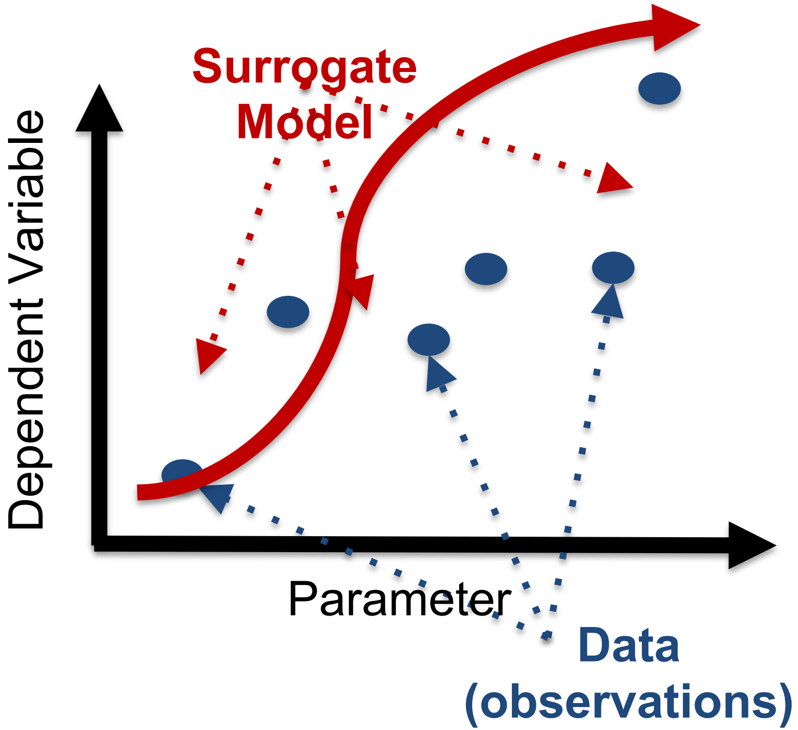
Surrogate-Based Modeling



- Data → **all** sampled data to create a **single global model**
- Model → fit a **polynomial** to data (continuous and differentiable)
- Best for→ finding underlying **global trends** when they exist

Image source: Johnston, R., Taufer

Surrogate-Based Modeling

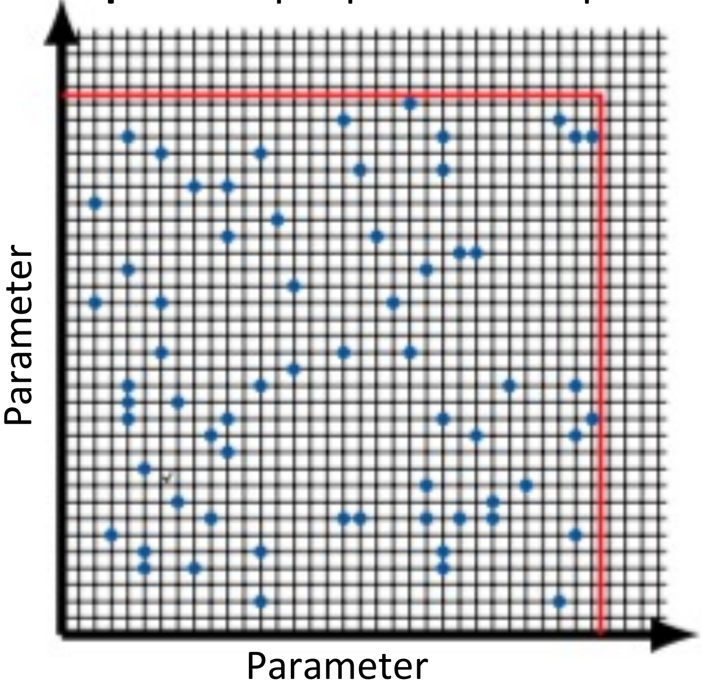


- Data → **all** sampled data to create a **single global model**
- Model → fit a **polynomial** to data (continuous and differentiable)
- Best for→ finding underlying **global trends** when they exist

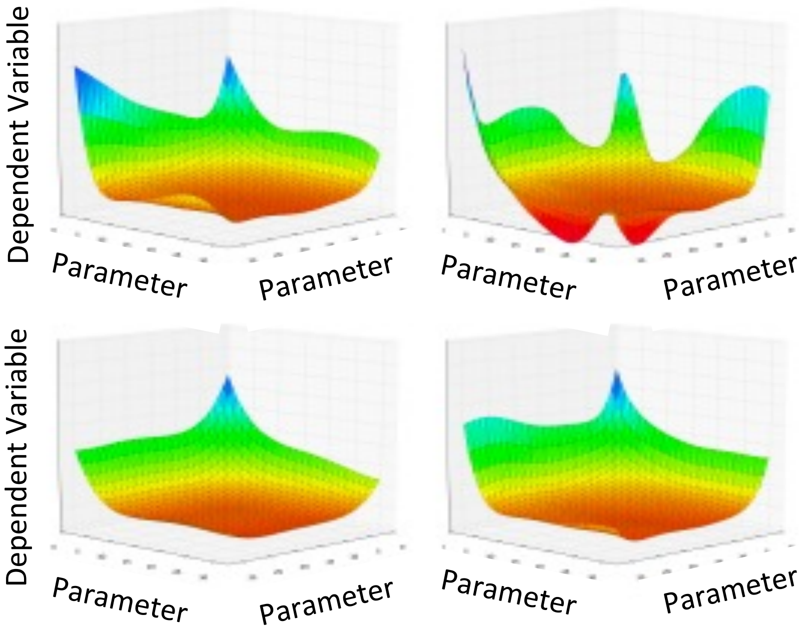
Image source: Johnston, R., Taufer

Surrogate-Based Modeling

Step 1: Sample parameter space

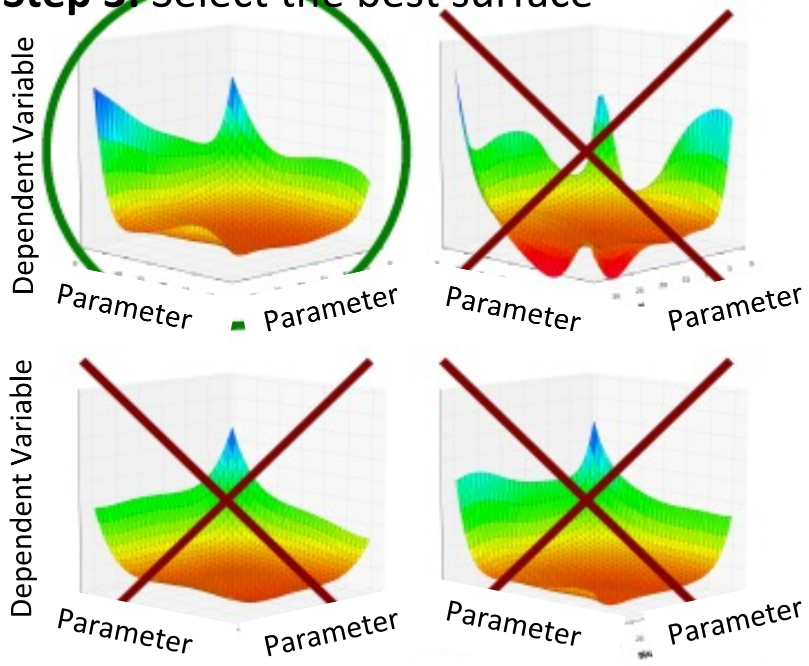


Step 2: Build candidate surfaces

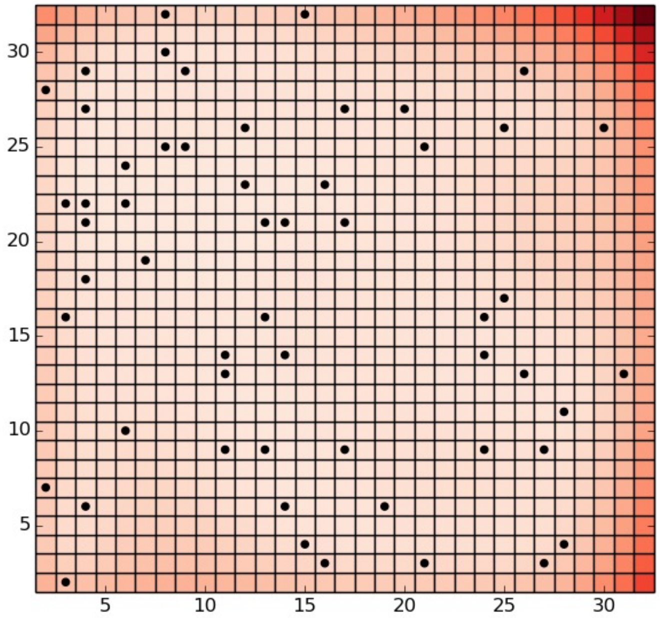


Surrogate-Based Modeling

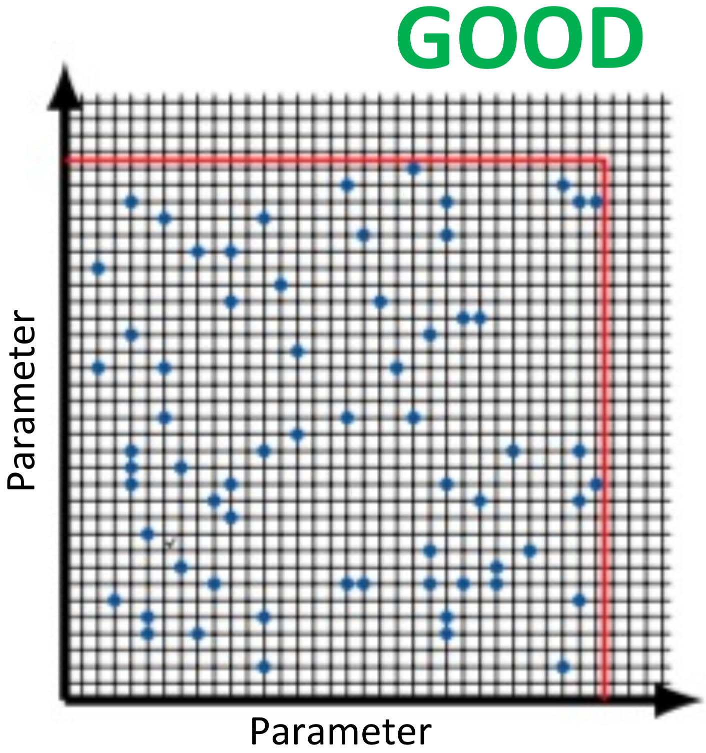
Step 3: Select the best surface

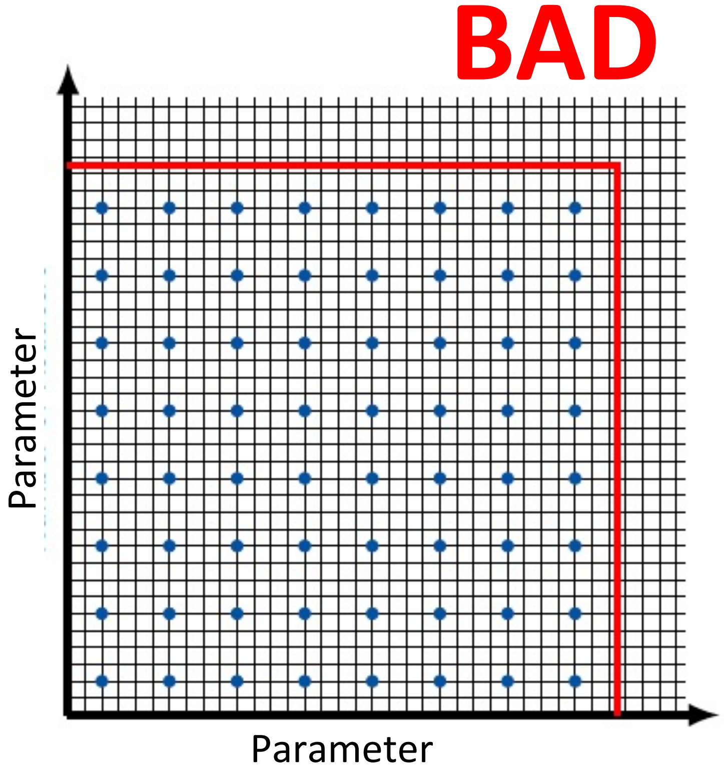


Step 4: Apply confidence interval



Step 1: Random sampling





Step 2: Building candidate surfaces

What kind of surfaces do we build?

- We represent our **surface** by a multivariate **polynomials**
- The candidate surfaces look like: **Z = B X**

- $z_1(x,y) = \beta_1 + \beta_2 x + \beta_3 y$ Degree 1
- $z_2(x,y) = \beta_1 + \beta_2 x + \beta_3 y + \beta_4 x^2 + \beta_5 xy + \beta_6 y^2$ Degree 2
- $z_3(x,y) = z_2(x,y) + \beta_7 x^3 + \beta_8 x^2 y + \beta_9 xy^2 + \beta_{10} y^3$ Degree 3

Step 2: Building candidate surfaces

Why build a polynomial surface?

- Polynomials are easy to describe and represent in memory
- Polynomials can generate quite complex surfaces
- Polynomials easily generalize to any number of variables

Step 2: Building candidate surfaces

- We construct best fit polynomial surfaces of degree *d* for each reasonable value of d, i.e. d = {1, 2, …, M} where M is small enough that the matrix X^TX is invertible

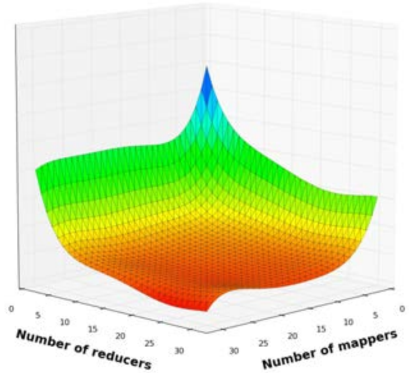
Step 2: Building candidate surfaces

How many points do we need to sample?

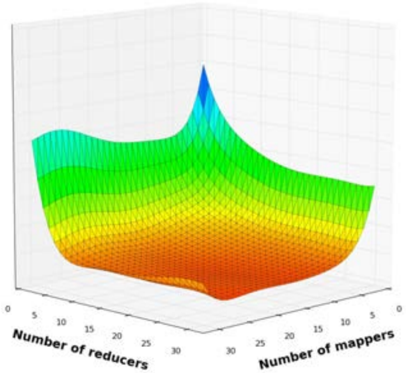
- To build the surface, we determine the coefficients B by solving the matrix equation:

  $X B = Z \dashrightarrow \mathbf{X^T X\ B = X^T Z}$

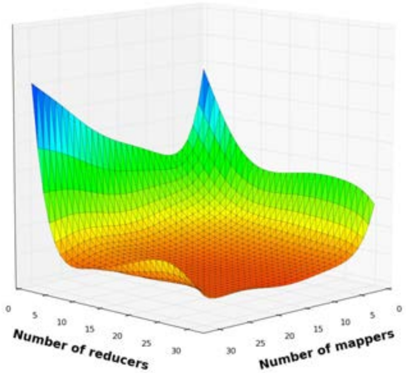
- If $\mathbf{X^T X}$ is not invertible, then there is not a unique solution for B
- If the number of samples taken is smaller than the number of terms in our polynomial, then $X^T X$ is not invertible
- To build a surface of degree d with *v* variables we need at least $\binom{d+v}{v}$



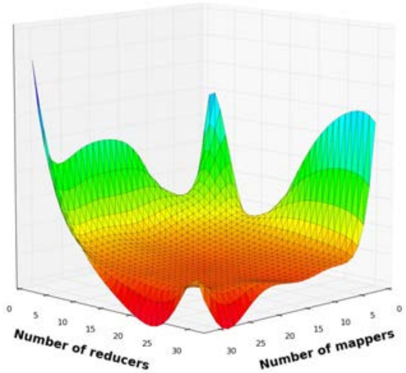
Degree 5



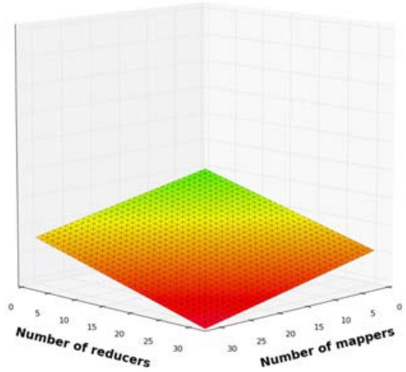
Degree 6



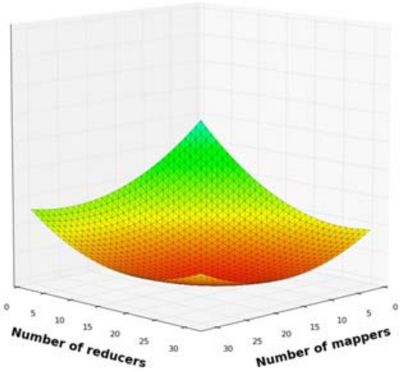
Degree 7



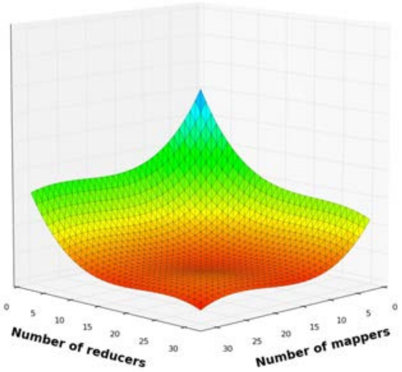
Degree 8



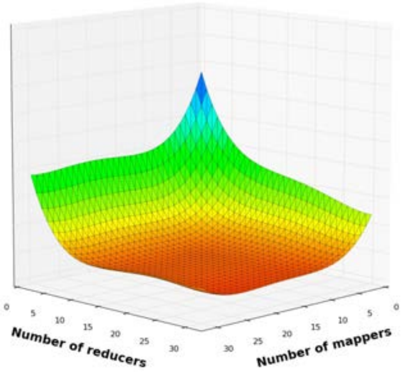
Degree 1



Degree 2

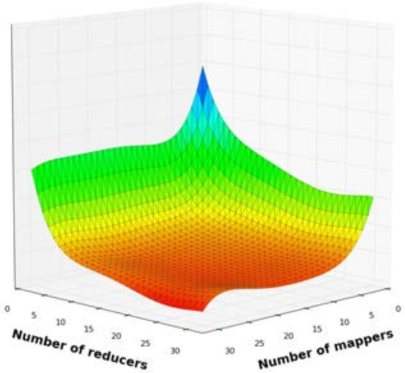


Degree 3

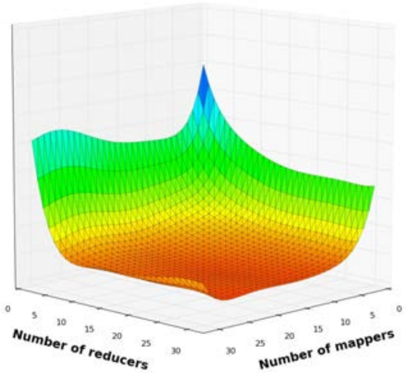


Degree 4

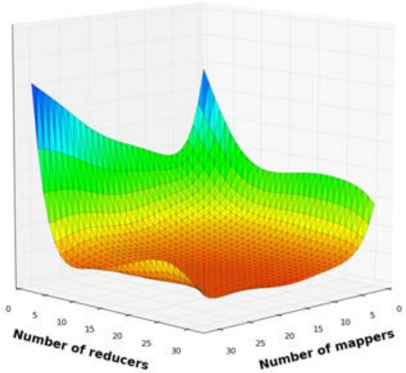
Which Model is Best?



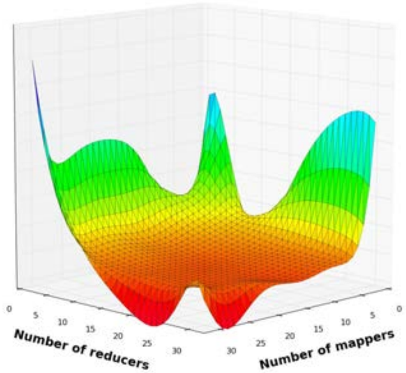
Degree 5



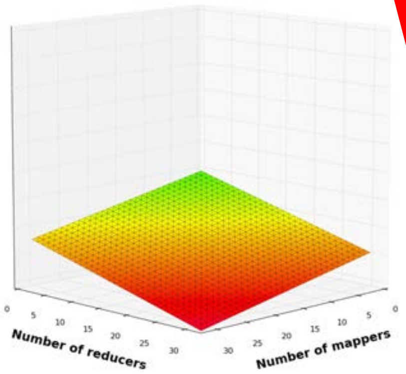
Degree 6



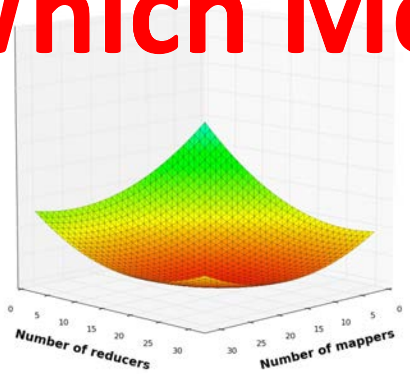
Degree 7



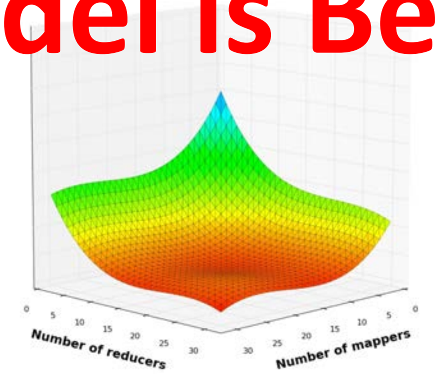
Degree 8



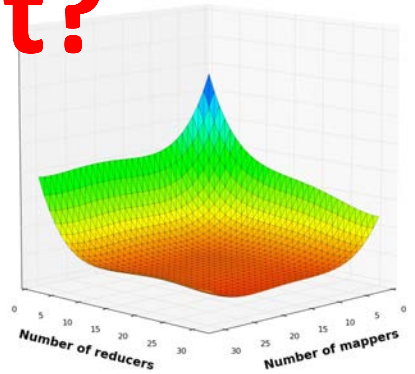
Degree 1



Degree 2



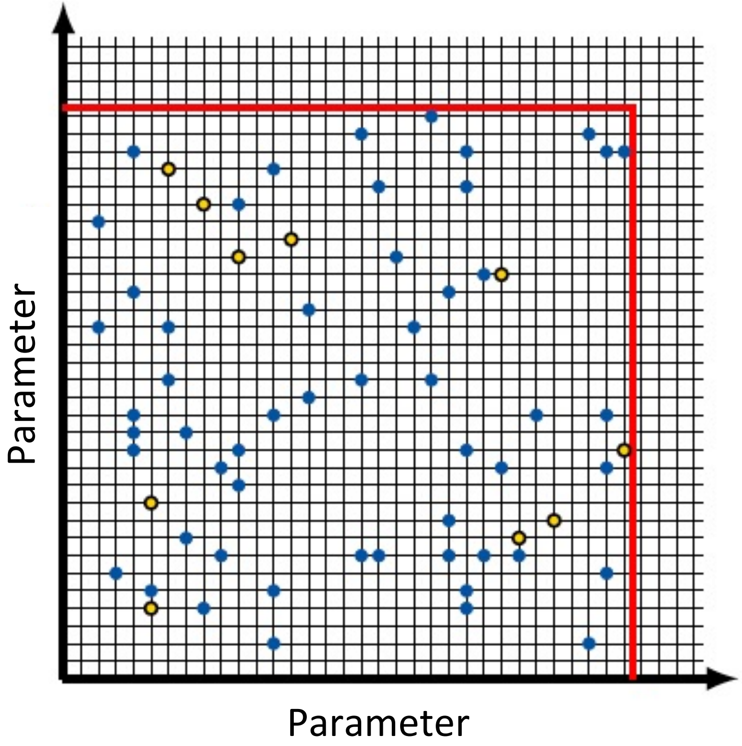
Degree 3



Degree 4

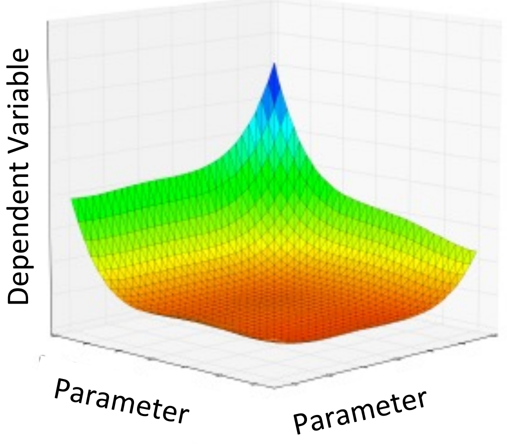
Step 3: Selecting the best surface using k-fold cross validation

- Partition samples into k sets of equal size
- k -1 sets are used for learning; one is use for testing
- Build best fit polynomial surface from learning set
- Use polynomial surface to predict unknown variables in testing set
- Use points in the testing set to compute SSE



Partition 1

- Learning Sets(k -1)
- Testing Set (1)

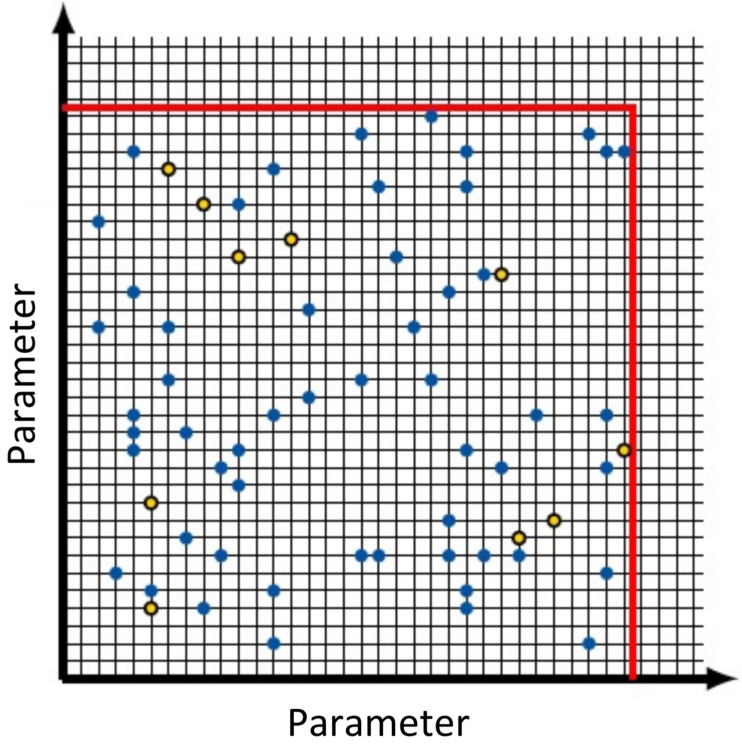


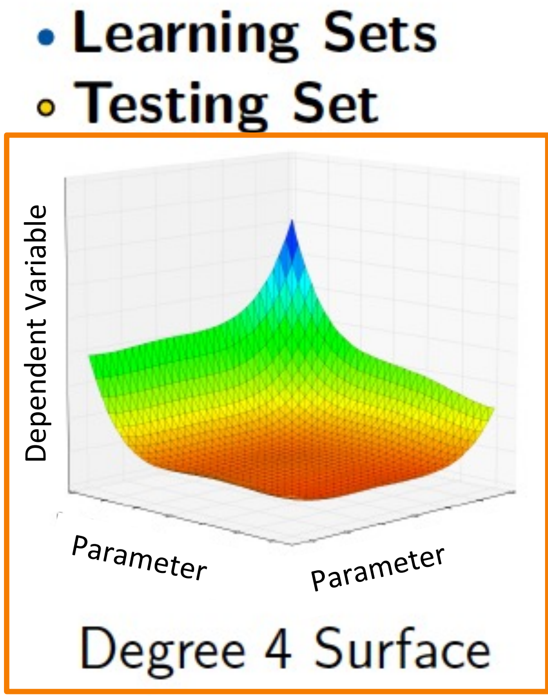
Degree 4 Surface

Step 3: Selecting the best surface using k-fold cross validation

Partition 1

- Partition samples into k sets of equal size
- k -1 sets are used for learning; one is use for testing
- **Build best fit polynomial surface from learning set**
- Use polynomial surface to predict unknown variables in testing set
- Use points in the testing set to compute SSE



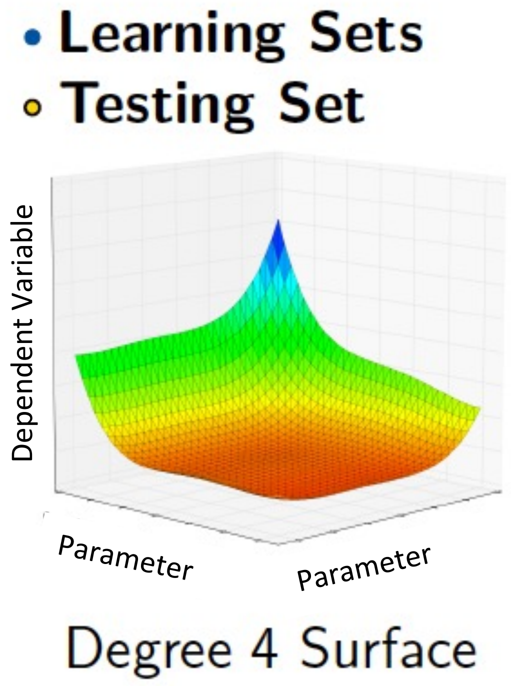


Step 3: Selecting the best surface using k-fold cross validation

Partition 1

- Partition samples into k sets of equal size
- k -1 sets are used for learning; one is use for testing
- Build best fit polynomial surface from learning set
- **Use polynomial surface to predict unknown variables in testing set**
- **Use points in the testing set to compute SSE**

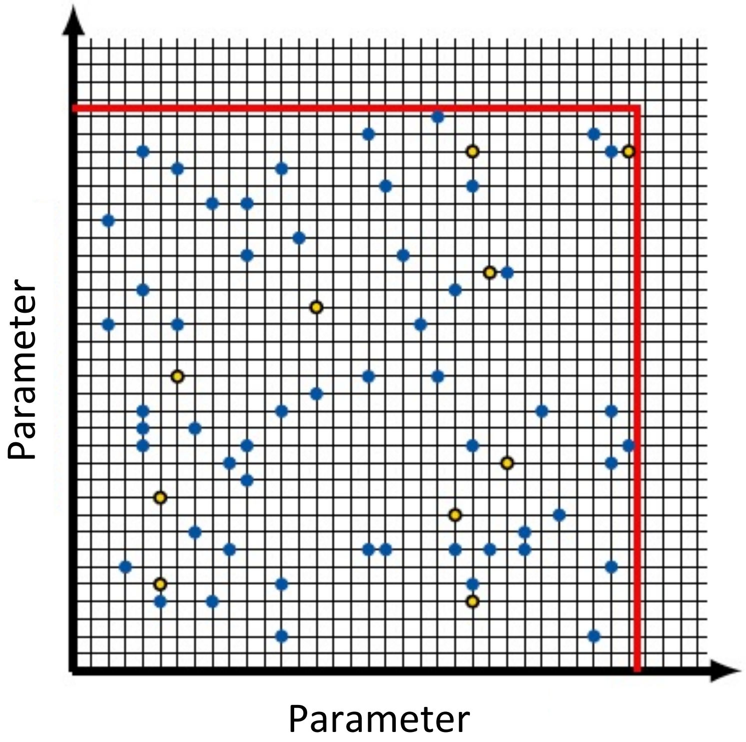
Compute SSE:
Observed variables
in testing points
vs
predicted variable
(using surface)
in testing pints

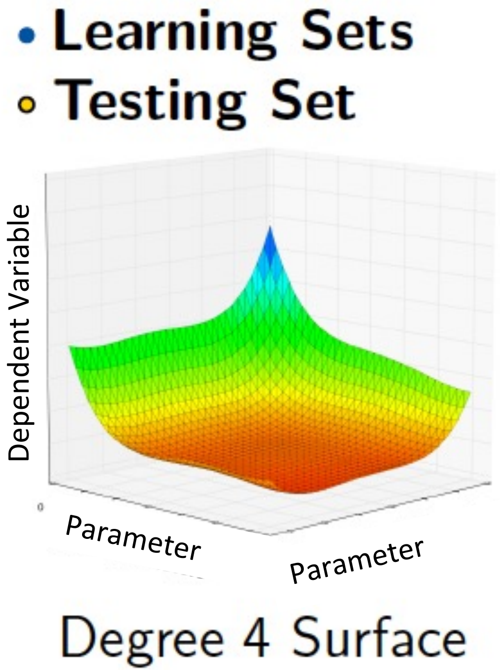


Step 3: Selecting the best surface using k-fold cross validation

Partition 2

- Partition samples into k sets of equal size
- k -1 sets are used for learning; one is use for testing
- Build best fit polynomial surface from learning set
- Use polynomial surface to predict unknown variables in testing set
- Use points in the testing set to compute SSE

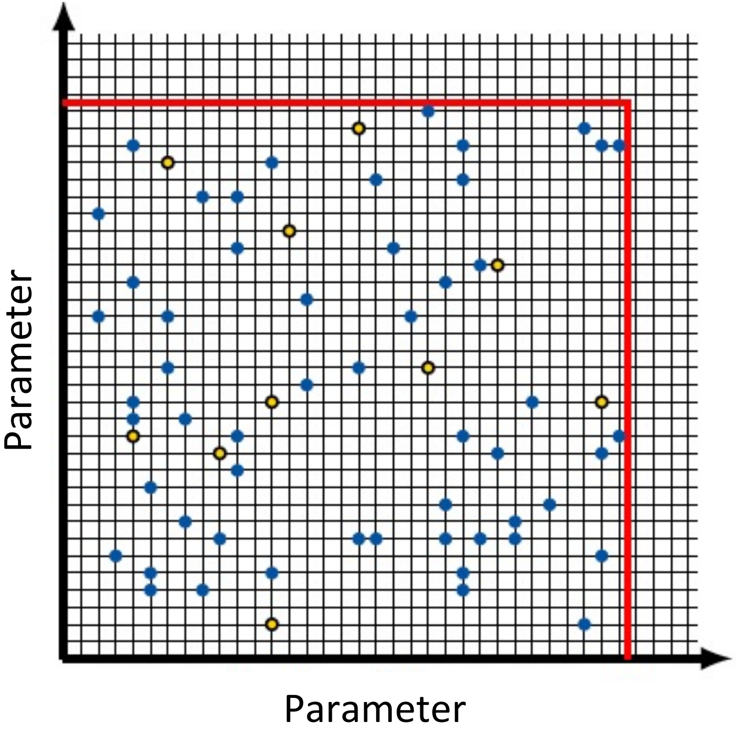


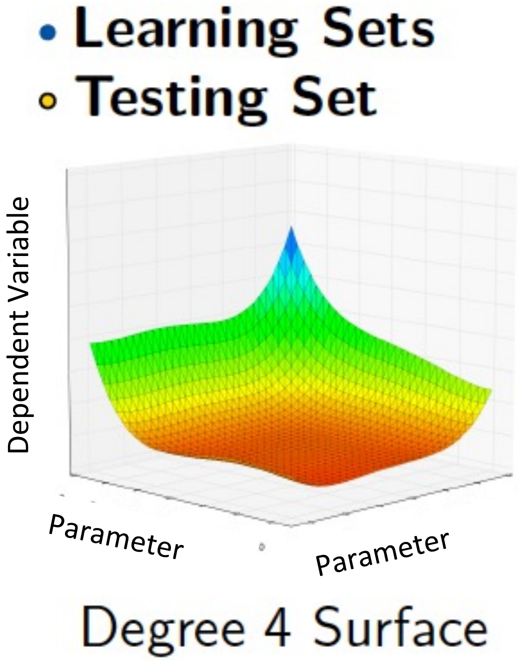


Step 3: Selecting the best surface using k-fold cross validation

Partition 3

- Partition samples into k sets of equal size
- k -1 sets are used for learning; one is use for testing
- Build best fit polynomial surface from learning set
- Use polynomial surface to predict unknown variables in testing set
- Use points in the testing set to compute SSE

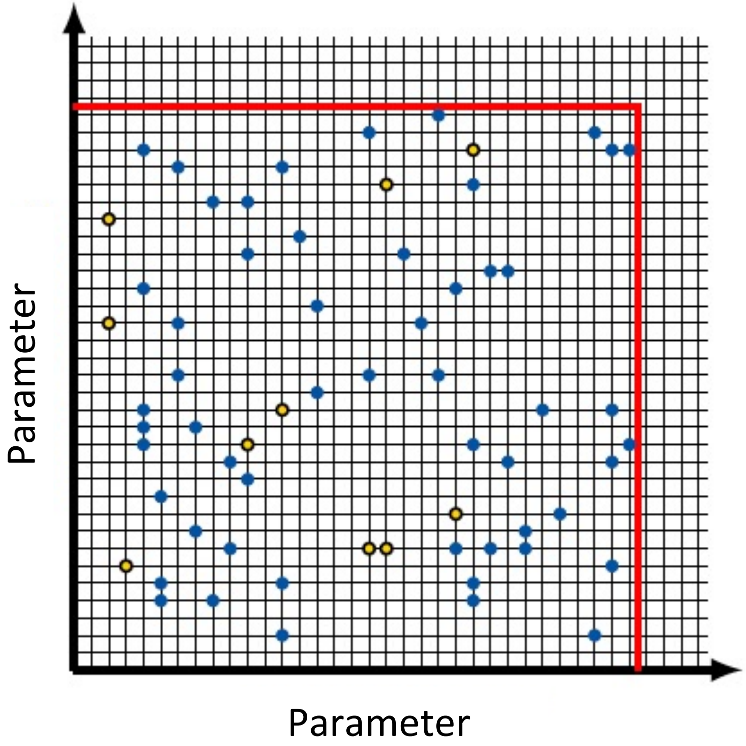


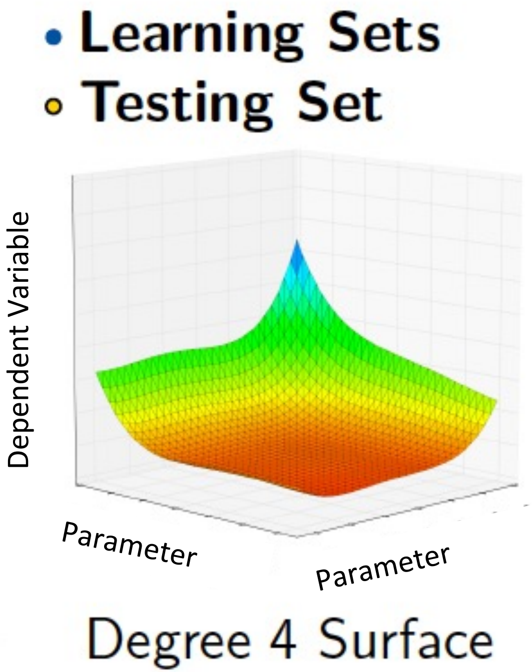


Step 3: Selecting the best surface using k-fold cross validation

Partition 4

- Partition samples into k sets of equal size
- k -1 sets are used for learning; one is use for testing
- Build best fit polynomial surface from learning set
- Use polynomial surface to predict unknown variables in testing set
- Use points in the testing set to compute SSE

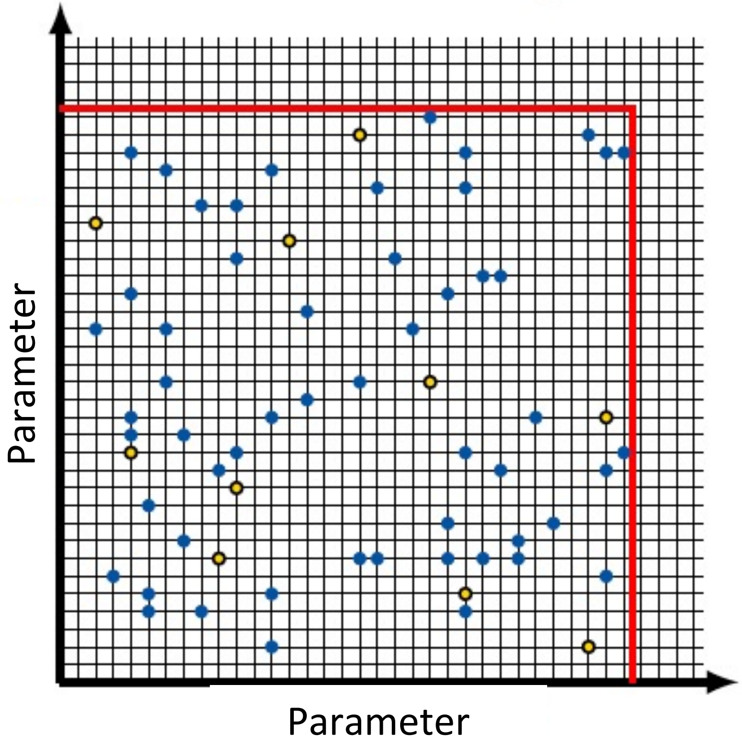


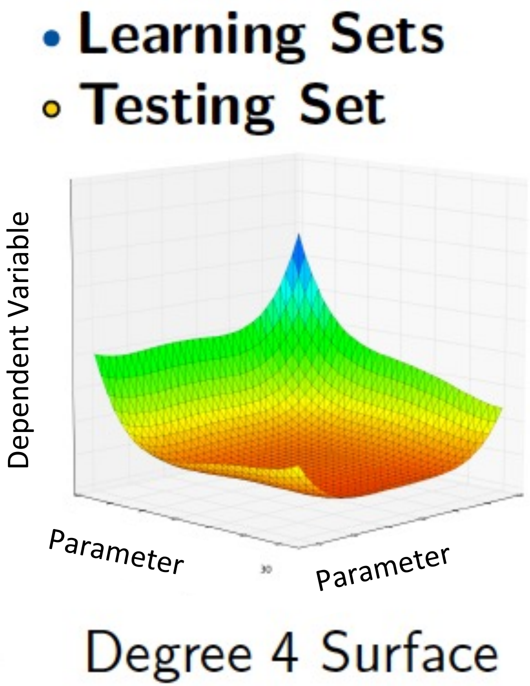


Step 3: Selecting the best surface using k-fold cross validation

Partition 5

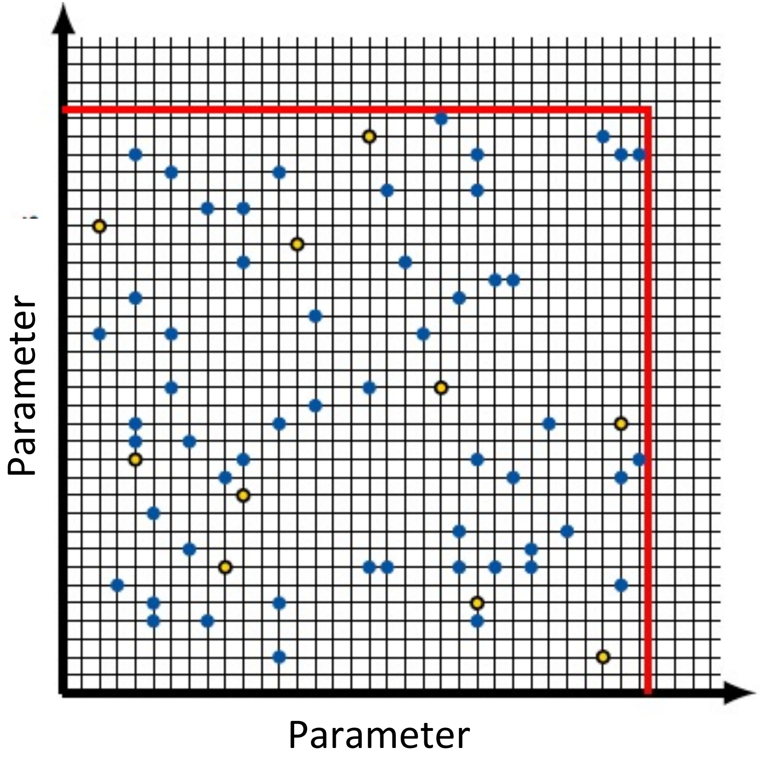
- Partition samples into k sets of equal size
- k -1 sets are used for learning; one is use for testing
- Build best fit polynomial surface from learning set
- Use polynomial surface to predict unknown variables in testing set
- Use points in the testing set to compute SSE

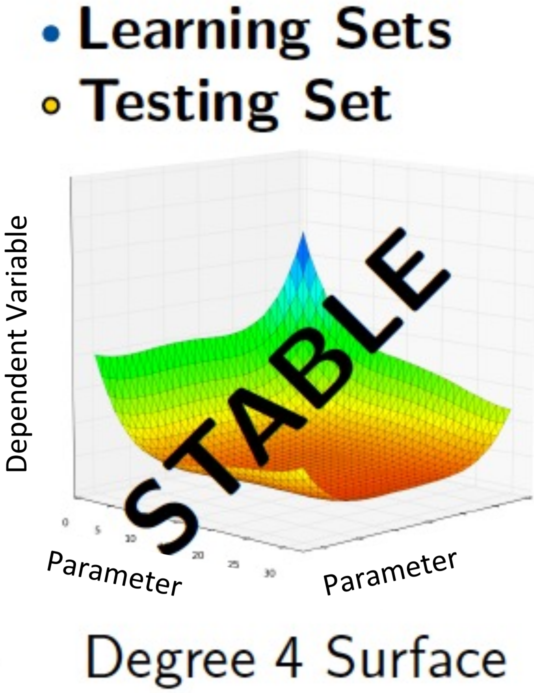




Step 3: Selecting the best surface using k-fold cross validation

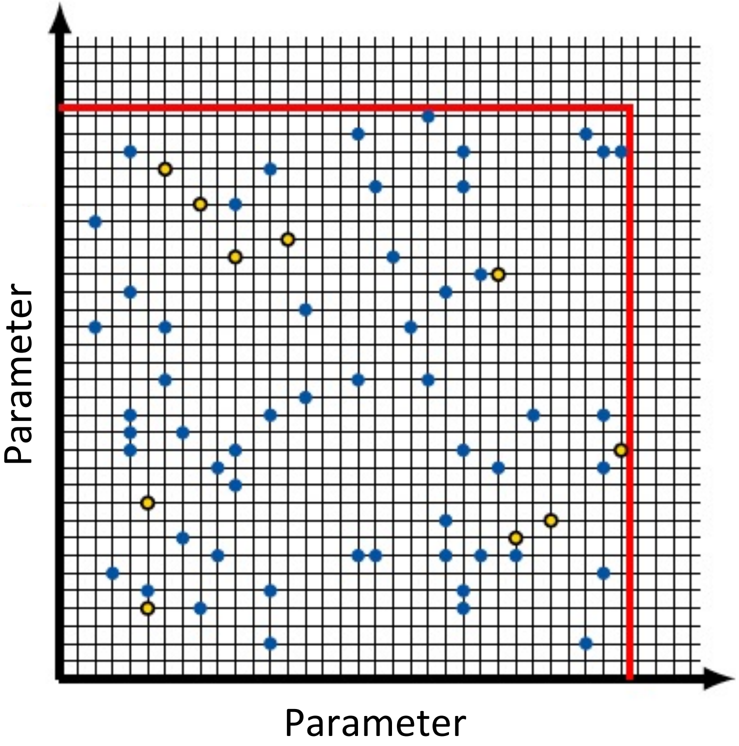
- Partition samples into k sets of equal size
- k -1 sets are used for learning; one is use for testing
- Build best fit polynomial surface from learning set
- Use polynomial surface to predict unknown variables in testing set
- Use points in the testing set to compute SSE

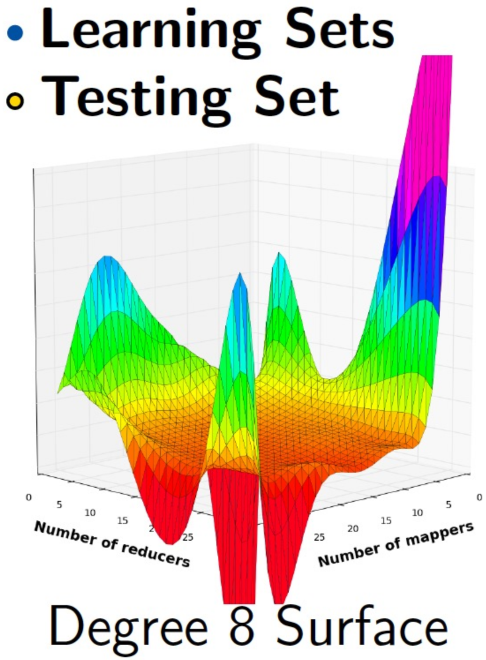




Step 3: Selecting the best surface using k-fold cross validation

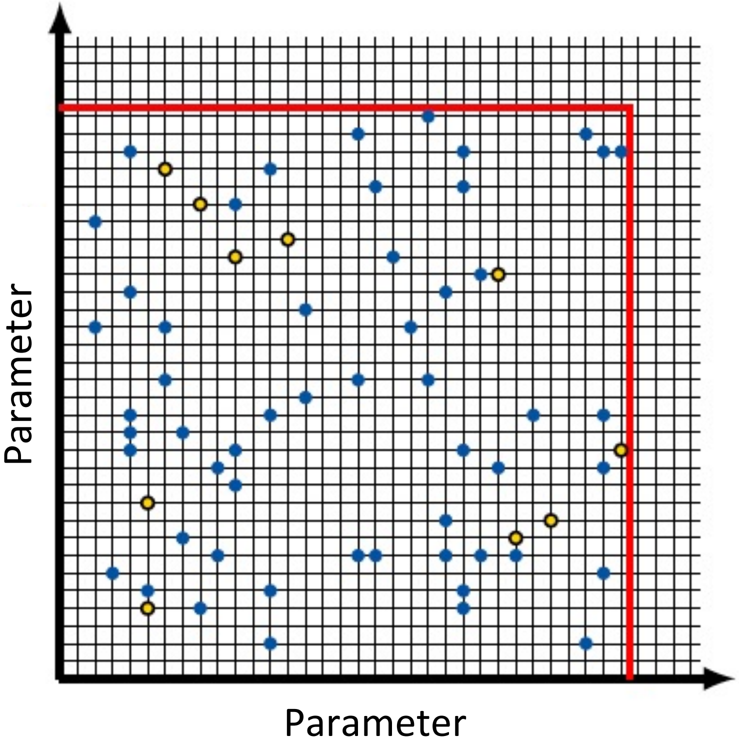
- Partition samples into k sets of equal size
- k -1 sets are used for learning; one is use for testing
- Build best fit polynomial surface from learning set
- Use polynomial surface to predict unknown variables in testing set
- Use points in the testing set to compute SSE

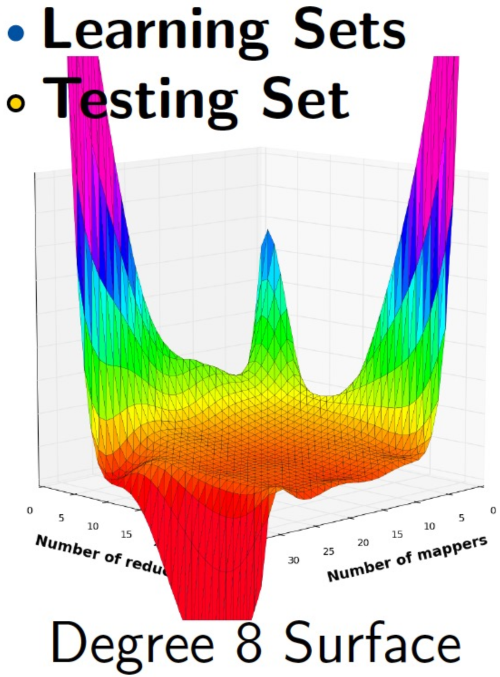




Step 3: Selecting the best surface using k-fold cross validation

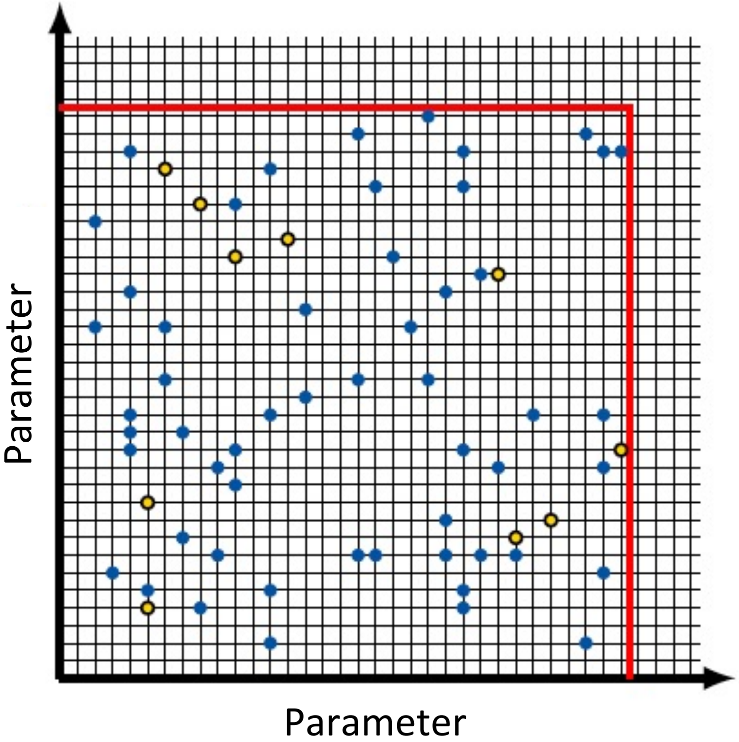
- Partition samples into k sets of equal size
- k -1 sets are used for learning; one is use for testing
- Build best fit polynomial surface from learning set
- Use polynomial surface to predict unknown variables in testing set
- Use points in the testing set to compute SSE



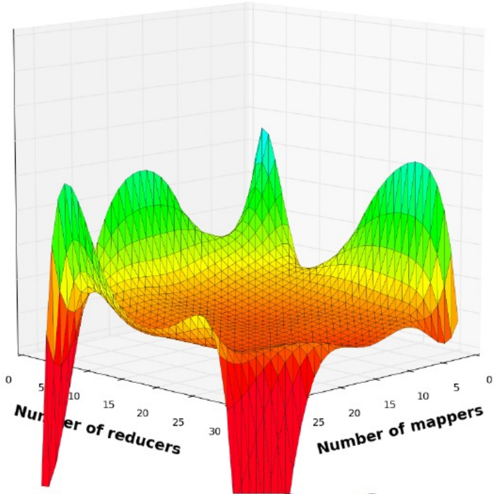


Step 3: Selecting the best surface using k-fold cross validation

- Partition samples into k sets of equal size
- k -1 sets are used for learning; one is use for testing
- Build best fit polynomial surface from learning set
- Use polynomial surface to predict unknown variables in testing set
- Use points in the testing set to compute SSE



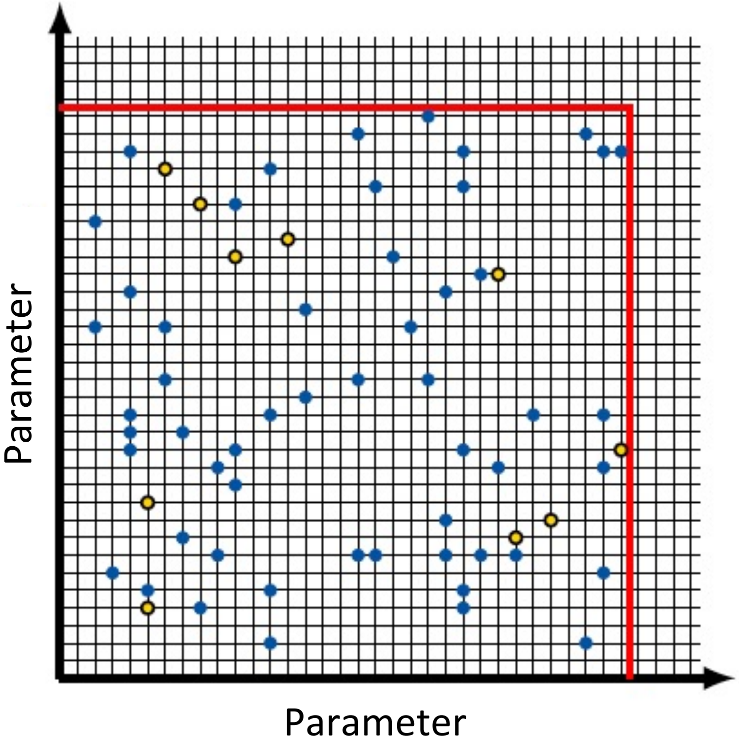
- Learning Sets
- Testing Set



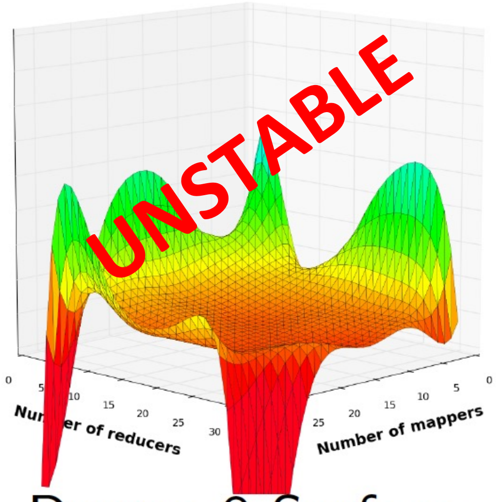
Degree 8 Surface

Step 3: Selecting the best surface using k-fold cross validation

- Partition samples into k sets of equal size
- k -1 sets are used for learning; one is use for testing
- Build best fit polynomial surface from learning set
- Use polynomial surface to predict unknown variables in testing set
- Use points in the testing set to compute SSE

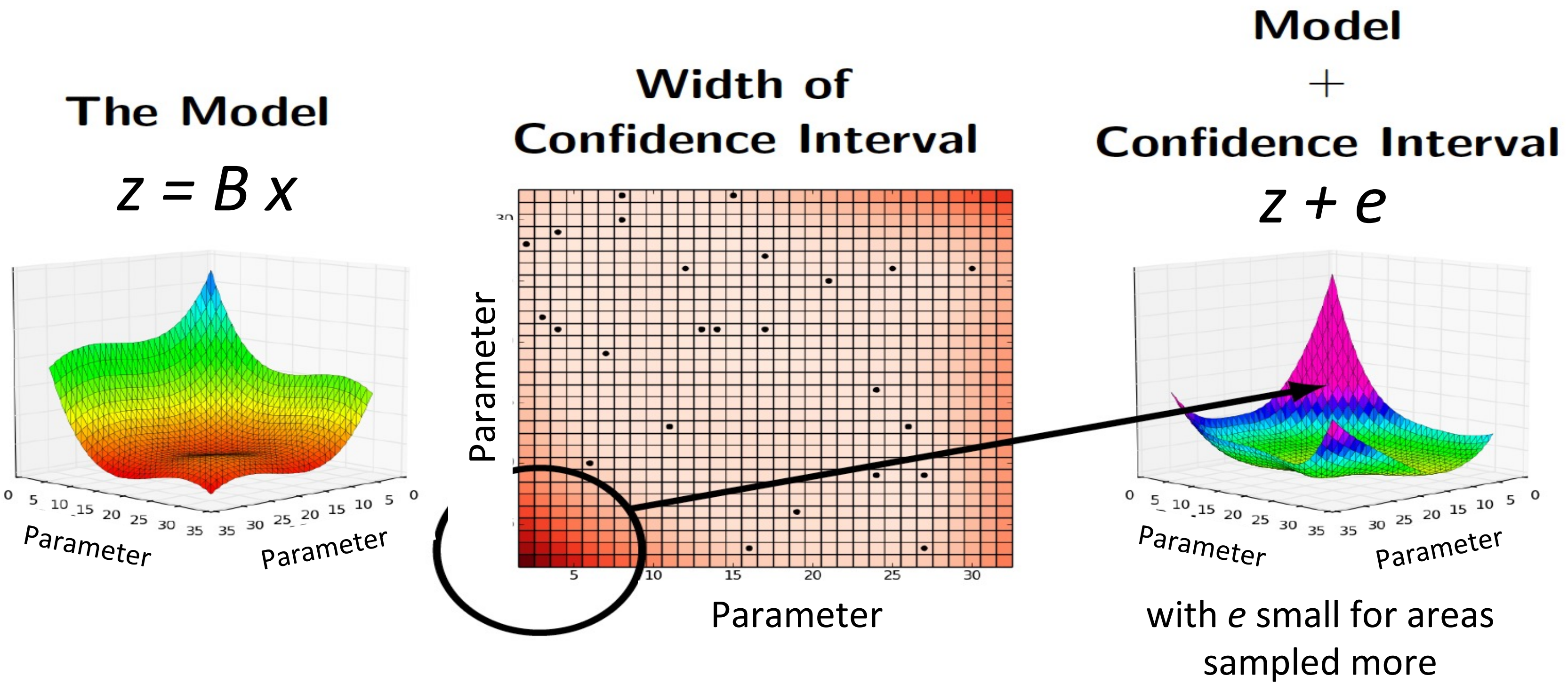


- Learning Sets
- Testing Set



Degree 8 Surface

Step 4: Applying a confidence interval



KNN and SBM

KNN:

→ Use **local data**

→ Compute k and distance kernel using cross validation automatically

→ Compute weighted means with the kernel (***many values***)

0 0 0 0

0 0 0 0 0

Surrogate based model (SBM):

→ Use **all sampled data**

→ Use regression to generate one single polynomial model (***single polynomial model***)

4 4 4 4

4 4 4 4 4

D. Rorabaugh, M. Guevara, R. Llamas, J. Kitson, R. Vargas, and **M. Taufer**. SOMOSPIE: A Modular SOil MOisture SPatial Inference Engine based on Data Driven Decisions. eScinece 2019.

Hybrid Piecewise Polynomial Modeling

HYbrid Piecewise POlynomial modeling (HYPPO)

k Nearest Neighbors

- Use **local** data
- Compute the average
  (*many simple local models*)

Surrogate-Based Modeling

- Use **all** sampled data
- Construct **one** **polynomial**
  (*single complex global model*)

Hybrid modeling ⍰ HYPPO

- Use **local** data
- Construct many **polynomials**
  (*many complex local models*)

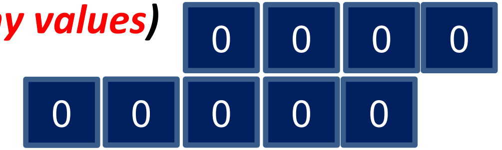
KNN, SBM, and HYPPO

KNN:

→ Use **local data**

→ Compute k and distance kernel using cross validation automatically

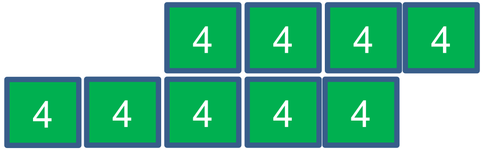
→ Compute weighted means with the kernel (***many values***)



Surrogate based model (SBM):

→ Use **all sampled data**

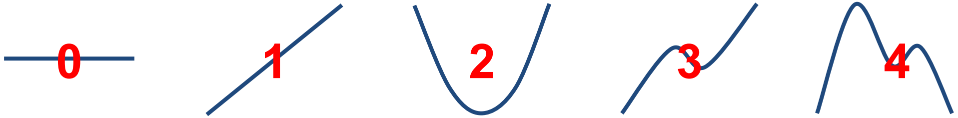
→ Use regression to generate one single polynomial model (***single polynomial model***)



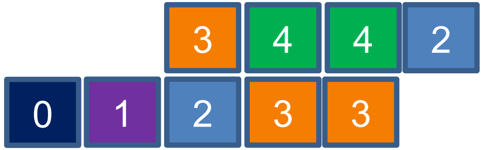
HYPPO (Hybrid Piecewise Polynomial Modeling):

→ Use **local data**

→ Determine local polynomial degree using cross validation



→ Use regression to generate local polynomial model (***many polynomial models***)

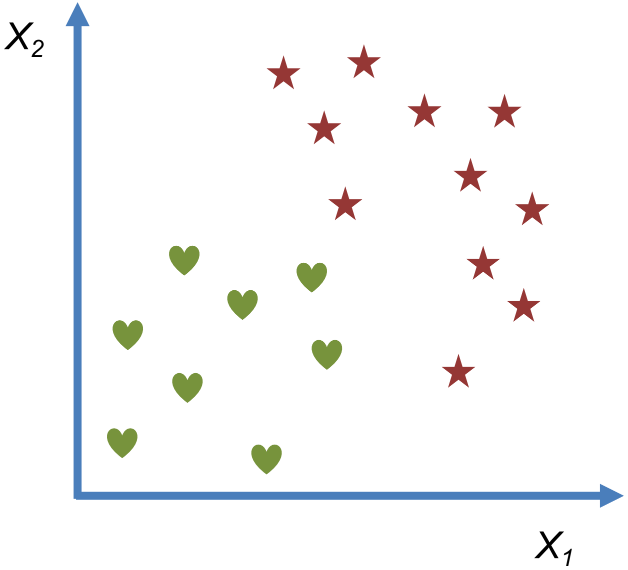


D. Rorabaugh, M. Guevara, R. Llamas, J. Kitson, R. Vargas, and **M. Taufer**. SOMOSPIE: A Modular SOil MOisture SPatial Inference Engine based on Data Driven Decisions. eScinece 2019.

BIG ORANGE BIG IDEAS

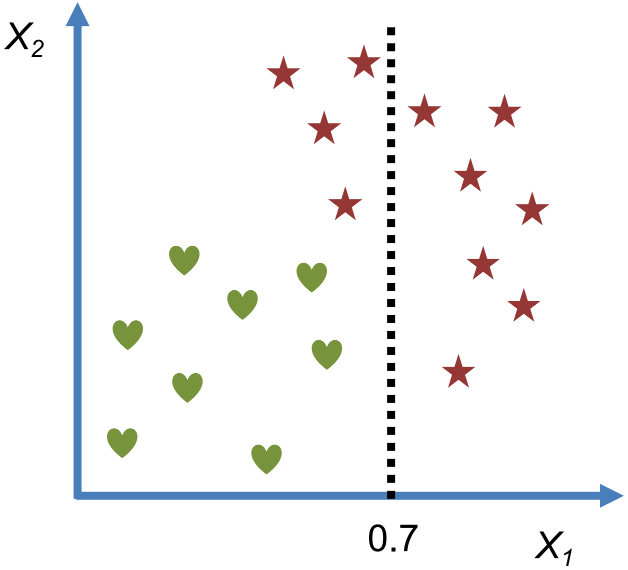
Random Forest

Divide and Conquer



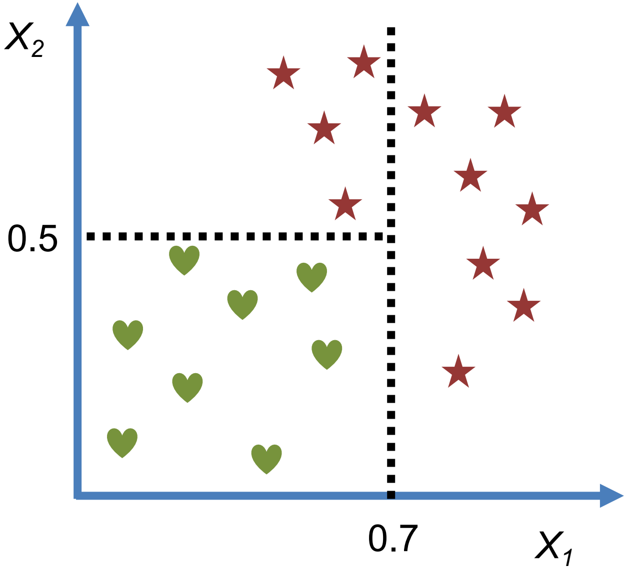
- Information gain is used to decide how to divide the dataset for classification
- The data is recursively divided until subdivision are all one class

Divide and Conquer



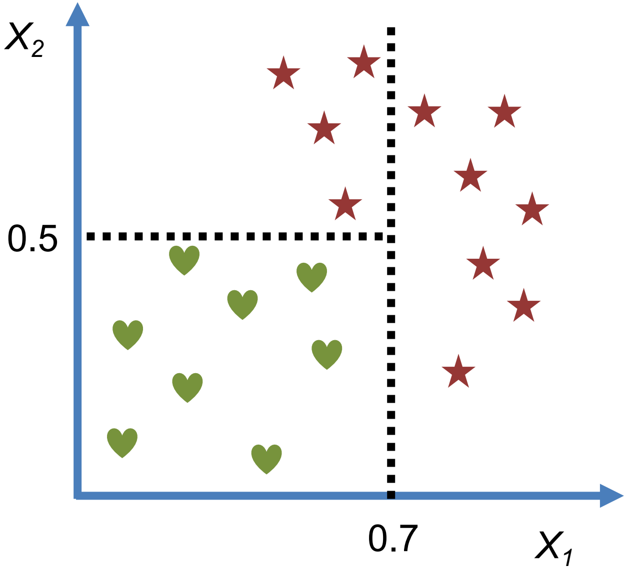
- Information gain is used to decide how to divide the dataset for classification
- The data is recursively divided until subdivision are all one class

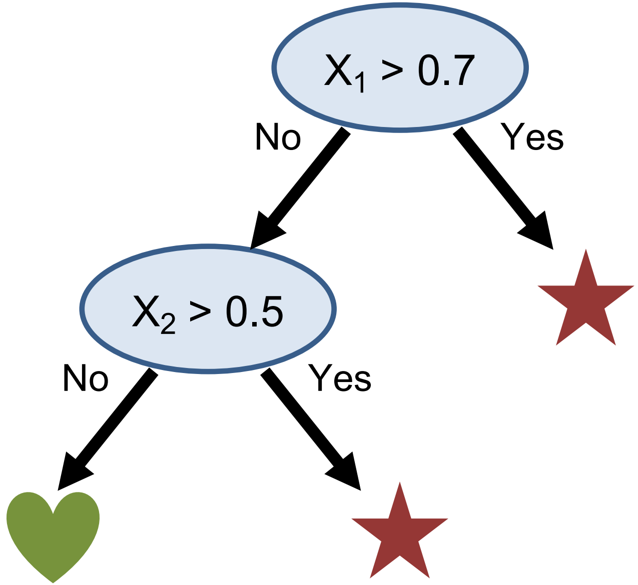
Divide and Conquer



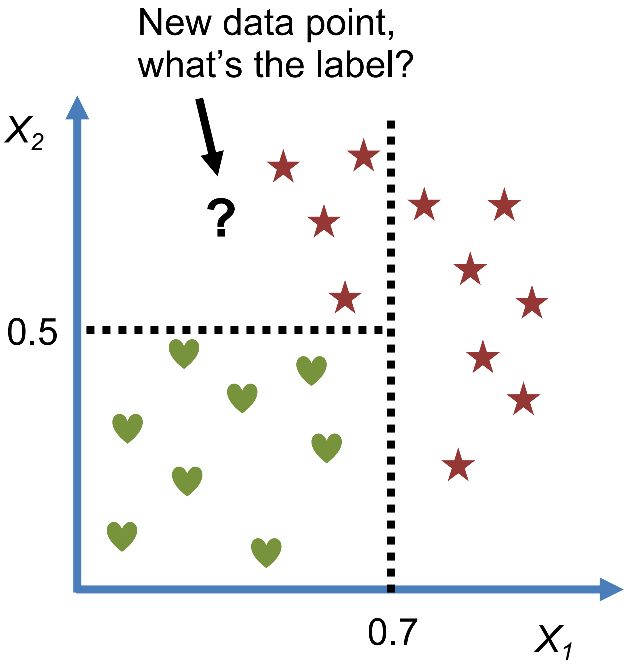
- Information gain is used to decide how to divide the dataset for classification
- The data is recursively divided until subdivision are all one class

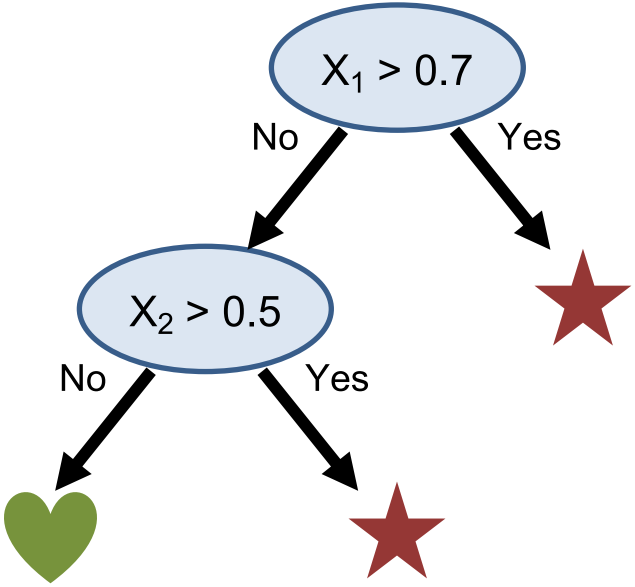
Divide and Conquer



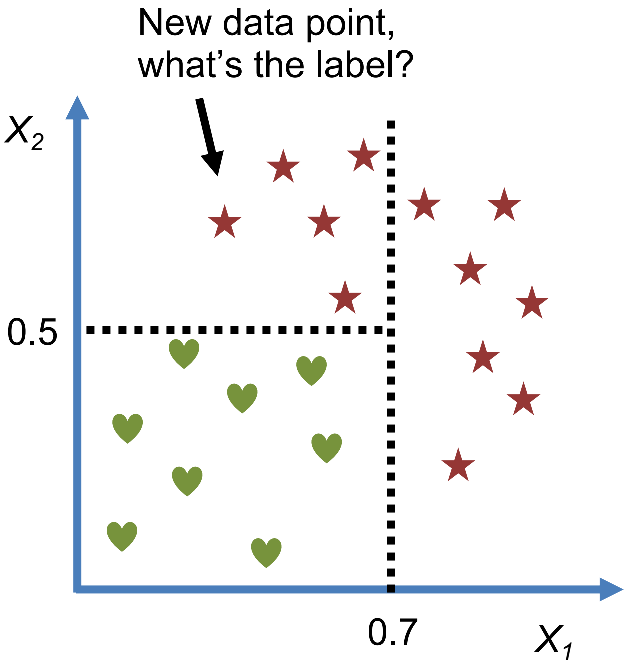


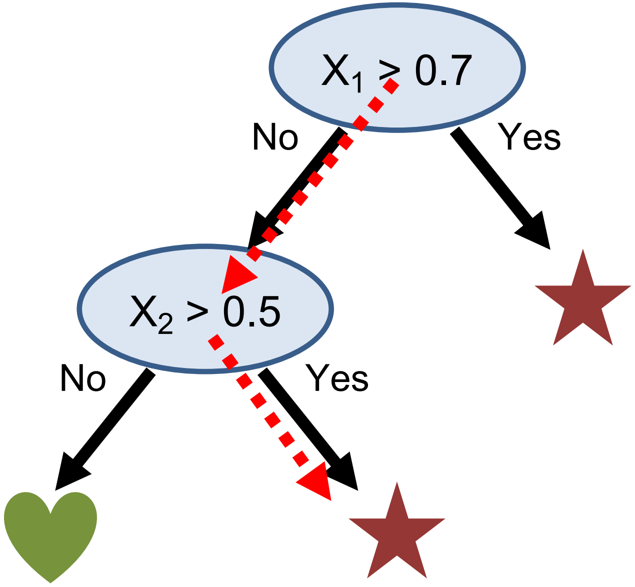
Divide and Conquer





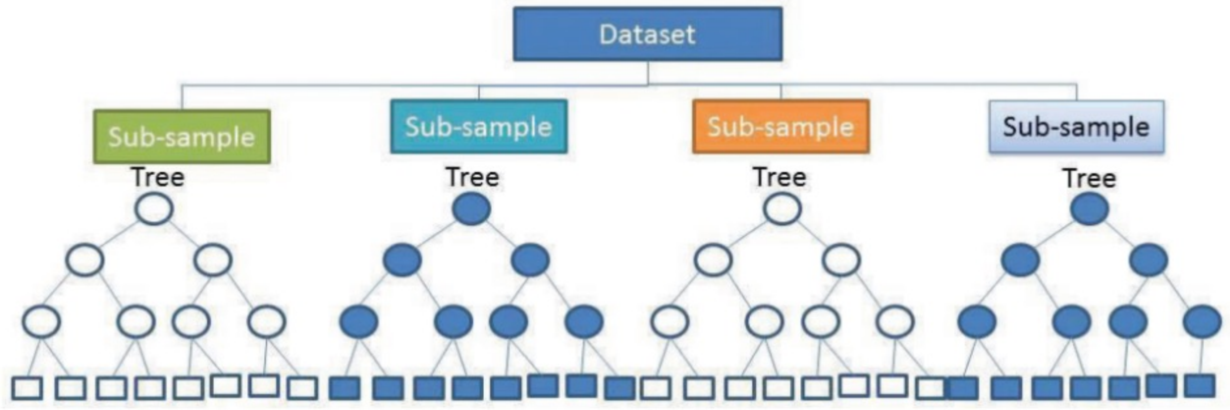
Divide and Conquer





Random Forests: Ensembles of Trees

- The prediction of many trees is better than the prediction of any one tree



- Each tree is built with a random sample of data rows and data features
- Each tree makes a prediction and the most popular prediction is returned

Reading

Reading

- T. Johnston, M. Alsulmi, P. Cicotti and M. Taufer. Performance Tuning of MapReduce Jobs Using Surrogate-Based Modeling. In *Proceedings of the International Conference on Computational Science (ICCS)*, pp. 49 – 59. Reykjavik, Iceland. June 1 – 3, 2015.
- T. Johnston, C. Zannin, and M. Taufer. HYPPO: A Hybrid, Piecewise Polynomial Modeling Technique for Non-Smooth Surfaces. In *Proceedings of the 28th IEEEE International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, pp. 1 – 8. Los Angeles, CA, USA. October 26 – 28, 2016.
- Breiman, L. Random Forests. *Machine Learning* **45,** 5–32 (2001). https://doi.org/10.1023/A:1010933404324

Live Chat

Live Chat

- GTC 2015 Keynote with Jeff Dean, Google

  https://video.ibm.com/recorded/60071572