Metamodeling

ME555 Lecture

(Max) Yi Ren

Department of Mechanical Engineering, University of Michigan

February 12, 2014

Motivation

Metamodeling is commonly used when the objective (and constraints) of an optimization problem can only be evaluated through experiments or simulations.

When to use metamodeling:

- Expensive computational cost of function and gradient evaluation
- No model available, only data
- Smooth out numerical noise

Also called surrogate modeling.

Basic steps:

1. Sample the design space using a sample set $\mathbf{X}$;
2. Get objective (and constraints) values from simulation or experiment, denoted as $\mathbf{y}$;
3. Train metamodels using the data $\mathbf{X}$, $\mathbf{y}$.

Ordinary least square regression

Let the independent variables be $\mathbf{X}$ (n by p), with n observations with p dimensions, and the ith dependent variable be y_i. A linear model assumes:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{1}$$

where $\boldsymbol{\varepsilon}$ is a random error from a certain distribution. The goal of OLS is to estimate the model parameters $\boldsymbol{\beta}$ so that the estimations $\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta}$ are close to the observed $\mathbf{y}$. When $\boldsymbol{\varepsilon}$ is i.i.d. and normal, this difference can be measured by a Euclidean distance:

$$\boldsymbol{\beta}^* = arg\min_{\boldsymbol{\beta}} \ ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2, \tag{2}$$

which can be solved analytically as:

$$\boldsymbol{\beta}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{3}$$

What can go wrong with this solution $\boldsymbol{\beta}^*$?

Performance measure

R^2 value:

$$R^2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (y_i - \bar{y})^2},$$

where $\mathbf{y}$ are from the testing data, $\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ are estimates of $\mathbf{y}$ and $\bar{y}$ is the mean of $\mathbf{y}$. We can use R^2 to evaluate the testing performance of the model. Higher R^2 indicates better performance.

Tip: If your regression model has low R^2 value, first try normalizing $\mathbf{X}$:

$$x_{ij} = \frac{x_{ij} - \bar{x}_j}{\delta_j},$$

where $\bar{x}_j$ and δ_j are the mean and deviation on dimension j.

Akaike information criterion

Occam's razor: Among competing hypotheses, the hypothesis with the fewest assumptions should be selected.

AIC: A measure of goodness of fit and model complexity, for a given set of data. Provides a means for model selection.

$$AIC = 2p - 2\ln(L), \tag{4}$$

where p is the number of parameters and L is the maximum likelihood.

In OLS

$$AIC = \ln\left(\sum_i (\hat{y}_i - y_i)^2/n\right) + \frac{2p}{n}, \tag{5}$$

where n is the sample size.

Akaike information criterion

AICc: AIC with a correction:

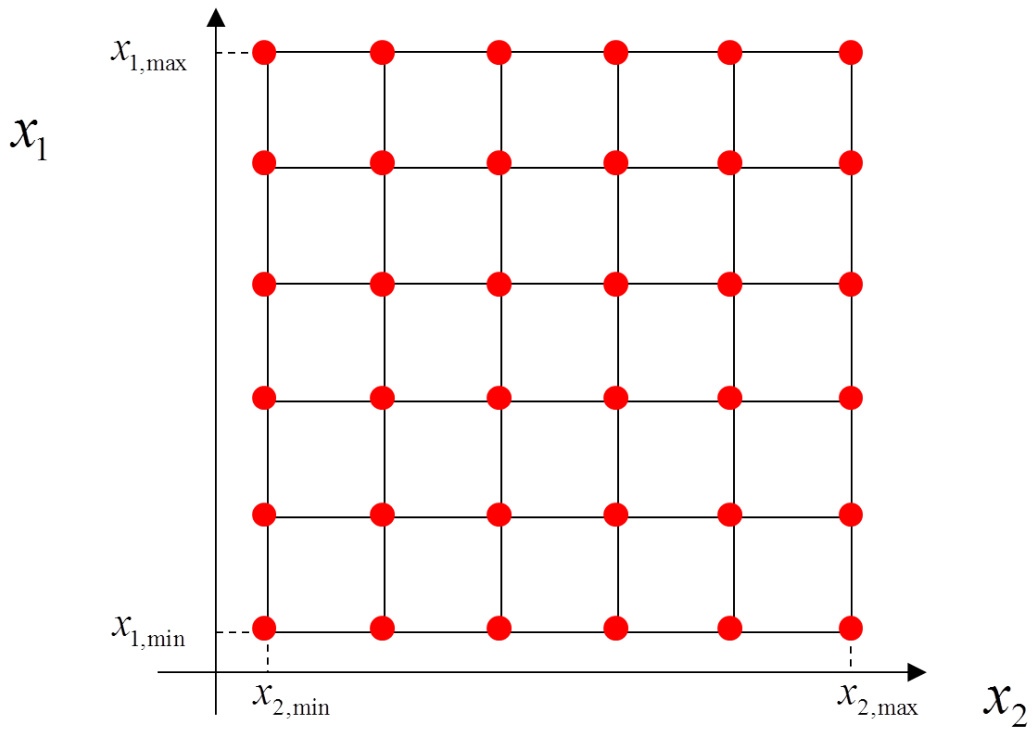
$$AICc = AIC + \frac{2p(p+1)}{n-p-1}. \tag{6}$$

Use AICc instead of AIC when $n/p < 40$.

Sampling

Design of experiments (optimal experiment design): efficiently sample the design space to create a statistical model with high prediction performance.

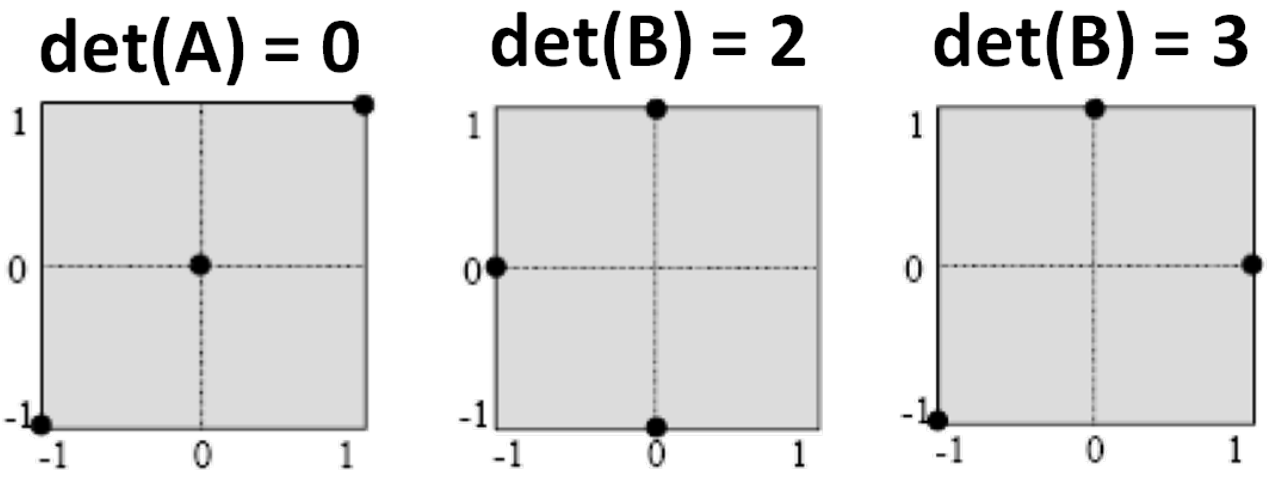
A naive way is to use *full factorial experiment*



Full factorial sampling costs l^p samples and will be intractable when l (the number of levels) or p (the number of variables) are large.

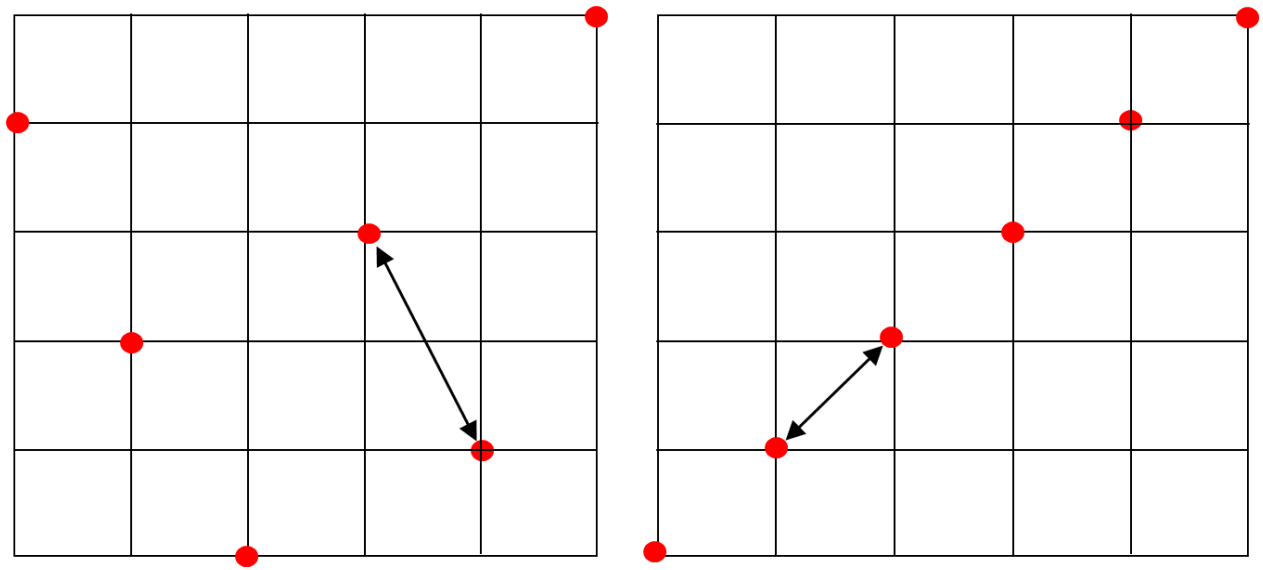
D-optimal design

For a fixed amount of samples, one may use a sample set with large determinant of $\mathbf{X}^T\mathbf{X}$.



Latin hypercube

Latin hypercube sampling (LHS) uses l samples regardless of the number of variables (p), and is therefore widely adopted.

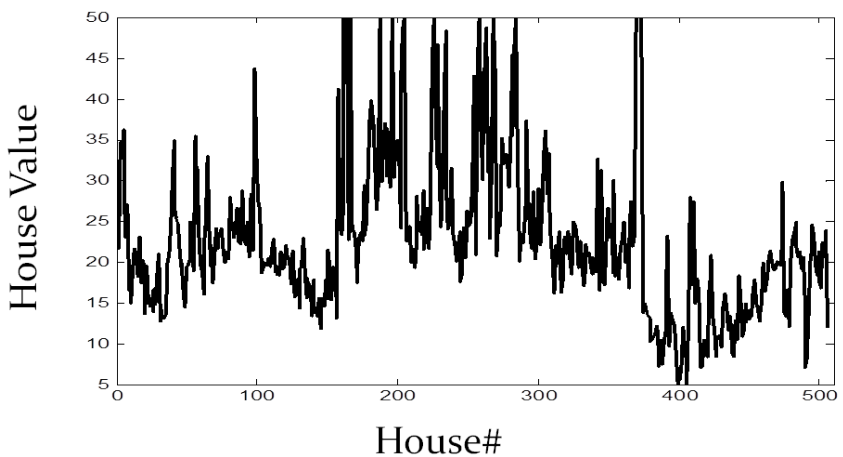


In LHS, there does not exist samples that share the same value on any variable. To implement LHS, one should also include dispersion criteria, e.g., maximizing the minimum distance between sample points, or minimizing the correlation.

Regression methods

When the function to model cannot be linearly approximated by design variables, or we don't know what features (e.g., polynomial terms) to use for modeling the observation, OLS may not work well.

An example from Matlab House.data: $\mathbf{X}$ (506×13): 506 houses with 13 parameters, $\mathbf{y}$ (506×1): house values.



	OLS	feed-forward NN	SVR RBF
Testing R^2	0.66	0.77	0.83

Table: Cross-validated testing R^2 on House.data

Feed-forward neural networks (NNFF)

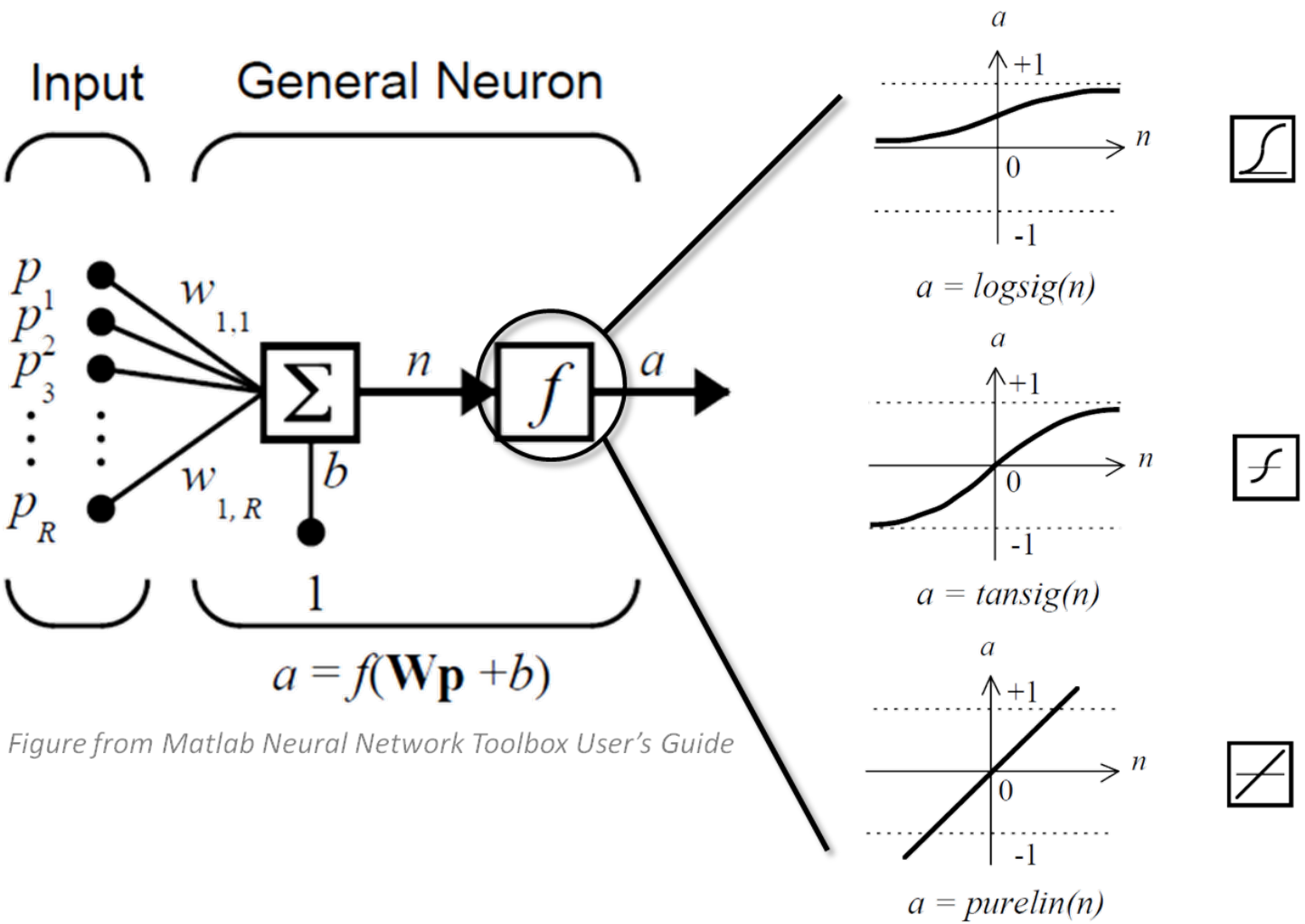


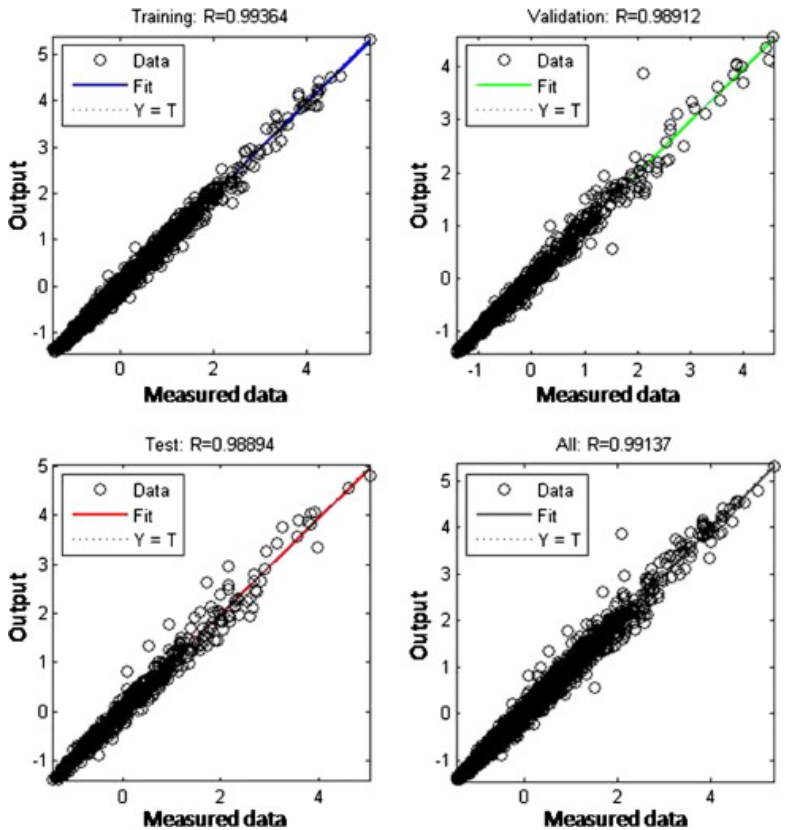
Figure from Matlab Neural Network Toolbox User's Guide

A simplest feed-forward neural net. One may add arbitrary number of layers and neurons to the model.

Feed-forward neural networks (NNFF)

We find optimal network parameters to minimize the mean-squared error (MSE) of the testing data. What algorithms can we use?

Matlab uses a portion of the training data for validation. The training (optimization) will terminate when gradient is close to zero or MSE of the validation set does not decrease for a few iterations.



Radial-basis neural networks (NNRB)

When sample $\mathbf{y}$ are deterministic, the following NNRB model can be used for interpolation purpose:

$$y(\mathbf{x}) = \sum w_i \exp(-\gamma ||\mathbf{x} - \mathbf{x}_i||^2)$$

, where $\mathbf{w}$ are network weights.

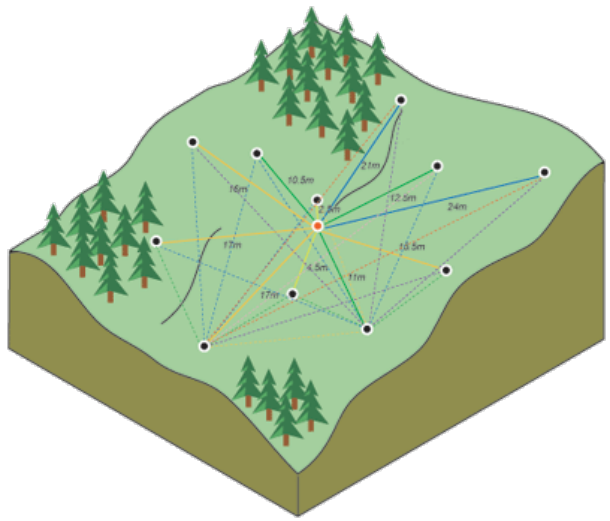
Let $r_j(\mathbf{x}) = \exp(-\gamma ||\mathbf{x} - \mathbf{x}_j||^2)$, with n samples we have $\mathbf{Rw} = \mathbf{y}$, where the matrix $\mathbf{R}$ is

$$\begin{bmatrix} r_1(\mathbf{x}_1) & r_2(\mathbf{x}_1) & \cdots & r_n(\mathbf{x}_1) \\ r_1(\mathbf{x}_2) & r_2(\mathbf{x}_2) & \cdots & r_n(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ r_1(\mathbf{x}_n) & r_2(\mathbf{x}_n) & \cdots & r_n(\mathbf{x}_n) \end{bmatrix}$$

It can be proved that $\mathbf{R}$ is non-singular if samples $\mathbf{X}$ are distinct, and thus the weights can be solved as $\mathbf{w} = \mathbf{R}^{-1}\mathbf{y}$.

Kriging

Kriging: A geostatistical techniques to interpolate the elevation of the landscape as a function of the geographic location at an unobserved location from observations of its value at nearby locations.



The Kriging model has

$$\hat{Y}(\mathbf{x}) = \sum_{i=1}^{n} w_i(\mathbf{x}) Y(\mathbf{x}_i),$$

where $Y(\mathbf{x})$ is a random field on $\mathbf{x}$, $w_i(\mathbf{x})$ is the weight measuring the similarity between $\mathbf{x}$ and $\mathbf{x}_i$.

Kriging

The simple Kriging model assumes $E[Y(\mathbf{x})] = 0$, which results in the model

$$\hat{y}(\mathbf{x}) = \mathbf{r}(\mathbf{x})^T \mathbf{R}^{-1} \mathbf{y},$$

where the vector $\mathbf{r}(\mathbf{x})$ measures the similarities between $\mathbf{x}$ and all samples $\mathbf{x}_i$, and the matrix $\mathbf{R}$ measures the similarities among all samples. When we use the radial-basis (Gaussian) function for measuring similarity, simple Kriging results in the same model as from NNRB.

When assuming $E[Y(\mathbf{x})] = \text{const}$, we will have the Kriging model:

$$\hat{y}(\mathbf{x}) = \hat{b} + \mathbf{r}(\mathbf{x})^T \mathbf{R}^{-1} (\mathbf{y} - \hat{b}\mathbf{1}),$$

where

$$\hat{b} = \frac{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{y}}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}}.$$

The prediction $\hat{y}$ at any sampled $\mathbf{x}$ matches the sampled value y. Therefore Kriging is widely used for metamodeling from computer simulations (with deterministic outputs).

Ridge regression (RR)

Recall the solution of OLS

$$\boldsymbol{\beta}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

When $\mathbf{X}^T\mathbf{X}$ is ill-conditioned, we can try

$$\boldsymbol{\beta}^* = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y},$$

where λ is an unknown parameter.

This solution corresponds to minimizing

$$||\mathbf{X}\boldsymbol{\beta} - \mathbf{y}||^2 + \lambda||\boldsymbol{\beta}||^2.$$

This objective tries to minimize MSE within a sphere of possible $\boldsymbol{\beta}$.

λ represents your believe of the observations, i.e., the larger λ is, the less believe you have. One can use cross-validation on the training data to find the optimal value of λ.

Support vector regression (SVR)

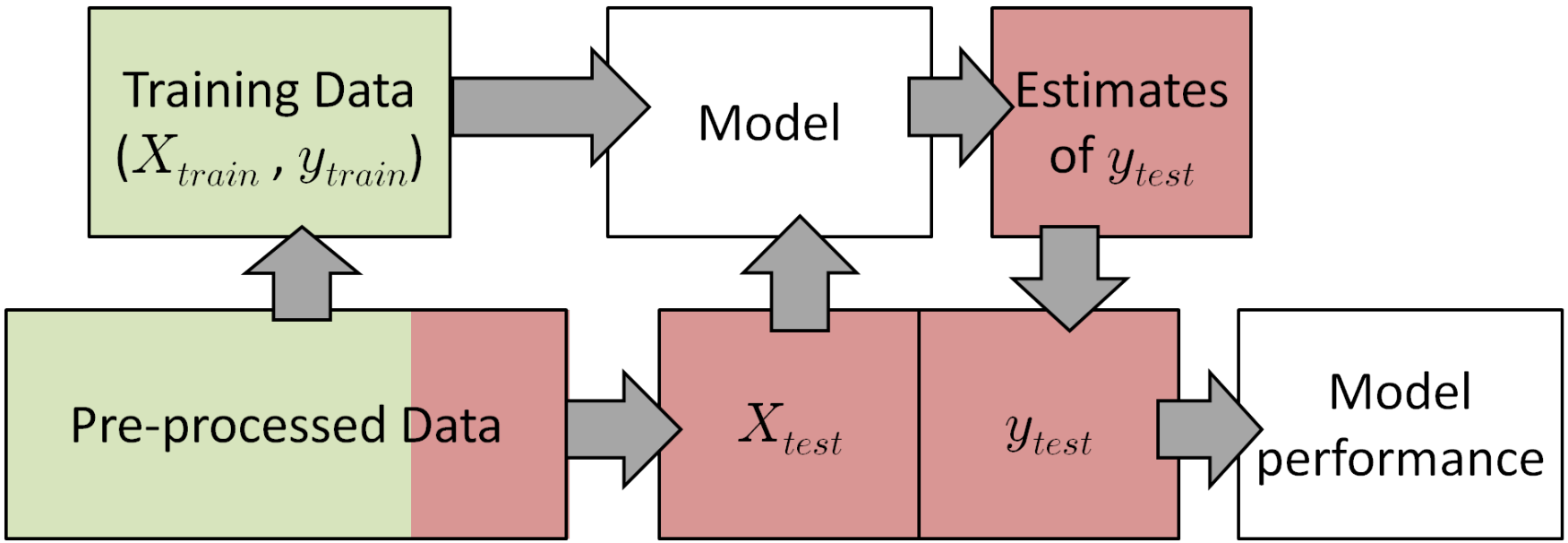
SVR is the regression version of the original support vector machine (SVM) for classification. The idea is to balance the training error (MSE) and model complexity to prevent over-fitting:

$$\begin{aligned} \min_{\boldsymbol{\beta},b,\boldsymbol{\xi},\boldsymbol{\xi}^*} \quad & ||\beta||^2 + C_1 \sum \xi_i + C_2 \sum \xi_i^* \\ \text{subject to} \quad & \boldsymbol{\beta}^T \mathbf{x}_i + b - y_i \leq \varepsilon + \xi_i, \\ & y_i - \boldsymbol{\beta}^T \mathbf{x}_i - b \leq \varepsilon + \xi_i^*, \\ & \xi_i,\ \xi_i^* \geq 0, \forall i. \end{aligned}$$

Similar to Kriging, with a definition of similarity, SVR can train nonlinear models. It will have the same analytical solution to Kriging when training error is forced to zero.

Training and testing

How do we know whether the model is good or not?



There is no rule for splitting the data into training and testing data. In practice one would use $1/5$ to $1/3$ of data for testing.

One can also use cross-validation for testing: (1) Partition data into sets, (2) Use all but one sets to create a model, (3) Use the remaining set to test the model, (4) Iterate through all sets and report averaged testing performance.

When each set has a size of 1, it is called leave-one-out cross-validation.

Summary

- Sample using Latin hypercube, $\mathbf{X}$ needs to have similar scale on each dimension;
- Always try OLS first;
- Use AIC (or other information criteria) when applicable, otherwise report performance on testing data (or use cross-validation).
- OLS, Kriging, RR, SVR are easier to tune than NNFF but NNFF can be more powerful if well-tuned;
- Choose a model with the best performance. Retrain with all data when training and testing is used.