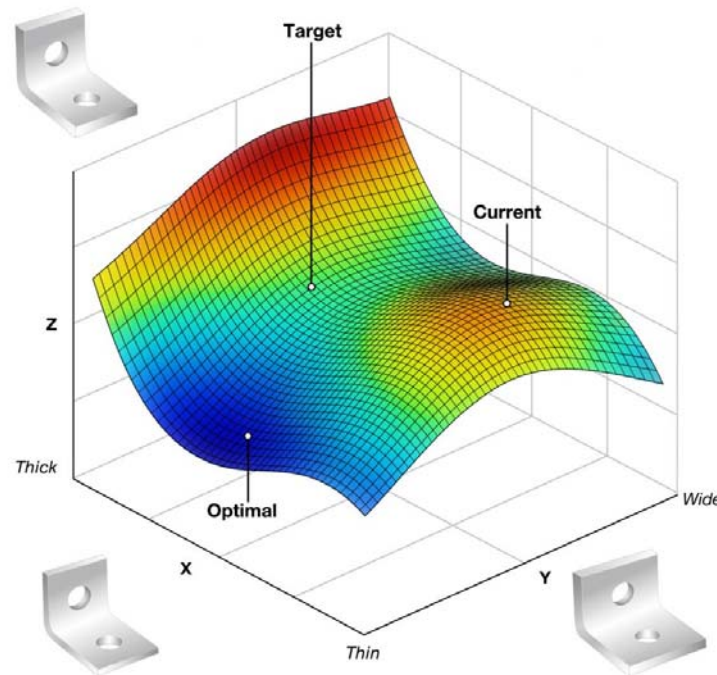


Introduction to Surrogate Modeling

Nam-Ho Kim



$$\underset{\text{truth}}{f(\mathbf{x})} \approx \underset{\text{surrogate}}{\hat{f}(\mathbf{x})}$$

Outline of surrogate modeling module

- Introduction to surrogate modeling
- Polynomial response surface
- Linear regression accuracy
- Sampling plans
- Neural network model
- Radial basis neural network
- Kriging surrogate

What is a surrogate model?

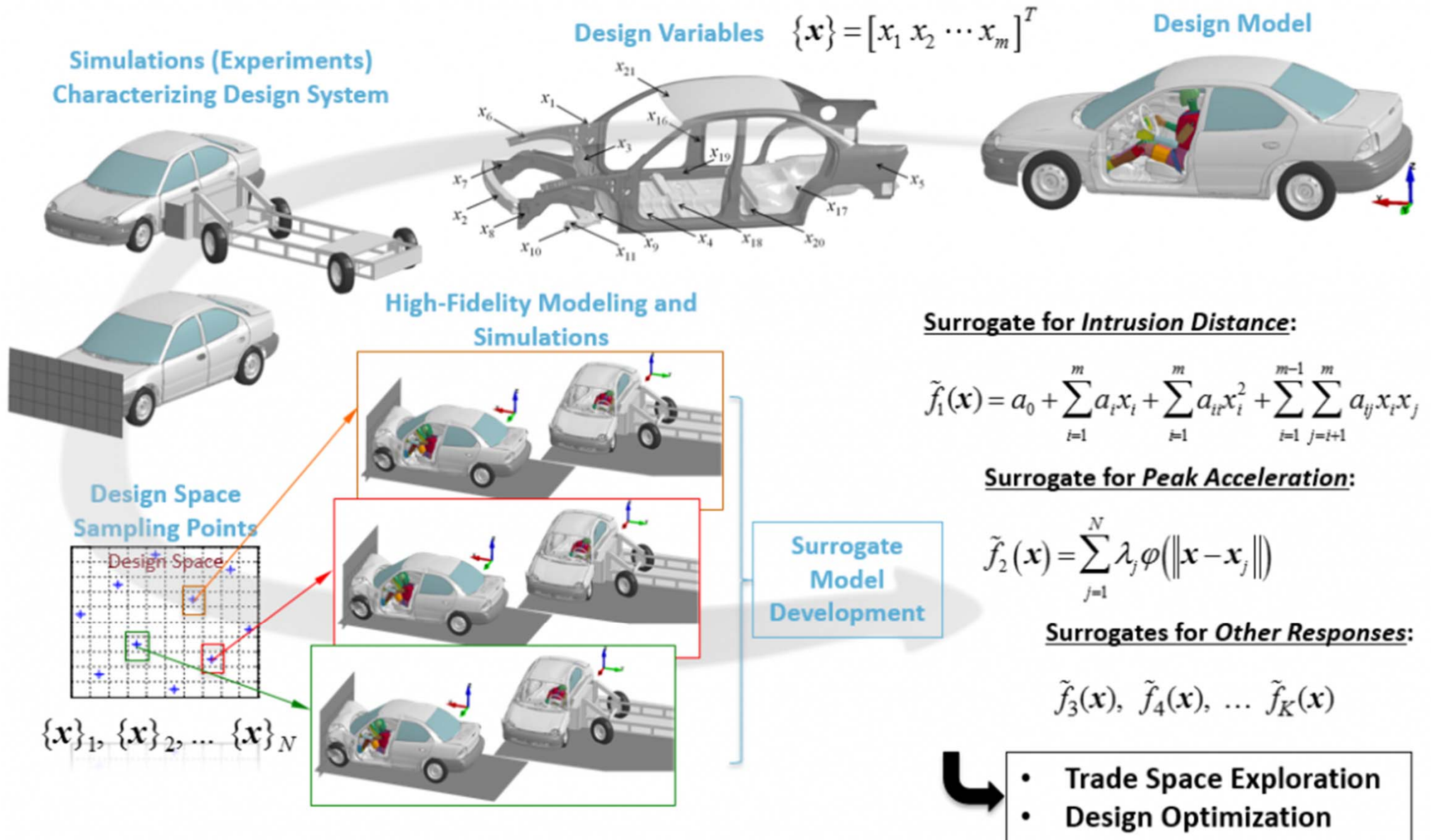
- An engineering method used when a quantity of interest (QoI) cannot be easily directly measured (calculated) so an approximate (**surrogate**) of the QoI is used instead
- Engineering design problems require experiments and/or simulations to evaluate design objective and constraints as a function of design variables
 - Ex) To find the optimal airfoil shape, an engineer simulates the airflow around the wing for different shape variables (length, curvature, material, ..)
- Single simulation is expensive, and yet, design optimization, design space exploration, sensitivity analysis and what-if analysis require thousands or millions of simulations ➡ **Use surrogate!**

What is a surrogate model? cont.

- Surrogate models, response surface models, metamodels or emulators
 - mimic (**approximate**) the behavior of the simulation model as closely as possible while being computationally cheap(er) to evaluate
 - Surrogate models does not concern physics or theory in the simulation. It only concerns input-output behavior
 - A simple mathematical model is constructed based on limited number of input-output pair of data (**samples**)
 - This approach is also known as behavioral modeling or **black-box** modeling
 - When only a single design variable is involved, the process is known as **curve-fitting**

Example of surrogate modeling

Surrogate Modeling of Crash-Induced Responses



Goals of surrogate modeling

- The generation of a surrogate that is as **accurate** as possible, using as **few simulation evaluations** as possible
 - Sample selection (sequential design, optimal experimental design (OED), **design of experiments** (DOE), or active learning)
 - Construction of the surrogate model and optimizing the model parameters (bias-variance trade-off)
 - Appraisal of the accuracy of the surrogate.
- The accuracy of the surrogate depends on the **number and location of samples** (experiments or simulations).
- Various design of experiments (DOE) techniques cater to different sources of errors, in particular, errors due to **noise** in the data or errors due to an **improper surrogate model**.

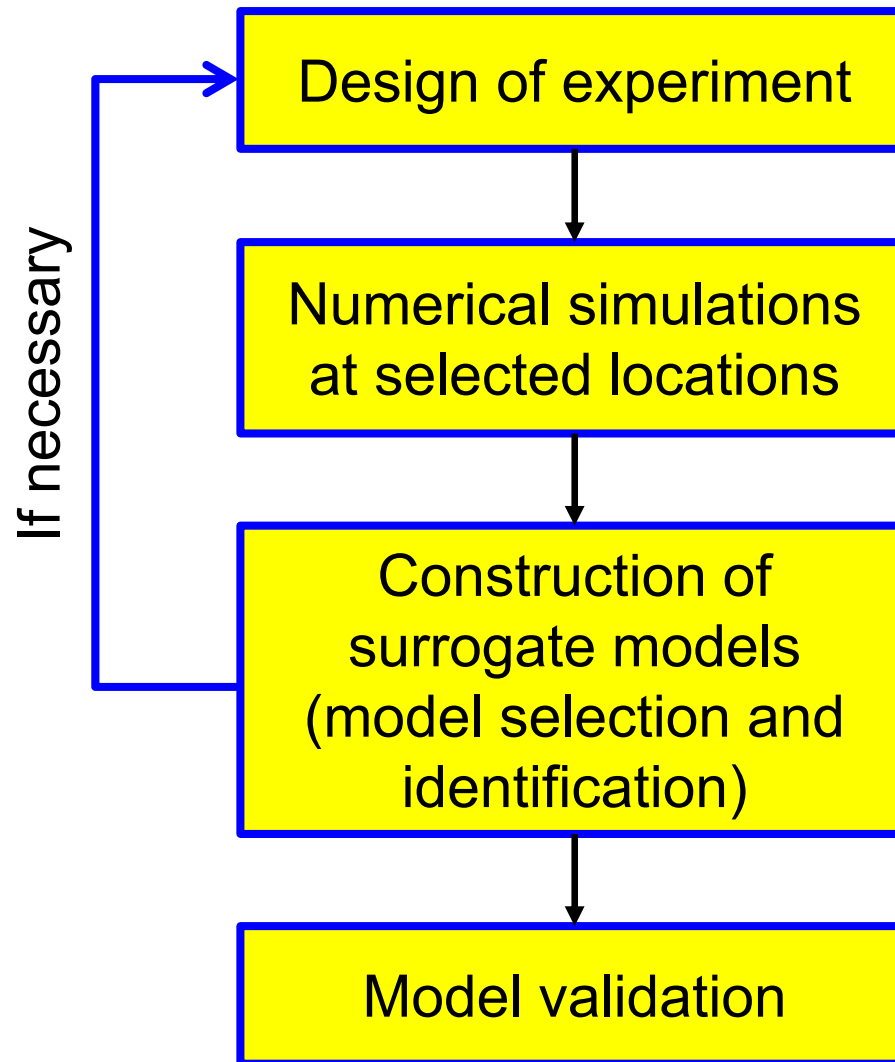
Which surrogate is the best?

- Polynomial response surfaces (PRS), Kriging, Radial basis function, Support vector machines, Space mapping, Artificial neural networks, and Bayesian networks
- When the nature of true function is not known a priori so it is not clear which surrogate model will be most accurate
- When the nature of true function is known, physics-based surrogates such as space-mapping based models are the most efficient
- There is **no consensus** on how to obtain the most reliable estimates of the accuracy of a given surrogate
- **Ensemble of multiple surrogates** can be used to reduce the risk of using a bad surrogate

Outline of surrogate modeling module

- Local approximation and local-global approximation
- Surrogate construction
- Linear regression accuracy 1, 2, 3
- Neural network model
- Radial basis neural network
- Kriging surrogate 1, 2
- Sampling plans 1, 2
- Nonlinear regression
- Multi-fidelity surrogate
- Moving least squares method 1, 2

Flow-chart of surrogate modeling



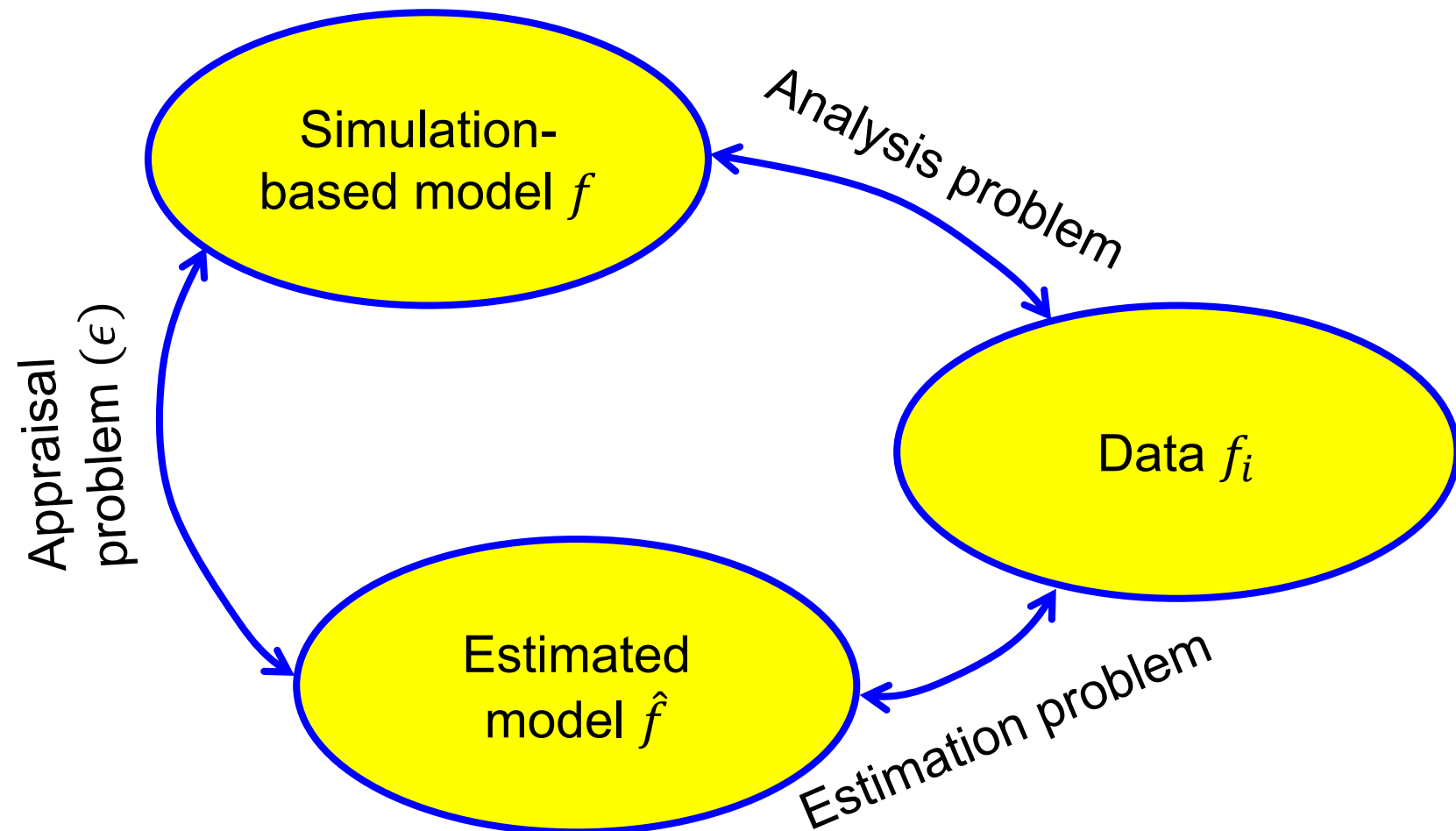
Flow-chart of surrogate modeling *cont.*

- Design of experiments (DOE):
 - Sampling plan in design variable space
 - How to assess the **goodness** of such designs
 - The number of samples is severely **limited** by computational cost
- Numerical simulations at selected locations
 - Expensive numerical simulation or experiments
 - Data may include random noise or error

Flow-chart of surrogate modeling *cont.*

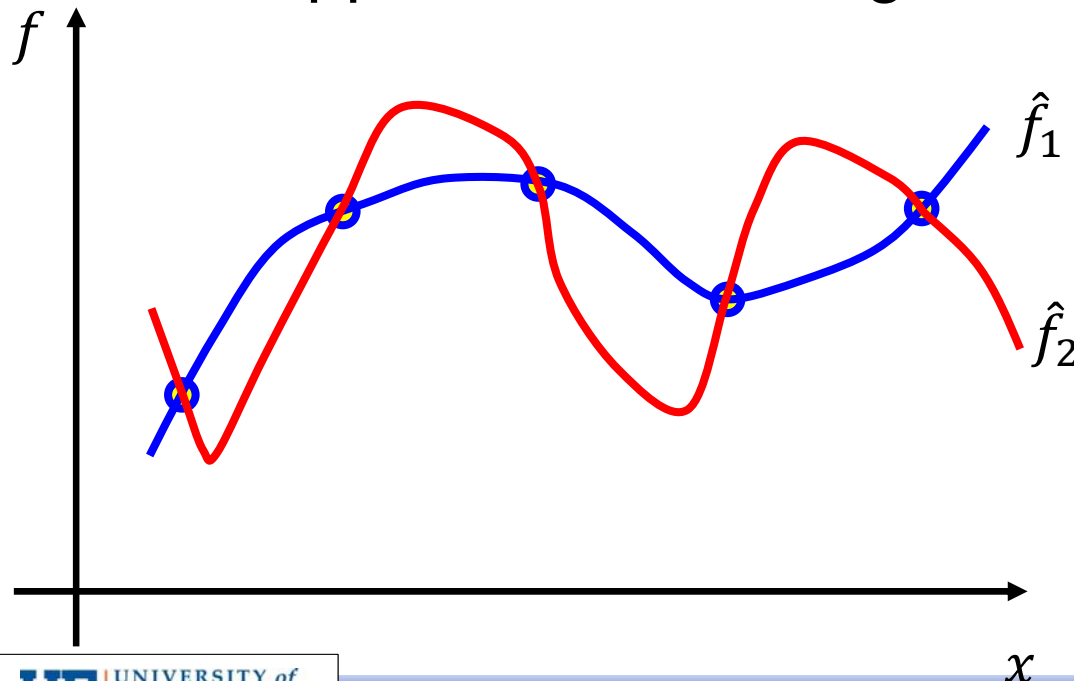
- Construction of surrogate model
 - Surrogate model selection
 - Model identification (determining unknown parameters)
- Model validation
 - Predictive capabilities of the surrogate model at unsampled points

Overview of surrogate modeling



Overview of surrogate modeling *cont.*

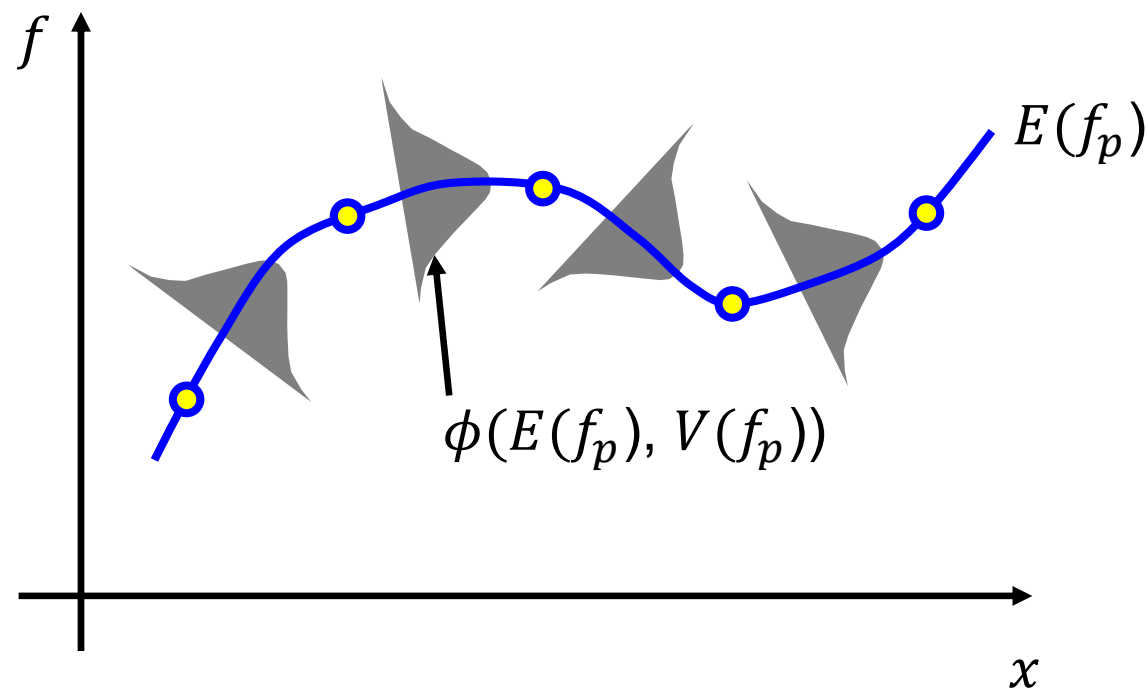
- Surrogate model: nonlinear inverse problem
 - We want to determine a continuous function (\hat{f}) as a function of design variables from a limited amount of data (\mathbf{f})
 - The data may be exact or noisy
- Model estimation: constructing a model from available data
- Model appraisal: assessing the errors ϵ attached to it



Multiple possibilities
of surrogates with
given data

Prediction uncertainty

- Surrogate prediction: $f_p(\mathbf{x}) = \hat{f}(\mathbf{x}) + \epsilon(\mathbf{x})$
- The prediction has expected value $E(f_p)$ and variance $V(f_p)$



Regularization

- Minimize error (loss function) + smoothness

$$\min_{\hat{f} \in H} \hat{f} = \frac{1}{N} \sum_{i=1}^N L[f_i - \hat{f}(\mathbf{x}_i)] + \lambda \int \|D^m \hat{f}\|_H d\mathbf{x}$$

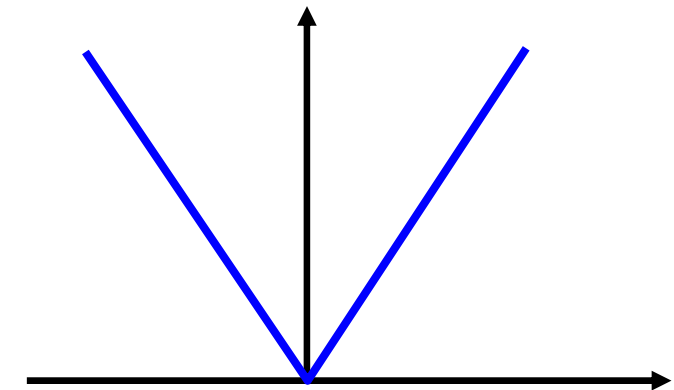
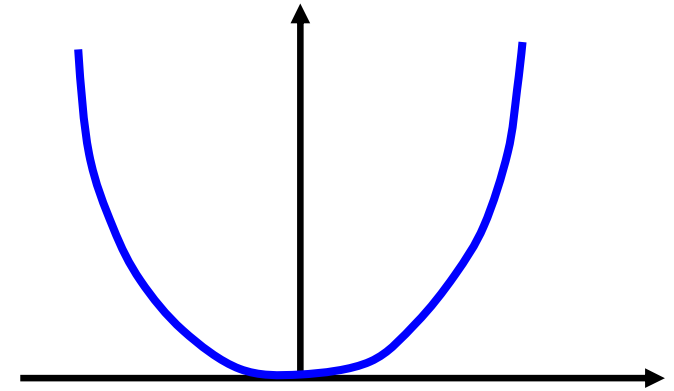
Closeness to the data

Smoothness of model

- H : family of surrogate models under consideration
- $L(x)$: a loss or cost function used to quantify the empirical error
- λ : regularization parameter ≥ 0
- $D^m \hat{f}$: m -th derivative of \hat{f} , a penalty term on smoothness

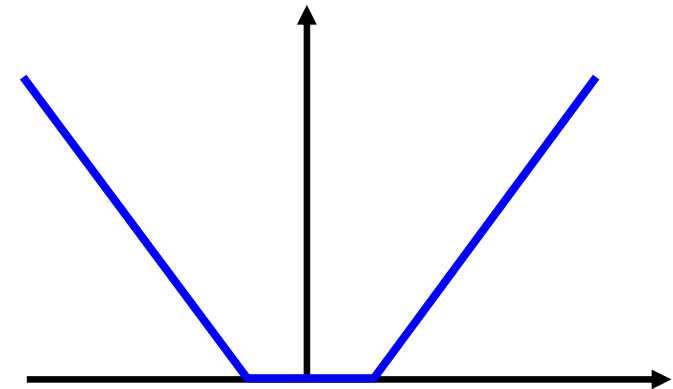
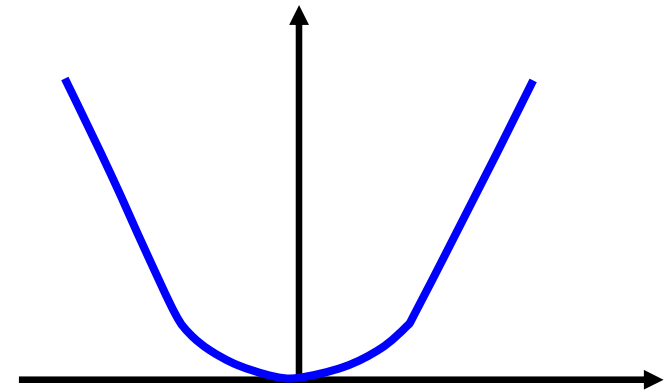
Loss function

- Loss function to determine the closeness of the model to the data
- Quadratic loss function (L2-norm):
 - Most commonly used (easy estimation of the parameters)
 - Sensitive to outliers
- Linear loss function (L1-norm):
 - Also called Laplace loss function
 - Use absolute value of the difference



Loss functions *cont.*


- Huber loss function:
 - Quadratic for small values of its argument and linear otherwise
- ϵ -loss function:
 - Popular in support vector regression surrogate
 - Error is considered to be zero when the difference is less than ϵ



Polynomial Response Surface



Background

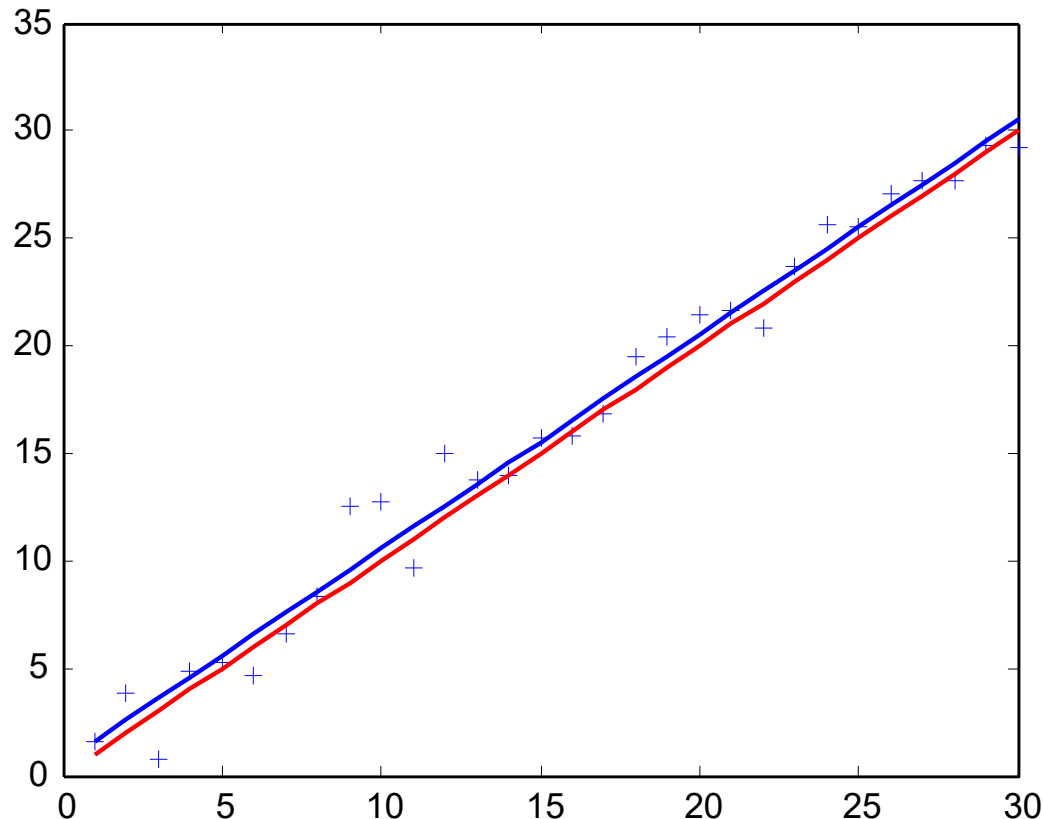
- Approximate (**surrogate**) discrete experimental data (**output**) with different conditions (**input**) using a simple polynomials
- Can compensate noise (ε) in measurement
- Later, extended to approximate simulation results with numerical noise
- Approximation  Accuracy must be checked
- **Polynomial response surface** (PRS) approximates data using a linear combination of polynomials
 - Bases are known (monomials), coefficients are unknown (need to be determined)

Curve fit metrics

- When we fit a curve to data we ask:
 - What is the error metric for the best fit?
 - What is more accurate, the data or the fit?
- This lecture deals with the following case:
 - The data is **noisy**
 - **The functional form of the true function is known**
 - The data is dense enough to allow us some noise filtering

Ex: Curve fitting

- Generate samples from $y = x$ (red) at $x=1,2,\dots,30$
- Add random noise $\sim N(0,1)$ and fit a linear polynomial (blue)
- Which one is more accurate? The fit or the data?



```
[p,s]=polyfit(x,y,1);  
yfit=polyval(p,x);  
plot(x,y,'+',x,x,'r',  
      x,yfit,'b')
```

With dense data, functional form is clear. Fit serves to **filter out noise**

Regression

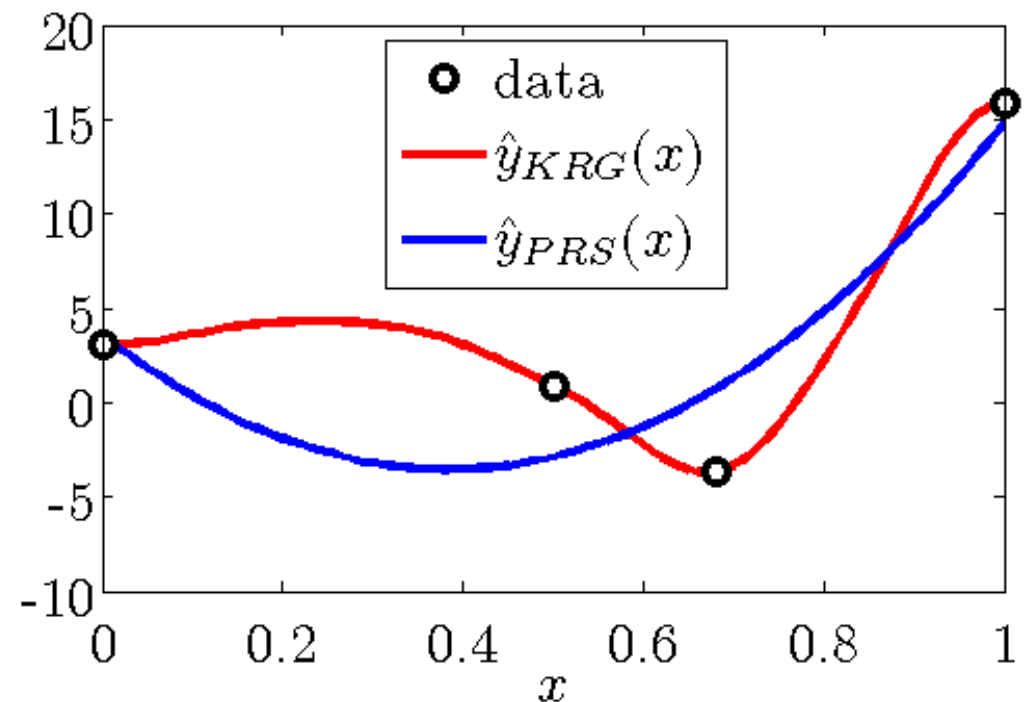
- The process of fitting data with a curve by minimizing the mean square difference from the data is known as **regression**
- Term originated from first paper to use regression dealt with a phenomenon called regression toward the mean (check Wikipedia)
- The polynomial regression on the previous slide is a simple regression, where we know or **assume the functional shape** and need to **determine only the coefficients**.

Surrogate (metamodel)

- The algebraic function we fit to data is called surrogate, metamodel or approximation.
- Polynomial surrogates were invented in the 1920s to characterize crop yields in terms of inputs such as water and fertilizer.
- They were called then “response surface approximations.”
- The term “surrogate” captures the purpose of the fit: using it instead of the data for prediction.
- Most important when data is expensive and noisy, especially for optimization.

Surrogates for fitting simulations

- Great interest now in fitting computer simulations
- Computer simulations are also subject to noise (numerical)
- Simulations are exactly repeatable, so noise is hidden.
- Some surrogates (e.g. polynomial response surfaces) cater mostly to noisy data.
- Some (e.g. Kriging) interpolate data.



Fitting a function with given data

- Approximate function
 - $y(\mathbf{x})$: response function (stress, displacement, cost, etc)
 - \mathbf{x} : vector of design variables, $\dim(\mathbf{x}) = n$
 - $\boldsymbol{\beta}$: vector of unknown parameters, $\dim(\boldsymbol{\beta}) = n_{\boldsymbol{\beta}}$

$$y(\mathbf{x}) = \hat{y}(\mathbf{x}, \boldsymbol{\beta}) + \epsilon$$

- $\hat{y}(\mathbf{x}, \boldsymbol{\beta})$: approximation of $y(\mathbf{x})$ (response surface)
 - ϵ : approximation error
- Goal: **determine $\boldsymbol{\beta}$ so that ϵ is minimized**

Question: What is the form of approximate function?

What measure is used to minimize the error?

How to determine β ?

- Assumption: The true function is unknown, but we can **evaluate it at discrete points**
- Perform n_y experiments: (\mathbf{x}_i, y_i) , $i = 1, \dots, n_y$

$$y_i = \hat{y}(\mathbf{x}_i, \boldsymbol{\beta}) + \epsilon_i, \quad i = 1, \dots, n_y$$

- We want to find β that will best fit the experiment data
- Ex) linear polynomials: $\hat{y}(\mathbf{x}, \boldsymbol{\beta}) = \beta_1 + \beta_2 x$
- Ex) rational function: $\hat{y}(\mathbf{x}, \boldsymbol{\beta}) = \frac{\beta_1}{x + \beta_2}$

Regression

- Finding β to best fit the data (minimize regression error)
- Need to define the regression error first
 - Root-mean-squared (RMS) error

$$e_{\text{rms}} = \sqrt{\frac{1}{n_y} \sum_{i=1}^{n_y} [y_i - \hat{y}(\mathbf{x}_i, \boldsymbol{\beta})]^2}$$

L_2 -norm

- Average error

$$e_{\text{av}} = \frac{1}{n_y} \sum_{i=1}^{n_y} |y_i - \hat{y}(\mathbf{x}_i, \boldsymbol{\beta})|$$

L_1 -norm

- Maximum error

$$e_{\text{max}} = \max_{n_y} |y_i - \hat{y}(\mathbf{x}_i, \boldsymbol{\beta})|$$

L_∞ -norm

Exercise

- In the curve-fitting example with the true function $y = x$
- Noisy data are fitted to a linear polynomial $y = 1.06x$
- The data at $x=10$ was $y_{10}=11$.
- What are (a) ε , (b) e_{10} , and (c) the surrogate error at $x = 10$?

Linear regression

- Exact $\boldsymbol{\beta}$ can be found when $n_y \rightarrow \infty$. For finite n_y , we can only find the estimate of $\boldsymbol{\beta} \rightarrow \mathbf{b}$
- **Linear regression**: $\hat{y}(\mathbf{x}, \boldsymbol{\beta})$ is a linear function of $\boldsymbol{\beta}$

$$\hat{y}(\mathbf{x}, \boldsymbol{\beta}) = \sum_{i=1}^{n_{\beta}} \beta_i \xi_i(\mathbf{x})$$

→ given shape function, **basis**

– Ex) linear approximation: $\xi_1 = 1, \xi_2 = x \rightarrow \hat{y}(x, \boldsymbol{\beta}) = \beta_1 + \beta_2 x$

Only $\boldsymbol{\beta}$ is unknown

Regress error

- Regression error vector

- with n_y samples (\mathbf{x}_i, y_i)

$$\mathbf{e} = \mathbf{y} - \mathbf{X} \cdot \mathbf{b}$$

$$e_j = y_j - \sum_{i=1}^{n_\beta} b_i \xi_i(\mathbf{x}_j)$$

Evaluation of basis
at sample points

- In matrix notation

$$\begin{Bmatrix} e_1 \\ e_2 \\ \vdots \\ e_{n_y} \end{Bmatrix} = \begin{Bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n_y} \end{Bmatrix} - \begin{bmatrix} \xi_1(\mathbf{x}_1) & \xi_2(\mathbf{x}_1) & \cdots & \xi_{n_\beta}(\mathbf{x}_1) \\ \xi_1(\mathbf{x}_2) & \xi_2(\mathbf{x}_2) & \cdots & \xi_{n_\beta}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \xi_1(\mathbf{x}_{n_y}) & \xi_2(\mathbf{x}_{n_y}) & \cdots & \xi_{n_\beta}(\mathbf{x}_{n_y}) \end{bmatrix} \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n_\beta} \end{Bmatrix}$$

\mathbf{X} : $n_y \times n_\beta$ design matrix

Determining unknown coefficients

- We will use RMS error
 - The magnitude of e_{rms} is not the focus, \mathbf{b} that minimizes e_{rms} is

$$e_{rms} = \sqrt{\frac{1}{n_y} \sum_{i=1}^{n_y} e_i^2} = \sqrt{\frac{1}{n_y} \mathbf{e}^T \mathbf{e}}$$

- Therefore, it is equivalent to minimize $\mathbf{e}^T \mathbf{e}$

$$\begin{aligned} \mathbf{e}^T \mathbf{e} &= (\mathbf{y} - \mathbf{X}\mathbf{b})^T (\mathbf{y} - \mathbf{X}\mathbf{b}) \\ &= \mathbf{y}^T \mathbf{y} - \mathbf{y}^T \mathbf{X}\mathbf{b} - \mathbf{b}^T \mathbf{X}^T \mathbf{y} + \mathbf{b}^T \mathbf{X}^T \mathbf{X}\mathbf{b} \\ &\quad \underbrace{\mathbf{y}^T \mathbf{X}\mathbf{b} = \mathbf{b}^T \mathbf{X}^T \mathbf{y}}_{\text{Scalar}} \end{aligned}$$

Determining unknown coefficients *cont.*

- Minimum of function = zero derivative

$$\frac{d}{d\mathbf{b}}(\mathbf{e}^T \mathbf{e}) = -2\mathbf{X}^T \mathbf{y} + 2\mathbf{X}^T \mathbf{X} \mathbf{b} = 0$$


$$\mathbf{X}^T \mathbf{X} \mathbf{b} = \mathbf{X}^T \mathbf{y}$$

Equation of linear regression
Normal equation

$$\mathbf{b} = \underbrace{(\mathbf{X}^T \mathbf{X})^{-1}}_{n_\beta \times n_\beta \text{ matrix}} \mathbf{X}^T \mathbf{y}$$

$n_\beta \times n_\beta$ matrix: **ill-conditioned** for large n_β

- Remedy: Solve $\mathbf{X} \mathbf{b} = \mathbf{y}$ using QR decomposition
- Computationally **efficient**. Nonlinear regression requires solving an optimization problem to determine \mathbf{b}

Ex) Linear approximation

- Linear function with 3 data

$$\hat{y}(x, \mathbf{b}) = b_0 + b_1 x, \quad y(0) = 0, \quad y(1) = 1, \quad y(2) = 0$$

- Apply data to the model

$$\begin{pmatrix} y(0) = 0 = b_0 + b_1 \cdot 0 \\ y(1) = 1 = b_0 + b_1 \cdot 1 \\ y(2) = 0 = b_0 + b_1 \cdot 2 \end{pmatrix} \Rightarrow \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{Bmatrix} b_1 \\ b_2 \end{Bmatrix}}_{\mathbf{b}} = \underbrace{\begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}}_{\mathbf{y}}$$

$$\mathbf{X}^T \mathbf{X} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ 3 & 5 \end{bmatrix} \quad \mathbf{X}^T \mathbf{y} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}$$

Ex) Linear approximation cont.

- Linear regression equation

$$\mathbf{X}^T \mathbf{X} \mathbf{b} = \mathbf{X}^T \mathbf{y} \quad \Rightarrow \quad \begin{cases} 3b_0 + 3b_1 = 1 \\ 3b_0 + 5b_1 = 1 \end{cases} \Rightarrow \begin{cases} b_0 = \frac{1}{3} \\ b_1 = 0 \end{cases}$$

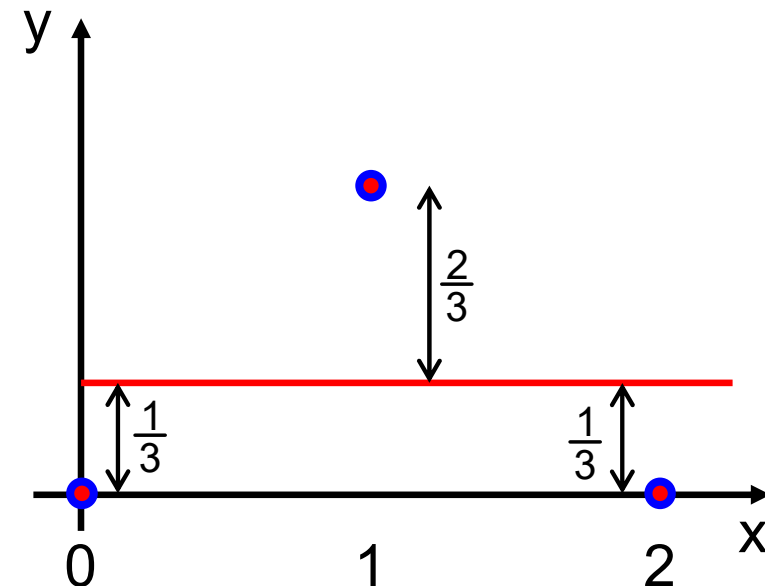
$$\hat{y}(x, \mathbf{b}) = \frac{1}{3}$$

Average value of data

- Errors

$$e_1 = -\frac{1}{3}, e_2 = \frac{2}{3}, e_3 = -\frac{1}{3}$$

$$e_{rms} = \sqrt{\frac{1}{3} \left[\left(-\frac{1}{3}\right)^2 + \left(\frac{2}{3}\right)^2 + \left(-\frac{1}{3}\right)^2 \right]} = 0.47$$



Ex) Linear approximation with other error metrics

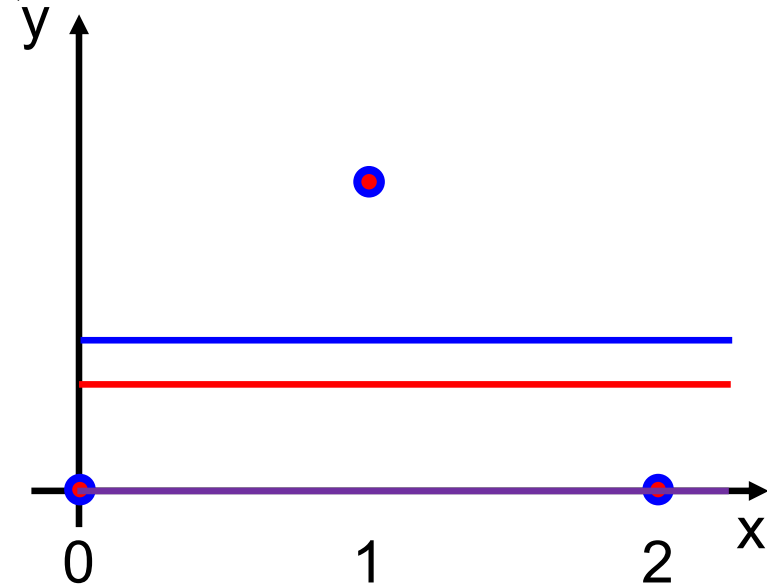
- Assuming other fits will lead to the form $\hat{y} = b$
- For **average error** minimize

$$3e_{av} = |0 - b| + |1 - b| + |0 - b| \Rightarrow b = 0$$

- For **maximum error** minimize

$$e_{\max} = \max(|0 - b|, |1 - b|, |0 - b|) \Rightarrow b = 0.5$$

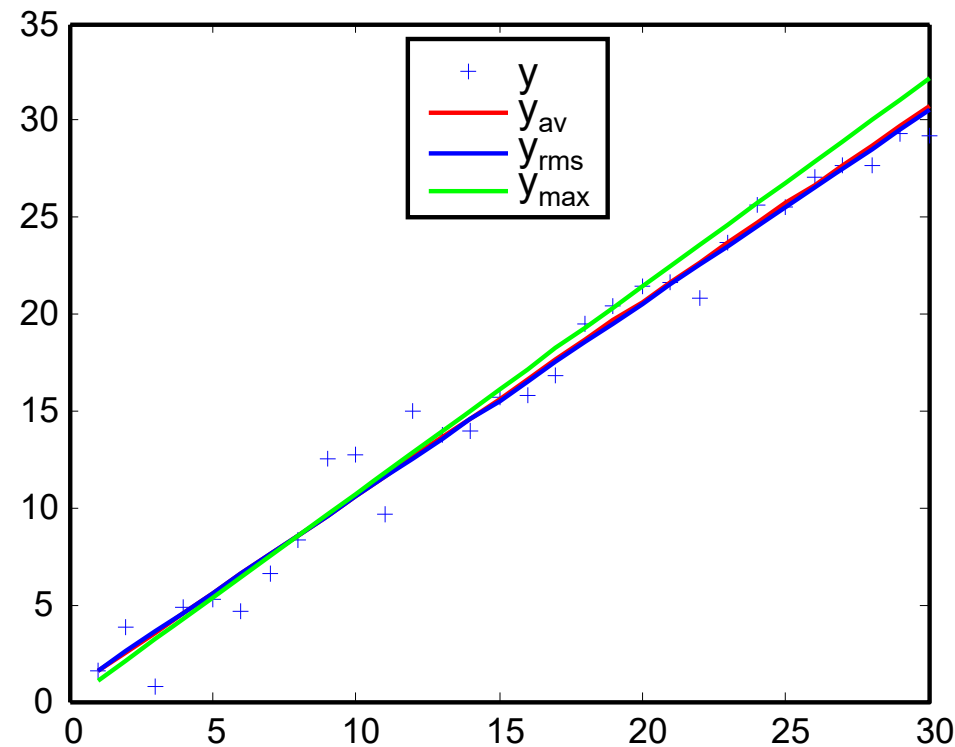
	RMS fit	Av. Err. fit	Max err. fit
RMS error	0.471	0.577	0.5
Av. error	0.444	0.333	0.5
Max error	0.667	1	0.5



Ex) Curve fit cont.

- Fitting $y = x$ with noisy data
- Use 3 error metrics to fit
- With dense data, difference due to metrics is small
- Max error:

```
f=@(b,x,y) max(abs(b(1)+b(2)*x-y))  
B=fminsearch(@(b) f(b,x,y),[0,1])
```



	RMS fit	Av. Err. fit	Max err. fit
RMS error	1.278	1.283	1.536
Av. error	0.958	0.951	1.234
Max error	3.007	2.987	2.934

$$\hat{y}_{rms} = 0.5981 + 0.997x$$

$$\hat{y}_{max} = 0.0003 + 1.0716x$$

$$\hat{y}_{av} = 0.5309 + 1.0067x$$

Exercises

- Find other metrics for a fit besides the three discussed in this lecture
- Redo the 30-point example with the surrogate $y = bx$
- Redo the 30-point example using only every third point ($x=3,6,\dots$). Compare the accuracy of the fit with regard to the true function. It is enough to use one error metric

Regression Accuracy

Polynomial response surface (PRS)



Global predictors of regression fidelity

- A measure to characterize the overall quality of a surrogate.
 - Goal: Evaluate accuracy of surrogate in design space
 - Reality: Errors are minimized at sample points
 - When $n_y = n_\beta$, surrogate passes through data points and $\varepsilon = 0$
- Equivalence measures (between surrogate and data)
 - Coefficient of multiple determination
 - Adjusted coefficient of multiple determination
- Prediction accuracy measures
 - Model independent: Cross validation error
 - Model dependent: Standard error

Errors at sample points

- After fitting surrogate $\hat{y}(\mathbf{x}, \mathbf{b}) = \sum_{i=1}^{n_\beta} b_i \xi_i(\mathbf{x})$, its prediction at data points are $\hat{\mathbf{y}} = \mathbf{X} \cdot \mathbf{b}$ and the errors are $\mathbf{e} = \hat{\mathbf{y}} - \mathbf{y}$

- Average error
$$e_{av} = \frac{1}{n_y} \sum_{i=1}^{n_y} |e_i|$$

- Maximum error
$$e_{max} = \max_{n_y} |e_i|$$

- RMS error
$$e_{rms} = \sqrt{\frac{SSe}{n_y}}$$

- Square-sum-error $SSe = \mathbf{e}^T \mathbf{e} = \mathbf{y}^T \mathbf{y} - 2\mathbf{b}^T \mathbf{X}^T \mathbf{y} + \mathbf{b}^T \mathbf{X}^T \mathbf{X} \mathbf{b}$

- These measures underestimate error at unsampled (prediction) points

- Errors are minimized at sample locations

Estimation of noise in data

- PRS assumes that the model $\hat{y}(\mathbf{x}, \mathbf{b}) = \sum_{i=1}^{n_\beta} b_i \xi_i(\mathbf{x})$ is **accurate** but the **data has a random noise** $\sim N(0, \sigma^2)$
- Unbiased estimate of σ

$$\hat{\sigma}^2 = \frac{\text{SSe}}{n_y - n_\beta}$$

- This is only reasonable when the model form is accurate
 - The model form error is embedded in the estimated noise

Ex) Fitting noise

- Equally spaced 5 samples of $y = x$ with noise $\sim N(0,1^2)$

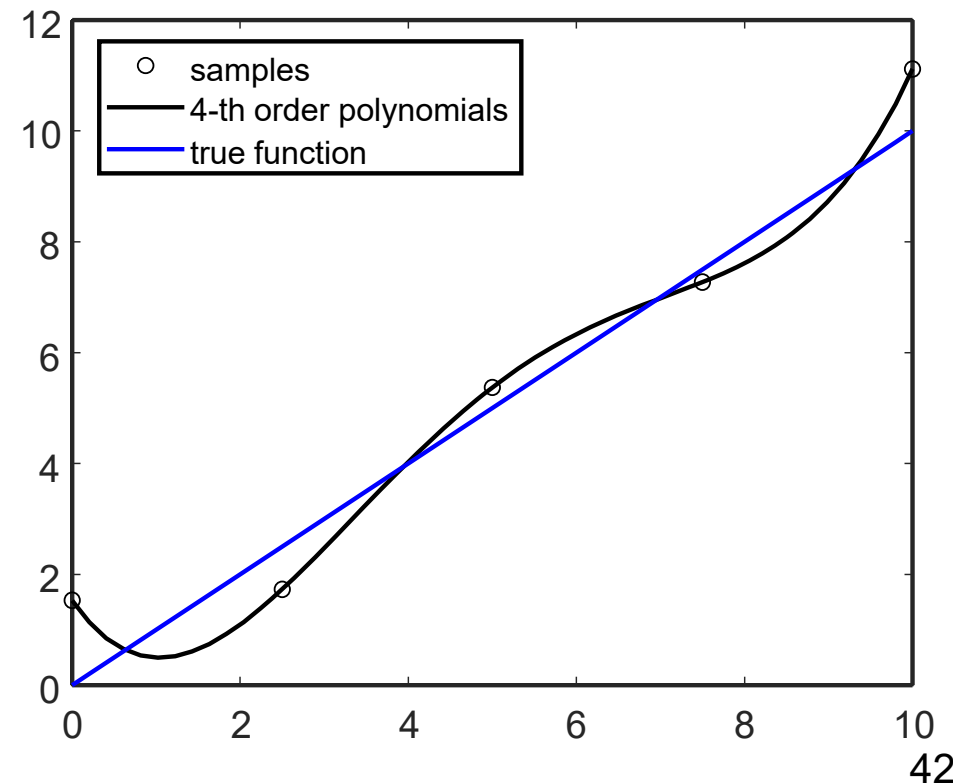
$x = [0, 2.5000, 5.0000, 7.5000, 10.0000]$

$y = [1.5326, 1.7303, 5.3714, 7.2744, 11.1174]$

- Fit the sample using 4-th order polynomials (5 coefficients)

$$\hat{y}(x) = 1.5326 - 2.1864x + 1.3397x^2 - 0.1970x^3 + 0.0094x^4$$

- Perfect fit for all data points
- The surrogate fits noise, not the trend (no noise canceling)



Ex) Estimated noise versus model error

- PRS assumes that the model is correct but data has noise
 - Estimated noise will be reasonable when the model is correct
 - If the model has an error, it will be included in the estimated noise
- Equally spaced 10 samples of $y = x^2$ with noise $\sim N(0,1^2)$

```
x=linspace(0,10,20);  
y=x.^2+randn(1,20);
```

- Fit the samples using linear and quadratic polynomials

$$\hat{y}_L(x) = -16.0902 + 10.0884x$$

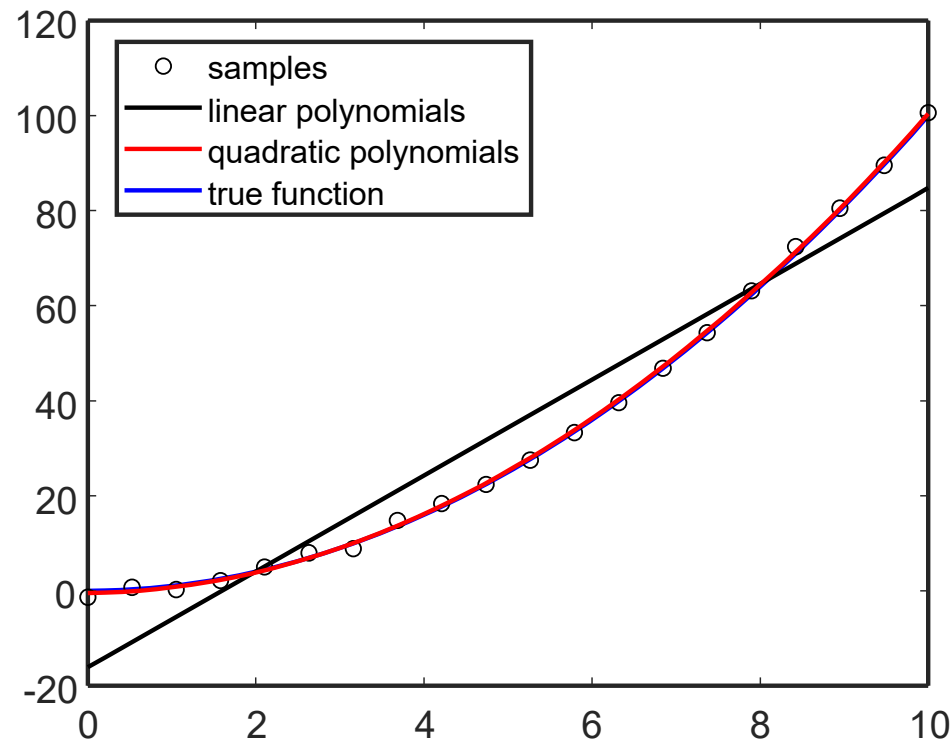
$$\hat{y}_Q(x) = -0.4847 + 0.2050x + 0.9883x^2$$

Ex) Estimated noise versus model error *cont.*

- Estimated noise

$$\hat{\sigma}_L = \sqrt{\frac{SSe}{20-2}} = 8.5807, \quad \hat{\sigma}_Q = \sqrt{\frac{SSe}{20-3}} = 1.7683$$

Include model error



Coefficient of multiple determination

- Equivalence of surrogate with data is measured by what fraction of variance in the data is captured by the surrogate.

$$SSy = \sum_{i=1}^{n_y} (y_i - \bar{y})^2, \quad SSr = \sum_{i=1}^{n_y} (\hat{y}_i - \bar{y})^2, \quad \bar{y} = \frac{1}{n_y} \sum_{i=1}^{n_y} y_i$$

Variation of data

Variation of prediction

Average of data

- Coefficient of multiple determination

$$R^2 = \frac{SSr}{SSy} = 1 - \frac{SSe}{SSy}$$

- Adjusted coefficient of multiple determination

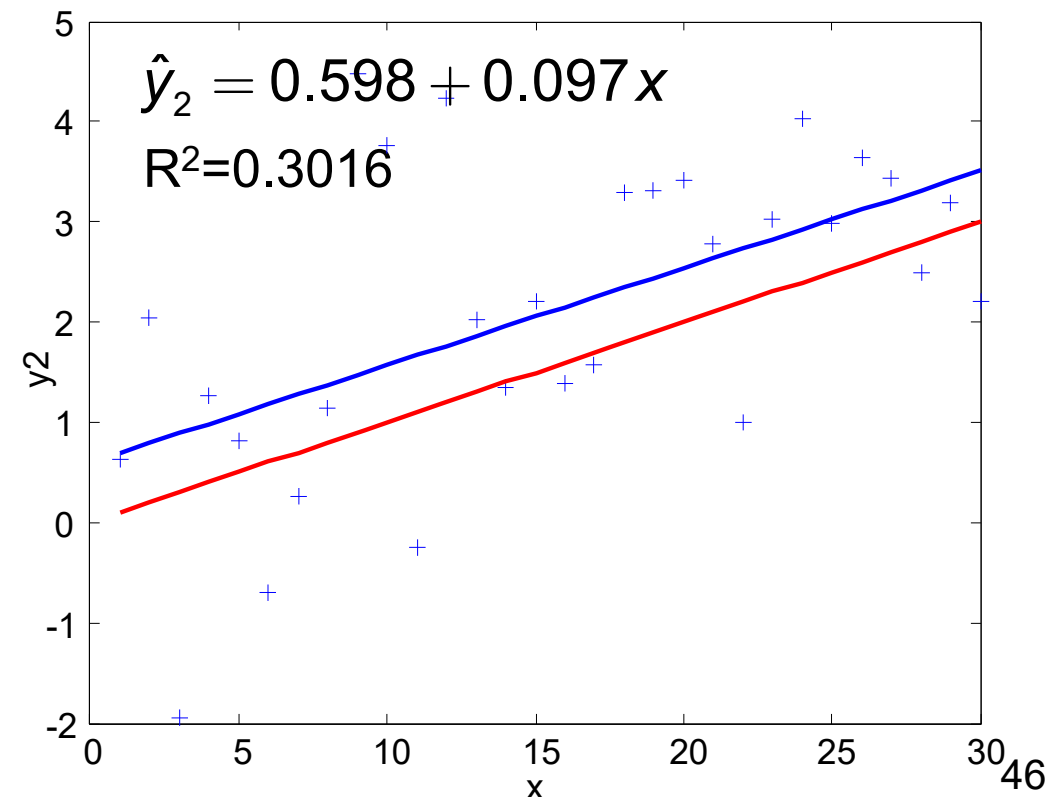
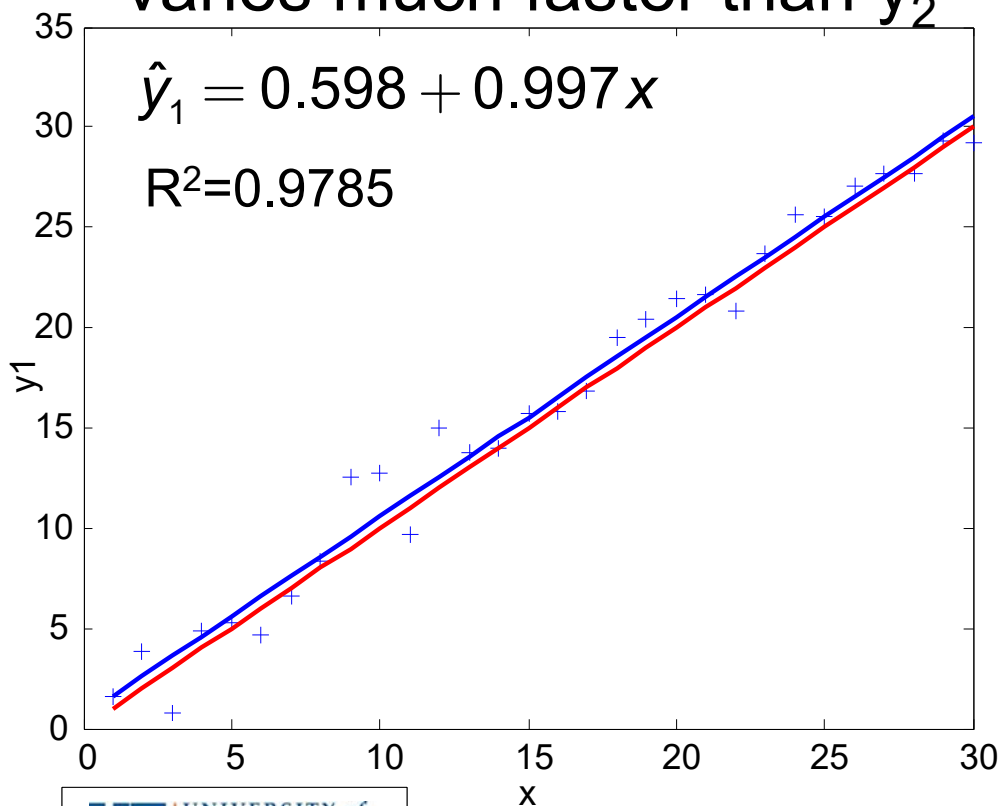
$$R_a^2 = 1 - (1 - R^2) \frac{n_y - 1}{n_y - n_\beta}$$

Penalize the number of coefficients

- Values larger than 0.9 is satisfactory
 - But not directly related to the magnitude of error

Ex) R^2 does not reflect accuracy

- Compare $y_1 = x$ to $y_2 = 0.1x$ plus noise $\sim N(0,1^2)$
- Estimate e_{av} and R^2 between the function (red) and surrogate (blue).
- e_{av} is similar, but R^2 are significantly different because y_1 varies much faster than y_2



Cross validation

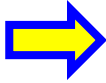
- Validation consists of checking the surrogate at a set of validation points.
- This may be considered wasteful because we do not use all the points for fitting the best possible surrogate.
- Cross validation divides data into n_g groups.
- Fit the approximation to $n_g - 1$ groups, and use last group to estimate error. Repeat for each group.
- When each group consists of one point, error often called **PRESS** (prediction error sum of squares)
- Calculate error at each point and then present RMS error

PRESS (prediction error sum of squares)

- Leave out data at point i
- Construct surrogate $\hat{y}_{(i)}$ \longleftrightarrow \hat{y} : Surrogate with all data
- PRESS residual: $e_{pi} = y_i - \hat{y}_{(i)}(\mathbf{x}_i)$
- Error: $e_i = y_i - \hat{y}(\mathbf{x}_i)$
- PRESS:
$$\text{PRESS} = \sqrt{\frac{1}{n_y - 1} \sum_{i=1}^{n_y} e_{pi}^2} = \sqrt{\frac{1}{n_y - 1} \sum_{i=1}^{n_y} [y_i - \hat{y}_{(i)}(\mathbf{x}_i)]^2}$$
 - This requires fitting n_y times. But it turns out that

$$e_{pi} = \frac{e_i}{1 - E_{ii}}$$

$$\mathbf{E} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \quad \text{Idempotent matrix}$$


$$\text{PRESS} = \sqrt{\frac{1}{n_y - 1} \sum_{i=1}^{n_y} \left(\frac{e_i}{1 - E_{ii}} \right)^2}$$

Scaled error with
 $1 - E_{ii}$ term
Requires one fitting

PRESS (prediction error sum of squares) *cont.*

- Property of the **idempotent matrix \mathbf{E}**

$$\hat{\mathbf{y}} = \mathbf{X} \cdot \mathbf{b} = \underbrace{\mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T}_{\mathbf{E}} \mathbf{y} = \mathbf{E} \cdot \mathbf{y}$$

Idempotent matrix \mathbf{E} : map $\mathbf{y} \rightarrow \hat{\mathbf{y}}$

– Large E_{ii} : large PRESS residual \rightarrow high influence point


- Variance of i th PRESS

$$V[e_{pi}] = V\left[\frac{e_i}{1 - E_{ii}}\right] = \left(\frac{\sigma}{1 - E_{ii}}\right)^2$$

Ex) Example: Linear regression

- True function: $y = x$
5 data: $(-2, -1.5), (-1, -1.5), (1, 1.25), (2, 1.75), (0, 0)$
- Linear fit: $y = b_1 + b_2x$

$$\mathbf{X} = \begin{bmatrix} 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} -1.5 \\ -1.5 \\ 0 \\ 1.25 \\ 1.75 \end{bmatrix}, \quad \mathbf{X}^T \mathbf{X} = \begin{bmatrix} 5 & 0 \\ 0 & 10 \end{bmatrix}, \quad \mathbf{X}^T \mathbf{y} = \begin{bmatrix} 0 \\ 9.25 \end{bmatrix} \Rightarrow \mathbf{b} = \begin{bmatrix} 0 \\ 0.925 \end{bmatrix}$$

 $\hat{y} = 0.925x$ $\mathbf{e} = \{0.35 \quad -0.575 \quad 0 \quad 0.325 \quad -0.1\}$

$$SSy = \sum_{i=1}^5 (y_i - \bar{y})^2 = 9.125, \quad SSe = \mathbf{e}^T \mathbf{e} = 0.56875$$

Ex) Linear regression *cont.*

- Coefficient of multiple determination

$$R^2 = 1 - \frac{SSe}{SSy} = 0.9377, \quad R_a^2 = 1 - (1 - 0.9377) \frac{4}{3} = 0.9169$$

Linear fit is satisfactory!!

- Error measures

$$e_{av} = \frac{1}{5} \sum_{i=1}^5 |e_i| = 0.27, \quad e_{max} = \max |e_i| = 0.575$$

$$e_{rms} = \sqrt{\frac{SSe}{5}} = 0.337$$


- Standard deviation of noise

$$\hat{\sigma} = \sqrt{\frac{SSe}{n_y - n_{\beta}}} = \sqrt{\frac{0.56875}{5 - 2}} = 0.4354 \quad \longleftrightarrow \quad \sigma = 0.395$$

Ex) Linear regression *cont.*

- Quadratic fit: $y = b_1 + b_2x + b_3x^2$

$$\mathbf{X} = \begin{bmatrix} 1 & -2 & 4 \\ 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} -1.5 \\ -1.5 \\ 0 \\ 1.25 \\ 1.75 \end{bmatrix}, \quad \mathbf{X}^T \mathbf{X} = \begin{bmatrix} 5 & 0 & 10 \\ 0 & 10 & 0 \\ 10 & 0 & 34 \end{bmatrix}, \quad \mathbf{X}^T \mathbf{y} = \begin{bmatrix} 0 \\ 9.25 \\ 0.75 \end{bmatrix} \Rightarrow \mathbf{b} = \begin{bmatrix} -0.1071 \\ 0.925 \\ 0.05357 \end{bmatrix}$$


$$\hat{y} = -0.1071 + 0.925x + 0.05357x^2$$

$$\mathbf{e} = \{-0.243 \quad 0.521 \quad -0.107 \quad -0.378 \quad 0.207\}$$

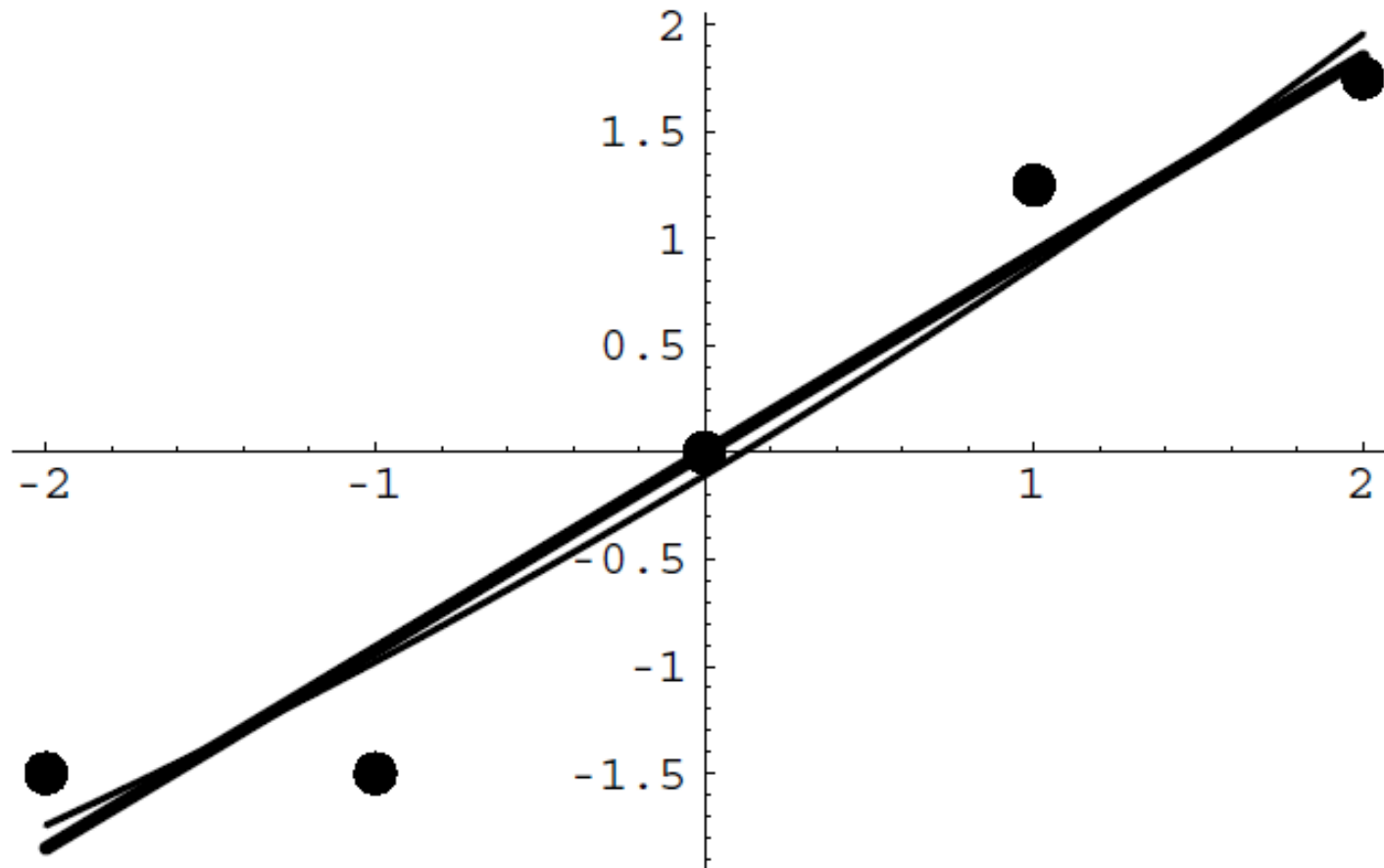
$$SSy = 9.125, \quad SSe = 0.52857, \quad R^2 = 0.9421, \quad R_a^2 = 0.8841$$

Improved Deteriorated

$$e_{av} = 0.291, \quad e_{max} = 0.521, \quad e_{rms} = 0.325, \quad \hat{\sigma} = 0.5141$$

Ex) Linear regression *cont.*

- we have not gained any predictive capabilities by adding the quadratic terms



Model based error for linear regression

- The common assumptions for linear regression
 - The **true function** is described by the functional form of the surrogate.
 - The **data** is contaminated with **normally distributed error** with the same standard deviation at every point.
 - The errors at different points are **not correlated**.
- Under these assumptions, the noise standard deviation (**standard error**) is estimated as

$$\hat{\sigma} = \sqrt{\frac{\mathbf{e}^T \mathbf{e}}{n_y - n_\beta}}$$

$$n_y \uparrow \Rightarrow \hat{\sigma} \downarrow$$

Large # of data makes surrogate more accurate than data

- $\hat{\sigma}$ is used as estimate of the **prediction error**. That is the error between the true function and the surrogate

Ex) Model-based error vs. PRESS

- 30 samples from $y = x$ and noise $\sim N(0,1^2)$

```
x=1:30; noise=randn(1,30); y=x+noise;
```

- Fitting linear function $\hat{y} = 0.5981 + 0.997x$

```
X=[ones(30,1),x'];
```

```
[B,BINT,R,RINT,STATS] = regress(y',X);
```

```
yfit=B(1)+B(2)*x;
```

```
error=y-yfit;
```

```
sigma=sqrt(error*error'/28)
```

Similar, due to finite samples

- Standard error = 1.3228
- $\text{mean}(\text{noise}) = 0.5519, \text{std}(\text{noise}) = 1.3$

```
mean(noise)
```

```
neutnoise=noise-mean(noise);
```

```
nnrms=sqrt(neutnoise*neutnoise'/29)
```

Ex) Model-based error vs. PRESS *cont.*

- PRESS = 1.3907

```
M=X'*X; E=X*inv(M)*X';
```

```
d=diag(E);
```

```
ep=error'./(1-d);
```

```
epress=sqrt(ep'*ep/29)
```

- Two errors are similar (1.3228 vs 1.3907)
 - And they are equal to $\text{std}(\text{noise}) = 1.3$
 - The actual error was only about 0.6 because the large amount of data filtered the noise.
- With less data, the differences will be larger.

Exercises

- The pairs (0,0), (1,1), (2,1) represent strain (millistrains) and stress (ksi) measurements.
 - Estimate Young's modulus using regression.
 - Calculate the error in Young modulus using cross validation both from the definition and from the formula.
- Repeat the example of $y = x$, using only data at $x = 3, 6, 9, \dots, 30$. Add noise $\sim N(0, 1^2)$.

Confidence in regression coefficients

- Random noise in data yields uncertainty in coefficients
 - Different set of random noise can fit different regression coefficients
 - We can consider \mathbf{b} as a random vector and estimate uncertainty
- Covariance matrix of coefficient vector \mathbf{b}

$$\Sigma_{\mathbf{b}} = [\mathbf{b} - E(\mathbf{b})][\mathbf{b} - E(\mathbf{b})]^T$$

$E(\mathbf{b})$: expected
(average) of \mathbf{b}

- Diagonal of $\Sigma_{\mathbf{b}}$: variance of b_i
- Off-diagonal of $\Sigma_{\mathbf{b}}$: correlation between b_i and b_j

$$\Sigma_{\mathbf{b}} = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}$$

Use estimated $\hat{\sigma}$

Confidence in regression coefficients *cont.*

- Standard deviation of **b**

$$se(b_i) = s_{b_i} = \hat{\sigma} \sqrt{(\mathbf{X}^T \mathbf{X})_{ii}^{-1}}$$

- Coefficient of variation of **b**


$$c_i = \frac{s_{b_i}}{|b_i|}$$

$c_i > 1$: very little confidence
on the coefficient


- t-statistics $= \frac{1}{c_i}$

- Backward elimination**: eliminate coefficient with the largest C.O.V.

Eliminating unimportant basis

- Importance of basis  magnitude of coefficients

$$\hat{y}(x_1, x_2) = 10.5 + \underset{\substack{\downarrow \\ \text{Linear } x_1 \text{ term} \\ \text{is not important}}}{0.01x_1} + 9.87x_2 - 5x_1^2 + 7.6x_1x_2 + \underset{\substack{\downarrow \\ \text{Quadratic } x_2 \text{ term} \\ \text{is not important}}}{0.02x_2^2}$$

 $\hat{y}(x_1, x_2) = 10.5 + 9.87x_2 - 5x_1^2 + 7.6x_1x_2$

- Remove those coefficients whose values cannot be estimated accurately for given data
 - These coefficients do not have much effect on the accuracy of the fit
 - May reduce prediction quality in the region where these coefficients have a large effect

Ex) Backward elimination

- Quadratic model of Example 3.2.1

$$\hat{y} = -0.1071 + 0.925x + 0.05357x^2$$

$$\hat{\sigma} = 0.5141$$

$$R_a^2 = 0.8841$$

$$(\mathbf{X}^T \mathbf{X})^{-1} = \begin{bmatrix} \frac{17}{35} & 0 & -\frac{1}{7} \\ 0 & \frac{1}{10} & 0 \\ -\frac{1}{7} & 0 & \frac{1}{14} \end{bmatrix}, \quad \Sigma_b = \begin{bmatrix} 0.1284 & 0 & -0.03776 \\ 0 & 0.02643 & 0 \\ -0.03776 & 0 & 0.01888 \end{bmatrix}$$

- Coefficient of variation

$$c_1 = \frac{\sqrt{0.1284}}{0.1071} = 3.35, \quad c_2 = \frac{\sqrt{0.02643}}{0.925} = 0.176, \quad c_3 = \frac{\sqrt{0.01888}}{0.05357} = 2.56$$




Eliminate \mathbf{b}_1

- Reduced quadratic model $\hat{y} = b_2x + b_3x^2$

Ex) Backward elimination *cont.*

- Reduced quadratic model $\hat{y} = b_1x + b_2x^2$

$$\mathbf{X} = \begin{bmatrix} -2 & 4 \\ -1 & 1 \\ 0 & 0 \\ 1 & 1 \\ 2 & 4 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} -1.5 \\ -1.5 \\ 0 \\ 1.25 \\ 1.75 \end{bmatrix} \Rightarrow \mathbf{b} = [0.925, 0.02205]$$

 $\hat{y}(x) = 0.925x + 0.02205x^2$

$$\mathbf{e} = [-0.262, 0.597, 0, -0.3038, 0.188]$$

$$SSy = 9.125, \quad SSe = 0.5522, \quad R^2 = 0.9394, \quad R_a^2 = 0.9193$$

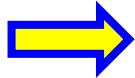
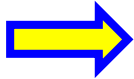
$$\hat{\sigma}^2 = \frac{SSe}{n_y - n_\beta} = 0.1841, \quad \hat{\sigma} = 0.429$$


Improved!



Ex) Backward elimination *cont.*

$$(\mathbf{X}^T \mathbf{X})^{-1} = \begin{bmatrix} \frac{1}{10} & 0 \\ 0 & \frac{1}{34} \end{bmatrix}, \quad \Sigma_{\mathbf{b}} = \begin{bmatrix} 0.0184 & 0 \\ 0 & 0.00541 \end{bmatrix}$$

 $c_1 = \frac{\sqrt{0.0184}}{0.925} = 0.147, \quad c_2 = 3.33$  Eliminate b_2

- Eliminate b_2  $\hat{y} = b_1 x$

$\hat{y}(x) = 0.925x, \quad \hat{\sigma} = 0.4354, \quad R_a^2 = 0.9169$  Slightly decreased

- Best fit: $\hat{y}(x) = 0.925x + 0.02205x^2$

Ex) Backward elimination *cont.*

- Change (perturb) data $y(1) = 1.35$ (8% perturbation)

- Quadratic fit

$$\hat{y} = -0.1071 + 0.925x + 0.05357x^2$$



$$\hat{y} = -0.0729 + 0.935x + 0.0465x^2$$

30%

1%

13% change

- After eliminating b_1

$$\hat{y}(x) = 0.925x + 0.02205x^2$$



$$\hat{y}(x) = 0.935x + 0.025x^2$$

1%

13% change

Prediction variance in linear regression

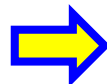
- Assumptions on noise in linear regression allow us to estimate the **prediction variance** due to the noise at any point.
- Prediction variance is usually large when you are far from a data point.
- We distinguish between **interpolation**, when we are in the convex hull of the data points, and **extrapolation** where we are outside.
- Extrapolation is associated with larger errors, and in high dimensions it usually cannot be avoided.

Prediction variance

- Linear regression model $\hat{y} = \sum_{i=1}^{n_\beta} b_i \xi_i(\mathbf{x})$

- Define $x_i^{(m)} = \xi_i(\mathbf{x})$, then $\hat{y} = \boldsymbol{\xi}^\top \mathbf{b}$

- Ex) $y = ax$, $V[x] = \sigma^2$

 $V[y] = a^2 V[x] = a^2 \sigma^2$

- Prediction variance:

$$V[\hat{y}(\mathbf{x})] = \boldsymbol{\xi}^\top \boldsymbol{\Sigma}_b \boldsymbol{\xi} = \hat{\sigma}^2 \boldsymbol{\xi}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \boldsymbol{\xi}$$

- Standard error of prediction

$$s_y = \hat{\sigma} \sqrt{\boldsymbol{\xi}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \boldsymbol{\xi}}$$

Data sensitivity

- **Data sensitivity**: how much the prediction at \mathbf{x} will vary due to a change in data y_i ?
- Linear regression surrogate

$$\hat{y}(\mathbf{x}) = \boldsymbol{\xi}^T \mathbf{b} = \boldsymbol{\xi}^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

- Differentiate $\hat{y}(\mathbf{x})$ with respect to i th component of \mathbf{y}

$$\frac{\partial \hat{y}(\mathbf{x})}{\partial y_i} = \left\{ \boldsymbol{\xi}^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \right\}_i$$

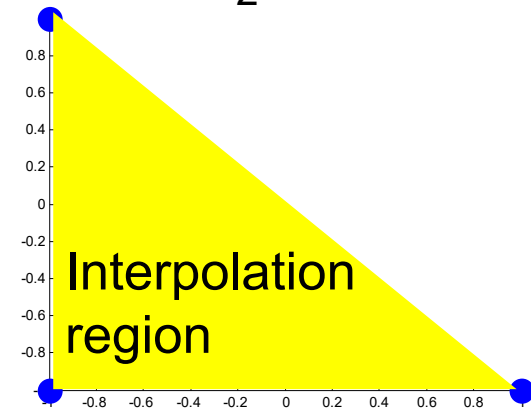
- Sensitivity of prediction with respect to change in data

Ex) Prediction variance

- For a linear polynomial RS $y = b_1 + b_2x_1 + b_3x_2$, find the prediction variance in the region $-1 \leq x_1 \leq 1$, $-1 \leq x_2 \leq 1$

(a) For data at three vertices

$$\mathbf{x}_1 = \begin{Bmatrix} -1 \\ -1 \end{Bmatrix}, \quad \mathbf{x}_2 = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix}, \quad \mathbf{x}_3 = \begin{Bmatrix} 1 \\ -1 \end{Bmatrix}$$



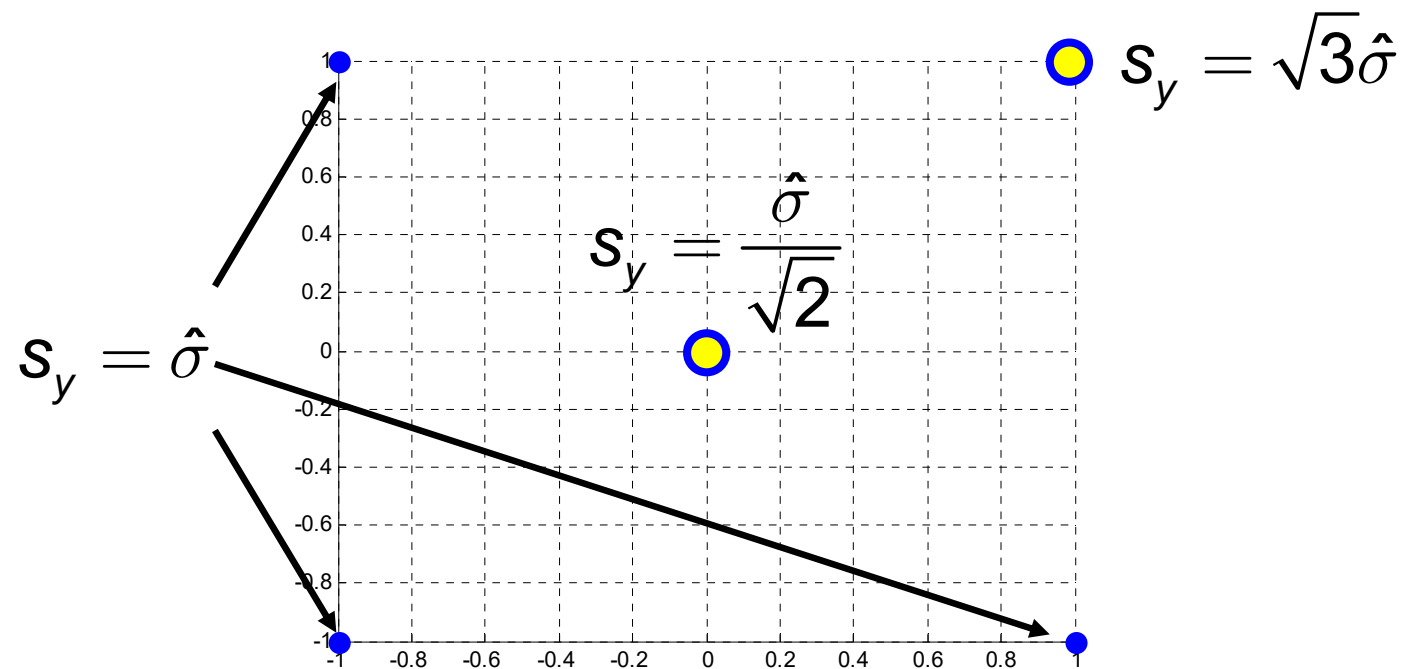
$$\xi = \begin{Bmatrix} 1 \\ x_1 \\ x_2 \end{Bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & -1 & -1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix}, \quad \mathbf{X}^T \mathbf{X} = \begin{bmatrix} 3 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{bmatrix}$$

$$(\mathbf{X}^T \mathbf{X})^{-1} = 0.25 \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

Ex) Prediction variance: Interpolation vs. Extrapolation

- Prediction standard error

$$s_y = \hat{\sigma} \sqrt{\boldsymbol{\xi}^T (\mathbf{X}^T \mathbf{X})^{-1} \boldsymbol{\xi}} = \hat{\sigma} \sqrt{0.5(1 + x_1 + x_2 + x_1^2 + x_2^2 + x_1 x_2)}$$

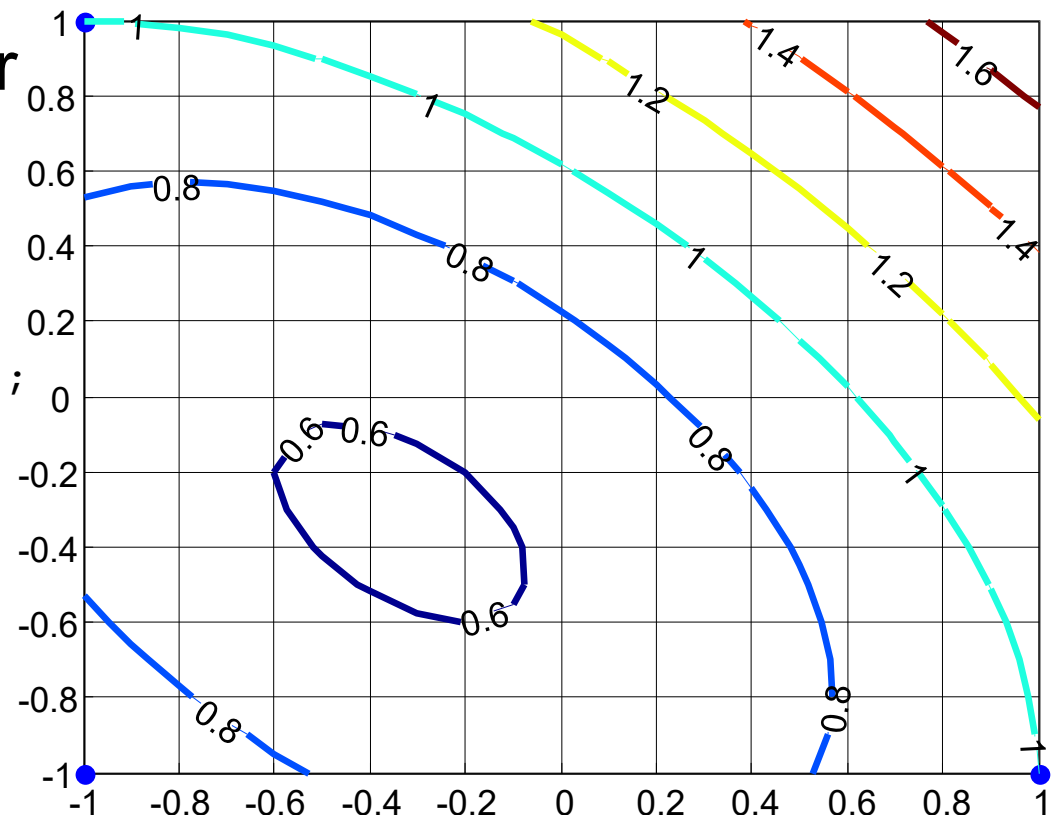


- Prediction error increased at an extrapolation point (1,1)

Ex) Prediction standard error contour

- Find the minimum error location by $\frac{\partial s_y}{\partial x_1} = \frac{\partial s_y}{\partial x_2} = 0$
- Minimum $s_y = \frac{1}{\sqrt{3}} \hat{\sigma}$ at $x_1 = x_2 = -\frac{1}{3}$
- What is special about this point?
- Prediction variance contour

```
x=[-1 -1 1]; y=[-1 1 -1];  
[X,Y]=meshgrid(-1:.1:1, -1:.1:1);  
Z=sqrt(.5*(1+X+Y+X.^2+Y.^2+X.*Y));  
v=linspace(0.6,1.8,7)  
scatter(x,y,'filled');  
grid on; hold on  
[C,h]=contour(X,Y,Z,v);  
clabel(C,h)
```




Ex) Data at four vertices

- Add additional data at (1,1)

$$\mathbf{x}_1^T = [-1, -1], \mathbf{x}_2^T = [-1, 1], \mathbf{x}_3^T = [1, -1], \mathbf{x}_4^T = [1, 1]$$

$$\mathbf{X} = \begin{bmatrix} 1 & -1 & -1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \\ 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{X}^T \mathbf{X} = 4 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

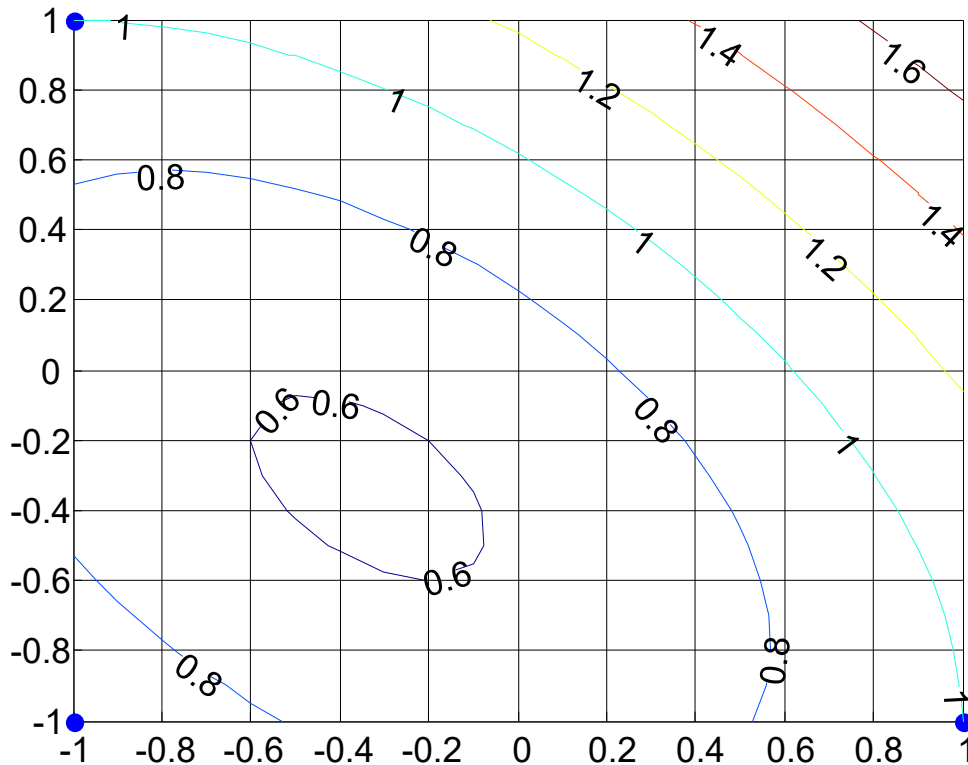

$$\xi^T (\mathbf{X}^T \mathbf{X})^{-1} \xi = 0.25(1 + x_1^2 + x_2^2)$$

- Error at vertices: $s_y = \frac{\sqrt{3}}{2} \hat{\sigma}$
- At the origin minimum is $s_y = \frac{1}{2} \hat{\sigma}$

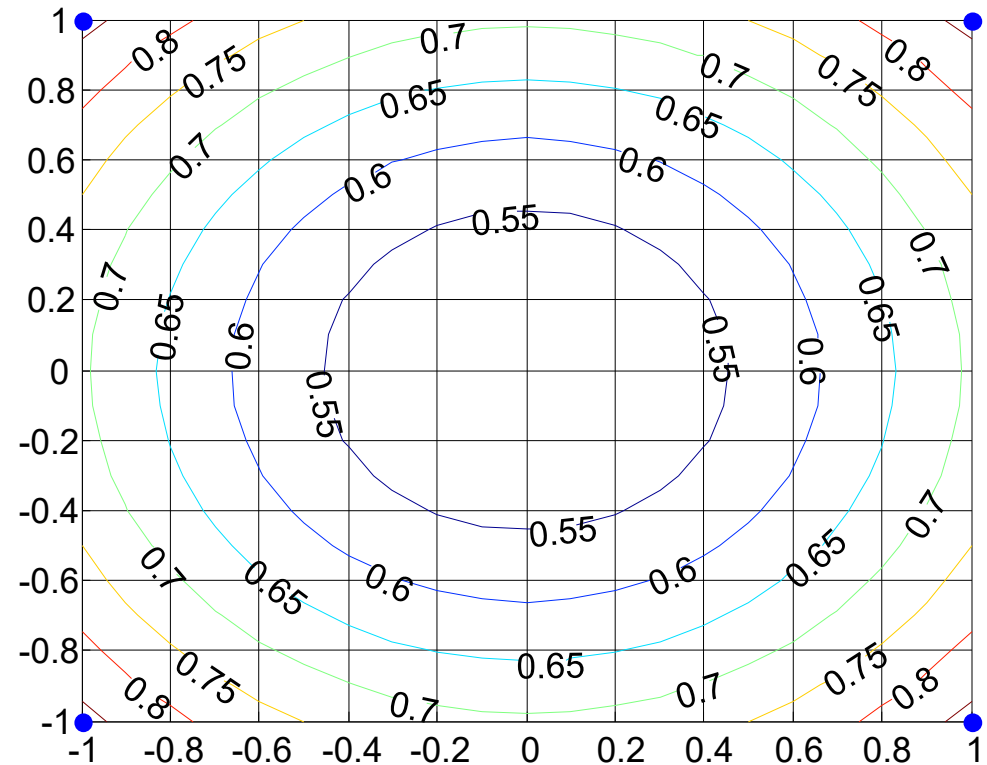
Surrogate is more accurate
than the data

Ex) Graphical comparison of standard errors

3 sampling points



4 sampling points



- Additional sample does not improve the low prediction variance, but significantly reduce the large prediction variance in the extrapolation region

Exercise

- Redo analytically the four point example, when the data points are not at the corners but inside the domain, at ± 0.7 . What does the difference in the results tells you?
- For a grid of 3x3 data points, compare the standard error contours for linear and quadratic fits.

Prediction variance with variable noise

- So far, we consider the case when noise of all data points are from an identical random distribution $\sim N(0, \sigma^2)$

$$V[\hat{y}(\mathbf{x})] = \sigma^2 \boldsymbol{\xi}^T (\mathbf{X}^T \mathbf{X})^{-1} \boldsymbol{\xi}$$

- In some cases, noises at different locations may have different magnitudes
- Data sensitivity can be used to estimate the prediction variance when each data has noise variance σ_i^2

$$V[\hat{y}(\mathbf{x})] = \sum \left(\frac{\partial \hat{y}(\mathbf{x})}{\partial y_i} \right)^2 \sigma_i^2$$

Ex) Example 3.2.1

- Linear fit the data

X	-2	-1	0	1	2
Y	-1.5	-1.5	0	1.25	1.75

$$\mathbf{X} = \begin{bmatrix} 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix} \quad \mathbf{y} = \begin{Bmatrix} -1.5 \\ -1.5 \\ 0 \\ 1.25 \\ 1.75 \end{Bmatrix} \quad \mathbf{X}^T \mathbf{X} = \begin{bmatrix} 5 & 0 \\ 0 & 10 \end{bmatrix} \quad \boldsymbol{\xi} = \begin{Bmatrix} 1 \\ x \end{Bmatrix}$$

- PRS

$$\hat{y}(x) = \boldsymbol{\xi}^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} = 0.925x$$

- Data sensitivity

$$\frac{\partial \hat{y}}{\partial \mathbf{y}} = \boldsymbol{\xi}^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T = 0.1 [2 - 2x \quad 2 - x \quad 2 \quad 2 + x \quad 2 + 2x]$$

Ex) Example 3.2.1 prediction variance at $x=3$

- Prediction variance

$$\begin{aligned} V[\hat{y}(\mathbf{x})] &= \sigma^2 \boldsymbol{\xi}^T (\mathbf{X}^T \mathbf{X})^{-1} \boldsymbol{\xi} \\ &= \sigma^2 \begin{bmatrix} 1 & x \end{bmatrix} \begin{bmatrix} 0.2 & 0 \\ 0 & 0.1 \end{bmatrix} \begin{Bmatrix} 1 \\ x \end{Bmatrix} = (0.2 + 0.1x^2) \sigma^2 = 1.1\sigma^2 \end{aligned}$$

- Variance of prediction

$$\frac{\partial \hat{y}}{\partial \mathbf{y}} = 0.1 \begin{bmatrix} -4 & -1 & 2 & 5 & 8 \end{bmatrix}$$

- If all data variances are the same, $V[\hat{y}(x)] = 1.1\sigma^2$
- If not, variance of y_5 is most important
 - y_5 is closest to $x = 3$

Bootstrapping

- When we calculate statistics from random data bootstrapping can provide **error estimates**.
- If we had multiple sets of samples, we could use them to estimate the error in the computation.
- With bootstrapping we perform the amazing feat of getting the error from a single set of samples.
- This is done by **resampling with replacement** of the same data.
 - We draw a samples from the original data without removing it so that the new sample may have **repetitions**.
- We repeat for many bootstrap samples to get a **distribution of the statistic of interest**.

Matlab bootstrap routine

- `bootstat = bootstrap(nboot,bootfun,d1,...)` draws `nboot` bootstrap data samples, computes statistics on each sample using `bootfun`, and returns the results in the matrix `bootstat`. `bootfun` is a function handle specified with `@`. Each row of `bootstat` contains the results of applying `bootfun` to one bootstrap sample.
- `[bootstat,bootsam] = bootstrap(...)` returns an `n`-by-`nboot` matrix of bootstrap indices, `bootsam`. Each column in `bootsam` contains indices of the values that were drawn from the original data sets to constitute the corresponding bootstrap sample

Ex) Sample mean

- Generate 10 random samples: $x = \text{randn}(1,10)$
 $x = [0.5377, 1.8339, -2.2588, 0.8622, 0.3188,$
 $-1.3077, -0.4336, 0.3426, 3.5784, 2.7694];$
- `[bootstat, bootsam]=bootstrp(1000,@mean,x);`
 - Generate 1000 bootstrap sets of samples and means
- `bootsam(:,1:5)` shows the first 5 sets of bootstrap samples

6	3	2	8	1
4	9	10	3	10
10	2	10	2	8
9	3	5	3	5
6	2	2	4	6
7	3	3	5	3
6	5	5	6	5
3	4	6	1	10
4	10	3	3	6
5	5	7	9	6

Each column contains the indices of one set of bootstrap samples. For example, the last column indicates that we drew $x(1)$, $x(10)$, $x(8)$, $x(5)$, $x(6)$, $x(3)$, $x(5)$, $x(10)$, $x(6)$, and $x(6)$. That is, we drew $x(6)$ three times, $x(5)$ and $x(10)$ twice.

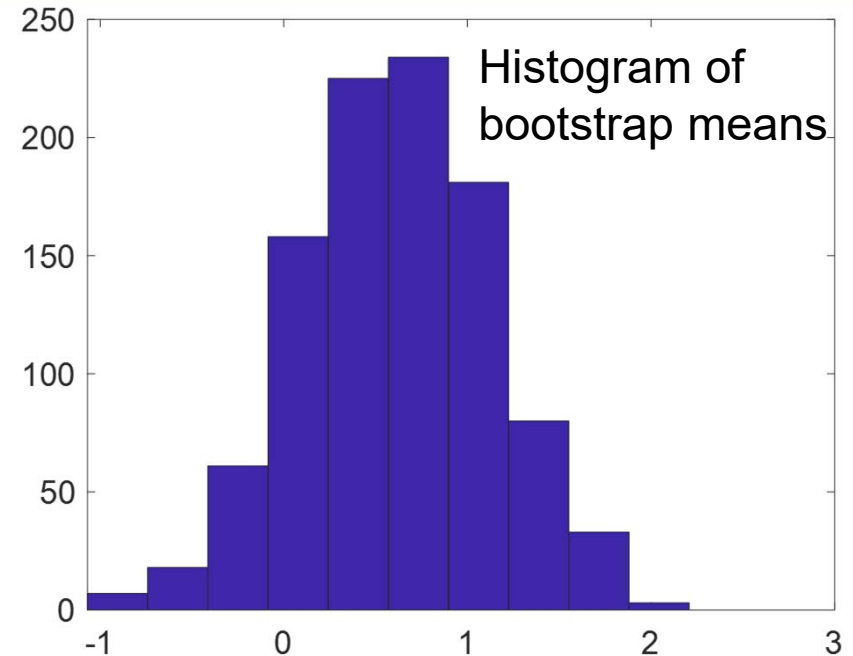
Ex) Statistics for sample mean

- $\text{mean}(x) = 0.6243$
- $\text{std}(x) = 1.7699$
- $\text{mean}(\text{bootstat}) = 0.6068$
- $\text{std}(\text{bootstat}) = 0.5191$

- Standard deviation of sample mean:

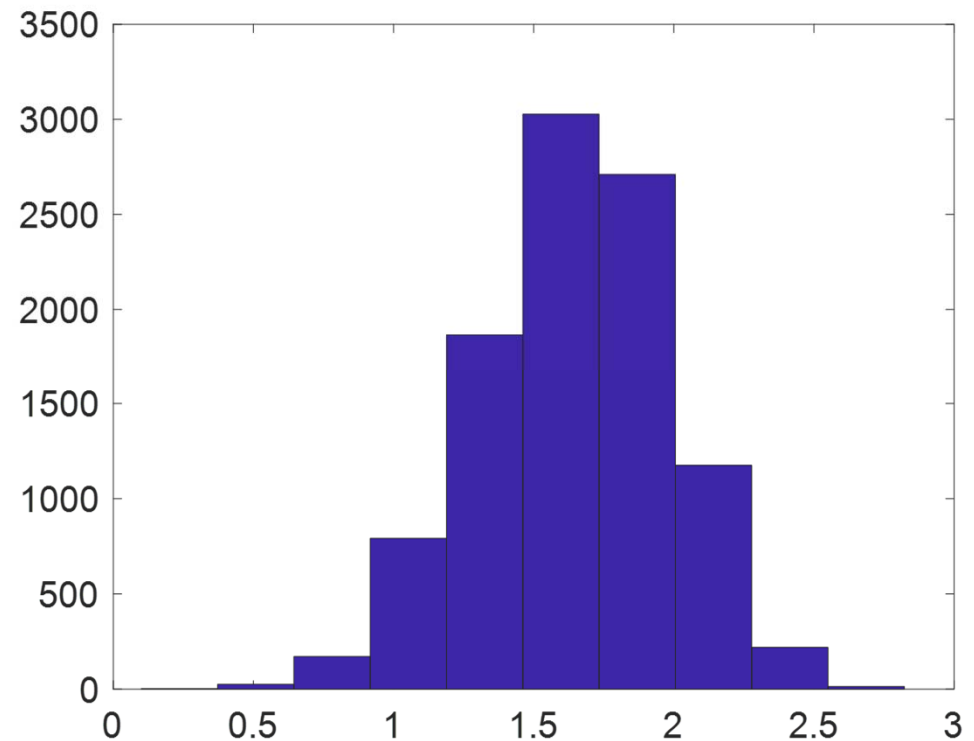
$$\sigma_{\mu} = \frac{\sigma_x}{\sqrt{n}} = 0.56$$

- In other cases we may not have a formula. May use bootstrapping to estimate accuracy of probability



Ex) Statistics of sample standard deviation

- `[bootstat,bootsam]=bootstrp(10000,@std,x);`
- `mean(bootstat)=1.6387`
- `std(bootstat)=0.3415`
- Check ratio
 - `a=randn(10,10000);`
 - `s=std(a);`
 - `mean(s) = 0.9728`
 - `std(s)=0.2302`
- Bootstrap ratio is 0.208, actual ratio 0.237



Exercise

- The variables x and y are normally distributed with $N(0,1^2)$ marginal distributions and a correlation coefficient of 0.7.
 - Generate a sample of 10 pairs and use bootstrap to estimate the accuracy of the correlation coefficient you obtain from the sample.
 - Compare to the accuracy you can get from a formula or by repeating step 1 many times.

Statistical View of Linear Regression



Linear regression

- Statistical model

- Data model:

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

- Prediction y are modeled by a deterministic function of inputs, $f(\mathbf{x}; \mathbf{b})$, which are contaminated by noise or some error defined by ϵ
 - The noise is assumed to be normally distributed with mean zero and variance σ^2

$$y|\mathbf{x} \sim N(f(\mathbf{x}; \mathbf{b}), \sigma^2)$$

- Conditional probability distribution of y given x is a normal distribution with mean function f and variance σ^2

$$p(y|\mathbf{x}) = N(f(\mathbf{x}; \mathbf{b}), \sigma^2)$$

- The principle of likelihood
 - How likely is it that I would have observed the outputs given the inputs?
 - The likelihood of observing the outputs is the conditional probability of making all the observations

$$p(y_1, y_2, \dots, y_n | \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$$

- We assume that the data are independent and identically distributed (iid)
- The joint probability of our measurements takes a factored form

$$p(\mathbf{y} | \mathbf{X}, \mathbf{b}, \sigma) = \prod_{i=1}^n p(y_i | \mathbf{x}_i) = \prod_{i=1}^n N(f(\mathbf{x}_i; \mathbf{b}), \sigma^2)$$

Maximum likelihood

- The principle of likelihood
 - The joint probability is likelihood function
 - Log-likelihood function is defined as

$$\begin{aligned} L = \log p(\mathbf{y}|\mathbf{X}, \mathbf{b}, \sigma) &= \sum_{i=1}^n \log p(y_i|\mathbf{x}_i, \mathbf{b}, \sigma) \\ &= \sum_{i=1}^n \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2} |y_i - f(\mathbf{x}_i; \mathbf{b})|^2\right) \end{aligned}$$

- Maximum likelihood = the stationary point of the log-likelihood

$$\frac{\partial L}{\partial \mathbf{b}} = \frac{1}{\sigma^2} (\mathbf{X}^T \mathbf{y} - \mathbf{X}^T \mathbf{X} \mathbf{b}) = 0$$

- Estimated model parameters
 - Same with estimated model parameters by minimizing RMS error

$$\hat{\mathbf{b}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad \hat{\sigma}^2 = \frac{1}{n} (\mathbf{y} - \mathbf{X}\hat{\mathbf{b}})^T (\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}) = \frac{1}{n} (\mathbf{y}^T \mathbf{y} - \mathbf{y}^T \hat{\mathbf{y}})$$

- Hessian matrix (information matrix) of likelihood function
 - A measure of uncertainty in the estimates

$$\frac{\partial L}{\partial \mathbf{b}} = \frac{1}{\sigma^2} (\mathbf{X}^T \mathbf{y} - \mathbf{X}^T \mathbf{X} \mathbf{b})$$

$$\frac{\partial^2 L}{\partial \mathbf{b} \partial \mathbf{b}^T} = -\frac{1}{\sigma^2} \mathbf{X}^T \mathbf{X}$$

Uncertainty in estimated parameters

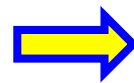
- Uncertainty is defined as covariance matrix

$$\text{cov}[\hat{\mathbf{b}}] = E[\hat{\mathbf{b}}\hat{\mathbf{b}}^T] - E[\hat{\mathbf{b}}]E[\hat{\mathbf{b}}^T]$$

$$E[\hat{\mathbf{b}}\hat{\mathbf{b}}^T] = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^TE[\mathbf{y}\mathbf{y}^T]\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}$$

$$E[\mathbf{y}\mathbf{y}^T] = E[(\mathbf{X}\mathbf{b} - \boldsymbol{\epsilon})(\mathbf{X}\mathbf{b} - \boldsymbol{\epsilon})^T] = \mathbf{X}\mathbf{b}\mathbf{b}^T\mathbf{X}^T + \sigma^2\mathbf{I}$$

$$E[\hat{\mathbf{b}}] = \mathbf{b}$$


$$\text{cov}[\hat{\mathbf{b}}] = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$$

$\hat{\mathbf{b}}$ is uncertain but \mathbf{b} is not.
 $E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] = \sigma^2\mathbf{I}$

- Hessian matrix is defined as the information matrix

$$\text{cov}[\hat{\mathbf{b}}] = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1} = -\left(\frac{\partial^2 L}{\partial \mathbf{b}\partial \mathbf{b}^T}\right)^{-1}$$

Uncertainty in prediction

- Prediction

$$\hat{y}(\mathbf{x}) = \boldsymbol{\xi}(\mathbf{x})^T \mathbf{b} \quad \text{cov}[\hat{\mathbf{b}}] = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}$$

- Estimate of the variance of the prediction

$$\text{Var}(\hat{y}) = \boldsymbol{\xi}(\mathbf{x})^T \text{cov}[\hat{\mathbf{b}}] \boldsymbol{\xi}(\mathbf{x}) = \boldsymbol{\xi}(\mathbf{x})^T \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1} \boldsymbol{\xi}(\mathbf{x})$$

$$\text{Std}(\hat{y}) = \sqrt{\boldsymbol{\xi}(\mathbf{x})^T \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1} \boldsymbol{\xi}(\mathbf{x})}$$

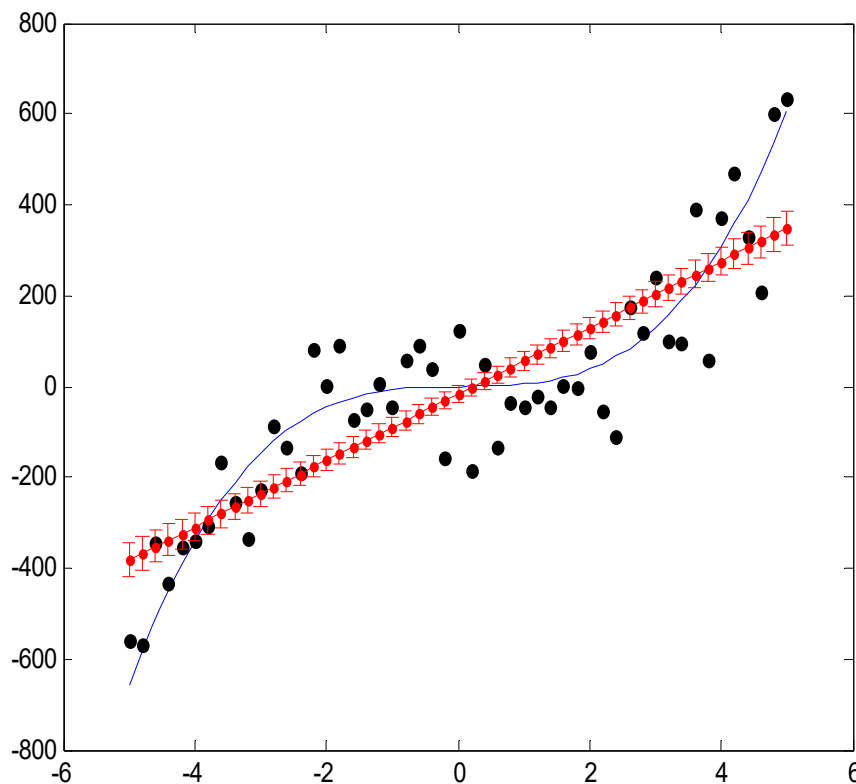
- 95% confidence intervals around the estimate

$$\hat{y} \pm 1.96 \text{Std}(\hat{y})$$

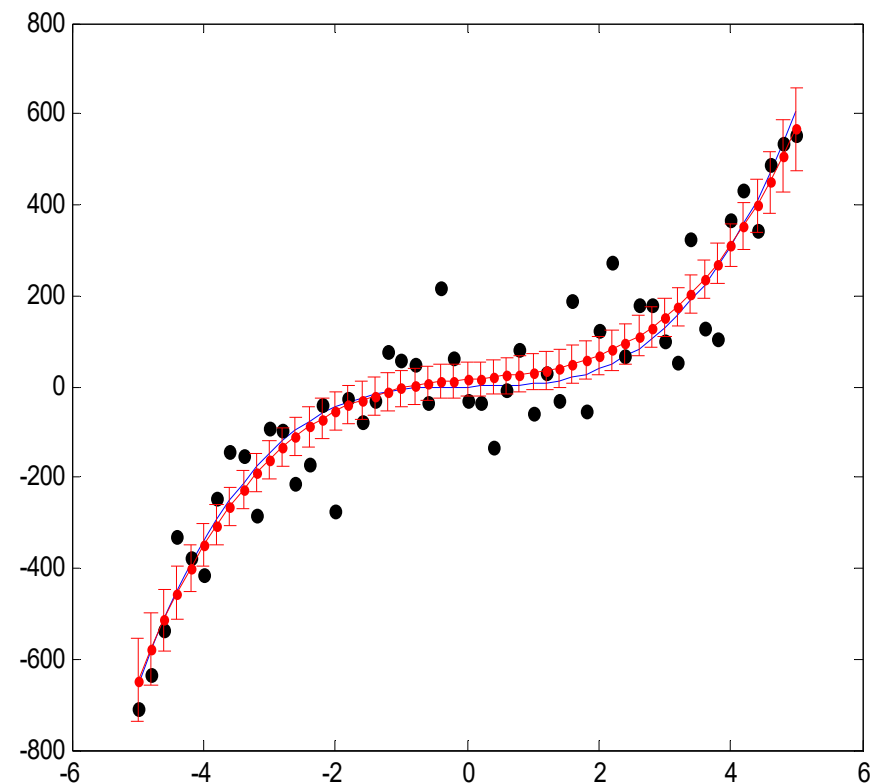
Example

▣ Estimation with different polynomial orders

- True equation: $y = 5x^3 - x^2 + x$ (the standard deviation of noise 100)
- 95% CI of the prediction



(K=1: linear)

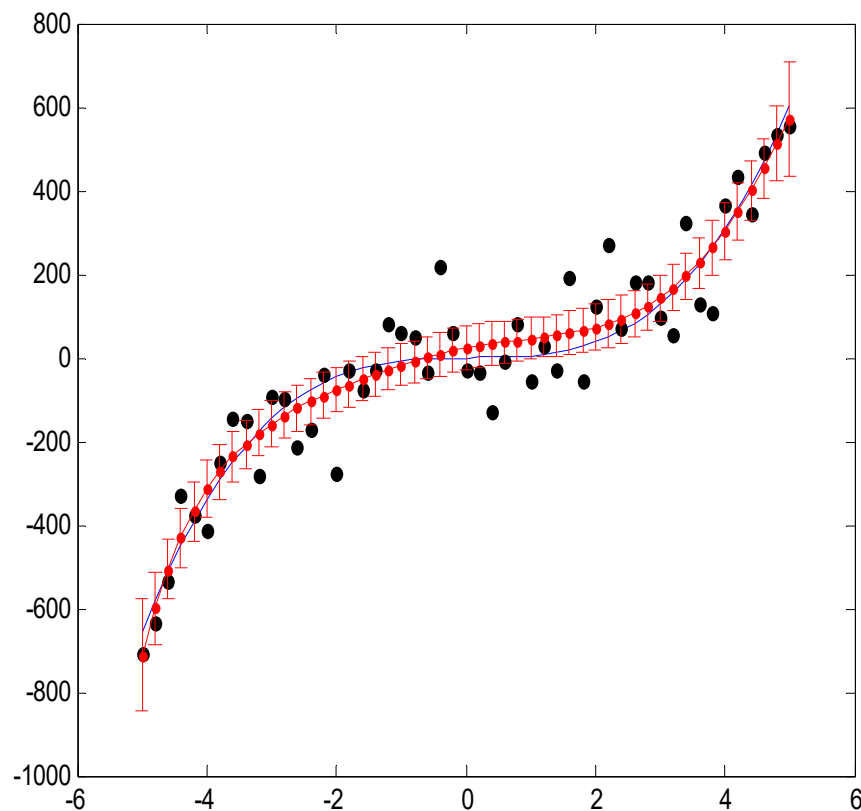


(K=3: cubic)

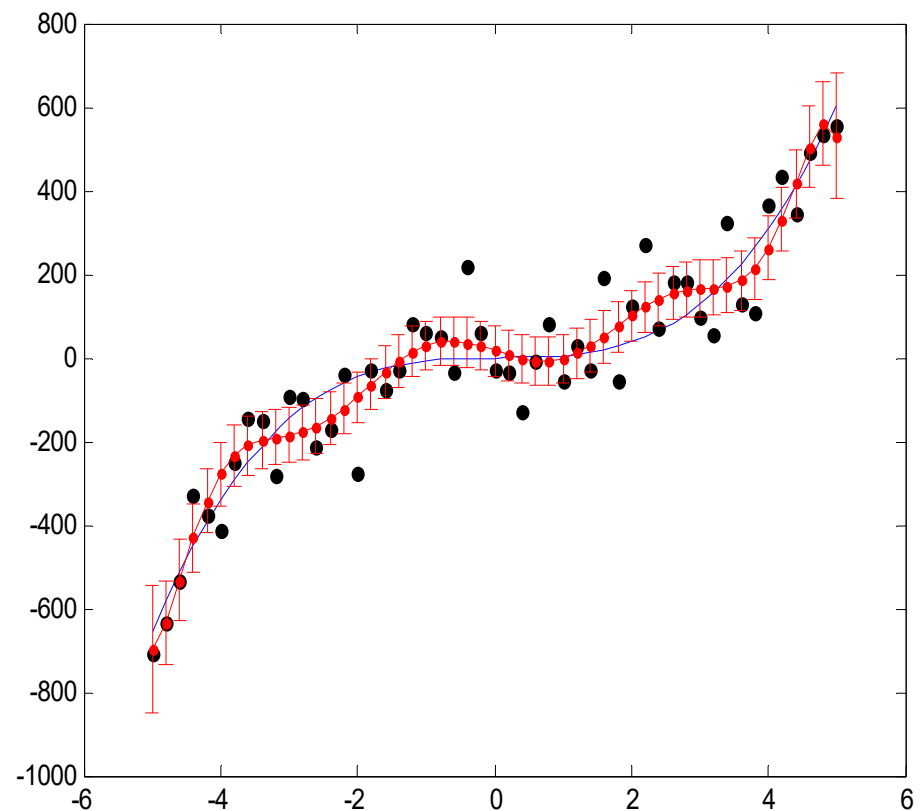
Example

▣ Estimation with different polynomial orders

- True equation: $y = 5x^3 - x^2 + x$ (the standard deviation of noise 100)
- 95% CI of the prediction



(K=6: sixth-order)



(K=10: 10th-order)