SE-598:

Data-Driven Design Methods

Lec. 06: *Surrogate Modeling*

Dr. Pingfeng Wang

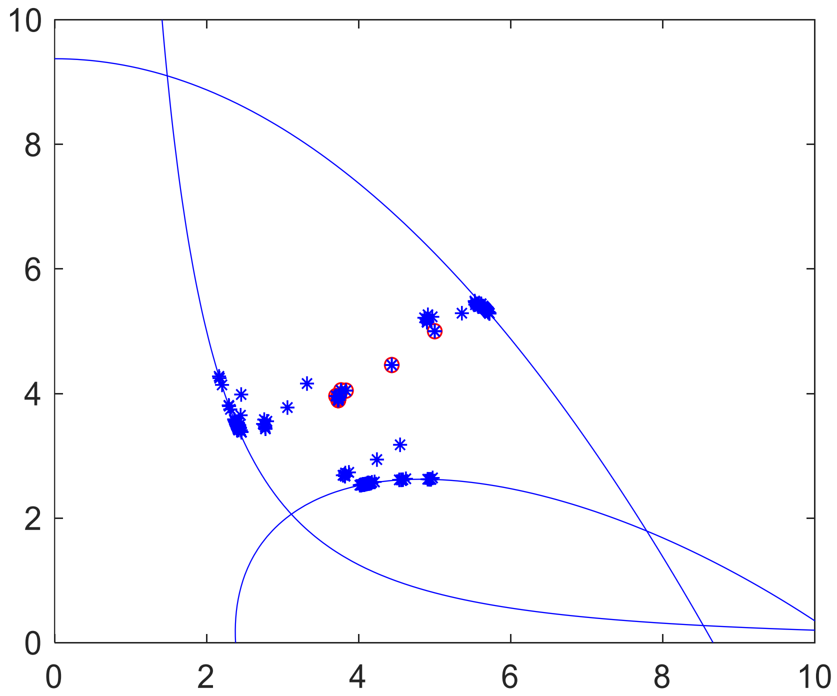
Associate Professor
Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign
Office: TB-310
Email: pingfeng@illinois.edu
Tel: 217-300-7078

Surrogate Modeling - Introduction

Surrogate Modeling - Introduction

- Motivation
- Surrogate Modeling Introduction
  - *Model form and parameters*
  - *Sample Points*
  - *Parameter estimation*
  - *Prediction Error*
  - *Model validation*
- Design with surrogate models

Motivation



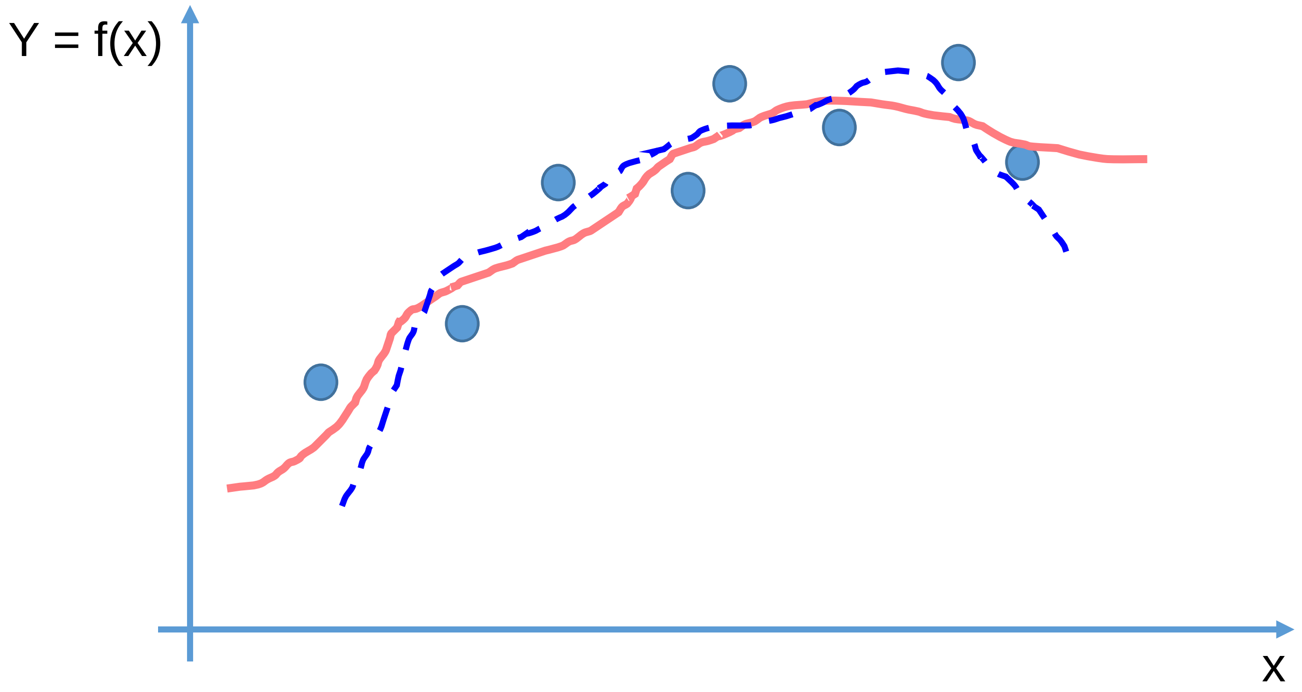

```
%%%% Plot the constraint contours %%%%%%%%%
[xp1, xp2] = meshgrid([0:0.1:10; 0:0.1:10]);

g1=1-xp1.^2.*xp2/20;
g2=1-(xp1+xp2-5).^2/30-(xp1-xp2-12).^2/120;
g3=1-80./(xp1.^2+8*xp2+5);

figure(1)
hold on
contour(xp1, xp2, g1, [0, 0],'-b');
contour(xp1, xp2, g2, [0, 0],'-b');
contour(xp1, xp2, g3, [0, 0],'-b');
```

A total of ***232 sample points*** have been used, as shown by “*” plotted in the figure.

We might be able to use a smaller number of sample points to build a model and approximate the design constraints.

How to develop a good model (surrogate?)

Overview: Surrogate modeling



Construct a metamodel

Stage 1: Preparing the data and choosing the modeling approach

- How to get $y = f(x)$
  - *Sampling methods (e.g. DOE, LHS etc.)*

- What is your f…?

- Simple model: $f(x,w) = w^T x + v$

- The model needs to be "trained" if you have decided the model form. Thus, the model parameters **w** (slope vector) and **v** (intercept) need to be determined

- How can we determine **w** and **v**?

Construct a metamodel (cont.)

Stage 2: Parameter estimation and model training

- Maximum Likelihood Estimation

Given a set of parameters w and $f(\mathrm{x},\mathrm{w})$, we can calculate the probability of the data set {**x**, **y**} having resulted from

$$P = \frac{1}{\left(2\pi\sigma^2\right)^{n/2}} \prod_{i=1}^{n} \left\{ \exp\left[-\frac{1}{2}\right] \left(\frac{y^{(i)} - f(\mathrm{x},\mathrm{w})}{\sigma} \right)^2 \varepsilon \right\}$$

Reverse: what is the likelihood of the parameters, given the data.

$$\min_{\mathrm{w}} \sum_{i=1}^{n} \frac{\left[y^{(i)} - f(\mathrm{x},\mathrm{w})\right]^2}{2\sigma^2} - n \ln \varepsilon$$

$$\min_{\mathrm{w}} \sum_{i=1}^{n} \left[y^{(i)} - f(\mathrm{x},\mathrm{w})\right]^2$$

Construct a metamodel (cont.)

Stage 2: Parameter estimation and model training

- Maximum Likelihood Estimation

Given a set of parameters w and $f(\mathrm{x},\mathrm{w})$, we can calculate the probability of the data set {**x**, **y**} having resulted from

$$P = \frac{1}{\left(2\pi\sigma^2\right)^{n/2}} \prod_{i=1}^{n} \left\{ \exp\left[-\frac{1}{2}\right] \left(\frac{y^{(i)} - f(\mathrm{x},\mathrm{w})}{\sigma} \right)^2 \varepsilon \right\}$$

Reverse: what is the likelihood of the parameters, given the data.

$$\min_{\mathrm{w}} \sum_{i=1}^{n} \frac{\left[y^{(i)} - f(\mathrm{x},\mathrm{w})\right]^2}{2\sigma^2} - n \ln \varepsilon$$

$$\min_{\mathrm{w}} \sum_{i=1}^{n} \left[y^{(i)} - f(\mathrm{x},\mathrm{w})\right]^2$$

Cross Validation

- Split the sample points {**x**, **y**} into roughly *n* subsets
- Remove each one of these subset and fit the model
- Find the difference between the predicted and the actual for the set that we set aside

$$\varepsilon_{CV}\left(\mathrm{w}\right)=\frac{1}{n}\sum_{i=1}^{n}L\left[y^{(i)},\,f^{-\zeta(i)}(\mathrm{x}^{(i)},\mathrm{w})\right]$$

Construct a metamodel (cont.)

Stage 3: Model testing

- Test Points
- Error Metrics:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=0}^{n_t} (y^{(i)} - y^{(i)})^2}{n_t}}$$

RMSE should be as small as possible.

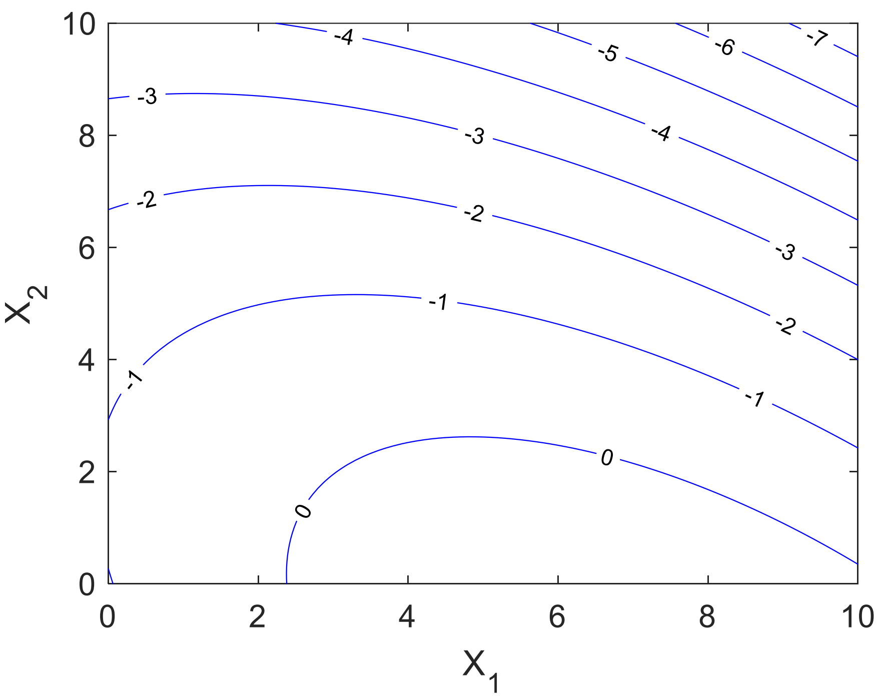
$$\mathrm{R}^2 = 1 - \frac{\sum_{i=0}^{n_t} (y^{(i)} - y^{(i)})^2}{\sum_{i=0}^{n_t} (y^{(i)} - \overline{y})^2} = 1 - \frac{RSS}{TSS}$$

R^2, coefficient of determination, is within [0, 1], and the higher the better.

Example: Polynomial models

$$f\left(x,m,w\right)=w_0+w_1x+w_2x^2+\cdots+w_mx^m=\sum_{i=0}^{m}w_ix^i$$

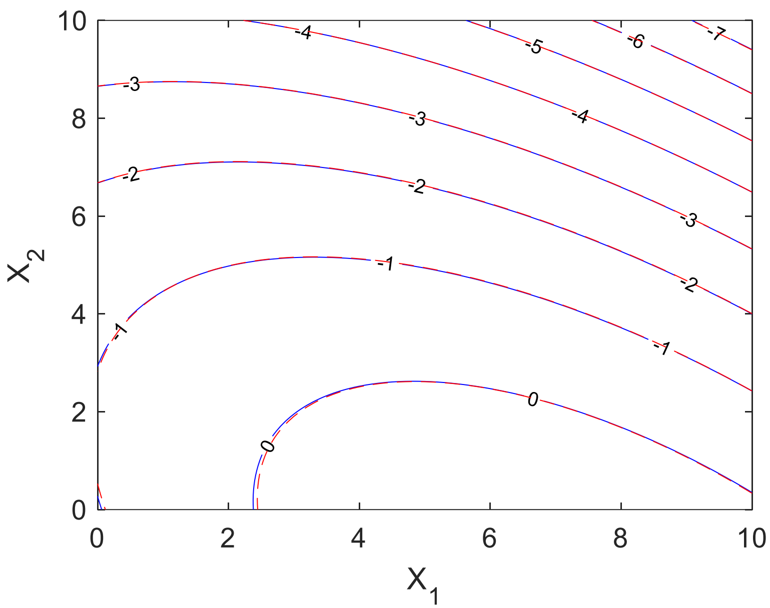
w can be found using least square approach.




```
% we first plot the true constraint contours for G = 0;
g=@(x1, x2) 1-(x1+x2-5).^2/30-(x1-x2-12).^2/120;
[xp1, xp2] = meshgrid([0:0.1:10; 0:0.1:10]);
figure(1)
hold on;
contour(xp1, xp2, g(xp1,xp2),'-b');
```

Can we use 100 sample points to fit each constraint?

Example: Polynomial models




```
% Initial trial with surrogate modeling
close all; clear; clc

% we first plot the true constraint contours for G = 0;
g=@(x1, x2) 1-(x1+x2-5).^2/30-(x1-x2-12).^2/120;
[xp1, xp2] = meshgrid([0:0.1:10; 0:0.1:10]);
figure(1)
hold on;
contour(xp1, xp2, g(xp1,xp2),'-b');

% Step 1:
% model form: 3rd order polynomials
% y = w1+w2*x1 + w3*x2 + w4*x1^2+x5*x2^2+w6*x1*x2
% prepare the data
xsamp = 1:1:10;
[xs1, xs2] = meshgrid(xsamp, xsamp);  % generating sample points
xs=[reshape(xs1, [], 1), reshape(xs2, [],1)];
ys = g(xs(:,1), xs(:,2));

% Step 2: fit the model parameters using least square
fun0 = @(w, x1, x2) w(1)+w(2)*x1+w(3)*x2 +w(4)*x1.^2+ ...
        w(5)*x2.^2 +w(6)*x1.*x2;
fun = @(w) sum((fun0(w, xs(:,1), xs(:,2))-ys).^2);

x0 = ones(1,6);
[w1,~] = fminunc(fun,x0)   %min MSE

% plot contours using predicted models

hold on;
contour(xp1, xp2, fun0(w1,xp1,xp2),'--r');
```