Numerical Optimization 14: Surrogate models

Qiang Zhu

University of Nevada Las Vegas

May 20, 2020

Overview

Surrogate Models

The surrogate models are designed to be smooth and inexpensive to evaluate so that they can be efficiently optimized from the given sampling points. A surrogate model $\hat{f}$ parameterized by θ is designed to mimic the true objective function f. The parameters θ can be adjusted to fit the model based on samples collected from f.

Suppose we have

- m design points: $\{x^1, x^2, \cdots, x^m\}$
- associated function evaluations: $\{y^1, y^2, \cdots, y^m\}$

For a particular set of parameters, the model will predict

$$\hat{y} = \{\hat{f}_\theta(x^1), \hat{f}_\theta(x^2), \cdots, \hat{f}_\theta(x^m)\}$$

In turn, this is a minimization problem

$$\min_\theta = ||y - \hat{y}||$$

Linear Models

A simple surrogate model is the linear model, which has the form

$$\hat{f} = w_0 + \boldsymbol{w}^T x \qquad \theta = \{w_0, \boldsymbol{w}\}$$

For an n-dimensional design space, the linear model has $n+1$ parameters, and thus requires at least $n+1$ samples to fit unambiguously.
Instead of having both w and w_0 as parameters, it is common to construct a single vector of parameters $\theta = [w_0, \boldsymbol{w}]$ and prepend 1 to the vector x to get

$$\hat{f} = \boldsymbol{\theta}^T \boldsymbol{x}$$

Finding an optimal $\boldsymbol{\theta}$ requires solving a linear regression problem:

$$\min_{\theta} ||\boldsymbol{y} - \hat{\boldsymbol{y}}|| \quad \text{or} \quad ||y - \boldsymbol{X}\boldsymbol{\theta}||$$

where $\boldsymbol{X}$ is a design matrix, $[(\boldsymbol{x}^1)^T; \cdots; (\boldsymbol{x}^m)^T]$

Basis Functions

The linear model is a linear combination of the components of $\boldsymbol{x}$:

$$\hat{f}(\boldsymbol{x}) = \theta_1 x_1 + \cdots + \theta_n x_n = \sum_{i=1}^{n} \theta_i x_i = \boldsymbol{\theta}^T \boldsymbol{x}$$

which is a specific example of a more general linear combination of basis functions.

$$\hat{f}(\boldsymbol{x}) = \theta_1 b(x_1) + \cdots + \theta_n b(x_n) = \sum_{i=1}^{n} \theta_i b(x_i) = \boldsymbol{\theta}^T b(\boldsymbol{x})$$

Linear models cannot capture nonlinear relations. There are a variety of other families of basis functions that can represent more expressive surrogate models. The remainder of this section discusses a few common families.

Polynomial Basis Functions

Polynomial basis functions consist of a product of design vector components, each raised to a power. Linear basis functions are a special case of polynomial basis functions.
In one dimension, a polynomial model of degree k has the form

$$\hat{f}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots = \sum_{i=1}^{k} \theta_i x^i$$

In two dimensions, a polynomial model of degree k has basis functions of the form

$$b_{ij}(\boldsymbol{x}) = x_1^j x_2^j \quad \text{for } i, j \in \{0, \cdots, k\}, i + j \leq k$$

Sinusoidal Basis Functions

Any continuous function over a finite domain can be represented using an infinite set of sinusoidal basis functions. A Fourier series can be constructed for any integrable univariate function f on an interval $[a, b]$

$$f(x) = \frac{\theta_0}{2} + \sum_{i=1}^{\infty} \theta_i^{\sin} \sin\left(\frac{2\pi i x}{b-a}\right) \sum_{i=1}^{\infty} \theta_i^{\cos} \cos\left(\frac{2\pi i x}{b-a}\right)$$

where

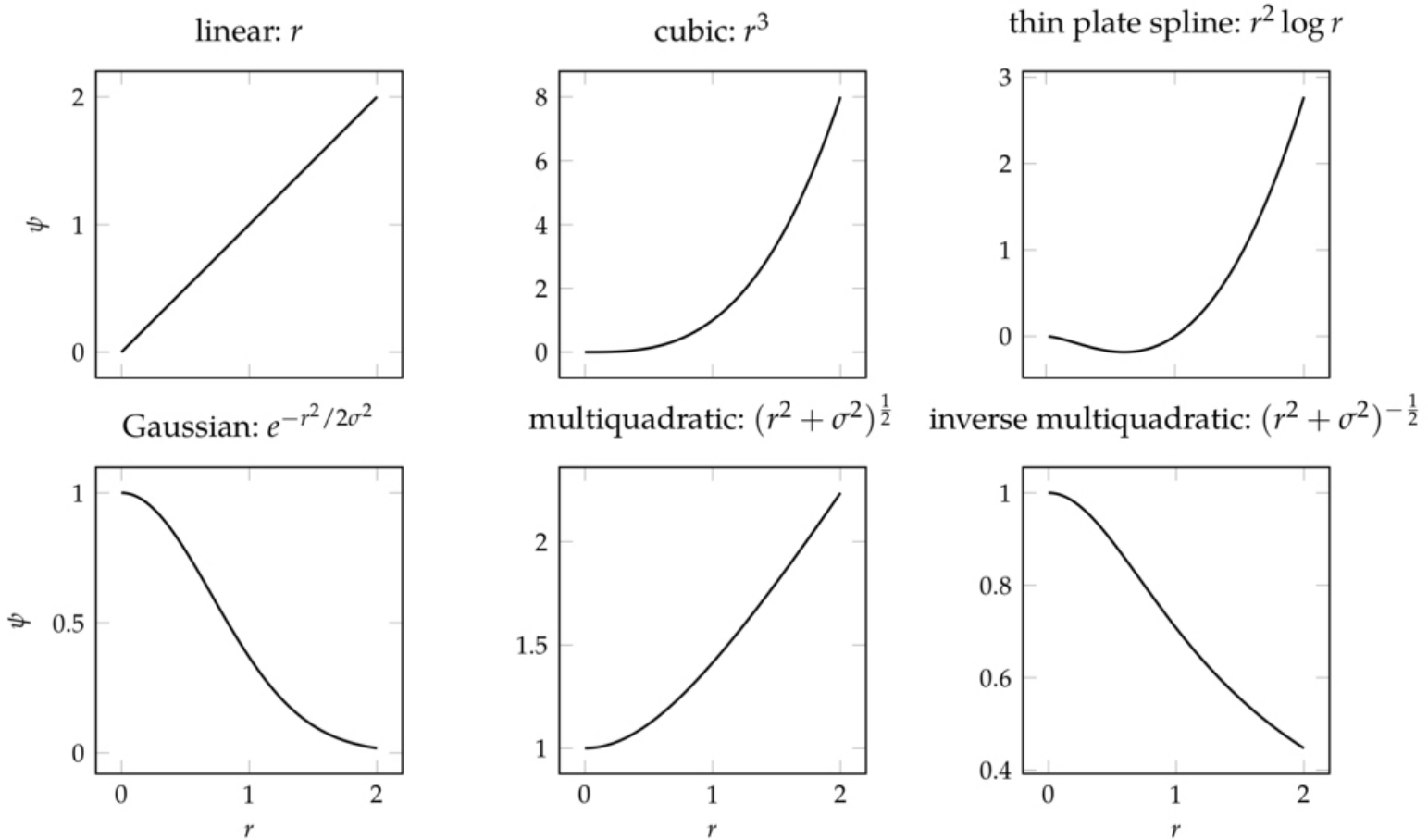
$$\theta_0 = \frac{2}{b-a} \int_a^b f(x) dx$$

$$\theta_i^{\sin} = \frac{2}{b-a} \int_a^b f(x) \sin\left(\frac{2\pi i x}{b-a}\right) dx$$

$$\theta_i^{\cos} = \frac{2}{b-a} \int_a^b f(x) \cos\left(\frac{2\pi i x}{b-a}\right) dx$$

Radial Basis Functions

A radial function Ψ is one which depends only on the distance of a point from some center point c, such that it can be written $\Psi(x, c) = \Psi(|xc|)) = \Psi(r)$.



Fitting Noisy Objective Functions

Models fit using regression will pass as close as possible to every design point. When the objective function evaluations are noisy, complex models are likely to excessively contort themselves to pass through every point. However, smoother fits are often better predictors of the true underlying objective function. A regularization term is added in addition to the prediction error in order to give preference to solutions with lower weights. The resulting basis regression problem with L2 regularization is:

$$\min_{\theta} ||\boldsymbol{y} - \boldsymbol{B\theta}||^2 + \lambda ||\boldsymbol{\theta}||_2^2$$

The optimal parameter vector is given by:

$$\boldsymbol{\theta} = (\boldsymbol{B^T B} + \lambda \mathbf{I}) \boldsymbol{B^T y}$$

where $\mathbf{I}$ is the identity matrix.

Model Selection

So far, we have discussed how to fit a particular model to data. We generally want to minimize generalization error, which is a measure of the error of the model on the full design space, including points that may not be included in the data used to train the model. One way to measure generalization error is to use the expected squared error of its predictions:

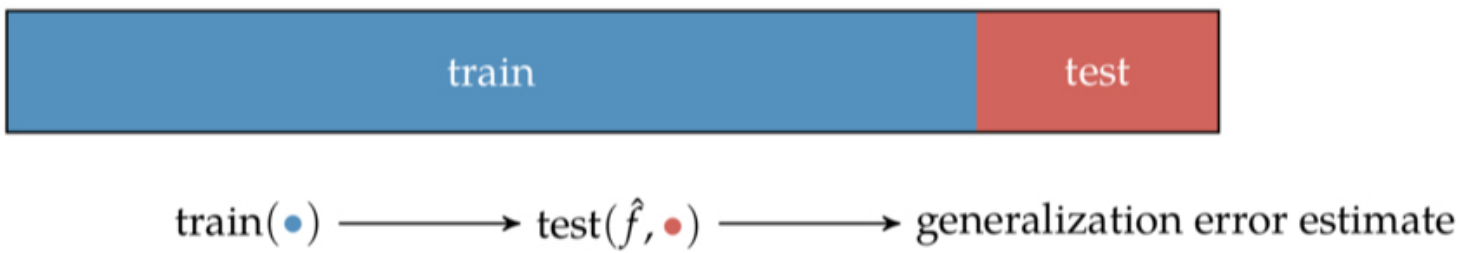
$$\epsilon_{\text{gen}} = \mathbb{E}_{\boldsymbol{x} \sim \boldsymbol{\chi}} \left[\left(f(x) - \hat{f}(x) \right)^2 \right]$$

which impossible to compute. It may be tempting to estimate the generalization error of a model from the training error by using the mean squared error (MSE) of the model evaluated on the m samples:

$$\epsilon_{\text{train}} = \frac{1}{m} \sum_{i}^{m} \left[\left(f(x^i) - \hat{f}(x^i) \right)^2 \right]$$

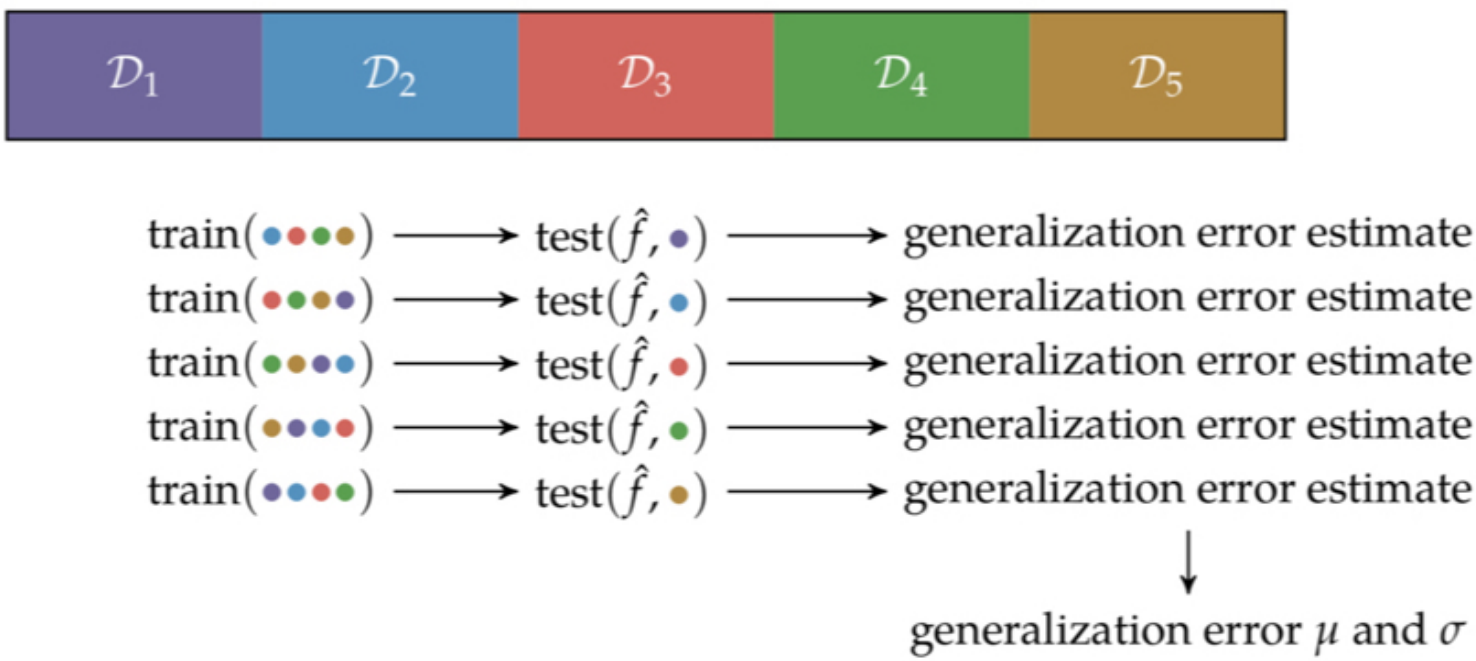
Holdout

A simple approach to estimating the generalization error is the holdout method, which partitions the available data into a test set D_h with h samples and a training set Dt consisting of all remaining $m - h$ samples. The training set is used to fit model parameters. The held out test set is not used during model fitting, and can thus be used to estimate the generalization error. Different split ratios are used, typically ranging from 50% train, 50% test to 90% train, 10% test, depending on the size and nature of the dataset. Using too few samples for training can result in poor fits, whereas using too many will result in poor generalization estimates.



Cross validation

Here, the original dataset D is randomly partitioned into k sets $D_1, D_2, \cdots, D_k$ of equal, or approximately equal, size. We then train k models, one on each subset of $k-1$ sets, and we use the withheld set to estimate the generalization error. The cross-validation estimate of generalization error is the mean generalization error over all folds



Summary

- Surrogate models are function approximations that can be optimized instead of the true, potentially expensive objective function.
- Many surrogate models can be represented using a linear combination of basis functions.
- Model selection involves a bias-variance trade off between models with low complexity that cannot capture important trends and models with high complexity that overfit to noise.
- Generalization error can be estimated using techniques such as hold out,k-fold cross validation, and the bootstrap.