Surrogate Modeling

1 Introduction

Capturing uncertainty in high-fidelity computational models can be costly on many domains or even infeasible. Here we follow the uncertainty quantification literature in building so called *surrogate models*, which are designed to approximate the input-output relation of high-fidelity models.

In the following we start by describing the data we are working with and then present the different kind of models we are using as surrogates.

2 Data

Say we want to analyze a model $\mathcal{M}$ that is parameterized with a parameter vector $\theta \in \Theta \subset \mathbb{R}^D$. Given a specific research question, we are interested in a *quantity of interest* (qoi), that we can compute from the model. We can write this relationship in a functional form, $qoi = \tilde{\mathcal{M}}(\theta)$ –here we use a similar notation as for the model since the functional form, i.e. the mathematical relationship between the quantity of interest and the parameters can be derived from the model, but it does not characterize the model in general. For notational convenience we will write $y = \tilde{\mathcal{M}}(\theta)$ hereinafter.

Further we assume that we obtained (maximum-likelihood) estimates $\hat{\theta}$ for θ, with corresponding (approximate) standard errors. We then sample θ_n from the multivariate normal distribution defined by the mean vector $\hat{\theta}$ and standard deviation given by the estimated standard errors, for $n = 1, \dots, N^*$ iterations —this corresponds to the asymptotic distribution of the maximum-likelihood estimator. For each θ_n we compute the quantity of interest, y_n, using the high-fidelity model.

Our general goal is to analyze how the uncertainty in the estimates, quantified by the standard errors, propagates through the model into the uncertainty of the quantity of interest. That is, we want to analyze the functional relationship $\theta \xrightarrow{f} y$ using data $\mathcal{D} := \{(y_n, \theta_n) : n = 1, \dots, N\}$.

2.1 Data Preprocessing and Model Validation

Data Preprocessing. The input data we use is drawn from a multivariate normal distribution. For all models in question we therefore first standardize the input data, i.e. we subtract mean and divide by the empirical standard deviation. This has multiple advantages. For input dimensions with little variation we ensure that we do not run into problems regarding numerical singularity. Further, in all linear models this makes coefficient values comparable.

Model Validation. In the surrogate modeling framework we are only concerned with obtaining the best possible prediction. To ensure that we are not picking up noise from the data we split the original data set into a training and testing (validation) data set. For fitting of the surrogate model parameters we only use the training data set and for validation we only use the testing data set. We measure the goodness of fit of our models using the mean absolute prediction error on the validation set.

3 Surrogate Models

For notational convenience we write $\mathbf{\Omega} = (\theta_1^\top, \ldots, \theta_N^\top)^\top \in \mathbb{R}^{N \times D}$, where the rows of $\mathbf{\Omega}$ denote the different draws from the asymptotic distribution used for training and the columns the different dimensions/ parameters of the model. Further, we write $\mathbf{y} = (y_1, \ldots, y_N)^\top \in \mathbb{R}^N$.

3.1 Linear Regression

Model:

$$\begin{aligned} y_n &= \beta_0 + \sum_{d=1}^{D} \beta_d \, \theta_n^{(d)} + \epsilon_n && (1) \\ &= \beta_0 + \beta^\top \theta_n + \epsilon_n && (2) \end{aligned}$$

Estimation:

$$\beta_0^*, \beta^* = \underset{b_0 \in \mathbb{R}, \boldsymbol{b} \in \mathbb{R}^D}{\operatorname{argmin}} \left\{ \|y - b_0 - \mathbf{\Omega}\boldsymbol{b}\|_2^2 \right\} \tag{3}$$

Prediction:

$$y_{test} = \beta_0^* + \beta^* \theta_{test} \tag{4}$$

3.1.1 Results Linear Regression

Here we report the results of for different training data sizes etc.

3.2 Polynomial Regression (2nd degree)

Model:

$$y_n = \beta_0 + \sum_{d=1}^{D} \beta_{1,d} \, \theta_n^{(d)} + \sum_{d=1}^{D} \beta_{2,d} \, (\theta_n^{(d)})^2 + \sum_{d \neq j} \beta_{3,dj} \, \theta_n^{(d)} \theta_n^{(j)} + \epsilon_n \tag{5}$$

Estimation: Write $\phi(\theta)$ for the vector containing linear terms, squared terms and interaction terms. Then, $y_n = \beta_0 + \beta \, \phi(\theta_n) + \epsilon_n$ and

$$\beta_0^*, \beta^* = \underset{b_0 \in \mathbb{R}, \boldsymbol{b} \in \mathbb{R}^{(2D + \binom{D}{2})}}{\operatorname{argmin}} \left\{ \|y - b_0 - \phi(\mathbf{\Omega})\boldsymbol{b}\|_2^2 \right\} \tag{6}$$

Prediction:

$$y_{test} = \beta_0^* + \beta^* \phi(\theta_{test}) \tag{7}$$

3.3 Ridge Regression (Tikhonov regularization)

Idea: Regularize coefficients to avoid overfitting and increase prediction accuracy on test data.

Model: Same as equation 1 or equation 5, depending on the degree of the polynomial used.

Estimation: Say we use a polynomial model of degree 1, then

$$\beta_0^*(\alpha), \beta^*(\alpha) = \underset{b_0 \in \mathbb{R}, \boldsymbol{b} \in \mathbb{R}^D}{\text{argmin}} \left\{ \|y - b_0 - \Omega \boldsymbol{b}\|_2^2 + \alpha \|\boldsymbol{b}\|_2^2 \right\} , \tag{8}$$

and

$$\beta_0^*, \beta^* = \beta_0^*(\alpha^*), \beta^*(\alpha^*) , \tag{9}$$

where α^* is chosen via cross validation on the training set.

Prediction: Same as equation 4 or equation 7, depending on the degree of the polynomial used, but using the regularized coefficients.

3.4 Neural Networks

Work in progress.

3.5 Boosted Trees

Work in progress.

3.6 Gaussian Processes

Work in progress.

4 Results

Here we compare the performance of the different surrogate models along two different margins that might be important to the applied researcher: the number of observations in the training data N_T, i.e. the number of draws from the asymptotic distribution, as this can be computationally intensive and along the number of parameters that are included in the prediction.

4.1 Varying N_T

4.2 Variable selection