Chapter 1 - Introduction

Weber Jakob

October 22, 2020

Contents

1 Introduction

The ongoing digitalization of complex, large-scale industrial plants is leading to a massive increase of process data. This data can be used to enhance the overall understanding of the characterizing physical process inside the plant. Modern observer and control concepts are used to enhance the efficiency and quality the plant. They use mathematical models of the ongoing process. Using an exact physical description as mathematical model for the relevant quantities is nevertheless often not feasible because of the complexity of the process as well as computational and measurement limitations.

Data driven approaches are state-of-the-art in many fields, including e.g. image and speech recognition. The usage of data driven methods, e.g. artificial neural networks, parametric models, etc., to model process quantities for which measurements are expensive or not practical, gains more and more influence and acceptance in the field of process control and optimization. The application of the specific algorithm depends on issues like data quality, interpretability of the model or computational efficiency.

In most process optimization tasks massive amounts of domain specific knowledge in form of physical theories and a priori knowledge is available. The combination of the use of domain knowledge and data driven modeling techniques is called hybrid modeling or grey-box modeling. It lies between the two modeling extrema of white-box models, which are derived from first principles and physical models, [white box source] and black-box models, which are derived from data only.**ashby1961introduction** The incorporation of this knowledge in state-of-the-art data driven approaches is not trivial and not solved for

some algorithms. Nevertheless, its inclusion should improve the interpretability, which itself is of importance in the context of the emerging field of explainable artificial intelligence (XAI). XAI refers to modeling approaches and techniques in which the main goal is that the resulting model is understandable by humans. **dovsilovic2018explainable**

In this thesis, we are going to develop an algorithm for efficient, static, multi-dimensional function approximation using a priori domain knowledge. The algorithm is based on structured additive regression using splines. **fahrmeir2013regression**. We use user-defined constraints to include the a priori domain knowledge in the fitting process. **hofner2011monotonicity** It should produce interpretable and efficient models based on the given domain knowledge. The incorporation of domain specific knowledge should improve the model quality and robustness as well as interpolation and extrapolation behavior in situations where the measured data is sparse and/or noisy. Further, we are going to evaluate the algorithm using noisy samples from artificial test functions with known behavior as well as real-world data collected in a heat treatment process.

2 Related Work

We will now discuss some of the most used data-driven algorithms. The discussion includes the following model approaches:

- Parametric models: Linear and polynomial regression
- Non-parametric models: Basis function models
- Artificial neural networks
- Look-up tables

and focuses on the interpretability, computational efficiency and the ability to include domain knowledge of the individual modeling approaches. The list given above is not intended to be complete.

The common starting point for the different data-driven modeling approaches is that we have some data $\{x_1^{(i)}, \ldots, x_q^{(i)}, y^{(i)}\}$ for $i = 1, \ldots, n$. We restrict the following discussion to the single-input setting, i.e. $\{x^{(i)}, y^{(i)}\}$ for $i = 1, \ldots, n$. The generalization to multiple input dimensions is given in the respective literature. The model function f is then used to predict the response or output variable y, i.e.

$$y = f(x)). \tag{1}$$

Therefore we are in the setting of supervised learning.

2.1 Parametric Models

According to Nelles, parametric models are defined as models that can describe the true process behavior using a finite number of parameters.**nelles2013nonlinear** An example is given by the linear regression model for one input variable x as

$$y = f(x) = \beta_0 + \beta_1 x. \tag{2}$$

Both parameters β_0 and β_1 allow for a direct interpretation as β_0 is the intercept, i.e. the output for the input $x = 0$, and β_1 is the slope, i.e. the constant defining the relationship between the increase of the output y with respect to the increase of the input x. The interpretability of linear regression models is therefore very high.

Linear regression models are widely used and part of standard software tools. Their parameters can be efficiently computed using the least squares algorithm. One major drawback is that they can only recover linear relationships between input and output variables. They are therefore quite restrictive and do not allow the incorporation of a priori domain knowledge.

An extension of the linear regression model is given by polynomial regression. Here, we try to model the output data y using a polynomial of degree p, i.e.

$$y = f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_p x^p. \tag{3}$$

Polynomial regression introduces more flexibility in the fitting process, since the restriction of linear relationship is relaxed to a polynomial relationship of degree p. As for linear models, the interpretability of the parameters is given. We can use the least squares algorithm for parameter estimation. The incorporation of a some priori domain knowledge is possible, e.g. as the degree of the polynomial regression model. One major problem of polynomial regression is that the model function becomes quite wiggly for high polynomial degrees p.

Linear and polynomial regression models are so-called global models. Their parameters act on the complete input space. This property makes the incorporation of specific a priori domain knowledge, e.g. unimodal behavior, difficult and in most cases nearly impossible for parametric models.

2.2 Non-parametric Models

Nelles defines non-parametric models as models which require a infinite number of parameters to describe a process exactly. **nelles2013nonlinear** In almost all practical applications this infinite series is approximated by a finite number of parameters using the basis function approach given by

$$y = f(x) = \sum_{i=1}^{M} \theta_i^{(l)} \Phi_i(x, \theta_i^{(nl)}) \tag{4}$$

with the linear parameters $\theta_i^{(l)}$, the basis functions $\Phi_i(.)$, the input variables x and the non-linear parameters $\theta_i^{(nl)}$. The output y is therefore given by a linear combination of M basis functions $\Phi_i(.)$. To model a non-linear relationship between y and x, the basis functions $\Phi(.)$ need to be non-linear. Commonly used basis functions are e.g. the *hat function*, the *Gaussian*, *splines* or the *hinge function*.

One commonly used algorithm using the basis function approach is called Multivariate Adaptive Regression Splines (MARS). **friedman1991multivariate**

MARS approximates data, as example again for a single input dimension, using the following model

$$y = \sum_{i=1}^{M} \theta_i \Phi_i(x) \tag{5}$$

using constant parameters θ_i. The basis functions $\Phi_i(.)$ are one of the following three alternatives:

1. $\Phi_i(x) = 1$, representing the intercept.
2. $\Phi_i(x) = \max(0, x - \text{constant}_i)$ or $= \max(0, \text{constant}_i - x)$, representing the *hinge function* h_i.
3. $\Phi_i(x) = h_i h_j$, representing a product of two *hinge functions*.

MARS fits the model using an recursive splitting approach. More information can be found in **friedman1991multivariate** and **friedman2001elements**. MARS models are more flexible compared to the parametric linear and polynomial regression models. As only hinge functions and products of hinge functions are used, MARS models are efficient and in general simple to understand and interpret. To our knowledge, there is currently no possibility to include a priori domain knowledge in the fitting process when using MARS.

Another widely used methods using basis functions is the use of *splines*. Splines are defined as piece-wise polynomials on a sequence of knots. Further information can be found in *Chapter Splines*. **deBoor1978practicalGuideToSplines** Using k splines to model some data, we obtain the following model formulation

$$y = f(x) = \beta_i \sum_{i=1}^{k} b_i(x) \tag{6}$$

with the spline basis functions b_i and the parameters β_i. The parameters can be calculated used the least squares algorithm. The usage of splines allows a lot of flexibility and computational efficiency. A priori domain knowledge can be incorporated using the approach given by Hofner.**hofner2011monotonicity**

The basis function approach in (4) may be extended by changing the parameters $\theta_i^{(l)}$ to more complex forms. An example for this is the so-called local linear neuro-fuzzy model, for which each parameter $\theta_i^{(i)}$ is changed to be a *local linear model* and each basis function $\Phi_i(.)$ is then called *validity function* determining the region of validity of the local linear model. **nelles2013nonlinear** The validity functions are normalized for any model input x, i.e.

$$\sum_{i=1}^{M} \Phi_i(x) = 1. \tag{7}$$

and typically chosen to be *Gaussian* functions, i.e.

$$\Phi_i(x) = a_i \exp\big(\frac{(x-\mu_i)^2}{\sigma_i^2}\big) \tag{8}$$

with the normalization constant a_i and the parameters μ_i and σ_i determining the location and scale of the Gaussian function. The output of the local linear neuro-fuzzy model using M local linear models is then given by

$$y = \sum_{i=1}^{M} (\beta_{i0} + \beta_{i1} x_1) \Phi_i(x). \tag{9}$$

The first term in the summation are the *local linear models*. The parameters β_{ij} for $i = 1, \ldots, M$ and $j = 0, 1$ as well as the parameters μ_i and σ_i from the validity functions Φ_i need to be optimized. This is done using the LOLIMOT algorithm. Further information is given in **nelles2013nonlinear**.

Local linear models as extension of linear models possess more flexibility with regards to non-linear relationships in the data. They can also be efficiently evaluated after the iterative training process. The interpretability is high since each local linear model contributes to the prediction according to its validity function. The ability to include a priori domain knowledge in the fitting process is currently not available.

2.3 Artificial Neural Networks

Artificial neural networks are currently the state-of-the-art solution method for many problems ranging from computer vision over time-series prediction to regression tasks. They are constructed as coarse model of the human brain, consisting of neurons which are connected by some weights. These connections are adapted in the learning process using an algorithm called "backpropagation". They utilize a high number of parameters to model hidden, high-dimensional relationships in the data. Further information can be found in standard textbook about neural networks, e.g. **bishop2006patternRecognition** or **goodfellow2016deep**.

In terms of modeling flexibility, artificial neural networks of sufficient size are proven to be able to represent a wide variety of functions by so-called universal approximation theorems.**cybenko1989approximation hornik1991approximation** The computational complexity of a neural network depends on its size, aka. the number of parameters. Large networks need many training samples to generate sufficiently accurate predictions. Artificial neural networks are an example of a black-box model. The inclusion of a priori domain knowledge into the learning process of neural networks is possible for specific types of knowledge using the concepts of hints, see **abu1990learning sill1997monotonicity**

2.4 Look-up Tables

A look-up table is an array of values, which allows to replace computational expensive computations with inexpensive array indexing operations. The values in the look-up table are most often computed and stored beforehand. To gain

higher resolution, interpolation techniques such as linear or quadratic interpolation may be applied to look-up tables.

Look-up tables are a standard tool in many fields. They are extremely efficient in terms of computation time. One problem that occurs is the exponential increase in size with the number of dimensions for the look-up table. As example, a 2×2-table needs to save 4 values, while a $2 \times 2 \times 2$ table already needs 8 values. Another problem is that the values in the look-up table may come from complex, computational or physical models.

Lattice regression tackles this problems by jointly estimating all lookup-table values by minimizing the regularized interpolation error on training data. **garcia2009lattice** They state that using ensembles of lookup-tables which combine several *tiny* lattices enables linear scaling in the number of input dimension even for high dimensions. **fard2016fast** They further state that lattice regression may be used to incorporate a priori domain knowledge like monotonicity, shape or unimodality into the fitting process. **gupta2016monotonic** **you2017deep**

3 Outline

The thesis is divided into 5 chapters: Chapter 2 provides an overview of the fundamental mathematical concepts used. We focus on the description of linear models, splines and the topic of structured additive regression. In chapter 3, we develop the algorithm using the concepts given in chapter 2. In chapter 4, we test the algorithm using different artificial functions and a priori domain knowledge as well as real-world data. Chapter 5 gives a summary and outlines future, possible work.

4 Stages in a thesis introduction

1. state the general topic and give some background
2. provide a review of the literature related to the topic
3. define the terms and scope of the topic
4. outline the current situation
5. evaluate the current situation (advantages/disadvantages) and identify the gap
6. state research question of proposed research
7. state research aims and/or research objectives
8. state the hypotheses
9. outline the order of information in the thesis
10. outline the methodology

This is the introduction chapter. It is going to contain the following points:

- Background Motivation: Like in the expose, data-driven approaches are necessary because of complexe phenomena
- Interpretable AI: Shift from black-box models to interpretable models
- Related work in XAI
- Goal of Thesis
- outline

References

abu1990learning Yaser S Abu-Mostafa. "Learning from hints in neural networks". In: *Journal of complexity* 6.2 (1990), pp. 192–198.

ashby1961introduction W Ross Ashby. *An introduction to cybernetics*. Chapman & Hall Ltd, 1961.

bishop2006patternRecognition Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

cybenko1989approximation George Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314.

deBoor1978practicalGuideToSplines Carl De Boor et al. *A practical guide to splines*. Vol. 27. springer-verlag New York, 1978.

dovsilovic2018explainable Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. "Explainable artificial intelligence: A survey". In: *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*. IEEE. 2018, pp. 0210–0215.

fahrmeir2013regression Ludwig Fahrmeir et al. *Regression models*. Springer, 2013, pp. 21–72.

fard2016fast Mahdi Milani Fard et al. "Fast and flexible monotonic functions with ensembles of lattices". In: *Advances in Neural Information Processing Systems*. 2016, pp. 2919–2927.

friedman1991multivariate Jerome H Friedman. "Multivariate adaptive regression splines". In: *The annals of statistics* (1991), pp. 1–67.

friedman2001elements Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, 2001.

garcia2009lattice Eric Garcia and Maya Gupta. "Lattice regression". In: *Advances in Neural Information Processing Systems*. 2009, pp. 594–602.

goodfellow2016deep Ian Goodfellow et al. *Deep learning*. Vol. 1. MIT press Cambridge, 2016.

gupta2016monotonic Maya Gupta et al. "Monotonic calibrated interpolated look-up tables". In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 3790–3836.

hofner2011monotonicity Benjamin Hofner, Jörg Müller, and Torsten Hothorn. "Monotonicity-constrained species distribution models". In: *Ecology* 92.10 (2011), pp. 1895–1901.

hornik1991approximation Kurt Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural networks* 4.2 (1991), pp. 251–257.

nelles2013nonlinear Oliver Nelles. *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*. Springer Science & Business Media, 2013.

sill1997monotonicity Joseph Sill and Yaser S Abu-Mostafa. "Monotonicity hints". In: *Advances in neural information processing systems*. 1997, pp. 634–640.

you2017deep Seungil You et al. "Deep lattice networks and partial monotonic functions". In: *Advances in neural information processing systems*. 2017, pp. 2981–2989.