Power Consumption Forecast using Nonlinear Regression Methodologies

Daniel Jaroszewski[1], Benedikt Sturm[2]

[1,2] *FCE Frankfurt Consulting Engineers GmbH, Frankfurt am Main, Hessen, 60549, Germany*
daniel.jaroszewski@frankfurtconsultingengineers.de
benedikt.sturm@frankfurtconsultingengineers.de

Abstract

In this paper, we present an innovative method to forecast time series. We focus on a long-term forecasting model dealing with main periodicities in addition to short term effects. The consolidation of Fourier transformations, which covers the basic oscillation of a time series, with machine learning algorithms approximating the error term, is at the heart of our forecasting model. Later on, we compare different regressions such as kernel and logistic regression, a combinatorial technique on sparse grids and finally a special representative of neural networks. Thereby we are able to forecast power consumption in certain locations of a given network and we show the results of those forecasts as functions of various inputs. The results presented are used for power demand planning of cities and are consequently prognostic in nature. In the context of Health Management, however, one usually works with anomaly detection and supervised learning methods. Nevertheless, a time series forecast in neighboring applications, e.g. the power consumption of a traction system in railway vehicles, could substantially benefit from these prognostic functionalities. This also means that deviations of physical quantities measured under real-time conditions from their expected behavior indicate a likely prevailing malfunction.

1. Introduction

In the course of smart grid activities, the power demand forecast becomes a basic ingredient for further optimization steps. Only if the future can be predicted well enough, the consequences such as the investment in battery/storage solutions and network effects depending on price policies will become solvable. This is the main motivation for power demand forecasts on any voltage level. We focus our activities on the distribution grid and are confronted with grid nodes, that are producing energy, consuming energy and a combination of both. Particularly, weather conditions have a strong impact on power production. In this paper we will present our data driven response surface solution.

2. Pre-processing

In this section we start with the analysis of a multidimensional observable input data set $\mathbf{X}$, representing a sample of the N-dimensional stochastic process $X = (X^1, X^2, ..., X^N)$. $\mathbf{X}$ consists of N signals and T records $(\vec{x}_i)_{i=1,...,T}$, such that $\mathbf{X} \in R^{T\times N}$ can be written as

$$\mathbf{X} = \begin{pmatrix} x_1^1 & x_1^2 & \cdots & x_1^N \\ x_2^1 & x_2^2 & \cdots & x_2^N \\ \vdots & \vdots & \ddots & \vdots \\ x_T^1 & x_T^2 & \cdots & x_T^N \end{pmatrix} = \begin{pmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vdots \\ \vec{x}_T \end{pmatrix} \tag{1}$$

On the other hand there is an output process $\mathbf{Y} \in R^{T\times K}$, which captures the measurements of the power consumption/ generation of a certain grid node. It is conceivable to distinguish signal characteristics such as idle power and the different phases. Here, our focus is on the prediction of the aggregated power signal. The K-dimensional stochastic process is defined as follows

$$\mathbf{Y} = \begin{pmatrix} y_1^1 & y_1^2 & \cdots & y_1^K \\ y_2^1 & y_2^2 & \cdots & y_2^K \\ \vdots & \vdots & \ddots & \vdots \\ y_T^1 & y_T^2 & \cdots & y_T^K \end{pmatrix} \tag{2}$$

Industrial data usually call for some pre-processing steps, which will be described below.

2.1. Comparability

This section is concerned with some primary steps applicable to huge data sets. The signal ranges vary from parameter to parameter, thus a data point standardization is required. Furthermore, high sampling frequency over a long-time period results in many records and this impacts the performance of all statistical analysis steps. Additionally, we must cope with problems such as measurement errors and signals carrying no information. To facilitate comparability, we will use normalized or standardized signals as inputs for all analysis methods.

Normalization

A normalization maps the entire signal range into the interval between zero and one and is defined as

$$x_i^j = \frac{x_i^j - \min(X^j)}{\max(X^j) - \min(X^j)} \tag{3}$$

where i denotes the i-th record and $\max(X^j)$ or $\min(X^j)$ are, respectively, the minimum and maximum of signal X^j of our training matrix.

Standardization

A standardization transforms a random variable such that the transformed variable has a mean of zero and variance of one

$$x_i^j = \frac{x_i^j - \mu^j}{\sigma^j} \tag{4}$$

2.2. Outlier Detection and Date time synchronization

In real life one cannot exclude certain errors in the data set. Within the progress and the extraction of data, measurement errors are possible. Moreover, in the case of different measurement sources a date time synchronization is indispensable.

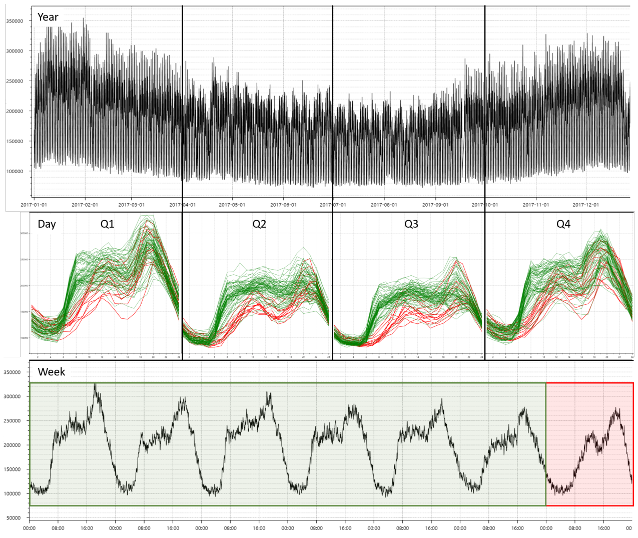


Figure 1. The top figure is the power times series for the year 2017; the middle figure shows daily power curves dependent on the season; the bottom figure shows the weekly behavior.

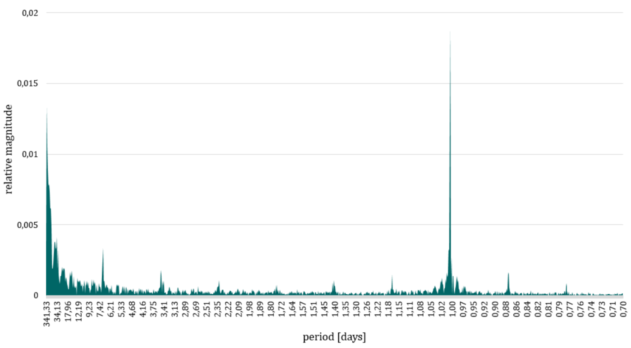


Figure 2. The periodogram represents the most relevant periods of a time series.

2.3. Record Dependencies

A record is defined as an N-dimensional process sample at a particular time. Depending on a distance metric, two consecutive records exhibit spatial nearness or not. Over time a multidimensional process passes frequently through signal range constellations represented by cluster points. We try to figure out which records are similar and collect them in clusters. Clustering refers to a method, which reduces large numbers of multidimensional records to a much smaller number of clusters. Moreover, the fact of having a highly periodic process extends the cluster notation. We will use cluster results, which includes the analysis of periodicities, to increase the computational performance on the one hand and to increase the accuracy of the forecasting model. Obviously, the power signal possesses oscillating effects. Figure 1 illustrates the most relevant periodicities. Particular emphasis should be put on daily curves marked in red, showing the different daily behavior especially in comparison between Sunday and workdays.

Therefore, the computation of the Fourier transformation decomposes the signal into frequencies as shown in figure 2. It is easy to see, that visually observed cycles can be found in the periodogram.

2.4. Record Classification

Obviously, the day of week is a good classifier for the node behavior. Thus, the forecast model takes the day of week as an input. Holidays or other running days have to be classified additionally. We also use non-linear regressions as classification methods. The ”Whit Monday” on the 5th of June 2017 for instance is a workday, which behaves like a typical Sunday. Such adaptations are integrated in the forecasting model and increase, in some cases, the forecast results significantly (see results in subsection 4.3.1.).

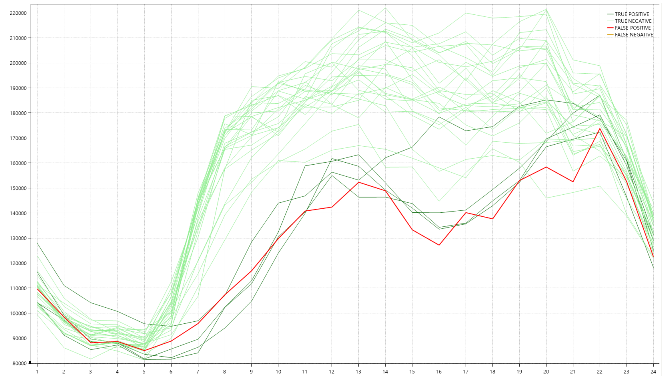


Figure 3. Daily power consumption curves between the 6th of June and the 5th of July 2017. Dark green = Sundays; Light green = other days; Red = 15th of June (holiday), which is not a Sunday.

3. FORECASTING MODEL

3.1. State Space

A good forecast does not only depend on the type of regression or model used, it also depends on the definition of the state space. First, we will give a short introduction to the state space, which we used in this paper. The state space depends on the following parameters. On the one hand, the state space includes all chosen time series data ($X = (\vec{x}_1, ..., \vec{x}_T)$), as well as weather forecast data ($Y = (\vec{y}_1, ..., \vec{y}_T)$). Both time series can be multidimensional. On the other hand, it includes technical parameters such as step size S, embedding window L and the aggregation of the parameters. Our pre-processing routine generates an equidistant, computable time series with an appropriate parameter choice.

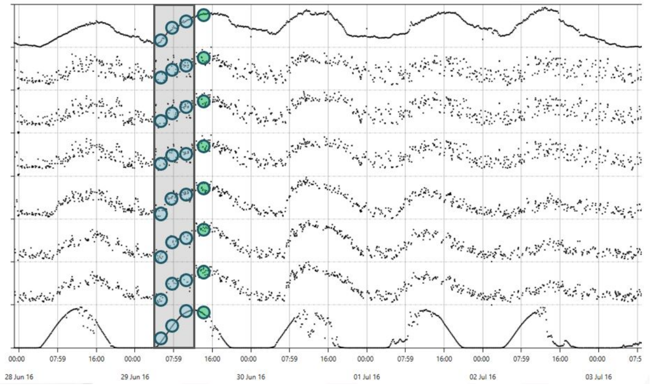


Figure 4. Sketch of the state space generation. Blue circles represent the aggregated values of the embedding window L=3, green circles represent the actual aggregated value in each coordinate direction.

From now on, if we write $X = (\vec{x}_1, ..., \vec{x}_T)$ or $Y = (\vec{y}_1, ..., \vec{y}_T)$, we always mean the time series transformed into the defined state space.

If we want to forecast a time series for a certain prognosis horizon H, we always try to approximate the relation between the input $X = (\vec{x}_1, ..., \vec{x}_{T-H})$ and the output $Y = (\vec{y}_{1+H}, ..., \vec{y}_T)$.

3.2. Time Series Forecasting Model

The long-term forecasting model uses response surface approximation techniques in order to forecast certain outcomes e.g. power. This scenario in particular, requires the input vector predictions such as weather, in order to improve the auto-associative estimates.

Response Surface Forecast Let Y be the observed multivariate process $Y = (\vec{y_1}, ..., \vec{y_T})$ and let $X = (\vec{x_1}, ..., \vec{x_T})$ define additional parameters such as weather. As introduced the forecasting model distinguishes between observable X and non-observable Y data sets. Time dependent features such as the day of week or the hour are important regressors because of periodicities. The following definition holds for the response surface method

$$Y_h = f_{h,d}(\hat{X}_h(t)) \tag{5}$$

where for given date time t, time dependent features are calculated:

- $h \in N_+$: time period of the forecasting day in [min] e.g., $h = \frac{x}{60}, x \in [1, \cdots, 1440]$
- $d \in [1, \cdots, 7]$: day of week of the forecast day

Response Surface with Classification In addition to the Response surface method daily curves are classified at the beginning of the training phase. The knowledge of special calendar days such as holidays can be approximated with a higher accuracy, if they are classified in the history. The estimation function is rewritten by

$$Y_h = f_{h,C(d)}(\hat{X}_h(t)) \tag{6}$$

where $C(d)$ is the classification function, i.e. learned feast days, for a certain day d.

Fourier transformation with Residua Response Surface Forecast The consolidation of the Fast Fourier transformation, which covers the basic oscillation of a time series, with machine learning algorithms approximating the error term, is at the heart of this forecasting model. The FFT approach is described by Press at al. (2007). Subsequently, the error function f is approximated by nonlinear regressions, whereby the error term is the consequence of the FFT approximation.

$$Y_h = \hat{Y}_h^{FFT} - f_{h,d}(\hat{X}_h) \tag{7}$$

Let F^s be the Fourier transform sorted by the magnitude and defined as follows

$$F^S(\hat{y}) = (F(\hat{y}_{\pi(1)}), \cdots, F(\hat{y}_{\pi(M)}))(8)$$

with

$$\pi = \{\pi(1), ..., \pi(M) : |F(\hat{y}_{\pi(1)})| \geq ... \geq |F(\hat{y}_{\pi(M)})|\} \quad (9)$$

By the aid of a distance-neighborhood

$$U_t^p = \{F(\hat{y}_{\pi(1)}), \cdots, Q_{(F^s)(p)}\} \tag{10}$$

where $Q_{(F^s)(p)}$ is the quantile function of the sorted Fourier transformation the estimate follows by

$$\hat{y}_t^{FFT} = \frac{1}{M} \sum_{y \in U_t^p} y \cdot e^{(2\pi i \frac{j \cdot k}{M})} \tag{11}$$

and p controls the amount of 'relevant' periods.

3.3. Non-Linear Regressions

The main challenge is to find a problem specific, non-linear regression minimizing the target function. We used four different regressions to forecast the power consumption. For more details, see the appendix.

4. VERIFICATION OF FORECASTING RESULTS

4.1. Error Functions

The objective is to find an appropriate error function. The following error functions are mostly used in the literature.

Mean Absolute Percentage Error

The mean absolute percentage error (mape) is a measure of prediction accuracy of a forecasting method in statistics, for example in trend estimation. It usually expresses accuracy as a percentage and is defined by the formula:

$$e = \frac{1}{n} \sum_{t=0}^{n} \left| \frac{y_t - y_t^*}{y_t} \right| \tag{12}$$

Symmetric Mean Absolute Percentage Error

The symmetric mean absolute percentage error (smape) is an accuracy measure based on percentage (or relative) errors. It is defined as follows:

$$e = \frac{1}{n} \sum_{t=0}^{n} \frac{|y_t - y_t^*|}{(|y_t| + |y_t^*|)/2} \tag{13}$$

Mean Absolute Error

The mean absolute error (mae) is a measure, as the name suggests, averaging the mean absolute error:

$$e = \frac{1}{n} \sum_{t=0}^{n} \left| \frac{y_t - y_t^*}{Y_{Range}} \right| \tag{14}$$

For comparability purposes we scale the error by the signal range.

4.2. Forecasting Results

In this section we present forecasting results depending on:

1. State Space
   - Input Parameters
   - Embedding Window - L
   - Step Size - S
2. Methodology
   - Response Surface - RS
   - Fourier transformation - FT
   - Response Surface with Classification Inputs - RSC
3. Regressions
   - Kernel
   - Logistic
   - RBFNN
   - Sparse Grids

The goal is to forecast the power consumption p of a certain location by using different settings (as described before). The input parameters are forecast weather data (temperature, humidity, sun etc.). To compare different models, we use the estimation of the mean absolute error (mae).

Tables 1 - a photovoltaic asset with consumers - and 2 - consumers only - present the results for two defined locations by using air temperature and global radiation (forecasts) as inputs. Further, the step size is one hour, and the embedding window is three hours. As shown in the following table, we examine different regressions, months and methodologies.

		RS			FT			RSC		
Month	Regression	mape	smape	mae	mape	smape	mae	mape	smape	mae
7-2017	Kernel	40.4	22.5	**2.9**	131.7	42.9	5.2	41.4	22.8	**2.9**
7-2017	Logit	63.8	26.2	3.0	156.3	52.2	8.5	73.2	26.5	3.2
7-2017	RBFNN	57.9	33.1	3.5	93.4	39.3	5.4	51.0	34.8	3.6
12-2016	Kernel	21.5	24.8	6.0	22.6	27.0	6.0	21.6	24.9	6.0
12-2016	Logit	17.6	20.1	5.0	34.0	43.6	8.8	18.5	20.9	5.0
12-2016	RBFNN	17.7	19.7	**4.7**	23.3	27.5	6.2	18.3	20.4	**4.9**

Table 1. Location 1 - summer versus winter.

		RS			FT			RSC		
Month	Regression	mape	smape	mae	mape	smape	mae	mape	smape	mae
7-2017	Kernel	5.9	5.6	**2.9**	7.8	7.7	3.8	6.1	5.9	3.0
7-2017	Logit	8.1	8.0	4.2	13.1	13.0	6.9	7.8	7.7	4.0
7-2017	RBFNN	6.3	6.2	3.0	8.0	8.4	4.1	6.5	6.4	3.1
12-2016	Kernel	7.5	7.7	5.3	7.0	7.1	4.7	7.2	7.4	5.1
12-2016	Logit	7.1	7.3	5.1	10.6	10.7	7.5	6.7	6.8	4.8
12-2016	RBFNN	6.8	6.8	4.6	6.1	6.3.	4.2	6.3	6.4	4.3

Table 2. Location 2 - summer versus winter.

There are some interesting observations:

- The error functions mape and smape are very sensitive with require to location 1. The reason is that the signal value is very close to zero. An actual value close to zero, increases the percentage error. In cases of nodes with power generation behavior, mae is the appropriate measure.
- The Response Surface (with/ without classification) seems to be a robust method

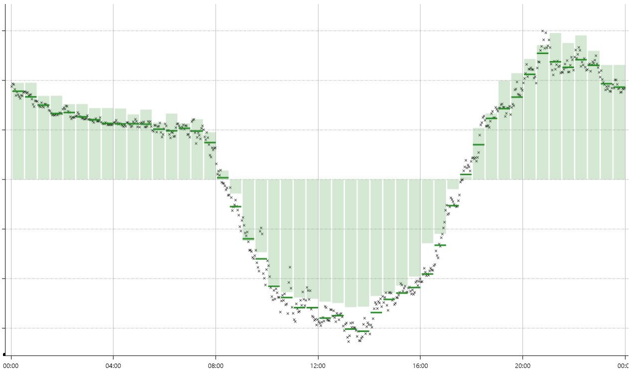


Figure 5. Daily power consumption forecast for location 1. Light green = prognosis. Green = real aggregated power consumption. Black markers = real power consumption.

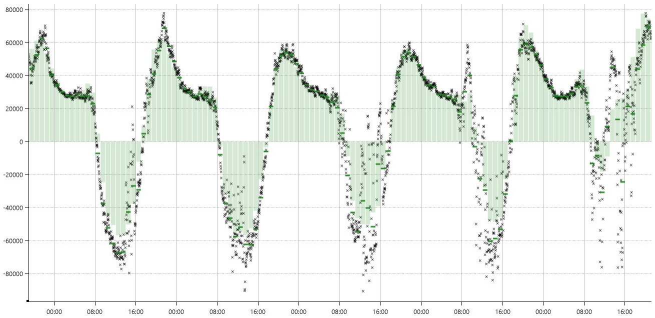


Figure 6. Monthly prognosis (kernel regression, July 17, Location1).

4.3. Influence of the methodology

The following tables 3 and 4 present the results of the RS- and FT-methodology and different input parameters (t:= temperature and r:= radiation; S=1 hour; L=3):

			FT			RS		
Location	Input	Regression	mape	smape	mae	mape	smape	mae
Location 1	t + r	Kernel	462.5	52.0	9.5	193.1	34.2	**3.2**
Location 1	t + r	Logit	866.8	57.3	8.6	102.2	34.5	**3.7**
Location 1	t + r	RBFNN	311.1	42.8	6.7	129.47	40.3	3.8
Location 2	t + r	Kernel	13.0	11.4	6.3	6.5	6.4	3.5
Location 2	t + r	Logit	13.8	13.8	6.8	9.9	10.2	5.5
Location 2	t + r	RBFNN	8.0	7.9	4.2	8.1	8.1	4.2
Location 1	r	Kernel	165.7	43.3	5.8	225.5	43.2	4.3
Location 1	r	Logit	656.7	57.0	9.3	106.2	45.3	4.8
Location 1	r	RBFNN	239.3	42.4	6.7	387.8	94.2	16.3
Location 2	r	Kernel	8.1	7.9	4.2	8.6	8.2	4.2
Location 2	r	Logit	12.4	12.9	6.9	11.3	11.4	6.0
Location 2	r	RBFNN	11.2	8.6	6.0	17.0	19.3	8.3
Location 1	t	Kernel	194.0	44.1	7.0	335.8	44.2	5.6
Location 1	t	Logit	937.5	63.1	10.7	234.0	38.3	4.5
Location 1	t	RBFNN	1034.1	44.1	8.0	307.1	39.7	4.8
Location 2	t	Kernel	7.6	7.4	3.9	6.3	6.1	3.3
Location 2	t	Logit	15.1	15.2	7.6	9.0	9.1	4.9
Location 2	t	RBFNN	8.1	8.1	4.3	6.9	6.8	**3.6**

Table 3. June 2017 - FT versus RS.

			FT			RS		
Location	Input	Regression	mape	smape	mae	mape	smape	mae
Location 1	t + r	Kernel	12.1	11.4	2.9	21.5	24.8	6.0
Location 1	t + r	Logit	16.1	16.4	4.1	17.8	20.0	5.0
Location 1	t + r	RBFNN	12.3	12.3	3.1	17.7	19.6	4.7
Location 2	t + r	Kernel	5.1	5.1	3.4	7.5	7.7	5.3
Location 2	t + r	Logit	8.2	8.6	5.3	7.2	7.3	5.1
Location 2	t + r	RBFNN	4.5	4.5	3.1	6.8	6.9	4.7
Location 1	r	Kernel	9.1	9.0	**2.3**	29.5	35.5	8.2
Location 1	r	Logit	12.2	12.6	**3.0**	27.0	32.1	7.4
Location 1	r	RBFNN	13.3	12.7	3.4	83.0	66.5	20.9
Location 2	r	Kernel	4.5	4.5	3.0	12.3	13.1	8.4
Location 2	r	Logit	5.9	6.0	4.1	12.1	12.8	8.2
Location 2	r	RBFNN	4.6	4.5	**3.0**	21.3	26.7	13.2
Location 1	t	Kernel	10.3	10.0	2.6	26.9	33.3	7.4
Location 1	t	Logit	18.3	18.6	4.6	20.5	24.0	5.5
Location 1	t	RBFNN	12.5	12.6	3.2	21.8	25.1	5.7
Location 2	t	Kernel	4.3	4.3	2.8	8.2	8.5	5.8
Location 2	t	Logit	9.1	9.6	6.0	7.2	7.4	5.1
Location 2	t	RBFNN	4.8	4.5	3.3	6.6	6.7	4.6

Table 4. December 2016 - FT versus RS.

Table 3 includes the results of the FT- and RS-Forecast in summer depending on the regression. It obvious, that in general the values of the mae of the RS-Forecast are lower than the FT-Forecast. This can be explained by a higher influence of weather such as temperature in summer. Contrary to this, the FT-Forecast is better in winter, see table 4. The next methodology extends the response surface model by feast day treatment.

4.3.1. Influence of holidays

Mondays in general have a common behavior. Moreover, calendar phenomena such as holidays can change the whole power consumption for a certain day. It makes sense to classify the forecasting day to get better results. For both locations, we forecasted the power consumption for the 5th of June 2017 (Whit Monday) and 1st of November 2016 (All Saints' Day) by changing only the methodology (additional settings: S=1 hour; L=2; Inputs = global radiation). Therefore, see table 5 and table 6.

The RSC-Forecast is a logical extension of the classical response surface technique and therefore produces better results in case of defined classifications and remain the same results in the ordinary case for certain periods and locations. The

		FT			RS			RSC		
Day	Regression	mape	smape	mae	mape	smape	mae	mape	smape	mae
2017-06-05	Kernel	45.0	32.0	3.8	53.4	40.4	3.6	53.4	40.4	**3.6**
2017-06-05	Logit	43.8	31.4	**4.1**	36.5	31.7	5.1	36.5	31.7	5.1
2017-06-05	RBFNN	233.7	83.3	16.4	182.8	72.2	13.8	182.8	72.2	**13.8**
2017-06-05	SparseGrids	85.0	45.1	8.4	68.8	58.3	6,3	68.8	58.3	**6.3**
2016-11-01	Kernel	13.4	13.8	**2.5**	22.9	30.3	3.2	25.3	34.7	3.6
2016-11-01	Logit	18.5	22.9	**2.9**	25.6	34.5	3.3	27.0	36.0	3.5
2016-11-01	RBFNN	73.0	43.8	**14.6**	99.8	74.9	18.8	102.9	80.3	19.4
2016-11-01	SparseGrids	32.4	38.3	**3.4**	31.5	37.9	3.5	26.7	33.8	3.5

Table 5. Location 1 - FT versus RSC.

		FT			RS			RSC		
Day	Regression	mape	smape	mae	mape	smape	mae	mape	smape	mae
2017-06-05	Kernel	21.6	18.7	9.4	25.7	21.9	10.9	6.6	6.2	**2.6**
2017-06-05	Logit	22.3	19.2	9.8	18.8	16.4	7.8	11.0	11.1	**4.7**
2017-06-05	RBFNN	85.0	54.5	35.0	36.9	40.8	17.2	21.1	26.1	**9.6**
2017-06-05	SparseGrids	29.3	23.3	12.2	20.5	17.9	8.9	8.4	7.9	**3.6**
2016-11-01	Kernel	12.8	12.5	6.0	20.6	17.6	11.0	5.1	5.0	**2.9**
2016-11-01	Logit	13.4	13.1	6.2	21.4	18.3	11.4	5.2	5.0	**3.0**
2016-11-01	RBFNN	55.7	37.9	26.5	32.3	35.9	17.4	23.3	29.6	**12.5**
2016-11-01	SparseGrids	12.9	12.6	5.9	24.2	20.6	13.2	7.5	7.1	**4.5**

Table 6. Location 2 - FT versus RSC.

RSC-methodology generates the best results for location 2 - only consumer - in winter and summer (see table 6). For location 1 - photovoltaic asset with consumer - a holiday is not a significant factor for the prognosis in winter. As observed before, the FT-Forecast is better than the RS- and RSC-Forecast in winter. In summer, we observe the same behavior as in subsection 4.3.: The effect of the weather is greater than the normal oscillation. Further, the holiday has another common effect on the power consumption (with exception to the logistic regression for the selected state space).

If we classify the 12th of June 2017, as a 'normal' Monday, there is no change to the RS-forecast. The FT-Forecast on that special day is better than the RS-method and worse than the RSC-method, because of supposed periodic holiday behavior.

4.4. Influence of the state space

From tables 3 and 4, it is observable, that the input parameters have a huge impact on the prognosis results, too. A well-chosen input space guarantees good forecasting results. It is observable, that global radiation and air temperature have a huge impact of the prognosis results. Without these parameters, it is not possible to approximate day-specific behaviors represented for instance by several weather conditions with the response surface methodology.

The embedding window L is another important ingredient. A larger embedding window catches the locally temporal state space representation for each time stamp. It does not directly yield a better forecasting KPI, but for regressions with more elasticity, a high degree of freedom improves the results. The results for different values of the embedding window L (and additional settings: S=1 hour; global radiation as input) are presented in tables 7 - 10.

As mentioned before, a larger embedding window implies a more precise input space. On the one hand, a higher dimensionality enables the opportunity to identify similar points/ separate non-similar points, on the other hand the performance decreases. It is conspicuous, that the Fourier methodology generates better KPI's in summer at location 2 (see table 8) in contrast to the observations in 4.3., which is explainable by the node behavior, location 2 - consumers only, and the modification of the value of the embedding window.

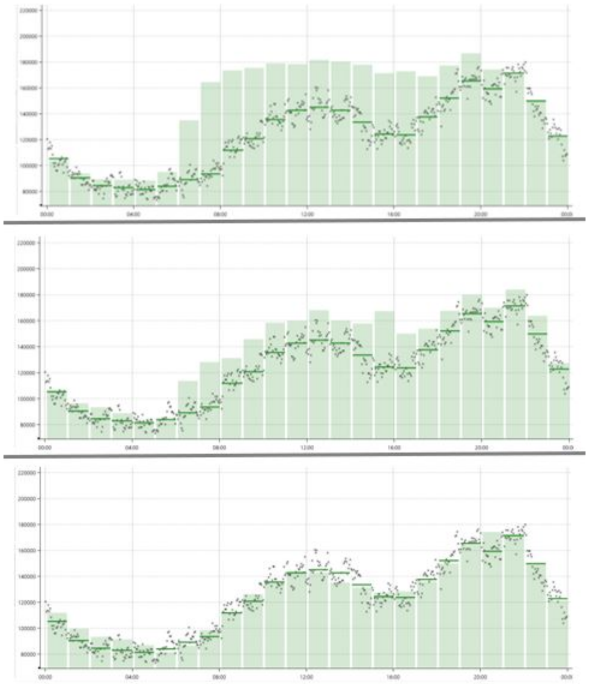


Figure 7. From top to bottom: RS, FT, RSC.

4.5. Influence of the regression

Summarizing the results of tables 3 - 10 it is observable, that kernel regression delivers the most robust forecasts. In cases of full observation, regressions with more degrees of freedom are getting more powerful. This is similar to the observation that the results of the RBFNN get better by increasing the embedding window. The combinatorial techniques of sparse grids do not perform well in high dimensions. Further, the choice of the 'right' regressions also depends on the seasonality (summer/ winter), the methodology and the state space.

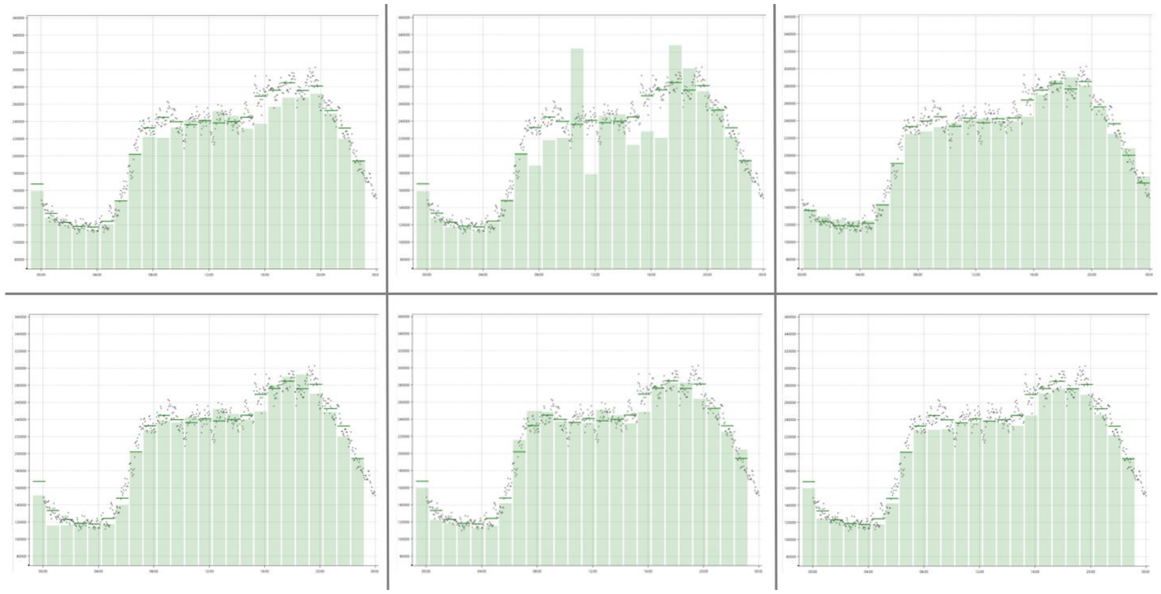


Figure 8. From left to right: Kernel, Logit, RBFNN. Top line: L=3. Bottom line: L=30. (all: methodology FT)

			FT			RS	
L	Regression	mape	smape	mae	mape	smape	mae
3	Kernel	165.7	43.3	5.8	225.5	43.2	4.3
3	Logit	656.7	57.0	9.3	106.2	45.3	4.8
3	RBFNN	239.3	42.4	6.7	387.8	94.2	16.3
6	Kernel	154.6	43.7	6.1	318.8	40.3	3.9
6	Logit	357.0	56.4	8.2	103.4	44.3	4.5
6	RBFNN	186.7	41.9	6.0	183.1	65.4	10.3
9	Kernel	156.3	43.9	6.4	212.2	35.6	3.6
9	Logit	361.7	59.7	8.3	225.1	42.5	**4.3**
9	RBFNN	188.8	41.8	5.9	80.8	35.9	**4.5**
12	Kernel	164.0	45.8	6.7	237.5	34.1	3.5
12	Logit	291.3	56.4	7.8	194.2	42.6	4.4
12	RBFNN	188.4	41.7	5.8	80.0	34.8	4.5
15	Kernel	166.0	45.2	7.1	235.4	31.4	**3.3**
15	Logit	243.6	50.5	7.0	170.7	42.5	4.6
15	RBFNN	184.5	41.2	5.8	91.0	35.9	4.5
18	Kernel	419.5	44.7	7.3	161.2	31.7	3.5
18	Logit	213.2	48.7	6.9	119.4	46.6	4.9
18	RBFNN	174.4	41.5	5.7	96.8	37.1	4.7
21	Kernel	677.6	45.5	7.5	189.0	30.0	3.4
21	Logit	314.3	47.4	6.9	289.3	48.5	5.1
21	RBFNN	270.2	41.4	5.7	113.3	37.5	4.9
24	Kernel	678.8	45.9	7.7	201.9	30.6	3.4
24	Logit	369.9	48.1	7.0	242.9	50.4	5.2
24	RBFNN	417.8	42.0	5.8	285.2	37.0	4.9

Table 7. June 2017 of Location 1 - Embedding window.

			FT			RS	
L	Regression	mape	smape	mae	mape	smape	mae
3	Kernel	8.1	7.9	4.2	8.6	8.2	4.2
3	Logit	12.4	12.9	6.9	11.3	11.4	6.0
3	RBFNN	11.2	8.6	6.0	17.0	19.3	8.3
6	Kernel	8.3	8.0	4.3	8.1	7.7	4.0
6	Logit	12.4	12.8	6.8	11.1	11.2	5.9
6	RBFNN	10.2	8.5	5.6	13.0	13.8	6.5
9	Kernel	8.8	8.5	4.6	7.1	6.9	3.7
9	Logit	13.0	13.4	6.8	10.9	10.9	5.8
9	RBFNN	8.2	7.8	4.3	11.5	12.2	6.2
12	Kernel	9.3	8.9	4.8	6.8	6.6	3.5
12	Logit	12.4	12.6	6.5	11.1	11.0	5.9
12	RBFNN	8.0	7.8	4.2	11.5	12.1	6.2
15	Kernel	9.6	9.0	4.9	6.6	6.4	**3.4**
15	Logit	11.6	11.7	6.1	11.6	11.3	6.1
15	RBFNN	7.8	7.5	4.1	10.9	11.0	5.8
18	Kernel	9.5	8.7	4.8	6.5	6.3	3.4
18	Logit	10.8	10.8	5.7	12.1	11.7	6.4
18	RBFNN	7.8	7.5	4.1	11.0	11.2	5.9
21	Kernel	9.7	8.8	4.8	6.5	6.4	3.4
21	Logit	10.4	10.4	5.5	12.6	12.2	6.6
21	RBFNN	7.6	7.4	4.0	11.0	11.3	6
24	Kernel	9.8	8.8	4.8	6.5	6.4	3.5
24	Logit	10.2	10.1	**5.4**	12.6	12.3	6.7
24	RBFNN	7.6	7.3	**3.9**	11.1	11.4	6.0

Table 8. June 2017 of Location 2 - Embedding window.

4.6. Conclusion

As presented in this chapter, external factors (such as location and seasonality) and important model features (such as methodology, state space and regression) obviously have a vital effect on the prognosis results. This is underlined by the following statements:

- In comparison to RS and FT, the RSC method improves the forecasting results on feast days.
- Consumer behavior for winter days can be forecast best with the FT method.
- Energy-generating behavior for summer days can be forecast best with the RS method.
- The most robust nonlinear regression is the Kernel regression.
- The RBFNN performs best in high-dimensional model parameter state spaces caused by the embedding window.
- The choice of state space (input parameters, step size and embedding window) is more relevant for less-periodic time series.

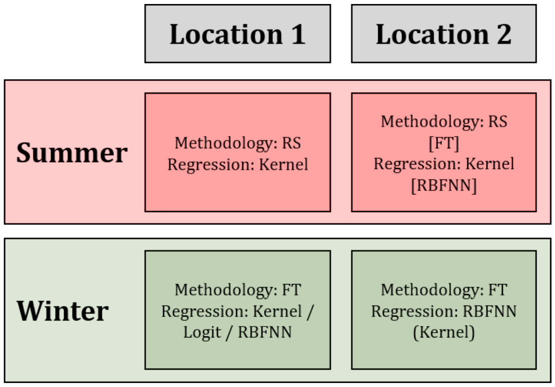
Figure 9 summarizes KPI's by best regressions in each circumstance. We classify our forecasting portfolio by the external factors such as location and seasonality. Recommendations are given in figure 10.

			FT			RS	
L	Regression	mape	smape	mae	mape	smape	mae
3	Kernel	9.1	9.0	**2.3**	29.5	35.5	8.2
3	Logit	12.2	12.6	3.0	27.0	32.1	7.4
3	RBFNN	13.3	12.7	3.4	83.0	66.5	20.9
6	Kernel	9.1	8.9	2.3	26.2	30.9	7.3
6	Logit	11.7	11.8	**2.9**	25.3	29.7	6.9
6	RBFNN	12.4	11.8	3.0	57.8	48.8	14.0
9	Kernel	9.6	9.4	2.4	24.0	27.9	6.7
9	Logit	12.5	12.6	3.1	23.5	27.3	6.5
9	RBFNN	12.4	11.8	3.1	24.6	28.8	6.7
12	Kernel	9.9	9.7	2.5	22.3	25.6	6.3
12	Logit	12.4	12.4	3.1	22.2	25.5	6.1
12	RBFNN	11.9	11.5	3.0	23.5	27.3	6.4
15	Kernel	10.2	10.0	2.6	20.6	23.5	5.8
15	Logit	13.0	13.0	3.2	21.2	24.2	5.9
15	RBFNN	11.4	11.1	2.8	22.8	26.5	6.3
18	Kernel	10.6	10.3	2.7	19.3	21.8	5.4
18	Logit	12.8	12.7	3.1	20.4	23.2	5.7
18	RBFNN	10.1	10.0	**2.5**	22.4	25.9	6.2
21	Kernel	10.7	10.4	2.7	18.1	20.1	5.1
21	Logit	12.2	12.2	3.0	19.8	22.4	5.5
21	RBFNN	10.1	10.1	2.5	22.2	25.5	6.2
24	Kernel	10.9	10.6	2.8	16.9	18.5	4.7
24	Logit	11.4	11.3	2.9	19.3	21.8	5.4
24	RBFNN	10.0	9.9	2.5	21.9	25.1	6.1

Table 9. Dec 2016 of Location 1 - Embedding window.

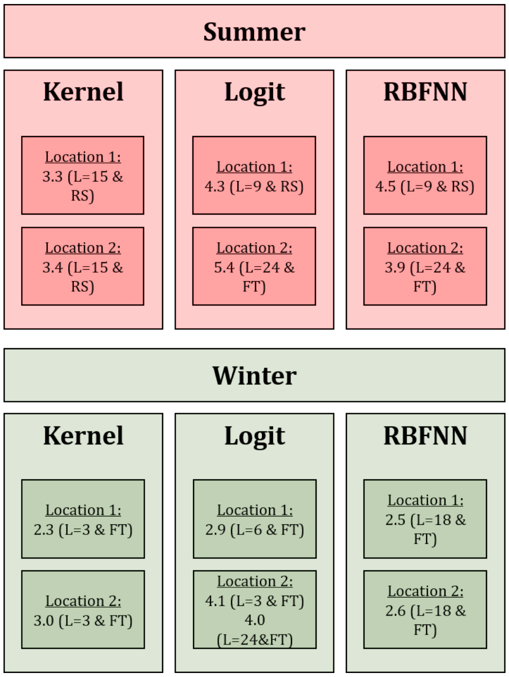
			FT			RS	
L	Regression	mape	smape	mae	mape	smape	mae
3	Kernel	4.5	4.5	**3.0**	12.3	13.1	8.4
3	Logit	5.9	6.0	**4.1**	12.1	12.8	8.2
3	RBFNN	4.6	4.5	3.0	21.3	26.7	13.2
6	Kernel	4.7	4.7	3.1	11.1	11.7	7.5
6	Logit	6.0	6.0	4.2	11.3	11.9	7.6
6	RBFNN	4.5	4.4	3.0	15.5	17.9	9.6
9	Kernel	4.9	4.9	3.3	10.0	10.4	6.8
9	Logit	5.9	6.0	4.1	10.2	10.6	7.0
9	RBFNN	4.5	4.4	2.9	11.9	13.1	8.2
12	Kernel	5.1	5.1	3.4	9.0	9.2	6.3
12	Logit	6.5	6.5	4.3	9.3	9.6	6.5
12	RBFNN	4.2	4.1	2.8	10.9	11.8	7.7
15	Kernel	5.2	5.2	3.4	8.2	8.4	5.9
15	Logit	7.0	7.3	4.5	8.8	9.0	6.2
15	RBFNN	4.2	4.1	2.8	9.9	10.3	7.0
18	Kernel	5.3	5.2	3.5	7.8	7.9	5.6
18	Logit	6.5	6.6	4.3	8.5	8.6	6.0
18	RBFNN	4.0	3.9	**2.6**	9.9	10.3	7.0
21	Kernel	5.3	5.2	3.5	7.5	7.5	5.3
21	Logit	6.5	6.5	4.3	8.3	8.4	5.9
21	RBFNN	4.1	4.0	2.7	9.8	10.2	6.9
24	Kernel	5.1	5.0	3.3	7.2	7.1	5.1
24	Logit	6.1	6.2	**4.0**	8.2	8.3	5.8
24	RBFNN	4.0	3.9	2.6	9.7	10.1	6.9

Table 10. Dec 2016 of Location 2 - Embedding window.



	Location 1	Location 2
Summer	Methodology: RS Regression: Kernel	Methodology: RS [FT] Regression: Kernel [RBFNN]
Winter	Methodology: FT Regression: Kernel / Logit / RBFNN	Methodology: FT Regression: RBFNN (Kernel)

Figure 10. Recommendations depending on the location and seasonality.



	Kernel	Logit	RBFNN
Summer	Location 1: 3.3 (L=15 & RS)	Location 1: 4.3 (L=9 & RS)	Location 1: 4.5 (L=9 & RS)
	Location 2: 3.4 (L=15 & RS)	Location 2: 5.4 (L=24 & FT)	Location 2: 3.9 (L=24 & FT)
Winter	Location 1: 2.3 (L=3 & FT)	Location 1: 2.9 (L=6 & FT)	Location 1: 2.5 (L=18 & FT)
	Location 2: 3.0 (L=3 & FT)	Location 2: 4.1 (L=3 & FT) 4.0 (L=24&FT)	Location 2: 2.6 (L=18 & FT)

Figure 9. KPI's by type of regression.

Summarizing our experiences in the area of power demand forecasting, one important lesson to be learned is the need to consider the various behavior types and seasonality. In order to perform best under all contingencies, the methodology has to be flexible. Especially, the periodicity of a node influences this choice. Beside the perennial difficulty to select 'optimal' regressors, the special knowledge of holidays impacting power demand is central to this model and will improve results significantly.

5. REFERENCES

ALT, H.W. (1999). *Lineare Funktionsanalysis*. Springer, 6. Auflage. Berlin.

BACKHAUS, K., ERICHSON, B., PLINKE, W. & WEIBER, R. (2000). *Multivariate Analysemethoden - Eine anwendungsorientierte Einführung*. 9. Auflage. Springer Verlag.

BUNGARTZ, H. J., GRIEBEL, M. (2004). *Sparse Grids*. Acta Numerica. Cambridge University Press.

GARCKE, J. (2004). *Maschinelles Lernen durch Funktionsrekonstruktion mit verallgemeinerten dünnen Gittern*. Dissertation. Bonn.

GARCKE, J., GERSTNER, T. & GRIEBEL, M. (2013). *Time Series Forecasting using Sparse Grids*. Artikel. Bonn und Frankfurt.

GRIEBEL, M. & KNAPEK, S. (2000). *Optimized general sparse grid approximation spaces for operator equations*.

Mathematics of computation.

GRIEBEL, M., SCHNEIDER, M. & ZENGER, C. (1992). *A combination technique for the solution of sparse grid problems*. IMACS. Netherlands.

HANKE-BOURGEOIS, M. (2009. *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens*. Vieweg+Teubner, 3. Auflage. Wiesbaden.

HASTIE, T., TIBSHIRANI, R. & FRIEDMAN, J. (2008). *The Elements of Statistical Learning*. Springer Second Edition. Stanford.

KIRSCH, A. (2011). *An Introduction to the Mathematical Theory of Inverse Problems*. Springer. Second Edition. Karlsruhe.

KRUSE, R., BORGELT, C., KLAWONN, F., MOEWES, C., RUSS, G. & STEINBRECHER, M. (2011). *'Computational Intelligence - Eine methodische Einfuehrung in künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze*. Vieweg, 1. Auflage. Wiesbaden.

KRIESEL, D. (2005). *Ein kleiner Überblick über neuronale Netze*. http://www.dkriesel.com/science/neural$_networks.ZETA2-DE.$

KANTZ, H. & RAGWITZ, M. (14:1935, 1945, 2004). Phase space reconstruction and nonlinear predictions for stationary and nonstationary markovian processes. *International Journal of Bifurcation and Chaos*.

KANTZ, H. & SCHREIBER, T. (2004). *Nonlinear Time Series Analysis*. Cambridge University Press. Dresden.

PRESS, W., TEUKOLSKY, S., VETTERLING, W. & FLANNERY, B. (2007). *Numerical Recipes - The Art of Scientific Computing*. 3rd Edition. Cambridge University Press.

RIEDER, A. (2003). *Keine Probleme mit Inversen Problemen*. Vieweg, 3. Auflage. Wiesbaden.

SMOLYAK, S. (1963). *Quadrature and interpolation formulas for tensor products of certain classes of functions*. Dokl. Akad. Nauk (1042-1045).

SCHOELKOPF, B. & SMOLA, A. (2002). *Learning with Kernels*. MIT Press. Cambridge, Massachusetts.

TIKHONOV, A. & ARSENIN, V. (1977). *Solution of Ill-Posed Problems*. Washington: Winston Sons.

TIKHONOV, A. & LEONOV, A. (1998. *Non-Linear Ill-Posed Problems*. Chapman and Hall.

ZENGER, C. (1990). Sparse Grids. In Hackbusch: *Parallel Algorithms for Partial Differential Equations*. 6th GAMM-Seminar Kiel. 1990. Band 31. Vieweg-Verlag.

6. APPENDIX

6.1. Non-linear regressions

6.1.1. Kernel Regression

Kernel regression is a non-parametric method computing the conditional expectation of a random variable through a kernel density estimation. It is applicable to different time series analysis problems and therefore Auto-Associative Kernel Regression is introduced first. As mentioned in the introduction a forecast can be accomplished by comparing the current system state to states in the temporal or spatial neighborhood of the historical data set.

Auto-Associative Kernel Regressions

The AAKR method is very useful in the detection of abnormal states $\vec{x} \in R^N$, by searching for the most similar points to the current one, interpreting a high difference as an abnormal behavior. Therefore, AAKR is commonly used for anomaly detection.
However, we will utilize kernel regression in order to locate similar points. The comparison between a new measurement $\vec{x}_t$ for time t and X as defined in chapter 2 is done by using the Nadaraya-Watson-Estimator defined below, which is usually a Gaussian kernel in combination with a Euclidean distance. The estimate is calculated through a weighted sum of all records in the training matrix X or, more precisely, the conditional expectation of a process X at time t given, we have observed the sample $\vec{x}_t$ for X_t

$$\vec{x_t}^* = \mathbf{E}[X_t|X_t = \vec{x}_t] = \int_{x\in X} xp(x|X_t = \vec{x}_t)\, dx \qquad (15)$$

$$\approx \frac{\sum_i^M K(\vec{x}_t, \vec{x}_i)\vec{x}_i}{\sum_i^M K(\vec{x}_t, \vec{x}_i)} \qquad (16)$$

with a Gaussian kernel

$$K(\vec{x}_t, \vec{x}_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(-\frac{1}{2\sigma^2}\|\vec{x}_t - \vec{x}_i\|^2\Big), \quad i = 1, \ldots, N \qquad (17)$$

or with the kernel for the nearest neighbor

$$K(\vec{x}_t, \vec{x}_i) = \delta(\mathbf{1}_A), \quad A = \{i =_{j=1,\ldots,M} \|\vec{x}_j - \vec{x}_t\|\} \qquad (18)$$

Please note that the distance measure has naturally an impact on the kernel weights. Typically, we think of the

- Euclidean distance

$$d(\vec{x}, \vec{y}) = \|\vec{y} - \vec{x}\|_2 = \sqrt{(\vec{y} - \vec{x})^\top(\vec{y} - \vec{x})} \qquad (19)$$

- Mahalanobis distance

$$d(\vec{x}, \vec{y}) = \|\vec{y} - \vec{x}\|_M = \sqrt{(\vec{y} - \vec{x})^\top S^{-1}(\vec{y} - \vec{x})}, \qquad (20)$$

where S is the covariance matrix.

The influence of different measures especially for highly correlated signals has to be analyzed.

Predictions with Kernel Regressions

The following analysis is discussed in Kantz and Schreiber (2004) and Kantz and Ragwitz (2004), where the scalar locally constant predictor is defined and this predictor is generalized to the vectoral case. Given a vector time series $X = (\vec{x}_1, \ldots, \vec{x}_T)$, and an embedding dimension L we construct ourselves a embedding vector $\mathbf{x}_n = (\vec{x}_n, \vec{x}_{n-1}, \ldots, \vec{x}_{n-L+1})$. In order to predict a time $\triangle t$ ahead starting in time t we first choose a ϵ and generate a neighborhood $\mathcal{U}_t$ of $\mathbf{x}_t$ defined by $\mathcal{U}_t = \{\mathbf{x}_n : \|\mathbf{x}_n - \mathbf{x}_t\|_L \leq \epsilon\}$, where $\|\mathbf{x}_n - \mathbf{x}_t\|_L = \sum_{i=0}^{L-1} \|\vec{x}_{n-i} - \vec{x}_{t-i}\|$. The locally constant predictor for the future time $t + \triangle t$ is then

$$\hat{\mathbf{x}}_{t+\triangle t} = \frac{1}{|\mathcal{U}_t|} \sum_{\mathbf{x}_n \in \mathcal{U}_n} \mathbf{x}_{n+\triangle t}. \qquad (21)$$

Remember that $\triangle t$ can be the next time step or even a time sequence $(1, \ldots, H)$ defining a trajectory with horizon H.
The Kernel Regression introduced above is required to propose the following prediction algorithm. In contrast to the simple locally Constant Predictor introduced in Kantz and Schreiber (2004) the predictor with AAKR is not an average of representative points. Instead it is a weighted sum over the whole data set, where similar points receive higher weights

$$\vec{x}^*_{t+\triangle t} = \frac{\sum_i^M K(\mathbf{x}_t, \mathbf{x}_i)\mathbf{x}_{i+\triangle t}}{\sum_i^M K(\mathbf{x}_t, \mathbf{x}_i)} \qquad (22)$$

Clusters derived in the time series analysis step are useful to increase the performance of the predictor, by referring to records of a specific cluster $C = C(\vec{x}_t)$, which is the nearest to the current observation $\vec{x}_t$. The estimate is defined as follows

$$\vec{x}^*_{t+\triangle t} = \frac{\sum_{x_i \in C} K(\mathbf{x}_t, \mathbf{x}_i)\mathbf{x}_{i+\triangle t}}{\sum_{x_i \in C} K(\mathbf{x}_t, \mathbf{x}_i)} \qquad (23)$$

Furthermore, similarity grouping separates stochastically in-dependent signals, such that the approach presented can be applied to similarity group. Thus, the identification of similar points is much more exact. For more details see for example Hastie at al. (2008), Kantz and Schreiber (2004) or Schoelkopf and Smola (2002).

6.1.2. Logistic Regression

Normally, logistic regression is used to find a probability that a measurement $\vec{y}_t$ belongs to a certain category (mostly two categories) by transforming a given observation $\vec{x}_t$ into an interval $[0, 1]$ (which is interpreted as the probability that $\vec{y}_t$ belongs to category 1).

$$P(Y_t = \vec{y}_t \mid X_t = \vec{x}_t) = \frac{\exp(\beta_0 + \vec{x}_t^T \cdot \vec{\beta}_1)}{1 + \exp(\beta_0 + \vec{x}_t^T \cdot \vec{\beta}_1)} \tag{24}$$

$$= \frac{1}{1 + \exp(-(\beta_0 + \vec{x}_t^T \cdot \vec{\beta}_1))} =: h_{\vec{\beta}}(\vec{x}_t) \tag{25}$$

An optimal solution of $\vec{\beta} = (\beta_0, \vec{\beta}_1)$ can be found by maximizing the logarithmic likelihood function using a gradient descent for example.

$$L(\vec{\beta} \mid x) = P(Y \mid X; \vec{\beta}) = \prod_{t=1}^{T} P(\vec{y}_{t+H} \mid \vec{x}_t; \vec{\beta}) \tag{26}$$

$$= \prod_{t=1}^{T} h_{\vec{\beta}}(\vec{x}_t)^{\vec{y}_t} (1 - h_{\vec{\beta}}(\vec{x}_t))^{(1-\vec{y}_t)} \tag{27}$$

and the logarithmic likelihood function divided by T:

$$\frac{\log(L(\vec{\beta} \mid X))}{T} \overset{!}{=} maximize \tag{28}$$

After the training step, a new prognosis for an observation $\vec{x}_{new}$ is done by solving

$$h_{\vec{\beta}^\star}(\vec{x}_{new}) = \vec{y}^\star_{new} \tag{29}$$

For more details see Backhaus at al. (2000) or Press at al. (2007).

6.1.3. Sparse Grids

The combination technique of sparse grids to approximate a functional relation is not a common method, this time. Zenger and Bungartz developed the theory about the usage of sparse grids in the early 90s. A huge advantage of using sparse grids is that every possible (non-linear, non-quadratic and so on) relation can be approximated. The combination technique uses different sparse grids with different density of points in each coordinate direction, but same basic function. The following theory depends on papers by Zenger, Bungartz, Garcke, Griebel and Gerstner (see Bungartz and Griebel (2004), Garcke (2004), Garcke at al. (2013), Griebel at al. (1992), Zenger (1990)).

A good overview to the theory of regularization of ill-posed problems and optimization is given by Alt, Rieder, Hanke-Borgeois and Tikhonov (see Alt (1999), Hanke-Bourgeois (2009), Kirsch (2011), Rieder (2003), Tikhinov and Arsenin (1977), Tikhonov at al. (1998)), the theory about error estimation is well described by Garcke, Knapek and others (see Griebel and Knapek (2000) for example).

The goal is to find a function f which approximates the relation between X and Y well:

$$\vec{y}_t \approx \vec{y}_t^\star = f(\vec{x}_t) = \sum_{i=1}^{N} \alpha_i \cdot \phi_i(\vec{x}_t) \tag{30}$$

where $\phi_i(\cdot)$ are basic functions and α_i are coefficients, which have to be optimized. To find a good solution of the approximation function f, we want to minimize the mean squared error (MSE). Further, we add a regularization term:

$$min \overset{!}{=} \frac{1}{T} \sum_{t=1}^{T} (\vec{y}_t - f(\vec{x}_t))^2 + \lambda \mathcal{S}(f) \tag{31}$$

$\mathcal{S}$ will be the quadratic L_2-norm of function f

$$\mathcal{S}(f) = \|\mathcal{C}f\|^2_{L_2} = \|\nabla \sum_{i=1}^{N} \alpha_i \phi_i(\cdot)\|^2_{L_2} \tag{32}$$

and can be interpreted as a smoothness operator, where ∇ is the gradient. If λ is increasing, the solution of f becomes smoother. Using the last two equations, we obtain the formula:

$$\frac{1}{T} \sum_{t=1}^{T} \left(\vec{y}_t - \sum_{i=1}^{N} \alpha_i \phi_i(\vec{x}_t) \right)^2 + \lambda \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \langle \nabla \phi_i(\cdot), \nabla \phi_j(\cdot) \rangle_{L_2} \tag{33}$$

By solving the minimization problem, we can reach the following analytic solution for fixed $k = 1, ..., N$:

$$\sum_{i=1}^{N} \alpha_i \left(T\lambda \langle \nabla \phi_i(\cdot), \nabla \phi_k(\cdot) \rangle_{L_2} + \sum_{t=1}^{T} \phi_i(\vec{x}_t) \phi_k(\vec{x}_t) \right) \tag{34}$$

$$= \sum_{t=1}^{T} \vec{y}_t \phi_k(\vec{x}_t) \tag{35}$$

And the same solution represented by matrices and vectors

$$(\lambda \cdot T \cdot C + B \cdot B^T)\vec{\alpha} = BY \tag{36}$$

$$\Leftrightarrow \vec{\alpha} = (\lambda \cdot T \cdot C + B \cdot B^T)^{-1} BY \tag{37}$$

where $\vec{\alpha}$ is the N-dimensional vector of all coefficients α_i and the two matrices B and C represented by

$$B = \begin{pmatrix} \phi_1(\vec{x}_1) & ... & \phi_1(\vec{x}_T) \\ \vdots & & \vdots \\ \phi_N(\vec{x}_1) & ... & \phi_N(\vec{x}_T) \end{pmatrix} \tag{38}$$

and

$$C = \begin{pmatrix} \langle \nabla\phi_1, \nabla\phi_1 \rangle_{L_2} & ... & \langle \nabla\phi_1, \nabla\phi_N \rangle_{L_2} \\ \vdots & & \vdots \\ \langle \nabla\phi_N, \nabla\phi_1 \rangle_{L_2} & .... & \langle \nabla\phi_N, \nabla\phi_N \rangle_{L_2} \end{pmatrix} \quad . \tag{39}$$

In the next sector, there will be a short introduction to sparse grids and the basic functions $\phi(\cdot)$. A sparse grid has the following parameters, first the dimension D, which depends on the size of the input vectors $\vec{x}_t$, and second, the maximal grid level L. We only attend to equidistant grids (for other examples see Smolyak in Smolyak (1963)). So, for a given dimension D, the grid level vector $\vec{L} = (l_1, ..., l_D)^T$ describes the density of points. For a fixed coordinate direction $h_d = 2^{-\vec{L}_d}$ is the mesh sizes for $d = 1, ..., D$ and describes the gap between two points. Further, we use a multi-index $\vec{i} = (i_1, ..., i_D)^T$ with $i_d \in \{0, ..., 2^{\vec{L}_d}\}$ for $d = 1, ..., D$ to represent all points of a certain sparse grid $\Omega_{\vec{L}}$.
In this paper, the basic functions $\phi(\cdot)$ are linear, piecewise functions:

$$\phi_{l_d,i_d}(\vec{x}_{t,d}) = \begin{cases} 1- \mid \frac{\vec{x}_{t,d}}{h_{l_d}} - i_d \mid, & \vec{x}_{t,d} \in I \\ 0, & otherwise \end{cases} \tag{40}$$

with

$$I = \vec{x}_{t,d} \in [(i_d - 1)h_{l_d}, (i_d + 1)h_{l_d}] \cap [0, 1] \tag{41}$$

where $\vec{x}_{t,d}$ is the d-th component of $\vec{x}_t$. We get the multi-dimensional basics functions by building the product over all coordinate directions

$$\Phi_{\vec{l},\vec{i}}(\vec{x}_t) = \prod_{d=1}^{D} \phi_{l_d,i_d}(\vec{x}_{t,d}) \quad . \tag{42}$$

Finally, the approximation function f is represented by

$$f_L^c(\vec{x}_t) = \sum_{q=0}^{D-1} (-1)^q \begin{pmatrix} D-1 \\ q \end{pmatrix} \sum_{|\vec{l}|=L+(D-1)+q} f_{\vec{l}}(\vec{x}_t) \tag{43}$$

and

$$f_{\vec{l}}(\vec{x}_t) = \sum_{i_1=0}^{2^{\vec{l}_1}} ... \sum_{i_D=0}^{2^{\vec{l}_D}} \alpha_{\vec{l},\vec{i}} \cdot \Phi_{\vec{l},\vec{i}}(\vec{x}_t) \quad . \tag{44}$$

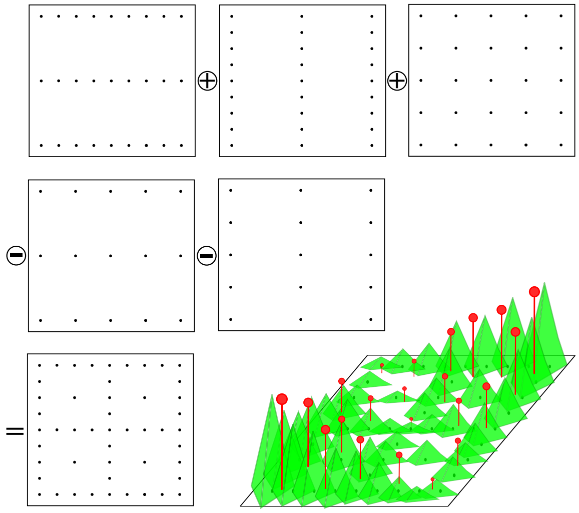


Figure 11. Combination technique of sparse grids (L=3 and D=2) and an example for a function approximation.

The dimension of function reconstruction depends on the maximum grid level G and the dimension of the input space D. By increasing maximum grid level G, the error is decreasing theoretically. In practice the error term is not decreasing whenever the maximum grid level is increasing. An optimal value for G must be found. Last, the evaluation of a new point also depends on the dimension of the input space and the maximum grid level:

- $dim(\Omega_G^D) = \mathcal{O}(\frac{1}{h_G} \log(\frac{1}{h_G})^{D-1}) = \mathcal{O}(2^G \log(2^G)^{D-1})$ with $h_G = 2^G$
- Error: $\|g - g_G^c\|_{L_p} = \mathcal{O}(h_G^2 \log(\frac{1}{h_G})^{G-1})$ $= \mathcal{O}(2^{2G} \log(2^G)^{D-1})$
- Costs of an evaluation: $\mathcal{O}(G^{D-1})$

6.1.4. Radial Basis Function Neural Networks

Radial basis function neural networks (RBFNNs) are a special representative of artificial neural networks (ANNs) with only one hidden layer (and one input and one output layer). ANNs have the huge advantage that ANNs are able to approximate nearly every kind of relation in data and that they automatically add new neurons or edges between two neurons and/ or delete existing neurons or edges between two neurons. The number of neuron of the input layer I is number of the dimension of the input vector. O is the number of neurons of the output layer which is the same as the dimension of the output vector. The number of neurons in the hidden layer H can vary. For the theory, we accept that H is a fixed number of neurons of the hidden layer. Every neuron of the hidden layer has an own activation function which is a representative of a radial Gaussian function

$$a_h(\vec{x}_t) = \exp\left(\frac{-r_h^2}{2 \cdot \sigma_h^2}\right) \tag{45}$$

with centre $\vec{c}_h$ for $h = 1, ..., H$ and the Euclidean distance

$$r_h = \|\vec{x}_t - \vec{c}_h\| = \sqrt{\sum_{i=1}^{I} (\vec{x}_{t,i} - \vec{c}_{h,i})^2} \tag{46}$$

where $\vec{x}_{t,i}$ is the i-th value in time t. Additionally, a multidimensional radial gaussian function with covariance matrix Σ can produce better results. (Remark: All weights of edges between the input neurons and hidden neurons are equal to one.)

The j-th output is calculated as follows

$$o_{t,j} = \sum_{h=1}^{H} w_{h,j} \cdot a_h(\vec{x}_t) \tag{47}$$

where $w_{h,j}$ is the weight of the edge between the h-th hidden neuron and the j-th output neuron (has to be optimized in the training part) and

$$\vec{y}^{\star}_{t+H} = (o_{t,1}, ..., o_{1,O})^T \tag{48}$$

for $h = 1, ..., H$ and $t = 1, ..., T$. The training phase includes certain teaching steps which try to optimize the following items:

- weights $w_{h,j}$ of the edges between the hidden neurons and output neurons
- number of hidden neurons H
- centres c_h of the activation function of the hidden neurons
- standard deviation σ_h of the activation function of the hidden neurons

Finally, for a new observation the expected output is calculated by summarizing all return values of all activation functions of the hidden neurons multiplied by their weights.

$$\vec{y}^{\star}_{new+H} = \begin{pmatrix} \sum_{h=1}^{H} w_{h,1} \cdot a_h(\vec{x}_{new}) \\ \vdots \\ \sum_{h=1}^{H} w_{h,O} \cdot a_h(\vec{x}_{new}) \end{pmatrix} \tag{49}$$

For more details of RBFNN's see the work of Kriesel (2005) or Kruse at al. (2011).