Data Analysis

Patrick Lang and Jürgen Franke

1 Data Sources

Today, due to the continuously advancing digitalization of production and business processes, data is being produced and often archived in an amount that only a few years ago would have been hardly imaginable. The drivers of this trend are the availability of numerous new sensor technologies and higher-performance data storage equipment. For many production processes in large industry, all potentially relevant adjustment and equipment parameters are now being recorded at high temporal resolution and then stored. Moreover, implementation of the Industry 4.0 concept, in which diverse, context-specific communication is to flow between production goods and production equipment, and between one production step and another, will lead to numerous additional data streams and, consequently, to a further significant increase in data volume.

The availability of ever more complex and precise measurement and analysis procedures also leads to the generation of larger quantities of data. One can think, for example, of the Next Generation Sequencing Procedure for genome analysis in the context of personalized medicine. Here, data on the order of terabytes can easily accrue with each analysis.

A further source for this flood of data is the increased networking of our world. One only has to consider the many data streams in the Internet, such as real-time stock market index updating; numerous social media with their own news channels; on-line service providers, such as eBay and Amazon, with their movements of customer data; or locally-resolved meteorological data streams. Moreover, in addition to current data, for

P. Lang (✉) · J. Franke
Kaiserslautern, Germany
e-mail: patrick.lang@itwm.fraunhofer.de

H. Neunzert, D. Prätzel-Wolters (eds.), *Currents in Industrial Mathematics*,
DOI 10.1007/978-3-662-48258-2_4

almost any question that can be asked, there exists a data base with corresponding historical data to answer it. Not only is the quantity of data increasing, but the opportunities of the individual for utilizing the publicly accessible flood of data are increasing as well.

Data, as it is generally understood, is not necessarily a structured combination of numerical values in the form of vectors, matrices, or time series; it can also refer to semi-structured or unstructured information, such as a simple piece of text. Due to its nature, the latter is not directly accessible to mathematical processing. Instead, it must first be prepared appropriately. The methods of information retrieval and text mining deal with this topic.

Media reports also currently feature the problems associated with "Big Data," which is typically characterized by the three "V's": volume, velocity, and variety. Volume refers simply to the size of such data sets, and velocity, to the speed with which streaming services can supply new data. Variety describes the heterogeneity of the data that might appear together in a common context. This brief description outlines the challenges facing the data analysis procedures that will be needed in the future.

2 Data Quality and Informational Content

The enormous amounts of existing and newly arising data remain relatively useless, unless we succeed in discovering new connections and knowledge within it. This is the main task of data mining and statistical learning theory, fields that have provided a multitude of algorithms for diverse scenarios (see [1] and [14]). Despite the existence of these methods and the software tools that accompany them, their use in the context of industrial production processes, for example, has not yet caught on widely. As shown in a joint project entitled "Supporting Decisions in Production Using Data Mining Tools," carried out by a consortium consisting of the ITWM, other Fraunhofer institutes, and representatives from the manufacturing industry, the disproportionately large adaptation efforts required for heterogeneous production domains and communication structures often cause significant difficulties. The lack of real-time capability for many of the analysis procedures also plays an important role here.

Generally speaking, especially in the context of dynamic systems, not all arbitrarily measured combinations of system inputs and outputs contain enough information in and of themselves to allow for complete identification of the system dynamics and generation of a corresponding system model. Discussions with customers from the manufacturing industry have consistently revealed that, although the adjustment and equipment parameters, for example, may indeed be highly temporally resolved, the product qualities to which they are assigned are only sampled randomly on a coarse time schedule. And there is another factor. Because the determination of these quality characteristics is often not automated, but performed manually in the lab, there are also long time delays

before the data becomes available. Taken as a whole, this often means that the potential of high-resolution input data can only be realized in a limited way for modeling product quality.

For successful, data-based system identification, it is also crucial to have data from different operating points and/or different dynamic excitation states. Otherwise, the resulting system models are only valid within a very limited area and are usually not suitable for use in subsequent optimization or control approaches. The most informative generation of process data is methodologically supported by the design of experiments (DOE) framework, which seeks to achieve the largest possible variance reduction in the model parameters being estimated by means of the smallest possible number of suitably chosen measurement points. In our projects, however, we regularly run up against technical or economic limits regarding specifications in the experimental design about the amount of data to be collected and the selected process points. The insertion of appropriate filters to protect against technically impossible parameter combinations is very helpful, but, for reasons of complexity, is usually only partly feasible. It should also be noted that the experimental design only delivers explicit formulas for determining the system input settings for models that are linearly dependent on their parameters. For nonlinear dependencies, no generally valid formulas can be specified in advance. Instead, the DOE plans themselves depend on the results of the executed measurements and are of an iterative nature.

In the life sciences–for example, when considering the expression patterns of the more than 20 000 human genes–there is also often a multitude of potential influencing factors that might explain a specific disease. However, one has only a small number of patients available who have been classified and analyzed.

Another crucial point in the evaluation of data quality is the proportion of disturbances contaminating the observed data. Particularly with measurement data, there is always contamination of this kind caused by the measurement-principle-dependent characteristics of the sensors being used. If the characteristics of the processes generating the disturbances are known with sufficient precision, then they can be modeled explicitly, and this model can be used to correct the data for the impact of the disturbances. In practice, however, one is often dealing with the simultaneous overlapping of several disturbance sources, and the resulting complexity often makes mechanistic modeling impossible. Instead, one describes the disturbances as the result of stochastic processes, which can be characterized by the appropriate distribution information. The frequently made assumption that this data follows a normal distribution can indeed be justified in many situations, due to the law of large numbers. There are, however, very many technical and biological questions for which this assumption is false. Nonetheless, many well-established procedures presume a normal distribution, along with the linearity of the underlying data-producing process dynamics. If one generalizes these assumptions, for example, in the field of state and parameter estimation, one then moves from the well-known Kalman filter based methods to the sequential Monte Carlo approach. This is a method that has been actively pursued for several years in the System Analysis, Prognosis, and Control Department in its work

with particle filters (see also "The Research—Robust State Estimations of Complex Systems").

In many application cases, it is not just disturbances in the data that cause difficulties. Often, the observed data sets are also incomplete, that is, some entries are missing. The values of some data sets may also be many times higher than the level of comparable data sets. The correct treatment of these defects and outliers, which can be caused by damaged sensors, for example, plays a decisive role in dealing with industrial data sources.

3 Data Integration and Pre-processing

The selection and allocation of suitable information-bearing quantities is crucial for the successful use of data analysis methods. In many industrial cases, this data is not to be found initially all together in some data warehouse, easily accessible to analysis. It is more likely to be distributed across different sources. Here, the spectrum runs from diverse databases to ASCII and/or Excel files to other, application-specific data formats. Occasionally, it still happens that certain data is only available in paper form and must first be digitalized. One initially looks for opportunities to extract the relevant data from all sources and bring it together into a higher-level data structure. Here, there are often problems in correctly assigning the data sets. In addition to solving these problems, one must also find suitable treatments for other incompatibilities, such as differing sampling rates among sensor data. Organizational challenges can arise when the needed data is distributed among different spheres of responsibility within different departments of a company.

As mentioned in the previous section, data sets are generally incomplete. One can almost always count on finding discontinuities and outliers. There are various procedures for identifying and adequately managing such problem cases, which must be chosen and executed according to the situation.

Along with integrating the data, one generally also subjects it to a normalization process and, possibly, a disturbance correction. Here as well, there are many procedures available for these work steps. In general, however, our project experience has taught us that, from the perspective of data analysis, it is desirable to retain as much control as possible over the entire chain of data processing steps. In accordance with this goal, one should always try to obtain data from project partners in its "rawest" form.

Moreover, to optimally select the next processing steps, it helps to first gain an overview of the data distribution. Especially for highly dimensional problems, one will make decisions on dimension reduction on the basis of the data's correlation structure and remove strongly correlated quantities from further consideration. In many cases, it also makes sense to execute the subsequent modeling steps not on the basis of the original data, but to draw upon compressed features instead. A well-known example of this is the principal component analysis, in which the original data is projected onto those sub-spaces

that explain the largest portion of variance in the data. If the corresponding background information is available, one attempts in this step—in the manner of grey box modeling—to transfer this knowledge into a set of appropriate features. For more on this topic, see Sect. 5.

4 Data-Based Modeling

In almost all mathematical modeling questions arising from practical applications, the existence of an adequate amount of real, measured data plays a decisive role in the success of the model design. Depending on the type of modeling, however, the requirements for the quantity and information content of the needed data fluctuate markedly. With so-called white box modeling, in which the model design is strongly guided by the explicit implementation of physical, biological, or economic laws, the data requirements are rather moderate and serve primarily scaling and calibration purposes. In contrast, so-called black box approaches assume purely data-driven modeling, with correspondingly high requirements on the quantity and information content of the available data. With so-called grey box modeling, a hybrid form of knowledge-driven and data-driven modeling, the data requirements lie somewhere in between. For the remainder of this chapter, we will be concerned primarily with questions of purely data-driven modeling. For further discussion of white box and black box modeling, refer also to "The Concepts—Modeling."

Data-driven modeling approaches come into consideration primarily when sufficiently informative measurement data is available and the interrelations and dynamics of the observed systems or processes resist explicit description due to their complexity. Two examples here are the extrusion of plastic components, including variation in the material recipe, and the crash behavior of carbon-fiber composite materials.

In general, data mining includes procedures with which relevant information can be extracted from complex data. Here, statistical learning methods model the data as results from random experiments. This perspective makes it possible to derive, verify, and better understand procedures for gaining information on the basis of statistical theory and intuition.

Statistical learning has a great deal in common with machine learning. With complex data, statistics must rely on appropriate, computationally intensive learning algorithms. Conversely, the statistical perspective in machine learning often allows one to understand when and why data analysis algorithms function and how they can be extended.

An important distinction of data mining problems lies in the type of data being observed. So-called structure-describing procedures, such as regression and classification, are normally confronted with the problem of approximating a target quantity Y (output, dependent variable) as accurately as possible using a function of the input quantity U (input, independent variable, predictor). The data forms a *random sampling* or *training set* $(U_1, Y_1), \ldots, (U_N, Y_N)$ of input variables U_j, together with the output variables Y_j. When learning the connections between input and output, one can therefore judge and optimize the system's performance on the basis of correct, observed values Y_j. In this case, one speaks of *supervised* learning.

With so-called structuring problems, in contrast, one has only input data $U_1, \ldots, U_N$, in which one wishes to identify structures such as clusters or low-dimensionality. Because there are no output variables that can serve as starting points for correcting errors in the learning results, this is also described as *unsupervised* learning. The features U_j are generally high-dimensional, and their structures usually cannot be simply visualized. Graphically representable projections onto two or three coordinate dimensions do not typically show the structures of interest. To make cluster formation or low dimensionality graphically visible, one must identify the most informative projections possible for this data.

5 Unsupervised Learning

With unsupervised learning, the focus is on characterizing the distribution and structure of the existing data. Along with observing standard quantities from the descriptive statistics, one is especially interested in discovering clusters and low-dimensional structures in the data. Here, there is also a strong overlap with the goals of data pre-processing, and unsupervised learning is therefore often used as a preparatory step in supervised learning problems.

One class of structuring problems arising in practice contains so-called variant management problems. Here, the input data describes the composition of complex products, such as commercial vehicles, for example, on the basis of their structural components. The goal is to find a sensible way to structure the product space, as defined by the customers of the associated company by means of the purchased products.

Here, the space should be approximated by the smallest possible number of representative products. This then allows one, in a subsequent step, to derive a plan for revising and reducing the necessary component spectrum and thus, decreasing inventory costs. The so-called cluster analysis is one method suitable for working on this question.

5.1 Cluster Analysis

One considers a finite set U of objects, each of which is described by the characteristics $U_1, \ldots, U_m$ of a number of attributes. The central prerequisite for the grouping of data is the existence of a dissimilarity or distance measure $d : U \times U \to R^{\geq 0}$, which permits measurement of the similarity between two objects; the larger the value $d(U_i, U_j)$, the more dissimilar are the objects U_i and U_j. In the cluster analysis, the goal is now to decompose the finite set U into pairwise disjoint groups or clusters $C_1, \ldots, C_r$:

$$U = \bigcup_{i=1}^{r} C_i, \quad C_i \bigcap C_j = \emptyset, \text{ for } i \neq j.$$

Such a decomposition is also called a partition of U. Each two objects within a cluster should be as similar as possible, whereas two objects from different clusters should be highly dissimilar. There are numerous algorithms for determining an optimal partition of U, which differ in search strategy and in the data types permissible for the features. The algorithms themselves frequently need specifications for the values of control parameters, such as the number of clusters to be sought, the minimal number of elements in a cluster, or the minimum dissimilarity between the objects of different clusters. Some algorithms also assume the specification of a start partition. This multitude of choices militates in favor of an external evaluation of the result partitions (in contrast to an evaluation within the algorithm regarding optimality) [8]. By comparing the results of a cluster algorithm for different parameter settings or start partitions, one can draw conclusions about, among other things, the stability of a result partition, the optimal number of clusters, and the coarse structure of the similarity space (U, d). The comparison of partitions can itself be accomplished by means of a distance measure

$$D : P(U) \times P(U) \to R^{\geq 0}$$

which is defined on the set $P(U)$ of all partitions of the set U. Such measures have been used for many years in biology and the social sciences. One possibility for comparing partitions is the information variation introduced in [11], which represents a metric based on an entropy approach.

5.2 Feature Selection

During the process of preparing a data-based regression model, the choice of which features one uses to build up the model is crucial. In our experience, this decision is significantly more important for successful modeling than the choice of a special model class. Although individual input quantities can be used as features, in many cases, one relies instead on the functional linking of different input quantities. Clues as to how one arrives

at the definition of the most information-rich features often come in the form of problem-specific expertise. Our project experience has shown us that these clues should definitely be followed. This helps to turn the original black box modeling at least partially grey.

Particularly in cases where there are no application-specific clues about feature definition, there is indeed in many applications the problem of a disparity between the high dimension of the input space and the relatively small number of existing input-output data pairs. Here, a dimension reduction is necessary, and one often carries out a principal component analysis of the input data. Restricting oneself to the principal components assigned to the largest singular values then delivers a corresponding subspace that is defined by the selected principal components. Another advantage of this approach is that the transformed data is uncorrelated and thus, in the case of normally distributed data, is even independent. Data that is given as a linear mixture of independent, arbitrarily distributed data sources can be decomposed into independent individual components using entropy based methods, such as independent component analysis (ICA) [9]. Entropy-based measures for quantifying the independence of two random variables, such as Mutual Information, are also often suitable for evaluating the explanatory power of a feature or a collection of features with regard to a given output quantity. On the basis of corresponding ranking criteria, one can then derive a variety of selection strategies for building up information-rich feature sets.

6 Supervised Learning

In the remainder of this section, we will consider supervised learning on the basis of input-output pairs (U_j, Y_j), $j = 1, \ldots, N$, which are modeled as independent and identically distributed (i.i.d.) realizations of random variables. For the sake of simplicity, we will only look at the case in which Y_j is one-dimensional. In contrast, the features U_j used to predict Y_j are typically highly dimensional in data mining. (U, Y) stands for a representative input-output pair that has the same distribution and is independent from the observed data.

The goal of the learning is to find a mapping f, so that $f(u)$ approximates or predicts "as well as possible" a new value Y, when the associated input value $U = u$ is known. In order to refine this, a loss function $L_f(u, y)$ is specified that measures the quality of the approximation. The most widely used loss function for regression problems is the quadratic forecasting error $L_f(u, y) = (y - f(u))^2$.

Statistical learning now attempts to find a classification or prediction function $f(u)$ that delivers a good approximation on average, i.e., for which the expectation value $R(f) = \mathrm{E}L_f(U, Y)$ is as small as possible. For regression problems with quadratic loss functions, the optimal prediction is

$$f(u) = m(u) = \mathrm{E}\{Y \mid U = u\}$$

of the *conditional expectation value* of Y, given that $U = u$ is known. Because the distribution of the data (U_j, Y_j) is unknown and can be quite arbitrary, the conditional expectation value $m(u)$, in practice, cannot be calculated. The goal of statistical learning is therefore to use the data to calculate approximations or *estimators* for this optimal function.

One demanding regression problem from the area of production is the quality prognosis of extruded plastic components. During extrusion, a mixture of plastic granules and other raw materials is melted in an extruder under the influence of temperature and pressure and, with the help of the extruder screw, pressed through an application-dependent mold. Such processes are used to manufacture window frame stock and insulation sheets, for example. Here, one is interested in the functional dependency of the thermal conductivity coefficient and the compressive strength of the extruded insulation sheets on the starting recipe and the settings of the equipment parameters, as well as on the various temperature zones along the extruder and the rotation speed of the extruder screw. Due to the complexity of the dependencies, an explicit modeling of the interactions is futile, and one resorts instead to historical production data and regression methods. The identified transformations then serve as the starting point for a subsequent process optimization by means of suitable Pareto optimization methods. For more, see "The Concepts—Optimization Processes".

A further example of a complicated regression problem from business economics is the calculation of the expected residual value of a leasing vehicle according to the specified duration of the contract. The value depends on numerous predictor variables, such as distance driven, vehicle model, engine, color, diverse equipment options, vehicle age, etc. If one knows the dependence of the residual value on this vehicle data, then one can estimate the capital value of the leasing inventory, plan future equipment packages so as to optimize the residual value, and so on. A similar regression problem is estimating the value of a house as a function of square footage, lot size, roof style, location, number of separate apartments, age and condition of the house, etc. What we are looking for is a forecasting function that predicts the price obtainable on the market as a function of all this data. In addition to providing support for specific purchase and sales decisions, this value information also plays an important role in appraising and mortgaging larger real estate projects.

In addition to regression problems for which the target quantity is continuous, so-called classification problems are also of practical importance. Here the Y_j only assume values in a finite set $\mathcal{K}$, which, for the sake of simplicity, correspond to the numbers $1, \ldots, K$ of the K classes. Figure 1 shows an example of a classification problem for two classes that is not separable by a linear classifier, but by a nonlinear one. Classification problems can be represented mathematically as special regression problems and thus will not be treated as a topic in their own right in the following discussion.

A challenging classification problem from economics is automatic fraud detection within the very large number of invoices that contracting firms submit to a company. On the basis of extensive information about the accounting data, such as amount and scope of the individual items, the identity of the invoicing party, etc., one uses statistical learning to decide whether there are any grounds for suspecting fraud and whether the invoice must therefore be examined more closely. An everyday example for the use of statistical-learning-based classification procedures are the spam filters in email accounts, which decide, on the basis of a large set of features, whether an incoming item is spam or a genuine email.

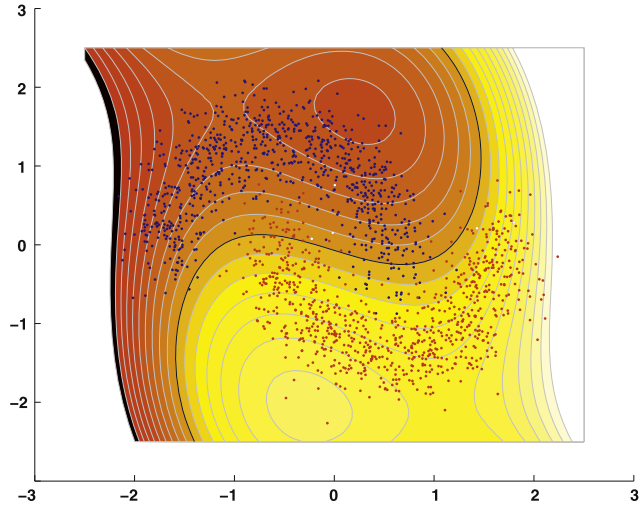

Fig. 1 Nonlinear classification problem: class 1: *blue*, class 2: *red*

A representative classification problem from the field of bio-informatics is identifying a so-called biomarker for a particular disease within a set of gene expression data. In other words, one searches for genes whose common expression pattern is characteristic for the presence and severity of the disease in question. If such a biomarker is found, then it can be used to manufacture disease-specific test kits, which allow one to quickly verify the presence of the disease.

6.1 Non-parametric Regression

If one defines the residuals $\varepsilon_j = Y_j - m(U_j)$, $j = 1, \ldots, N$, then they have conditional expectation values $\mathrm{E}\{\varepsilon_j \mid U_j = u\} = 0$. That is, U_j contains no information about which average value ε_j will assume. One usually assumes that the ε_j are i.i.d., which means that the following standard model of non-parametrical regression [3, 6] applies for the data:

$$Y_j = m(U_j) + \varepsilon_j, \quad j = 1, \ldots, N, \ \mathrm{E}\varepsilon_j = 0, \tag{1}$$

where $U_1, \ldots, U_N$ are i.i.d. and independent from the likewise i.i.d. $\varepsilon_1, \ldots, \varepsilon_N$. Moreover, one also usually assumes that the residuals possess a finite variance: $\mathrm{var}\,\varepsilon_j < \infty$.

In contrast to classical regression analysis, where the regression function $m(u)$ is assumed to be known except for a few parameters, non-parametric regression, and thus statistical learning as well, does not need these restrictive pre-requisites. Weak regularity assumptions about $m(u)$, such as twice continuous differentiability or quadratic integrability with respect to the distribution of U_j, are sufficient. The estimation procedure makes it possible to use the data to "learn" a predictive function that is largely unknown at the start.

Non-parametric regression approaches are not restricted to the standard model (1). For example, the residuals ε_j can also depend on the independent variables U_j. One example is the heteroscedastic regression model

$$Y_j = m(U_j) + \varepsilon_j = m(U_j) + \sigma(U_j)\eta_j, \quad j = 1, \ldots, N, \tag{2}$$

with i.i.d. η_j, for which $\mathrm{E}\eta_j = 0$ and var $\eta_j = 1$. Here, it is not only the average, but also the variability of Y_j that depends on U_j. The term $\sigma^2(u)$ is the conditional variance $\mathrm{var}\{Y_j \mid U_j = u\}$ of Y_j, given that $U_j = u$, and it can also be estimated using the same procedures as for $m(u)$.

An important class of problems that one repeatedly encounters in practice is characterized by dynamic developments in the target quantity over time. The above methods are also used in the corresponding non-parametrical time series analysis; one merely abandons the assumption that the U_j are independent. For example, if one sets $U_j = (Y_{j-1}, \ldots, Y_{j-p})$, then the result is a non-parametrical auto-regression model

$$Y_j = m(Y_{j-1}, \ldots, Y_{j-p}) + \varepsilon_j, \quad j = 1, \ldots, N,\ \varepsilon_1, \ldots, \varepsilon_N \text{ i.i.d. with } \mathrm{E}\varepsilon_j = 0.$$

In this case, the auto-regression function m delivers the best prediction of the value Y_j of the time series at time j, using the last p observations $Y_{j-1}, \ldots, Y_{j-p}$, inasmuch as the average quadratic prediction error is minimized. Correspondingly, one obtains non-parametrical versions of the ARCH models from (2), which play an important role in risk measurement in financial statistics.

6.2 Empirical Risk Minimization

The predictive function $m(u) = \mathrm{E}\{Y_j \mid U_j = u\}$ minimizes the expected loss $R(f) = \mathrm{E}(Y - f(U))^2$ (also known as *risk*) relative to f. With empirical risk minimization, in order to estimate $m(u)$, the risk is first estimated from the data, taking reference here to the law of large numbers, by

$$\widehat{R}(f) = \frac{1}{N}\sum_{j=1}^{N}\big(Y_j - f(U_j)\big)^2. \tag{3}$$

Depending on the application, other loss functions might be more suitable, such as the L^1-risk, as defined by adding the absolute deviations of the amounts. Particularly for multi-dimensional target quantities, the search for an optimal loss function is commensurately complex. One must also consider that many prominent learning algorithms take advantage of the special characteristics of a quadratic loss function, in particular for the derivative formation. Thus, one must assume that there are significantly fewer suitable learning algorithms for more general loss functions. Particularly with classification problems, one

is also often dealing with the kinds of problems for which the costs caused by a misclassification depend on the original class affiliation; that is, they are often particularly non-symmetric. Let us consider here a healthy person who is incorrectly classified as sick, and a sick person who is classified as healthy. While in the former case, a superfluous therapy is prescribed that is possibly accompanied by quite unpleasant side effects and unnecessary monetary costs, in the latter case, a possibly life-saving treatment is withheld from a sick person who requires it to survive. Arriving at a loss function that accurately reflects the characteristics of the problem under investigation and that can also be efficiently minimized is, in many cases, a key milestone in a successful data-based modeling endeavor.

One then attains an estimator for m by minimizing the empirical risk $\widehat{R}(f)$. Minimizing across all measurable functions, or even merely across all twice continuously differentiable functions, leads to a function $\hat{f}$, however, that interpolates the data, that is, $Y_j = \hat{f}(U_j)$, $j = 1, \ldots, N$. Such a solution is unserviceable for use in predicting future data, since it models exactly the random disturbances ε_j in the collected random samples, instead of adequately reflecting the general form of dependency between the random quantities U and Y.

There are three strategies that allow empirical risk minimization to circumvent this problem:

- Localization, that is, restricting the averaging in the empirical risk to those U_j lying in the neighborhood of that point u, at which one wants to estimate $m(u)$;
- Regularization, that is, imposing variation limitations on f that rule out interpolating solutions;
- Restricting the set of functions across which (3) is minimized, which leads to the class of *sieve estimators*.

In the following sections, we will discuss important further aspects and implementations of these strategies.

6.3 Local Smoothing and Regularization

The idea of local smoothing for the estimation of a largely arbitrary regression function $m(x)$ can be derived directly from the law of large numbers: when $Y_1, \ldots, Y_N$ i.i.d., with expectation value $\mathrm{E}Y_j = m_0$, then the random sample average for $N \to \infty$ converges almost surely toward m_0:

$$\frac{1}{N} \sum_{j=1}^{N} Y_j \xrightarrow[\text{a.s.}]{} m_0.$$

If, in regression model (1), $m(u)$ is smooth–for example, twice continuously differentiable—then m is approximately constant within a small neighborhood around u. This means that, for small $h > 0$

$$m(z) \approx m(u), \quad \text{when } \|z - u\| < h. \tag{4}$$

If one now averages only those observations Y_j in the neighborhood of u, that is, with $\|U_j - u\| < h$, then, for all $\mathrm{E}Y_j \approx m(u)$, that is, for large N,

$$\hat{m}(u,h) = \frac{1}{N(u,h)} \sum_{j=1}^{N} 1_h\big(\|U_j - u\|\big) Y_j \approx m(u), \quad \text{with } N(u,h) = \sum_{j=1}^{N} 1_h\big(\|U_j - u\|\big) \tag{5}$$

in which $1_h(z) = 1$ for $-h \le z \le h$, and $= 0$ otherwise. $N(u,h)$ is the number of observations in the neighborhood of u. Local smoothing of the data, that is, averaging of the data in the neighborhood of u, delivers a convenient estimator for $m(u)$. One obtains a convergence of $\hat{m}(u,h)$ towards $m(u)$ for one-dimensional U_j, for example, for $N \to \infty$, $h \to 0$ and $Nh \to \infty$.

The local averaging is based on assumption (4), for $z = U_j$, an assumption that becomes better and better as the distance between U_j and u decreases. This suggests therefore the idea of weighting the contribution of Y_j to the local averaging according to how closely U_j lies to u. Instead of a simple average, one then obtains a weighted local average. One example of this is the *kernel estimator*, in which the weights are generated by a function $K(u)$ known as a *kernel*. Typical choices for K are probability densities, that is, $K(u) \ge 0$ and $\int K(u)du = 1$.

With a simple local average (5) and, in general, with kernel estimators, the bandwidth h determines the size of the area used for local averaging. This leads to problems in estimating $m(u)$ when there are only a few observations U_j in the neighborhood of u. Therefore, drawing on the same insight, *k-nearest-neighbor estimators* do not average across a fixed neighborhood surrounding u. Instead, they average across a fixed number k of data points. Those data points Y_j are chosen for which U_j lies closest to u, that is, the averaging is performed across the k nearest neighbors to u.

At first glance, *regularization estimators* appear to follow an entirely different approach than localized smoothing procedures for ruling out interpolation when minimizing an empirical risk. In (3), $\widehat{R}(f)$ measures how well the function values $f(U_j)$ fit to the observations Y_j. In order to avoid over-fitting, an auxiliary condition $r(f) \le c$ is placed on the minimization of $\widehat{R}(f)$, where $r(f)$ is a measure for the variation of the function f. As a result, when N is large, the strongly-fluctuating interpolating functions or nearly interpolating functions are ruled out as solutions. For some regularization estimators, an asymptotic equivalence to special kernel estimators can be shown (see [10] and [13]). Because the latter allow for a simple asymptotic theory, corresponding distribution approximations

can be transferred to the regularization estimators and used for hypothesis tests and the calculation of confidence intervals and quantiles.

A recognized problem with the use of local smoothing procedures, one that, unfortunately, arises frequently with applications relevant to practice, is the so-called "curse of dimensionality." When these procedures are applied in the direct form described here for input spaces U with high dimension d, then, except for extremely large random samples, the neighborhoods determined by h are almost empty for many values of u. As a result, the random error is not averaged out. With k-nearest-neighbor estimators, the design does indeed ensure that averaging is always performed across k values. But here, when the number of dimensions is large, the adaptively selected neighborhoods are necessarily very large. This corresponds to the choice of a very large bandwidth h for kernel estimators, which leads to a systematic distortion of the estimator.

Especially when working on attractors of nonlinear dynamic systems that have been reconstructed using phase space methods, the above next neighbor methods can often be used successfully. Indeed, here, we have relevant project experience in connection with the risk evaluation of electrocardiogram data. In this context, the dimensions d being observed are small to medium in size, and there is a relatively large data set. Nevertheless, it is definitely advisable to use efficient procedures when searching for each of the nearest neighbors; a naive implementation quickly reaches its performance limits. See [7], also.

6.4 Sieve Estimators

Sieve estimators dispense with localization or regularization as a means of avoiding overadaptation or, worse, interpolation of the data. Instead, they achieve this by restricting the function class across which the empirical risk (3) is to be minimized. In order to still achieve the necessary flexibility and avoid limiting assumptions about the estimated function $m(x)$, the function class $\mathcal{F}_N$ being considered here grows with the random sample size N. A sieve estimator therefore solves the minimization problem

$$\min_{f \in \mathcal{F}_N} \widehat{R}(f).$$

To ensure that the resulting function estimator $\hat{m}_N(x)$ converges to $m(x)$, the function classes $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \cdots$ must possess a *universal approximation characteristic*. That is, for each regression function m being considered, there must be a suitable N and an $m_N \in \mathcal{F}_N$, so that m_N approximates the function m with sufficient accuracy. There are various possibilities for refining this requirement. The function classes $\mathcal{F}_N$ are typically parametric, that is, they contain only functions that have been specified, except for a single parameter $\theta \in R^p$. Actually, just as in classical statistics, one adapts a parametric model to the data, but allows the model to be mis-specified. That is, one allows the function

m being estimated to lie outside of $\mathcal{F}_N$. The non-parametric consistency of the procedure is achieved by allowing the parameter dimension $p = p(N)$ to grow as the number N of data grows. Next, we will briefly discuss the three most important function classes.

As a starting point for designing the function classes $\mathcal{F}_N$, one often resorts to *series expansions* relative to orthogonal basis functions. Accordingly, the number of summands then depends on N. Sieve estimators can also be derived for non-orthogonal functions, provided that a universal approximation characteristic applies. In order to guarantee the stability of the estimator, one usually carries out an additional regularization of the coefficients of the series expansion. For corresponding convergence results, see [3].

The starting point for *partition estimators* is a disjoint decomposition of the domain of the input variables. Each of the estimators is then constant on each set of this partition, and the corresponding values are calculated as the average of the observations lying within the set. If the partitions become finer and finer as N grows, then $\mathcal{F}_N$ possesses the universal approximation characteristic. A data-adaptive choice of partitions is advantageous. In many cases, tree-based methods are used here, and the corresponding estimators are then called *classification* or *regression trees*. See [2]. These approaches are useful for practical applications requiring the estimator to be interpretable, such as is almost always the case in medical applications, for example. Here, in very rare cases, one accepts a black box whose decisions may indeed be correct, but cannot necessarily be explained or argued satisfactorily. In particular, rule bases for decision-making can also be derived directly from the classification trees. This allows the plausibility of this procedure to then be evaluated in discussions with experts in the application domain.

Neural networks (see [4, 5], and [12]), originally developed as models for signal processing in the brain, represent an important class of sieve estimators. The best known of these are the feed-forward networks. In addition to the input and output layers, these networks possess at least one nonlinear, hidden layer of so-called neurons. These lead with the *activation function* ψ to the following class of functions:

$$\mathcal{F}_N = \left\{ f(x) = v_0 + \sum_{k=1}^{H} v_k \psi \left(w_{0k} + \sum_{\ell=1}^{d} w_{\ell k} x_\ell \right); v_k, w_{\ell k} \in R \right\}$$

with the parameter $\theta = (v_0, \ldots, v_H, w_{01}, \ldots, w_{dH})' \in R^{(d+2)H+1}$. The classes $\mathcal{F}_N$ of output functions of feed-forward networks possess the universal approximation characteristic when the number H of neurons grows as a function of N. The practical success of neural networks is the result of the existence of fast algorithms, particularly the back propagation algorithm and suitable modifications [15], which allow the network parameters to be learned within an acceptable time, even for large data sets N. An important point for successfully learning the underlying dependencies in the given data is the selection of a neural network whose size is adapted to the informational content of the data. The next section describes approaches for doing this.

7 Data-Adaptive Complexity Selection

All non-parametrical regression estimators contain tuning parameters with which the variation or complexity of the function can be controlled. They are utilized to force the estimation procedure to adapt to an adequate description of the actual dependency structure between input U_j and output Y_j, instead of reproducing irrelevant random effects that are inconsequential for the prediction of future data. With kernel estimators, the tuning parameter is the bandwidth h; with next-neighbor estimators, it is the number k of neighbors; and with sieve estimators, it is basically the number of free parameters of the function class $\mathcal{F}_N$. There is a variety of procedures that allow for data-adaptive selection of these tuning parameters.

The choice of tuning parameters is closely connected with the bias-variance dilemma and the problem of finding a balance between over-adaptation (overfitting) to the data and insufficient adaptation (underfitting) to the data. If the estimator is allowed too much freedom, overfitting will result; the estimator $\hat{m}$ adapts itself not only to the desired function m, but also tries to model parts of the random error ε_j. Conversely, if the estimator is allowed too little freedom, the result is underfitting. Here, the variability of the function estimator $\hat{m}$ is indeed small, but it deviates systematically from the function m being estimated, since the bias $\mathrm{E}\hat{m}(u, h) - m(u)$ is large. Accordingly, it is also unsuitable for predicting future data.

The goal of the data-adaptive selection of tuning parameters is an estimator of the function m that is as good as possible and that delivers optimal predictions. The average estimation error should be as small as possible, but is unknown. Therefore, one generally proceeds by splitting the data into training data and validation data; the training data is then used to calculate the estimator and the validation data is used to compare different estimators with different tuning parameters or complexity. When there is only a small amount of data available, and the estimation quality suffers significantly because some of the data must be put aside for validation purposes instead of being used for the estimation, then the cross-validation approach can be used [6]. This approach uses the data more efficiently, but at the cost of appreciably higher computation time.

8 Concluding Remarks

Our experience with industrial data analysis questions shows that an application-specific problem formulation, combined with the selection of suitable data sources and the features derived from them, plays the central role. Here, as much expertise as possible from every application domain should be brought to bear on the modeling process. The success of the endeavor generally depends more on this expertise than on the choice of a special machine-learning procedure.

Nonetheless, in all cases, the quality and informational content of a given data set also implicitly set an upper limit to the maximum attainable quality for learning a dependency structure based on the data. Here, it is very important to suitably adapt the complexity of

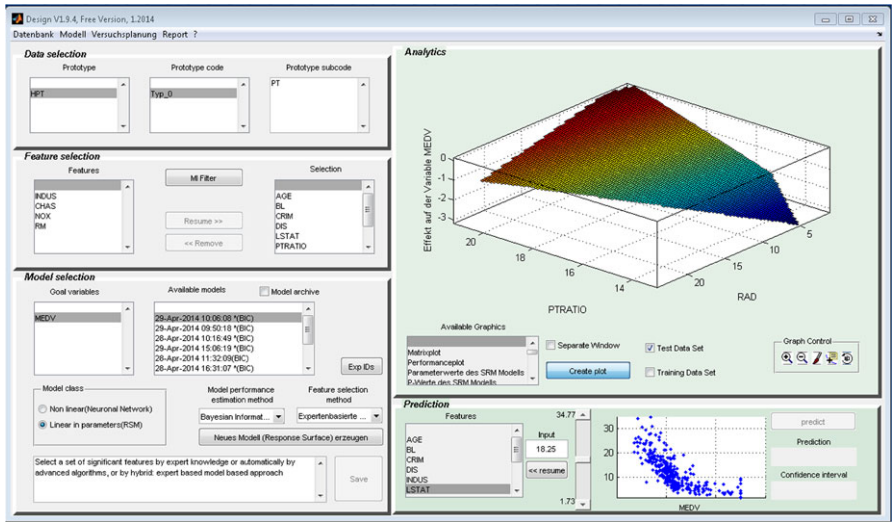

Fig. 2 Data mining platform "Design"

the chosen model approach to this informational content. Acknowledging and integrating any additionally available expertise and domain-specific knowledge is always beneficial.

To promote acceptance of data mining procedures in industry, it is important, on the one hand, to supply high-performance algorithms that take into account the corresponding requirements and restrictions regarding run-time or data volume. At the same time, it is also crucial to support the user in selecting procedure parameters and interpreting and evaluating the results. Toward this end, we in the System Analysis, Prognosis, and Control Department have developed the analysis platform "Design" (Fig. 2). It can be easily adapted to diverse application contexts and data structures, and it contains a selection of effective machine learning algorithms. At the same time, however, it relieves the user of much of the work of setting critical procedure parameters.

References

1. Bishop, C.M.: Pattern Recognition and Machine Learning. Information Science and Statistics. Springer, Berlin (2008)
2. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. CRC Press, Boca Raton (1984)
3. Györfy, L., Kohler, M., Krzyźak, A., Walk, H.: A Distribution-Free Theory of Nonparametric Regression. Springer, Berlin (2002)
4. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning, Data Mining, Inference, and Prediction. 2nd edn. Springer, Berlin (2008)
5. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall, New York (1999)
6. Härdle, W.: Applied Nonparametric Regression. Cambridge University Press, Cambridge (1990)

7. Kantz, H., Schreiber, T.: Nonlinear Time Series Analysis. Cambridge University Press, Cambridge (1997)
8. Knaf, H.: Distanzen zwischen Partitionen – zur Anwendung und Theorie. Technical Report 226, Fraunhofer ITWM, Kaiserslautern (2013)
9. Lee, T.W.: Independent Component Analysis: Theory and Applications. Kluwer Academic, Norwell (2000)
10. Linton, O., Härdle, W.: Nonparametric regression. In: Banks, D., Kotz, S. (eds.) Encyclopedia of Statistical Science, vol. X. Wiley, New York (1998)
11. Meila, M.: Comparing clusterings – an axiomatic view. In: Proceedings of the 22nd International Conference on Machine Learning, vol. 84, pp. 1003–1013 (2005)
12. Montavon, G., Orr, G., Müller, K.R.: Neural Networks: Tricks of the Trade. 2nd edn. Lecture Notes in Computer Science, vol. 7700. Springer, Berlin (2012)
13. Silverman, B.: Spline smoothing: the equivalent variable kernel method. Ann. Stat. **12**, 898–916 (1984)
14. Vapnik, V.N.: Statistical Learning Theory. Adaptive and Learning Systems for Signal Processing, Communications and Control. Wiley, New York (1998)
15. White, H.: Some asymptotic results for learning in single hidden-layer feedforward network models. J. Am. Stat. Assoc. **84**, 1003–1013 (1989)