Quantitative Externalization of Visual Data Analysis Results using Local Regression Models

Krešimir Matković[1], Hrvoje Abraham[2], Mario Jelović[2], and Helwig Hauser[3]

[1] VRVis Research Center, Vienna, Austria, `Matkovic@VRVis.at`
[2] AVL-AST d.o.o., Zagreb, Croatia, `{Hrvoje.Abraham|Mario.Jelovic}@AVL.com`
[3] University of Bergen, Norway, `Helwig.Hauser@UiB.no`

Abstract. Both interactive visualization and computational analysis methods are useful for data studies and an integration of both approaches is promising to successfully combine the benefits of both methodologies. In interactive data exploration and analysis workflows, we need successful means to quantitatively externalize results from data studies, amounting to a particular challenge for the usually qualitative visual data analysis. In this paper, we propose a hybrid approach in order to quantitatively externalize valuable findings from interactive visual data exploration and analysis, based on local linear regression models. The models are built on user-selected subsets of the data, and we provide a way of keeping track of these models and comparing them. As an additional benefit, we also provide the user with the numeric model coefficients. Once the models are available, they can be used in subsequent steps of the workflow. A model-based optimization can then be performed, for example, or more complex models can be reconstructed using an inversion of the local models. We study two datasets to exemplify the proposed approach, a meteorological data set for illustration purposes and a simulation ensemble from the automotive industry as an actual case study.

Keywords: Interactive Visual Data Exploration and Analysis, Local Regression Models, Externalization of Analysis Results

1 Introduction

In the currently evolving information age, both data exploration and analysis become increasingly important for a large variety of applications and both interactive visualization as well as computational methods (from statistics, machine learning, etc.) establish themselves as indispensable approaches to access valuable information in large and complex datasets. With interactive visualization, the analyst is included in the knowledge crystallization loop and thus also open-ended and ill-defined exploration and analysis questions can be investigated, often also on the basis of data with certain deficiencies (noise, errors, etc.). With computational data analysis, exact quantitative results can be achieved, based on advanced and fast algorithms that also often are completely automated. In visual analytics, one key question is whether we can successfully combine both

approaches to integrate the mutual advantages in hybrid solutions, based both on interactive visualization and on computational data analysis.

One special challenge with interactive visual data exploration and analysis is the question of how to effectively and efficiently externalize valuable findings such that following steps in an application workflow can successfully build on them. Only very few works in visualization research [32, 27, 17] have so far focused on this question and suggested selected solutions. In particular the quantitative externalization of findings from qualitative interactive visual data analysis is genuinely difficult, while many workflows clearly would benefit from solutions that could pass on results in quantitative form—think, for example, of an analyst, who studies some relevant data curves in a graph view and wishes to use their inclination (quantitatively) in a subsequent work process.

In this paper, we now propose a new solution for quantitatively externalizing findings from interactive visual data exploration and analysis. We describe a method that enables the analyst to interactively measure certain data relations in a visualization. This is realized by locally modeling selected data relations of interest with a linear data model and then externalizing the model parameters from this process. For several reasons, most importantly including their stability properties and their simplicity, we focus on linear local models in this work— clearly, many other, non-linear models could be considered for this purpose, as well. While linear models often are too simple for global data approximations, they often provide good results locally. In order to fit the linear models locally to selected data, we use several different regression methods, depending on which of these methods achieves the best results. We present our solution in the context of a system with coordinated multiple views that enables such an externalization through interactive means.

In our solution, we assume the user to be involved in an iterative, interactive data exploration and analysis process. During the visual data drill-down, the user instantiates locally a linear modeling process of selected subsets of data. The corresponding model parameters are then returned back to the user in a quantitative form. Models and data are also shown together in the visualization. In this way, the user can easily interpret the findings, and, since the modeling results are available explicitly, rank the findings in order to choose those to use subsequently.

Already in 1965, John Tukey pointed out that combining the power of a graphical presentation with automatic computer analysis would enable more successful solutions [31]. Later, Anscombe [1] illustrated how important it is to also see the data, in addition to considering statistical measures. Nonetheless, a recent study by Kandogan et al. [11] explains that still data analysts do not regularly use visualization due to a lack of means to quantify analysis results.

The main contribution of this paper is thus not a new visual metaphor— we use standard views. Instead, we integrate solutions from machine learning into visualization (modeling by regression) in order to quantitatively externalize valuable findings from interactive data studies. We also suggest to keep track of the computed models and we provide a fast and intuitive way to instantiate new

models in the visualization. This way, a powerful combination of automatic and interactive data analysis is realized, combining valuable advantages from both approaches, i.e., the quantitative results from regression modeling, and the user-steered local modeling from the visualization. The quantitative externalization of otherwise qualitative results makes them easier to describe and rank, while the visualization is useful to spot and understand shortcomings and imprecisions of the automatically fitted models.

In this paper, we focus on complex data, which, in addition to scalar independent and dependent data, also contains families of curves, i.e., time-dependent attributes. We deploy a coordinated multiple views system, which supports on-the-fly data derivation and aggregation as an important basis for our approach. The interactive approach makes modeling very quick and efficient and also easier accessible for domain experts, who are not experts in machine learning or statistics. In the following, we first introduce the new approach along with a relatively simple meteorology example (for illustration purposes), before we then evaluate it informally based on an application case in the automotive industry.

2   Related Work

Our research is related to several fields. Interactive visual analysis (IVA) facilitates knowledge discovery in complex datasets by utilizing a tight feedback loop of computation, visualization and user interaction [13, 14, 29]. IVA provides an interactive and iterative data exploration and analysis framework, where the user guides the analysis [26], supported by a variety of computational analysis tools. The interactive visual analysis exploits human vision, experience, and intuition in order to analyze complex data. Tam et al. identify the potential of so called "soft knowledge", which is only available in human-centric approaches [28], including the ability to consider consequences of a decision and to infer associations from common sense.

The interactive exploration process is mostly qualitative. Recent research, however, focuses increasingly on quantitative aspects. Radoš et al. [24] structure the brushing space and enhance linked views using descriptive statistics. Kehrer et al. [12] integrate statistical aggregates along selected, independent data dimensions in a framework of coordinated, multiple views. Brushing particular statistics, the analyst can investigate data characteristics such as trends and outliers. Haslett et al. [6] introduce the ability to show the average of the points that are currently selected by the brush.

Lampe and Hauser [17] support the explanation of data by rapidly drafting, fitting and quantifying model prototypes in visualization space. Their method is related to the statistical concept of de-trending, where data that behaves according to a model is de-emphasized, leaving only the residuals (potentially outliers and/or other model flaws) for further inspection. Piringer et al. [23] introduce a system for the visual evaluation of regression models for simulation data. They focus on the evaluation of the provided models, while we focus on

the description of data relations by means of local regression models. We exploit on-the-fly data aggregation as described by Konyha et al. [15].

Shao et al. [25] present new research on combing regression modeling and interactive visual analysis. They build models based on selected subsets of data, as we do here, but they depict them on-the-fly during interaction. Neither do they provide a system for any house-keeping of models, or for the comparison of models. They also do only depict modeling results visually, while we provide models coefficients as well as quality-of-fit indicators.

In this work, we focus on an engineering example, while complex data is also common in other domains. Holzinger [7] introduces a concept of interactive machine learning for complex medical data, where a human-in-the-loop approach is deployed. The approach has been evaluated as a proof-of-concept study [8] and as a means to analyze patient groups based on high-dimensional information per patient [10].

In this paper, we make use of the common least squares, the Lasso, and the Huber regression models, described, for example, in standard literature on regression modeling [4].

3 Data Description and Problem Statement

In this paper, we focus on complex data in the form of records that contain different types of attributes. In contrast to the conventional approach, where attributes are scalar values (numerical or categorical), we also address complex attributes, i.e., curves (time-dependent attributes). Such a data organization is more natural for many cases in science and engineering.

We illustrate our approach based on a simple data set describing meteorological stations in the United States [21]. Global summaries per month are used, containing the statistics of 55 climatological variables. Each record corresponds to a single station with the following scalar attributes: longitude, latitude, elevation, state, and station name. Further, we also study two curve attributes: the mean temperatures per month throughout the year and the according mean precipitation values. Figure 1 illustrates the data. Figure 2 shows all stations as points in a scatterplot and temperature and precipitation curves in two curve views. The curve view depicts all curves plotted over each other. A density mapping is deployed and areas where curves are more dense can be seen, accordingly.

We differentiate independent from dependent attributes and some of our dependent attributes are curves. In our data model, the independent part of a data point is described as $\mathbf{x} = (x_1, \ldots, x_m)^\top$, i.e., a point in $\mathbb{R}^m$, and the corresponding dependent output part $\mathbf{y} = (y_1, \ldots, y_n)^\top$, i.e., a point in $\mathbb{R}^n$, where m can be seen as the number of control parameters and n as the number of outputs. This assumes that there is a function $\mathbf{S}$ that maps inputs to outputs. This function can be a numerical simulation or a measurement method:

$$\mathbf{y} = \mathbf{S}(\mathbf{x}) \tag{1}$$

ID	Latitude	Longitude	Elevation	State	station Name	Temp (t)	Prec (t)
11084	31.0581	-87.0547	25.9	AL	REWTON 3 SSE		
12813	30.5467	-87.8808	7	AL	AIRHOPE 2 NE		
13160	32.8347	-88.1342	38.1	AL	AINESVILLE LOCK		
...	...	...	...	...	...	...	...

Fig. 1. Structure of complex data, including also curves as attributes. Temperatures and precipitation values are stored as functions of time. The curves in the table are only symbolic, the actual curves have different shapes.

In the case of an ensemble of simulations or measurements, we then have a set of pairs: $E = \{(\mathbf{x}_j, \mathbf{y}_j)\}$. As indicated above, any y_i can also be a curve $y_i(t)$, given at a particular sequence of t-values.

Interactive visual analysis is a proven method for analyzing such data. However, if we want to quantify and compare results, we have to deploy quantitative analysis. If we, for example, assume that there is a correlation between the maximum yearly temperatures and the latitude of the weather station, we easily can show a corresponding scatterplot and see if there is such a relation. Figure 3 shows such a scatterplot. But how can we communicate our findings? And moreover, once we can quantify it, how can we compare it with other findings?

We thus propose to locally fit linear regression models for user-selected subsets of data, and then to return the model values to the user. This way, the findings are quantified and externalized. Accordingly, they can be also compared, the best ones can be identified and then used in subsequent tasks. In fact, there are many relevant application scenarios, where a model of the data (or a data subset) is needed. If a process has to be optimized, for example, a regression model can be very useful. Further, if we want to reconstruct our simulation or measurement, i.e., find inputs which correspond to a desired output, a corresponding linear model can easily be inverted and we thus can easily derive target input values. All these operations assume a model which is a good representation of the data (globally and/or locally). Our approach makes such analysis tasks possible, combining the best from interactive and from automatic data analysis.

4 Interactive Regression Modeling

We deploy linear regression models to quantify local analysis results. In order to build a regression model we first extract scalar aggregates from the curve attributes. The attributes of interest strongly depend on the analyst's tasks.
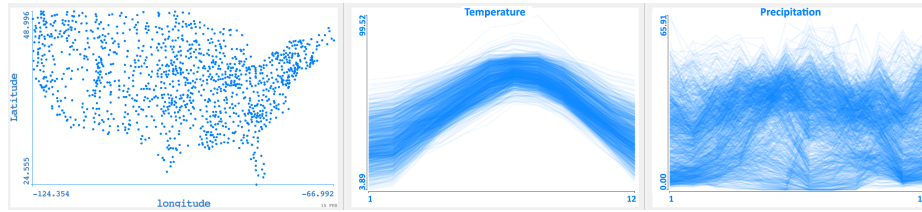
Fig. 2. A visualization of the illustrative, meteorological data. The scatterplot on the left mimics a map of the weather stations in the United States. The curve views show the monthly mean temperatures and precipitation values for each station. The temperature curves are quite similar in their shape (cold winters, warm summers), while the precipitation curves exhibit more variation.
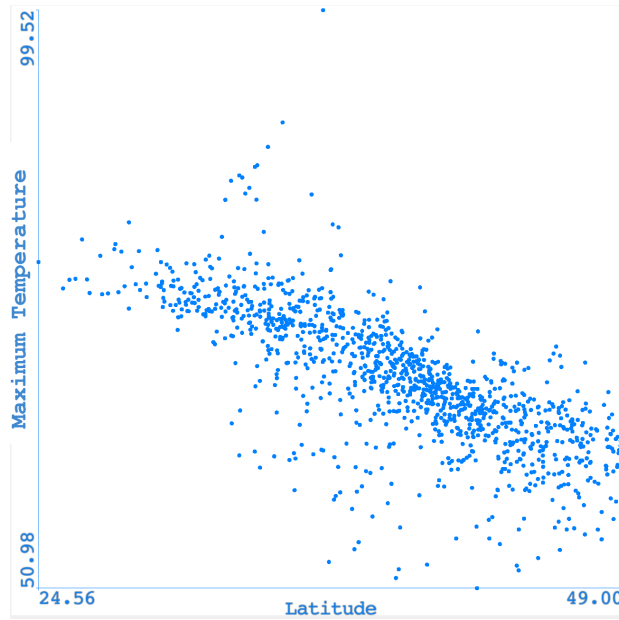
Fig. 3. A scatterplot showing the relation between latitude and the maximum monthly temperature value, i.e., the maximum of the temperature curves shown in Figure 2. As expected, southern stations have higher temperatures.

Accordingly, there isn't any predefined set of attributes which would be valid for all data sets and all cases, but the interactive, on-demand derivation of such aggregates proves useful instead [15].

In the following, we first summarize the models we use and then we illustrate the main idea using the meteorological data set and simple scalar aggregates. A more complex case which includes complex aggregates is described in the case study section.

4.1 Linear Regression Models

The most standard linear regression model that we use is the common least squares method, as proposed already by Legendre in 1805 [18] as well as by Gauss in 1809 [5]. Both applied it to astronomical observations in order to determine the orbits of planets around the Sun.

The objective function for a dataset with N M-dimensional inputs x_{ij}, output vector y_i and the regression coefficients vector $\mathbf{w} = (w_0, w_1, \ldots, w_M)^\top$ is

$$F_{LS}(\mathbf{w}) = RSS(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - w_0 - \sum_{j=1}^{M} w_j \, x_{ij} \right)^2. \tag{2}$$

This objective function is also known as the Residual Sum of Squares or just as $RSS(\mathbf{w})$.

Lasso regression, being in principle very similar to the least squares method, adds an additional constraint to the minimization of the objective function in order to limit the extent of the fitting coefficients:

$$F_{Lasso}(\mathbf{w}) = RSS(\mathbf{w}), \ \sum_{j=1}^{M} |w_j| \leq t. \tag{3}$$

It was named and analyzed in detail by Tibshirani in 1996 [30], after Breiman's influential paper in 1995 [2], introducing the *nonnegative garrote* method, and Frank & Friedman's 1993 paper [3], where the same constraint is considered as one form of the *generalized ridge penalty*, but without any analysis and results.

The Lasso-regularization is controlled via tuning parameter t and for a sufficiently large t the method is equivalent to the least squares approach. Generally, Lasso regression ensures a more stable result for some classes of base functions, such as polynomials, and it can be also used for feature selection as it tends to reduce the regression coefficients of less important inputs to zero (or close to 0). For this reason it is often used in the analysis of multidimensional data, machine learning, etc.

Another interesting property is that it also can be used to determine minimal models when the number of regression parameters is greater than the number of input cases, e.g., fitting a 10th degree polynomial to just 6 data points, a case in which the least squares method would just return one of many non-unique solutions (or none at all). It is important to notice, however, that the method is not scale-invariant, so the data has to be normalized in a certain way (*standardized*) to get useful results.

Huber regression follows a similar approach, but divides the residuals

$$r_i = y_i - w_0 - \sum_{j=1}^{M} w_j \, x_{ij} \tag{4}$$

into two classes: *small* and *big* residuals. For some given parameter δ, a quadratic function is used for *small* residuals ($|r_<|/\sigma \leq \delta$), and linear absolute values for *big* residuals ($|r_>|/\sigma > \delta$), where σ is determined during the minimization together with the regression coefficients w_j:

$$F_{Huber}(\mathbf{w}, \sigma) = \sum \left(\left(\frac{r_<}{\sigma} \right)^2 + \left| \frac{r_>}{\sigma} \right| \right). \tag{5}$$

This approach was introduced by Huber in 1964 [9] and it ensures that a small number of outliers does not have a big influence on the result (as they would if the quadratic form would be used for them, as well). Due to the reduced sensitivity to outliers, Huber regression belongs to the important class of *robust regressors*.

The fitting score of a regression model is measured by the determination coefficient R^2, defined as

$$R^2 = 1 - \frac{u}{v}, \quad u = \sum_{i=1}^{N}(y_i - z_i)^2, \quad v = \sum_{i=1}^{N}(y_i - \overline{y})^2 \tag{6}$$

where y_i, $\overline{y}$, and z_i are the dataset outputs, the mean of the outputs, and the model-predicted values, respectively. The highest possible score is 1. When the model returns just the output mean $\overline{y}$, the score is 0; and bad models get (arbitrarily) negative scores.

4.2 Interactive Modeling

It is essential to enable the user to interactively select scalar aggregates during the analysis. In our solution, it is anytime possible to compute new aggregates and to thereby extend the data table by additional synthetic data attributes. Often, it is not fully clear in the first place which aggregates indeed are most useful and all scalar aggregates that we describe in this paper were computed on the fly during the data exploration and analysis. This solution brings important flexibility and reduces the pressure on the analyst to define all necessary aggregates in advance. In the following, we illustrate our main idea by selecting three basic aggregates, i.e., the minimum, the maximum, and the mean of the temperature and of the precipitation curves. Accordingly, our data table then has six additional scalar columns.

A reasonably compact regression model, which successfully captures all data relations for all weather stations across the entire United States, relating longitudes and latitudes (as independent attributes) and the six scaler aggregates of the temperatures and the precipitation values (as dependent data), would be very challenging to construct (if possible at all). Also, one needs to assume that there are important additional factors with an influence on the temperature and precipitation values (like elevation, etc.). Accordingly, we simply dismiss the idea of creating a global model, in particular it is clear that we cannot expect to find a useful linear global model. Instead, we focus on local modeling of selected data
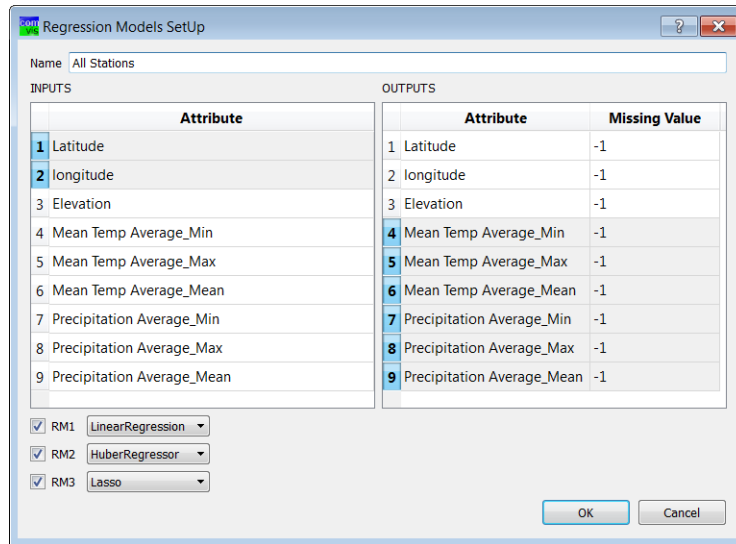
Fig. 4. A model is created for all data or for a subset which is specified by means of brushing. The user enters the name of the subset (*All Stations* here), and specifies which attributes are considered as independent (*Latitude and Longitude* here), and which are dependent. Further, the user can select up to three models to be built.

Fig. 5. A part of the user interface which shows the regression models' parameters. For each output column the overall quality of fit measure—fitting score R^2—is depicted. Further, for each output column (dialog is cropped in this figure) the values of the coefficients of the models are shown. This way, the user can compare multiple models. This output is then the starting point for any subsequent use of the models.

subsets, providing also the possibility to select which regression model to select. In a regression model specification dialog we set independent and dependent variables (see Figure 4), and three different models are computed automatically.

The results of the computation are depicted in two ways. On the one hand they are shown in a table and on the other hand they can be also visualized. The table specifies the model name, input parameters, and output parameters. Further, we show the fitting score R^2 for each output parameter, and the intercept and linear coefficients for each of input parameters and for every output parameter fit (see Figure 5). In contrast to some interactive applications, which
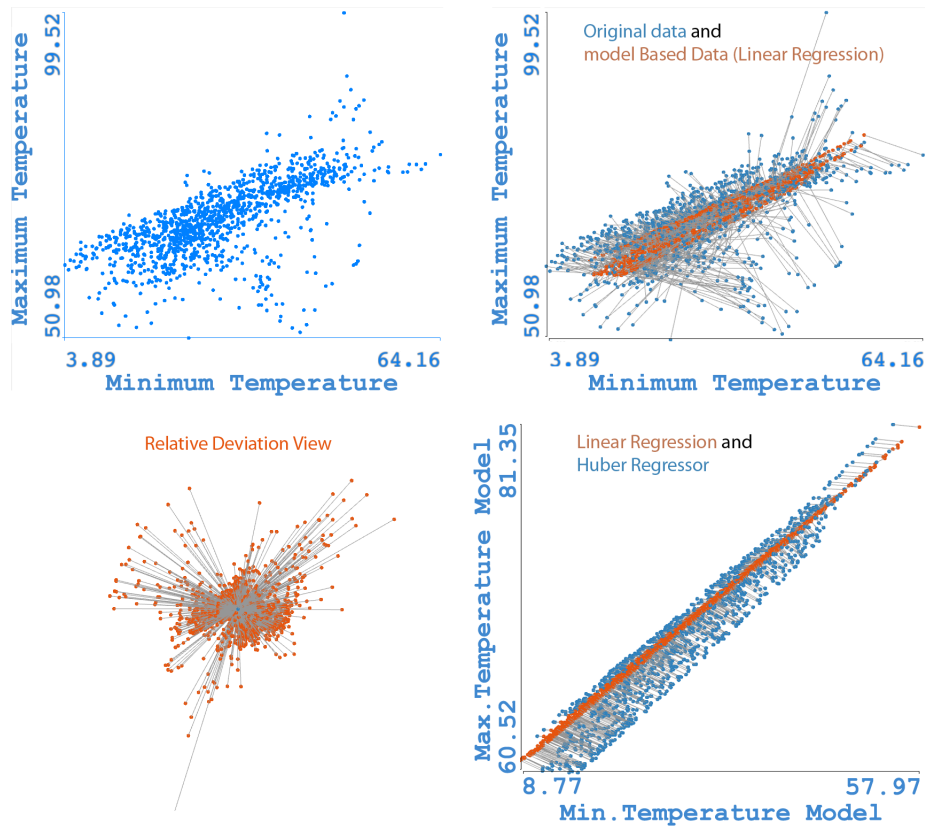
Fig. 6. Visualization for model evaluation. The scatterplot in the top left corner shows the minimum and the maximum temperatures for all stations. The top right view shows the same data in blue, and the vales computed using the liner regression in orange. Corresponding points are connected to visualize deviations. The bottom left view shows relative deviations only (imagine all blue points at the origin). The bottom right view compares two models, here liner regression and a Huber model.

do not offer a way to keep the data about the models, we keep the table as long as it is not explicitly deleted. By doing so, we make it possible for the user to compare different models, and to chose the best one for subsequent processing. In particular, the findings are also externalized in this way. The different models are computed using different subsets of data, and different modeling methods.

The quality-of-fit measure alone is often not sufficient to evaluate the models. It gives a good hint on model precision, but visualization can revel much more insight here. This is especially true for Huber and other robust regression models as the influence of outliers and the definition of *good* and *bad* heavily depend on the dataset structure and the context.
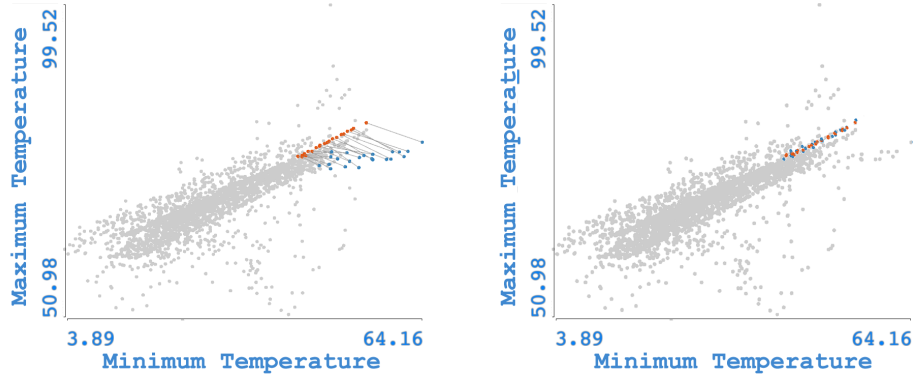
Fig. 7. Original points (in blue) and fitted points (in orange) for all Florida stations. The scatterplot on the left shows the points fitted using a global regression model, and the scatterplot on the right shows the points fitted using the model created for the Florida stations only. The local model, as expected, provides a much better fit.

In order to visualize the results, we use a modified scatterplot which shows original data points and the corresponding points, computed using a model, at the same time. The points' color differ, and we also show thin connecting lines between corresponding points. If the analyst is interested in relative error values, both for visualization and for interaction, we also can show relative deviations by placing all original points in the origin. In addition, the same technique can be used to compare different regression models with each other. Figure 6 shows different visualizations of the computed models. The top-left scatterplot shows the original data (minimum vs. maximum temperatures of all meteorological stations also shown in Figure 2). The top-right scatterplot shows the same data in blue and temperatures as computed using a liner regression model in orange. In an ideal case the points would perfectly coincide. In our case, however, large deviations are visible as expected. The view in the bottom-left shows relative deviations only—all blue points are aligned in the origin, and the orange points show the relative deviation for each station—we can see certain directions of particularly large errors. The bottom-right figure depicts how a Huber regression model (blue points) differs from linear regression (orange points). The data shown in the table in Figure 5 represent the same models. Obviously, there are multiple ways to visualize model accuracy. After long discussions with a domain expert, and after considering several options, we agreed to use these modified scatterplots. On the one hand, our users are used to scatterplots, and on the other hand, we are also able to meet main requirements posed by the domain, i.e., to show how models differ from original data, to see the error characteristics, and to visually compare different model results. We plan to extend the number of compared items to more then two in the future, expecting that we would need a more formal evaluation of such a design then, also.
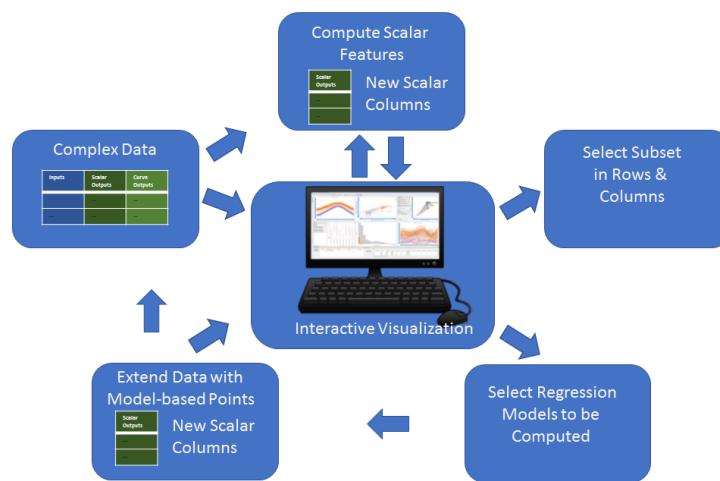
Fig. 8. The suggested workflow for externalizing findings. A tight interplay of interactive visualization and regression model building is necessary in order to use the resulting models in subsequent processes.

Instead of aiming at a global model for all the data, we focus on modeling parts of the data with local models (and considering a collection of them instead of one global model). In a way, this is related to piece-wise modeling, as for example with splines. One important aspect of our solution is the interactive instantiation and placement of local models. The user simply brushes a subset of data points in a view, activates the modeling dialog, and the models are computed and integrated, accordingly.

Figure 7, for example, shows all meteorological stations in Florida. The left scatterplot shows the model computed for all points—only the Florida stations are highlighted. The right scatterplot shows a liner regression model which is computed for the stations in Florida only. As weather characteristics are comparably similar across the state, the local regression model represents the data much better. Now that we have a well-fitted model for Florida, we could easily use it to estimate temperatures or precipitation values in other locations in Florida, or we could invert it, and find locations for desired temperatures and precipitation values. Note that we do all modeling using the scalar aggregates of the temperature and the precipitation curves. Clearly, we can also show the original curves in order to check the models in more detail.

This simple example illustrates the suggested workflow for interactive local modeling of complex data, also illustrated in Figure 8, that unfolds as an iterative and interactive visual analytics process, where the user can initiate a computation of new features whenever needed, and the computation of new regression models for selected subsets of data at any time. The visualization, depicted in the center of the diagram, is an essential part, and it is used as the control

mechanism for all analysis steps—all steps are initiated from the visualization, and all results are then visible to the user in return. Importantly, this workflow now includes that valuable findings are explicitly described in terms of the parameters (coefficients) of all computed models. The visualization also provides essential means to compare and evaluate the individual models.

Along with our research, we implemented this new workflow in ComVis [19], a visual analytics system based on coordinated multiple views. Regression modeling is realized on the basis of scikit-learn [22], a Python library for machine learning by calling the respective methods in scikit-learn package from ComVis.
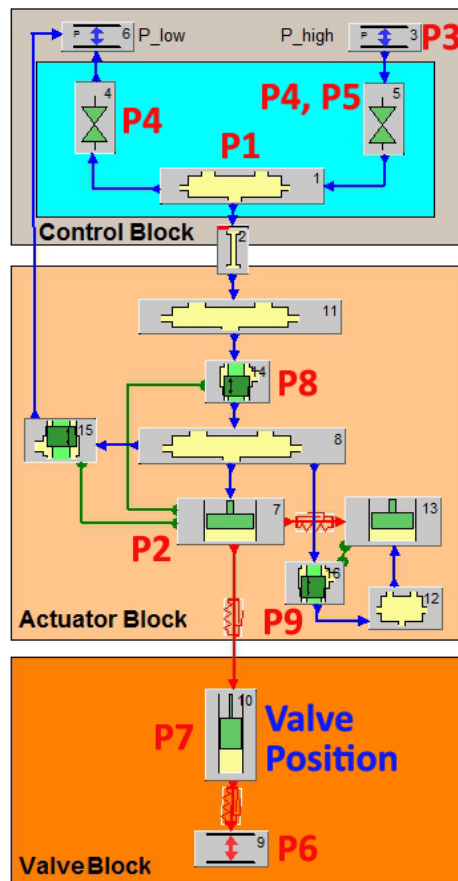
Fig. 9. The simulation model for the computation of the simulation ensemble. The control parameters are shown in red next to the corresponding elements. The output values are indicated in blue.
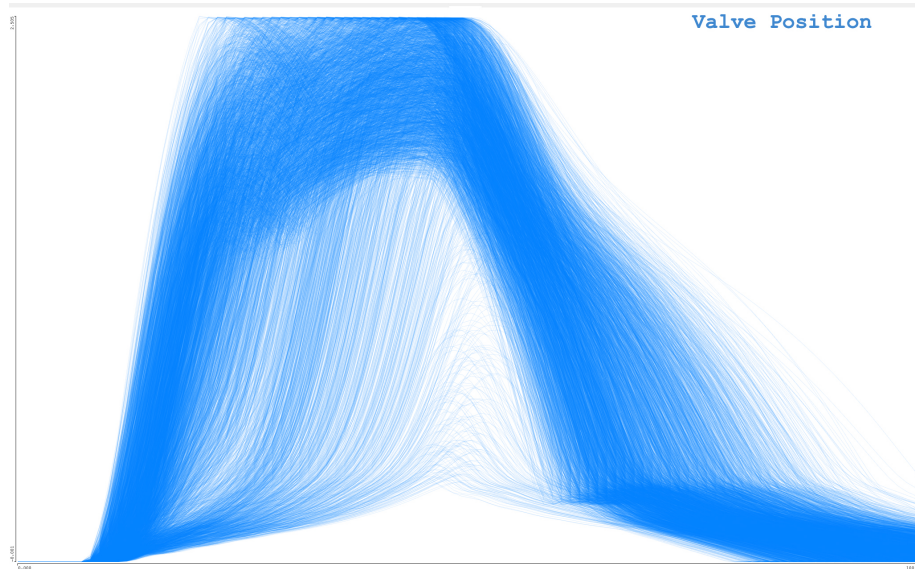
Fig. 10. Almost 5000 curves are shown from the simulation ensemble. Each curve shows the valve position for one hundred degrees of crankshaft revolution. The valve position is only one of many attributes in the dataset.

5   Case Study

In the following, we present a case study from the automotive simulation domain. We used our new interactive local modeling solution to analyze a Variable Valve Actuation (VVA) system for an automotive combustion engine. Optimizing VVA solutions is an active research field in the automotive industry and it is closely related to the development of new four-stroke engines. A precise control of the opening and the closing times of the intake and the exhaust valves is essential for an optimal engine operation. Conventional systems use a camshaft, where carefully placed cams open and close the valves at specific times, dependent on the mechanical construction of the cams. Variable valve actuation makes it then possible to change the shape and timing of the intake and exhaust profiles. In our case, we deal with a hydraulic system, i.e., an electronically controlled hydraulic mechanism that opens the valves independently of the crankshaft rotation.

Understanding and tuning of VVA systems is essential for automotive engineers. The valves' opening directly influences combustion, and therefore, also emission and consumption. A complete analysis of such a system is certainly beyond the scope of this paper. Still, we briefly present our joint evaluation in context of this case study, based on expertise in automotive engineering as well as in interactive visualization.

We study simulation data that consists of nine independent parameters and two dependent curve-attributes and it was computed based on the simulation

model shown in Figure 9. The independent parameters are: actuator volume size ($P1$), actuator piston area ($P2$), inflow pressure ($P3$), opening/closing time ($P4$), maximum flow area ($P5$), cylinder pressure ($P6$), valve mass ($P7$), port cut discharge coefficient ($P8$), and damper discharge coefficient ($P9$).

We computed simulation output for 4993 combinations of the control parameters. Here we focus on the valve position curves which describe the valve position relative to the closed state as a function of the crankshaft angle (see Figure 10). The valve opens when the curve rises and it closes when the curve declines; at the zero value of y-axis the valve is completely closed.

We see a great amount of variation in the curves' shapes with some rising steeply, some finishing early, and some not opening much at all. We needed a set of suitable scalar aggregates that describe these curves sufficiently well so that we could derive appropriate regression models for the data. Eventually, an important related task is optimization, for example supported by interactive ensemble steering [20]. In our case, we first aimed at extracting valuable findings, based on a visual analysis session of a user with automotive engineering expertise (supported by a visualization expert).

In order to properly capture the valve behavior, we decided to derive the following scalar aggregates during the analysis:

- *area under the curve*: quantity of mixture that enters/exits the cylinder
- *time of maximum opening*: time span during which the valve is open more than 98% of its maximum
- *time of opening*: first time when the valve opening is greater than 0
- *average opening of the value*: corresponds to the mean flow resistance
- *average valve opening velocity*: the average opening velocity from the start of the opening until 98% of the maximum is reached
- *velocity and acceleration at maximal opening*: corresponds to the force and moment when the valve hits its maximal opening
- *average valve closing velocity*: this velocity is computed for two ranges, i.e., one steeper and one less steep part of the curve
- *velocity and acceleration at closing*: corresponds to the force and moment when the valve closes again

Based on this derivation, the data set is extended by ten additional attributes. We select all data and compute regression models. As expected, we cannot capture all relevant relations between the inputs and the outputs by one global, linear model. Figure 11 shows a selection of deviation plots for a global model and we see overly large deviations, making it immediately clear that a more detailed approach is needed.

During the exploratory process, the analyst drills down into the data, and focuses on selected subsets. It is straight-forward to select relevant subsets of the data in the visualization (like in Figure 10, using a line brush [16] that extends over a subset of the shown curves). In our case, we started with selecting the curves that rise quickly, stay open for a long time, and then close smoothly (see Figure 12, on the left). New models were then created for these curves. In
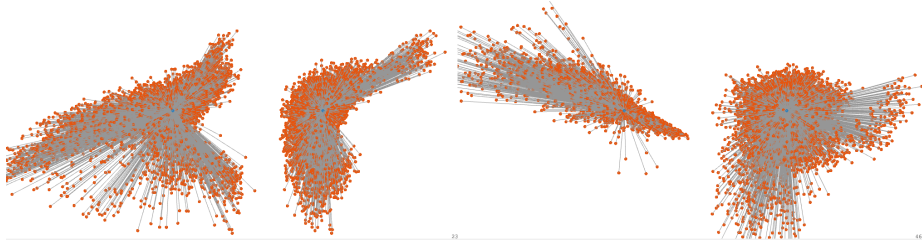
Fig. 11. A selection of deviation plots for a global model, showing that a more detailed analysis is needed.

the visualization, it becomes clear that the deviations are much more moderate, indicating more useful models. Figure 12 (on the right) shows some of the deviation plots (the images are cropped, but drawn using the same scale as in Figure 11). The derived (local) models are precise enough to be used in optimization or ensemble steering, but the corresponding parameter space domain has to be considered, of course.

Once satisfied with the local models (according to the visualization of all deviations), the analyst checks the regression models coefficients, and the findings can be described quantitatively. Figure 13 shows such a case. The analyst saw (in a scatterplot) that the opening velocity (slope of the curves when they rise) clearly depends on the P5 input parameter. He then decided to compute an according model. A quick model check (Figure 13 on the right) showed that the deviations were smaller for the medium values of the opening velocities. The details view then revealed:

$$velocity_{opening} = 0.0435 \cdot P5 + 0.0523 \tag{7}$$

Computing the same regression model, only taking medium velocity into account results in the following equation:

$$velocity_{opening} = 0.0246 \cdot P5 + 0.0881 \tag{8}$$

As a result, the analyst gained a quantitative understanding of how the average opening velocity depends on the P5 parameter in the middle opening velocity range. Finding an inverse function is then trivial, so parameter estimation for any desired velocities can be easily made.

The process continued, and the analyst selected new subsets. Figure 14 shows a screenshot of one display taken during the analysis session. Ideally, the analysis is conducted on multiple displays. Several different views are used simultaneously in a continuous interplay between interactive and automatic methods.

6 Discussion, Conclusion, and Future Work

The quantification and externalization of findings is often essential for a successful data exploration and analysis and in this paper we show how local linear
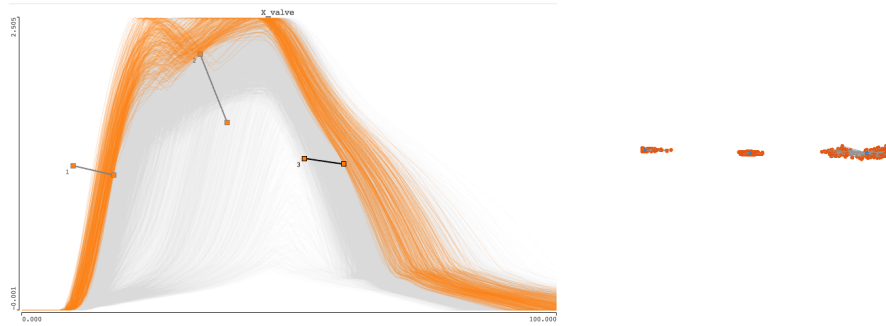
Fig. 12. After an interactive visual drill-down procedure, the expert focuses on a subset of desired curve shapes—fast opening and large integral value. New models are computed using only this subset of the data. The deviation plots on the right show that the according models are much more precise. Explicit model coefficients are also available, and remain visible during the entire analysis session.
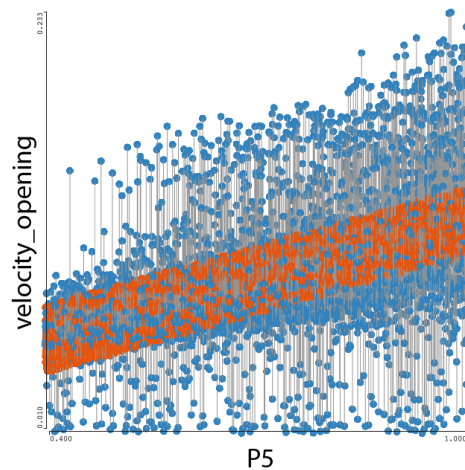
Fig. 13. A linear dependency between parameter P5 and the opening velocity is assumed after seeing this image. The model fits medium velocity values relatively good. Accordingly, the analyst computes another model, and results are indeed a bit better.

regression models can be used for this purpose. The resulting models are easy to comprehend and easy to invert, for example, during optimization. Our informal evaluation in the domain of automotive engineering showed that model reconstruction and the quantitative communication of findings are two very important analysis tasks. Due to the integration of modeling with visualization, we achieve a valuable mixed-initiative solution that not only accelerates the process of modeling, but also provides valuable means to model evaluation and comparison. Compared to a less integrated approach, e.g., when first exporting data
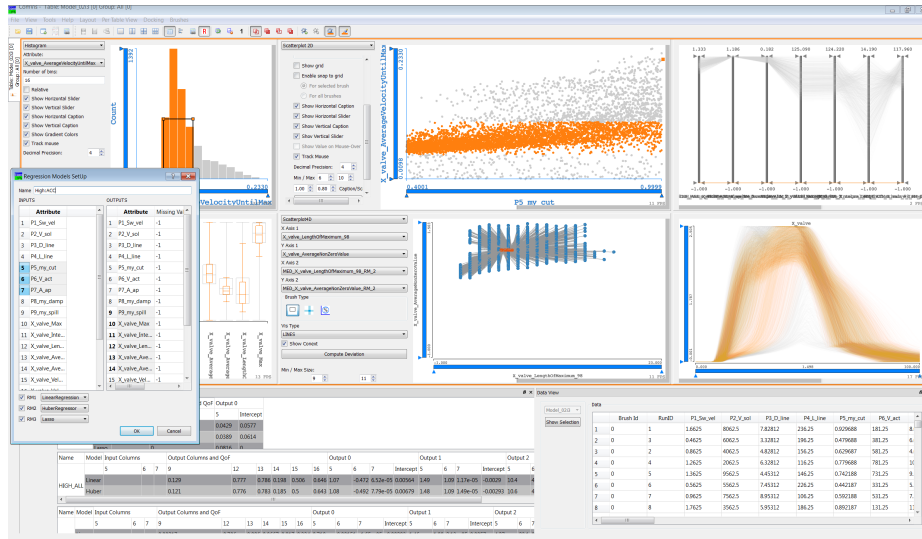
Fig. 14. A screenshot of an analysis session. Many views are used simultaneously, and the expert seamlessly switches between automatic and interactive analysis in an iterative interactive loop.

subsets from a visualization system, then modeling these subsets in a separate package, before then bringing the results back into the visualization, we now can iterate much more swiftly over multiple model variations and thus increase the likelihood of eventually deriving high-quality results.

Keeping the model data available throughout an entire analysis session, enables the comparison of different models in order to defer the choice of which model to use in subsequent analysis steps up to a point in the process, where enough information has been gathered. We also observe that users do explore and analyze the data more freely, when they know that previous findings are still available (related to the important undo/redo functionality in most state-of-the-art production software products).

All in all, we see this work as a first step towards even better solutions for the externalization of findings from visual analytics, here by means of regression models. We plan to add more complex models and to improve the model keeping mechanism. Currently, we do not support any automatic ranking of the models, or any kind of guidance in the selection of potentially suitable models. Additional quality-of-fit measures also may be implemented. Further, we plan to improve the visual exploration of the models' parameters (coefficients, quality-of-fit measures, etc.), also capitalizing on interactive visual data exploration and analysis, all in the same framework. Also the integration of other, non-liner models is relatively straight-forward, even though an according solution—while certainly more powerful—is likely to become more complex, also. An even better evaluation and a more thorough case study is also subject of future work.

7 Acknowledgements

References

1. Anscombe, F.J.: Graphs in statistical analysis. The American Statistician 27(1), 17–21 (1973)
2. Breiman, L.: Better subset regression using the nonnegative garrote. Technometrics 37(4), 373–384 (Nov 1995), http://dx.doi.org/10.2307/1269730
3. Frank, I.E., Friedman, J.H.: A statistical view of some chemometrics regression tools. Technometrics 35(2), 109–135 (1993), http://www.jstor.org/stable/1269656
4. Freedman, D.: Statistical Models: Theory and Practice. Cambridge University Press (2005)
5. Gauss, C.: Theoria motus corporum coelestium in sectionibus conicis solem ambientium. sumtibus F. Perthes et I. H. Besser (1809)
6. Haslett, J., Bradley, R., Craig, P., Unwin, A., Wills, G.: Dynamic graphics for exploring spatial data with application to locating global and local anomalies. The American Statistician 45(3), 234–242 (1991), http://www.jstor.org/stable/2684298
7. Holzinger, A.: Interactive machine learning for health informatics: when do we need the human-in-the-loop? Brain Informatics 3(2), 119–131 (Jun 2016)
8. Holzinger, A., Plass, M., Holzinger, K., Crişan, G.C., Pintea, C.M., Palade, V.: Towards interactive Machine Learning (iML): Applying Ant Colony Algorithms to Solve the Traveling Salesman Problem with the Human-in-the-Loop Approach, pp. 81–95. Springer International Publishing, Cham (2016)
9. Huber, P.J.: Robust estimation of a location parameter. Ann. Math. Statist. 35(1), 73–101 (03 1964), http://dx.doi.org/10.1214/aoms/1177703732
10. Hund, M., Böhm, D., Sturm, W., Sedlmair, M., Schreck, T., Ullrich, T., Keim, D.A., Majnaric, L., Holzinger, A.: Visual analytics for concept exploration in subspaces of patient groups. Brain Informatics 3(4), 233–247 (Dec 2016)
11. Kandogan, E., Balakrishnan, A., Haber, E., Pierce, J.: From data to insight: Work practices of analysts in the enterprise. Computer Graphics and Applications, IEEE 34(5) (2014)
12. Kehrer, J., Filzmoser, P., Hauser, H.: Brushing moments in interactive visual analysis. In: Proceedings of the 12th Eurographics / IEEE - VGTC Conference on Visualization. pp. 813–822. EuroVis'10, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2010)
13. Keim, D., Andrienko, G., Fekete, J.D., Görg, C., Kohlhammer, J., Melançon, G.: Visual Analytics: Definition, Process, and Challenges. In: Kerren, A., Stasko, J.T., Fekete, J.D., North, C. (eds.) Information Visualization, Lecture Notes in Computer Science, vol. 4950, pp. 154–175. Springer Berlin Heidelberg (2008)
14. Keim, D.A., Kohlhammer, J., Ellis, G., Mansmann, F.: Mastering the Information Age - Solving Problems with Visual Analytics. Eurographics Association (2010), http://books.google.hr/books?id=vdv5wZM8ioIC

15. Konyha, Z., Lež, A., Matković, K., Jelović, M., Hauser, H.: Interactive visual analysis of families of curves using data aggregation and derivation. In: Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies. pp. 24:1–24:8. i-KNOW '12, ACM, New York, NY, USA (2012)
16. Konyha, Z., Matković, K., Gračanin, D., Jelović, M., Hauser, H.: Interactive visual analysis of families of function graphs. IEEE Transactions on Visualization and Computer Graphics 12(6), 1373–1385 (2006)
17. Lampe, O.D., Hauser, H.: Model building in visualization space. In: Proceedings of Sigrad 2011 (2011)
18. Legendre, A.: Nouvelles méthodes pour la détermination des orbites des comètes. Méthode pour déterminer la longueur exacte du quart du méridien, F. Didot (1805)
19. Matkovic, K., Freiler, W., Gracanin, D., Hauser, H.: Comvis: A coordinated multiple views system for prototyping new visualization technology. In: 2008 12th International Conference Information Visualisation. pp. 215–220 (July 2008)
20. Matković, K., Gračanin, D., Splechtna, R., Jelović, M., Stehno, B., Hauser, H., Purgathofer, W.: Visual analytics for complex engineering systems: Hybrid visual steering of simulation ensembles. IEEE Transactions on Visualization and Computer Graphics 20(12), 1803–1812 (Dec 2014)
21. Oceanic, N.N., Administration), A.: Climate data online. https://www.ncdc.noaa.gov/cdo-web/datasets/ (2017), [Online; accessed 19-June-2017]
22. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
23. Piringer, H., Berger, W., Krasser, J.: HyperMoVal: Interactive Visual Validation of Regression Models for Real-Time Simulation. Computer Graphics Forum (2010)
24. Radoš, S., Splechtna, R., Matković, K., Đuras, M., Gröller, E., Hauser, H.: Towards quantitative visual analytics with structured brushing and linked statistics. Computer Graphics Forum 35(3), 251–260 (2016), http://dx.doi.org/10.1111/cgf.12901
25. Shao, L., Mahajan, A., Schreck, T., Lehmann, D.J.: Interactive regression lens for exploring scatter plots. Computer Graphics Forum (Proc. EuroVis) p. to appear (2017)
26. Shneiderman, B.: Inventing Discovery Tools: Combining Information Visualization with Data Mining. Information Visualization 1(1), 5–12 (Mar 2002)
27. Shrinivasan, Y.B., van Wijk, J.J.: Supporting exploration awareness in information visualization. IEEE Comput. Graph. Appl. 29(5), 34–43 (Sep 2009)
28. Tam, G.K.L., Kothari, V., Chen, M.: An Analysis of Machine-and Human-Analytics in Classification. IEEE Transactions on Visualization and Computer Graphics (2016)
29. Thomas, J.J., Cook, K.A.: A visual analytics agenda. IEEE Computer Graphics and Applications 26(1), 10–13 (Jan/Feb 2006)
30. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) 58(1), 267–288 (1996), http://www.jstor.org/stable/2346178
31. Tukey, J.: The technical tools of statistics. American Statistician 19 (1965)
32. Yang, D., Xie, Z., Rundensteiner, E.A., Ward, M.O.: Managing discoveries in the visual analytics process. SIGKDD Explor. Newsl. 9(2), 22–29 (Dec 2007), http://doi.acm.org/10.1145/1345448.1345453