What You See Is What You Can Change: Human-Centered Machine Learning By Interactive Visualization

Dominik Sacha[a,*], Michael Sedlmair[b], Leishi Zhang[c], John A. Lee[d], Jaakko Peltonen[e], Daniel Weiskopf[f], Stephen C. North[g], Daniel A. Keim[a]

[a] *University of Konstanz, Germany*
[b] *University of Vienna, Austria*
[c] *Middlesex University, UK*
[d] *Université catholique de Louvain, Belgium*
[e] *Aalto University and University of Tampere, Finland*
[f] *University of Stuttgart, Germany*
[g] *Infovisible, Oldwick NJ, USA*

Abstract

Visual analytics (VA) systems help data analysts solve complex problems interactively, by integrating automated data analysis and mining, such as machine learning (ML) based methods, with interactive visualizations. We propose a conceptual framework that models human interactions with ML components in the VA process, and that puts the central relationship between automated algorithms and interactive visualizations into sharp focus. The framework is illustrated with several examples and we further elaborate on the interactive ML process by identifying key scenarios where ML methods are combined with human feedback through interactive visualization. We derive five open research challenges at the intersection of ML and visualization research, whose solution should lead to more effective data analysis.

Keywords: Machine Learning, Information Visualization, Interaction, Visual Analytics

---

1. Introduction

Real-world data analysis usually relies heavily on both automatic processing and human expertise. Data size and complexity often preclude simply looking at all the data, and make machine learning (ML) and other algorithmic approaches attractive, and even inevitable. However, the power of ML cannot be fully exploited without human guidance. It remains a challenge to translate

*Corresponding author
Email address: dominik.sacha@uni-konstanz.de (Dominik Sacha)

Preprint submitted to Neurocomputing. To appear under DOI: 10.1016/j.neucom.2017.01.105 *April 28, 2017*

real-world phenomena and analysis tasks, which are often under-specified, into ML problems. It is difficult to choose and apply appropriate methods in diverse application domains and tasks. More importantly, it is crucial to be able to incorporate the knowledge, insight, and feedback of human experts into the analytic process, so that models can be tuned and hypotheses refined.

In a typical setting, domain experts use ML and visualization methods provided by common software tools (*e.g.*, SPSS, R, Tableau) "out of the box". Realistically, the domain experts' proficiency in ML may be limited, and the underlying computations may not be transparent and comprehensive enough to provide the feedback needed to guide model refinement. Visualizations are often used to display the ML model results without offering interactions that trigger recalculations. This results in a very standardized configuration of the ML and visualization pipelines based on default parameters that domain experts may not know how to adapt. The situation may be improved by having domain experts collaborate with data scientists, improving the effectiveness of analysis, but also leading to a much more costly iterative design process. ML researchers usually know how to tune models directly in ML platforms (*e.g.*, Matlab, R, Python) and provide results to domain experts. However, domain experts generally find it necessary to learn how models behave and how to evaluate results to provide useful feedback.

By integrating ML algorithms with interactive visualization, visual analytics (VA) aims at providing visual platforms for analysts to interact directly with data and models [1]. Tam et. al [2] illustrated in case studies that human-centric ML can produce better results than purely machine-centric methods. In such cases, an analyst is enabled to steer the computation and interact with the model and data through an interactive visual interface. Despite much effort to date, though, solutions from ML and VA are still not interwoven closely enough to satisfy the needs of many real-world applications [3, 4]. For example, in existing toolkits (such as WEKA, Elki, or javaML), tight integration between interactive visualization and ML process is missing. Most of these tools present modeling results as static visualizations; interactions are often limited to command line interfaces or user interface controls that are not intuitive and accessible to end-users. Toward better integration of ML and VA, in recent years conceptual frameworks that characterize the interplay between them have been proposed [1, 3, 4, 5]. It appears most frameworks were designed from the perspective of interactive visualization, focusing on the role of the "human in the loop". A closer connection between visualization and common ML paradigms (such as unsupervised and (semi-) supervised learning; classification, regression, clustering, etc.) including specifics of these methods (*e.g.*, SVM vs. random forests in classification) and their implementations is needed. In this paper, we put a sharper focus on scenarios in which complementary ML and VA methods are combined, and propose a framework for a tighter relationship between ML and VA. To do so, we identify aspects of automated ML techniques that are amenable to interactive control, and illustrate these with examples. We further describe human factors within this process that should be considered carefully in the design of interactive visual ML systems, and enumerate analysis scenarios. The proposed conceptual framework opens perspectives on new ways of combining automated and interactive methods, which will lead to better integrated, and, ultimately, more effective data analysis systems.

Researchers in both ML and visualization have realized for some time that closer collaboration could help to solve this problem. An interdisciplinary team with experts from the ML and visualization communities was formed at a Dagstuhl Seminar on "Bridging Information Visualization with Machine Learning" [6]. The framework proposed in this study is the outcome of several iterations of discussions, feedback, and framework refinements made by this team. The initial version of this framework [6] was based on a survey of several earlier frameworks and systems combining ML and interactive visualization. Subsequently, the framework was refined by applying it to a larger set of example applications (identified in the visualization, ML, and HCI literature) and by incorporating external feedback from experts, such as conference submission reviews. This led us to a process of framework refinement, carried out over 1.5 years, including extensions and simplifications, validation, and evaluation by analyzing existing VA systems and ML techniques. This paper extends an initial report in ESANN 2016 [7] to include an extended review of prior work, a more detailed framework, examples of providing automated support for each stage, identification and description of scenarios where analysis and feedback take place, and additional discussion throughout.

The rest of this paper is structured as follows. Section 2 discusses related work on the interplay of machine learning and human feedback. Section 3 introduces our conceptual framework and the key stages in its interactive pipeline, and they are illustrated with examples in Section 4. Section 5 examines the human interaction loop in more detail, describing the stages of action and analysis scenarios where interaction occurs. Section 6 identifies five challenges and associated opportunities in creating systems that fully use the framework. Section 7 gathers conclusions and final discussions.

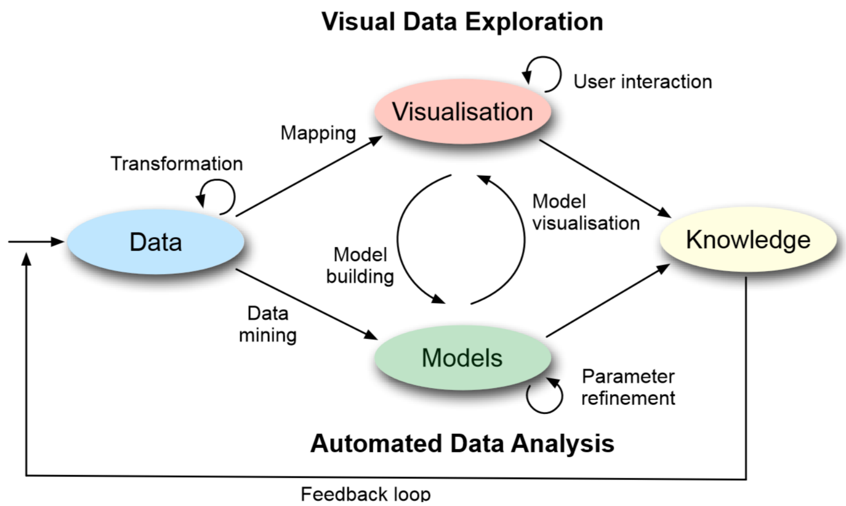

Figure 1: Visual Analytics Framework by Keim et al. [1]. ML interactions are related to model building and parameter refinement.

2. Related Work

The literature describes related models that capture the interplay between ML system components and human feedback loops. We will discuss several different perspectives on this topic, divided into VA models, interaction taxonomies, interactive ML, and human-centered design. This section concludes with a high-level summary for interested readers without ML expertise.

Visual Analytics Models. Pipeline-based models such as the *Reference Model for Information Visualization* [8] or the *Knowledge Discovery Process in Databases* (KDD) [9] usually contain feedback loops that cover all the subcomponents with the potential for user interaction. In the standard VA model [1], the analysis process is characterized by interactions between data, visualizations, models of data, and users, for knowledge discovery (see Figure 1). ML interaction in this framework is aimed at model building and parameter refinement. Sacha et al. extended this model [4] to encompass the process of human knowledge generation. This extended model clarifies the role of humans in knowledge generation, and highlights the importance of supporting tighter integration of human and machine. Several other models focus on a clear depiction of the human data analysis process, including Pirolli and Card's sensemaking process [10], and Pike et al.'s science of interaction [11]. Endert et al. characterized the interaction process between a human analyst and automated analysis techniques as the "human *is* the loop" [12] and proposed a model for coupling cognition and computation [3]. More recently, Chen and Golan [13] provided an abstract model to describe six classes of human-machine workflows in combination with an information-theoretic measure of cost-benefit. Their model allows one to analyze workflows composed of machine computations and human interactions supported by different "levels" of visualizations. All these models reflect a high-level understanding of system and human concepts.

Interaction & Task Taxonomies. Another set of models related to our endeavor seek to characterize and organize the tasks and interactions in a visual data analysis process. For example, Brehmer and Munzner [14] propose a comprehensive visualization task taxonomy. However, model interactions only arise in tasks they refer to as "aggregate" or "derive" tasks. Landesberger et al. [15] define a taxonomy that includes interaction and data processing. Their taxonomy provides two types of data processing interactions: data changes, such as editing or selecting data, and processing changes, such as scheme or parameter changes. They incorporate Bertini and Lalanne's [16] distinction of human intervention levels, that distinguishes, for example, between scheme tuning (*e.g.*, parameter refinement) and scheme changing (*e.g.*, changing the model) interactions. Mühlbacher et al. [17] investigate and categorize several types of user involvement for black box algorithms with different characteristics. The characterization of interactions in our framework is orthogonal to these taxonomies and extends them with a dedicated view on interaction with ML components.

Interactive Machine Learning. While the above models were strongly framed from the viewpoint of visualization and VA, there is also growing interest in the ML community to incorporate human interaction more fully in the analysis and learning processes. A typical scenario would be that a human observes or explores the current state of a learning system, and explicitly or implicitly guides an ongoing training process. This is often referred to as *human-in-the-loop machine learning*, or more broadly, *interactive machine learning*. The classical example is recommender systems that infer users' personal preferences from previous choices. User can provide continuous feedback, such as by recording additional choices, or by explicitly scoring (liking/disliking) individual items [18, 19, 20]. In similar scenarios, the user provides class labels, for which a machine learning classifier is (continuously) trained. This concept is strongly linked to the topic of *active learning* [21, 22], which aims at efficient choices of samples during training epochs to achieve fast

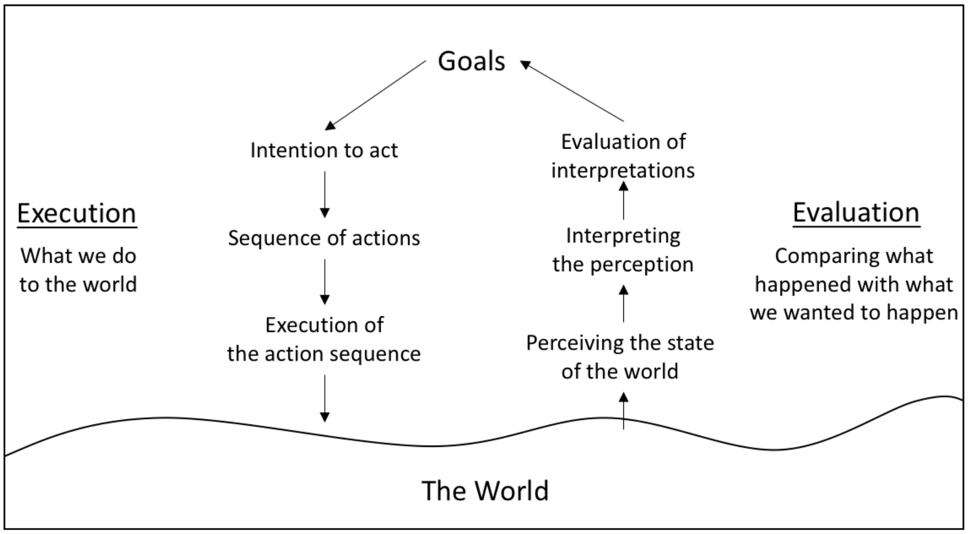

Figure 2: Stages of interaction [24].

convergence of a learning algorithm. An increased demand for involving user feedback is underscored by recent work investigating more elaborate user models or more intricate forms of interaction. For example, Amershi et al. [5] propose a set of high-level paradigms by which user involvement in ML may be characterized, similar to models that have been discussed in the VA community [1]. They also stress the importance of accounting for user behavior, and the potential benefits of collaborative research between ML experts and the human-computer interaction community. Similarly, Groce and colleagues investigate sample selection strategies to test classifiers effectively via systematic feedback requests to end users [23].

Human-Centered Design. Another perspective on the analysis process is provided by the human-computer interaction (HCI) domain. We are able to adopt commonly known concepts and terms in our interactive ML setting, considering the interplay between human and machine. A famous example is Norman's *Stages of Action* cycle [24], shown in Figure 2. At the center of the cycle are the *Goals* that a human analyst wants to achieve. Norman distinguishes between two major stages of an interaction: *Execution* and *Evaluation*. In the *Execution* phase, the human (1) forms an intention to act and (2) specifies a sequence of actions that is (3) finally executed to the world. Subsequently, in the *Evaluation* phase, the state of the world has to be (1) observed, (2) interpreted, and (3) finally compared and evaluated with respect to the initial goals. Norman further describes the distances or "gulfs" between the human goals and the world that need to be bridged when humans interact with (digital) interfaces. The *Gulf of Execution* describes the problem when the human does not know how to perform an action, whereas the *Gulf of Evaluation* indicates that humans are not able to evaluate the result of an action. Human-centered user interface design attempts to bridge these gaps. User interface features need to be visible and offer *Affordances* to the end user. These (perceived) affordances are relationships between a person and a physical/digital object, and suggest how the object might be used [24]. *Visual Cues* (*e.g.*, visual elements, icons, or animations that attract attention) may guide the end user during the analysis process. On the one hand, a system should communicate the progress of ongoing computations or the quality of results to the analyst. On the other hand, visual cues may guide the analyst to "handles" or objects that can be manipulated within the interface. In this respect, the concept of *direct manipulation* [25] has been demonstrated to enable intuitive operations to end users. Interactive visualizations of ML model structures and data items allow direct interaction that is more convenient and more easily interpreted than text commands.

Machine Learning Overview. A wide range of ML algorithms and methods have been proposed and employed in practice. One way to distinguish these methods is based on their learning paradigm, which is either *supervised* (examples of system inputs and desired outputs are both provided), *unsupervised* (no desired outputs are specified), or *semi-supervised* (not all outputs are available, typically only a few). While supervised methods aim to learn the input-output relationship from the provided examples, unsupervised methods attempt to extract hidden structures from them. These learning paradigms can be instantiated into specific categories of ML tasks. *Regression* aims to best predict any form of continuous outputs as a function of the inputs. *Classification* aims to predict class labels or memberships associated with the inputs. On the unsupervised side, *clustering* aims to identify groups or hierarchies present in data. Similarly, *dimensionality reduction* and *manifold learning* both aim to identify linear or nonlinear relationships between the observed variables and to represent the subspace where most of the data variation happens with fewer latent variables. These are a few examples of emblematic ML tasks, among many others, like association rule learning, missing value imputation, time series prediction and outlier detection, novelty detection. In practice, several of these abstract tasks are combined in a data flow to solve real-world problems and analysis. For example, one might apply dimensionality reduction (to mitigate the computational impact of working with high dimensional data) before applying regression or time series prediction. Such combinations of methods, though useful in practice, lead to composite models with heterogeneous parameterization, which are difficult to train, and time-consuming to validate.

In summary, both VA and ML communities have noticed the gaps between automatic ML and human interactions in data analytics systems, which limit their effectiveness in solving real world application problems. Various models to conceptualize the potential integration of ML and interactive visualizations have been proposed. These models, however, still have either a strong human/visualization focus, or a strong algorithmic focus. In this paper we propose a new conceptual framework that covers both aspects with the objective of providing a more systematic view of how interactive visualization and ML algorithms can be integrated in practice.

3. Human-Centered Machine Learning Framework

As shown in Figure 3, our framework unifies, embeds, and extends existing theories on interactive ML and VA by integrating and generalizing observations from emergent case studies and examples. The framework combines typical ML and VA pipeline components (A–D) with an analysts' iterative evaluation and refinement process (E). An analyst can interact with the individual stages in this pipeline through a visual interface (D), which acts as a mediator or "lens" between the human and the ML components (*dashed arrows*). Changes are then sent back to the visual interface and presented to the analyst (*solid arrows*). The dark blue boxes in the figure denote examples of automated methods that support the analyst in performing specific interactions. Our framework illustrates that a multidisciplinary perspective combining ML and VA is needed to provide usable and accessible access to end-users (domain experts). For example, data operations, visualization techniques, and human-computer interaction (blocks A, D, and E in Figure 3) are addressed in the visualization community, whereas ML algorithms, setups, and optimization (blocks B and C) are core to ML research. Next, we detail the interactions involved in each stage of the analysis process, and discuss possible automatic support to facilitate these interactions.

Edits & Enrichment (A). In ML, data is often seen as fixed or immutable, but many VA tools support data cleaning, wrangling, editing, and enrichment, which is essential in many applications

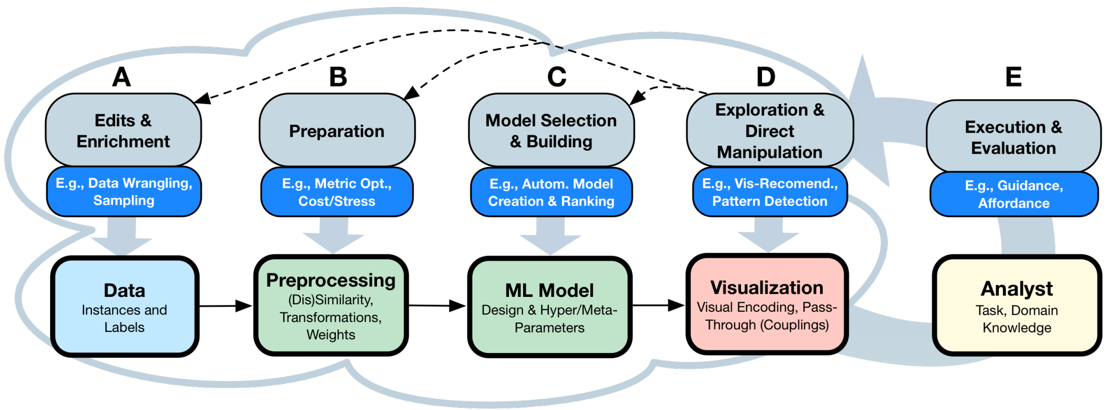

Figure 3: Proposed conceptual framework: A reference interactive VA/ML pipeline is shown on the left (A–D), complemented by several interaction options (light blue boxes) and exemplary automated methods to support interaction (dark blue boxes). Interactions derive changes to be observed, interpreted, validated, and refined by the analyst (E). Visual interfaces (D) are the "lens" between ML models and the analyst. Dashed arrows indicate where direct interactions with visualizations must be translated to ML pipeline adaptations. The colors of the pipeline components refer to the ones defined in the VA process model [1] shown in Figure 1.

[26]. For example, in a typical active learning scenario in ML, a domain expert may want to incrementally add labels to data while training a classifier in order to inject domain knowledge and improve the quality of the classifier. Another example of Edits & Enrichment interaction is the testing of "what-if scenarios" on the data. The analyst might want to change or remove some data points and see the effect to test certain assumptions about the data. Data editing is often followed by a "warm restart" of the ML pipeline, iteratively propagating results to the analyst. From an ML perspective, data editing combined with user feedback can be seen as a form of cross-validation/bootstrap. In these techniques, the ML model is re-trained with "modified" data, either data held out in cross-validation and leave-one-out, or changed into some other observed instance that is then reintroduced in the bootstrap sample. However, traditional cross-validation/bootstrap are performed with strict rules about how data are held out or modified, to pursue statistical goals about generalization performance, whereas the editing and feedback discussed here are performed by the user to carry out a task, not constrained to a specific mathematical formalization of the task. Hence data editing might be seen as a kind of "meta" cross-validation, requiring proper quality assessment for the user's task.

Automatic Support: Various statistical and ML techniques exist for preconditioning and processing data. These techniques can be implemented in interactive systems for data wrangling, such as missing value detection and replacement, sampling, and data transformation. When datasets are large, the analyst can apply sampling techniques (*e.g.*, vector quantization or hierarchical clustering) to derive a representative subset for interactive analysis. Similar methods can also be used for efficient labeling, for example, by adding an annotated class label to all items in a particular cluster.

Preparation (B). Many ML pipelines or VA workflows incorporate preprocessing steps that are selected and adjusted by a-priori domain experts. Being outside of the scope of the central ML model, the design options and parameters in these steps often have a different status. For instance, they may not be subject to cross-validation in some cases. While edit-and-enrich interactions focus on persistent changes to data (possibly individual items), preparation interaction applies a uniform, transient transformation of features to a larger set of observations. Typical preparation activities include transformation of data such as standardization, scaling, Fourier or wavelet transforms, and weightings. Weightings include filtering (0-weights) of data items, as well as feature selection [27]. In this respect, we often observe a gap in the "judgment of (dis)similarity" between a human and the "default" metrics used in ML methods. Analysts usually focus on specific features or subsets within their data. This requires feature weightings, or defining more complex (dis)similarity functions.

Automatic Support: ML offers several measures and methods to optimize *Preparation*. For example, feature weighting can be supported in the form of relevance [28], metric [29, 30], or kernel learning [31]. Other setups make use of cost functions or stress for optimizing parameterizations of preprocessing steps. Furthermore, other methods such as correlation (e.g., Pearson correlation) or factorial analysis support analysts in understanding feature dependencies within data.

Model Selection & Building (C). An essential idea of VA is to enable analysts to interact directly with ML models so they can integrate domain knowledge into the analysis process. In *Model Selection*, analysts choose among various ML algorithm families, or a set of pre-built model results. Another possibility is to build ML model ensembles interactively. *Model Building* interactions focus on changing a given ML model through the adjustment of model parameters. While internal model parameters are usually optimized automatically, others, such as design or form, and meta- or hyper-parameters, need to be adapted by the analyst according to their assumptions. Model building interactions can lead to ML model changes that affect its *form*, *constraints*, *quality*, and *accuracy*.

Form parameters define basic structures (such as the number of nodes in a neural network), whereas *constraints* reflect more detailed assumptions (*e.g.*, defining fixed anchor points in a dimensionality reduction algorithm). In some applications it is also desirable to tune the *quality* and *accuracy* of the ML result (*e.g.*, by interacting with the confusion matrix of a classifier).

Automatic Support: Model Selection can be supported automatically or semi-automatically, for example, with Akaike or Bayes information criteria, cross validation, bootstrap [32], etc. These techniques assess the quality of a model based on its complexity (roughly, the number of free parameters) and the generalization error, which allows different ML models to be compared and ranked. When there are too many models to exhaustively compare all using model selection methods, higher-level *Model Building* can also be supported by automatic methods, especially for metaparameter optimization (i.e., parameters that cannot be tuned by optimization within the model family and where the analyst is not able to provide "useful" feedback). For example, heuristic approaches such as genetic algorithms can be applied to select features and a regularization parameter for support vector machine classifiers based on the quality (cross-validated performance or theoretical performance bounds) of the classification result [33].

Exploration & Direct Manipulation (D). The various characteristics, parameters, and results of all pipeline stages can be presented to the analyst as visualizations in a user interface. On the one hand, data can be visualized using a plethora of known visualization techniques. On the other hand, visualizations of ML components can be presented as well. VA aims to combine the two variants by incorporating ML results or patterns (e.g., identified groups, classes, or outliers) into data representations. We found visual representations for different parts of the ML pipeline, such as data and model spaces (Figure 4-a), pre-built model variants including their characteristics (Figure 4-b) and quality (Figure 4-a/b/d), but also the ML structures (Figure 4-d). Interactive visualizations that allow for *Direct Manipulation* of visual objects make ML interactions amenable to analysts. Usually, simple *Exploration* interactions, such as changing a graphical encoding, or navigating within views, do not feed back to ML components but help the analyst to understand and interpret the visualization. However, the preceding discussion also mentioned several situations where interactions in visual interfaces are "passed through" to ML changes that trigger recalculation of the ML pipeline, as indicated by dashed arrows in Figure 3. This concept has become known as "semantic interaction" that maps intuitive observation level interactions in a visualization to

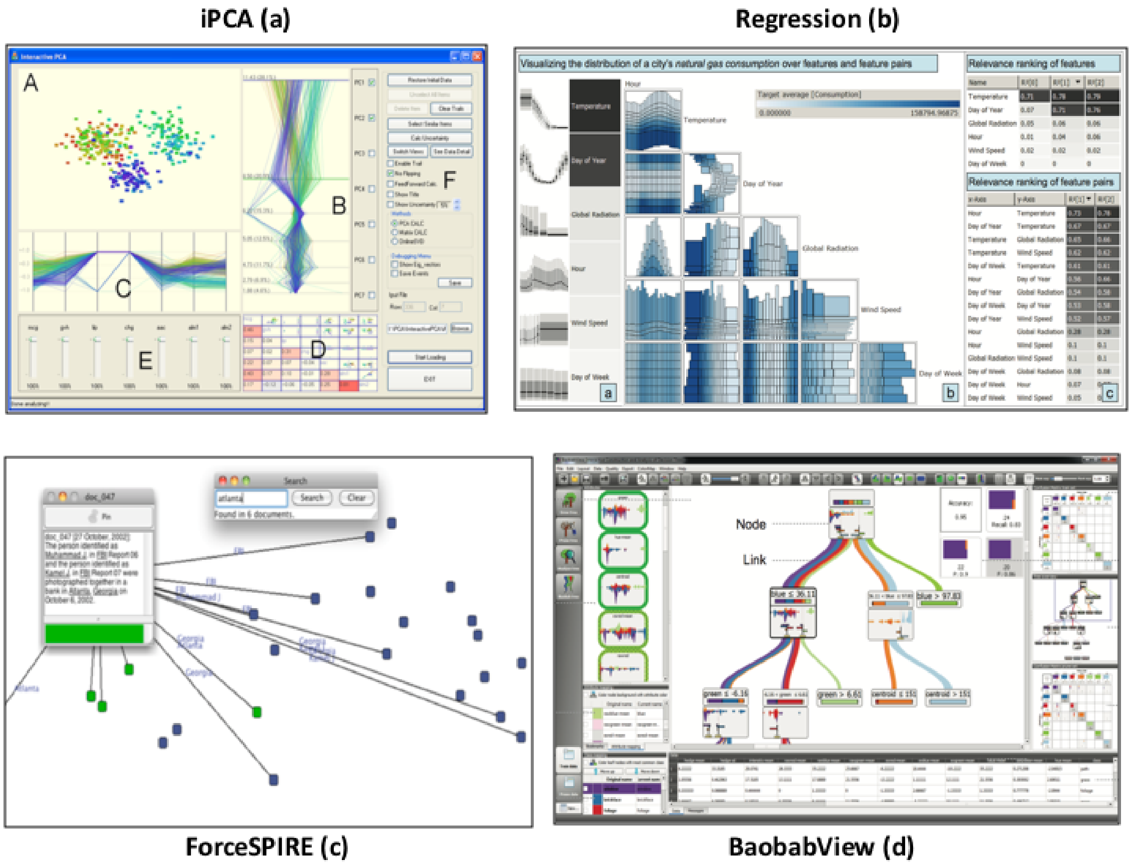

Figure 4: A selection of examples that effectively involve analysts into the ML process by interactive visualization. Courtesy of Jeong [34], Mülbacher [35], Endert [36], van den Elzen [37].

appropriate ML changes [36].

Automatic Support: Automated methods can be used to detect and highlight specific visual patterns, such as class separation, correlation, outliers, or sequences. These methods imitate human perception with the goal of better helping human analysts find interesting, visible patterns in the data. Aupetit and Sedlmair [38], for instance, provide a rich set of over 2000 measures that automatically detect visual class separation patterns. Similarly, different visualization techniques (e.g., scatterplots, parallel coordinates, or matrices) may be employed to provide different perspectives of data and ML results. In summary, recommending "interesting' visualizations has excellent potential for improving the effectiveness of data analysis.

Execution & Evaluation (E). This step involves the entire interactive and iterative analysis process, including all the above-mentioned interaction components *A-D*. Interactive visualizations (D) not only serve as an aid or "lens" that facilitates the process of interpretation and evaluation of ML results, but also make the execution of ML interactions amenable to analysts. In an ideal VA system, analysts actively engage in an iterative process of observing, interpreting, and evaluating the system's outputs, followed by subsequent execution of interactions to refine the analysis. This duality of interaction design goals has been characterized by Norman's pioneering work on *Stages of Action* [24] (see Section 2 for more details). However, the system should actively enable and support this duality by providing usable and interpretable visual interfaces considering human-centered design, such as affordances [24, 39], direct manipulation [25], and interpretable representations. We will provide a more detailed perspective on this human loop in Section 5 describing an analyst's thinking, sense-making, and reasoning process influenced by various human factors.

Automatic Support: Boy et al. [40] investigated visual cues as perceived affordances in a "suggested interactivity" study, with the goal of providing guidance to analysts. Furthermore, the analysis process itself can be recorded (as a sequence of interactions) and visualized to enable browsing through various analysis states, adding analytic provenance capabilities [41]. In this context, the quality of an interaction result can be measured and compared within such a sequence to automatically distinguish beneficial from detrimental changes. For example, Kapoor et al. measured the accuracy of a classifier before and after interaction [42]. However, such measures are rarely available today, especially in more exploratory or speculative types of analysis.

4. Example Systems

Our framework was inspired by studying current data analysis systems that engage analysts through interactive visualization. In this section, we discuss ForceSPIRE [36] as an example of how interactive visualization can be integrated into each stage of an automatic analysis process. We will also briefly review some other relevant examples, and map their interactions to our framework to show how it covers many different types of interactive visualization, as well as its potential for identifying interactions missing in the analysis process.

4.1. ForceSPIRE

ForceSPIRE [36] is an interactive text visualization and analysis system. It takes a collection of text documents as input and shows them in a force-directed layout, driven by document similarity (measured by comparing common terms). The analyst can explore documents in this spatial representation, and take advantage of domain knowledge about them by means of interactions such as document movement, highlighting, annotation and search. Below is a mapping of these interactions to our framework:

Edits & Enrichment (A). The data (text documents) can be enriched with annotations, for example, by adding topic-terms that do not explicitly occur in the text.

Preparation (B). Document similarities are derived from common terms, which are transferred into a weighted feature vector for each document. Term weights are adapted based on user interactions, such as highlighting and searching for specific terms. In addition, term weights may be updated if the analyst rearranges document positions in the layout. Documents that are moved closer are considered more similar, and the term weights are adapted accordingly.

Model Selection & Building (C). A force-directed graph model is derived to create a two dimensional spatialization of the documents. Document nodes are treated as physical objects, where the number of entities or terms per document defines its mass, so larger documents move more slowly. Document similarities are represented by graph edges or ideal springs connecting related document nodes. Spring forces are calculated from common terms and an importance value or weight per term. The analyst may add constraints to the document layout by pinning specific nodes to fixed positions.

Exploration & Direct Manipulation (D). Documents are visualized as nodes that can be opened or closed on demand. This allows the analyst to explore documents, inspect details on demand, and provide feedback as needed. The visualization offers direct manipulation interactions that can be translated to ML-pipeline adaptions. These "semantic interactions" adapt the underlying term-weight model or add constraints into the document layout. In this way, the analyst can enrich the spatialization with semantic meaning to support the human reasoning process.

Execution & Evaluation (E). Each interaction that is performed results in an observable behavior within the visualization. For example, new clusters may emerge as a result of pinning a document in the layout, as some documents may move further away, and others may move closer to the pinned document. In such cases, the analyst needs to interpret and validate the results and provide further feedback.

4.2. Other Examples

We found many other examples of VA systems described in the literature that support aspects of the proposed analysis process. In this section, we briefly review a few examples that illustrate a wide variety of realizations. Table 1 summarizes these examples in terms of the framework components (A–E).

Inter-Active Learning [43] is a concept proposed by Höferlin et al. aimed at extending active learning. In line with standard active learning approaches, the proposed system allows the analyst to iteratively add class labels (A) to train a classifier. In contrast with traditional active learning, the analyst can also pose queries to identify points to be labeled. Additionally, the analyst is provided with multiple views that visualize the classifier quality (D) and let the analyst tune parameters of the classifier (C).

DataWrangler [44] provides a good example of supporting Edits & Enrichment (A). The system automatically validates selected data and suggests data transformations to the analyst to implement data editing operations.

iPCA [34] (Figure 4-a) is an interesting example that offers different perspectives and interactions within data and model space. The analyst can edit or remove data items (A) in linked views (e.g., scatter plot, parallel coordinates, or matrix views) (D) and observe the results (E). It is also possible to define dimension loadings using sliders (B).

Dis-Function [45] learns distance functions from user feedback. When the analyst rearranges data points in a two dimensional embedding (D), a calculation of a new distance functions (or feature weights) is triggered (B). By immediately revealing the resulting changes (updated scatter plot and bar chart), the approach gives analysts a convenient way of exploring alternative configurations of preprocessing steps (E).

The partition-based framework by Mühlbacher and Piringer [35] (Figure 4-b) provides the analyst with different pre-built regression model variants that can be selected and refined (C). The system visualizes these regression models in a matrix view in combination with further features and quality information (D). The analyst can apply feature-transformations and offers preparation parameters that are used for feature partitioning (B). All interactions trigger recalculations of the regression model variants and quality metrics enable the analyst to evaluate these changes (E).

BoababView [37] (Figure 4-d) visualizes the structure of a decision tree in combination with data, feature, and quality information (D). The analyst can perform tree operations (e.g., split or merge nodes) and adapt specific parameters, such as split points values (C). The analyst can inspect and follow changes of the data flow within the tree, and evaluate the precision of the classifier at any stage (E).

EnsembleMatrix [42, 46] lets analysts build classifier ensembles by discovering several combination strategies (C). The system offers several confusion matrix visualizations for each classifier with a combined main matrix, as well as a linear classifier combination view (D). An extension of the tool further measures the accuracy of an ensemble classifier before and after user interaction to support analysts in their evaluation process (E).

Voyager [47] provides the analyst with visual recommendations for faceted browsing through a series of automatically generated visualizations (D) that match the underlying data's characteristics with user preferences.

DimStiller [48] allows analysts to design and validate several steps of a dimensionality reduction workflow (A–D). This approach makes it possible to compare alternative workflows and validate each step of the ML pipeline to identify which phases can be improved (E).

Table 1: Examples grouped by framework components (A–E).

Part	Examples
(A)	Adding class labels [43], adjusting & removing data [34], adding textual annotations [36], suggesting transformations [44]
(B)	Re-arrange data points [45], dimension loadings [34], term weightings [36], preparation parameters [35]
(C)	Parameter tuning [43], making model selections [35], building ensemble classifiers [46], defining constraints in a force-directed layout [36], tree operations [37]
(D)	Multiple linked views [34, 43], 2D-spatialization with direct interactions [36, 45], pre-built ML variants [35], tree-visualizations [37], (confusion) matrix [34, 37, 46], browsing visualizations [47]
(E)	Responsive visualization updates [34, 35, 37, 36, 45], ML workflow design [48], measuring interaction quality [42]

5. Human-Centered Machine Learning Loop

In this section, we focus more closely on the analyst's feedback loop, described as an iterative cycle of *Executions* and *Evaluations* (shown in Figure 3-E and in more detail in Figure 5). This loop "connects" an analysis system and an analyst whose ability to provide feedback depends on individual factors, as well as the visual interfaces of the system. We will outline individual human factors of the analyst, and elaborate on potential stages of action in more detail (Figure 5-left). Subsequently, we enumerate several analysis scenarios to illustrate types of feedback that can be provided by an analyst (Figure 5-right).

5.1. Stages of Action

We adopt Norman's *Stages of Action* [24] to distinguish two phases within the human-in-the-loop model, shown in Figure 5. This model describes the interplay and collaboration between the system and the analyst. While the previous section described details of the analysis system, we now shift our focus to the analyst, and describe the phases of *Execution* and *Evaluation* in more details.

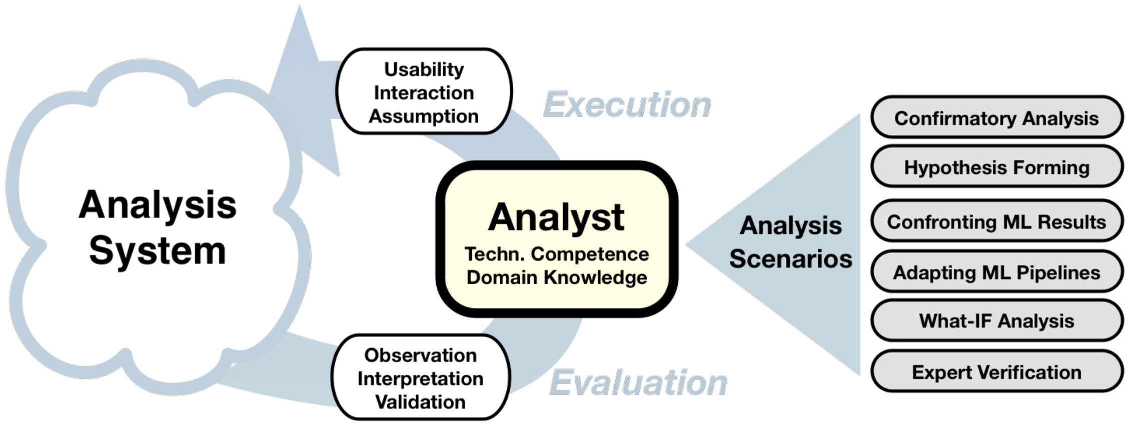

Figure 5: Human-centered ML process loop shown in more detail. The loop reveals important characteristics of analytic activity; the right hand side shows different analysis scenarios.

The Analyst. The analyst forms goals based on individual prior knowledge, and assumptions about the data/visual interface. The ability of analysts to provide feedback depends on factors such as technical competence (*e.g.*, expertise in data analysis, ML, mathematics, statistics, or visualization) and application domain knowledge (e.g., biology, business, digital humanities, or sports). Analysts may have highly diverse backgrounds and therefore varying level of skills and related capabilities. Consequently, they may provide different kinds of feedback and "take on different roles". Typically, data scientists, such as ML experts with strong mathematical skills, can train specialized ML models and techniques, but may miss significant anomalies in data generation or collection. Conversely, domain experts may be very aware of these details, but overlook important properties of models they might approach as off-the-shelf black boxes. To overcome this, in interdisciplinary research, ML experts, visualization experts, and data owners usually work in collaborative teams. In this case, visualization provides a common platform for communication. These differences and gaps between different types of users can be addressed by a *Liaison*, a person sharing language and knowledge from the application domain and from the visualization domain, with the goal of mediating communication issues [49].

Execution. Applying Norman's *Stages of Execution* to our interactive ML setting (1) the intentions to act may be based on assumptions about the ML model or the data at hand, (2) the sequence of actions describes the different ML pipeline adaptions, and (3) the actual execution is realized through the visual interface (or visualization) and heavily depends on its usability. As pointed out in the previous paragraph, analysts usually need user interfaces tailored to their individual capabilities. Visual metaphors and actions (e.g., moving points, or providing labels) need to be accessible and familiar. Command line interfaces and specific parameters are often operable for data scientists, however application domain experts often expect simple and intuitive user interfaces to provide feedback. Note that the visual interactions should faithfully reflect and translate the analysts assumptions to ML pipeline adaptions. If the analyst is not able to perform a desired action (e.g., because of poor usability of the user interface) there is a gap between human and machine, also known as "gulf of execution" [24].

Evaluation. Before the analyst is enabled to provide (further) feedback, he/she has to *Evaluate* the current state of the system. In our described VA/ML pipeline, changes made by the analyst or feedback given by the analyst (should) cause (1) observable reactions in the analysis system. These observations—in our context usually represented as visual patterns (e.g., groups, sequences, outliers)—have to be (2) interpreted by the analyst who can leverage his/her domain knowledge. Finally, the analyst has to (3) validate and verify the derived insights according to previous goals and assumptions. Visual interfaces should therefore allow the analyst to compare different states of the analysis system, by switching between visualization results before and after the computations. Animations and transitions between states or progressive/intermediate ML results may enhance the interpretability of complex ML models. Design studies have to be conducted in order to "bring the entire ML pipeline closer" to the domain experts mental models, language, and metaphors [50]. Interpretability is essential for evaluating the obtained results and also for providing further feedback in subsequent loops/iterations [51]. Note that misinterpretations may cause poor feedback and therefore impair the ML pipeline configuration. This gap between machine and the human is known as "gulf of evaluation" [24]. Especially in ML when the analyst is presented with a final result, it is often a challenge to find out "why" the result is not good enough. Several methods may be combined into complex pipelines, making it hard to assess the quality of the individual blocks.

5.2. Analysis Scenarios

This section enumerates six analysis scenarios illustrating a variety of strategies and feedback that can be incorporated in a visual interactive ML setup. Notice that some scenarios overlap, and can be combined or switched during an analysis session. Specifically, the first two scenarios (*confirmatory analysis* and *hypothesis forming*) can be seen as higher level analysis goals, in contrast with the latter four scenarios.

Confirmatory Analysis. An ML model is built on assumptions about the domain and data at hand. In an interactive ML session, analysts may make use of different ML types to model and confirm hypotheses. In this activity, they often correct and refine model parameters to more closely match their assumptions; they also may need to generate and collect evidence to either verify or falsify hypotheses [4]. Such evidence may be provided by statistical tests, or by inspecting visual patterns generated by a more complex ML algorithm. For example, a grouping of similar observations can be computed by clustering, or classification. However, analysts also have expectations about groupings and may need to check whether their assumptions are consistent with the ML results. In many cases, techniques do not fit "out of the box" and need to be refined by an expert.

Hypothesis Forming. Another analysis goal is to generate, form and refine hypotheses. In this case, the analysis is more exploratory, and ML models can be invoked and visualized to get broad overviews. Several unsupervised ML methods are effective for revealing certain structures that are otherwise hidden in data (e.g., feature selection, dimensionality reduction, clustering, outlier or novelty detection). Visualizations support the analyst in spotting patterns that can be investigated in more detail. Such patterns can be, for example, manifolds, outliers, sequences, clusters, or trends. Note that spotting patterns may be the result of pure serendipity during analysis. However, once a pattern has been spotted, an analyst generally needs to discover "why" the pattern exists, and consequently forms more concrete hypotheses and may switch to confirmatory analysis.

Confronting ML Results or Structures. In an iterative ML process, the analyst provides feedback about results to the ML pipeline. Domain experts, who are able to exploit domain knowledge, can effectively adapt ML results if they spot errors within visualizations that do not match their assumptions or prior knowledge. For example, they can re-organize automatically generated groups [52] or adjust class labels [53]. Furthermore, parameters or weights can be tuned to adapt ML structures to focus the analysis on specific features, or to determine the granularity of the ML algorithms (e.g., the number of clusters desired). Reorganizing points ("declaring distances") can correct distances between specific observations when they are known to the analyst (e.g., [45]).

Adapting ML Pipelines. Depending on the data and analysis task at hand, the ML pipeline may not be "comprehensive" enough and may require reconfiguration to accommodate additional ML computations or features of the dataset. ML models can be thought of as atomic components that can be combined and then require some meta-assessment, with the difficulty that validation faces combinatorial growth and can become intractable. An ML algorithm could, for example, require additional pre-processing, specific feature selections and transformations. If ML models become "too complex" some parts of the ML pipeline may need to be simplified or even removed (for example, in case of model overfitting).

"What-If"-Analysis. Interactions enable the analyst to experiment with an ML pipeline and observe how it reacts to changes or feedback. This may contribute to better understanding of how the model behaves, even without ML expertise. An example is to investigate how the final ML results are affected by adapting dimension loadings in a problem of dimensionality reduction (such as in iPCA [34]). In such a case, the analyst can identify which data items are affected and related to specific features. The same can be done with manipulating data observations. The analyst may examine what happens if a particular observation is present or absent in the data.

Expert Verification. ML models aim to detect structure and patterns in data, such as trends. However, in a real world use case, patterns have to be cross-checked based on "external" knowledge. One possibility is to apply the ML model to other external data sources to test whether the pattern recurs as in a kind of manual validation or test procedure. These data sources may be obtained from another database or repository. However, if such data is not available, the domain expert has to judge whether observed structures or patterns are plausible and useful. In this case, several domain experts may collaboratively discuss the outcome, or design further experiments.

6. Challenges & Opportunities

On the path toward systems that fully implement the proposed framework, we encounter several important research challenges that must be overcome. We identified five relevant challenges at the intersection of ML and visualization research. We will describe how joint research in these areas opens up novel opportunities to advance practical data analysis.

Designing Interaction for ML Adaption. A variety of ML algorithms, offering a broad set of design options and parameters, have been described in the literature. Yet, we find no generic way to interface ML with visualization. Current visual analytics systems are often restricted to working with a small set of ML techniques and parameters. Furthermore, within current interfaces, switching between ML models is likely to disrupt a human's sense of context in the analysis process. To address, this, new approaches are needed that support analysts in making sense of such model changes. In addition, existing examples such as ForceSpire and iPCA nicely illustrate how understandable direct interactions can be combined with model changes in a straightforward manner. Direct manipulation has proven effective and easy-to-learn for accessing computational tools [25]. It has, however, not been extensively explored in the context of ML so far. Often, ML models are designed for unique, static configurations, whereas in VA iterative refinement is needed. Mapping user inputs to more complex algorithmic actions along the entire ML pipeline remains an open problem. One key question is how to translate "simple" interactions within the visual interface to data manipulation, preprocessing, or ML model-adaption operations and combinations thereof. —*Opportunities:* Central to our conceptual framework is the idea that the underlying ML design options and meta-parameters, which usually cannot be optimized automatically, can often instead be steered by convenient, iterative user interactions. Accessible interactions and smooth transitions between different ML models will help analysts to develop intuition or form mental models [54] about the underlying data, as well as about the function or behavior of complex ML methods. Consider the case of switching between different ML models: at what point does the system realize—from user feedback—that the current ML model might not be the most appropriate one anymore? It could then suggest an alternative model, and smoothly transition to it. Instead of linear projection with PCA, it might, for instance, suggest a more complex nonlinear dimensionality reduction method like multidimensional scaling or t-SNE. Continuous model spaces [55, 56] contribute some preliminary ideas towards such solutions, which are dependent on the ML models' meta- or hyper-parameters and their interpretability. Further, more general ways to apply and adapt ML through expert feedback (*e.g.*, labeling or rating) would allow us to take advantage of a larger, more powerful set of ML methods. The previous examples demonstrate that there is vast space for future research, given the great variety of available ML techniques and their associated parameter spaces. A joint effort from both the ML and VA communities is needed to face this research challenge.

Guidance. Another major challenge is to adequately support application domain experts in steering the ML pipeline. Analysts can be overwhelmed by the wide range of ML models and parameters, along with the challenges of working with large data sets. Moreover, analysis problems are often incompletely defined, and change over time, resulting in a complex analysis process with much trial-and-error. Consequently, analysts may change, adapt, or switch tasks frequently. While analysts may bring crucial domain-dependent information to problem-solving, they often lack advanced programming skills and statistical expertise, and therefore require assistance and guidance (*e.g.*, by providing recommendations about operations on data and alternative models.) *—Opportunities:* It is important to better understand the tasks, practices, and stumbling blocks of domain experts (which likely differ from those of visualization or ML experts). Adopting a design study methodology is a viable approach towards gaining better understanding of such user characteristics [50] and providing appropriate guidance. Furthermore, enhanced measures and tools could point analysts to interesting data, parameterizations, and ML models through automatic recommendations. While many measures exist, both depicting data and perceptual characteristics (e.g., [57]), currently it is not well understood how they can be effectively exploited in interactive analytical processes. Consider a relevance feedback approach where the learning system retrieves a set of interesting visualizations based on iterative user feedback. In each iteration the analyst marks the presented results either as positive (interesting) or negative (uninteresting) [53]. How could the system detect if a pattern was spotted and the analysis task changes from overview to detail? Therefore, we envision the usage of analytic provenance which "captures the interactive exploration process and the accompanied human reasoning process during sensemaking" [41]. This information could guide the analysis process to meet the analysts' needs, which might be derived from their behavior. In the VA community, research has been carried out on capturing, visualizing, and reusing analytic provenance. However, more work is needed on modeling such information to shape or refine the overall analysis as well as specific ML methods. Doing this is an interesting research problem that will require expertise from the ML community.

Measuring Quality & Consistency. In the rich human-in-the-loop analysis process we envision, it is crucial to ensure both *ML model quality* and *visualization quality.* Yet, these two types of quality assurance do not always align well. For example, in a visual representation of a data embedding, after dimensionality reduction, there might be a trade-off between preservation of the original data structure, and readability of patterns, due to intrinsically high dimensionality. While measures have been developed that describe each of these aspects of quality in data analysis (*e.g.*, [55], [56], [57], and [58]), the challenge is to help analysts to find the right balance between them, so that meaningful analysis is enhanced. Beyond measuring ML and visualization quality, our framework suggests a third type of quality assessment, which is the *level of consistency* between the ML model and the analyst's expectations. The goal of data analysis is to extract reliable and relevant knowledge from data. Assuming that there exists some "ground truth" to back up such knowledge, it is the goal of ML and visualization to reveal it with high fidelity. At the same time, the user will have a priori knowledge and expectations, which in the ideal case should closely match what the analysis reveals. While an ML model will surely seek to accurately describe the data, essential pieces of information or context known by analysts may be unavailable to an algorithm. In this case, the set of ML assumptions may be incomplete, a challenge often encountered in exploratory data analysis. —*Opportunities:* To externalize this missing information, it is important to check consistency between what the ML model presents and what the analyst expects. If inconsistent, the analyst should either suspect a problem with the ML model and provide feedback about missing information, or accept that the expected patterns were not found in the data. If consistent, analysts usually conclude there is a confirmation of their expectation. Note, although consistency between human and machine is desirable, it does not guarantee correct reflection of the underlying ground truth in the data per se. Currently, consistency checks are often done manually. Automatic methods that systematically check consistency, highlight inconsistencies and recommend any needed remediation could help. Joint effort from both ML and VA communities is needed to enhance these measures, especially by combining and bridging them.

Handling Uncertainty. There are several stages in our framework where uncertainty might be dealt with explicitly. Uncertainty may arise from several sources of unreliability or vagueness, such as data described by probability density descriptions, missing data, or even systematic errors in modeling. Visualizations themselves can also introduce uncertainty, for example, due to resolution or contrast effects [59]. In our framework, uncertainty implicitly propagates through the pipeline and eventually may affect the analyst's trust-building process [60]. Properly describing, quantifying, and formally propagating uncertainty in all pipeline stages will be a major challenge in developing robust, effective tools. —*Opportunities:* Alternative visualizations can be generated and investigated to raise the analyst's awareness of the sources of uncertainty within the ML pipeline. Furthermore, analysts can be supported in (interactively) exploring, understanding and reducing uncertainty [61]. Better integration of uncertainty measures from data, preprocessing, ML models, and visualization can be expected to provide a holistic perspective and understanding of uncertainty. There is much previous work on visualization techniques to display data uncertainty of spatial data, such as volume or flow visualization [62, 63]. We find less work on uncertainty visualization of abstract data, such as high-dimensional data visualization [64]. As abstract data is typical in ML applications, there is a need for improved uncertainty visualization along the analytical pipeline outlined in our framework. A joint effort by ML and visualization researchers is needed to handle uncertainty within the entire pipeline.

ML-Vis Interoperability. A final challenge arises because most existing ML algorithms, toolkits and libraries were not designed to support interactive visualization. Scalability problems in computation may cause long delays that impede interaction; parameters may not be adaptable or visible; and relevant information (e.g., quality measures or internal ML structures) may be inaccessible. The ability to communicate these types of algorithmic information and to take advantage of them to construct better user interfaces is often described as "opening the black box" [17]. Especially in the case of direct human interactions, it is often difficult to speed up the ML computations enough to provide the desired responsive behavior of the visualization. Another challenge is to train ML techniques from interactive inputs, which typically are few in number. *—Opportunities:* The visualization community could benefit from ML algorithms and libraries that meet specific requirements, such as exposing intermediate or progressive results, and providing meaningful and interpretable parameters or handles to integrate them with interactive visualizations. Additional information, such as model structures, preprocessings, and quality information can be visualized. Recently, novel VA systems have been described that provide approximated or progressive computations with interactive and steerable visualizations (e.g., progressive t-SNE [65] or progressive PCA [66]). These examples suggest considerable potential for an expanded, generalized integration of ML with visualization. In summary, both communities could gain much from library and framework designs informed by the requirements of both interactive visualization and ML.

7. Conclusions

We propose a framework that characterizes important forms of interaction that are possible with ML components in a VA process. In general, such interaction offers considerable potential for improved support of ML interpretability, understandability, evaluation and refinement. We found that a multidisciplinary perspective can bridge the gaps between automated ML methods and human reasoning. The proposed framework offers a balanced perspective on the design and configuration of the analytic pipeline, incorporating important aspects of both automated techniques and human interaction. Of course, current VA tools and ML components pose many interesting challenges for future work at the intersection of visualization and ML. To address these challenges, closer collaboration between ML and visualization researchers will be vital.

The conceptual framework proposed in this report was developed over a period of 1.5 years. During this process, this framework was iteratively validated and refined, comparing it with previous models and real-world systems as illustrated in Section 4. By doing this, we were able to verify how well the framework fits such models and systems. This process lead to both extending the framework to add missing features, and summarizing and generalizing common aspects of it across multiple example systems and existing models. The main benefit was to ground our theoretical framework in real-world systems and applications.

A full evaluation of any theoretical framework, assessing how useful it will be to others, is beyond the scope of a single paper. An effective theoretical model should be expected to stand the test of time, and to repeatedly demonstrate its validity [67]. It is our hope that empirical evaluations of future systems and solutions implementing the proposed framework will add evidence to prove its worth [68]. By guiding researchers and practitioners in the design of novel data analysis solutions, in the long run its “real” value can be established.

References

[1] D. A. Keim, J. Kohlhammer, G. P. Ellis, F. Mansmann, Mastering the Information Age - Solving Problems with Visual Analytics, Eurographics Association, 2010.

[2] G. L. Tam, V. Kothari, M. Chen, An analysis of machine- and human-analytics in classification, IEEE Trans. Visualization and Computer Graphics (TVCG) PP (99) (2016) 1–1. doi:10.1109/TVCG.2016.2598829.

[3] A. Endert, C. North, R. Chang, M. Zhou, Toward usable interactive analytics: Coupling cognition and computation, in: Proc. ACM SIGKDD Workshop Interactive Data Exploration and Analytics (IDEA), 2014.

[4] D. Sacha, A. Stoffel, F. Stoffel, B. Kwon, G. Ellis, D. Keim, Knowledge generation model for visual analytics, IEEE Trans. Visualization and Computer Graphics (TVCG) 20 (12) (2014) 1604–1613. doi:10.1109/TVCG.2014.2346481.

[5] S. Amershi, M. Cakmak, W. Knox, T. Kulesza, Power to the people: The role of humans in interactive machine learning, AI Magazine 35 (4) (2014) 105–120.

[6] D. A. Keim, T. Munzner, F. Rossi, M. Verleysen, Bridging Information Visualization with Machine Learning (Dagstuhl Seminar 15101), Dagstuhl Reports 5 (3) (2015) 1–27. doi:10.4230/DagRep.5.3.1.

[7] D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, D. Weiskopf, S. C. North, D. A. Keim, Human-Centered Machine Learning Through Interactive Visualization: Review and Open Challenges, in: 24th European Symposium on Artificial Neural Networks, ESANN, 2016, pp. 641–646.

[8] S. K. Card, J. D. Mackinlay, B. Shneiderman, Readings in information visualization - using vision to think, Academic Press, 1999.

[9] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From data mining to knowledge discovery in databases, AI Magazine 17 (3) (1996) 37–54.

[10] P. Pirolli, S. Card, The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis, in: Proc. Int. Conf. Intelligence Analysis, Vol. 5, 2005, pp. 2–4.

[11] W. A. Pike, J. T. Stasko, R. Chang, T. A. O'Connell, The science of interaction, Information Visualization 8 (4) (2009) 263–274. doi:10.1057/ivs.2009.22.

[12] A. Endert, M. S. Hossain, N. Ramakrishnan, C. North, P. Fiaux, C. Andrews, The human is the loop: new directions for visual analytics, Journal of Intelligent Information Systems 43 (3) (2014) 411–435.

[13] M. Chen, A. Golan, What may visualization processes optimize?, CoRR abs/1506.02245.

[14] M. Brehmer, T. Munzner, A multi-level typology of abstract visualization tasks, IEEE Trans. Visualization and Computer Graphics (TVCG) 19 (12) (2013) 2376–2385. doi:10.1109/TVCG.2013.124.

[15] T. von Landesberger, S. Fiebig, S. Bremm, A. Kuijper, D. W. Fellner, Interaction taxonomy for tracking of user actions in visual analytics applications, in: Handbook of Human Centric Visualization, Springer, 2014, pp. 653–670.

[16] E. Bertini, D. Lalanne, Surveying the complementary role of automatic data analysis and visualization in knowledge discovery, in: Proc. ACM SIGKDD Workshop Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration, 2009, pp. 12–20. doi:10.1145/1562849.1562851.

[17] T. Muhlbacher, H. Piringer, S. Gratzl, M. Sedlmair, M. Streit, Opening the black box: Strategies for increased user involvement in existing algorithm implementations, IEEE Trans. Visualization and Computer Graphics (TVCG) 20 (12) (2014) 1643–1652. doi:10.1109/TVCG.2014.2346578.

[18] G. Jawaheer, P. Weller, P. Kostkova, Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback, ACM Trans. Interactive Intelligent Systems 4 (2) (2014) 8:1–8:26. doi:10.1145/2512208.

[19] P. Melville, V. Sindhwani, Recommender systems, in: C. Sammut, G. I. Webb (Eds.), Encyclopedia of Machine Learning, Springer, 2010, pp. 829–838. doi:10.1007/978-0-387-30164-8.

[20] F. Ricci, L. Rokach, B. Shapira, P. B. Kantor, Recommender Systems Handbook, 1st Edition, Springer-Verlag, 2010.

[21] D. Cohn, L. Atlas, R. Ladner, Improving generalization with active learning, Machine Learning 15 (2) (1994) 201–221. doi:10.1023/A:1022673506211.

[22] B. Settles, Active Learning, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2012. doi:10.2200/S00429ED1V01Y201207AIM018.

[23] A. Groce, T. Kulesza, C. Zhang, S. Shamasunder, M. M. Burnett, W. Wong, S. Stumpf, S. Das, A. Shinsel, F. Bice, K. McIntosh, You are the only possible oracle: Effective test selection for

end users of interactive machine learning systems, IEEE Trans. Software Engineering 40 (3) (2014) 307–323. doi:10.1109/TSE.2013.59.

[24] D. Norman, The Design of Everyday Things, Basic Books, 2002.

[25] B. Shneiderman, Direct manipulation: A step beyond programming languages, IEEE Computer 16 (8) (1983) 57–69. doi:10.1109/MC.1983.1654471.

[26] S. Kandel, J. Heer, C. Plaisant, J. Kennedy, F. van Ham, N. Riche, C. Weaver, B. Lee, D. Brodbeck, P. Buono, Research directions in data wrangling: Visualizations and transformations for usable and credible data, Information Visualization 10 (4) (2011) 271–288. doi:10.1177/1473871611415994.

[27] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.

[28] M. Kaden, M. Lange, D. Nebel, M. Riedel, T. Geweniger, T. Villmann, Aspects in classification learning - Review of recent developments in Learning Vector Quantization, Foundations of Computing and Decision Sciences 39 (2) (2014) 79–105.

[29] P. Schneider, B. Hammer, M. Biehl, Adaptive relevance matrices in learning vector quantization, Neural Computation 21 (2009) 3532–3561.

[30] K. Bunte, P. Schneider, B. Hammer, F.-M. Schleif, T. Villmann, M. Biehl, Limited rank matrix learning, discriminative dimension reduction and visualization, Neural Networks 26 (1) (2012) 159–173.

[31] A. Rakotomamonjy, F. R. Bach, S. Canu, Y. Grandvalet, Simplemkl, Journal of Machine Learning Research 9 (Nov) (2008) 2491–2521.

[32] B. Efron, R. Tibshirani, Improvements on cross-validation: The .632+ bootstrap method, Journal of the American Statistical Association 92 (438) (1997) 548–560. doi:10.2307/2965703.

[33] H. Fröhlich, O. Chapelle, B. Schölkopf, Feature selection for support vector machines by means of genetic algorithm, in: Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, 2003, pp. 142–148.

[34] D. H. Jeong, C. Ziemkiewicz, B. D. Fisher, W. Ribarsky, R. Chang, iPCA: An interactive system for PCA-based visual analytics, Computer Graphics Forum 28 (3) (2009) 767–774. doi:10.1111/j.1467-8659.2009.01475.x.

[35] T. Mühlbacher, H. Piringer, A partition-based framework for building and validating regression models, IEEE Trans. Visualization and Computer Graphics (TVCG) 19 (12) (2013) 1962–1971. doi:10.1109/TVCG.2013.125.

[36] A. Endert, P. Fiaux, C. North, Semantic interaction for visual text analytics, in: ACM SIGCHI Conf. Human Factors in Computing Systems (CHI), 2012, pp. 473–482.

[37] S. van den Elzen, J. van Wijk, Baobabview: Interactive construction and analysis of decision trees, in: IEEE Conf. Visual Analytics in Science and Technology (VAST), 2011, pp. 151–160. doi:10.1109/VAST.2011.6102453.

[38] M. Aupetit, M. Sedlmair, Sepme: 2002 new visual separation measures, in: IEEE Pacific Visualization Symp., 2016, pp. 1–8.

[39] R. J. Crouser, R. Chang, An affordance-based framework for human computation and human-computer collaboration, IEEE Trans. Visualization and Computer Graphics (TVCG) 18 (12) (2012) 2859–2868. doi:10.1109/TVCG.2012.195.

[40] J. Boy, L. Eveillard, F. Détienne, J. Fekete, Suggested interactivity: Seeking perceived affordances for information visualization, IEEE Trans. Visualization and Computer Graphics (TVCG) 22 (1) (2016) 639–648. doi:10.1109/TVCG.2015.2467201.

[41] K. Xu, S. Attfield, T. J. Jankun-Kelly, A. Wheat, P. H. Nguyen, N. Selvaraj, Analytic provenance for sensemaking: A research agenda, IEEE Computer Graphics and Applications 35 (3) (2015) 56–64. doi:10.1109/MCG.2015.50.

[42] A. Kapoor, B. Lee, D. S. Tan, E. Horvitz, Learning to learn: Algorithmic inspirations from human problem solving, in: Proc. of the Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012.

[43] B. Höferlin, R. Netzel, M. Höferlin, D. Weiskopf, G. Heidemann, Inter-active learning of ad-hoc classifiers for video visual analytics, in: IEEE Conf. Visual Analytics in Science and Technology (VAST), 2012, pp. 23–32. doi:10.1109/VAST.2012.6400492.

[44] S. Kandel, A. Paepcke, J. M. Hellerstein, J. Heer, Wrangler: interactive visual specification of data transformation scripts, in: ACM SIGCHI Conf. Human Factors in Computing Systems (CHI), 2011, pp. 3363–3372. doi:10.1145/1978942.1979444.

[45] E. Brown, J. Liu, C. Brodley, R. Chang, Dis-function: Learning distance functions interactively, in: IEEE Conf. Visual Analytics in Science and Technology (VAST), 2012, pp. 83–92. doi:10.1109/VAST.2012.6400486.

[46] J. Talbot, B. Lee, A. Kapoor, D. Tan, Ensemblematrix: interactive visualization to support machine learning with multiple classifiers, in: ACM SIGCHI Conf. Human Factors in Computing Systems (CHI), 2009, pp. 1283–1292. doi:10.1145/1518701.1518895.

[47] K. Wongsuphasawat, D. Moritz, A. Anand, J. D. Mackinlay, B. Howe, J. Heer, Voyager: Exploratory analysis via faceted browsing of visualization recommendations, IEEE Trans. Visualization and Computer Graphics (TVCG) 22 (1) (2016) 649–658. doi:10.1109/TVCG.2015.2467191.

[48] S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner, T. Möller, Dimstiller: Workflows for dimensional analysis and reduction, in: IEEE Conf. Visual Analytics in Science and Technology (VAST), 2010, pp. 3–10. doi:10.1109/VAST.2010.5652392.

[49] S. Simon, S. Mittelstdt, D. A. Keim, M. Sedlmair, Bridging the Gap of Domain and Visualization Experts with a Liaison, in: E. Bertini, J. Kennedy, E. Puppo (Eds.), Eurographics Conference on Visualization (EuroVis) - Short Papers, The Eurographics Association, 2015. doi:10.2312/eurovisshort.20151137.

[50] M. Sedlmair, M. Meyer, T. Munzner, Design study methodology: Reflections from the trenches and the stacks, IEEE Trans. Visualization and Computer Graphics (TVCG) 18 (12) (2012) 2431–2440. doi:10.1109/TVCG.2012.213.

[51] A. Vellido, J. D. Martín-Guerrero, P. J. G. Lisboa, Making machine learning models interpretable, in: 20th European Symposium on Artificial Neural Networks, ESANN 2012, 2012.

[52] S. Amershi, J. Fogarty, D. Weld, ReGroup: Interactive machine learning for on-demand group creation in social networks, in: ACM SIGCHI Conf. Human Factors in Computing Systems (CHI), ACM, 2012, pp. 21–30. doi:10.1145/2207676.2207680.

[53] M. Behrisch, F. Korkmaz, L. Shao, T. Schreck, Feedback-driven interactive exploration of large multidimensional data supported by visual classifier, in: IEEE Conf. Visual Analytics in Science and Technology (VAST), 2014, pp. 43–52. doi:10.1109/VAST.2014.7042480.

[54] Z. Liu, J. Stasko, Mental models, visual reasoning and interaction in information visualization: A top-down perspective, IEEE Trans. Visualization and Computer Graphics (TVCG) 16 (6) (2010) 999–1008. doi:10.1109/TVCG.2010.177.

[55] J. Venna, J. Peltonen, K. Nybo, H. Aidos, S. Kaski, Information retrieval perspective to nonlinear dimensionality reduction for data visualization, Journal of Machine Learning Research 11 (2010) 451–490.

[56] J. A. Lee, E. Renard, G. Bernard, P. Dupont, M. Verleysen, Type 1 and 2 mixtures of Kullback-Leibler divergences as cost functions in dimensionality reduction based on similarity preservation, Neurocomputing 112 (2013) 92–108. doi:10.1016/j.neucom.2012.12.036.

[57] E. Bertini, A. Tatu, D. A. Keim, Quality metrics in high-dimensional data visualization: An overview and systematization, IEEE Trans. Visualization and Computer Graphics (TVCG) 17 (12) (2011) 2203–2212.

[58] M. Sips, B. Neubert, J. P. Lewis, P. Hanrahan, Selecting good views of high-dimensional data using class consistency, Computer Graphics Forum 28 (3) (2009) 831–838.

[59] T. Zuk, S. Carpendale, Visualization of uncertainty and reasoning, in: Proceedings of the 8th International Symposium on Smart Graphics, SG '07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 164–177. doi:10.1007/978-3-540-73214-3_15.

[60] D. Sacha, H. Senaratne, B. C. Kwon, G. P. Ellis, D. A. Keim, The role of uncertainty, awareness, and trust in visual analytics, IEEE Trans. Visualization and Computer Graphics (TVCG) 22 (1) (2016) 240–249. doi:10.1109/TVCG.2015.2467591.

[61] C. D. Correa, Y. Chan, K. Ma, A framework for uncertainty-aware visual analytics, in: IEEE Conf. Visual Analytics in Science and Technology (VAST), 2009, pp. 51–58. doi:10.1109/VAST.2009.5332611.

[62] A. Pang, C. M. Wittenbrink, S. K. Lodha, Approaches to uncertainty visualization, The Visual Computer 13 (8) (1997) 370–390.

[63] K. Brodlie, R. A. Osorio, A. Lopes, A review of uncertainty in data visualization, in: J. Dill, R. Earnshaw, D. Kasik, J. Vince, P. C. Wong (Eds.), Expanding the Frontiers of Visual Analytics and Visualization, Springer, 2012, pp. 81–109.

[64] M. Skeels, B. Lee, G. Smith, G. G. Robertson, Revealing uncertainty for information visualization, Information Visualization 9 (1) (2010) 70–81.

[65] N. Pezzotti, B. Lelieveldt, L. van der Maaten, T. Hollt, E. Eisemann, A. Vilanova, Approximated and user steerable tsne for progressive visual analytics, IEEE Trans. Visualization and Computer Graphics (TVCG) PP (99) (2016) 1–1. doi:10.1109/TVCG.2016.2570755.

[66] C. Turkay, E. Kaya, S. Balcisoy, H. Hauser, Designing progressive and interactive analytics processes for high-dimensional data analysis, IEEE Trans. Visualization and Computer Graphics (TVCG) 23 (1) (2017) 131–140. doi:10.1109/TVCG.2016.2598470.

[67] K. Popper, The logic of scientific discovery, Routledge, 2005.

[68] A. Dix, Theoretical analysis and theory creation, Research Methods in Human-Computer Interaction (2008) 175–195.