Feedback-Driven Interactive Exploration of Large Multidimensional Data Supported by Visual Classifier

Michael Behrisch *Student Member, IEEE*, Fatih Korkmaz, Lin Shao and Tobias Schreck *Member, IEEE*

Abstract—
The extraction of relevant and meaningful information from multivariate or high-dimensional data is a challenging problem. One reason for this is that the number of possible representations, which might contain relevant information, grows exponentially with the amount of data dimensions. Also, not all views from a possibly large view space, are potentially relevant to a given analysis task or user. Focus+Context or Semantic Zoom Interfaces can help to some extent to efficiently search for interesting views or data segments, yet they show scalability problems for very large data sets. Accordingly, users are confronted with the problem of identifying *interesting views*, yet the manual exploration of the entire view space becomes ineffective or even infeasible. While certain quality metrics have been proposed recently to identify potentially interesting views, these often are defined in a heuristic way and do not take into account the application or user context. We introduce a framework for a feedback-driven view exploration, inspired by relevance feedback approaches used in Information Retrieval. Our basic idea is that users iteratively express their notion of interestingness when presented with candidate views. From that expression, a model representing the user's preferences, is trained and used to recommend further interesting view candidates. A decision support system monitors the exploration process and assesses the relevance-driven search process for convergence and stability. We present an instantiation of our framework for exploration of Scatter Plot Spaces based on visual features. We demonstrate the effectiveness of this implementation by a case study on two real-world datasets. We also discuss our framework in light of design alternatives and point out its usefulness for development of user- and context-dependent visual exploration systems.

Index Terms—View Space Exploration Framework, Interesting View Problem, Relevance Feedback, User Preference Model

1 Introduction

Our current data-agnostic society is driven by the prevalent perception that most data contains valuable information, which can be retrieved in a later information retrieval process. To this end, all kinds of data are stored and analyzed. The business consultancy McKinsey even forecasts that the "data scientist" will become one of the most important jobs in the US in the coming decade [24]. While the collected data may be rich in information, it is still highly challenging identifying appropriate views on the data sets. As an example, an n-dimensional numeric data set allows to render $(n \times (n-1)/2)$ distinctive views only by using a projection onto two distinctive dimension axis. This spans a large exploration space in which interesting views need to be identified. To make matters worse, the most valuable data views exists in relation with the users' current tasks, intentions, and current context.

A range of approaches to deal with the *interesting view problem* were developed over the years. For example, *Focus+Context* systems [28, 3] lead the users in an overview to areas of interest and let them explore these areas with drill-down mechanisms. Focus+Context systems are an established and approved method, but often tend to be expert systems restricted to specific data set characteristics and/or user interactions. These systems potentially introduce misleading abstractions, which ultimately can lead to wrong exploration paths. Semantic zoom interfaces [5] help also to deal with this problem. Here, the user explores the data set at varying levels of abstraction/detail, starting with a highly aggregated version of the underlying data. The more the user "zooms" into the data, the more details become assessable. Alternatively, cluster-based navigation systems partition the exploration space into a range of distinctive clusters that are represented by a small amount of prototypes. Choosing the prototypes relates directly to the interesting views problem. A further well-understood, yet simple, approach to tackle the interesting views problem is to focus on faceted search algorithms that operate on the available meta data. This requires a manual annotation and insertion of meta data, which is often prone to errors or missing values.

Directly related to the interesting views problem is that a *query formulation* [23] on complex data sets is difficult. This is primarily due to the fact that the collected data sets are multivariate and high-dimensional in nature. To tackle this problem, novel querying mechanisms, such as query-by-example or sketch-based interfaces were developed. Here, the systems rely on the assumption that users have a priori an initial understanding of the interesting patterns under investigation. This explicit definition of interest is time consuming, particularly if interest rules need to be updated during the exploration process.

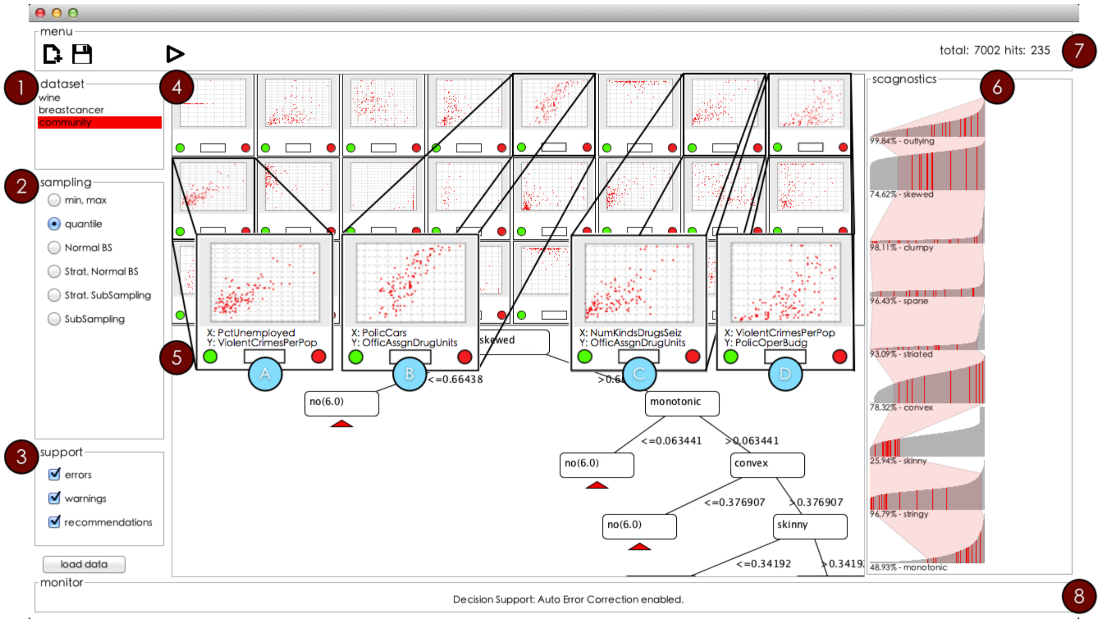
In this paper we present a novel approach to the interesting view problem, which focuses on the interplay between the user and an automatic decision-support system. In an iterative work flow the user assesses whether a set of presented views are of interest or not. These views can be arbitrary, but suitable, visualizations for the high-dimensional data exploration task at hand. A classification system learns from the previous user decisions, while notifying the user in case of potentially wrong decision paths and major decision path divergence. The general idea is inspired by multimedia retrieval approaches, where the user's explicit relevance feedback on retrieval results is used to recommend additional previously unseen results [23]. In contrast to the major work in this field, the presented relevance feedback mechanism is incorporated into a feedback loop, which adapts to the earlier user decisions.

Our approach relies on the basic assumption that for most and even complex data visualizations a comprehensive set of feature vector descriptors can be found, either in the data-, in the image space or in a combination of the latter, that can be mapped to its analytic benefit.

The outlined approach has to be seen as a framework for an interactive relevance feedback driven data exploration process. One of the benefits of the framework is that a great variety of design alternatives can be applied without changing the fundamental approach. We are showing one reference implementation of the framework by using

- *Michael Behrisch is with the Universität Konstanz, Germany. E-mail: michael.behrisch@uni-konstanz.de.*
- *Fatih Korkmaz is with the Universität Konstanz, Germany. E-mail: fatih.korkmaz@uni-konstanz.de.*
- *Lin Shao is with the Universität Konstanz, Germany. E-mail: lin.shao@uni-konstanz.de.*
- *Tobias Schreck is with the Universität Konstanz, Germany. E-mail: tobias.schreck@uni-konstanz.de.*

Manuscript received 31 March 2013; accepted 1 August 2013; posted online 13 October 2013; mailed on 4 October 2013.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.



○ min, max
● quantile
○ Normal BS
○ Strat. Normal BS
○ Strat. SubSampling
○ SubSampling

☑ errors
☑ warnings
☑ recommendations

Decision Support: Auto Error Correction enabled.

Fig. 1. The user interacts in the View Space Explorer by choosing relevant or irrelevant examples (4) from a small sample set. An incremental decision tree visualization (5) and a feature tube visualization (6) help to assess the exploration convergence. Specific decision support intervention points can be enabled/disabled in (3). Additional decision support notifications are shown in (8).

an incremental decision-tree classification to guide the user in a large scatter plot exploration space. However, it has to be mentioned that we are not restricting ourselves to scatter plot visualizations, but allow any type of visualization technique as long as a descriptive feature vector space can be found.

The remainder of the paper is structured as follows. In Section 2 we discuss related work. In Section 3 we introduce the general idea of the interactive relevance feedback driven data exploration framework. The following Section 4 describes our instantiation of the exploration framework and discusses the design decisions and alternatives. Section 5 details on the implementation of the decision support system and its implications, while Section 6 shows the visual interface for the view space exploration. Next, in Section 7 we present the results of two case studies on real-life data. In Section 8 we discuss limitations and possible extensions. Finally, Section 9 concludes the paper.

2 Related Work

Our work relates to interactive and automatic approaches for view selection, relevance-driven information retrieval, and systems which capture user feedback to guide the analysis process.

2.1 Interest-Driven Data Filtering for Visual Analysis

Methods for visual data analysis need to handle increasingly large data sets. As the data size grows, so does the space of data views, which are possible, given large data spaces and view parameters. Then, analysts run risk of overlooking interesting views if relying only on interactive navigation. To this end, intelligent methods for compressing and filtering data for potential patterns of interest has recently become a research focus. Overview-based approaches aim to generate effective layouts over many candidate data portions, to efficiently spot patterns of interest. Examples include the Value-and-Relation display [36], which lays out pixel-oriented views based on their data similarity. Another example is [33], where many time series are shown by small glyphs which are layed out based on data similarity.

Besides overview approaches, automatic filtering of views for potential structures of interest has been proposed. For scatter plots, the Scagnostics approach [34] automatically analyzes structures in scatter plots, which can be used to rank and filter. Recently, a clustering-based overview approach was presented in the ScagExplorer [8]. In case class information is given, scatter plots can be filtered for discriminative views by class consistency measures [29]. Also, projection pursuit approaches, such as initially presented by Friedman and Tukey [13], try to identify interesting 2D subspaces in high-dimensional data (mostly depicted by scatter plot views). Further heuristic interestingness filters for Scatter- and Parallel Coordinate plots have been discussed in [31, 9] and may narrow down the potentially large search space for high-dimensional data. In [32], an explorative overview of subspaces contained in high-dimensional data based on mutual differences and clustering quality properties was introduced.

2.2 Relevance-Driven and Image-Based Retrieval

In Information Retrieval, similar to Information Visualization, users search for relevant information, but often without being able to precisely specify the pattern they are looking for. In context of document retrieval, relevance feedback [2] allows to incrementally refine the user query. Based on a set of example documents, users assign a degree of relevance on them, based on the context of their information need. This assignment information in turn is used to iterate the search, e.g., by query term expansion or by weighting of query terms, based on the subset of relevant documents. This mechanism abstracts from the specific query formulation by the user, but may implicitly capture the user information need. Relevance feedback methods have also been intensively applied in content-based image retrieval [26, 30] and shown to improve the retrieval performance.

Many image retrieval systems so far rely on low-level image features, such as color histograms, edge histograms, or texture measures [10], which are heuristically combined to form distance functions. Relevance feedback methods for image retrieval may operate on these low-level feature representations in various ways. One option is to construct a new query vector by averaging the feature vectors of all image examples marked by the user as relevant. Another option is to train a classifier (e.g., SVM or Decision Tree [14]) from the set of relevance information provided by the user.

2.3 Relevance-Driven Analytics and Distinction of our Work

According to our observation, the majority of Visual Analytics approaches which incorporate interest-driven data filtering rely either on a) fixed heuristics for fully automatic filtering, or b) on fully interactive filter specification by users. However, fixed heuristics may not

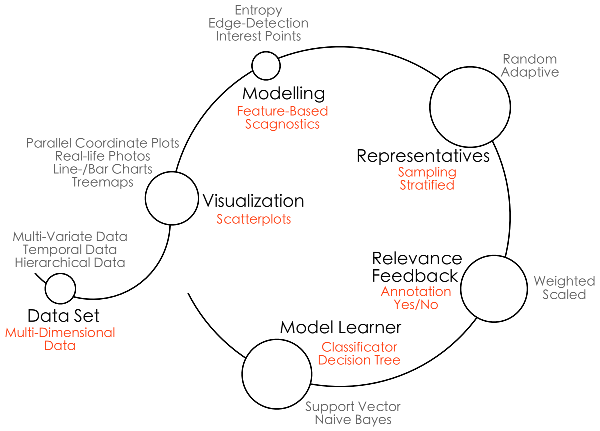


Fig. 2. The interaction flow in the Feedback-Driven View Exploration Framework: The user chooses a dataset of interest. A meaningful visualization type is selected (automatically or manually). The underlying data is described by means of feature descriptors. A range of representatives are shown to the user, which are interactively tagged for their relevance in the exploration process. A model learner tries to reflect the user's preferences and shows a new representatives set to the user.

necessarily map to a given users' information need, which may depend on data and context. Moreover, fully interactive search may not be feasible due to large search spaces. Surprisingly few works provide user-adaptive data filtering heuristics. In [22], intelligent visual analytics queries are proposed. The user marks a section within a given visualization as interesting; the system then computes certain distribution measures given in the data section, and automatically retrieves similar data segments from a larger database. The assumption is that the additionally retrieved data will add to the user information need. In [16], user data navigation is supported by a Bayes classification approach. The method learns to distinguish between interesting and uninteresting data sections while users pan and zoom an information landscape. The classifier is then utilized to suggest navigation paths of interest to a given user.

Two further recent works exploit user interaction to improve the analysis process. In [4], users interact by with the marks in a 2D projection of high-dimensional data, to express their notion of data correspondences. This input is used to adapt the data similarity function and re-project the data. Along similar lines, the approach in [12] allows users to interact with the positioning of documents in a 2D document landscape collection, to express document-level relationships. The system then learns and highlights the document terms which are most descriptive of the expressed document relationships.

In our approach, we apply ideas of relevance feedback-driven image retrieval to the problem of exploring large view spaces. Based on user examples, a decision tree is trained to identify additional interesting views based on Scagnostics features. We instantiate the approach for scatter plots described with Scagnostics features. Our approach is novel in that we a) introduce the concept of relevance feedback to scatter plot exploration, and b) that we make explicit the gained knowledge by a decision tree, which is used to guide and monitor the user exploration process. In that, our approach is related to [16] where a Bayes classifier is used to navigate a 2D information landscape. Our approach differs from [4, 12] in that the latter works consider user feedback in one single 2D view of the data, which is continuously updated. We here aim at retrieving sets of relevant 2D views in an iterative process. Furthermore, we do not update a data similarity function but use a Decision Tree classifier to capture user feedback and expressed interestingness relationships.

3 A Framework for Feedback-Driven View Exploration

The basic idea of a feedback-driven view exploration approach is to put users into a steering position to determine what they want. Figure 2 outlines the main work flow in the semi-automated exploration process. In a normal sequence of actions the user chooses a data set under investigation and decides for a meaningful visualization to assess the underlying data. The framework uses an appropriate feature descriptor from the data- and/or the image space to represent the data. The resulting feature vectors are the basis for the visualizations. In case of a feature descriptor operating on the data space characteristics, such as the data distribution or compressibility, will be represented. An example would be to measure the convexity of a scatter plot. Image features will be used to reflect the visualization's –or depiction's– characteristics. An example would be to measure the number of interest points for a real-world image. For the overall exploration the choice of the feature descriptor is crucial, since every descriptor is only capable of reflecting certain characteristics of the underlying data.

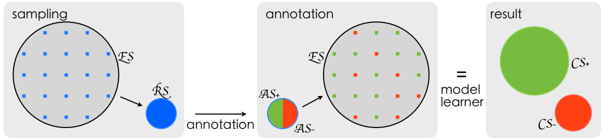


Fig. 3. Four different sets are distinguished in the approach: (1) The exploration set $\mathcal{ES}$ contains all possible views. (2) A sampled version of the exploration set will be presented to the user ($\mathcal{RS}$). (3) The user annotates this set for interesting, respectively uninteresting, views ($\mathcal{AS}^+$ versus $\mathcal{AS}^-$). (4) A classifier learns a mapping of the exploration set into potentially interesting views ($\mathcal{CS}^+$), respectively uninteresting views ($\mathcal{CS}^-$).

As Figure 3 depicts, from a potentially very large exploration set, denoted as $\mathcal{ES}$, only a limited amount of visualizations can be presented initially to the user. We will denote this subset as the representation set $\mathcal{RS}$. The choice of the items in $\mathcal{RS}$ can be random, deterministic, or iteratively adaptable (cf. Section 5). In the general feedback-driven view exploration framework the representation choice adapts according to the user's decisions. In an exploratory search phase a uniformly distributed sample should be made available, while in a confirmatory search only subpopulations of $\mathcal{ES}$ need to be presented. Generally, users might not be able to manually assess the entire data set. Hence, after a broad beginning only parts of the exploration space will be presented to the user. From $\mathcal{RS}$ the users can either choose visualizations of interest or express their dislike. Thus, an implicit knowledge gets explicitly available and accessible to the framework. A model learner is used to reflect the expressed user preferences by classifying the unseen items in $\mathcal{ES}$ as potentially relevant, denoted as $\mathcal{CS}^+$, or potentially irrelevant, denoted as $\mathcal{CS}^-$. We can assess the model learner's (un-)certainty in the classification. Relevant and irrelevant items can be matched to both classification sets $\mathcal{CS}^+$and $\mathcal{CS}^-$ to find visualizations with an (un-)certain interestingness mapping.

The task is now to find a good mapping $f : \mathcal{ES} \mapsto \mathcal{RS}$, such that the user on the one hand will find interesting patterns and on the other hand is still able to explore the dataset without loosing too many details. A secondary goal is to let the user iterate only a few times through the feedback loop.

Another basic idea of the feedback-driven view exploration framework is that user decisions should have an impact on the exploration. Hence, the task of a *decision support system*, as described in Section 5, is to assess the stability and convergence of the exploration path. Figure 4 outlines potential intervention points in the feedback-driven view exploration framework. As an example, the choice of the data set implies a (feature-based) modeling of the data. While this choice might be appropriate in an early exploration phase, it might be too restrictive, respectively broad, in the later phases. Thus, it might be beneficial to switch from a scatter plot visualization view to a parallel coordinate visualization with an appropriate modeling scheme. One heuristic for this recommendation could be that the exploration path stays rather unspecific and does not converge even after a number of iterations. Another intervention point is the outcome of the model learner. Here, the decision support acts as a supervision instance, which, e.g, allows assessing the certainty of the classification subsystem.

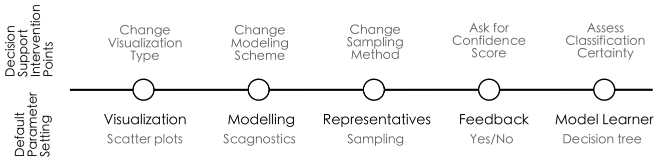


Fig. 4. The Decision Support can change essential parameters in the Relevance Feedback Driven View Exploration Framework if the exploration convergence stagnates: For example, it can recommend switching to a more appropriate visualization type with an appropriate modeling scheme or ask for a confidence score if the user annotation is obviously misleading.

4 Exemplified Instantiation of Feedback-Driven View Exploration Framework

In the following section we want to present one instantiation of the general semi-automatic exploration work flow from Section 3. In each of the following sections we will outline the overall idea for the respective work flow step, describe the current implementation and reflect our design rationale by describing alternatives and further prospects.

4.1 Visualization

The general idea of the *Visualization* step is to illustrate the given exploration set ($\mathcal{ES}$) in a reasonable manner. This step initiates our framework's interaction pipeline in Figure 2. The choice of the visualization technique is important and depends on the given data set. Effective visualizations help in the decision-making process, while reading ineffective visualizations can be time-consuming and potentially leads to wrong decisions.

In our exemplified instantiation of the feedback-driven view exploration framework we use *scatter plot visualizations* to represent a continuous high-dimensional data set. The choice of this visualization has several reasons. First of all, scatter plots prove to be powerful and intuitive visualizations for user decisions. They are used in a large variety of domains and are familiar to most users. Second, we can separate the high-dimensional data set into individual plots allowing the user to judge the importance over all pairwise dimension combinations.

As mentioned above, the choice of the visualization technique depends on the data set under consideration. In our case, scatter plots are appropriate, but in case of other data types, such as temporal, hierarchical or textual data, alternative visualization techniques are better suited for the view space exploration in our pipeline. To name just a few alternative examples, line charts are suitable for temporal data, treemaps can represent hierarchical data, and word clouds can be used to depict text content.

4.2 Modeling

The general idea of the *Modeling* step is to characterize all visualizations of the exploration set $\mathcal{ES}$ and to compute a uniform model for the *Model Learning* step (cf. Section 4.5). It computes a feature-based vector for every visualization. By this means, the similarity for each individual visualization can be automatically compared and used for further sampling or classification methods.

We decided to use the *Scagnostics* approach [34] to characterize the scatter plot contents, since it is capable of describing point distributions by meaningful measures. This approach extracts a nine dimensional feature vector characterizing the scatter plots for: outlying, skewed, clumpy, convex, skinny, striated, stringy, straight, and monotonic features. These measures can characterize the shape of scatter plots well. Thus, the decision classifier can identify the user's preferences in the form of "find more dense" or "find highly coherent scatter plots".

Depending on the chosen visualization (cf. Section 4.1), different descriptors have to be used to extract feature vectors. While, *Kernel Density Estimators* or *Regressional features* could be used to extract suitable features for scatter plot point distributions, image-based descriptors will be more appropriate to describe real world images. In the case of structure conveying visualizations, such as treemaps or matrices, one option is to apply an *Edge-Histogram Descriptor* or line extraction algorithm [11] to describe the general shape of the visualizations content. For text visualizations, a dictionary-based approach can be applied to compare the textual content inside visualizations.

4.3 Representatives

The general idea of the *Representatives* step is to select a manageable number of items from the exploration set $\mathcal{ES}$. This presentation set, denoted as $\mathcal{RS}$, is presented and judged by the user. Hence, its functionality highly influences the exploration process. The selection procedure is exchangeable in our implementation of the view space exploration pipeline.

In the current implementation, we are experimenting with content-based *sampling-based* approaches, such as discussed in [15], to select a range of items in $\mathcal{ES}$. A Min-/Max sampling option selects two representatives for each of the feature value ranges (cf. Section 4.2). For the Scagnostics example, 18 representatives can be judged by the user: two items (one min-value and one max-value representative) for each of the nine Scagnostics features. To increase the number of items in $\mathcal{RS}$ and to reflect the data distribution, a quantile sampling, a (stratified) bootstrapping and a stratified normal sampling method can be applied. The user can select how many items should be retrieved, resulting in $|features| \times requestedSamplingItems$ items in $\mathcal{RS}$.

One reason to apply sampling is that $\mathcal{RS}$ is available instantaneously without much computational effort. One obvious disadvantage is that the sampling potentially shows a series of outliers in the data distribution. However, this effect can be neglected in case of the quantile sampling method ($amountQuantiles > 2$).

As stated above further design alternatives are possible and may be considered if the representation items are not perceived as appropriate. One computational expensive solution would be to apply a density-based clustering in every projection pane of the feature space. N modeling features would lead then to N projection panes. From the clustering results a range of representatives could be selected by choosing, e.g., the medoid of the found clusters.

4.4 Relevance Feedback

The general idea of the *Relevance Feedback* mechanism in a semi-automatic exploration pipeline is to give the user the ability to steer the retrieval process. The user can categorize the presented items into relevant, irrelevant, and neutral examples. Relevant annotated items, denoted as $\mathcal{AS}^+$, represent potential *hits* for the search process. Irrelevant examples, denoted as $\mathcal{AS}^-$, depict items that lead to wrong or uninteresting search paths.

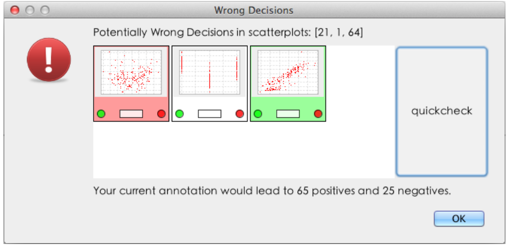
In the current implementation, depicted in Figure 1, the user can click on green and red buttons to express his like, respectively dislike.

Alternatively to the binary decision approach, a weighted relevance feedback could be implemented for finer granularity assessment of the user feedback. In this case, the users would have to judge the interestingness of the presented items in terms of a linear scale. Also a star rating, as it is known from product reviews, would be possible. We decided against a weighted relevance feedback system, since these kinds of decisions might be hard to judge for the user in the beginning of the exploration process and involve additional interaction overhead.

4.5 Model Learning

The goal of the *Model Learning* step is to reflect the user's preferences. In the best case, the system learns the user's intention after only one iteration and retrieves only positive examples. The worst case scenario is that the model learning cannot grasp the user's intention after a finite number of steps, leading to always negative examples. The pipeline would then iterate until all irrelevant items are excluded and only relevant examples are left. Hence, the search eventually converges.

A model learner has to be able to revise and refine previous decisions. Whenever the system restricts existing decisions we assume that the user also refined his/her understanding of the explored items. Thus, we assume that the exploration path is "correct". On the other hand, a revision of existing decisions corresponds to a potentially "wrong" exploration path.

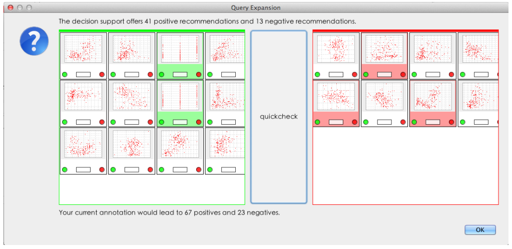


Wrong Decisions

Potentially Wrong Decisions in scatterplots: [21, 1, 64]

Your current annotation would lead to 65 positives and 25 negatives.

Fig. 5. Potentially wrong decisions are intercepted by the decision support system to keep the model learning in a consistent state. The outcome of each decision can be anticipated without applying it to the model learner by using the quick-check functionality button.



Query Expansion

The decision support offers 41 positive recommendations and 13 negative recommendations.

Your current annotation would lead to 67 positives and 23 negatives.

Fig. 6. Additional meaningful decisions can be recommended to the user by retrieving the most similar scatter plots to the already relevant, respectively irrelevant, annotated views.

On top of the exploration steering function of this component we put another prerequisite on the system: It should externalize its decisions in a visual depiction (cf. Section 6.2). In our current implementation we decided to use a classification system to approximate the user's preferences. Our model learner is inspired by the idea of an iterative decision tree, such as presented in [35]. In contrast to a normal decision tree, iterative decision trees retain most of its structure after the initial training. This allows the user to perceive the structural development over time and can only be achieved if a fundamental confinement of the decision tree algorithms gets derestrict: Nodes corresponding to a parameter (-range) can occur multiple times in the same decision tree. However, in line with the decision tree logic, these multiple occurrences may not violate already applied range restrictions (value > 0.5 leads to *yes*, but also value ≤ 0.5 should lead to *yes*).

In a standard course of action, we are expanding the decision leafs in each learning iteration of the pipeline. We are differentiating between *outer decisions* and *inner decisions*. While outer decisions modify the outer boundary of the decision space formed by the n selected features (cf. Section 4.2), inner decisions lead to subareas in the already excluded/included decision space, which should be included, respectively excluded, from the search.

For outer decisions two cases are possible without violating the idea of a decision tree: 1) A *yes* node, representing a set of relevant classified items, on a *yes path* gets split up. In this case, the user found that the classification is too unspecific and should be narrowed. One example for this case is shown in Figure 12 (c), where the monotonic feature range was modified from $[0.09, 1.0]$ to $[0.12, 1.0]$. 2) A *no* node,representing a set of irrelevant classified items, on a *no path* gets split up. In this case, the user found that the classification is too specific and should be broaden.

Inner decisions are improving constraints set in earlier decisions. Here, also two alternatives are possible without violating the decision tree idea: 1) A *no* node on a *yes path* gets split up. In this case the user found that learned constraint is limiting the exploration and should be made less restrictive. 2) A *yes* node on a *no path* gets split up. In this case the user found that learned constraint was not restrictive enough and should be strengthened. One example for this case is shown in Figure 12 (c), where a monotonic value above 0.006 alone would lead to a positive classification. This classification gets restricted by the sparsity feature descriptor below 0.11. In both presented cases of an inner decision parallel decision paths, or split-ups, could be a result.

Alternatively, adaptive learning systems could be applied to learn the user preference model. Here, for example multi-agent learners, such as presented in [27] could incorporate likelihood considerations into the learning process, which would be beneficial if many views show similar content. The application of neural networks, such as presented in [37], could be an alternative. Both mentioned model learners are able to learn non-linear decision boundaries in high-dimensional decision spaces. We decided against these sophisticated methods due to the following reasons: 1) They mostly cannot meet our prerequisite of being visually interpretable/traceable. 2) Their application would lead to immense computational efforts and results in long waiting times for the users. 3) Many of these approaches require a full training after each of the user decisions. Support-Vector machine classification has been considered, but is not yet implemented. Here, visual depictions are available, such as presented in [17]. The training process is more complex than with the presented approach, but still feasible and a full training is not always necessary as [6] demonstrates.

5 Enhanced Decision Support for Feedback-Driven View Exploration

One of the primary advantages of the presented exploration framework (cf. Section 3) is that it allows for an automatic supervision of the exploration process. This supervision can be used to investigate and monitor the actions taken by the user. Thus, it becomes possible to make use of a user feedback loop whenever an action is not meaningful, potentially incorrect, or could be improved on the fly.

Users are notified about a potential intervention with the help of dialogs. These dialogs contrast the current user selection with an automatically created suggestion. Most importantly, the decision support system forecasts both options' outcome and presents them to the user.

In the case of conflicting decisions (cf. Section 5.1) between the user and the decision support system, the user decision are preferred to the algorithmic decisions.

In the following we are referring to our implementation of the feedback-driven view exploration framework as it is described in Section 4. Specifically, we are rendering scatter plot visualizations modeled/described by the scagnostic feature set. We are applying a sampling-based approach to find representatives. The user gives binary feedback, whether an item is relevant or rather irrelevant; the incremental decision tree algorithm classifies the exploration set $\mathcal{ES}$ into positive $\mathcal{CS}^{+}$and negative classified items $\mathcal{CS}^{-}$.

5.1 Handling Potentially Wrong Feedback Decisions

Decisions are ambiguous and potentially wrong if the same view (scatter plot) has been marked both irrelevant and relevant in the current iteration. In both cases the user has to revise and disambiguate the current decision in an *Error Dialog*, depicted in Figure 5.

The error dialog allows previewing the decision outcome with the help of a *quick check functionality*. Its purpose is to anticipate the $\mathcal{CS}^{+}$ and $\mathcal{CS}^{-}$ outcome without applying the decisions to the classification model learner.

If this kind of error occurs multiple times the decision support system suggests enabling an auto-highlighting functionality that keeps track of the annotation sets and holds them in a consistent state.

5.2 Handling Missing Decisions

Missing decisions can occur whenever the same scatter plot is shown multiple times in one presentation set and the user marks a scatter plot as relevant, respectively irrelevant, but does not apply the same choice on the second occurrence of the scatter plot. Multiple occurrences can happen, e.g., when applying sampling-based representation finding approaches on a small data set. In case of Min-/Max sampling, multiple presentations of the same item are even likely and cannot be ruled out.

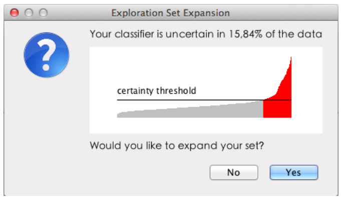
The decision support system keeps track of these missing values and fills them automatically to retain a consistent learning model.

5.3 Recommending Additional Decisions

The decision support system is able to do more than a mere failure handling. If the user is satisfied with her/his relevance feedback in one iteration, the intra-presentation set similarity to the positive and negative examples is calculated. If the similarity for an unannotated scatter plot is high to one of the items in $\mathcal{AS}^+$ or $\mathcal{AS}^-$ it becomes an annotation candidate for the respective annotation set. More specifically, we are calculating for every view in $\mathcal{AS}^+$ and $\mathcal{AS}^-$ a ranked list of similar views from $\mathcal{ES}$. We are using the Euclidean distance on the Scagnostics feature vectors for calculating the similarity score. These ranking lists are unified for each annotation class by taking into account (a) a minimum similarity threshold –since we want to show only highly similar views– and (b) the potential reoccurrence of one view in the candidate lists –since we want to eliminate duplicate candidate views. The outcome of including annotation candidates into $\mathcal{AS}^+$ and $\mathcal{AS}^-$ are presented to the user in the *query-expansion dialog* shown in Figure 6. Here again, the user has the functionality to anticipate (quick-check) the results without applying them to the model learner.

5.4 Exploration Set Expansion

Another decision support system functionality aims at assessing the search stability and convergence. For example, if the decision tree classifies more than 50% (user-parameter) of $\mathcal{ES}$ as irrelevant in the first iteration a great variety of potential scatter plot patterns may be lost. If the selected parameter is exceeded, the decision support system evaluates the *classification certainty* by comparing all irrelevant classified items ($\mathcal{CS}^-$) to the items in the annotation set $\mathcal{AS}^+$ and $\mathcal{AS}^-$. We construct a certainty ranking for all items in $\mathcal{CS}^-$. Again, we are using the Euclidean distance on the Scagnostics feature vectors for calculating the ranking score. The subset of items in $\mathcal{CS}^-$ that have a distance higher than an adaptive threshold are treated as uncertain classification decisions and may be taken again into the exploration set $\mathcal{ES}$ for a further refinement. The certainty threshold increases with the number of feedback iterations. In other words, the fewer decisions have been taken by the user the less uncertainty is accepted. Figure 7 shows the exploration set expansion dialog.


Exploration Set Expansion

Your classifier is uncertain in 15,84% of the data

Would you like to expand your set?

Fig. 7. The classification system's certainty is assessed in the background. A histogram view shows the number of (un-)certain decisions. If the classifier eliminates a great variety of scatter plot patterns (red bars), the user may decide to retain uncertain scatter plots. Uncertain decisions correspond to the scatter plots whose distance to the negative annotated set is larger than an adaptive certainty threshold.

6 View Space Explorer

In the following we will describe our graphical user interface for the Feedback-Driven View Exploration Framework. Figure 1 depicts the visual interface, consisting of the *View Explorer*, the primary interaction component, and a range of meta visualizations, which help to track changes in the exploration process.[1]

6.1 View Explorer

The View Explorer displays the presentation set $\mathcal{RS}$ to the user. It is depicted in Figure 1 (4). The scatter plot selection process is described in Section 4.3. The displayed views are ordered according to the feature descriptor and the selected sampling method; i.e. for each applied feature descriptor one high and one low value in the case of Min-/Max sampling. While an alternative option would have been to sort the views according to their feature vector similarity, the applied ordering allows perceiving the feature descriptor's value ranges more effectively. A detail view for the user annotation options is depicted in Figure 9. A tool tip shows, next to the scatter plot's id and its axis metadata, a visual indication of the sampling set choice; the circled number represents the selected 2^{nd} quantile.

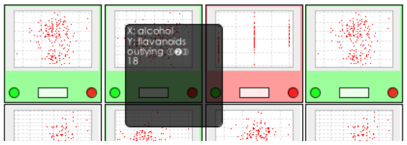

X: alcohol
Y: flavanoids
outlying ①❷③
18

Fig. 9. Users can annotate scatter plot views as uninteresting, neutral, or interesting with the red, white or green buttons. A tool tip shows that the view was selected into the presentation set $\mathcal{RS}$, since its outlying value is in the 2^{nd} quantile of the respective feature range.

6.2 Meta Visualizations

Two meta visualizations help the user to assess if an exploration path leads to a convergence (only interesting views). On top of that, the decision support system uses the displayed data to quantitatively assess the exploration convergence.

Feature Tube: Figure 10 shows the *Feature Tube*, a stacked histogram view per feature descriptor (cf. Section 4.2). The histograms are sorted in ascending order to reveal the feature's value distributions. The current decision path corresponds to an interval selection in the n-dimensional feature space, where n is the number of features under consideration. In our case, nine Scagnostics feature histograms are rendered. We are showing the decision path by a tube overlay, highlighting the selected feature intervals of interest. The overlay can be used to assess the specificity of the search. A narrow tube relates to a highly specific query –potentially in an advanced status of the exploration process– while a broad tube shows that the exploration is unspecific. Selected intervals can vary in their density of contained items (or views). Dense/sparse intervals show that the current exploration specificity maps to many, respectively few, possible views. By perceiving the change of Feature Tube between two model learning phases, users are able to judge their exploration advancement. Brushing and Linking is used to retrieve a scatter plot selection from the view explorer in all feature histograms.

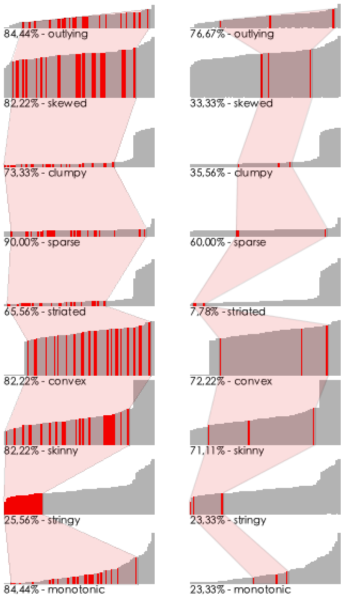
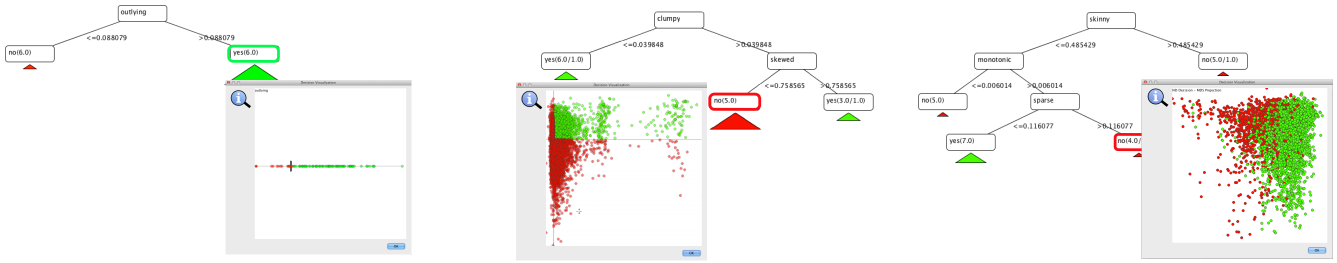


Fig. 10. The Feature Space Tube represents visually the exploration process advancements. A narrow tube overlays (right) corresponds to a highly specific query, while a broad tube (left) represents an unspecific exploration.

[1] A video showing the functional components and the main interaction work flow is available in the supplementary material.



(a) 1D decision on one classification attribute (b) 2D decision on two classification attributes (c) nD decision on n classification attributes

Fig. 12. The incremental decision tree allows assessing the complexity of the formulated exploration query. Additional meta visualizations depict the value distribution for 1D classification decisions (decision on one classification attribute), 2D decisions in a confusion matrix and nD decisions in a MDS projection of the classified items similarity.

Incremental Decision Tree: Figure 1 (5) and Figure 12 shows the *Incremental Decision Tree*, a visualization for the current classifier decisions. In an incremental decision tree every framework iteration corresponds to one level of the decision tree: Level 1 decisions correspond to the projection of the n-dimensional decision space onto the one dimensional subspace of the corresponding classification attribute. Level 2 decisions span a confusion matrix, in which the true-positive (upper right) field corresponds to all positively classified items ($\mathcal{CS}^{+}$), the true-negative field (lower left) corresponds to all negatively classified items ($\mathcal{CS}^{-}$). The items in false-positive (upper left) and false-negative (lower right) are potential mis-classifications and cannot be rejected without any reservation. Thus, they remain in the exploration set $\mathcal{ES}$. Level decisions > 2 can be depicted with a MDS projection of the pairwise item similarity. We use of the classical MDS implementation in MDSJ [25] for our purposes. During the annotation phase the user sees the tree visualization, as depicted in Figure 1 (5). However, if the user wants to get more details about the model learner meta visualizations for the 1D, 2D and nD levels are shown in a dialog on top of the incremental decision tree (Figure 12). The purpose of these meta visualizations is to allow the user to interpret how many items the classifier rejected from the exploration. Furthermore, the user can perceive how many items where close to the calculated decision border.

6.3 Measures of Exploration Convergence

For a quantitative assessment of change in the exploration process we are quantitatively measuring the *appearance development* of the feature tube and the incremental decision tree visualization. For the feature tube we are storing the covered tube area for each pipeline iteration and calculate the areal difference of the two shapes. If the difference area decreases in two subsequent iterations, the search converges gradually and we are able to measure a *convergence factor*. If the area change stagnates or even increases the user did not advance in the view exploration. For the incremental decision tree, we are able to measure a binary convergence factor. Either, in the negative case the model learner returns the same decision tree several times (no further learning improvement) or new tree leafs need to be added (learning progress). If the classification training results in the same tree twice we are interpreting the result as an exploration stagnation.

A wide range of other convergence measurements are possible. The simplest is to relate the number of items in $\mathcal{ES}$ before and after an application of the classifier. Another option is to calculate a similarity value for the decision tree appearance before and after the application of the classifier, as e.g., presented for general tree structures in [19]. However, this option can only be applied if the decision tree is built from scratch, rather than not incrementally. In the future we are planning to experiment with an adaptive convergence measure that takes multiple decision criteria into account. In all cases of a slow or stagnating convergence we are applying *counter measurements* to steer the exploration process. The first measurement is to intervene in the representation finding (cf. Section 4.3). In our case, we are changing the used sampling function. The next level of intervention is to increase the number of suggestions for annotation candidates (cf. Section 5.3). While the number of suggestions decreases in the normal convergence cases, we are here boosting the lowest similarity, such that more (even less similar) items are recommended for annotation.

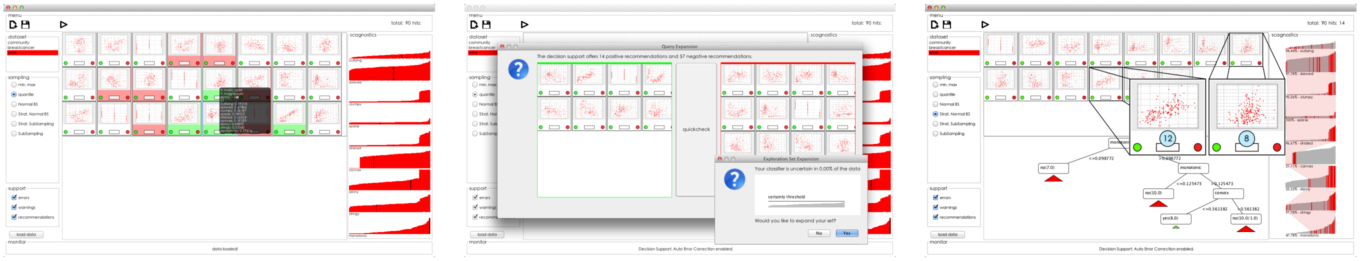
7 Case Studies

To present the applicability of our View Space Explorer we showcase two exemplary exploration scenarios on two well-known, real-life numeric datasets: The *Wine* and the *Communities and Crime* data set, both from the UCI Machine Learning Repository [1]. The two chosen datasets vary in size and complexity. We chose the wine dataset, because of its low number of scatter plot axis combinations and thus its understandability. This allows us to focus on the decision support's interventions in the exploration process. The second dataset (communities) is significantly more complex in the number of dimensions and allows us to focus –next to the presentation of (hidden) patterns– on the convergence development like it is necessary for information retrieval systems. We conducted our case studies with Ph.D. and Master students from the computer science area.

In the case studies we are searching for correlations in the axis combinations. The exploration of data correlations is challenging, since users must be able to describe the correlation to be found. In the feedback-driven view exploration approach we assume that users do not know in advance, which (Scagnostic) feature might be beneficial for the current question.

Wine Data Set The Wine data set results from a chemical analysis of 178 wines sub-categorized into three classes (red, white, rose wines). In total the dataset comprises 14 numeric continuous attributes, such as alcoholic strength, color intensity, or magnesium. We derived 90 axis combinations and converted them to scatter plots -short SPs- with 178 data points each.

Figure 11 shows us a sequence of interactions on the wine dataset. In the first iteration (Figure 11 (a)) the user annotated four relevant and six irrelevant SPs from 24 initially presented SPs. The presentation set $\mathcal{RS}$ results from a normal bootstrapping sampling of the initial 90 SPs. A review of the four relevant annotated SPs shows that they share a high *monotonic* value. Satisfied with the first annotation round the user clicks on *Apply* and sees a dialog, in which the decision support recommends adding 15 SPs to $\mathcal{AS}^{+}$ and 68 SPs to $\mathcal{AS}^{-}$ (Figure 11 (b) background). The user declines both recommendations. Hence, the classifier is trained on the initial annotation set and results in 27 positively and 63 negatively classified SPs. A review of the classification uncertainty shows that all decisions were certain (Figure 11 (b) front). Thus, the user can assume that no SP is lost due to a misclassification. The feature tube reveals that the highest concentration is in the monotonic value. It appears to be beneficial to choose SPs with a high monotonic value for this task. In the second feedback round the user selects seven relevant and six irrelevant SPs and the decision support recommends adding one relevant and four irrelevant SPs. Once again the user declines all recommendations. The subsequent classification results –again– in the same 27 positively and 63 negatively classified SPs. No more exploration progress is apparent and thus the user will see the same SPs in each subsequent feedback round. Accordingly,



The decision support offers 14 positive recommendations and 57 negative recommendations.

Your classifier is uncertain in 0.00% of the data

Would you like to expand your set?

Decision Support: Auto Error Correction enabled.

(a) In the first annotation round the user selects four relevant and six irrelevevant scatter plots.

(b) The decision support recommends adding a range of similar scatter plots to both annotation sets; After the classifier is trained all decisions are judged as certain.

(c) The final result shows several data correlations. E.g. The amount of ash and flavoid influences a wine's color.

Fig. 11. Finding correlations in the Wine data set; After three annotation iterations the exploration set with initially 90 scatter plots is reduced to 14 scatter plots, containing the annotated data correlations.

the decision support switches the sampling method to "Stratified bootstrapping". In a final annotation round six relevant and five irrelevant SPs are chosen and the exploration finishes with 14 SPs.

In the 14 result SPs some patterns become visible (Figure 11 (c)). For example, the SPs 8 and 12 show that the wines' color is positively correlated with the amount of ash and the amount of flavonoids. A meta research reveals: "Flavonoids are antioxidant compounds found in plants, as well as tea, red wine and chocolate, . . . " [20].

Communities and Crime Data Set The Communities and Crime dataset combines socio-economic, law enforcement, and crime data for the US in the years 1990 to 1995. After a filtering of missing values the dataset contains 123 dimensions. All in all, 7002 scatter plots were generated from the possible axis combinations.

Figure 12 shows a sequence of actions on this data set. The final result of the exploration is given in Figure 1. Our goal is again to find hidden data correlations that can be semantically interpreted.

In order to decrease the exploration set not too much in an early exploration phase, the user selects SPs with a high skewness. This feature describes indirectly the data density and gives an intuition about the interrelation of two dimensions. As Figure 12 (a) depicts, the user selects five relevant and six irrelevant SPs. Subsequently, the decision support recommends 2247 positive and 2045 negative SPs. This large amount of recommendations indicates that the decision support is uncertain about the exploration path direction. The user declines all recommendations and receives a classification uncertainty of 12.28% for the irrelevant classified SPs. Moreover, the uncertainty visualization in Figure 12 (a - front) depicts that the majority of distances is significantly above the uncertainty threshold. Thus, the user decides to expand the exploration set with the uncertain SPs. All in all, the exploration set is reduced to 5251 SPs, leading to an exploration set decrease of 25.0%.

In the following second annotation round the user tries to narrow down the exploration and annotates thus six irrelevant and five relevant SPs with a stronger visible correlation. The annotation is depicted in Figure 12 (b). Due to the reason that the recommendation threshold is adapted iteratively, a significantly lower amount of SPs (91 relevant; 344 irrelevant) gets recommended. However, the user once again declines all recommendations. In the subsequent certainty assessment the user sees that in 28.81% of the irrelevant classified decisions the classifier is uncertain. Hence, the user decides again to retain all uncertain scatter plots. All in all, the exploration set is reduced to 2536 SPs, leading to an exploration set decrease of 51.7%.

In the third annotation round, depicted in Figure 12 (c), the user searches for SPs with a rather round scatter plot distribution, which relates to the Scagnostics convexity feature. Thus, the user annotates seven irrelevant and six relevant SPs and the decision support recommends adding one relevant and seven irrelevant SPs. One positive recommendation is accepted. The classification uncertainty is 34.12%. Accordingly, the exploration set is expanded with all uncertain SPs. All in all, the exploration set is reduced to 1900 SPs, leading to an exploration set decrease of 25.1%.

In the final annotation round the user chooses to filter SPs with a low density distribution, which relates to a strong skinny value. These dimension combinations often occur if a categorical value is related to a numeric value. Thus, six relevant and six irrelevant SPs are annotated. The decision support recommends adding 1 relevant SP. The classification uncertainty is 9.74%. In this annotation round the exploration set is not expanded anymore. All in all, the exploration set is reduced to 235 SPs, leading to an exploration set decrease of 87.6%. The search converges and we see the final result in Figure 1.

According to the result set depicted in Figure 1 we can come to the following conclusions: First, we found that a high crime rate exists in areas, where the percentage of households with public assistance income is high. A similar correlation exists with the percentage of unemployment in these areas (SP A). Second, we found that in these areas the police budget is higher than in areas with a low crime rate (SP D). Third, we found that the police budget correlates with the number of police cars (SP B). Generally, SP C shows that more drug units exist the larger is the variety of drug types in that area.

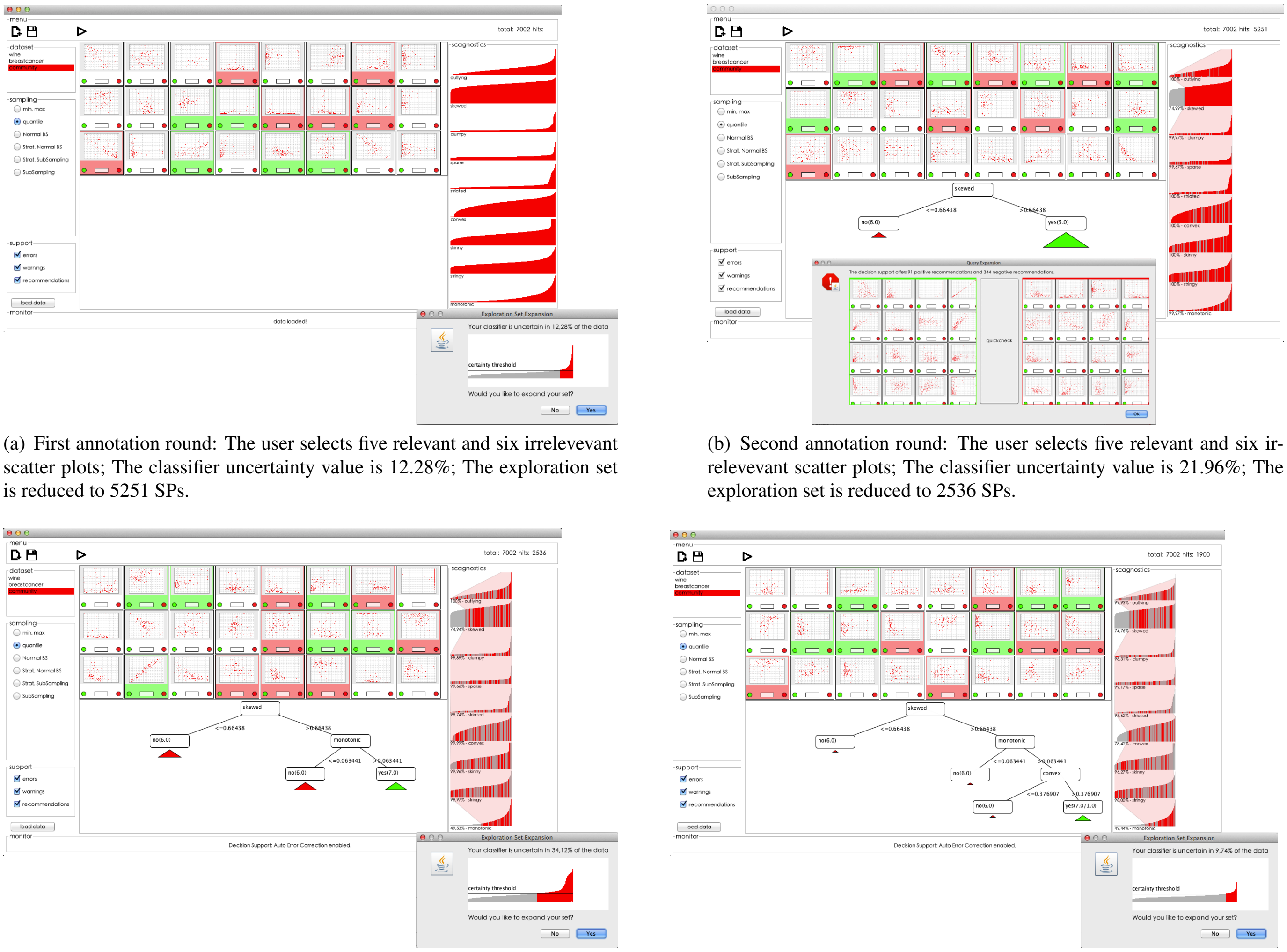
8 Discussion and Extension

While our technique has proven useful, we have identified several areas where improvements or alternatives should be explored.

Firstly, we want to apply the view space exploration framework on data sets which allow for more than one meaningful visualization. As an example, we could represent the Wine or Communities and Crime data sets from Section 7 also with Parallel-Coordinate-Plots. This would allow representing different data aspects more prominently. In these cases, the view selection space will increase drastically, leading to new visualization and computation challenges. In line with this question we want to investigate when proven visualization techniques for building overviews without abstractions, esp. Scatterplot-, Parallel-Coordinate-, or Generalized Plot Matrices [7, 18, 21], can be outperformed with our iterative exploration approach. On the other hand, for arbitrary visualizations, which can not be represented in a small multiple manner, layout based approaches will not scale due to the screen space restrictions.

Secondly, we are already experimenting with a tight integration of the decision support system with the user feedback loop and the model learner. While the intervention process appears to improve the exploration, many alternatives to assess the search stability and convergence are possible. One particular research challenge is the degree of intervention in the process. Certain user groups might wish for guidance, others might feel uncomfortable with this supervision. Related to this question is that it might be beneficial to enhance Decision Support with a query negation function, allowing query suggestions, like 'Show non-correlated variables, instead of correlated'.

Thirdly, we are planning to experiment with implicit and finergranular relevance feedback mechanisms. Specifically, a range of design options becomes available, ranging from time-to-click measures



data loaded!

Your classifier is uncertain in 12.28% of the data

Would you like to expand your set?

The decision support offers 91 positive recommendations and 344 negative recommendations.

Decision Support: Auto Error Correction enabled.

Your classifier is uncertain in 34.12% of the data

Your classifier is uncertain in 9.74% of the data

(a) First annotation round: The user selects five relevant and six irrelevant scatter plots; The classifier uncertainty value is 12.28%; The exploration set is reduced to 5251 SPs.

(b) Second annotation round: The user selects five relevant and six irrelevant scatter plots; The classifier uncertainty value is 21.96%; The exploration set is reduced to 2536 SPs.

(c) Third annotation round: The user selects six relevant and seven irrelevant scatter plots; The classifier uncertainty value is 34.12%; The exploration set is reduced to 1900 SPs.

(d) Fourth annotation round: The user selects six relevant and six irrelevant scatter plots; The classifier uncertainty value is 9.74%; The exploration set is reduced to 235 SPs.

Fig. 12. Finding correlations in the Communities and Crime data set; After four annotation rounds the exploration set with 7002 scatter plots is reduced to 235 scatter plots, containing data correlations.

to eye-tracking approaches, to assess relevance of a view. It will also be interesting to explore, similar to the document term highlighting mechanism of [12], ways to highlight which local patterns would be responsible for a given scatter plot view being relevant for a user.

9 Conclusion

The interesting view problem is prevalent in visual data exploration approaches whenever the number of available data views to be investigated exceeds the user's willingness or ability to judge the views. While previous approaches centered around establishing effective overview abstractions to guide the user to interesting views, the focus is shifting to an automated calculation of interestingness. Whenever a proper definition of interestingness can be established it needs to be transformed into a quality measure. However, in most cases the formulation of interestingness is neither possible nor stable. Rather so, the understanding of interestingness develops in the sequence of actions taken on the data.

In this paper we motivate an iterative and interactive view space exploration approach that does not rely on any visual abstractions. We introduced a general feedback-driven view exploration framework in which a relevance feedback mechanism is applied to retrieve user preferences. On the automatic side, the system learns the user preferences and finds new interesting views by applying a classification on the data. The more the search advances, the more user preferences are learned and transformed to specific queries on the underlying data. To showcase our general ideas, we presented one instantiation of the feedback-driven view exploration framework. In this instantiation we render scatter plot visualizations modeled by the Scagnostic feature set. It has to be noted that the general idea is not limited to scatter plots, but allows rather any type of visualization technique as long as a descriptive feature vector space can be found. A sampling-based approach is used to find potential interesting views from an exploration set. In every interaction loop the exploration set is incrementally reduced by implicit queries resulting from the user's binary relevance feedback. An incremental decision tree algorithm classifies the exploration set into potentially relevant and irrelevant items. A novel decision support system is applied on top of the framework that supervises the decision process. It is used, on the one hand, to handle potentially inconsistencies in the annotation process and, on the other hand, to evaluate automatically the exploration convergence.

We evaluated our approach with a case-study driven evaluation, conducted to showcase its usefulness on two real-world data sets. The results reveal that our feedback-driven view space exploration framework shows to be effective and enhances the user understanding.

In conclusion, the presented feedback driven view space exploration framework serves as a basis for a range of visual analytics systems that allow tackling the interesting view problem.

References

[1] K. Bache and M. Lichman. University of California (UCI) machine learning repository. Online, June 2013. http://archive.ics.uci.edu/ml/.

[2] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval - the concepts and technology behind search, Second edition*. Pearson Education Ltd., Harlow, England, 2011.

[3] P. Baudisch, N. Good, and P. Stewart. Focus plus context screens: Combining display technology with visualization techniques. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, UIST '01, pages 31–40, New York, NY, USA, 2001. ACM.

[4] E. Brown, J. Liu, C. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 83–92, Oct 2012.

[5] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[6] G. Cauwenberghs and T. Poggio. Incremental and decremental support vector machine learning. *Advances in neural information processing systems*, pages 409–415, 2001.

[7] W. S. Cleveland. *Visualizing Data*. Hobart Press, 1993.

[8] T. N. Dang and L. Wilkinson. ScagExplorer: Exploring Scatterplots by Their Scagnostics. In *Pacific Visualization Symposium (PacificVis), 2014 IEEE*, pages 73–80, March 2014.

[9] A. Dasgupta and R. Kosara. Pargnostics: Screen-space metrics for parallel coordinates. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1017–1026, Nov 2010.

[10] T. Deselaers, D. Keysers, and H. Ney. Features for image retrieval: an experimental comparison. *Information Retrieval*, 11(2):77–107, Apr. 2008.

[11] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, Jan. 1972.

[12] A. Endert, P. Fiaux, and C. North. Semantic interaction for visual text analytics. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, page 473–482, New York, NY, USA, 2012. ACM, ACM.

[13] J. Friedman and J. Tukey. A projection pursuit algorithm for exploratory data analysis. *Computers, IEEE Transactions on*, C-23(9):881–890, Sept 1974.

[14] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Elsevier Ltd, Oxford, 3rd edition, 2011.

[15] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.

[16] C. G. Healey and B. M. Dennis. Interest driven navigation in visualization. *IEEE Trans. Vis. Comput. Graph.*, 18(10):1744–1756, 2012.

[17] F. Heimerl, S. Koch, H. Bosch, and T. Ertl. Visual classifier training for text document retrieval. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 18(12):2839–2848, 2012.

[18] J. Heinrich, J. Stasko, and D. Weiskopf. The parallel coordinates matrix. *EuroVis–Short Papers*, pages 37–41, 2012.

[19] M. Hess, S. Bremm, S. Weissgraeber, K. Hamacher, M. Goesele, J. Wiemeyer, and T. von Landesberger. Visual exploration of parameter influence on phylogenetic trees. *IEEE Computer Graphics and Applications*, 99(PrePrints):1, 2014.

[20] J. Hope. Chocolate and red wine can help stave off diabetes: High levels of antioxidants can regulate blood glucose levels. Online, January 2014. accessed 29 March 2014.

[21] J.-F. Im, M. McGuffin, and R. Leung. Gplom: The generalized plot matrix for visualizing multidimensional multivariate data. *Visualization and Computer Graphics, IEEE Trans. on*, 19(12):2606–2614, Dec 2013.

[22] D. A. Keim, D. Morent, J. Schneidewind, M. C. Hao, and U. Dayal. Intelligent Visual Analytics Queries. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST 2007)*, 2007.

[23] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[24] McKinsey Global Institute. Big data: The next frontier for innovation, competition, and productivity. Online, May 2011.

[25] C. Pich. Mdsj: Java library for multidimensional scaling (version 0.2). online, 2009. Available at http://www.inf.uni-konstanz.de/algo/software/mdsj/.

[26] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Trans. Circuits Syst. Video Techn.*, 8(5):644–655, 1998.

[27] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.

[28] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages*, VL '96, pages 336–, Washington, DC, USA, 1996. IEEE Computer Society.

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

[30] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(7):1088–1099, 2006.

[31] A. Tatu, G. Albuquerque, M. Eisemann, P. Bak, H. Theisel, M. A. Magnor, and D. A. Keim. Automated analytical methods to support visual exploration of high-dimensional data. *IEEE Trans. Vis. Comput. Graph.*, 17(5):584–597, 2011.

[32] A. Tatu, F. Maass, I. Faerber, E. Bertini, T. Schreck, T. Seidl, and D. A. Keim. Subspace Search and Visualization to Make Sense of Alternative Clusterings in High-Dimensional Data. In *Procedings of IEEE VAST*, pages 63–72. IEEE CS Press, 2012.

[33] M. O. Ward and Z. Guo. Visual exploration of time-series data with shape space projections. *Comput. Graph. Forum*, 30(3):701–710, 2011.

[34] L. Wilkinson, A. Anand, and R. Grossman. Graph-theoretic scagnostics. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*. IEEE Computer Society, 2005.

[35] H. Yang and S. Fong. Incrementally optimized decision tree for noisy big data. In *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, BigMine '12, pages 36–44, New York, NY, USA, 2012. ACM.

[36] J. Yang, D. Hubball, M. O. Ward, E. A. Rundensteiner, and W. Ribarsky. Value and relation display: Interactive visual exploration of large data sets with hundreds of dimensions. *IEEE Trans. Vis. Comput. Graph.*, 13(3):494–507, 2007.

[37] B. Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.