DOI: 10.1111/cgf.13404

Eurographics Conference on Visualization (EuroVis) 2018
J. Heer, H. Leitte, and T. Ropinski
(Guest Editors)

Volume 37 (2018), Number 3

Interactive Visual Exploration of Local Patterns in Large Scatterplot Spaces

Mohammad Chegini[1], Lin Shao[1], Robert Gregor[1], Dirk J. Lehmann[2], Keith Andrews[1], and Tobias Schreck[1]

[1]Graz University of Technology, Graz, Austria
[2] University Magdeburg, Magdeburg, Germany

Abstract

Analysts often use visualisation techniques like a scatterplot matrix (SPLOM) to explore multivariate datasets. The scatterplots of a SPLOM can help to identify and compare two-dimensional global patterns. However, local *patterns which might only exist within subsets of records are typically much harder to identify and may go unnoticed among larger sets of plots in a SPLOM. This paper explores the notion of local patterns and presents a novel approach to visually select, search for, and compare local patterns in a multivariate dataset. Model-based and shape-based pattern descriptors are used to automatically compare local regions in scatterplots to assist in the discovery of similar local patterns. Mechanisms are provided to assess the level of similarity between local patterns and to rank similar patterns effectively. Moreover, a relevance feedback module is used to suggest potentially relevant local patterns to the user. The approach has been implemented in an interactive tool and demonstrated with two real-world datasets and use cases. It supports the discovery of potentially useful information such as clusters, functional dependencies between variables, and statistical relationships in subsets of data records and dimensions.*

Categories and Subject Descriptors (according to ACM CCS): [Human-centered computing]: Visualization—Visualization systems and tools

1. Introduction

Visual data analysis has become a key area in computer science and an established approach to empower domain experts. For the particular case of *multivariate data*, i.e. tabular data having a large number (n) of dimensions and (m) of data records, analysis with the aid of *scatterplots* (bivariate projections of pairs of dimensions) has become popular.

For a multivariate data space of n dimensions, n^2 pairwise scatterplots are required to completely visualise the space ($n^2/2$ if transposed plots are eliminated). A matrix of scatterplots representing every pairwise combination of plots is called a *scatterplot matrix (SPLOM)*. The properties of a SPLOM for the purpose of data analysis have already been widely explored in the literature.

While these techniques focus on exploring patterns within the data at a global level, little attention has yet been paid to the analysis of local structures and patterns. For the purposes of this paper, a *pattern* is defined as a set of points in a scatterplot (i.e. from two of the n dimensions) contained within a specified bounding box. Points outside the bounding box are not part of the corresponding pattern. A *global pattern* comprises all the points in a particular scatterplot; in other words, the bounding box covers the entire scatterplot. A *local pattern* comprises a subset of points in a scatterplot. A *query pattern* is a pattern defined by the user, typically by interactively dragging a box.

One or more *descriptors* can be defined to characterise a pattern. A descriptor is a function taking a pattern (set of points) as input and producing a feature vector as output. *Shape-based descriptors* are based on the visual properties of the pattern, for example by subdividing the pattern into grid cells and calculating features such as the relative density of points in each cell. *Model-based descriptors* are derived mathematically, for example from a regression model, where individual features might be determined by evaluating the regression function at specific points.

The similarity between patterns can be defined in terms of similarity in the feature space of each corresponding descriptor, based on a distance metric such as the L1 metric or quadratic form distance [BUS10]. In practice, best results were often achieved using a similarity function defined as a weighted combination of distance metrics, subject to minimum thresholds for two measures of pattern matching quality. This is discussed in detail in Section 3.

With regard to local pattern analysis in scatterplots, Shao et al. [SMSL17] previously proposed a scheme to explore and display regression models for interactively selected local regions of a scatterplot. However, that work only considered a single scatterplot (2

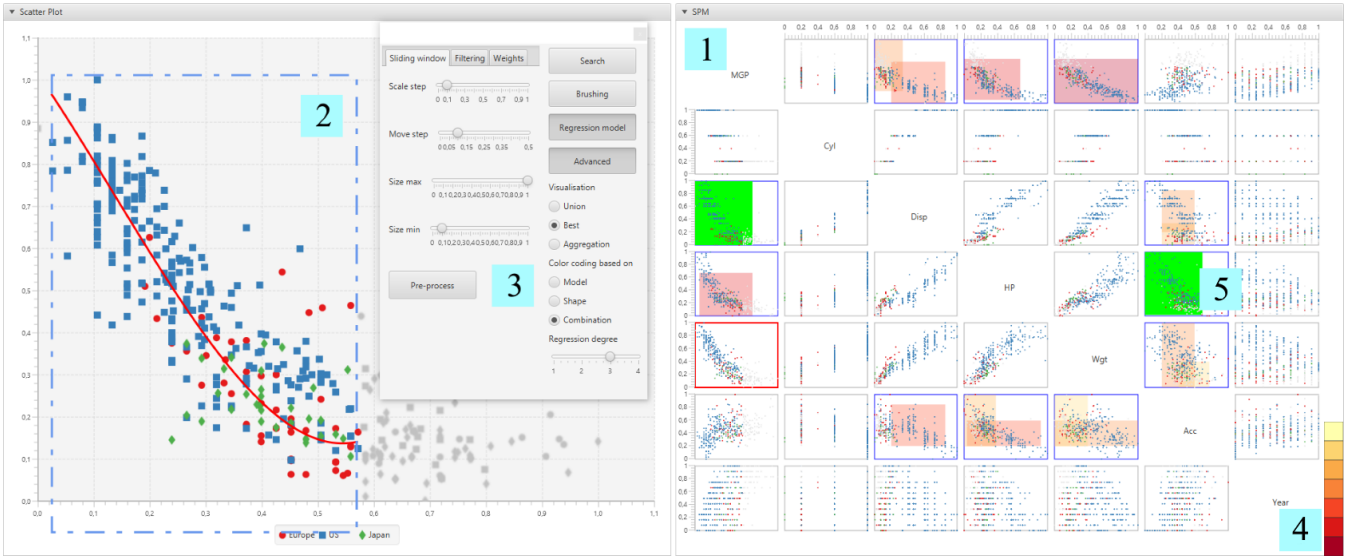


Figure 1: *The SPLOM view [1] can be examined for interesting patterns (rectangular regions). The colours of records (points) simply indicate the class they belong to. A query pattern can be specified interactively by dragging a bounding rectangle [2] in a scatterplot of interest. The query pattern can further be adjusted using the floating toolbox [3]. Matching patterns in the SPLOM are highlighted by coloured overlays according to their similarity to the query pattern. Dark red ones are the most similar and yellow ones are least similar [4]. The patterns with the green overlay are patterns marked by the user as relevant [5].*

of the n dimensions) independently of any others. This paper extends that approach to search for similar local patterns within *all other* scatterplots in the SPLOM. Matching patterns are ranked by similarity and presented to the user for further exploration. Visual highlighting is used to indicate matching local patterns within other scatterplots. The user can refine the search by selecting the patterns most relevant to their interests. A relevance feedback mechanism is then used to identify and recommend additional potentially relevant results. Figure 1 shows an overview of the approach.

For example, consider a dataset where the dimensions are attributes of a country (such as population, GDP, etc.) and the records are data from various countries and years (such as Japan 2010). Choosing an interesting local pattern and being able to explore similar local patterns within other scatterplots opens up a powerful new way to discover relationships between subsets of records across the entire multidimensional space. Appropriate interaction mechanisms allow the user to inspect the ranked set of matching local patterns and refine their query to explore further. While this paper focuses on the search for local patterns in other dimensions, the approach can be easily extended to finding patterns in other scatterplots of the same dimensions. For example, consider a dataset containing information about customers in various quarters of the year. Each quarter can be shown as a scatterplot, allowing the analyst to find similar patterns in other quarters. Another use case is searching for similar patterns in one scatterplot and on different clusters. For example, in the countries dataset, one could be interested in countries following the same pattern in the GDP-Population scatterplot.

The main goal was to develop an approach to find local patterns in a scatterplot space, without prior knowledge about the dataset. The design of a search algorithm for defined local patterns differs fundamentally from the design of a system dealing with general use cases. If the patterns of a dataset are known, it is possible to add carefully tailor-made descriptors. If no prior knowledge about the dataset exists, a general algorithm and descriptors to search for patterns are needed. For this reason, an approach which takes into account shape and model-based descriptors, various parameters, and a relevance feedback module was introduced. The approach can be further customised for specific datasets. The contributions of this paper are:

1. A set of interactive strategies to select local patterns of interest in one or more scatterplots.
2. An approach facilitating local pattern exploration by suggesting similar local patterns in other scatterplots across the entire SPLOM, based on relevance feedback.
3. Shape-based and model-based descriptors to characterise local patterns.
4. A similarity metric to determine the best matching local patterns in the rest of the multidimensional space.

The rest of this paper is structured as follows: Section 2 discusses related work. Section 3 explains in detail the concepts behind the proposed novel approach. The prototype implementation is presented in Section 4. Section 5 discusses use cases illustrating the utility of the system for analysing real-world datasets. Section 6 discusses open problems and future work, and Section 7 concludes the paper.

2. Related Work

The approach described in this paper is related to several research areas of visual analytics. The three most important directions which influenced the work are local scatterplot segmentation, visual retrieval techniques, and visualisation of scatterplot patterns.

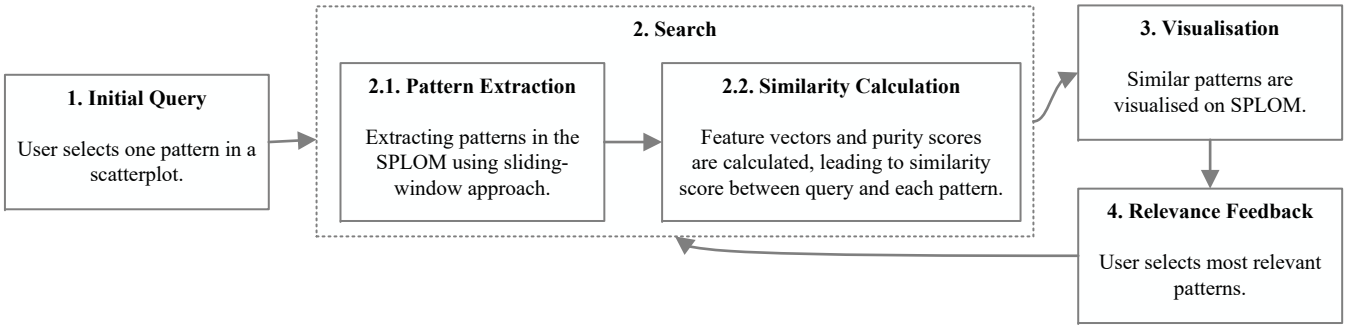


Figure 2: *Finding similar local patterns in a SPLOM. First, the user selects a rectangular portion of a scatterplot to create the initial query 1. Next, all identifiable patterns are extracted from the SPLOM using a sliding-window approach. Their feature vectors and purity scores are calculated, leading to a similarity score between each pattern and the query 2. The most similar patterns are visualised in the SPLOM 3. The user can then indicate which of the matching patterns are most relevant to their needs, thus refining the query 4.*

2.1. Scatterplot Segmentation

In recent years, the segmentation of local patterns has become an accepted and even essential part of data analysis in many fields, including genome research [ESBB98], trajectory analysis [MJE02], and image processing [FR97].

Data scientists often use data mining techniques in combination with information visualisation to present extracted patterns visually for human perception. Mayorga and Gleicher [MG13] describe an automatic abstraction approach which groups dense data points to reveal the relationship between data subgroups. Shao et al. [SSB+16] extracted local scatterplot motifs to create a visual overview of frequent patterns, which was then used to rank scatterplot views based on an adapted TF-IDF algorithm from information retrieval. The idea was to automatically determine weights of interest for a pattern, by comparing their occurrence frequencies within and among scatterplots. Another approach uses sensitivity coefficients from flow field analysis to highlight the local variation of one variable in relation to another [CCM10]. Chen et al. [CCM+14] used a hierarchical multi-class sampling technique to create new visual abstraction schemes for scatterplot visualisations. Sedlmair et al. [STMT12] proposed a taxonomy of visual cluster separation factors in scatterplots and a data-driven framework for evaluating visual quality measures [STMT15].

The technique described in this paper uses a sliding-window approach for segmentation, which compares a query against many possible matching candidate positions and areas, hence implicitly and heuristically segmenting the data.

2.2. Search Techniques for Scatterplot Retrieval

Another essential part of the work in this paper is the description and retrieval of scatterplot patterns. This research topic has been extensively investigated and addressed in recent works. Some of the pioneering work includes the Scagnostics approach by Wilkinson et al. [WAG05], which characterises 2D point distributions in a multi-dimensional Euclidean space using graph-theoretic measures. The approach can be used to search for patterns based on density, skewness, shape, outliers, and texture [MTL18]. Similar approaches were developed with the aim of finding similar patterns in other application domains such as time series or image retrieval [NAW13; NW14].

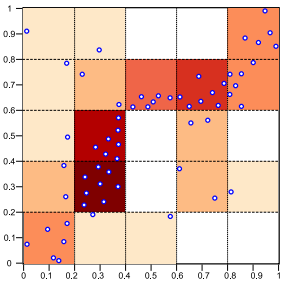
Scherer et al. [SBS11] introduced a goodness-of-fit approach based on regression models to find functional dependencies between pairs of variables in a dataset. To search for patterns of interest, the user can either enter the query directly as a formula or sketch a scatterplot. Scherer et al. [SvLS13] extended the approach to compare sets of scatterplots based on a bag-of-words model derived from scatterplot descriptors. Scherer et al. [SvLS12] compared scatterplot descriptors for effectiveness in finding globally similar scatterplots based on a defined ground truth dataset. Shao et al. [SBS+14] considered image-based features for sketch-based search in scatterplot data, including real-time feature extraction of the sketch.

Interesting work also exists to automatically detect a pattern in higher-dimensional data spaces. For instance, Tatu et al. [TMF+12] introduced a subspace search algorithm which suggests a set of subspaces of interest.

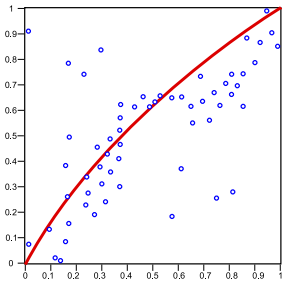
The work described in this paper differs in that the query is selection-based instead of sketch-based and the search algorithm combines both model-based and shape-based descriptors to specifically address local patterns. Moreover, the user can select the most relevant matches to further refine the search query.

2.3. Visualisation of Local Patterns

Various visual approaches have been proposed to present local properties in a scatterplot. Yates et al. [YWS+14] described an enhanced SPLOM representation called Glyph SPLOM, which links heatmap properties to a SPLOM. Instead of showing all single scatterplots, it uses glyphs to visually encode similarity features based on the occupancy of the scatterplot quadrants. The Regression Lens [SMSL17] is an example of how local properties of a scatterplot can be displayed interactively. Users can apply an interactive regression analysis on a local portion of the data and immediately see the best fitting model on the plot. Eisemann et al. [EAM14] describe interactive visualisation of distinct patterns of data within a given scatterplot (a hierarchy of localised scatterplots), which allows the user to explore dense areas in a scatterplot. In this paper, a

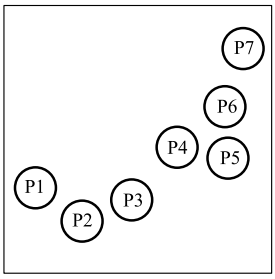


(a) *A shape-based descriptor defined by point density in a 5×5 grid, where the cell colour indicates relative density.*

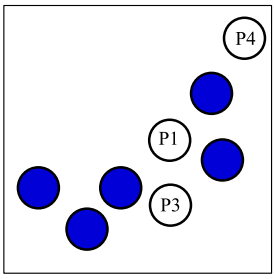


(b) *A model-based descriptor defined by a quadratic regression model.*

Figure 3: *Shape-based and model-based descriptors.*



(a) *User-defined query pattern containing seven points.*



(b) *A matching pattern containing eight points, of which three are contained in the query.*

Figure 4: *A user-defined query pattern and one matching pattern.*

combination of these approaches is used to enhance the visualisation of local patterns and facilitate the exploration of the dataset.

There are two essential aspects when visually analysing local patterns, namely, visualising and aggregating local patterns, and supporting visual comparison between them. Schreck and Panse [SP07] used class labels to group data points in a scatterplot, and show properties of the contained points using aggregation by bounding boxes, circles or convex hulls. Also, colour or blur was used to convey properties of the groups. Tominski et al. [TFJ12] suggested three visual comparison methods based on the natural behaviour of users when comparing charts: side-by-side, shine through, and folding interaction. Gleicher et al. [GAW+11] argued that since comparing complex objects is difficult, a promising strategy is the abstraction of complexity. They presented three types of comparative visualisation: juxtaposition, superposition, and explicit encoding. The work described in this paper uses all three approaches.

3. Exploring Local Patterns in Scatterplots

After discovering an interesting local pattern in one scatterplot, an analyst sometimes wants to search for similar local patterns in the rest of the SPLOM to investigate otherwise hidden relationships such as correlations between dimensions in a subset of the dataset. This section presents a novel visual analytics approach to select, search for, visualise and refine the search for local patterns. Figure 2 illustrates the search pipeline.

To begin a search for local patterns, the analyst draws a bounding box around a set of points in a scatterplot of interest, which specifies the initial query (Step 1 in Figure 2). The system then extracts a set of patterns from the entire space of scatterplots in the SPLOM. These patterns are built by successively translating and scaling a bounding box over each scatterplot. The bounding box moves over a scatterplot by a discrete *translation step size* and scales from the smallest size until it fits the whole scatterplot by a discrete *scaling step size*. All the patterns generated are added to the resulting *set of patterns*, which holds all extracted patterns (Step 2.1).

Feature vectors are extracted to describe each pattern using both model-based and shape-based descriptors. Shape-based descriptors represent the perceptual similarity of patterns. Since they characterise the data according to appearance, they have some limitations [PKF+16]. Model-based descriptors (currently from regression models) are used to capture the relationship between points in a pattern (Step 2.2). Since it can also be important that both the query and a pattern from the set of patterns contain a larger set of identical records, an overlap of records in the query and the target pattern is computed, and is used to filter the results. To this end, purity scores (Section 3.2) are introduced. Purity scores indicate how many records from the query, exist in a pattern.

A ranking of patterns is determined by comparing the query and patterns using the descriptors, and the best matching patterns are then visually highlighted in the SPLOM (Step 3). The user can now select the patterns most relevant to their needs and the relevance feedback module refines the search parameters based on user's feedback and searches for new patterns (Step 4).

3.1. Model-Based and Shape-Based Descriptors

Shape-based descriptors use shape information to characterise a pattern. The pattern is partitioned into a grid of cells. Then, a 2D histogram is calculated, in which each feature represents the density of points in the corresponding cell. The density is calculated by N_{subset} / N_{total}, where N_{subset} is the number of points in the cell and N_{total} is the total number of points (records) in the pattern. Figure 3a shows a simple 5×5 grid of a pattern. This descriptor is scale-invariant.

Model-based descriptors characterise relationships between points in a pattern. For example, a local regression model estimates the relationship among variables in the local area of a scatterplot. Based on the notion of the regression model, a regression-based descriptor is used. Since ranges of the regression models are not the same, it is not possible to compare two feature vectors created by two different regression models unless values of points are normalised. For this reason and also to keep the descriptor scale-invariant, the points are normalised. Then, a linear, cubic, quadratic or 4th-degree regression model is calculated. Assuming the regression model is $f : X \rightarrow Y$ and $X \in \{0.0, 0.1, \ldots, 0.9, 1.0\}$, the feature vector is built by iterating over the domain X. Therefore, there are 11 features in the feature vector. Figure 3b shows an example of

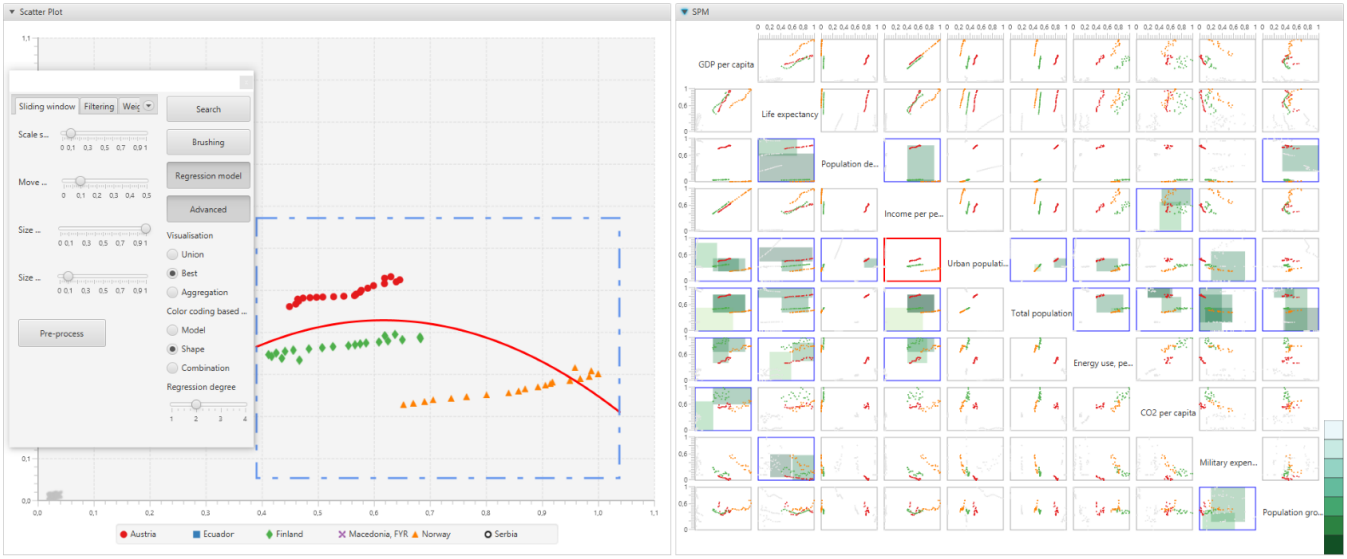


Figure 5: *A snapshot of the application showing the Countries dataset from the World Bank [TWB18]. The scatterplot outlined in red in the SPLOM is selected and is shown enlarged in the scatterplot view to the left. The user has already specified a query pattern and chosen to visualise the patterns in aggregation mode using shape-based descriptor colour-coding. In aggregation mode, all patterns similar to the query are highlighted in the SPLOM. The greenish colour-coding indicates the strength of the similarity according to shape-based descriptors.*

a regression model for a pattern. The function to calculate the distance between two feature vectors is: $\frac{\sum_{i=0}^{n}(|y_1^i - y_2^i|)}{n}$.

3.2. Purity Scores for Pattern Comparison

Since the set of records in the query and a pattern are not always equal, it is useful to consider the similarity of records. Inspired by pattern recognition algorithms, purity scores $P_{precision}$ and P_{recall} are introduced. The $P_{precision}$ is calculated by dividing the total number of records shared between both patterns, N_{shared}, by the total number of records in the target pattern, $N_{pattern}$. This score shows how many matched records exist in the target pattern. In contrast, P_{recall} is defined as N_{shared} divided by N_{query}, which is a total number of points in the query. This score indicates what percentage of records is repeated from the query in the target pattern:

$$P_{precision} = N_{shared} \;/\; N_{pattern} \tag{1}$$

$$P_{recall} = N_{shared} \;/\; N_{query} \tag{2}$$

Both scores are between zero and one. It is possible that the analyst prefers to filter out patterns with low purity scores. Figure 4 gives an example of purity scores. Figure 4a shows a query with seven records. Figure 4b shows a target pattern containing three records from the query. In this case, the purity scores are $P_{precision} = 3 \;/\; 7$ and $P_{recall} = 3 \;/\; 8$.

3.3. Ranking Algorithm

A ranking algorithm is used to obtain a ranked list of patterns similar to a query pattern. The ranking algorithm incorporates the distance between descriptors (Section 3.1) of the query and candidate patterns, as well as purity scores between them (Section 3.2).

In the first step, L_1-distances between descriptors of the query and all candidate patterns are calculated. For each descriptor type, separate distance measures of d_s (shape descriptor) and d_m (model descriptor) are obtained. Both of these distance sets are then min-max normalised to $[0, 1]$. For the ranking, both distances are combined with a parametrised weight and scaling. Subsequently, purity scores are computed among query and candidate patterns to filter the ranking.

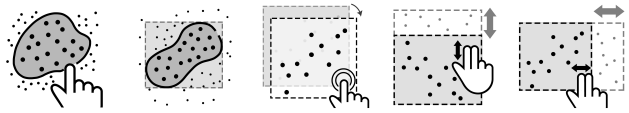
The effective ranking computed by the algorithm can be formalised as a descending ordered list with respect to the similarity score $s : (q, p)$ (Formula 3) between the query pattern q and each candidate pattern p:

$$s(q,p) = \begin{cases} 0 & \text{if } P_{precision} < P_{pmin} \text{ or } P_{recall} < P_{rmin} \\ w_m(1-d_m) + s(1-w_m)(1-d_s), & \text{otherwise} \end{cases} \tag{3}$$

where w_m and $1 - w_m$ are weights assigned to the descriptors, P_{pmin} and P_{rmin} are minimum thresholds for purity scores, and s is an additional distance scaling coefficient. Scaling distances after they have been normalised might further improve ranking quality, since the distribution within the distance spaces of d_m and d_s is likely to differ significantly. This cannot be compensated for by min-max normalising both distributions. In practice, the greatest interest is directed towards patterns having the smallest distances to a query and not towards those patterns yielding the largest distances.

The ranking algorithm can be customised by adjusting several parameters in a graphical dialogue to better reflect the user's notion of similarity for the respective dataset (or domain) at hand:

- $\boldsymbol{w_m}$ within $[0, 1]$ balances the weight between distances d_s and d_m obtained for each descriptor type.
- $\boldsymbol{s}$ is an arbitrary real-valued number that can be used to fine-tune the weighting. In particular, it might compensate distortion for



(a) *Selection* **(b)** *Lens* **(c)** *Moving* **(d)** *Scaling* **(e)** *Scaling*

Figure 6: *Possible user interactions with the scatterplot. (a) The user first draws an arbitrary shape around the points. (b) The bounding box appears around the records. (c) The user can translate the box. (d) By placing two fingers on the top side of the box, the user scales the box vertically. (e) By placing two fingers on the right side of the box, the user scales the box horizontally.*

the first results in case of very differently shaped distributions within the distance spaces obtained by d_s and d_m

$\boldsymbol{P_{pmin}}$, $\boldsymbol{P_{rmin}}$ are thresholds in [0, 1] that steer the filtering of the ranking by purity scores, where 0 disables the filtering. If both are set to 1, only candidate patterns sharing all points with the query will yield non-zero similarity.

shape descriptor resolution is an additional parameter within $\{2, 3, 4, 5\}$ which controls the spatial resolution (number of grid cells) of the shape descriptor.

model descriptor degree is an additional parameter which controls the degree of the polynomial regression used for the model descriptor. In the current system, it is within $\{1, 2, 3, 4\}$.

3.4. Relevance Feedback Algorithm

By examining the ranking algorithm above, the set of parameters involved in computing a similarity ranking of patterns for a respective query can be identified. For example, in Formula 3, the values for w_m, P_{pmin} and P_{rmin} can be any number in [0,1] and the coefficient s can be an arbitrary number. Moreover, four shape-based and four model-based descriptors are used, which affect d_m and d_s.

As mentioned in the previous section, the system provides a graphical dialogue, through which the user may tweak individual parameters of the ranking algorithm. However, for many users without a background in information retrieval, tweaking these parameters is difficult. Even with more in-depth understanding of the ranking algorithm, tweaking the parameters in a meaningful way is highly specific to the characteristics of the currently used dataset.

This issue is addressed by applying a relevance feedback module, which derives parameter values from user-provided examples of similar patterns. After an initial search, the module enables the user to select multiple patterns from the result set, which best match his or her notion of similarity. There is no need for manual parameter tuning, since the user provides feedback by indicating which patterns are most relevant to their current needs. In essence, the user tweaks the parameter set indirectly through relevance feedback. This method is widely used in information retrieval systems.

The relevance feedback module then evaluates the rankings of the selected result patterns with a large number of parameter configurations and selects the configuration C_k which minimises the aggregated rankings of the selected patterns. The operation is shown in Formula 4, where $u_1, u_2, \ldots, u_j$ are the patterns selected by the user and $ranking_{C_k}(q, u_i)$ denotes the ranking position obtained for pattern u_i and query q when a certain parameter configuration C_k is used (see Section 3.3):

$$\underset{C_k}{\operatorname{argmin}}\left(\sum_{i=1}^{n} ranking_{C_k}(q, u_i)\right) \tag{4}$$

To keep the delay in the graphical user interface to a minimum and also to avoid numerical issues in certain cases, the problem is not addressed continuously. Instead, all aggregated ranking scores are computed over a discrete set of possible parameter configurations for the user selected set of similar patterns. Since both the descriptors and the respective distances can be precomputed, a ranking can be computed with almost negligible computational cost (see Section 3.5 for details). It is hence possible to evaluate several thousand parameter configurations without noticeable delay in the graphical user interface.

In the current system, all configurations C_k are obtained as the 6-fold Cartesian product of the discrete value ranges for the individual parameters of the ranking algorithm. In addition to the value constraints for the parameters that are already mentioned in Section 3.3, the weights w_m used for evaluation are within $\{0.0, 0.2, 0.4, 0.5, 0.6, 0.8, 1.0\}$ and the scaling s is within $\{1.0, 1.5, 2.3, 3.4\}$. The numbers are chosen heuristically. In the current implementation, only a single value is used as range for P_{pmin} and P_{rmin}, but both are computed based on the user selected set of patterns according to Formula 5 and 6. In these equations, 0.2 is a number derived heuristically from various search results:

$$P_{pmin} = \min_{u_i}\Big(P_{precision}(q, u_i)\Big) - 0.2 \tag{5}$$

$$P_{rmin} = \min_{u_i}\Big(P_{recall}(q, u_i)\Big) - 0.2 \tag{6}$$

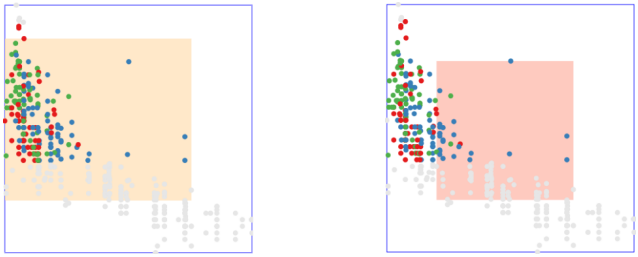
3.5. Complexity of the Algorithm

The run-time of the algorithm has two aspects. The first is the pre-processing required for a particular dataset. The second is the time required to search for patterns during user interaction. For the first aspect, three variables play a role:

Step size for sliding window: Both the translation step size and the scaling step size directly affect the number of patterns extracted per scatterplot. For example, if the translation step size is 0.1 and the scaling step size is 0.2, at most 385 patterns are extracted. In practice, some windows do not include any data points and therefore the number of extracted patterns is reduced.

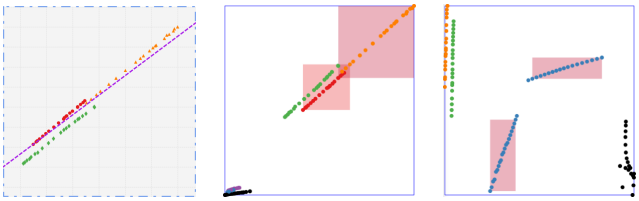
Number of dimensions: If a data set has n dimensions, the maximum number of extracted patterns is $n \times (n-1) \times 385$. For the example step sizes above, a dataset with 10 dimensions would see at most 34650 patterns extracted.

Number of descriptors: For each pattern, a number of feature vectors are extracted. This number directly affects the computation time. In this paper, four feature vectors are used for shape-based descriptors and another four for model-based descriptors. If calculating a feature vector for a pattern costs, say, time t_d, the final maximum time for pre-processing the example dataset above would be $34650 \times t_d$.



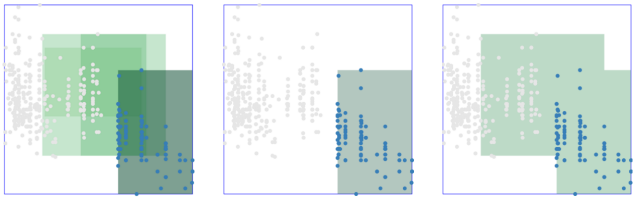
(a) *Search with purity scores.* **(b)** *Search without purity scores.*

Figure 7: *The coloured records (red, green, and blue) are chosen by the user in the query. In (a), the user chooses to filter only patterns having a high similarity of records with the query, while in (b) there is no restriction.*



(a) *The query.* **(b)** *First scatterplot.* **(c)** *Second scatterplot.*

Figure 8: *(a) The user selects a query. Since both shape and model-based feature vectors are scale-invariant, all four patterns in scatterplots (b) and (c) are marked as similar.*



(a) *Aggregation* **(b)** *Best-matches* **(c)** *Union*

Figure 9: *Three different techniques are used to highlight matching patterns in a scatterplot. In (a), all matching patterns are shown in aggregation. In (b), the best matching patterns are highlighted. In (c), the union of all matching patterns is highlighted.*

For the second, run-time aspect of the algorithm during user interaction, the number of parameters and patterns extracted in the pre-processing stage play an important role. For the configuration described in this paper, $7 \times 4 \times 7 \times 4 \times 4$ searches are conducted per pattern. If t_s is the time for one search, then a maximum time of $34650 \times 3136 \times t_s$ is required for one query.

4. System Overview

To support the proposed approach, a prototype application has been implemented on a vertically-mounted Eyevis 84-inch multi-touch display with a resolution of 3840×2160 pixels and a frame rate of 60 Hz [eye18]. This setup is being extended for multimodal, multi-user scenarios. The prototype application is written in Java, using JavaFX for the user interface and the TUIO [KBBC05] and TUIOFX library [FB15] for multi-touch interaction. The application consists of two linked views, a scatterplot, and a SPLOM. The user can open a scatterplot from the SPLOM in a new window or in the existing scatterplot view. The records in scatterplots and the SPLOM are coloured based on their (predetermined) class labels. By using a large, high-resolution display, visualising a multi-dimensional dataset on a SPLOM is supported.

4.1. Constructing a Query

To construct a query, the analyst first selects a scatterplot from the SPLOM. The selected scatterplot is shown on the left panel. As shown in Figure 6, the analyst draws an arbitrary closed shape in the scatterplot to select a set of records (points). While previous studies suggest creating the search query based on a sketch [SBS+14], this proposed free-form selection technique enables the user to search for local patterns. The query is built by fitting the minimum sized rectangle around the selected records. Inspired by the work of Shao et al. [SMSL17], a regression model is visualised within the rectangle as well. In this paper, the regression model is used to help the analyst obtain an abstraction of the pattern for a better understanding of the final query. The abstraction of information is a significant step in information visualisation to reduce cognitive efforts to interpret the data [GAW+11]. The analyst can scale the rectangle to include more records. Moreover, for more fluid interaction with large multi-touch screens, a floating toolbox is provided as shown in Figure 5. The analyst can manipulate the rectangle with one hand and the floating toolbox with the other hand simultaneously. Once selection of query points is finished, the analyst taps the search button to initiate a search.

4.2. Search

After query selection, the system searches for similar patterns in the SPLOM. Patterns in the SPLOM are extracted and all feature vectors are pre-calculated in multiple threads as the dataset is loaded into the application. On a standard PC with an Intel Core i7 CPU and 18 GB of RAM, a dataset containing ten dimensions, 240 records, producing a total of 5941 patterns, requires 26 seconds for pre-calculation. The calculated values are stored in a cache. Each pattern contains four shape-based and four model-based feature vectors.

4.2.1. Set of Patterns

To achieve better performance, a set of patterns is created from the SPLOM just once after the dataset is loaded. A set of patterns per scatterplot is generated by the sliding-window approach. A pattern with less than 5 points is ignored. For a dataset containing 240 records and ten dimensions, the system extracted 5941 patterns. The translation and scaling steps of the window are adjustable by the analyst. As shown in Figure 5, the analyst can change the step sizes by clicking on the Advanced button in the toolbox.

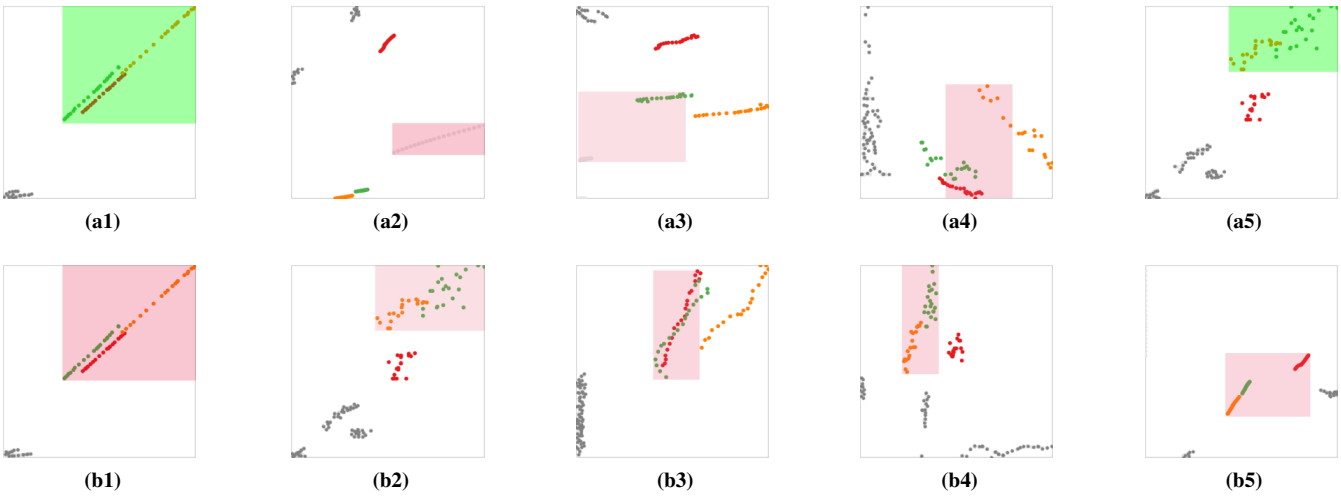


Figure 10: *The results of the query shown in Figure 8a. The first row shows the best matches in the initial search results, where the weight of both shape-based and model-based descriptors are the same. The second row shows the five best matches after the user chose the first and the last matches in the initial search result and the relevance feedback module has adjusted the search parameters accordingly.*

4.2.2. Purity Scores and Feature Vectors

The Advanced options in the floating toolbox allow the analyst to manipulate the parameters of the ranking algorithm. The options are located in the Filtering and Weights tabs. For example, in Figure 7(a), the analyst chose 50% for Min $P_{precision}$ value in the Filtering tab, while in Figure 7(b) the value is set to 100%. Instead of configuring the parameters manually, the analyst can rely on the relevance feedback module.

4.3. Similarity Visualisation

After the similarity search algorithm has determined the distance between items in the set of patterns and the query, the application visualises the patterns. Three techniques are used to visualise them: visualising a differing number of patterns based on the user's need, interactive brushing, and colour-coding based on distance.

Firstly, the user can choose between three options to manage the number of visually highlighted patterns: aggregation, best-matches, and union. As shown in Figure 9a, by choosing aggregation, all similar patterns according to the similarity search algorithm are visualised. This method may show overlapping patterns and creates rectangles that do not exist. To avoid this, the user can select the best-matches option, shown in Figure 9b. In this method, if two rectangles overlap by more than 70%, the more similar pattern will remain. The union in Figure 9c combines all patterns into one shape and the colour of the shape is the average of patterns combined.

Secondly, by brushing the selected points in scatterplots, the similarity scores are visualised. This method makes the user aware of the similarity between selected records and patterns.

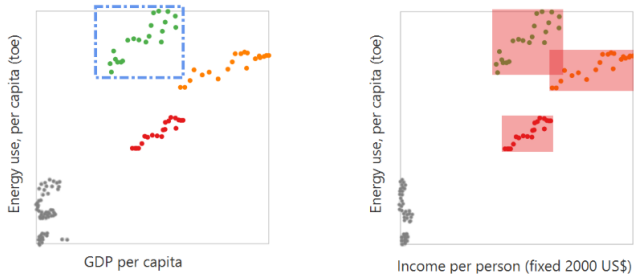
Pandey et al. [PKF+16] showed that judging similarity between plots purely according to their appearance may be misleading. Therefore, thirdly, to avoid relevance feedback favouring shape-based descriptors, a colour-coding function to show similarity according to different descriptors was implemented. Three multi-hue colour palettes are used to indicate the distance between the query feature vectors and pattern feature vectors based on shape-based descriptors (green shades), model-based descriptors (purple shades), and their combination (red shades). Figure 1 and Figure 5 show the colour palettes.

4.4. Relevance Feedback Module

As an example of relevance feedback, the top row of Figure 10 shows the top five query search results from the query in Figure 8a. The second row shows the refinement after relevance feedback. Records in the query have a positive linear relation. In the first row, the system declares five patterns as most similar to the query, whereby $w_m = 0.5$, $P_{pmin} = 0.00$ and $P_{rmin} = 0.00$ were used. The grid size of the shape-descriptor is 2×2 and the regression model in the model-descriptor is quadratic. The user indicates that the first and fifth patterns (Figure 10a1 and Figure 10a5) are relevant and the patterns highlighted in green.

The relevance feedback module determines a new set of parameters in which $w_m = 0.2$, $P_{pmin} = 0.80$ and $P_{rmin} = 0.57$ with a grid size of 3×3 for shape-based descriptors and linear regression model for model-based descriptors. The pattern in Figure 10a2 is ignored since the similarity score thresholds are not satisfied. The patterns in Figure 10a3 and Figure 10a4 are similar to the query by the model-based descriptor, but since the weight for it is low, they are taken out of the ranking. Also, the $Purity_{recall}$ and $Purity_{precision}$ are below the thresholds. As shown in the second row of Figure 10a3, the new patterns are more visually similar to the query.

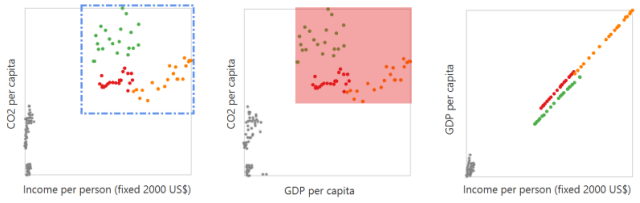
This example shows the usefulness of the relevance feedback module. Since the search algorithm is scale-invariant, some found patterns have a significantly different slope to the query. The system does not know the meaning behind units within scatterplots, therefore the scaling and angle of slope may not be meaningful.



(a) *The initial query (Finland) in the scatterplot of Energy Use against GDP per Capita.*

(b) *Similar patterns are found in the scatterplot of Energy Use against Income per Person.*

Figure 11: *Three local patterns are found, indicating similar behaviour of three countries in another scatterplot.*



(a) *The initial query.*

(b) *A similar pattern in another scatterplot.*

(c) *A local positive correlation.*

Figure 12: *Since the result pattern looks very similar to the initial query, a possible relationship might be expected between the two horizontal scatterplot dimensions.*

5. Use Cases

Any multivariate dataset can be loaded into the application. In this section, three query scenarios are discussed using a subset of the Countries dataset from the World Bank [TWB18]. The dataset contains ten dimensions and 126 records. The dimensions are attributes of countries such as GDP per Capita, Life Expectancy, and Population Growth. The records contain information about each country for each year between 1995 and 2015. Here, the data for six countries was used: Austria, Ecuador, Finland, Macedonia, Norway and Serbia. Figure 5 shows the dataset as visualised in the system. All three query scenarios were refined using the relevance feedback module. For simplicity, only the initial query and final results are presented.

The first query scenario is shown in Figure 11. The user selected all data points corresponding to Finland (green) in the scatterplot of Energy Use plotted against GDP per Capita. After refining the search with relevance feedback, three similar patterns belonging to Austria, Finland, and Norway were found in the scatterplot of Energy Use against Income per Person. This result is perhaps to be expected, since all three nations are developed countries in Europe and GDP and Income per Person are generally related. The parameters of the search are $P_{pmin} = 0.00$, $P_{rmin} = 0.00$, $s = 1.0$, $w_m = 0.5$, 4×4 resolution of the shape-based descriptor, and a linear model-based descriptor.

After investigation of the first query, the user decides to investigate the scatterplot of CO2 per Capita against Income per Person. This time, the user selects all three aforementioned countries to form the initial query, as shown in Figure 12. As may have been suspected, the same pattern is found in the scatterplot of CO2 per Capita against GDP per Capita. Again, the user suspects a possible relationship between Income per Person and GPD per Capita. The assumption is valid since there is a local positive correlation for these three countries, as shown in Figure 12c. After checking with other similar queries, the user concludes that the Income per Person dimension is redundant and that keeping GDP per Capita is sufficient for their purposes. The parameters of the search are $P_{pmin} = 0.80$, $P_{rmin} = 0.80$, $s = 1.0$, $w_m = 0.2$, 3×3 resolution of the shape-based descriptor, and a quadratic model-based descriptor.

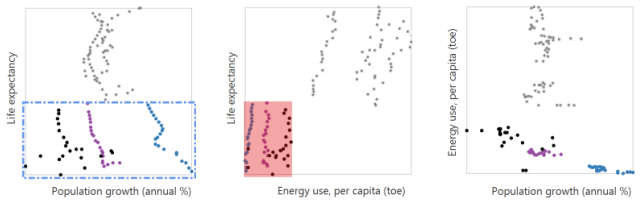
The user continues to explore the dataset by examining the scatterplot of Life Expectancy against Population Growth, as shown in Figure 13a. The user selects the data points comprising Macedonia, Serbia, and Ecuador at the bottom of the scatterplot. The most similar pattern is located in the scatterplot of Life Expectancy against Energy Use. This pattern looks like a flipped version of the query. The user decides to inspect the scatterplot of Energy Use against Population Growth to look for any local relationship between records in the query. As shown in Figure 13c, a local negative correlation exists between the points of these three countries, but no correlation is apparent when all of the points in the scatterplot are considered. The parameters of the search are $P_{pmin} = 0.59$, $P_{rmin} = 0.65$, $s = 1.0$, $w_m = 0$, 5×5 as the resolution of the shape-based descriptor, and a quadratic model-based descriptor. Since $w_m = 0$ in this search, the model-based descriptor is ignored.

6. Discussion and Future Work

The presented approach allows users to search for similar local patterns in a set of scatterplots, helping users explore multidimensional datasets by comparing patterns. More specifically, the approach focuses on finding related patterns with regard to shape-based and model-based similarity across different regions, dimensions, and record subsets of a larger SPLOM space.

Users can initiate a search by interactively selecting a region of a scatterplot as an initial query. An obvious extension would be to include a sketch-based interface where query patterns can be sketched in free form. Currently, the user must manually inspect the SPLOM to find a suitable query pattern in a scatterplot. A more scalable approach for a larger SPLOM would be to include a clustering step to identify representative local patterns (e.g., using density-based clustering). Then, an overview of cluster prototypes could be offered to the user to choose a query pattern, optionally editing this using sketching or by blending with other prototype patterns.

Through experimentation, it was found that similarity search of scatterplot patterns depends on the chosen descriptors and dataset. When inappropriate descriptors are used, the search results may be perceived as dissimilar by the user, although they are similar according to the definition of the descriptors. A relevance feedback approach allows users to tune search parameters implicitly, steering the system towards a notion of relevance fitting their current



(a) *The initial query.* **(b)** *A similar pattern in another scatterplot.* **(c)** *A local negative correlation.*

Figure 13: *There is some similarity between the initial query on the left and the matching local pattern on the right. There is a local negative correlation between the two horizontal scatterplot dimensions.*

needs. In informal experiments, it was observed that this approach can return more relevant patterns. More formal evaluation regarding information retrieval measures would be interesting. The usability of the relevance feedback interface needs to undergo some more formative evaluation. In addition, a comparative study of the system with and without relevance feedback would be desirable.

Since the algorithm is designed for general use cases, some false positive patterns are found, due to not having a clear definition of patterns for a specific dataset. These matches can be excluded later by manual parameter tuning or use of the relevance feedback module. However, in future, it would be desirable to incorporate active learning into the relevance feedback module, instead of one-time feedback from the user. Also, the user might be able to deselect undesired patterns rather than selecting the relevant matches.

An underlying question of scatterplot similarity is how the perception of patterns in scatterplots by analysts can be modelled, and eventually described by descriptors. In [PKF+16], an experiment to assess how analysts describe specific patterns in scatterplots was presented, which found that these were not easy to model using Scagnostics features. The shape and model-based descriptors used in the proposed approach are one choice, but additional features could be defined (or even learned from training data) to describe patterns more compatible with user perception and interpretation.

Since the descriptors are parameters in the approach, additional ones can be added to the system in the future. It would be particularly interesting to learn which descriptors work better with which kinds of dataset. After experimenting with L1, L2, and Quadradic Form distance functions, L1 was found to be good enough for the chosen datasets, possibly because of the coarseness of the descriptor's grid. It is possible that for other datasets and use cases, other distance functions might work better. Thus, it could be interesting to add distance functions as a parameter.

Since a sliding-window approach is used, the search may return many possible positions and areas of similarity within a given scatterplot. A simple rectangle is currently used to highlight matching patterns. It would be interesting to research more advanced visual representations of local matches in a scatterplot. Also, if the data is labelled, instead of using the sliding-window approach, algorithmic complexity could be reduced by only comparing clusters of records with the same label, especially when the scatterplots are dense. Conducting visual cluster separation in scatterplots would be another alternative to the sliding-window approach.

Finally, it would be interesting to develop ground truth and benchmark datasets to further compare algorithms for local pattern discovery in scatterplots, adding to existing benchmarks for global scatterplot features [SvLS12].

7. Concluding Remarks

This paper presented an approach to search for local patterns in the scatterplots of a scatterplot matrix. Model-based and shape-based descriptors are used to compare the initial query pattern with other patterns. Relevance feedback is then used to refine the search. An implementation of the approach has shown its usefulness for various datasets and works in near real-time. Finally, the limitations and possible extensions of the approach were discussed.

Acknowledgment

This work was partially funded by the Pro^{2}Future Competence Center for Products and Productions of the Future. Pro^{2}Future is funded within the Austrian COMET Program – Competence Centers for Excellent Technologies – under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Science, Research and Economy and of the Provinces of Upper Austria and Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

[BUS10] Beecks, C., Uysal, M. S., and Seidl, T. "Signature Quadratic Form Distance". In *Proc. ACM International Conference on Image and Video Retrieval (CIVR 2010)*. (Xi'an, China). 2010, pages 438–445. doi:10.1145/1816041.1816105.

[CCM+14] Chen, H., Chen, W., Mei, H., Liu, Z., Zhou, K., Chen, W., Gu, W., and Ma, K.-L. "Visual Abstraction and Exploration of Multi-Class Scatterplots". In *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pages 1683–1692. doi:10.1109/TVCG.2014.2346594.

[CCM10] Chan, Y.-H., Correa, C. D., and Ma, K.-L. "Flow-Based Scatterplots for Sensitivity Analysis". In *Proc. IEEE Symposium on Visual Analytics Science and Technology (VAST 2010)*. 2010, pages 43–50. doi:10.1109/VAST.2010.5652460.

[EAM14] Eisemann, M., Albuquerque, G., and Magnor, M. "A Nested Hierarchy of Localized Scatterplots". In *Proc. Conference on Graphics, Patterns and Images (SIBGRAPI 2014)*. 2014, pages 80–86. doi:10.1109/SIBGRAPI.2014.14.

[ESBB98] Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. "Cluster Analysis and Display of Genome-Wide Expression Patterns". In *Proc. National Academy of Sciences* 95.25 (1998), pages 14863–14868. doi:10.1073/pnas.95.25.14863.

[eye18] eyevis. *EYE-LCD-8400-QHD-V2*. Apr. 13, 2018. http://eyevis.de/en/products/lcd-solutions/4k-ultra-hd-lcd-monitors/84-inch-4k-uhd-lcd.html.

[FB15] Fetter, M. and Bimamisa, D. "TUIOFX—Toolkit Support for the Development of JavaFX Applications for Interactive Tabletops". In *Proc. International Conference Human-Computer Interaction (INTERACT 2015)*. Springer, 2015, pages 486–489. doi:10.1007/978-3-319-22723-8_44.

[FR97] Friedman, N. and Russell, S. "Image Segmentation in Video Sequences: A Probabilistic Approach". In *Proc. 13th Conference on Uncertainty in Artificial Intelligence (UAI'97)*. Morgan Kaufmann, 1997, pages 175–181.

[GAW+11] Gleicher, M., Albers, D., Walker, R., Jusufi, I., Hansen, C. D., and Roberts, J. C. "Visual Comparison for Information Visualization". In *Information Visualization* 10.4 (2011), pages 289–309. doi:10.1177/1473871611416549.

[KBBC05] Kaltenbrunner, M., Bovermann, T., Bencina, R., and Costanza, E. "TUIO: A Protocol for Table-Top Tangible User Interfaces". In *Proc. 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005)*. 2005. http://mtg.upf.edu/node/408.

[MG13] Mayorga, A. and Gleicher, M. "Splatterplots: Overcoming Overdraw in Scatter Plots". In *IEEE Transactions on Visualization and Computer Graphics* 19.9 (2013), pages 1526–1538. doi:10.1109/TVCG.2013.65.

[MJE02] Mann, R., Jepson, A. D., and El-Maraghi, T. "Trajectory Segmentation Using Dynamic Programming". In *Proc. 16th International Conference on Pattern Recognition*. Volume 1. 2002, pages 331–334. doi:10.1109/ICPR.2002.1044709.

[MTL18] Matute, J., Telea, A. C., and Linsen, L. "Skeleton-Based Scagnosticss". In *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2018), pages 542–552. doi:10.1109/TVCG.2017.2744339.

[NAW13] Nhon, D. T., Anand, A., and Wilkinson, L. "TimeSeer: Scagnostics for High-Dimensional Time Series". In *IEEE Transactions on Visualization and Computer Graphics* 19.3 (2013), pages 470–483. doi:10.1109/TVCG.2012.128.

[NW14] Nhon, D. T. and Wilkinson, L. "PixSearcher: Searching Similar Images in Large Image Collections through Pixel Descriptors". In *Proc. International Symposium on Visual Computing (ISVC 2014)*. Volume 8888. LNCS. Springer, 2014, pages 726–735. doi:10.1007/978-3-319-14364-4_70.

[PKF+16] Pandey, A. V., Krause, J., Felix, C., Boy, J., and Bertini, E. "Towards Understanding Human Similarity Perception in the Analysis of Large Sets of Scatter Plots". In *Proc. 2016 CHI Conference on Human Factors in Computing Systems*. ACM. 2016, pages 3659–3669. doi:10.1145/2858036.2858155.

[SBS+14] Shao, L., Behrisch, M., Schreck, T., von Landesberger, T., Scherer, M., Bremm, S., and Keim, D. A. "Guided Sketching for Visual Search and Exploration in Large Scatter Plot Spaces". In *Proc. EuroVis Workshop on Visual Analytics (EuroVA 2014)*. 2014. doi:10.2312/eurova.20141140.

[SBS11] Scherer, M., Bernard, J., and Schreck, T. "Retrieval and Exploratory Search in Multivariate Research Data Repositories Using Regressional Features". In *Proc. ACM/IEEE Joint Conference on Digital Libraries (JCDL'11)*. ACM, 2011, pages 363–372. doi:10.1145/1998076.1998144.

[SMSL17] Shao, L., Mahajan, A., Schreck, T., and Lehmann, D. J. "Interactive Regression Lens for Exploring Scatter Plots". In *Computer Graphics Forum* 36.3 (2017), pages 157–166. doi:10.1111/cgf.13176.

[SP07] Schreck, T. and Panse, C. "A New Metaphor for Projection-based Visual Analysis and Data Exploration". In *Proc. Electronic Imaging Conference on Visualization and Data Analysis*. Volume 6495. SPIE. 2007. doi:10.1117/12.697879.

[SSB+16] Shao, L., Schleicher, T., Behrisch, M., Schreck, T., Sipiran, I., and Keim, D. A. "Guiding the Exploration of Scatter Plot Data Using Motif-Based Interest Measures". In *Journal of Visual Languages & Computing* 36 (2016), pages 1–12. doi:10.1016/j.jvlc.2016.07.003.

[STMT12] Sedlmair, M., Tatu, A., Munzner, T., and Tory, M. "A Taxonomy of Visual Cluster Separation Factors". In *Computer Graphics Forum* 31.3pt4 (2012), pages 1335–1344. doi:10.1111/j.1467-8659.2012.03125.x.

[STMT15] Sedlmair, M., Tatu, A., Munzner, T., and Tory, M. "Data-Driven Evaluation of Visual Quality Measures". In *Computer Graphics Forum* 34.3 (2015), pages 201–210. doi:10.1111/cgf.12632.

[SvLS12] Scherer, M., von Landesberger, T., and Schreck, T. "A Benchmark for Content-Based Retrieval in Bivariate Data Collections". In *Proc. 2nd International Conference on Theory and Practice of Digital Libraries*. Volume 7489. LNCS. 2012, pages 286–297. doi:10.1007/978-3-642-33290-6_31.

[SvLS13] Scherer, M., von Landesberger, T., and Schreck, T. "Visual-Interactive Querying for Multivariate Research Data Repositories Using Bag-of-Words". In *Proc. 13th ACM/IEEE Joint Conference on Digital Libraries*. (Indianapolis, Indiana, USA). 2013, pages 285–294. doi:10.1145/2467696.2467705.

[TFJ12] Tominski, C., Forsell, C., and Johansson, J. "Interaction Support for Visual Comparison Inspired by Natural Behavior". In *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), pages 2719–2728. doi:10.1109/TVCG.2012.237.

[TMF+12] Tatu, A., Maass, F., Färber, I., Bertini, E., Schreck, T., Seidl, T., and Keim, D. "Subspace Search and Visualization to Make Sense of Alternative Clusterings in High-Dimensional Data". In *Proc. IEEE Conference on Visual Analytics Science and Technology (VAST 2012)*. 2012, pages 63–72. doi:10.1109/VAST.2012.6400488.

[TWB18] TWB. *Countries and Economies*. The World Bank Group. Apr. 2018. https://data.worldbank.org/country.

[WAG05] Wilkinson, L., Anand, A., and Grossman, R. "Graph-Theoretic Scagnostics". In *Proc. IEEE Symposium on Information Visualization (InfoVis 2005)*. 2005, pages 157–164. doi:10.1109/INFVIS.2005.1532142.

[YWS+14] Yates, A., Webb, A., Sharpnack, M., Chamberlin, H., Huang, K., and Machiraju, R. "Visualizing Multidimensional Data with Glyph SPLOMs". In *Computer Graphics Forum* 33.3 (2014), pages 301–310. doi:10.1111/cgf.12386.