Making Machine Intelligence Less Scary for Criminal Analysts:

Reflections on Designing a Visual Comparative Case Analysis Tool

Wolfgang Jentner · Dominik Sacha · Florian Stoffel · Geoffrey Ellis · Leishi Zhang · Daniel Keim

Abstract A fundamental task in Criminal Intelligence Analysis is to analyze the similarity of crime cases, called Comparative Case Analysis (CCA), to identify common crime patterns and to reason about unsolved crimes. Typically, the data is complex and high dimensional and the use of complex analytical processes would be appropriate. State-of-the-art CCA tools lack flexibility in interactive data exploration and fall short of computational transparency in terms of revealing alternative methods and results. In this paper, we report on the design of the Concept Explorer, a flexible, transparent and interactive CCA system. During this design process, we observed that most criminal analysts are not able to understand the underlying complex technical processes, which decrease the users' trust in the results and hence a reluctance to use the tool. Our CCA solution implements a computational pipeline together with a visual platform that allows the analysts to interact with each stage of the analysis process and to validate the result. The proposed Visual Analytics workflow iteratively supports the interpretation of the results of clustering with the respective feature relations, the development of alternative models, as well as cluster verification. The visualizations offer an understandable and usable way for the analyst to provide feedback to the system and to observe the impact of their interactions. Expert feedback confirmed that our user-centered design decisions made this computational complexity less scary to criminal analysts.

Keywords Crime Intelligence Analysis · Visual Analytics · Clustering · System Design · Human-Computer Interaction · Sequential Pattern Mining · Text Analysis · Dimensionality Reduction

Wolfgang Jentner*
E-mail: wolfgang.jentner@uni.kn
ORCID: 0000-0003-1045-6020

Dominik Sacha*
E-mail: dominik.sacha@uni.kn

Florian Stoffel*
E-mail: florian.stoffel@uni.kn

Geoffrey Ellis*
E-mail: geoffrey.ellis@uni-konstanz.de

Leishi Zhang
Middlesex University, London, UK
Tel.: +44-208-411-2843
E-mail: l.x.zhang@mdx.ac.uk

Daniel Keim*
E-mail: daniel.keim@uni.kn

*Address & Telephone
Universitätsstraße 10, 78464 Konstanz
Tel.: +49 (0) 7531 88-3064

1 Introduction

Comparative Case Analysis (CCA), also called Similar Fact Analysis (SFA) [29] is an important tool for criminal investigation and crime theory extraction [28]. Given a collection of crime reports, the idea is to analyze the commonalities between crime cases in order to support reasoning and decision making. For example, examining solved crimes that have similar characteristics as an unsolved crime may help the analyst generate a new hypothesis during a criminal investigation, and understanding the uneven distribution of crimes in terms of

offender/s unknown approached school changing rooms, from side of building, opened insecure fire exit door, gained entry, stole items belonging to football teams, mainly money and jewellery, made good their escape.

Fig. 1 A typical Modus Operandi (MO) of a burglary crime report. The extracted terms are underlined.
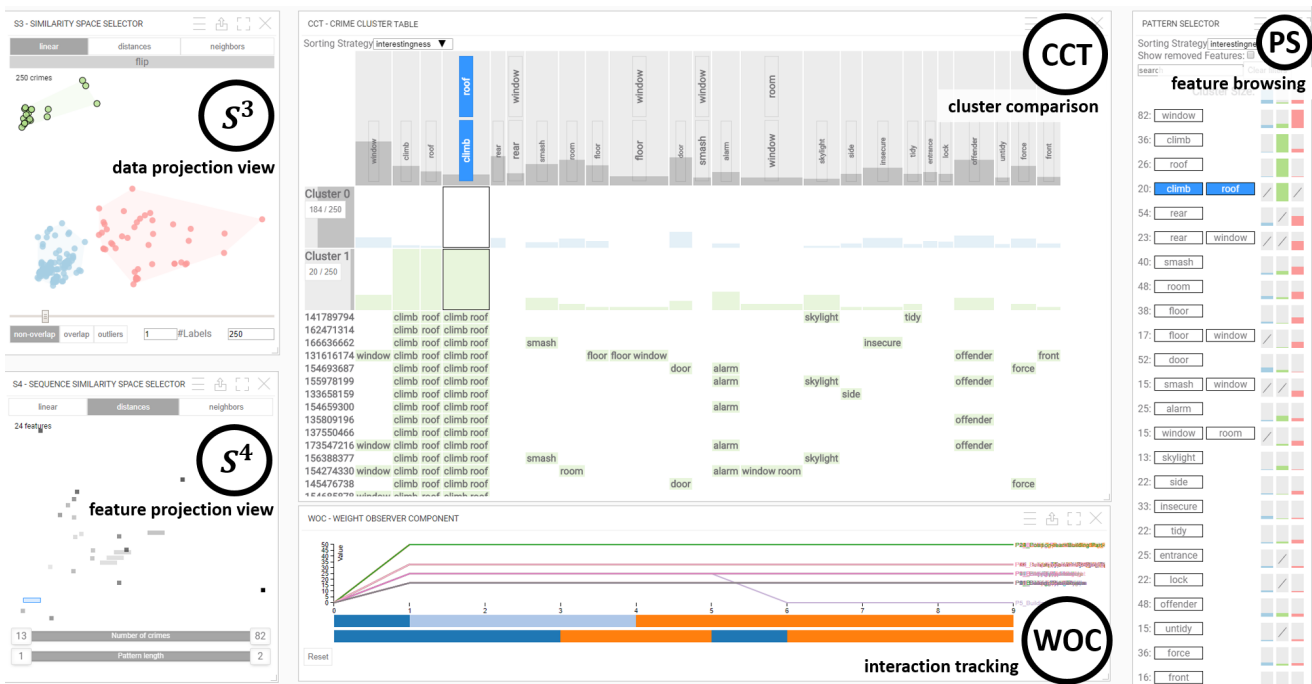
Fig. 2 Concept Explorer: A visual, interactive interface for Comparative Case Analysis. Crime cases and clusters are shown in the center within the Crime Cluster Table (CCT). On the left hand side, a hybrid analysis perspective on the data and feature space is provided: A two-dimensional embedding of the crime similarities and the clustering is shown in the Similarity Space Selector (S^3). Another two-dimensional embedding of the feature similarities based on the shared crimes is shown in the Sequence Similarity Space Selector (S^4). The respective features are also shown in the Pattern Selector (PS) on the right hand side. Tracked interactions and configurations are displayed in the Weight Observer Component (WOC). All views are linked and allow criminal analysts to develop and verify alternative clusterings from different tightly integrated perspectives.

spaces, types of offenders and victims may help the police to allocate police resources more effectively [12]. The latter lies in the responsibility of a Tactical Analyst (TA) who examines sets of crimes periodically to find new trends. CCA starts with the extraction of relevant headings (features) that are considered to be useful for the understanding of the crime cases. Information is then collated under the headings, resulting in a CCA table where each row is a crime case. As well as common headings such as day of week or time of day, the main focus is on extracting concepts from free text fields such as the Modus Operandi (MO) (see Figure 1). Manually analyzing an excessive number of such crime cases (extracting and analyzing the relevant information for each crime) is a tedious and complex task for criminal analysts. General purpose analysis tools (e.g., IBM I2 [20], Jigsaw [36]) and existing analysis approaches from text or high-dimensional data analysis (e.g., Ruppert et al. [30] and Jäckle et al. [21]) can be applied to criminal intelligence analysis. However, most of the work does not allow the analyst to develop and validate computational alternatives (transparency of the results) and does not allow the user to form the familiar structured CCA tables. In many real world data analysis scenarios, it is necessary to iteratively improve, adapt and combine a set of analysis methods to solve the analysis task. This

results in complex pipelines that need to be analyzed from different perspectives.

In this paper, we present our ongoing research on the design of a visual comparative case analysis tool called the Concept Explorer, which comprises several component views (Figure 2). The work is part of the EU-funded project "Visual Analytics for Sense-making and Criminal Intelligence Analysis (VALCRI)" [4]. The aim of the project is to develop a Visual Analytics (VA) system to improve the effectiveness of current criminal intelligence analysis solutions. According to our police partners, traditionally CCA is carried out manually on a spreadsheet. The task becomes increasingly difficult due to the growing volume and complexity of today's crime data. When introduced to automatic analysis techniques, such as feature extraction and clustering, that could help with the analysis tasks, the analysts found them "scary", principally due to the lack of understanding of the algorithms and the impossibility to examine alternatives. In order to design a CCA tool that capitalizes on the machine intelligence and at the same time provides sufficient level of usability, we designed our system in close collaboration with one police officer with an extensive analysis background, and received feedback on a regular basis from several police forces across Europe. The system design is based on a number

of analytical tasks we derived through the discussion with our end users, including:

Task 1. Understand Cluster Characteristics: A major task of CCA is to identify groups of crimes that have similar patterns and to understand the key features that define their main characteristics.

Task 2. Develop Alternative Clusterings: The analyst needs to be able create several alternative clustering results.

Task 3. Verify Cluster Robustness: The analyst needs to verify the robustness and stability of clustering result. This includes examining changes of grouping caused by different feature weightings (i.e., removing or adding features) as well as checking if the clustering result is stable across different computation methods.

Driven by these tasks, we designed a VA approach in a user-driven design study with TAs and other domain experts. The system instantiates the process model for interactive Dimensionality Reduction (DR) proposed by Sacha et al. [33] with the aim to provide an interactive visual platform for the analyst to examine groups of similar crimes as well as their main characteristics.

The main contribution of this paper is a comprehensive and flexible criminal intelligence analysis tool that implements a hybrid analysis approach to interactively analyze the data and feature space in parallel (Figure 2). The system takes free text fields of crime reports as input, extracts key features from the reports using a series of NLP techniques, calculates frequent sequences of the key features, and allows the analyst to select features of interest and set their weight/importance for similarity computation. The result is displayed in a 2D data similarity space that can be clustered. The features are displayed simultaneously in a similar fashion allowing the exploration and interpretation of the feature space. A table, inspired by the traditional spreadsheet table, combines the two spaces and enables the tactical analyst to undertake a CCA.

Additionally, we elaborate the design process that was carried out over a period of 2 years and we are able to report and subsequently reflect on four major design phases. Figure 3 depicts this system evolution with four instances of the framework. Each step embeds the DR pipeline (bottom row) in an iterative exploration process (right) with several ways to provide interactive feedback to the underlying analytics (top row).

The work is based on a previous short paper publication which described an intermediate state of our current solution [31]. The work was published at the international EuroVis workshop on Visual Analytics (EuroVA). The related work in the next section high-

lights various related VA systems as well as systems specialized on criminal intelligence analysis. Section 3 details four phases of the design process showing the evolution of each component and ultimately the whole system. We describe the current system in Section 4 and explain how the components are embedded into the VALCRI framework. Additionally, we present a use case, reflecting on its use by TAs. Feedback from the experts on the current system is reported in Section 4.2 and then lessons learned during the design process are summarized in Section 5. Conclusions are drawn in Section 6 and future perspectives are outlined.

2 Related Work

Our analysis approach combines many analytical techniques, such as textual feature extraction, sequential pattern mining, high-dimensional data analysis, and visual interactive clustering applied to criminal intelligence analysis. We illustrate these with examples.

2.1 Comparative Case Analysis

Comparative Case Analysis (CCA) is based on the notion of comparison, which is a fundamental technique used by many social science and scientific domains [11]. CCA starts with processing the text to extract key features, followed by reasoning and sense making based on similarity comparison. One challenge of CCA is to extract features. In the literature this is a manual process as presented by Bennell et al. [7] who manually extracted features from Modus Operandi (MO) of 86 solved commercial burglaries committed by 43 serial offenders in order to compare the similarity between burglary case. The findings were used to examine if a high-degree of similarity between them enables different cases to be validly linked to a common offender. This requires a significant amount of work even with this relatively small amount of data. Another challenge is the comparison. Given a set of crimes, what to compare and how to compare has to be decided by the analyst [29]. Work carried out by Canter et al. [10] used the *Jaccard coefficient* to measure the proportion of co-occurring features in crimes. The work also applied *multidimensional scaling* on the data to investigate the consistency of features across organized and disorganized cases. The research revealed that disorganized features were either easy to identify or occur more commonly, probably due to their vast number compared to organized features. To the best of our knowledge, no work has been reported on automatic feature extraction, feature selection and weighting for CCA.

2.2 Automated Feature Extraction for CCA

For the feature generation, we use a custom framework based on components from Stanford CoreNLP [26] and Apache OpenNLP [1]. For characterization of concepts and automated class assignments, two different resources, Wordnet [27] and Framenet [6] are used. Besides customized retrieval and classification methods, the analytic parts are based upon state-of-the-art techniques as described by Manning et al. [25] or Jurafsky and Martin [24].

For our system, we use a Sequential Pattern Mining (SPM) algorithm to mine for frequent sequences of terms occurring in the MO of the crime reports (see Figure 1 for an example). The problem was formally defined by Agrawal et al. [5]. In order to avoid redundant patterns, we mine for a set of closed sequential patterns [19, 43]. We use a Dimensionality Reduction (DR) on the mined frequent patterns and visualize them in a feature similarity space. Similarity measures for sequential patterns exist [34], however, in order to be consistent with the data similarity space, we use a binary feature vector containing the crime reports where a bit is set to one if the sequence occurs in that crime report.

2.3 Visual Analytics for CCA

Automatic analysis methods such as feature extraction, pattern mining, clustering and DR provide effective means of analyzing large amount of crime data and extract patterns in it. But Visual Analytics (VA) tools for supporting CCA are scarce. Software systems such as IBM I2 [20] and Jigsaw [36] were developed for the general purpose of Criminal Intelligence Analysis but little work has been carried out to improve on the manual CCA process. Jäckle et al. proposed a projection based approach [21] for analyzing similarity between textual data items but the approach does not allow police officers to form the customary structured tables. The Spherical Similarity Explorer system developed by Zhang et al. [45] allows the analyst to project crime data onto a spherical surface for similarity analysis. The tool focuses on one DR algorithm with limited interaction possibilities.

2.4 Interactive Visual Machine Learning

A VA system should effectively involve the analyst by interacting with the data and the models at different stages of the analytical pipeline in order to iteratively improve, adapt, and combine analysis methods to solve the analysis task at hand [32]. Recent work by Sacha et al. [33] surveyed existing visual DR tools and highlighted interaction possibilities to improve the effectiveness of the tools. The interpretability of results and the usability

of interactive DR systems, especially for domain expert users (without technical and data analysis background) is a major area for improvement.

Existing visual text analytics approaches such as IN-SPIRE [40] (and its predecessors [15, 8]), or recent works described by Ruppert et al. [30], shed light on the possibility of automatically processing textual documents to obtain and explore document clusters. These systems adopt different DR and/or clustering techniques to generate visual embeddings of the high-dimensional data to enable the analyst to compare the similarity between data items and examine interesting patterns in the data. Given that DR and clustering are complex processes that involve a series of selection, computation and validation, input from human analyst is often beneficial and largely unavoidable. Wenskovitch et al. [39] provide an overview of how to combine DR and clustering and also recommend design decisions.

2.5 Hybrid Views

Hybrid views, also often referred as dual views, aim to provide simultaneous access to the data and feature space. Van der Corput and Van Wijk [38] are using I^F-F^I tables to support access to both spaces. Turkay et al. [37] and Yuan et al. [44] use two tightly coupled scatter plots. We follow this strategy by creating these scatter plots through DR. However, additionally we use one table were both, data and features, are combined and the clusters generated in the data space can be interpreted. Demiralp [13] uses a heatmap-matrix diagram in combination with a scatter plot in order to interpret clustering results. We follow this approach, however, we utilize bar charts in a table to enable the user to perform a cluster comparison.

3 Design Study Methodology

We adopted a design study methodology [35] to iteratively build and refine our Visual Analytics (VA) approach based on several rounds of expert feedback from different user groups. We worked in close collaboration with one expert with a data analysis background on a regular basis while we conducted less frequent expert evaluations with different police forces. In the early phases, the feedback was provided as we demonstrated prototypes or versions of the tool to small expert groups. Later, the experts had to use the tool to perform particular tasks with a given data set. We are able to reflect on four major design phases:

Phase 1. Proof of Concept: The research focus was to test if we can extract useful features from the given
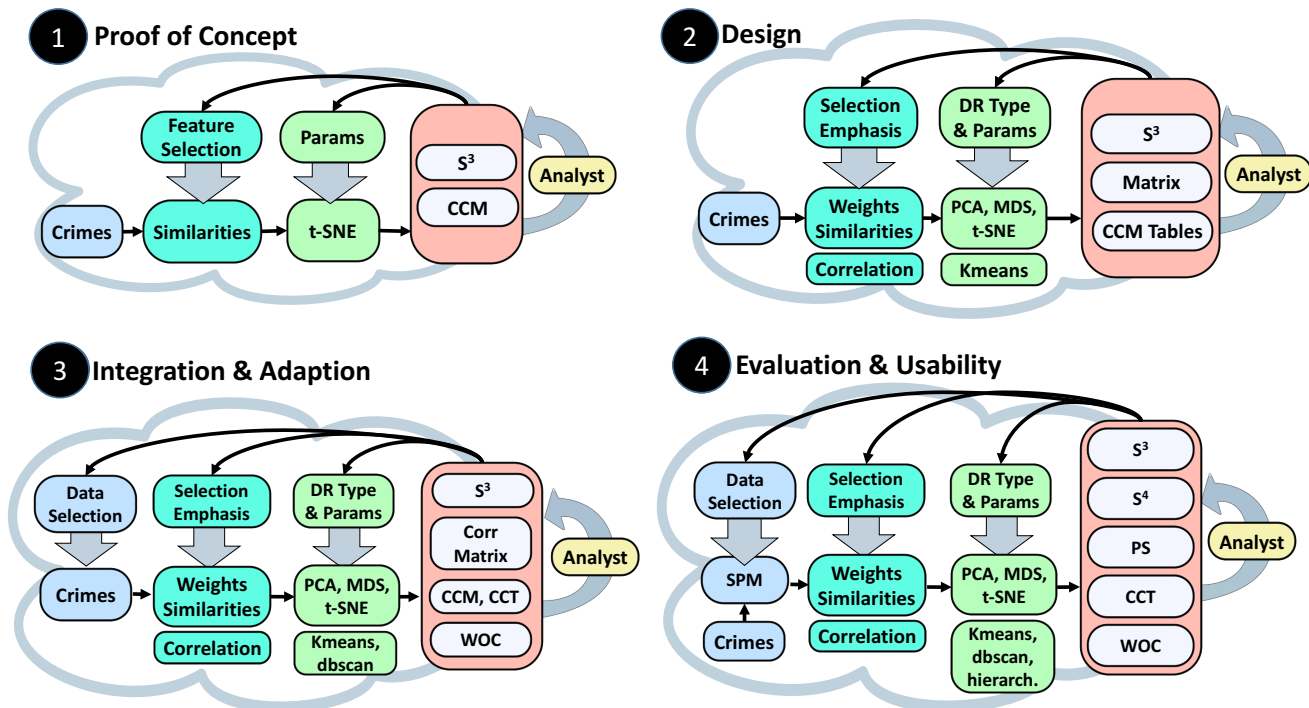
Fig. 3 The four phases of our design study. Each is represented by an instantiation of the visual interactive Dimensionality Reduction (DR) process described by Sacha et al. [33]. The pipelines show what core-parts were modified and extended during the phases. The pipeline delineates (from left to right) data (blue), pre-processing and feature generation (turquoise), DR and clustering methods (green), and visual interfaces (red). End-user involvement increases as the project progresses.

crime cases and if DR makes sense. The result was a basic pipeline (Figure 3, Phase 1) with some basic interactive visualizations: a 2D projection and a crime matrix which represent early versions of S^3 and CCT (Figure 2).

Phase 2. Design: We enriched the computational and interaction capabilities. The resulting system calculated similarities and crime clusters based on a variety of configurations and parameterizations with visual components for each pipeline stage (S^3, Matrix, CCM Tables - as described later in this section). However, during guided demonstrations of the prototype, the criminal analysts were overwhelmed by the apparent computational complexity.

Phase 3. Integration & Adaption: All developed components were integrated into the consortium's VAL-CRI framework. With the additional components, such as a geographic map or timeline, we could investigate some more realistic use cases together with our domain experts. This resulted in reducing the computational complexity and simplifying the user interface, as stated in our previous work [31] (Figure 3, Phase 3).

Phase 4. Evaluation & Usability: Observations during task-based evaluation sessions with crime analysts from our partner police forces were particularly useful, especially as one of our users reported being "scared to death" by the clustering when they were asked to

work independently with the system. Various changes were introduced to alleviate this. Henceforth, the User Interface (UI) strictly separated different configurations and the parameter tuning process was greatly simplified. We incorporated a Sequential Pattern Mining (SPM) algorithm and extended our analysis in order to improve the feature selection and emphasis process. A second perspective on the feature space (hybrid analysis of data and feature space) was added to help the users understand relations and patterns of crime clusters. The current version of the system is shown in Figure 3, Phase 4.

Phase 5. End-User Training & Evaluation: With the current prototype, we are now transitioning into the end-user training and evaluation phase, with the aim to further improve some UI elements.

The following sections describe the evolution of the major components of these design phases, giving an insight into some of the important design decisions made.

3.1 Feature Generation

We developed a text analytics solution that generates feature vectors for each input Modus Operandi (MO) text (Figure 1), where each position in the feature vector
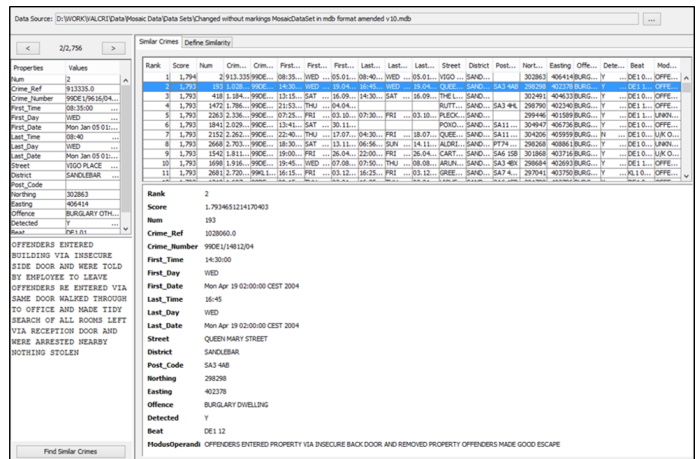
1 **Proof of Concept**
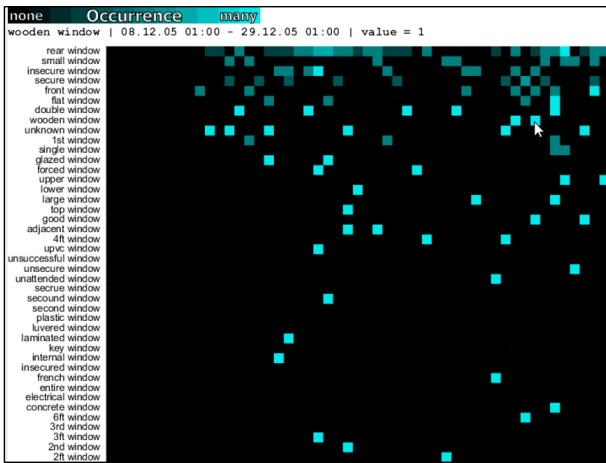
2 **Design**

Fig. 4 Phase 1 shows a visual interface to visualize occurrences of adjective noun combinations where the frequency is mapped to colors from black (zero occurrences) to cyan (max occurrences). A prototype to demonstrate similarity in the generated feature space is displayed in phase 2. Phase 3 and 4 are omitted because no further visual prototype was developed. Instead, the feature extraction was refined with experts and a sequential pattern mining algorithm was introduced.

refers to a concept indicating the occurrence of a term in the input data. This effectively generates the headings of a Comparative Case Analysis (CCA) table, automating the data processing conducted by police analysts.

Phase 1: To find some structure in the text data, we implemented a basic Natural Language Processing (NLP) pipeline, based on part-of-speech tagging and a lemmatizer to cope with inflected terms. This counted occurrences of distinct, adjective and noun combinations. The visualization (Figure 4, Phase 1) was a heat map-like matrix view, that displayed each unique adjective plus noun combinations (rows) per input document (columns). This provided an overview of our analysis results for a large number of documents. However, this generic approach yielded too many results to be of use, although the approach of representing the MO text fields by short extracts was regarded as possibly useful.

Phase 2: To reduce the number of possible features, we experimented in two different directions: 1) implemented a tf-idf based term weighting scheme [24]; and 2) designed a data and offense-specific text analysis matching system, that utilized domain knowledge in order to identify relevant parts of the MO. Domain experts provided good insights into the structure of current CCA table creation and what they regarded as potentially relevant features. We implemented a prototype using the selected terms for similarity-based retrieval of data records (see Figure 4, Phase 2), where the cut-off of terms and selections of the types of term combinations (bigrams and trigrams) could be adjusted. Using a variety of settings, police analysts compared the results with the actual crime reports. However, the feedback sessions were not

promising, as the results were considered to be fairly random.

Phase 3: It became clear, that the CCA task uses features and concepts that are very specific to the offense and, hence, we started to integrate sets of concept terms in the text analysis process. A two-stage analysis process was developed, that firstly identifies all possible combinations of corresponding terms (unigrams, bigrams, and trigrams) and then created the text feature space based on offense specific concept lists. As the VALCRI crime dataset was mostly burglaries, a relevant set of concept terms were collected in order to demonstrate the text analysis process to our end-users. These included 8 different types of concepts, for example, parts of a building, colors, and a frequency-based list from the dataset. We discussed missing or wrongly identified features with the domain experts.

Phase 4: We extended the amount of expert knowledge by refining the concept lists (feedback from Phase 3). Experiments with the S^3 prototype (see Section 3.2) showed, that some of the created features were too fine-grained to be of good use for the CCA process. Additionally, crime investigators are, for example, interested in crimes where a burglar enters the building through a window. A simple extracted term *window*, however, does not provide enough information as it probably includes crimes where the burglar exited through the window. To cope with such problems, we experimented with a SPM algorithm, which extracts frequent sequences of terms as they occur in the crime reports [22]. The order of the terms is important but gaps are allowed to filter out extremely infrequent terms that would obstruct a

sequence from being frequent. Sequences allow a more fine grained similarity space. For example, the sequence *window steal door* probably describes only MOs where a burglar enters through a window and exits through a door. The experts reported that features consisting of term-sequences are useful as they provide a better picture of the MO. We make use of a minimum support parameter ($minSup$) set to 5%, so a pattern has to occur in 5% of the crimes to be included in the results. Hence, reducing the amount of data will reveal more fine-grained patterns containing more terms (see Section 4.1).

3.2 Similarity Space Selector – S^3

S^3, a *data projection view,* provides a simple interface for the crime investigators to understand the relations and similarities among multiple crimes across different DR and clustering results. It represents the two-dimensional data space with crimes arranged according to feature similarities (i.e., if they contain similar crime patterns).

Phase 1: A first prototype explored the capability of DR techniques to spread out the data based on the sparse extracted concepts. Initially we used the number of occurrences of each concept in a crime and tried out different feature combinations (e.g., building parts of a house combined with actions or movements, e.g., *smashed door*). The web-based t-SNE implementation [3] provided promising results (Figure 5, Phase 1) with each dot representing one crime-report. The central view shows the current projection while the small multiples on the side offer alternatives, based on different concept sets, that might show promising patterns to the user. Feedback showed that the actual number of occurrences were not important and that users have difficulties in identifying the extent of possible clusters in the projected space.

Phase 2: The underlying data structure was changed to a binary feature vector where each bit represents one term. The effectiveness of other DR algorithms were explored. Figure 5, Phase 2 shows the different DR results for PCA [17], MDS [9] and t-SNE [2] in the columns left to the main plot. The rows show projections for specific feature configurations (e.g., all features or subsets of specific concept families, such as movements, colors or building parts), which formed a matrix of small multiples. Clicking on any of these small multiples moved it to the central view, with the previous one joining a history list. The visual clustering was improved by coloring a convex hull (and the points) based on the results of k-means clustering [17]. The CCA table (see Section 3.3) is tightly coupled with the S^3 component,

showing concept data from crime reports in selected clusters. However, experiments demonstrated that users' trust in the system was low because they did not understand the projection techniques. Although we anticipated that the multiple plots would generate interesting patterns, they just added confusion.

Phase 3: To simplify the UI, we reduced the component to its main view, giving the user the option to change the DR algorithm, and also added an animated transitions between the results (Figure 5, Phase 3). Initially, clusters were recomputed directly after changing either the importance (weight) of a term or applying a different DR algorithm. However, it was difficult for analysts to keep track of the changes, despite using animated transitions, and this reduced their understanding of the impact of their actions. Automatic re-clustering was therefore disabled which allowed our users to track the animated transitions of the dots (crimes) and clusters. More importantly, it was now possible to track and assess the cluster robustness (by investigating the distortion of the cluster hull and the crime colors) and the impact of the changes (e.g., feature removal or change to type of DR). This step was a quantum leap for the users to better understand the different DR techniques and to understand the cluster dynamics. The user must actively re-cluster the data to obtain new cluster colors and hulls. Here, the animated transitions of the colors and the cluster hulls are also helpful. The DBSCAN clustering technique [17] was added, which provided better results in some cases. However, it was still up to the analyst to select a clustering technique and to enter the respective parameters (k for k-means, or *eps* and $minPts$ for DBSCAN). The transition from Phase 2 significantly reduced complexity for the user interface but manual parameterization still confused some users to such as extent that one end-user stated: "Your clustering scares me to death!".

Phase 4: Figure 5, Phase 4 illustrates that the UI has been further simplified. Buttons were renamed (PCA to *linear*, MDS to *distances* and t-SNE to *neighbors*) giving the user a better understanding of what they can expect from the different DR techniques. A similar measure was taken for the visual clustering algorithms: k-means to *non-overlap* (because the clusters are clearly separated), hierarchical clustering [17] to *overlap* (this clustering algorithm was added to allow overlapping clusters) and DBSCAN to *outliers* (as this density based algorithm allows outliers that are not part of any cluster). Moreover, the parameters for each clustering algorithm were replaced by one single slider to control the respective parameters. The slider has the same semantics as it shows more clusters when it is dragged to the right.
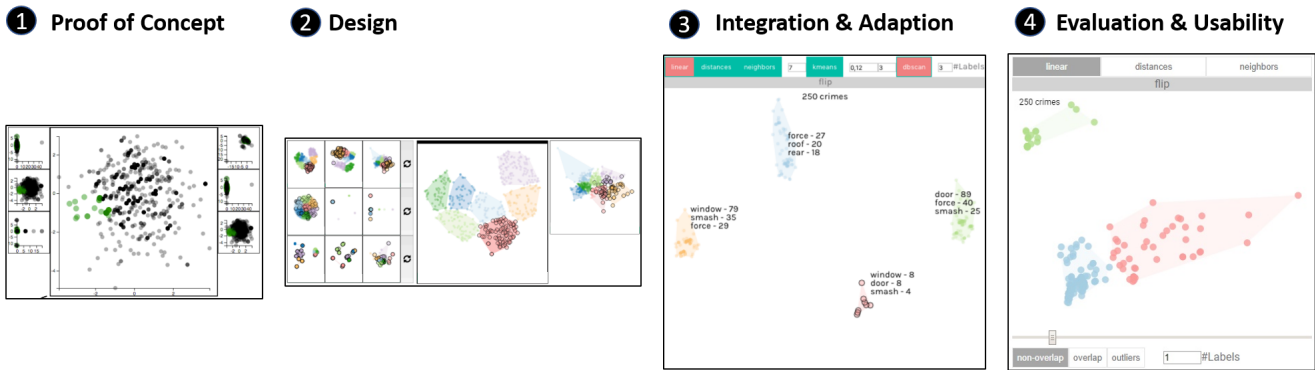
Fig. 5 The evolution of S^3 beginning with a proof of concept to test the usefulness of dimensionality reduction for displaying crime report similarities. We experimented with different DR and clustering algorithms and weight models in the second design phase. Phase 3 shows the integration and simplification of the components to focus on one projection. The user interface complexity was further reduced in Phase 4, for example, by introducing a simple slider interaction for all visual clusterings and by strictly separating different computations (DR – top, clustering – bottom).

Pre-computing all clusterings for specific parameters enables our users to explore the parameter spaces of the respective clusterings with a simple slider interaction providing instant feedback. Another beneficial measure is the clear distinction between the DR techniques (on the top) and the visual clustering techniques (below).

3.3 Crime Cluster Table – CCT

The CCT supports the CCA task and is, therefore, a central component of our UI. Crime investigators manually maintain such spreadsheets where crimes are listed with user identified crime characteristics as columns.

Phase 1: Our first visualization was a *Crime Comparison Matrix* (CCM, Figure 6, Phase 1). Here, the extracted concept-term combinations (e.g., *Building Part→door*) are displayed as rows (sorted in a descending order according to their overall frequency) and the crime cases as columns. The cells are color-coded when the respective term occurs in the crime report. The analysts had no difficulty in understanding the component. However, it showed that such a matrix does not scale well with many crimes and/or many features and that a single representation cannot be used for comparing different sets of crimes.

Phase 2: Multiple CCMs (Figure 6, Phase 2) were used to support the comparison task of the clusters that are generated by the S^3 component (*data projection view*). The coloring of the cells corresponds to the color of clusters in S^3. The users welcomed this step as it enabled them to interpret the clusters. We understood that the scalability issue is not yet solved.

Phase 3: In a first step, the multiple CCMs were reduced into a single CCM (Figure 6, Phase 3). The cells are colored according to the respective cluster and the

feature weights are mapped to the labels' font-size. The latter can be changed in steps by mouse clicks. To cope with the scalability issues, a secondary view called the *Crime Classification Table (CCT)* was developed, taking advantage of the two-level hierarchy in the features (e.g., *Building Part→door*). Here, only the concepts are displayed in separate rows while the corresponding terms, if they occur in the crime report, are directly written into the cells. Semantic zooming was introduced that decreased the font-size of the views and reduced the spacing. This enabled many crimes/features on a single display but could make it hard to read the labels. The users stated that they felt comfortable with this view and thus, were bolder to experiment with the given features. We observed that the users mainly counted how often a feature occurred in a cluster.

Phase 4: In the transition to this phase, we learned that one key to understanding a cluster is to check how frequent a feature occurs in a cluster and whether it does not occur in any other cluster. Such distinctive features are interesting to the users and can be steered by their weights. The CCT shown in Figure 6, Phase 4 was modified to display aggregated clusters as rows [31] and the feature frequencies as columns. This simplified the display considerably, whilst still making it possible to investigate individual crimes by expanding clusters. The overall frequency of a feature is displayed in the header via a bar chart in the background of each label. Similarly, the size of the clusters (number of crime reports) is displayed in each cluster summary row. The features (columns) can be sorted by frequency, alphabetically, or interestingness. The latter favors the most distinctive features of the clusters (sorting left to right) in order to support the analyst in interpreting the differences of the clusters (Section 1, Task 1). To reflect the change
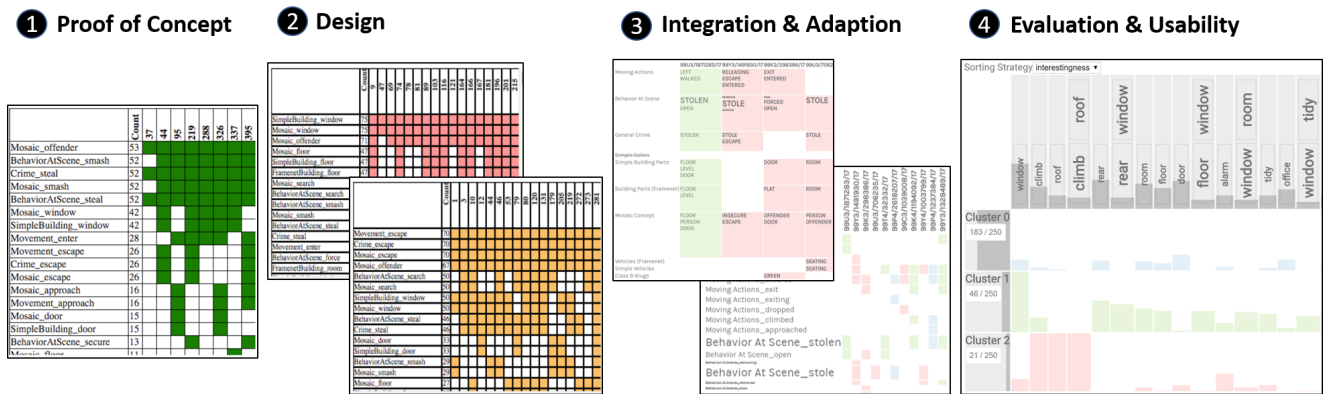
Fig. 6 The evolution of the CCT component starts with an instantiation of a CCA spreadsheet table. Multiple instances are created in Phase 2 to compare different clusters. The table is modified in Phase 3 to increase its scalability. In Phase 4, the table is optimized for comparing clusters.

of emphasis, the component was renamed as the *Crime Cluster Table (CCT)* (Figure 2).

3.4 Sequence Similarity Space Selector – S^4

S^4 is a *feature projection view* that offers an important perspective on the feature space, supporting the feature selection and emphasis task to improve the data clusters of the projections in the *data projection view* S^3. The visual clusters are only as good and useful as the features, therefore, the user needs to understand feature characteristics of the analyzed dataset including overlaps, redundancies, correlations and outliers.

Phase 1: An early version of this view (Figure 8, Phase 1) consisted of a basic feature frequency lists. Even though it did not reveal any similarities, the overall frequency of a feature is important for an analytical task as detailed later on in this section. We experienced that similarity is difficult to visualize in a list view. Therefore, we started to develop two separate components. The following phases will outline the evolution of the similarity view. The further development of the list view is detailed in Section 3.5.

Phase 2: During the design phase, we focused on revealing similar features and added a correlation-matrix, as depicted in Figure 7, Phase 2. The matrix shows highly co-occurring features in a saturated blue color and mutually exclusive features in a saturated red color. The correlation values for hovered feature combinations are shown in a tool tip. Negatively correlating features can be used to split the data, whilst positively correlating features are largely redundant. As the latter tend to dominate the projection, users could directly remove them by clicking on the respective cells.

Phase 3: The matrix did not scale well to an increasing feature space and showed many uninteresting, less correlating feature pairs, which are not regarded as that

useful by the experts. To overcome these problems, we created a correlation sorted list where all combinations with a correlation between −0.3 and 0.3 are omitted (Figure 7, Phase 3). This simplified the navigation and allowed the users to spot the interesting correlations faster.

Phase 4: User feedback revealed that more fine grained feature combinations are needed to distinguish crime reports (see Section 3.1, Phase 4). With experience of the *data projection view* S^3, the users better understood the concept of DR. Therefore, we provided a similar view for the feature space: the *Sequence Similarity Space Selector* (S^4) which illustrates the similarities of the features (sequential patterns) based on shared data items (crimes). The length of a sequence (number of terms) is mapped onto the length of the rectangles and the support of a feature is mapped onto the opacity (Figure 7, Phase 4). To help the user better distinguish between data and feature space, all items in the feature space use a rectangular shape and no color coding while the data space uses rounded corners. The bottom part of the component is replaced by two range sliders. The upper one sets the frequency range so it is possible to exclude features with either low or high frequencies; for example in the removal of outliers (see Section 4.1, Step 5). The lower slider sets bounds for the length of a sequence. Longer sequences are typically more specific and, thus, less frequent. To provide rapid feedback, the chart is updated whilst dragging the slider. On release, the DR is recalculated and the updated weights are propagated to the other components.

3.5 Pattern Selector – PS

The Pattern Selector allows the user to browse and explore multiple feature patterns.
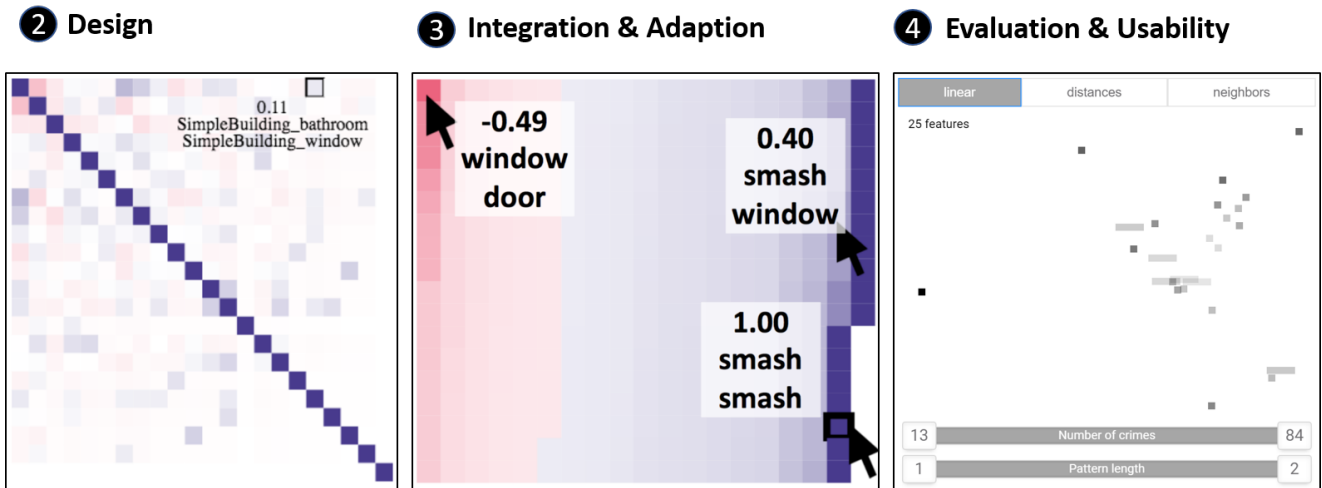
Fig. 7 S^4 evolution: Starting with a simple list showing the frequency of the features (see Figure 8, Phase 1). Phase 2 introduces a correlation matrix where highly co-occurring features are displayed in blue, negative correlations are displayed in red. Phase 3 shows a wrapped list instead of the matrix to simplify the navigation. The component is drastically changed in Phase 4 and uses the analogy of the S^3 component showing similarities in the feature space.
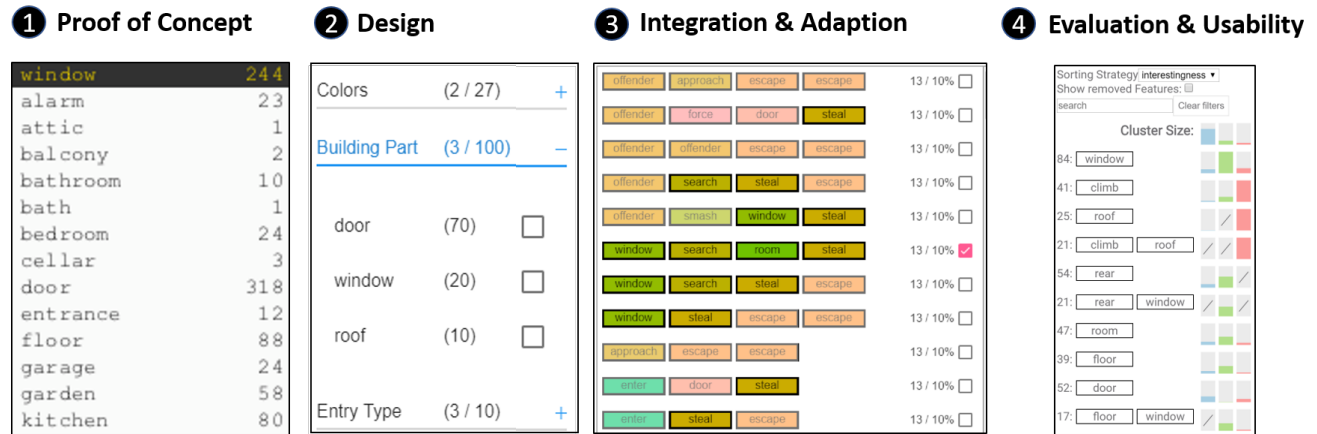

Fig. 8 The evolution of the Pattern Selector component starts with a list displaying the features and their frequencies. A simple frequency list is used in Phase 1. In Phase 2, a table is created that makes use of the two-level hierarchy of the features. A separate prototype uses a list to display sequences. This prototype is integrated and slightly modified to show the clustering information as generated by S^3.

Phase 1 & 2: In Phase 1, this component was equal to the feature space component (see Section 3.4, Phase 1). In Phase 2 (Figure 8), the concept selector leverages the two level hierarchy of the terms (e.g., *door*) and their corresponding concepts (e.g., *Building Part*). Both are represented in a table where concepts can be expanded to view the individual terms. The number of occurrences, as well as the selection, were displayed accordingly. Deselected terms set the corresponding weights, used in the DR, to zero. Disabling a concept disables all underlying terms.

Phase 3: The first external prototype using the sequences [22] was not linked with the *data projection view* S^3, however, the end-users stated that sequences containing three or more items provide enhanced information to understand the underlying MO. In Figure 8,

Phase 3, the sequential ordering of terms is shown for each pattern. Additionally, the terms are color coded to enable users to detect patterns in the sequences. The list can be filtered by entering a term in the search field.

Phase 4: The prototype was integrated into the VAL-CRI framework and revised based on user's feedback (Figure 8, Phase 4). The colors were removed to comply with the design decision of not using colors in the feature views (see Section 3.4, Phase 4). The number of occurrences is displayed in the first column. The list can be sorted as in the CCT (see Section 3.3, Phase 4). The cluster information is displayed as bar charts representing the frequency of the feature in each cluster. The color of the clusters is linked to S^3 (*data projection view*) and the top row shows the size of each cluster.

Clicking a feature in the pattern selector provides several actions: i) sets the weight (0 removes from display), ii) filter for crimes that contain this feature, and iii) filter for crimes that do not contain this feature. This drill-down operation can be performed multiple times, with all other components updating automatically (see Section 3.1, Phase 4 and Section 4.1).

3.6 Weight Observer Component – WOC

The WOC (Figure 2) provides analytic provenance [42] and captures user interactions [14]. It was initially designed as a tool for the developers to track and understand how the Concept Explorer was being used. It tracks and visualizes the feature weights in a multi line chart and the DR and clustering configuration in state-history charts (Figure 2). The end-users did not find it particularly useful but suggested that it could be part of a reporting feature, outlining their exploration of the data. The Security, Ethics, Privacy & Legal (SEPL) board highlighted its crucial role in court cases when analysts have to justify their decision making. We also observed that the component can be useful as a bookmarking feature to save and load configurations and feature weights for specific analytical tasks.

4 The Concept Explorer

The Concept Explorer's components are embedded into the VALCRI framework which provides additional components such as a timeline or a map view. The Concept Explorer targets the structurization of the Modus Operandi (MO) through feature extraction (Section 3.1) and the exploration of crimes, as well as the extracted features. The VALCRI framework features a web-based dashboard design [41] in the front-end and a Java-based back-end to perform more complex operations such as the Dimensionality Reduction (DR) and clustering. The user can open a canvas on a screen, each with multiple components which can be arranged and resized freely. The components are tightly coupled to provide a better analytical understanding of the data and its features. In general, hovering over a feature (e.g., in the *feature projection view S^4*) will highlight the feature in other components (e.g., Pattern Selector) with all crime reports, described by that feature, highlighted as well. Similarly, hovering over a crime report highlights features within that report. This linking and brushing capability is important in understanding the influences of features in the data similarity space (see Section 4.1). The Similarity Space Selector (S^3, *data projection view*) creates the clusters and the cluster information is broadcast to the other components, such as the Crime Cluster

Table (CCT). Filters can be applied by all components to reduce the crime report data set and enable users to drill down for a specific set of crime reports containing a user-defined set of features.

4.1 Use Case

The crime set being investigated is normally specific to a region and a time range and this can be obtained with the respective components available in the VALCRI framework (map and timeline). Additionally, the set is filtered by search terms to receive similar types of crimes. This use case emerged from our experiences within collaborative evaluation sessions with different user groups. It comprises a set of different analytical steps that have been considered useful to solve a variety of analytical tasks. The following use case demonstrates this functionality and has been conducted in multiple paired-analytics and user training sessions.

Step 1: The user is interested in burglaries in *schools*. After opening the data and feature projection views S^3 and S^4, the hybrid view is arranged as in Figure 9, Step 1. S^4 (left) shows three exposed dark quadratic rectangles representing to three feature sequences, containing a single term, that occur frequently. These terms are *door* (red), *rear* (blue), and *window* (green). The fact that these features are exposed and are highly saturated, suggests to the user that the data similarity space visible in S^3 (right) is mainly separated by these features. We have annotated the regions where the crime reports are located in the same colors as in S^4. The linking and brushing features of the components are used to obtain this insight.

Step 2: The user is further interested in these features and increases the weights for the features *door* and *window* and applies a new clustering to better distinguish the crime reports (Section 1, Task 2). The results are visible in Step 2.1 where S^3 shows four clusters. The yellow and the green clusters, circled in green, contain crime reports with the feature *window*. The yellow and the red clusters, circled in red, contain *door*. The blue cluster, on top, does not contain any of the these features. All clusters contain crime reports with the feature *rear* meaning that the similarity space is currently not separated by this (Section 1, Task 1). The user is also interested in the feature *rear* and therefore increases its weight. The *data projection view* updates immediately resulting in the view given in Step 2.2 (note that all clusters are rather distorted). The lower part of the clusters, circled in blue, consists of crime reports containing the feature *rear*. The user manually triggers a re-clustering and also increases the number of clusters using the lower slider in S^3 (Section 1, Task 2). The result can be seen

Fig. 9 A frequently observable use case starting with the initial data. The user identifies the main features separating the data space (Step 1) and increases the weights (importance) of interesting features and defines a new clustering (Step 2). A CCA analysis with detail on demand follows (Step 3). The cluster robustness across different DRs is tested afterward (Step 4). The use-case ends with a drill-down operation including the pruning of the feature space (Step 5).

in Step 2.3. This sub-workflow presented in Step 2 can be frequently observed. We call it "cluster-mitosis".

Step 3: The feature-characteristics of the clusters can be examined using the Pattern Selector (Step 3.1) and the CCT (Step 3.2) (Section 1, Task 1). Cluster 5 (Figure 9, Step 3.2 (dark-blue)) contains only crime reports that have all three features. This cluster is located in the bottom center location in S^3 (Step 2.3). With the CCT (Step 3.2), the user can now perform typical CCA tasks, such as comparing the features of the clusters to spot interesting co-occurring features. The feature sequences *rear window* and *rear door*, framed in purple, are only present in clusters 3, 4 and 5 where the single-term sequences are present. The bars displaying the frequency of the feature in the clusters are not full, showing that there are some crimes which contain the feature sequence *door rear*. But this sequence is too infrequent (less than 5% of the crimes; see Section 3.1, Phase 4) to be in the feature result set and therefore is not visible as a column in the table. Furthermore, the gray bars in the header, show that the feature sequence *rear window* is more frequent than *rear door*. The user expands cluster 5 in the CCT to inspect the individual crime reports. A similar view is visible in Figure 2. By clicking on one crime report, the crime-card opens showing more details of that crime including the MO (Step

3.3). As this cluster only contains crime reports holding all three features, the user can find these features in the text. Due to the order of the terms, the crime also contains the sequences *rear door* and *rear window*.

Step 4: The user checks the other projection methods (Section 1, Task 3) such as "distance" (MDS) and "neighbors" (t-SNE). Whilst the "distances" only shows that the clusters expanded a little (Step 4.1), the "neighbors" projection shows a different picture (Step 4.2). This projection favors neighborhoods and therefore shows identical crimes in non-overlapping rings. These *crime-rings* can be important in the users' analysis. The user learns that there are a few crime-rings containing the feature sequence *climb roof*. The feature is highlighted in Step 4.2 (right side) and the crime-reports are highlighted with a black border in Step 4.2 (left side). The user is further interested in these crimes and filters the crime data set on the feature *climb roof* (drill down).

Step 5: The remaining dataset contains 46 crime reports. However, the set of features has increased to 110 because a pattern must occur in at least 5% of the crime reports to remain in the result set which are only two crimes in this case. These longer and more specific sequences can be already interpreted by the experts without the need to read the MO (Section 3.5, Phase 3). S^4 shows an outlier circled in yellow in Step 5.1. At
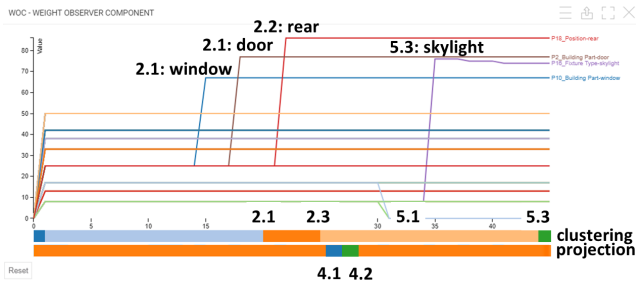
Fig. 10 The use case as it is recorded in the WOC component. The line chart represents the weights of the features. The upper state-history chart represents the clustering. The different colors represent different clustering algorithms or changes in the parameters. The lower state-history chart shows changes of the DR algorithms.

this location, three features *climb*, *roof* and *climb roof* are overplotted. These features are outliers since they describe all crimes in the result set. Thus, these features are uninteresting and do not influence the DR in S^3. The user removes these features by changing the range slider as indicated by the red arrow in Step 5.1. The user sets the range of the support for a feature to *2 - 40*. Features that have less or more occurrences are removed by setting their weights to 0. This change only affects the outliers. The remaining feature set contains 107 features and their similarity space is redrawn (Step 5.2). Note that this did not change the data similarity space in S^3. Whilst browsing the features in the Pattern Selector, the user spots one feature *climb roof skylight* and repeats the cluster-mitosis step to obtain a cluster for this feature. These features are described by the red cluster (Step 5.3). They are overplotted in S^4 (Figure 9, Step 5.2 (purple circle)) because they are redundant.

This use case was captured by the WOC (Section 3.6) as shown in Figure 10. Going from left to right, it shows the weights (importance) were increased for the features *window* and *door*. The upper state-history chart then shows that a new clustering was triggered manually (change from light blue to orange). Afterwards, the weight for feature *rear* was increased in Step 2.2. A re-clustering was executed in Step 2.3 which is visible in the change of color from orange to light-orange in the upper state-history chart. The user experimented with the projections as shown in the lower state-history chart (Steps 4.1 and 4.2). The Tactical Analyst (TA) proceeded with a drill-down for *climb roof* and then removed uninteresting features (their weight was changed to 0 in Step 5.1). The cluster-mitosis step was repeated with the feature *skylight* for Step 5.3.

The use cases represents a possible workflow, highlighting many of the features of the Concept Explorer. As a toolset, the respective Steps (1-5) can be freely combined and repeated to explore the data in depth.

4.2 Expert Feedback

We obtained feedback for the current system from a software developer and data analyst developing solutions for police forces (internal expert). We also presented the system to criminal investigators of the German police who are not part of the consortium (external experts) and did not have any training on the system.

The experts reported that the extracted features are relevant for their analytical tasks and the navigation in the data and feature space is easy due to the rapid updates and the linking and brushing capabilities. The crime investigators stated that without any user training the system seems to be very complex at first glance, however, it is definitely relevant for TAs as it can provide a much better overview for (large) sets of crimes. Tooltips provide relevant details on demand. Additionally, it avoids the cumbersome and very time consuming manual extraction of the features. The internal expert states that the addition of the S^4 and Pattern Selector components provides valuable functionality and are highly relevant for the tasks of the TAs. All experts agree that the existing instantiation of the CCA table provides easily understandable access to the data and feature space and is suitable to perform CCA tasks. The external experts remarked that it would be necessary for users to add new concepts to the concept lists when dealing with new MOs and crime types.

The WOC is criticized as it looks complex and labels are missing. The experts affirm that such a component is useful for the analyst and others in order to explain the decision making. However, the actual numbers for the weights are necessary in order to generate reports.

5 Lessons Learned

The system was developed in collaboration with domain experts over a period of 2 years. Hence, we are able to enumerate observations and lessons learned.

Our initial User Interface (UI) comprised multiple scatter plots that show visual embeddings using different configurations (Dimensionality Reduction (DR) types, feature subsets, etc.). Without much training, our end users reported that it was difficult to understand the results obtained with the different settings. They considered the concept of DR to be very abstract and found it hard to interpret and trust the result shown in scatter plots where the "meaning of axis" is missing. Although there exists work by Gleicher [18] to provide meaningful axes, this issue became irrelevant over time as the users learned that only the distance, not the actual position

is meaningful. As the CCT UI was familiar in some respects, it helped the users interpret the results of DR and visual clustering. We also recommended that the criminal analysts start with a few clusters that can be interpreted more easily. This allowed them to anchor particular areas (or specific crimes) in the projection which could be related to some specific patterns (e.g., a particular area contains all the "roof" crimes). The cluster slider allows them to easily increase the number of clusters whilst the animated transitions enables the tracking of positional changes. Interacting with the system and observing the changes helped the analysts to understand how the methods work and how they can interpret the results. While the users always showed interest, it was observable that the main interactions drifted to the spreadsheet tables even though such a comparison required much more effort. Due to the simplifications of the S^3 component, plus a training effect over time, the users became more confident and experimented more with the component. However, we also learned that it is essential to provide the analysts with tools they are familiar with (e.g., the spreadsheets) and the interpretablity of the results is the key to building trust in the system, as is providing useful interactive feedback. It is also worth mentioning that the system helped us (as developers) to understand the extracted data. We realized that some features occur with high frequency while others are very sparse. The user can now refine this with the help of the S^4 component using the range sliders. Training of the users with the S^3 component made this transition easier.

We experienced that consistency in such a complex system is vital. Users will immediately ask why certain linking and brushing capabilities are not working in one component or working differently in another. This consistency includes design decisions such as shapes and colors. Consistency is also fundamental in helping the user relate and navigate the data and feature spaces. Combining these two in the CCT provides additional context. As with many systems, tooltips are a useful aide-memoire as well as an easy method to show details on demand.

Users typically show interest in novel components and methods. With guidance and explanations the experts were able to solve the tasks reasonably well. However, on their own, the users were much more hesitant because they missed guidance from the system or an analyst. To overcome this, substantial effort has to be put in to lowering the complexity of a component whilst not sacrificing too much functionality. This includes changing labels so they express what the users can expect when they interact with it. Here, users and developers should agree on terms for the labels. Additional demon-strations and training helped the user to build trust in the system.

Most of the users are not interested in technical details. Others are interested in everything, but as non-computer scientists, they are unlikely to understand complex concepts and importantly, we should not expect them to understand. We learnt that it is better to explain how to react in certain situations. For example, during demonstrations we explained that the feature space of the S^4 component "does not look nice" as much space is wasted when there are outliers (see Section 4.1, Step 5.1). In this situation, the user has to rate the interestingness of such features and react with defining a different weight for it. A clear separation in the UI helps the users to remember the associated tasks.

Automation comes at a cost to interpretability. Whilst it saves time, it may greatly increase the complexity, resulting in a major decrease of trust in the system. For instance, disabling the automatic clustering really helped the users to better understand the difference between the DR and the visual clustering. In the words of the users: the DR is responsible for moving the crimes while the clustering changes the colors. Additionally, it helps the user to spatially anchoring individual crime reports and clusters across different projections.

Sliders with sensible limits and direct visual feedback and are a great way to encourage the user to trying different settings. In early versions, the users were reluctant to try out different alternatives for weights, projections and clusterings. We also provided them with a cluster interpretation strategy. Starting with a few clusters, allows the users to interpret and understand the major areas and features within the projections, that can be iteratively refined. We implemented this exploration strategy with the clustering slider.

Like many Visual Analytics (VA) tools, the scalability of our system is limited. Our domain experts suggested a typical targeted analysis task (e.g., looking at crimes happened in last three months in a specific region) involves no more than 500 crimes. For our use cases the tool worked reasonably well on 2000 crimes with 300 features. However, calculating the distances and sorting is bounded by computational complexity. PCA and MDS work the fastest. T-SNE is the slowest. Calculating all clustering parameterizations does not take too long as only 2 dimensions are covered. In any case, the user sees visual cues hinting that a computation is being performed. Although the users state that waiting for a certain computation is acceptable, we do not freeze nor disable any component, but allow the user to continue the exploration.

The Sequential Pattern Mining (SPM) processing step, after the feature extraction, offered valuable advan-

tages. The number of features is reduced as rare features are pruned and the relevance of a feature is related to the overall size of the current dataset being analyzed. This means that in smaller and more specific datasets the features are likely to contain more terms and are therefore more fine-grained. The use of patterns (sequences of terms) allows crime reports to be better differentiated. Additionally, this step allowed us to preconfigure default weights based on the length of a sequence. Here, we assume that a longer sequence is more valuable to the user, but typically describes less crime reports (due to the antimonotonicity of patterns [5]).

We incorporated many changes in the UI to remove or hide complexity. As in many projects, it took time to find a common level of expert language to efficiently discuss features and issues. We found it beneficial to attend workshops where experts explained their daily work and detailed their analytical tasks, including the problems they typically face. Frequent software updates maintains a welcome degree of familiarity with the system, helping the experts keep abreast of the changes and reduces the amount of the training. This also increases the motivation of the experts, encouraging them to be more open in stating issues in using the tool, which eventually increases the productivity.

In contrast to the work of Johansson et al. [23], the Concept Explorer is specifically designed for expert users. However, many parallels in the lessons learned are visible. Users are easily overwhelmed by complexity which can be partially compensated by frequent user training but removing or hiding complexity yields a more sustainable effect. Our end users also expressed the wish for more guidance but in the same breath, they wanted to keep the freedom and transparency of the current system (which is required in the criminal intelligence analysis domain). We experienced positive feedback during paired-analytics sessions as guidance from the developers combined with the expert knowledge has a positive synergy. Overall, we observe a tradeoff between guidance and the transparency across different algorithmic alternatives.

6 Conclusions and Future Perspectives

We presented our research in designing an interactive Comparative Case Analysis (CCA) system in collaboration with domain experts and detail on how components and the overall system has changed over time in a design study. The current system provides a powerful tool using a hybrid approach to simultaneously analyze and explore the data and an automatically generated feature space. Dimensionality Reduction (DR) techniques

are utilized in a similar fashion to visualize the similarity spaces. The hybrid view aids the users in drawing conclusions on the effects of features in the data space. The tight coupling of multiple components allows access to the data from different perspectives. Our DR pipeline implementation supports a variety of interactions but we observed and learnt that analysts may be overwhelmed by an excessive number of visual alternatives and configuration options. To tackle this problem we allow the users to interpret the results and interact directly with them in the crime table (the tool that they are familiar with). This helped them to understand and importantly, build trust in the computations. Our visual interaction design is generalizable to other data types and applications. To this end, we now include additional structured metadata, such as the weekday or known offender properties (e.g., gender, age) in our analysis.

In future work, we aim to enrich the table interactions with semantic mappings to DR pipeline adaption (inspired by Endert et al.'s work on semantic interaction [15, 16, 14]). For example, we want to allow the Tactical Analyst (TA) to re-arrange columns or rows to derive feature weights. An automatic sorting of the cluster-rows, for example, based on the td-idf measure could support the analyst in the CCA task. Similarly, we want to automatically derive which DR type is closest to the analyst's feedback (e.g., when the analyst declares two clusters as similar).

Based on the feedback of the experts we will also improve the WOC by hiding unchanged features in the line chart and showing them only on demand. Furthermore, a filter list can be used to selectively compare different histories of feature sets. Labels, as well as the values in numbers will be added to allow simplify the generation of reports. Annotations in the WOC can enable the analyst to explain and justify their decision making. Eventually, this may enable us to generate templated reports automatically.

The VALCRI project is in its final phase that will focus on the end-user training and evaluations. Our plan is to quantitatively measure which interactions are used, to capture the analysis processes of different analysts, and to collect more qualitative feedback.

One important aspect of the Concept Explorer as a tool for TAs is to manually add new concepts. Additionally, users reported that many concepts are not useful for a specific case, so a preconfiguration step selecting and adding features will be beneficial.

We provide insight into the development of a rather complex tool and show how we gradually decrease and hide the complexity from the user during the development in order to make it "less scary".

References

1. Apache opennlp. https://opennlp.apache.org/. Accessed: 2017-09-13.
2. T-sne-java. https://github.com/lejon/T-SNE-Java. Accessed: 2017-12-04.
3. T-sne-javascript. https://github.com/karpathy/tsnejs. Accessed: 2017-12-04.
4. Visual analytics for sense-making and criminal intelligence analysis, http://www.valcri.org/, last retrieved 14th sep.,2017.
5. R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering, March 6-10, 1995, Taipei, Taiwan*, pages 3–14, 1995.
6. C. F. Baker, C. J. Fillmore, and J. B. Lowe. The berkeley framenet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL '98, August 10-14, 1998, Université de Montréal, Montréal, Quebec, Canada. Proceedings of the Conference.*, pages 86–90, 1998.
7. C. Bennell and D. V. Canter. Linking commercial burglaries by modus operandi: tests using regression and ROC analysis. *Science & Justice*, 42(3), 2002.
8. L. Bradel, C. North, L. House, and S. Leman. Multi-model semantic interaction for text analytics. *IEEE Conf. on Visual Analytics in Science and Technology (VAST)*, pages 163–172, 2014.
9. U. Brandes and C. Pich. Eigensolver methods for progressive multidimensional scaling of large data. In M. Kaufmann and D. Wagner, editors, *Graph Drawing, 14th International Symposium, GD 2006, Karlsruhe, Germany, September 18-20, 2006. Revised Papers*, volume 4372 of *Lecture Notes in Computer Science*, pages 42–53. Springer, 2006.
10. D. V. Canter, L. J. Alison, E. Alison, and N. Wentink. The organized/disorganized typology of serial murder: Myth or model? *Psychology, Public Policy, and Law*, 10(3):293–320, September 2004.
11. D. Collier. The comparative method. In *POLITICAL SCIENCE: THE STATE OF DISCIPLINE II*, pages 105–118, 1993.
12. N. Cope. Intelligence led policing or policing led intelligence?: Integrating volume crime analysis into policing. *Br. J. Criminol.*, 44:188–203, 2004.
13. Ç. Demiralp. Clustrophile: A tool for visual clustering analysis. 2016.
14. A. Endert. *Semantic Interaction for Visual Analytics: Inferring Analytical Reasoning for Model Steering.* Synthesis Lectures on Visualization. Morgan & Claypool Publishers, 2016.
15. A. Endert, P. Fiaux, and C. North. Semantic interaction for sensemaking: Inferring analytical reasoning for model steering. *IEEE Trans. on Visualization and Computer Graphics*, 18(12):2879–2888, 2012.
16. A. Endert, P. Fiaux, and C. North. Semantic interaction for visual text analytics. In *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)*, pages 473–482, 2012.
17. E. Frank, M. A. Hall, and I. H. Witten. The WEKA workbench. In M. Kaufmann, editor, *Data Mining: Practical Machine Learning Tools and Techniques.* Springer, 2016.
18. M. Gleicher. Explainers: Expert explorations with crafted projections. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2042–2051, 2013.
19. A. Gomariz, M. Campos, R. Marín, and B. Goethals. Clasp: An efficient algorithm for mining frequent closed sequences. In *Advances in Knowledge Discovery and Data Mining, 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part I*, pages 50–61, 2013.
20. IBM. Ibm i2 intelligence analysis platform.
21. D. Jäckle, F. Stoffel, S. Mittelstädt, D. A. Keim, and H. Reiterer. Interpretation of dimensionally-reduced crime data: A study with untrained domain experts. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017) - Volume 3: IVAPP, Porto, Portugal, February 27 - March 1, 2017.*, pages 164–175, 2017.
22. W. Jentner, G. Ellis, F. Stoffel, D. Sacha, and D. A. Keim. A Visual Analytics Approach for Crime Signature Generation and Exploration. In *The Event Event: Temporal & Sequential Event Analysis, IEEE VIS 2016 Workshop*, 2016.
23. J. Johansson, T. Opach, E. Glaas, T. S. Neset, C. Navarra, B. Linnér, and J. K. Rød. Visadapt: A visualization tool to support climate change adaptation. *IEEE Computer Graphics and Applications*, 37(2):54–65, 2017.
24. D. Jurafsky and J. H. Martin. *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2nd Edition.* Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International, 2009.
25. C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval.* Cambridge University Press, 2008.
26. C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60, 2014.
27. G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.
28. NPIA. National policing improvement agency: Professional practice on analysis, 2008.
29. J. Prowse and E. Bennett. *Working Manual of Criminal Law.* Carswell Legal Pubns, Mar 2000.
30. T. Ruppert, M. Staab, A. Bannach, H. Lücke-Tieke, J. Bernard, A. Kuijper, and J. Kohlhammer. Visual interactive creation and validation of text clustering workflows to explore document collections. *Electronic Imaging*, 2017(1):46–57, 2017.
31. D. Sacha, W. Jentner, L. Zhang, F. Stoffel, and G. Ellis. Visual Comparative Case Analytics. In M. Sedlmair and C. Tominski, editors, *EuroVis Workshop on Visual Analytics (EuroVA).* The Eurographics Association, 2017.
32. D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. What you see is what you can change: Human-centered machine learning by interactive visualization. *Neurocomputing*, 268:164–175, 2017.
33. D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):241–250, 2017.

34. H. Saneifar, S. Bringay, A. Laurent, and M. Teisseire. S2MP: similarity measure for sequential patterns. In *Data Mining and Analytics 2008, Proceedings of the Seventh Australasian Data Mining Conference (AusDM 2008). Glenelg/Adelaide, SA, Australia, 27-28 November 2008, Proceedings*, pages 95–104, 2008.

35. M. Sedlmair, M. D. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Trans. on Visualization and Computer Graphics*, 18(12):2431–2440, 2012.

36. J. T. Stasko, C. Görg, and Z. Liu. Jigsaw: supporting investigative analysis through interactive visualization. *Information Visualization*, 7(2):118–132, 2008.

37. C. Turkay, P. Filzmoser, and H. Hauser. Brushing dimensions - A dual visual analysis model for high-dimensional data. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2591–2599, 2011.

38. P. van der Corput and J. J. van Wijk. Exploring items and features with i^f, $\dot{f}^i$-tables. *Computer Graphics Forum*, 35(3):31–40, 2016.

39. J. Wenskovitch, I. Crandell, N. Ramakrishnan, L. House, and C. North. Towards a systematic combination of dimension reduction and clustering in visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 2017.

40. J. A. Wise. The ecological approach to text visualization. *JASIS*, 50(13):1224–1233, 1999.

41. B. L. W. Wong, C. Rooney, and N. Kodagoda. White paper: Analyst user interface: Thinking landscape as design concept. Technical report, Middlesex University London, January 2017.

42. K. Xu, S. Attfield, T. J. Jankun-Kelly, A. Wheat, P. H. Nguyen, and N. Selvaraj. Analytic provenance for sensemaking: A research agenda. *IEEE Computer Graphics and Applications*, 35(3):56–64, 2015.

43. X. Yan, J. Han, and R. Afshar. Clospan: Mining closed sequential patterns in large datasets. In *Proceedings of the Third SIAM International Conference on Data Mining, San Francisco, CA, USA, May 1-3, 2003*, pages 166–177, 2003.

44. X. Yuan, D. Ren, Z. Wang, and C. Guo. Dimension projection matrix/tree: Interactive subspace visual exploration and analysis of high dimensional data. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2625–2633, 2013.

45. L. Zhang, C. Rooney, L. Nachmanson, B. L. W. Wong, B. C. Kwon, F. Stoffel, M. Hund, N. Qazi, U. Singh, and D. A. Keim. Spherical similarity explorer for comparative case analysis. In *Visualization and Data Analysis 2016, San Francisco, California, USA, February 14-18, 2016*, pages 1–10, 2016.