J Intell Inf Syst
DOI 10.1007/s10844-014-0304-9

The human is the loop: new directions for visual analytics

Alex Endert · M. Shahriar Hossain · Naren Ramakrishnan · Chris North · Patrick Fiaux · Christopher Andrews

Received: 10 October 2012 / Accepted: 2 January 2014

Abstract Visual analytics is the science of marrying interactive visualizations and analytic algorithms to support exploratory knowledge discovery in large datasets. We argue for a shift from a 'human in the loop' philosophy for visual analytics to a 'human is the loop' viewpoint, where the focus is on recognizing analysts' work processes, and seamlessly fitting analytics into that existing interactive process. We survey a range of projects that provide visual analytic support contextually in the sensemaking loop, and outline a research agenda along with future challenges.

Keywords Visual analytics · Clustering · Spatialization · Semantic interaction · Storytelling

A. Endert
Pacific Northwest National Laboratory, Richland, WA 99352, USA
e-mail: alex.endert@pnnl.gov

M. S. Hossain
Department of Computer Science, University of Texas at El Paso,
El Paso, TX 79968, USA
e-mail: mhossain@utep.edu

N. Ramakrishnan (✉) · C. North · P. Fiaux
Department of Computer Science, Virginia Tech, Blacksburg, VA 24060, USA
e-mail: naren@cs.vt.edu

C. North
e-mail: north@cs.vt.edu

P. Fiaux
e-mail: pfiaux@vt.edu

C. Andrews
Department of Computer Science, Mount Holyoke College,
South Hadley, MA 01075, USA
e-mail: andrews@mtholyoke.edu

Published online: 28 January 2014

1 Introduction

The emerging field of *visual analytics* seeks to address the needs of exploratory discovery in big data (Kielman et al. 2009; Thomas and Cook 2005). The approach is to marry the big data processing capabilities of analytics with the human intuitive capabilities of interactive visualization. The rationale is that data is too large for purely visual methods, requiring the use of data processing and mining; yet, the desired tasks are too exploratory for purely analytical methods, requiring the involvement of human analysts, using visualization as a medium for human interaction with the data. This approach must be situated within an understanding of human cognitive reasoning processes. Thus, visual analytics research necessitates an interdisciplinary approach.

Targeted tasks in visual analytics are those that are exploratory in nature, where the questions are ill-defined or unknown *a priori* and training data is not available. Tasks are strategic in nature, and must be translated into operational questions during the course of the analysis. For example, in intelligence or business analysis, analysts may be confronted with large amounts of textual information that they must make sense of. Stasko points out that while text analytics and visualizations are helpful in structuring the information, eventually the analyst must "read and understand the actual text documents" to gain semantic insight and report a finding (Stasko et al. 2008). Cybersecurity analysts must defend networks against attack or misuse. While known attack methods may be easily detectable by pattern analysis, creative new attacks are continually being developed by innovative adversaries. The analysts goal here is to seek, identify, track, understand, prevent, and document, such attacks (Fink et al. 2009).

To date, exemplar research in visual analytics has varied in its emphasis on the visual or the analytics, and the degree of interaction. Simoff et al. (2008) discuss the challenge of transitioning from interaction between computational analytic runs, to interaction during analytic runs. Keim et al. (2010) describe visual analytics as a problem solving process following the mantra: 'analyze first; show the important; zoom, filter, and analyze further; details on demand.' For example, Jigsaw (2008) supports visual analytics of text collections by first conducting entity extraction and link analysis, and then enabling users to explore the results in a variety of visual representations. Van Wijk et al. (1999) demonstrate the use of iterative model testing and refinement by experts to develop a final visual representation that communicates a valuable insight. InSpire (2012) and StreamIt (2011) exploit complex topic modeling to visualize document collections, and users can make parameter adjustments (e.g., by changing keyword weights) to compute entirely new views of the collection. iPCA (2009) users can navigate a principal component analysis model with sliders for adjusting model parameters, thus manipulating the role of eigenvalues and eigenvectors in data reduction.

Interaction is thus the critical glue that integrates analytics, visualization, and human analyst. But how should this interaction be designed? A common phrase used to describe interactive analytics is 'human in the loop,' representing the need for analytic algorithms to occasionally consult human experts for feedback and course correction. However, we believe human-in-the-loop thinking leads to inevitable usability problems, as analysts are presented with results out of context, without understanding their meaning or relevance, and interactive controls are algorithm specific and difficult to understand. In place of the flood of data, analysts are confronted with navigating a flood of disconnected algorithms and their parameters/settings.

Our hypothesis is that we must move beyond human-in-the-loop to 'human **is** the loop' analytics. The focus here is on recognizing analysts' work processes, and seamlessly fitting analytics into that existing interactive process. For example, Pirolli and Card's model of the sensemaking loop for analysts (Pirolli and Card 2005) (see Fig. 1) describes the complex interactive process that analysts conduct. The two major sub-loops involve foraging for relevant information and synthesis of hypotheses. The dual search loop involves the cognitively challenging process of generating hypotheses from found evidence, and simultaneously searching for evidence that supports potential hypotheses, while managing the potential effects of cognitive bias (Heuer 1999). This philosophy means that algorithms must be redesigned from the ground up to fit into this model, learning from the interactions that analysts are already performing in their sensemaking process and displaying results naturally within the context of that process. In this article, we present several examples of this approach to visual analytics and a research agenda to realize it.

2 Interaction in visual analytics

To emphasize the relevance of interaction, and to illustrate through examples the 'human is the loop' philosophy, we survey four projects from our group. The projects can be variously classified (see Table 1) in terms of the problem domain they study and in terms of the granularity of interaction.

The two broad analysis tasks we consider are related to clustering and storytelling. Clustering (Jain et al. 1999) needs almost no introduction to this audience. As a classical technique for data analysis it has become increasingly repurposed for new uses, with the advent of novel applications in bioinformatics (Sese et al. 2004; Ernst et al. 2005; Xu et al. 2002; Monti et al. 2003), intelligence analysis (Petrushin 2005; Liang et al. 2003; Baron and Freedman 2008), and web modeling (Miao et al. 2009; Aghabozorgi and Wah 2009; Cadez et al. 2003). Clustering is closely related to spatialization and dimension reduction, where the goal is to ensure that a dataset is laid out spatially in a way that reflects the user's notions of dissimilarity or distance.

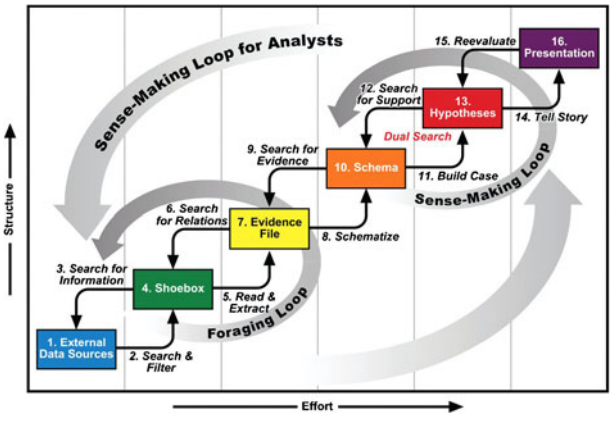


Fig. 1 The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. From Pirolli and Card (2005)

Table 1 Four projects that straddle multiple granularities of 'human is the loop' interaction

Project	Type of user interaction	Analysis task	User input	Visual feedback
ForceSPIRE	Instance-level interaction (instances = data points)	Spatializing	Implicit (semantic interactions)	Updated spatialization
Scatter-Gather	Bundle-level interaction (bundles = clusters)	Clustering	Explicit (scatter gather constraints)	Updated clustering
Analyst's Workspace	Instance-level interaction (instances = documents)	Storytelling	Explicit (path constraints)	Updated stories
Bixplorer	Bundle-level interaction (bundles = biclusters)	Storytelling	Explicit (entity constraints)	Compositional patterns

Of recent interest has been the ability to impart prior domain knowledge to data mining algorithms in the form of constraints (Wang and Davidson 2010; Davidson et al. 2007; Wagstaff et al. 2001; Davidson and Ravi 2005), clustering nonhomogeneous datasets (Hossain et al. 2010; Momtazpour et al. 2012), or providing expressive forms of user input (Alonso and Talbot 2008; Hwang et al. 2011; Huang and Mitchell 2006). In the below sections we are motivated by how users can steer the iterative process by which users can inspect clustering or spatialization outcomes, and how the system can provide feedback using visual analytic means. In particular, our desire was to provide natural interfaces for users by which they can critique results and, at the same time, operationalize their input into an effective mechanism to recluster the results. Our thesis is that 'a little domain knowledge goes a long way', and enabling the user in the loop to supply input can be significantly more effective than trying to design a clever clustering algorithm.

The second analysis task we study involves storytelling (Kumar et al. 2006; 2008), the investigative process of 'connecting the dots' between seemingly disconnected information (Hossain et al. 2011, 2012a, b, c; Wu et al. 2012). Storytelling is an accepted metaphor in analytical reasoning and in visual analytics (Thomas and Cook 2005). (By storytelling, we do not mean creative writing activities, e.g., composing a novel, or designing an animated movie (Kelleher and Pausch 2007), but rather the task of connection building between desired end-points.) Different researchers have employed this metaphor in different contexts. For instance, it has been used to denote generating event timelines (Guha et al. 2005), filling in the gaps in chains of evidence, threading information across dialogs, tracking collective reasoning patterns across a corpus (Rzhetsky et al. 2006), information organization based on narrative structures (Kuchinsky et al. 2002), topic tracking, and, in general, deciphering genealogy from a collage of information (Shaparenko and Joachims 2007). The common theme to all of them is their ability to present spatial/temporal/spatio-temporal progressions of multifaceted information. Many software tools exist to support story building activities (Eccles et al. 2008; Hsieh and Shipman 2002; Wright et al. 2006; i2group). Analysts are able to lay out evidence according to spatial cues and incrementally build connections between them. Such connections can then be chained together to create stories which either serve as end hypotheses or as templates of reasoning that can then be prototyped. However, sophisticated analytic support for storytelling remains a significant research frontier. We describe how we have demonstrated visual analytic approaches for exploring connections in document collections and for building stories between possibly disparate end points.

The other distinction being made in Table 1 refers to whether user input and control of the knowledge discovery process occurs at the level of instances (i.e., the original data points) or at the level of a higher-level abstraction as a result of some grouping/bundling process. Two examples of such grouping could be clusters or biclusters (Madeira and Oliveira 2004), described in greater detail below. Both forms of interaction are relevant for different applications. Finally, as Table 1 shows, it is helpful to view all projects through a common lens, viz. the type of user input they accommodate and how the visual feedback is presented back to the user in the context of their analytic process.

2.1 ForceSPIRE

ForceSPIRE (Endert et al. 2012a) is a visual analytics system (Fig. 2) to generate meaningful spatializations from text data, i.e., laying out documents visually such that the layout reflects user notions of similarity and distance. ForceSPIRE supports visual data exploration through interpreting the user interaction and performing implicit model steering operations (Endert et al. 2012b). Such user interactions are tailored towards the domain expertise and

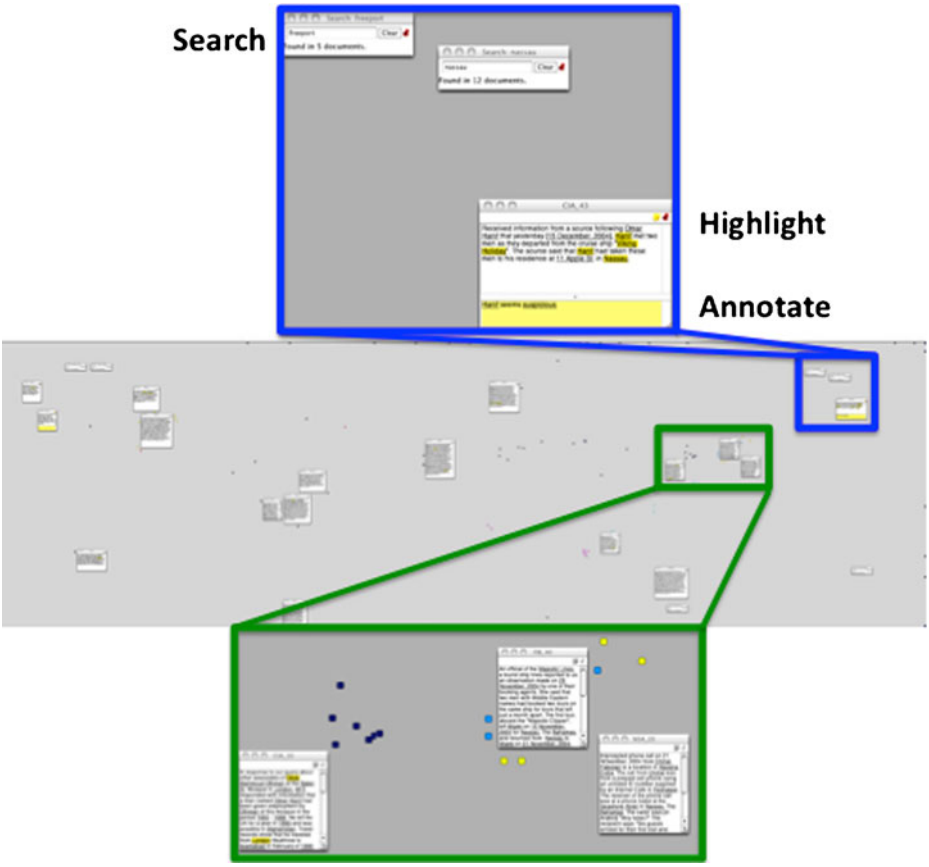


Fig. 2 ForceSPIRE can automatically generate spatializations from text that respect user's interactions

tasks of users, while providing implicit computational support are called *semantic interactions*. As a result, these semantic interactions such as repositioning documents, highlighting phrases as they read a document, annotations, and search results train the underlying dimension reduction model towards understanding the features important to the user. For example, by the user moving two documents closer together in the spatialization, ForceSPIRE can determine the characteristics responsible for the similarity through metric learning techniques. The resulting computation incrementally adjusts the spatialization in accordance with this user input.

It is instructive to contrast ForceSPIRE with approaches that require explicit user input for model steering. For example, tools such as IN-SPIRE (PNNL 2012) enable model steering through users explicitly selecting features (keywords) through a menu. While more expressive, such explicit user input does not provide the flexibility that may be desired for visual data exploration.

Implicit user feedback entails providing the feedback of the model through the visualization, rather than explicitly via the weighted dimensions. For example, ForceSPIRE provides an updated spatialization as a result of a semantic interaction (an entity viewer window also exists, where dimension weighting can be individually viewed and adjusted). Similarly, previous work on observation-level interaction also uses the updated spatialization as a medium for communicating the learned domain knowledge (Endert et al. 2011).

Liu et al. (2011) describe how after performing similar observation-level interaction, the system can provide explicit feedback to the user regarding the model learning. Through performing this type of observation-level interaction, the users are given a set of weights that correspond to their newly generated spatialization. As such, this work focuses on explicitly showing the user the dimensions that were adjusted based on their interaction (i.e., the feedback from the system to the users).

However, how can a system support a mixture between these two forms of feedback? One can see that as the number of dimensions increase (and become more abstract), explicit feedback may not be effective or meaningful to the users. Further, the results of a user study of ForceSPIRE (where explicit feedback can be obtained by the entity viewer window) shows that users may not prefer, or need, this form of feedback (Endert et al. 2012a). Similarly, users may require some feedback to gauge what information the system is learning based on their interaction, and given the ability to provide more fine-grained model steering (e.g., steering at the entity-level, rather than at the document level).

One possibility is to maintain this feedback within the spatialization. That is, instead of providing a separate view for the explicit feedback, augmenting the spatialization to include this sort of information may be beneficial. For example, ForceSPIRE includes entity underlining within the text of a document to inform users of which keywords are entities in the model. However, this depth of information could be increased, to highlighting words on a color ramp based on their weight. Then, if users find inconsistencies in the entity weighting scheme, adjustments can be made, and the bi-directional learning can continue.

To enable the implicit model learning of ForceSPIRE and semantic interaction, an inversion of the mathematical projection model is used. The decision of inverting a mathematical projection model may be a good fit for systems where the semantic interactions are primarily observation-level interactions. However, other forms of semantic interactions may not lend themselves to directly inverting a projection model (e.g., highlighting text, performing a search, etc.). One possibility for these forms of interaction is to create a forward model for each of these, by which the inversion can take place. For example highlighting can be automated given the weights of entities. Then, as users manually highlight (or change the highlighting that the system recommended), the system can invert the model

used for highlighting to maintain mathematically valid visualizations. The fundamental principles of semantic interaction still apply to these interactions, as they generalize beyond spatializations and observation-level interactions.

The updating of the spatial layout in ForceSPIRE is very important, as it provides the opportunity to show the user what has changed from one layout to the other. That is, it provides the user feedback on what the system has learned from their previous semantic interaction. Models that are incremental in nature (where the calculation of the lowest-stress configuration is incrementally obtained) more easily support this concept, as the user can observe the model achieving the state. For example, users can gain insight into both the characteristics of the model, as well as the weighting vector, through observing a force-directed model settling out. Figure 3 illustrates a series of spatializations showing the progression of the spatialization when using ForceSPIRE. The co-creation of the spatial layout of the dataset through semantic interaction fosters visual data exploration and

Fig. 3 Sample interactions in ForceSPIRE

sensemaking. An important design principle here is that the incremental learning of the model closely mirrors the incremental formalism (Hsieh and Shipman 2002) exhibited by the analyst's sensemaking process, both conceptually and spatially.

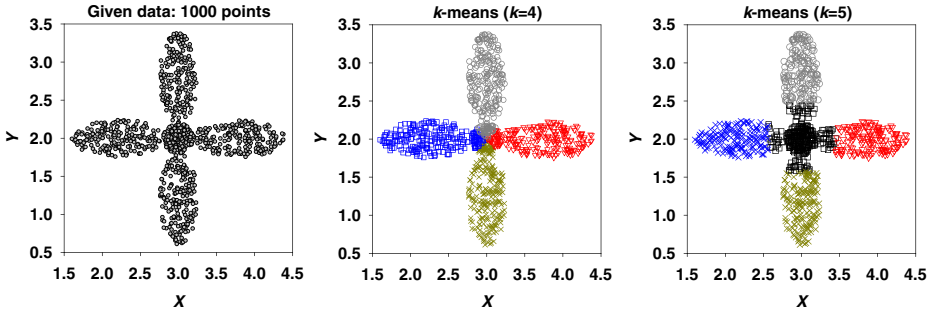
2.2 Scatter-gather

We now turn to our second example of a visual analytic framework for clustering. This framework aims to incorporate user input at the level of clusters, rather than instances.

In our experiences working with diverse application scientists, we have identified an interaction style (Pirolli et al. 1996)—scatter/gather clustering—that helps users iteratively restructure clustering results to meet their expectations. As the names indicate, *scatter* and *gather* are dual primitives that describe whether clusters in a current segmentation should be broken up further or, alternatively, brought back together. By combining scatter and gather operations in a single step (referred to as *scatter-gather clustering*), we support very expressive dynamic restructurings of data.

To illustrate the idea of scatter/gather clustering, we use a synthetic dataset composed of 1000 two-dimensional points (see Fig. 4(a)). The dataset is composed of four petals and a stalk each containing 200 points. When the user applies simple k-means clustering, with a setting of four clusters (i.e., $k = 4$), the flower is divided into four parts as shown in Fig. 4(b) where the petals are indeed in different clusters, but each of the petals also takes up one-fourth of the points from the stalk of the flower. When a setting of five clusters is used, the user obtains the clustering shown in Fig. 4(c). It is evident that the five clusters generated by k-means are not able to cleanly differentiate the stalk from the petals.

A conventional clustering algorithms like k-means does not take user expectation as an input to produce better clustering results. Even constrained clustering algorithms would require an inordinate number of user interactions to clearly separate the stalk from the petals. In the scatter-gather clustering framework, the user can provide an input to the algorithm regarding the expected outcome as shown in Fig. 5. The constraints shown in the middle of the figure should be read both from *left to right* and from *right to left*. Reading from left to right, we see that the user expects the four clusters to be broken down (scattered) into five clusters. Reading from right to left, we see that the stalk is expected to gather points from all current clusters, but there is a one-to-one correspondence between the desired petals to the original petals. Figure 5 shows that the results of such a scatter/gather clustering provide



(a) Original data (b) k-means (k=4) (c) k-means (k=5)

Fig. 4 Clustering the flower dataset. (**a**) The dataset has 1000 2D points arranged in the form of a flower. (**b**) Result of k-means clustering with $k = 4$. (**c**) k-means clustering with $k = 5$. Points from the stalk spill over into the petals

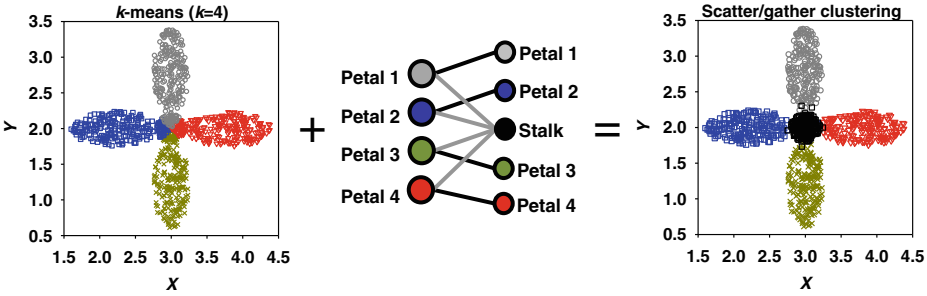


Fig. 5 Clustering the flower dataset with user provided input: Scatter/gather constraints when imposed over a clustering with four clusters yields five clusters with well-separated petals and center with the stalk, unlike Fig. 4(c)

well-separated petals and stalk, unlike the result provided by simple k-means with k=5 (as shown in Fig. 4(c)). Thus, instead of being frustrated by choosing a seemingly arbitrary parameter value for k, the analyst directly manipulates the cluster reorganization scheme. The interaction fits the analyst's cognitive process of incrementally redistributing specific clusters to test hypotheses.

The way in which constraints from Fig. 5 are incorporated to revise a clustering is covered in detail in (Hossain et al. 2012c). Essentially, we prepare a contingency table relating the current clustering to the target clustering, and use a non-linear optimization framework to propagate the given mean prototypes through the contingency table, to identify prototypes for the target clustering.

Figure 6 illustrates the use of scatter-gather clustering by an analyst studying the bat biosonar system. The expert is trying to find partitions of a woolly horseshoe bat ear. The expert at first partitions the object into two clusters using k-means clustering (Fig. 6(a)). The expert finds the partitions interesting. He observes that the boundary and the vertical ridges are in the same cluster (green), and the rest of the ear is in another cluster. This fosters a thought in the expert's mind that the vertical ridges could be separated to form a new cluster. The expert also believes that there could be less prominent layers in the borders of the ear. Being unsure about the constraints, the expert provides a uniform scatter/gather constraint table of size 2×3 indicating that he desires three clusters out of the two clusters.

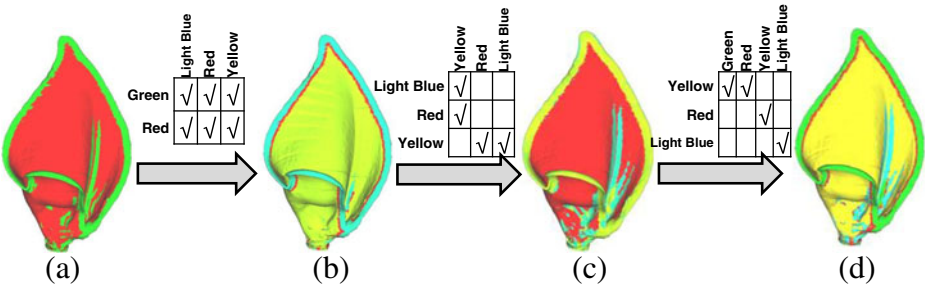


Fig. 6 An example of interactive scatter/gather clustering of a woolly horseshoe bat ear. The expert partitions the ear into four clusters beginning from a setting of two clusters. *(a) to (b)*—The expert supplies a 2×3 constraint table to generate three clusters from two, and the vertical ridge is lost in the result; *(b) to (c)*—the expert supplies constraints in a 3×3 table to retrieve the vertical ridge; *(c) to (d)*—the expert provides constraints in a 3×4 matrix to scatter the border into two layers but to keep the rest of the clusters the same

Our scatter/gather clustering provides the result shown in Fig. 6(b). The partitioning of Fig. 6(b) was able to pick up two border layers, but the vertical ridges now diminish inside the surrounding cluster. At this point, the expert believes that it is more important to reveal the shape of the vertical ridges rather than discovering the layers in the boundary. The expert now provides an S/G constraint table to merge two boundaries (light blue and red), and split the mid region of the ear (yellow) into two clusters. The resulting clusters are shown in Fig. 6(c) where the vertical ridges are well separated in one cluster. The expert now desires to split the border into two layers that he previously merged. Setting up an S/G constraint table of size 3×4 as shown in the middle of (c) and (d) objects of Fig. 6, the user obtains four clusters. These four clusters contain two layers of border (green and red), vertical ridges (light blue), and the flat region of the ear (yellow).

Unlike the way user interaction is used in ForceSPIRE the reader should note that user input is given here not at the instance level (i.e., specific data points) but at the cluster level, viz. which clusters should be broken up or brought back together. Thus, scatter-gather provides a fundamentally different type of interaction paradigm for visual analytics that fits into the analyst's process of redistributing clusters.

2.3 Analyst's workspace

We now shift our attention to navigating and mining large document collections. Analyst's Workspace (AW) is a visual analytics environment that i) closely mimics information organization layouts employed by analysts, ii) relates multiple representations to accommodate different strategies of exploration, and iii) provide automated algorithmic assistance for foraging connections and hypothesis generation. It is primarily targeted at datasets such as the VAST (Symposium on Visual Analytics Science and Technology) 2011 Challenge dataset (*Mini Challenge 3: Investigation into Terrorist Activity*). This dataset contains 4,474 documents, which are primarily synthetic news stories from a fictitious city newspaper, and the goal is to uncover the nature of a threat embedded in the document collection. Most of this collection is actually noise, with only about thirteen of the documents being relevant to uncovering the plot. Another feature of this dataset is that even if the analyst uncovers all thirteen documents, some analysis is still required to actually determine the actual form of the underlying threat.

AW provides the user with a plethora of interaction tools for use with large screen displays (e.g., familiar click-and-drag selection rectangles, multi-click selections) as well as information organization facilities (e.g., graph layout, temporal ordering). Because these operations are local, they only affect the local area or the currently selected documents and hence enable the analyst to freely mix spatial metaphors (see Fig. 7).

While the primary visual elements in AW are full text documents, we also provide support at the entity level. Documents are marked up based on extracted entities, and the analyst can use context menus to quickly identify new entities and create aliases between entities (Fig. 8). Double clicking an entity of interest in a document opens an entity object, which is initially displayed as a list of documents in which that entity appears. Entities can also be collapsed down to a representational icon (Fig. 9), and AW automatically draws links between entities when they co-occur in a document. These two features allow the analyst to rapidly construct and explore social networks, which are commonly used tools in intelligence analysis.

AW also provides basic facilities for text-based search. Search results are displayed as lists of matching documents in the space, like the entities. The documents are color coded to tell the analyst the state of a document: open, previously viewed, or never viewed.

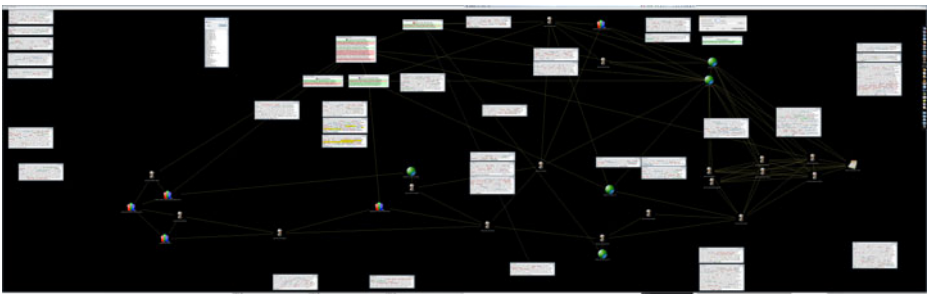


Fig. 7 An active session in Analyst's Workspace. Full text documents and entities share the space, with a mixture of spatial metaphors, such as clusters, graphs and timelines all in evidence. The *yellow lines* are the links of the derived social network

Visual links play a strong role in AW. These allow a number of relationships to be expressed, freeing spatial proximity to be used to express more complex relationships more directly related to making sense of the dataset.

While Analyst's Workspace is designed to be support a flexible approach to sensemaking, it does encourage a particular analytic approach that we observed being used by the analysts. This is a strategy that Kang et al. referred to as "Find a Clue, Follow the Trail" (Kang et al. 2009). In this strategy, the analyst identifies some starting place and then branches out the investigation from that point, following keywords and entities.

In AW, a starting point can be provided by the entity browser Fig. 10, which allows the analyst to order entities by the number of occurrences in the dataset. The analyst opens this entity and gets a list of documents in which this entity appears. The analyst then works through these documents, opening new entities or performing searches as new clues are found. Since all of the search results are independent objects in the space and there is a visual record of which documents have been visited, AW can support both a breadth-first and a depth-first search through the information. As the investigation progresses, the analyst uses

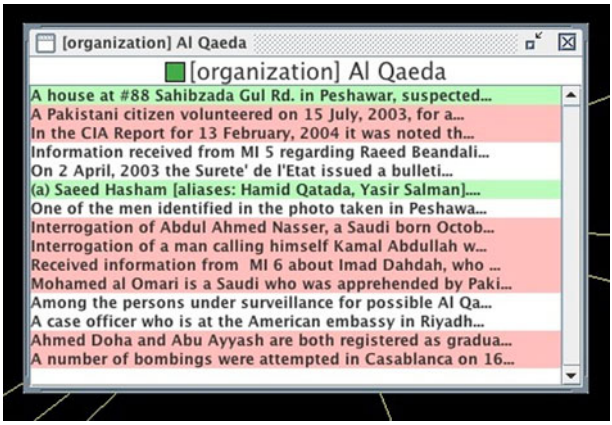


Fig. 8 An 'Al-Qaeda' entity viewed in AW displaying a list of the files in which this entity appears. The *green* files are currently open in the workspace, the *red* have been viewed and rejected by the analyst, and the *white* files have not yet been viewed

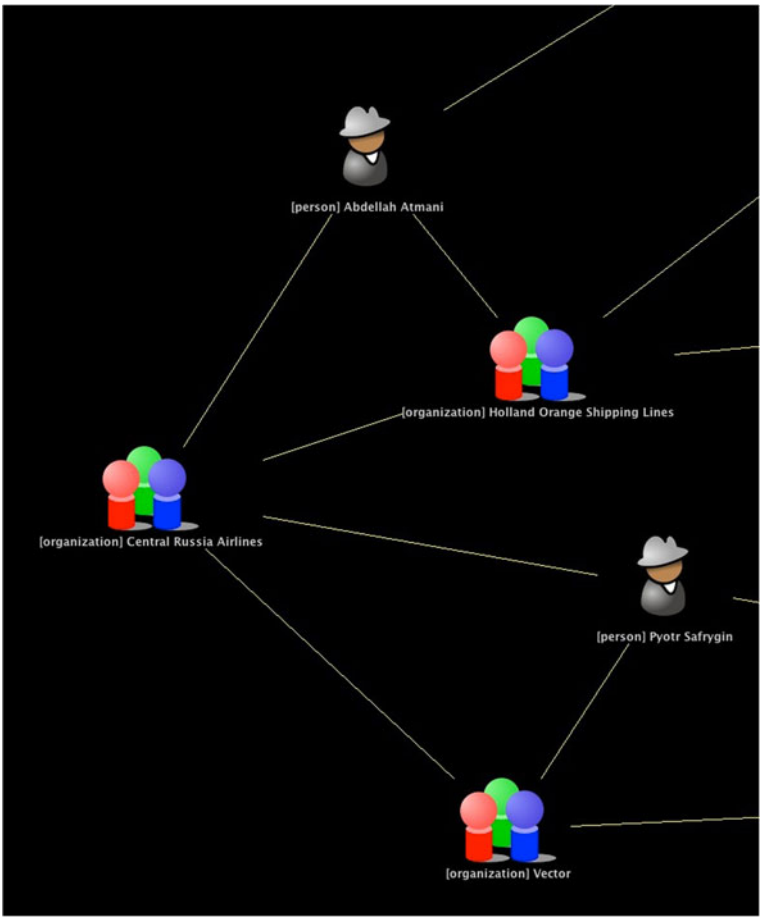


Fig. 9 A section of the generated social network from an AW session. Here, the entities have all been collapsed down to iconified form

the space to arrange the information as it is uncovered, building and rebuilding structures to reflect his or her current understanding of the underlying narrative.

While this approach has been shown to be fairly effective (Kang et al. 2009), it does not permit greater characterization of the dataset and does not support more complex questions that the analyst might ask. For example, this approach relies entirely on the analyst to pick the right keywords and entities to "chase," and can miss less direct lines of investigation. It is common for terrorists to use multiple aliases or code words that can easily thwart this approach. However, it is possible that common patterns of behavior or other document similarities might help the analyst to uncover some of these connections.

AW's story generation framework is exploratory in nature so that, given starting and ending documents of interest, it explores candidate documents for path following, and heuristics to admissibly estimate the potential for paths to lead to a desired destination. The generated paths are then presented to the AW analyst who can choose to revise them or adapt them for his/her purposes.

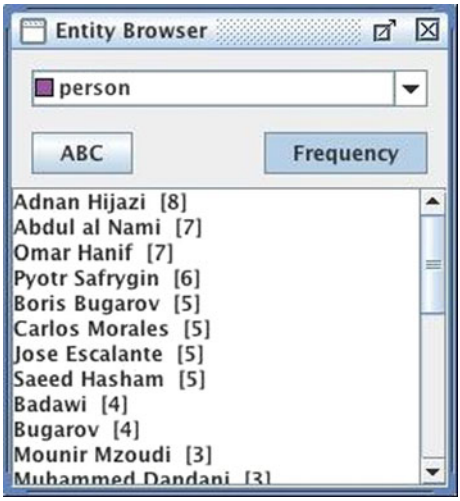


Fig. 10 AW's entity browser, here showing the people identified in the dataset, sorted by the number of documents in which each appears

A story between documents d_1 and d_n is a sequence of intermediate documents d_2, $d_3, \ldots, d_{n-1}$ such that every neighboring pair of documents satisfies some user defined criteria. Given a story connecting a start and an end document, analysts perform one of two tasks: they either aim to strengthen the individual connections, possibly leading to a longer chain, or alternatively they seek to organize evidence around the given connection. The notions of *distance threshold* and *clique size* are used to mimic these behaviors.

The distance threshold refers to the maximum acceptable distance between two neighboring documents in a story. Lower distance thresholds impose stricter requirements and lead to longer paths. The clique size threshold refers to the minimum size of the clique that every pair of neighboring documents must participate in. Thus, greater clique sizes impose greater neighborhood constraints and lead to longer paths. These two parameters hence essentially map the story finding problem to one of uncovering clique paths in the underlying induced similarity network between documents.

Figure 11 describes the steps involved in generating stories for interaction by the AW analyst. For document modeling, a bag-of-words (vector) representation is used where the terms are weighted by tf-idf with cosine normalization. The search framework has three key computational stages:

1. construction of a concept lattice,
2. generating promising candidates for path following, and
3. evaluating candidates for potential to lead to destination.

Of these, the first stage can be viewed as a startup cost that can be amortized over multiple path finding tasks. The second and third stages are organized as part of an A* search algorithm that begins with the starting document, uses the concept lattice to identify candidates satisfying the distance and clique size requirements, and evaluates them heuristically

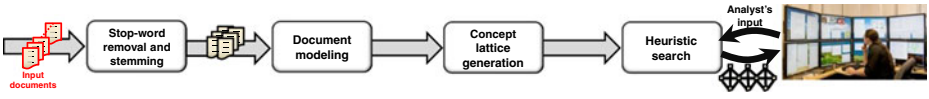


Fig. 11 Pipeline of the storytelling framework in AW

for their promise in leading to the end document. Hossain et al. (2011; 2012a) describe the storytelling algorithms in great details.

The analyst may also need the discovery of paths through the dataset to be more efficient. For example, the analyst may have uncovered that a revolutionary in South America shares the same last name as a farmer in the Pacific Northwest who has been implicated in some nefarious affairs and wishes to ask if there is any link between them other or if their last name is a coincidence. An exhaustive background check of the two men is possible through AW if the dataset is relatively small, but it is an indirect and time consuming process.

Figure 12 shows an example of the usage of AW and our algorithms. In this scenario, the analyst requests a story connecting a pair of interesting documents. The algorithm returns a story but the analyst is not satisfied with parts of the story. The analyst then requests information about documents in the surrounding neighborhood of an intermediate document. Having explored the local neighborhood, the analyst identified two additional documents that form a more meaningful connection and extends the original story. An important design principle here is that the invocation and output of the storytelling algorithms occurs within the analyst's spatial layout, thus fitting naturally into their cognitive sensemaking process. The end points of the story provide spatial anchors for the new information.

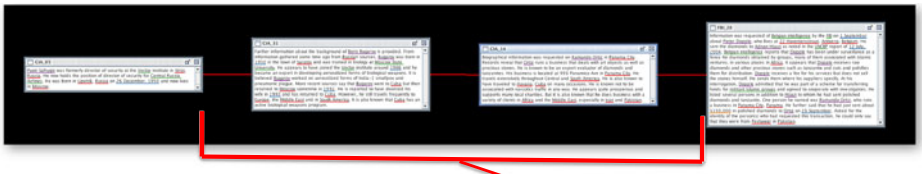
2.4 Bixplorer

Bixplorer is a visual analytics prototype (Fiaux 2012) that supports interactive exploration of textual datasets in a spatial workspace using biclusters. A bicluster, or biclique, is a complete bipartite subgraph in a relation, i.e., where every entity in one set is connected to all entities of another set. Biclusters across entity types serve as an important abstraction by 'bundling' relationships into cohesive units that are key navigation aids as well as units of knowledge discovery in themselves.

Consider Fig. 13 involving a relation capturing attendance of students in specific classes, we might infer a bicluster involving a set of students {S1,S2,S3} all of whom attend the same set of classes {C1,C2,C3,C4}. Biclusters are typically maximal, i.e., additional students and additional classes cannot be added into the bicluster because they will not have a relation to each other (in the original matrix).

Since biclusters are discovered in a single relation, we can 'compose' biclusters discovered separately across two relations by (approximately) matching the biclusters across the common domains. Jin et al. (2008) present this approach to identify compositional patterns in multi-relational datasets. As shown in Fig. 14, biclusters from three different relations can be chained using the common interfaces of people (between the first and second relation) and places (between the second and third relation). The results of such compositions can be read sequentially from one end to the other, not unlike a story. For instance in the scenario from Fig. 14, we might learn about 'a group of faculty from CS and other departments', many of whom 'are planning a trip to Austin, Texas and nearby places', the dates of which are approximately aligned with 'the second week of May 2012'; this might lead us to infer that they are likely HCI researchers planning to attend the CHI'12 conference. Documents supporting these relationships can then be inspected to gather evidence for this hypothesis. Thus, by relating biclusters across multiple relations we can 'bundle' relationships from a diversity of domains in a coherent manner. Such bundling and composition constitute one of the key features of Bixplorer.

Bixplorer is closest in spirit to hybrid matrices and node-link diagrams. NodeTrix, the work of Henry et al., allows exploration of social networks through a hybrid visualization of adjacency matrices (for dense subgraphs) and node-link diagrams (for sparse connections

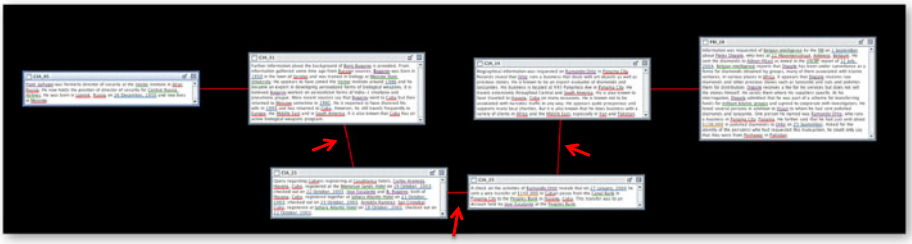


The generated story between the two endpoints. The system has identified two linking documents, and connected them together into a linked story.

The analyst requests a story connecting a pair of interesting documents.

A list of the neighbors of the third document. The lines provide visual links to open documents.

Unsatisfied with the strength of the connection, the analyst requests information about documents in the surrounding neighborhood (i.e., within the local clique).



New connections have been manually added to extend the story

Having explored the local neighborhood, the analyst has identified two additional documents that form a more meaningful connection and extends the original story.

Fig. 12 Illustration of interactively finding a story in AW

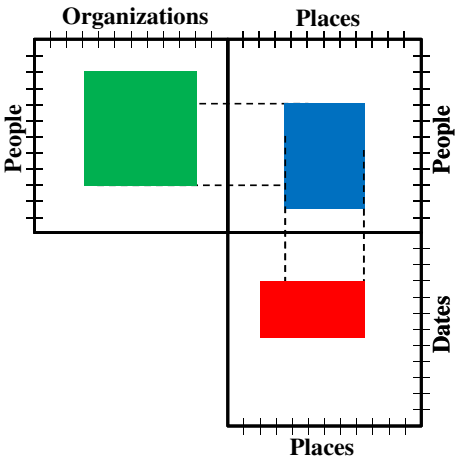
between the subgraphs) (Henry et al. 2007). Through clustering and linking clusters, users can explore relationships of a single type, such as co-authorship between authors. NodeTrix generates initial clusters, and then allows users to group or ungroup nodes to explore how they interact with the layout. OntoTrix by Bach et al. extends this technique to work with ontologies with multiple types of relationships (Bach et al. 2011). Thus allowing clustering and linking nodes of different types within the same graph. Bixplorer is different in that we use biclusters as the key unit of information organization rather than clusters and individual relationships.

Fig. 13 Example bicluster extracted from a student to classes relationship. *Dark cells* represent relationships, *orange cells* represent relationships part of this specific bicluster

Bixplorer uses closed itemset mining algorithms such as CHARM (Zaki and Hsiao 2002) and LCM (Uno et al. 2003); the results of such algorithms are then chained and made available for sensemaking (Fig. 15). Initially, the workspace is empty. Throughout the course of their analysis, users add documents and biclusters into the workspace. The workspace enables users to organize and visualize biclusters and documents together, and the links between them, in a single space. Figure 16 shows Bixplorer on a large, high-resolution display. Previous studies and tools have shown that a spatial workspace such as this enables users to create spatial representations (e.g., clusters, timelines, etc.) to capture their insights about the dataset (Andrews et al. 2010; Shipman and Marshall 1999; Endert et al. 2012). As such, biclusters and documents can be repositioned within the space by the user. A 'Link to...' function from the context menu allows users to create custom links between elements. User-defined links are shown in blue, whereas white links are computationally determined by the data mining.

We conducted a user study of Bixplorer with the Atlantic Storm dataset. Initial text extraction and mining was done offline, resulting in 437 unique entities, 4257 relationships, and 1001 biclusters. We learnt that each of the users was successful in integrating

Fig. 14 Chaining biclusters through multiple relations by approximately matching sets of entities across common domains



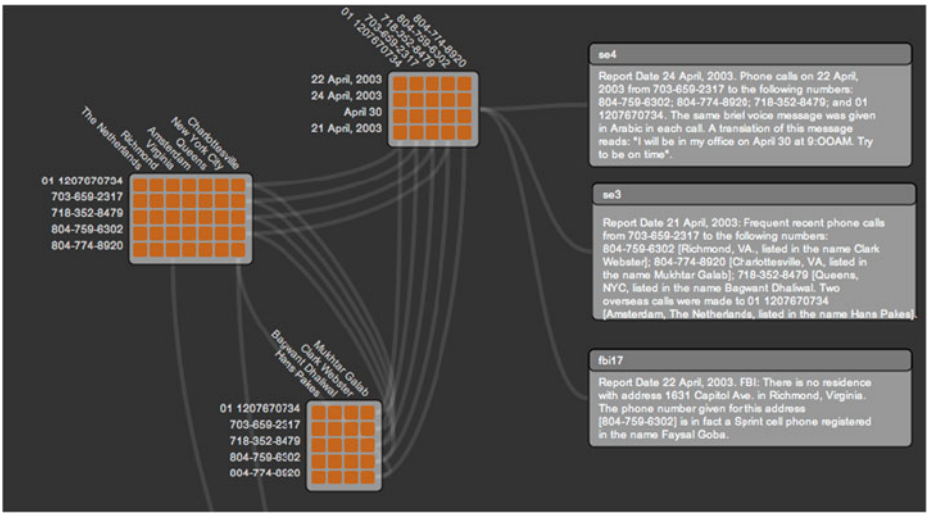


Fig. 15 Sample area of graph workspace with biclusters and documents connected

biclusters into the spatial analysis of the dataset, leveraging the visual representation of relationships in a variety of ways. Although none of the users in this study had previous experience or knowledge of biclusters, each of them was able to quickly integrate biclusters into their process.Biclusters were used to quickly scan relationships, to provide an overview of relationships involving a specific document, and to transition between the overview to the documents that are contained in the bicluster. Thus, user explored bicluster chains by intermittently injecting documents into the chain. This enabled a rapid exploration of the dataset, and users were able to quickly follow leads of suspicious entities and identify the latent plot. Biclusters also played a significant role in the final analytic product of the users. The spatial workspace was used to visually maintain the biclusters and documents that the users deemed relevant. Therefore, their findings were based on not only the documents, but also the biclusters. Users referred to the biclusters as a collection of evidence through which two or more documents were connected. Also, users found biclusters to be a useful label for a particular region of the workspace, capturing and representing the relationships

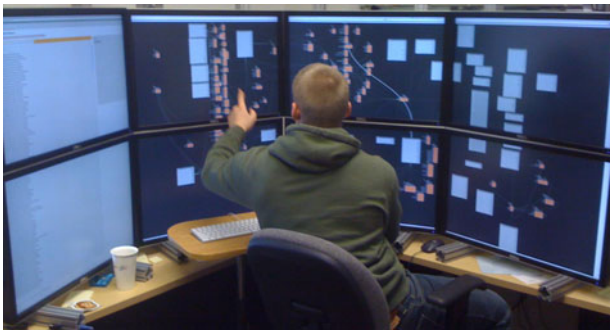


Fig. 16 Bixplorer on a large, high-resolution display

there at a high level. Thus, biclusters are a powerful visual representation of entity relationships within a data set. The encouraging results of this study show potential for future work exploring the benefits of biclusters not only as a visual representation of relationships, but also as a complex glyph with which users can interact.

3 Future opportunities

We have given an overview of four varied visual analytics projects, each of which provides rich capabilities for human interaction. We now present some possible themes that can serve to make interaction even more central, thus helping further the 'human is the loop' philosophy.

3.1 Mixing interaction modes

Users refer to information in different regions of spatializations with different contexts and metaphors (Andrews et al. 2010; Robinson 2008). Common metaphors include topical clusters, timelines, geospatial layouts, and social networks. Users frequently mix metaphors within the same workspace as either separate or nested schemas (Andrews et al. 2010; Robinson 2008). These metaphors may be well defined or ambiguous, and may evolve over time. This mixed-metaphor use of a spatialization poses challenges to layout and clustering models that are generally designed to compute a single model layout across the entire visualization. For example, iCluster (Drucker et al. 2011) which enables direct manipulation of a cluster model, could be combined with ForceSpire (Endert et al. 2012b) to enable dynamic layout of clusters, in much the same way as analysts currently do manually.

Challenge 1: How do we detect, interpret, compute, and visualize mixed models that represent mixed metaphors?

Challenge 2: How can we learn which model best captures the user's domain knowledge based on the layout?

Existing work has manually identified users' spatial metaphors (Andrews et al. 2010; Robinson 2008). Work in spatial parsers has developed heuristics for recognizing certain patterns (Marshall et al. 1994). Currently, tools make assumptions regarding user intentions (Endert et al. 2012b) or require explicit interaction by the user, such as switching views.

One way to organize mixed models is to operate at multiple levels of scale (Table 2). When all data points can feasibly be displayed on the screen, dimensionality reduction (DR) models can be used to lay out space, but this is less appropriate for larger datasets where the data points overfill the screen. At larger scales, cluster models can be aggregate data into visual groups. At even larger scales, information retrieval (IR) algorithms become essential to streaming or sampling data to dynamically display relevant data. A consistent direct manipulation approach to interaction can be applied across each level of scale. For example, IR algorithms can query for data relevance based on dimension weights learned by DR models, and learn from user actions such as placing uninteresting data in the 'trash pile.'

Challenge 3: How should direct manipulation be used to steer models across multiple scales?

Table 2 Multi-scale models

Levels of scale	Display scale	Database scale	Cloud scale
Usage Description	System lays out data according to users spatial organization feedback	System groups clusters of data in the layout according to users grouping feedback	System uses layout to query very large data and retrieve additional relevant data
Data scale of manipulation	<1 Million	<1 Billion	>1 Trillion
Algorithms	Dimensionality reduction	Clustering, Classification, Topic modeling	Information retrieval, sampling, streaming
Visualization	Spatial layout; Visual proximity = similarity	Groups, hierarchy, containment; Visual group = similarity	Salience, 3rd dimension; Visual salience = similarity
Interactive feedback for machine learning	Similarities, dimension weights, object weights	Group counts and contents, centroid landmarks, labels	Object relevance, keyword dimensions and weights

ForceSpire can be viewed as initial steps in this direction; it combines several of these techniques (e.g. document repositioning, highlighting, annotations, searching) in one system that tightly couple with the underlying dimension reduction model (Endert et al. 2012b). In addition to providing spatial constraints, the fundamental enhancement of this form of interactions is the ability to provide these constraints directly on the information (e.g., pinning a document to a specific location), and performing interactions that change the dimension-weighting scheme applied to the underlying dimension reduction models. For example, highlighting a phrase that contains a set of keywords implies increasing the weight of the corresponding dimensions (Endert et al. 2012b). The spatialization updates to reflect the incremental insights generated, creating a symbiotic relationship between the user's sensemaking and the system's machine learning.

3.2 Expressive forms of feedback for data mining algorithms

We have re-iterated the importance of user-provided feedback but thus far the forms of feedback considered are typically critiques of current results or preferences or constraints of desired outputs. It is not difficult to contemplate more structured and more expressive forms of feedback that will require significant re-tooling of algorithms.

> Challenge 4: Can we design expressive forms of feedback more naturally adapted to the visual forms of interaction that users desire?

For instance, in the storytelling algorithm described above, users typically are able to provide feedback in the form of 'I would prefer this story NOT use this document' or 'I would prefer that the story provide a justification for why this entity participates in it.' Such a feedback is quite non-trivial to translate back into the algorithmic machinery. This is because the algorithm is geared toward finding paths through document similarity networks, and thus the constraint must be translated into inequalities involving paths, and solved simultaneously to ensure that previously discarded paths become superior.

Going forward, it is conceivable that as visual analytics applications proliferate we will need to be more organized in terms of how we design user feedback/interaction mechanisms and the way in which such feedback is incorporated into constraints:

Challenge 5: Can we develop a taxonomy of user feedback and algorithmic constraints that can be used to standardize application development?

3.3 Space as a medium between human intuition and machine learning

Visualizations are intended to provide a visual representation of structure within information, with the purpose of illuminating patterns, relationships, trends, and other observable features within a dataset. Through continuous visual exploration, the features within the visual representation establish meaning to the user. For example, a two-dimensional spatialization of text reports may initially reveal themes or groups of related information. However, through exploration, more meaning and insight is generated, and the spatial layout begins to help the user construct a mental model of the data (Andrews et al. 2010; Endert et al. 2012).

An emerging opportunity for visual analytics is combining the computational advantages of data mining with the cognitive abilities of users by considering the visualization as a medium for interaction and analysis, and therefore an artifact to help facilitate common ground between user and system (Clark and Brennan 1991). Common ground (an understood shared knowledge between two or more parties) is created by the computational generation of the visualization, and the user-driven exploration and interaction with it.

Challenge 6: How can visualizations serve as artifacts for common ground between algorithms and users?

For instance, consider a two-dimensional spatialization created algorithmically (see Fig. 17). Similarity between data points is typically shown as relative Euclidean distance

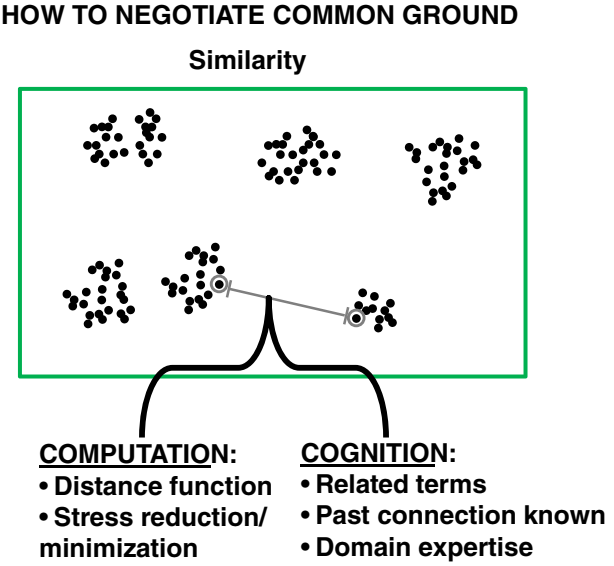


Fig. 17 Negotiating common ground between computation and cognition

between any two points. Computationally, this distance was determined by a distance function that calculated, based on several features of the data, and how far those two points should be apart. Such an output is easily interpretable by the user. However, users often have domain knowledge that contradicts the features used in distance function calculations. To share this knowledge with the system, they can interact with parameters of the distance reduction algorithm to reflect this knowledge. As a result, the system and user engage in a discourse to facilitate the process of common ground.

To strengthen this process, user interactions can be designed to occur directly within the visual representation of the data, instead of directly on model parameters. User interaction approaches such as semantic interaction (Endert et al. 2012), relevance feedback (MacArthur et al. 2002), and distance function learning (Brown et al. 2012) help facilitate this capability. These approaches present opportunities to engage users with metric learning, semi-supervised machine learning, and other computationally valuable methods of model steering without requiring expertise in data mining. Further, the ability for these approaches to enable the interactions purely within the visual space strengthens the process of common ground.

Challenge 7: How can domain knowledge be captured and communicated spatially?

In instantiating these features, the important distinction is in how the user communicates knowledge back to the system. Instead of directly manipulating model parameters, the insights that are gained spatially can be communicated spatially. If the user identifies two documents that are computationally placed far apart (implying dissimilarity), the distance function can be trained by relocating those two points closer together in the spatialization. As a result, the domain knowledge of the user is captured, interpreted, and extrapolated across the entire dataset (Endert et al. 2012), resulting in other data points correcting their relative distances from each other. The success of these approaches for user interaction in visual analytics has the potential to transform the analytic workflow of visual analytics users. Instead of structuring sensemaking around the computational models, the focus shifts back to thinking visually while maintaining the computational advantages of data mining.

3.4 Towards design principles

Our examples suggest design principles for ‘human is the loop’. User input and visual feedback are conducted and presented within the context of the analyst’s process. User input includes both the algorithm invocation command as well as the parameters and settings for the algorithm execution. Implicit steering is perhaps the ultimate form of in-context input as it passively takes advantage of interactions the analysts are already performing anyway (Endert et al. 2011), and the already existing objects/parameters of those interactions.

Yet, explicit steering can be carefully inserted within context as well. In the AW example, the user may explicitly invoke the algorithm to find connections, but the parameters evolve directly out of the user’s spatial layout and analytic process. Analysts frequently pose hypothetical connections by drawing a dotted line between entities, and thus also can trigger a connection finding algorithm. This perhaps suggests a potential implicit approach in which the invocation is automatic for proximal objects and numerous connections are visualized as a background distribution. Thus, space becomes the medium for computation.

At the opposite end of the spectrum would be completely out-of-context approaches. For example, the user might be required to export the data and load it into a separate algorithm while specifying numerous complex parameters, and then compare results back to their manual layout. The design tension is to strive for as much in-context as possible, while

preserving user control and expressiveness. It should be noted that the implicit approach, while appearing indirect to algorithm designers since interpretation is required, appears direct to the users because the operations are on objects of their concern, and in the domain of their expertise. Such implicit approaches map more closely to the user's flow of analysis (Elmqvist et al. 2011). When users stay in this 'cognitive zone' (Green et al. 2009), they can more effectively engage in sensemaking. Empirical evidence suggests that users prefer the implicit approach (Endert et al. 2012a) when carefully designed.

Interactions must also be cumulative. In many cases, the analyst must come to a conclusion incrementally (Shipman and Marshall 1999). If the conclusion were given to the analyst at the very beginning, it is likely that the analyst would not understand nor recognize it as a meaningful conclusion because it would be out of context. The analyst needed to experience the process. Sensemaking is inherently situated. Furthermore, there typically is not a single conclusion, but rather the analyst explores multiple alternative hypotheses so as to avoid confirmation bias (Heuer 1999). Thus, algorithms must incrementally adapt and compute over potentially large interaction data throughout this process.

This approach also suggests a highly integrated design in which many algorithms are simultaneously responding to user input. We are not suggesting a single panacea tool, but rather a compositional approach. In sensemaking for example, there are numerous opportunities for better integrating the foraging and synthesis halves of the sensemaking process (Andrews and North 2012).

4 Conclusion

We have provided a tour of visual analytics projects with a peek into the type of capabilities that might be enabled in the future. Beyond the interactive visualization and computational construction of semantically associated information objects, our goal is to ultimately understand how human analysts makes sense of data. The traditional viewpoint is that users can specify reasoning structures or frameworks and algorithms can help fill in the blanks. But it is not clear that such a viewpoint advances the user's conceptualization. We have argued that if space, visual entities, and algorithms become material objects that support joint reasoning between human and the machine, then users can perform actions that establish understanding to the algorithms, and be rewarded with results that fit naturally in the context of their analytic process. This can significantly further the cause and objectives of visual analytics research.

Acknowledgments This work is supported in part by the Institute for Critical Technology and Applied Science, Virginia Tech, and the US National Science Foundation through grant CCF-0937133.

References

Aghabozorgi, S.R., & Wah, T.Y. (2009). Recommender systems: incremental clustering on web log data. In *ICIS '09* (pp. 812–818).

Alonso, O., & Talbot, J. (2008). Structuring collections with scatter/gather extensions. In *SIGIR '08* (pp. 697–698).

Alsakran, J., Chen, Y., Zhao, Y., Yang, J., Luo, D. (2011). STREAMIT: dynamic visualization and interactive exploration of text streams. In *PACIFICVIS '11* (pp. 131–138).

Andrews, C., Endert, A., North, C. (2010). Space to think: large high-resolution displays for sensemaking. In *CHI '10* (pp. 55–64).

Andrews, C., & North, C. (2012). Analyst's workspace: an embodied sensemaking environment for large, high resolution displays. In *VAST '12*.

Bach, B., Pietriga, E., Liccardi, I., Legostaev, G. (2011). OntoTrix: a hybrid visualization for populated ontologies. In *WWW '11* (pp. 177–180).

Baron, A., & Freedman, M. (2008). Who is who and what is what: experiments in cross-document co-reference. In *EMNLP '08* (pp. 274–283).

Brown, E.T., Liu, J., Brodley, C.E., Chang, R. (2012). Dis-function: learning distance functions interactively. In *VAST '12*.

Cadez, I., Heckerman, D., Meek, C., Smyth, P., White, S. (2003). Model-based clustering and visualization of navigation patterns on a web site. *Data Mining and Knowledge Discovery*, *7*(4), 399–424.

Clark, H.H., & Brennan, S.A. (1991). Grounding in communication. In *Perspectives on socially shared cognition*. Washington, DC: APA Books.

Davidson, I., Ravi, S., Ester, M. (2007). Efficient incremental constrained clustering. In *KDD '07* (pp. 240–249).

Davidson, I., & Ravi, S.S. (2005). Clustering with constraints: feasibility issues and the k-means algorithm. In *SDM '05* (pp. 201–211).

Drucker, S.M., Fisher, D., Basu, S. (2011). Helping users sort faster with adaptive machine learning recommendations. In *INTERACT '11* (pp. 187–203).

Eccles, R., Kapler, T., Harper, R., Wright, W. (2008). Stories in GeoTime. *Information Visualization*, *7*(1), 3–17.

Elmqvist, N., Moere, A.V., Jetter, H.-C., Cernea, D., Reiterer, H., Jankun-Kelly, T.J. (2011). Fluid interaction for information visualization. *Information Visualization*, *10*(4), 327–340.

Endert, A., Fiaux, P., Chung, H., Stewart, M., Andrews, C., North, C. (2011). ChairMouse: leveraging natural chair rotation for cursor navigation on large, high-resolution displays. In *CHI EA '11* (pp. 571–580).

Endert, A., Fiaux, P., North, C. (2012a). Semantic interaction for sensemaking: inferring analytical reasoning for model steering. In *VAST '12*.

Endert, A., Fiaux, P., North, C. (2012b). Semantic interaction for visual text analytics. In *CHI '12* (pp. 473–482).

Endert, A., Fox, S., Maiti, D., Leman, S., North, C. (2012). The semantics of clustering: analysis of user-generated spatializations of text documents. In *AVI '12* (pp. 555–562).

Endert, A., Han, C., Maiti, D., House, L., Leman, S., North, C. (2011). Observation-level interaction with statistical models for visual analytics. In *VAST '11* (pp. 121–130).

Ernst, J., Nau, G., Joseph, Z. (2005). Clustering short time series gene expression data. *Bioinformatics*, *21*, i159–i168.

Fiaux, P. (2012). Solving intelligence analysis problems using biclusters. Blacksburg, VA: Master's thesis, Virginia Tech. http://scholar.lib.vt.edu/theses/available/etd-02202012-084450/.

Fink, G.A., North, C.L., Endert, A., Rose, S. (2009). Visualizing cyber security: usable workspaces. In *VizSec '09* (pp. 45–56).

Green, T.M., Ribarsky, W., Fisher, B. (2009). Building and applying a human cognition model for visual analytics. *Information Visualization*, *8*(1), 1–13.

Guha, R., Kumar, R., Sivakumar, D., Sundaram, R. (2005). Unweaving a web of documents. In *KDD '05* (pp. 574–579).

Henry, N., Fekete, J.-D., McGuffin, M.J. (2007). NodeTrix: a hybrid visualization of social networks. *TVCG*, *13*(6), 1302–1309.

Heuer, R. (1999). *Psychology of intelligence analysis*. CIA: Center for the study of intelligence.

Hossain, M.S., Akbar, M., Polys, N.F. (2012). Narratives in the network: interactive methods for mining cell signaling networks. *Journal of Computational Biology*, *19*(9), 1043–1059.

Hossain, M.S., Andrews, C., Ramakrishnan, N., North, C. (2011). Helping intelligence analysts make connections. In *AAAI '11 workshop on scalable integration of analytics and visualization (WS-11-17)* (pp. 22–31).

Hossain, M.S., Butler, P., Boedihardjo, A.P., Ramakrishnan, N. (2012a). Storytelling in entity networks to support intelligence analysts. In *KDD '12* (pp. 1375–1383).

Hossain, M.S., Gresock, J., Edmonds, Y., Helm, R., Potts, M., Ramakrishnan, N. (2012b). Connecting the dots between PubMed abstracts. *PLoS ONE*, *7*(1), e29509.

Hossain, M.S., Ojili, P.K.R., Grimm, C., Mueller, R., Watson, L.T., Ramakrishnan, N. (2012c). Scatter/gather clustering: flexibly incorporating user feedback to steer clustering results. In *VAST '12*.

Hossain, M.S., Tadepalli, S., Watson, L., Davidson, I., Helm, R., Ramakrishnan, N. (2010). Unifying dependent clustering and disparate clustering for non-homogeneous data. In *KDD '10* (pp. 593–602).

Hsieh, H., & Shipman, F.M. (2002). Manipulating structured information in a visual workspace. In *UIST'02* (pp. 217–226).

Huang, Y., & Mitchell, T.M. (2006). Text clustering with extended user feedback. In *SIGIR '06* (pp. 413–420).

Hwang, I., Kahng, M., Lee, S. (2011). Exploiting user feedback to improve quality of search results clustering. In *ICUIMC '11* (Vol. 5, pp. 68:1–68:5).

i2group. The analyst's notebook. http://www.i2group.com/us. Accessed 08 Oct 2012.

Jain, A.K., Murty, M.N., Flynn, P.J. (1999). Data clustering: a review. *ACM Computing Surveys*, *31*(3), 264–323.

Jeong, D.H., Ziemkiewicz, C., Fisher, B., Ribarsky, W., Chang, R. (2009). iPCA: an interactive system for pca-based visual analytics. *Computers and Graphics Forum*, *28*(3), 767–774.

Jin, Y., Murali, T.M., Ramakrishnan, N. (2008). Compositional mining of multirelational biological datasets. *ACM Transactions Knowledge in Discovery Data*, *2*(1), 1–35.

Kang, Y., Grg, C., Stasko, J. (2009). The evaluation of visual analytics systems for investigative analysis: deriving design principles from a case study. In *VAST* (pp. 139–146).

Keim, D.A., Mansmann, F., Thomas, J. (2010). Visual Analytics: how much visualization and how much analytics?. *SIGKDD Exploration Newsletter*, *11*(2), 5–8.

Kelleher, C., & Pausch, R. (2007). Using storytelling to motivate programming. *Communications of the ACM*, *50*(7), 58–64.

Kielman, J., Thomas, J., May, R. (2009). Foundations and frontiers in visual analytics. *Information Visualization*, *8*(4), 239–246.

Kuchinsky, A., Graham, K., Moh, D., Adler, A., Babaria, K., Creech, M.L. (2002). Biological storytelling: a software tool for biological information organization based upon narrative structure. *ACM SIGGROUP Bulletin*, *23*(2), 4–5.

Kumar, D., Ramakrishnan, N., Helm, R., Potts, M. (2006). Algorithms for storytelling. In *KDD '06*.

Kumar, D., Ramakrishnan, N., Helm, R., Potts, M. (2008). Algorithms for storytelling. *IEEE Transactions on Knowledge and Data Engineering*, *20*(6), 736–751.

Liang, J., Abidi, B., Abidi, M. (2003). Automatic x-ray image segmentation for threat detection. In *ICCIMA '03* (pp. 396–401).

Liu, J., Brown, E.T., Chang, R. (2011). Find distance function, hide model inference. In *VAST '11* (pp. 289–290).

MacArthur, S.D., Brodley, C.E., Kak, A.C., Broderick, L.S. (2002). Interactive content-based image retrieval using relevance feedback. *Computer Vision and Image Understanding*, *88*(2), 55–75.

Madeira, S.C., & Oliveira, A.L. (2004). Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions Computer Biology Bioinformatics*, *1*(1), 24–45.

Marshall, C.C., Shipman, III F.M., Coombs, J.H. (1994). VIKI: spatial hypertext supporting emergent structure. In *ECHT '94* (pp. 13–23).

Miao, G., Tatemura, J., Hsiung, W., Sawires, A., Moser, L. (2009). Extracting data records from the web using tag path clustering. In *WWW '09* (pp. 981–990).

Momtazpour, M., Butler, P., Hossain, M.S., Bozchalui, M.C., Ramakrishnan, N., Sharma, R. (2012). Coordinated clustering algorithms to support charging infrastructure design for electric vehicles. In *The ACM SIGKDD international workshop on urban computing, UrbComp '12* (pp. 26–133).

Monti, S., Tamayo, P., Mesirov, J., Golub, T. (2003). Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, *52*, 91–118.

Petrushin, V. (2005). Mining rare and frequent events in multi-camera surveillance video using self-organizing maps. In *KDD '05* (pp. 794–800).

Pirolli, P., & Card, S. (2005). The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *ICIA '05*.

Pirolli, P., Schank, P., Hearst, M., Diehl, C. (1996). Scatter/gather browsing communicates the topic structure of a very large text collection. In *CHI '96* (pp. 213–220).

PNNL (2012). Pacific Northwest National Laboratory, IN-SPIRE Visual Document Analysis. http://in-spire.pnnl.gov/. Accessed 08 Oct 2012.

Robinson, A.C. (2008). Design for synthesis in geovisualization. University Park, PA: PhD thesis, Pennsylvania State University.

Rzhetsky, A., Iossifov, I., Loh, J.M., White, K.P. (2006). Microparadigms: chains of collective reasoning in publications about molecular interactions. *Proceedings of the national academy of sciences, USA*, *103*(13), 4940–4945.

Sese, J., Kurokawa, Y., Monden, M., Kato, K., Morishita, S. (2004). Constrained clusters of gene expression profiles with pathological features. *Bioinformatics*, *20*(17), 3137–3145.

Shaparenko, B., & Joachims, T. (2007). Information genealogy: uncovering the flow of ideas in non-hyperlinked document databases. In *KDD '07* (pp. 619–628).

Shipman, F.M., & Marshall, C.C. (1999). Formality considered harmful: experiences, emerging themes, and directions on the use of formal representations in interactive systems. *CSCW*, *8*, 333–352.

Simoff, S., Bhlen, M., Mazeika, A. (2008). Visual data mining: an introduction and overview. In S. Simoff, M. Bhlen, A. Mazeika (Eds.), *Visual Data Mining* (Vol. 4404, pp. 1–12). Berlin/Heidelberg: Springer.

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

Thomas, J.J. & Cook, K.A. (Eds.), (2005). Illuminating the path: the research and development agenda for visual analytics. IEEE Computer Society Press.

Uno, T., Asai, T., Uchida, Y., Arimura, H. (2003). LCM: an efficient algorithm for enumerating frequent closed item sets. In *FIMI03*.

Van Wijk, J.J., & Van Selow, E.R. (1999). Cluster and calendar based visualization of time series data. In *INFOVIS '99* (pp. 4–9).

Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S. (2001). Constrained *k*-means clustering with background knowledge. In *ICML '01* (pp. 577–584).

Wang, X., & Davidson, I. (2010). Flexible constrained spectral clustering. In *KDD '10* (pp. 563–572).

Wright, W., Schroh, D., Proulx, P., Skaburskis, A., Cort, B. (2006). The sandbox for analysis: concepts and methods. In *CHI '06* (pp. 801–810).

Wu, H., Mampaey, M., Tatti, N., Vreeken, J., Hossain, M.S., Ramakrishnan, N. (2012). Where do i start? algorithmic strategies to guide intelligence analysts. In *ACM SIGKDD workshop on intelligence and security informatics ISI-KDD '12* (pp. 3:1–3:8).

Xu, Y., Olman, V., Xu, D. (2002). Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees. *Bioinformatics*, *18*(4), 536–545.

Zaki, M., & Hsiao, C. (2002). Charm: an efficient algorithm for closed itemset mining. In *SIAM international conference on data mining* (pp. 457–473).