Managing Discoveries in The Visual Analytics Process *

Di Yang, Zaixian Xie, Elke A. Rundensteiner and Matthew O. Ward
Worcester Polytechnic Institute
{diyang|xiezx|rundenst|matt}@cs.wpi.edu

ABSTRACT

Visualization systems traditionally focus on graphical representation of information. They tend not to provide integrated analytical services that could aid users in tackling complex knowledge discovery tasks. Users' exploration in such environments is usually impeded due to several problems: 1) valuable information is hard to discover when too much data is visualized on the screen; 2) Users have to manage and organize their discoveries off line, because no systematic discovery management mechanism exists; 3) their discoveries based on visual exploration alone may lack accuracy; and 4)they have no convenient access to the important knowledge learned by other users. To tackle these problems, it has been recognized that analytical tools must be introduced into visualization systems. In this paper, we present a novel analysis-guided exploration system, called the Nugget Management System (NMS). It leverages the collaborative effort of human comprehensibility and machine computations to facilitate users' visual exploration processes. Specifically, NMS first helps users extract the valuable information (nuggets) hidden in datasets based on their interests. Given that similar nuggets may be rediscovered by different users, NMS consolidates the nugget candidate set by clustering based on their semantic similarity. To solve the problem of inaccurate discoveries, localized data mining techniques are applied to refine the nuggets to best represent the captured patterns in datasets. Visualization techniques are then employed to present our collected nugget pool and thus create the nugget view. Based on the nugget view, interaction techniques are designed to help users observe and organize the nuggets in a more intuitive manner and eventually faciliate their sense-making process. We integrated NMS into XmdvTool, a freeware multivariate visualization system. User studies were performed to compare the users' efficiency and accuracy in finishing tasks on real datasets, with and without the help of NMS. Our user studies confirmed the effectiveness of NMS.

1. INTRODUCTION

Visualization systems traditionally focus on building graphical depictions of relationships among information in a human comprehensible format. By doing so, they help users to better understand the information. This means that the users can either learn facts that are difficult to discover without the graphical depiction, or the users' knowledge regarding some facts can become deeper or more precise. The usefulness of visualization systems has been well established [13; 14; 17].

*This work is supported under NSF grant IIS-0414380.

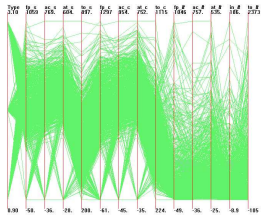

Figure 1: "AAUP" dataset visualized with Parallel Coordinates

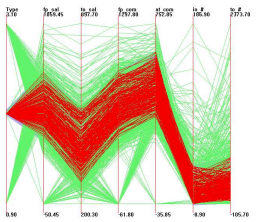

Figure 2: Complete cluster on seven dimensions of "AAUP"

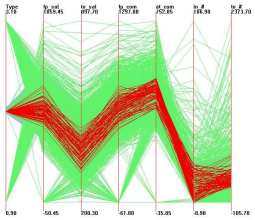

Figure 3: One "partial cluster" found by users

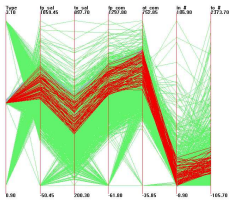

Figure 4: Another similar yet not identical "partial cluster"

Recently, visual analytics [15] has been proposed as a means to solve complex knowledge discovery tasks in many important fields, ranging from homeland security and credit fraud detection to financial market analysis. Solving such tasks usually requires analysts to perform complicated and iterative sense-making processes [5; 7]. Thus, it has been recognized that relying on analysts' perceptual power alone to conduct visual exploration may not always be the most effective method to solve these problems.

To fully support visual analytics, visualization systems have to be improved by tackling some key challenges. While we use clustering examples in Figures 1 - 4 to illustrate the these challenges, our goal over time is to support a rich set of patterns, including trends, outliers and associations. **1) Overloaded Displays:** When too much information is visualized on the screen, effective knowledge discovery is difficult. For example, as shown in Figure 1, when a dataset, even with modest numbers of records and dimensions, is visualized, overloaded displays can make knowledge discovery a time-consuming process. **2) Disorganized Discoveries:** Since there is no systematic discovery management mechanism provided by visualization systems themselves, users have to manage and organize their discoveries off line on their own. For example, some users, either due to rich domain knowledge or after a long time of exploration, may be able to identify the patterns (e.g., the cluster highlighted in red in Figure 2). Unfortunately, she may

not be able to store it in the system nor easily retrieve it for future exploration. Even if the systems provide some simple recording functionality, since a pattern may be repeatedly visited, redundant recordings may be generated (e.g., the clusters in Figures 3 and 4 are very similar). Such redundancy causes information overload that may hinder the future use of those recordings. **3) Inaccurate Discoveries:** Discoveries found by user's perceptual power alone may be inaccurate. For example, the "clusters" found by users in Figures 3 and 4 are actually subparts of a complete cluster depicted in Figure 2. Such inaccurate discoveries may lead to low-quality decision making (i.e., this user may miscount the population of the whole cluster, if she works on the "partial cluster" in Figure 3). **4) Isolated Knowledge:** Even if valuable knowledge may have already been uncovered, there is no convenient mechanism for users to access and share it. For example, a user interested in "clusters" in the dataset may spend a lot of time to find the one mentioned in Figure 2, even if it may have already been previously discovered.

Previous efforts to tackle these problems can be roughly classified into two categories. 1) User-driven: In this category, while the knowledge discovery process still relies on users' perceptual power, a variety of visual interaction mechanisms, such as zooming, filtering, color coding and dynamic querying, are offered by the visualization systems to facilitate exploration [1; 17]. Our framework applies these techniques to allow users to best use their perceptual power during visual exploration. 2) Data-driven: Data-driven techniques aim to expedite knowledge discovery with the help of the analytical power of machines. Data mining algorithms [4; 9; 23], which detect useful patterns or rules in large datasets, fulfill an important role here. These techniques are employed in our framework to improve the accuracy of discoveries.

More recently, some initial efforts have emerged to take advantage of both human perceptual abilities and computational power of computers to deal with the challenging process of knowledge discovery [15]. Visual data mining (VDM) [3; 8] involves users in the mining process itself, rather than being carried out completely by machines. In VDM, visualizations are utilized to support a specific mining task or display the results of a mining algorithm, such as association rule mining. However, VDM offers little help for knowledge organization and management, thus it does not support an iterative and comprehensive sense-making process. Our framework takes a different approach from VDM, that is, we put users at the first stage of the knowledge discovery process and apply data mining techniques as secondary method to refine and enhance what the users have already identified as interesting during their initial exploration. [5] proposed interactive tools to manage both the existing information and the synthesis of new analytic knowledge for sense-making in visualization systems. This work so far has not paid much attention on how to consolidate the users' discoveries. Collaborative visual analytics [7] introduced computational power into the sense-making process with a focus on supporting the exchange of information among team members. [6] proposed a framework to track the history of the knowledge discovery process for visualization systems. It created a generalized model so the tracking can be done across multiple application, systems, individuals and locations.

In this work, we design, implement and evaluate a novel analysis-guided exploration system, called the Nuggets Management System (NMS), which leverages the collaborative effort of human intuition and computational analysis to facilitate the process of visual analytics. Specifically, NMS first extracts nuggets based on both the explicit and implicit indication of users' interest. To eliminate possible redundancy among the collected nuggets, NMS combines similar nuggets by conducting nugget clustering. Then, data mining techniques are applied to refine the nuggets and thus improve their accuracy in capturing patterns present in the dataset. Visualization techniques are applied to the nugget pool and thus create an overview of the nugget space, which we call the nugget view. Furthermore, interaction techniques are designed based on the nugget view. By interacting with the nugget view, users will have more flexibility in observing the nuggets and be able to manage (e.g., users can attach annotations [11]) and organize nuggets (e.g., users can select a set of nuggets as the evidence to support a hypothesis) to support their sense-making processes. Lastly, the well-organized nugget pool can be used to guide users' exploration in both user- and system-initiated manners.

To verify the feasibility of NMS, we have integrated it into XmdvTool [17], a freeware tool developed at WPI for visual exploration and analysis of multivariate data sets. The main contributions of this paper are:

- We introduce a novel framework of analysis-guided visual exploration to facilitate visual analytics of multivariate data.
- We design a nugget combination solution that reduces the potential redundancy among nuggets.
- We present a nugget refinement solution, which utilizes data analysis techniques to improve the accuracy of the nuggets in capturing patterns in datasets.
- We present techniques for visualization and interactions with the nugget space, which allow users to observe and organize nuggets in an intuitive manner.
- We describe user studies evaluating the effectiveness of NMS. The user study demonstrates that NMS is able to enhance both the efficiency and accuracy of knowledge discovery tasks.

This work is an extension to two previous conference papers [20; 21]. The majority of the new material concerns the visualization of the nugget space and the sense-making process based on it, which are presented in Sections 5 and 6. The remainder of this paper is organized as follows: Section 2 introduces Nugget Extraction. Section 3 describes the techniques used in Nugget Combination. Nugget Refinement techniques are discussed in Section 4. Finally, we describe experimental evaluation in Section 7.

2. NUGGET EXTRACTION

2.1 Definition of Nuggets

Generally, a nugget is some valuable information extracted from the dataset, which could be clusters, outliers, associations and any other patterns. Additional attributes of a nugget, such as a name and annotations, can be attached to it as well. In our current implementation, a nugget is defined by a subset of the multivariate data plus the bounding box containing it.

The concept of nuggets is independent of the display methods in multivariate visualization systems, such as Parallel Coordinates, Scatterplot matrices and Glyphs [17]. We use Parallel Coordinates to demonstrate the examples in this paper. Thus visually a nugget appears as a blue band across the axes, which represents the query ranges on each dimension, and the red (highlighted) lines that indicate the selected records (result) of the query.

2.2 Nugget Extraction Based on User Interest

Nugget extraction can be achieved by observing a user's exploration process (user-driven) or by conducting analysis of the patterns existing in the data (data-driven). The NMS framework is

compatible with the nuggets derived using either of these two methods. Data mining algorithms for pattern detection have been extensively studied in the KDD community and any of these methods could be plugged into our framework. Here, we instead focus on nugget extraction via user-driven methods. The main benefit of user-driven methods is that we can bring into play the advantage of human perceptual and cognitive abilities to identify patterns in a knowledge discovery process. In NMS, the nuggets can be extracted based on either the explicit or implicit indication of users' interest. The specific techniques supporting those two nugget extraction methods are discussed in detail in [20].

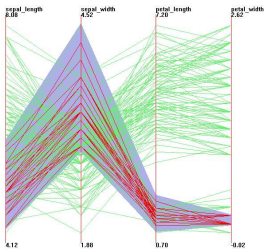

Figure 5: A nugget capturing a cluster in the "Iris" dataset

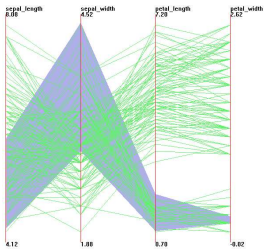

Figure 6: A nugget with no data record included

3. NUGGET COMBINATION

Relying on nugget extraction alone suffers from several problems. 1) Nugget redundancy may arise, because as the users navigate in the datasets, similar nuggets with slightly different boundaries are likely to represent the same data features. 2) An excessively large nugget pool generated during a long exploration period may make it difficult for users to access individual nuggets. 3) Continuous growth of the nugget population may lead to low system performance. An efficient method is needed to keep the nugget pool of modest size yet with high representativeness. Several techniques, such as sampling, filtering and clustering of nuggets may be employed to achieve this goal. We chose clustering, which groups similar nuggets and generates representatives for each group.

3.1 Distance Metrics

Clustering aims to group objects based on their similarities. It requires a distance measure that expresses the domain specific similarity between objects. To solve this problem, we developed distance metrics to capture the distances between any pair of nuggets.

Query Distance: Nuggets are defined by both queries and their results. So, naturally, nuggets defined by similar queries should be considered to be more similar than those defined by rather different queries. Thus our problem can be transformed into quantifying the similarity of queries. The major principle utilized in previous work [18; 19] for measuring query similarity (QS) between Nugget A and Nugget B can be summarized as:

$$QS(A,B) = \frac{QA \cap QB}{QA \cup QB} \quad (1)$$

Note that QA and QB are the qualifiers of these two queries. We adopt this idea as the basic principle for our query similarity measure on individual dimensions. We have also studied several important refinements to this basic idea, which enhance it to handle different types of domains (discrete, continuous, nominal) and at best level capture the visual similarity of nuggets. We have also extended the previous metric defined for a single dimension to now be applicable for multiple dimensions. Details of these techniques can be found in [21]. After we've normalized the acquired query similarities (between 0-1), we can easily calculate the query distances (QD) as shown in Formula 2:

$$QD(A,B) = 1 - QS(A,B) \quad (2)$$

Data Distance: However, nuggets are not only characterized by their queries (profile), but also by the results of the queries obtained when applying the queries to a particular dataset (content). As shown in Figures 5 and 6, two nuggets generated by very similar queries may be rather different in terms of actual data content. The former contains a cluster, while the latter is empty. Clearly, we need to enhance the capability of our distance metrics by also considering the "contents" of the nuggets. Now, the problem is how we can measure distance between two subsets of a multi-dimensional dataset. Previous work to tackle such problems [2; 12] can be classified into two main categories: statistical and transform-cost approaches. Below, we will introduce our proposed algorithm based on extending a basic transform cost algorithm.

In transform-cost approaches, the distance between two objects is expressed as the minimum cost of transforming one object to another. A well known algorithm that relies on Transform Cost is the Nearest Neighbor Measure (NNM) [12]. But unfortunately, NNM is a population-insensitive algorithm. It may lead to bad comparison results in our case, because comparing nuggets with different populations is going to be the norm in our work. We propose a new algorithm called the Exact Transformation Measure (ETM).

First, we formulate the problem. Given dataset $D, |D| = m$, and datasets A and B, $A \subseteq D, B \subseteq D, |A| = a, |B| = b, 0 \leq a \leq b \leq m, |A \cap B| = l, |B| - |A \cap B| = n$. Assume data points in D can be viewed as geometrically distributed in the value space based on their values in different dimensions. Our goal is to transform A to be exactly equal to B with minimum cost. To solve this problem, simply moving data points in A to their nearest neighbors in B will fail in many cases, because it is neither globally optimal nor sensitive to population. Thus, in order to achieve the transformation with minimum cost, we define three types of operations:

- *Move(x, y): given $x \in A, y \in B$, move x to the position where y lies.*
- *Add(x, y): given $y \in B$, add a new data point x to A at the same position where y lies.*
- *Delete(x) $x \in A$, delete x from A.*

By using "Move" and "Add", we are guaranteed to always be able to transform A to B, since A always has a smaller or equal sized population to that of B. However, simply relying on "Move" and "Add" will impose "forced matches", which may not always lead to the capture of the real distance between two datasets. Figure 7 shows an example of two 2-dimensional datasets where moving and adding are not sufficient to make a cost effective transformation plan. Members of datasets A and B are represented as white and black points, respectively. If only "Move" and "Add" are used, we have to match some data points in A with data points in B that are far away from them. In the worst case, the existence of a few "outlier" data points that do not have a "near neighbor" close to them will eliminate opportunities for many other data points to be matched with their real nearest neighbors.

To deal with this disadvantage of "forced matches", we use a "Delete" operation. With it, we no longer need to suffer from "forced matches", because for a given data point in A, "Move" is no longer the only option for it. We can choose "Delete", if moving it will bring too

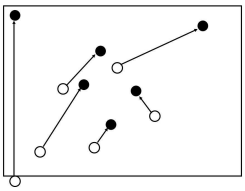

Figure 7: Trasforming A to B with moving and adding only

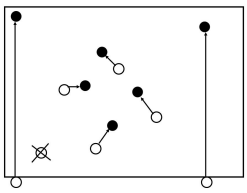

Figure 8: Transforming A to B with, moving, adding &deleting

much global cost. However, how to make an optimal transformation plan, which has the minimum cost, is still a complex problem. In order to tackle this problem, we need to study the cost of each operation first. **Cost(M[x,y]):** The cost of moving a data point x to y is equal to the normalized Euclidean distance between x and y (between 0-1). **COA:** Cost of adding a new point and **COD:** Cost of deleting an existing point are both estimated values that have a negative association with $|A|$.

Having set the costs of all our transfer operations, we now establish our solution for finding an optimal (most cost-effective) transformation plan. We note that making such an optimal transformation plan is non-trivial. Fortunately, the Hungarian Assignment [16] which was designed for finding minimum cost bipartite matches, provides a good approach to solving this problem. The algorithm takes an $n \times n$ matrix as input. Each row in the matrix represents a data point in A, and each column represents a data point in B. Then each entry is filled with the distance between the row and the column it belongs to. The algorithm returns a minimum cost match in $O(n^3)$ time.

Once we make a proper input matrix, the Hungarian Assignment Method will generate an output matrix representing the optimal matches. When the output matrix has been produced, by simply summing all the values in the input matrix entries that match an entry location with a "0" in its output matrix, and dividing the sum by $|B|$, we get the Data Distance (DD) between two nuggets.

Nugget Distance: Finally, we combine the Query Distance $(QD[X,Y])$ and Data Distance $(DD[X,Y])$ to present the Nugget Distance $(ND[X,Y])$ between any pair of nuggets X and Y.

$$ND[X,Y] = \alpha \cdot QD[X,Y] + \beta \cdot DD[X,Y] \quad (\alpha + \beta = 1) \qquad (3)$$

where α and β are experimentally derived weights. Note that ND will be normalized (between 0 to 1).

For more details about nugget distance, please see [21].

3.2 Nugget Clustering

Once we have computed the distances between nuggets, any generic clustering algorithm can be applied to conduct nugget clustering. The clustering process consolidates our nugget pool by removing redundant nuggets while keeping good representativeness. Besides the automatic nugget clustering to the whole nugget pool, our system also supports manual nugget clustering with provided interaction techniques based on visualized the nugget space. It may lead to more meaningful clusters for each individual user, because domain expert knowledge might be more effective than generic clustering algorithms. Moreover, this could even save the cost of running an expensive global clustering algorithm against all the collected nuggets. The visualized nugget space and the specific interaction techniques supporting manual nugget clustering are introduced in Section 5.

4. NUGGET REFINEMENT

4.1 Benefits from Nugget Refinement

In this section, we introduce the concept of using data mining techniques to refine the candidate nuggets extracted from users' logs. Such a refinement can be performed when a nugget was made because users were searching for some identifiable pattern types, such as clusters and outliers. For example, assume a user was searching for a cluster in the dataset, and for some reason, she missed part of it (Figure 9). Then, NMS will refine the nugget to capture the complete cluster (Figure 10).

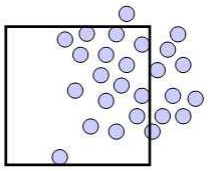

Figure 9: A nugget which captures the main body of a cluster but misses part of it

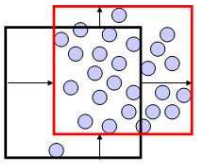

Figure 10: The refined nugget which captures the complete cluster

Nugget refinement offers two main advantages over both pure log analysis and mining techniques of the data itself. Firstly, log analysis techniques, for example, the nugget extraction introduced in Section 2, rely on users' actions only, without any help from computational analysis of the datasets and their properties. Thus they may lack accuracy in nugget specification. Nugget refinement likely improves the accuracy by exploiting both of them. Secondly, even assuming the system knows the specific pattern type a user is interested in, in many cases the user is not searching for all possible patterns but only for certain patterns of this type. This makes running expensive global pattern detection algorithms not cost effective and unrelated patterns detected may even cost users more effort to isolate the useful ones. We chose density-based clustering [4] and distance-based outlier detection [9] as our sample pattern detection algorithms, which are popular algorithms in the data mining field.

4.2 Techniques for Nugget Refinement

The refinement process is divided into two phases, called the match and refine phases.

In the match phase, we aim to match the identified nuggets with patterns "around them" within the data space. In other words, our goal is to determine which patterns users were searching for when these specific nuggets were made. In this work, we concentrate nuggets refinement on two important pattern types, clusters and outliers.

The concept of "Match" is used to judge whether some data patterns or the major parts of these patterns primarily contribute to a nugget. If it is the case, we call the nugget and these patterns "matched". The nuggets may be "matched" with more than one pattern. Or, put differently, a nugget may contain several patterns. Technically, to match a nugget with patterns, we have to compute two important factors that each represent one side of the match:

- *Participation Rate* (PR) : A pattern P should be matched with a nugget N, only if most of its members, if not all, participate in (are covered by) the nugget. For example, in Figure 11, for the cluster at the left side, data points 2, 3, 4, 5, 6 are covered by the nugget. So, we use PR to present how much of a pattern P is covered by a nugget N.

$$PR(N,P) = \frac{P.population \cap N.population}{P.population} \quad (4)$$

- *Contribution Rate* (CR) : Since "match" is two-directional, while PR just expresses one direction, namely, nugget to pattern, we introduce CR to capture the opposite direction, from patterns to nugget. This shows how much a whole or partial pattern contributes to the nugget. Moreover, because a nugget is decided by a query and the results of this query, we consider both the selected area and data population of the pattern and the nugget when calculating CR.

$$CR(N,P) = \frac{P.area \cap N.area}{2 * N.area} + \frac{P.population \cap N.population}{2 * N.population} \quad (5)$$

Next we show a specific example of how to calculate PR and CR between a nugget and a cluster (the left side cluster on Figure 11).

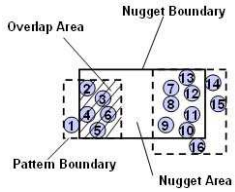

Figure 11: A nugget which captures the main bodies of two clusters

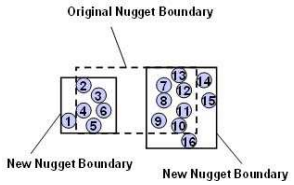

Figure 12: The refined nuggets which each capture a complete cluster

The covered pattern population ($P.population \cap N.population$) equals 5 (containing data points 2, 3, 4, 5, 6), and the pattern population ($P.population$) equals 6. So $PR = 5/6 = 0.83$. The Nugget Area ($N.area$) in this example is the area denoted by the Nugget Boundary. The Pattern Area ($P.area$) is indicated by the Pattern Boundary. Overlap Area ($P.area \cap N.area$) is the overlap area depicted by the shaded area in the figure. Let's assume Overlap Area/Nugget Area=0.3. The concept of "Area" here extends to the hypervolume when the number of dimension increases. We also know that the Nugget Population equals 12. So CR= (0.3+5/12)/2=0.39

Now we use PR and CR to match a nugget with the patterns around it. We use $MatchRate(P,N)$ to express the result of a match between a nugget N and all patterns of type P. Based on the match results, we classify nuggets into different categories. Here we concentrate our discussion on clusters. Techniques to handle other pattern types can be found in [20].

$$MatchRate(C,N) = \sum_{1 \le i \le n} PR(C_i,N) * CR(C_i,N) > T \quad (6)$$

Where C_i's are all the cluster patterns fully or partially covered by the nugget. T is a threshold which decides whether the nugget and the patterns match. In this case, a nugget is matched with one or more clusters. In other words, the main components of this nugget are clusters.

The match phase reveals what type of patterns a user was likely searching for. We now describe the refinement phase. If a nugget is classified into the first two categories mentioned above, we finish nugget refinement using the following two steps, called splitting (if necessary) and modification.

Splitting: If a nugget is composed of more than one pattern, we could split it into several new nuggets, each representing one pattern only. Because we already know all patterns the users were searching for from the match phase, simply putting all the members of each pattern into a new nugget will finish this job.

Modification: For the nuggets representing a single pattern only, the modification is to make the nugget boundaries exactly the same as the pattern boundaries.

In Figure 12, we show the new nuggets after nugget refinement. Each now represents one pattern only.

As with nugget clustering, nugget refinement can either be automatically performed against all the collected nuggets, or users could manually pick the nuggets of interest from the nugget space view as refinement candidates. Specific interaction techniques to support manual nugget refinement are also discussed in Secction 5.

5. NUGGET SPACE VISUALIZATION

Up to now, we have obtained a set of nuggets, each of which is either defined by users or is generated by extraction, combination and/or refinement. We call this set the *nugget space*. A natural requirement from analysts is a visual overview of these nuggets. In this section, we propose a visualization approach, the *MDS nugget starfield*, to present such an overview to users, which is inspired by the VaR display [22] developed by Yang et al. In addition, some interaction techniques are discussed to help users expose patterns in nugget space and do maintenance on nuggets.

5.1 MDS Nugget Starfield

Figure 13 shows an MDS nugget starfield to present a nugget space obtained from the cars dataset. Each glyph is an overview of a nugget and we generate this layout using an MDS algorithm [10]. The proximity among nugget positions reflects nugget distances. Assume that we have N nuggets in this nugget space, the procedure to get such a starfield is as follows: (1) We calculate nugget distances and record them into an N $\times$ N matrix. (2) This matrix is regarded as the input to an MDS algorithm [10], which generates a position for each nugget. (3) Each nugget overview is rendered in the position obtained from the MDS algorithm.

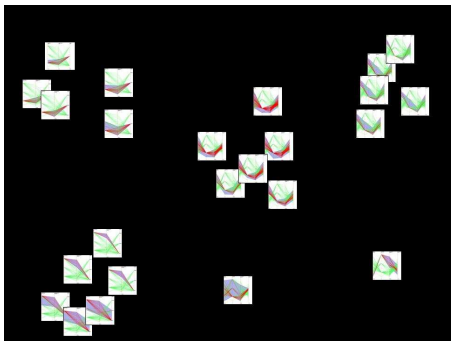

Figure 13: MDS Nugget Starfield. Each glyph represents a nugget. The distances among glyphs are determined by nugget distances.

The advantage of this approach is obvious. First, users can easily observe the distribution of nuggets and clusters in nugget space since this layout conveys the distance among nuggets. Then different actions, such as manual clustering and refinement, can be easily applied to some nuggets. For example, in Figure 13, we can see that there are four nugget clusters and two outlier nuggets. For each cluster, we can do nugget maintenance using interaction techniques as discussed in the following subsection.

5.2 Interactions on Starfield Layout

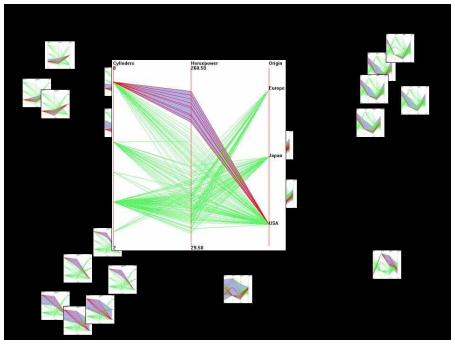

Figure 14: When users select a nugget on the MDS Starfield, a popup dialog shows its details.

We propose a set of interaction techniques to help users explore nugget space and maintain nuggets. These interactions include:
Focus+Context: If users want to see details of one nugget, an original view can be displayed as shown in Figure 14.

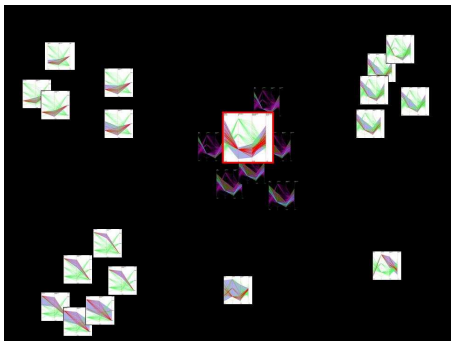

Figure 15: When users select a group of nuggets, the NMS system merges these queries and generate a new nugget.

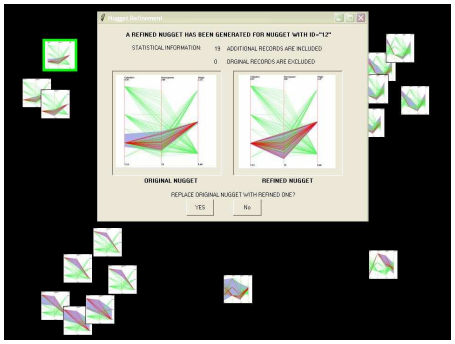

Figure 16: NMS refines a selected nugget and shows the results in a new dialog.

Nugget Brushing: We allow users to select a subset of nugget space based on attributes such as the person who defined this nugget, or when it was created or visited. In such a way, users can focus on important nuggets or those of interest. For example, selecting only those nuggets which were visited recently helps users focus on important nuggets and de-emphasizes old and less interesting ones.
Move: Users can move one nugget to make it closer to or farther from other nuggets using "drag and drop". The reason why we introduce this action is that nugget distance in the users' mind might be different from our distance formulas or the MDS algorithm result. Such an action allows users to correct the error from algorithms based on the knowledge of domain experts. It is possible for experts to make a mistake, thus we allow users to restore to the distances generated by our distance formulas or the MDS algorithm.
Compare: This action can popup a dialog to show the distances (query, data, nugget) between two nuggets selected by the user. The query arguments are also listed in this dialog to facilitate user's comparison.
Manual Clustering: As discussed before, NMS can do clustering on existing nuggets. However, users might not be satisfied with the automated clustering result. Thus our system allows users to select several nuggets and group them. A new nugget will be shown to replace these selected nuggets. For the flexibility, we do not delete the results from the clustering algorithms. Users can switch between automatic and manual results. An example of manual clustering is shown in Figure 15.
Automatic Refinement: Figure 16 shows the interface to do nugget refinement based on the algorithm discussed in Section 4. After the user select one nugget of interest, NMS runs the algorithm to refine the selected nugget and then pop up a dialog to show the results. If the user is satisfied with the refined results, NMS can replace the original nugget with the refined one.
In the above actions, *Move* and *Manual Clustering* potentially enable our system to learn some domain knowledge from experts, which is proposed as a part of our future work.

6. NUGGET-BASED SENSE-MAKING

In this section, we introduce techniques to support nugget-based sense making. As described in [5; 7], visualization-based sense making is usually a complicated and interactive process supported by continuous interacting with visualized evidence sets. Besides evidence collecting, another major task is to reveal the interrelations among individual pieces of evidence. Thus, beyond visualization techniques for evidence sets, namely, the nugget space in our case, more sophisticated nugget organization mechanisms need to be provided to support nugget-based sense making.
In particular, we use two distinct but interlaced phases to summarize the nugget-based sense making process. They are nugget selection and interrelation discovery. This is because a user's opinion of a certain hypothesis is eventually formed based on a selected set of evidence and also the interrelations among them. We use an important sense making model, which is hypothesis assessment, to demonstrate the techniques we develop to support nugget-based sense making. However, the general principles we propose for nugget-based sense making and the design for "hypothesis views" can easily be adapted to several other models, such as future prediction and alternative comparison. Before we present the details of nugget organization mechanisms, we introduce the hypothesis view, which acts as the nugget organization bed for hypothesis assessment in our system. As shown in Figure 17, a hypothesis view for a certain hypothesis is mainly composed of a "overview", a "support view" and a "refute view". Initially, all the nuggets related to the given hypothesis are collected in the "overview". The nuggets which are considered to be positive or negative evidence for the hypothesis are later separated from the "overview" and put into the "support view" and "refute view" respectively.

6.1 Nugget Selection

After long term exploration by multiple users, a single dataset may have accumulated a certain amount of nuggets in its nugget space.

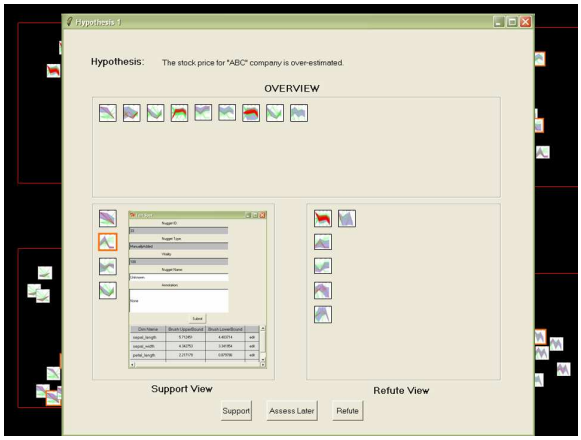

Figure 17: An example of "Hypothesis View" over four datasets. Totally 19 nuggets are collected as evidence to access a hypothesis: The stock value of 'ABC' company is over-estimated. 4 of them are considered to be positive evidence and thus being put in support view (the user is expanding the second one to view the details of it). 6 of them are judged to be the negative evidence and put in refute view. Others remain in the overview.

Many, if not most, of these nuggets may be unrelated to a certain hypothesis, because different users may have different intentions when exploring the dataset, and even for a single user, her exploration may be for pursuing multiple goals. Moreover, to assess a hypothesis (e.g., whether the stock value of a certain company is over-estimated), an analyst may need to analyze the evidence from multiple data resources (e.g., datasets recording stock market transactions, datasets recording company profits). Thus, our nugget selection techniques should allow users to efficiently select related nuggets from multiple nugget spaces. In NMS, such selections could either be achieved by users' direct operations on the system interface (e.g., drag nuggets from nugget spaces into the hypothesis view), or by automatically importing the nuggets, which fulfill the queries submitted by users, to the hypothesis view. NMS provides functionality, such as sorting and querying on statistical information and keyword based search on additional attributes, to help users quickly access the nuggets of interest. Figure 17 shows an example of selecting nuggets from four different nugget spaces.

6.2 Interrelation Discovery

Interrelation discovery against selected evidence is a critical yet complicated task for sense-making. It aims to incrementally integrate individual evidence pieces to form larger evidence pieces, i.e., sub-opinions, until the final opinion is reached. Here we note that there may exist numerous interrelation models among the evidence and some of them may be very complex. In this work, we describe several simple interrelation models among nuggets and the prototype in our system that helps alalysts use them in the hypothesis views. We show examples of the interrelation models we support in figure 18.

Group relation: Group relation is one of the simplest interrelation models. It indicates that a group of evidence together supports or refutes some component of a hypothesis. This is similar to nugget selection, because even among the positive or negative evidence for a hypothesis, different evidence may support or refute the hypothesis from different aspects. In NMS, users can group nuggets in the support or refute view and the system marks different groups with frames in different colors. Moreover, in a group of nuggets, some of them may have greater importance than others and act as the "core evidence" for the whole group, which is distinguished from others by a wider frame.

Sequence relation: Sequence relations among nuggets are based on group relations but involve the concept of a time sequence. This is important, because, in many cases, a group of evidence makes most sense when they are considered in a certain order. For example, in order to assess the hypothesis that the current gas price is at a wave bottom, a user may have to gather the national average gas prices for previous months, and analyze them in a time sequence. In NMS, users can define sequence relations among nugget groups. In particular, to express the order of two nuggets, users can stretch an arrow from one to another. Finally, the whole "nugget string" connected by the arrows forms a "storyline", which supports or refutes the hypothesis from one aspect (In the current version of our system, we do not allow "cycles" in a storyline).

the support or refute view, we also introduce an external relation between positive and negative evidence, which is a contradiction. Although, in general, the positive and negative evidence eventually contradicts with each other, the "contradiction" we define here refers to a direct conflict between two specific pieces of evidence. For example, a nugget extracted from dataset A shows that the gas price in Massachusetts is lower than the national average, while another nugget extracted from dataset B may indicate the opposite. Such direct conflict usually indicates uncertainty or errors in the datasets. Figuring out these contradictions will help users to be aware of "bad" evidence, and eventually avoid data from unreliable data sources. As shown in figure 18, in NMS, two nuggets involved in contradiction are marked by dashed frames and connected by a dashed line.

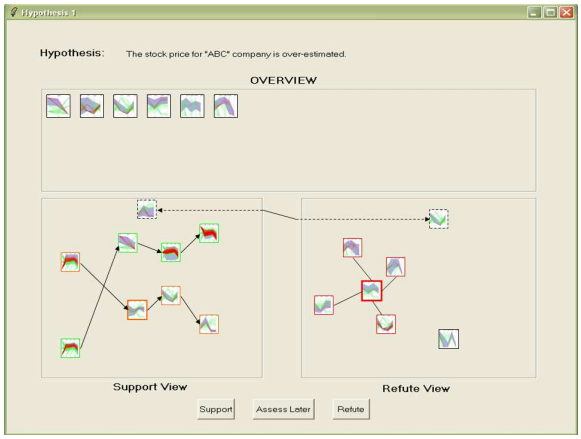

Figure 18: An example of interrelation discovery based on hypothesis view. 5 nuggets in refute view are grouped together to form an evidence group, and the one in the center is considered to be the "core evidence" of this group. 2 groups of nuggets, 4 in each, are recognzied to be evidence sequences and thus linked together by arrows in the support view. A pair of nuggets are pointing to each other by a double-direction dashed arrow, which indicates that they are contradicted with each other.

7. EVALUATIONS

In order to show the effectiveness of NMS, we have performed some preliminary user studies to compare user efficiency and accuracy when solving tasks with and without the help of NMS. We divided 12 students into 4 groups, 3 users per group. All 4 groups

were asked to finish the same 5 knowledge discovery tasks, which were based on 3 real datasets; three groups (group 2, 3, 4) were supported by NMS. All the users were encouraged to finish the tasks as quickly and correctly as possible. Our user studies showed that NMS may not only greatly improve users' time efficiency when solving knowledge discovery tasks, but also it can enhance users' accuracy of finishing these tasks. Details of the experimental setup, methodology and results of these user studies can be found in [20]. Here we note that these user studies were mainly designed to evaluate the functionalities of the analytical components of NMS, namely the nugget extraction, nugget combination and nugget refinement. The nuggets in these user studies were displayed with very basic visualization techniques, such as pull-down item lists. The advanced visualization and interaction techniques introduced in Section 5 and 6 were not employed in these user studies.

To further evaluate the functionalities of the nugget visualization (NV) and nugget-based sense making (NBSM) components in NMS, we conducted a preliminary case study to compare the effectiveness of NMS alone and NMS(NV+NBSM). In this case study, we invited 4 users (all WPI graduate students, but different from those involved in the previous user studies) to analyze the nuggets pool collected during previous users' exploration. They were asked to identify the most "well-mined" nuggets, i.e., those that capture users' interest best, and also to eliminate the "misinterpreted" nuggets, which were most likely generated by misinterpretation of users' interest. In this case study, 2 of the 4 users used NMS first and then NMS(NV+NBSM), while other 2 used them in the reverse order. Our case study showed that all 4 users were much more efficient in terms of both time spent and accuracy when using NMS(NV+ NBSM) to perform the tasks. Moreover, all 4 users gave comments that NMS(NV+NBSM) was "useful" and easier to use. This result was expected, because the nuggets view gave an intuitive overview of the interrelations among nuggets and the nugget sense-making component provides convenient mechanisms to organize the nuggets.

8. CONCLUSION

In this paper, we introduce a framework for analysis-guided visual exploration of multivariate data. Our system (NMS) leverages the collaborative effort of human intuition and machine computations to extract, combine, refine and visualize the valuable information (nuggets) hidden in large datasets. NMS also provides functionality to support users' sense-making processes based on the nugget space. Our preliminary evaluations indicate that NMS may greatly improve users' time efficiency when solving knowledge discovery tasks. It may also be able to enhance users' accuracy in finishing these tasks, although more complicated tasks are needed to validate this. Our future work includes expanding the recognizable nugget types(nugget extraction), pattern types(nugget refinement) and nugget interrelation models (nugget-based sense-making). Automatic mining for interrelations among nuggets and techniques to guide users' further exploration with the well-organized nugget spaces will also be investigated. Finally, more comprehensive user studies that involve more users and more complex tasks will be a major component of our future work.

9. REFERENCES

[1] A.Inselberg. Multidimensional detective. *Proc. of IEEE InfoVis*, pages 100–107, 1997.

[2] G. Cobena, S. Abiteboul, and A. Marian. Detecting changes in xml documents. In *ICDE*, pages 41–52, 2002.

[3] M. C. F. de Oliveira and H. Levkowitz. From visual data exploration to visual data mining: A survey. *IEEE Trans. on Vis. and Computer Graphics*, 9(3):378–394, 2003.

[4] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.

[5] D. Gotz, M. Zhou, and V. Aggarwal. Interactive visual synthesis of analytic knowledge. *IEEE VAST*, pages 51–58, 2006.

[6] D. P. Groth. Tracking and organizing visual exploration activities across systems and tools. *Proc. IEEE Conf. Information Visualization*, pages 11–16, 2007.

[7] P. Keel. Collaborative visual analytics: Inferring from the spatial organization and collaborative use of information. *IEEE VAST*, pages 137–144, 2006.

[8] D. A. Keim. Information visualization and visual data mining. *IEEE Trans. on Vis. and Computer Graphics*, 7(1):100–107, 2002.

[9] E. M. Knorr, R. T. Ng, and V. Tucakov. Distance-based outliers: Algorithms and applications. *VLDB J.*, 8(3-4):237–253, 2000.

[10] J. Kruskal and M. Wish. *Multidimensional Scaling*. Sage Publications, 1978.

[11] J. Lin and B. Katz. Question answering from the web using knowledge annotation and knowledge mining techniques clustering,. In *Proc. CIKM*, pages 116–123, 2003.

[12] E. A. Riskin. Optimal bit allocation via the generalized bfos algorithm. *IEEE Transactions on Information Theory*, 37(2):400–412, 1991.

[13] S.Havre, E.Hetzler, K.Perrine, E.Jurrus, and N.Miller. Interactive visualization of multiple query results. *Proc. of IEEE InfoVis*, 2001.

[14] B. Shneiderman. Tree visualization with tree-maps: A 2d space-filling approach. *ACM Trans on Graphics, Vol. 11(1), p. 92-99*, Jan. 1992.

[15] J. J. Thomas and K. A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society, Los Alamitos CA, 2005.

[16] K. Tranbarger and F. P. Schoenberg. The hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 12(2), 1957.

[17] M. Ward. Xmdvtool: Integrating multiple methods for visualizing multivariate data. *Proc. of Visualization '94, p. 326-33*, 1994.

[18] J. Wen, J. Nie, and H. Zhang. Clustering user queries of a search engine. In *World Wide Web*, pages 162–168, 2001.

[19] G. Xue, H. Zeng, Z. Chen, W. Ma, and Y. Yu. Clustering user queries of a search engine. In *Proc of the European Conference on Information Retrieval*, pages 330–344, 2005.

[20] D. Yang, E. A. Rundensteiner, and M. O. Ward. Analysis guided visual exploration of multivariate data. *VAST*, 2007. to appear.

[21] D. Yang, E. A. Rundensteiner, and M. O. Ward. Nugget discovery in visual exploration environments by query consolidation. In *Proc. CIKM*, 2007. to appear.

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

[23] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: an efficient data clustering method for very large databases. *SIGMOD Record, vol.25(2), p. 103-14*, 1996.