Analysis Guided Visual Exploration of Multivariate Data

Di Yang* Elke A. Rundensteiner† Matthew O. Ward‡§

Worcester Polytechnic Institute

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; 4) and 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 extracts the valuable information (nuggets) hidden in datasets based on the interests of users. Given that similar nuggets may be re-discovered 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. Lastly, the resulting well-organized nugget pool is used to guide users' exploration. To evaluate the effectiveness of NMS, 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.

Keywords: Visual Analytics, Visual Knowledge Discovery, Discovery Management, Analysis Guided Exploration

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical user interfaces H.2.8 [Database Management]: Database Applications—Data mining I.5.3 [Pattern Recognition]: Clustering—Similarity Measures

1 Introduction

Visualization systems traditionally focus on building graphical depictions of relationships among information in a human comprehensible format. By doing so, they help their users to better understand the information. This means the users can either learn some facts that are not easy to discover without the graphical depiction, or the users' knowledge to some facts can become deeper or more precise. The usefulness of visualization systems has been well established [22, **?**, 18].

*e-mail: diyang@wpi.edu

†e-mail:rundenst@cs.wpi.edu

‡e-mail:matt@cs.wpi.edu

§This work is supported under NSF grant IIS-0414380 and a grant from the NSA.

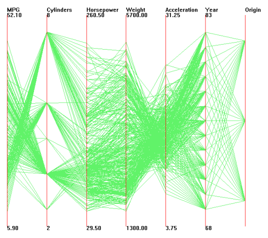


Figure 1: "Cars" dataset visualized with Parallel Coordinates

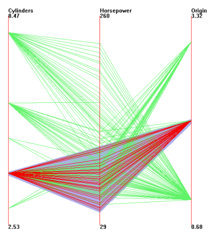


Figure 2: Complete cluster on three dimensions of "Cars"

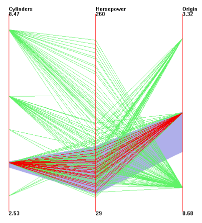


Figure 3: One "partial cluster" found by users

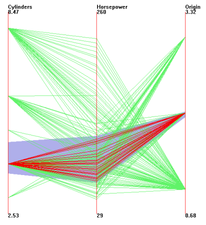


Figure 4: Another similar yet not identical "partial cluster"

Recently, visual analytics [20] has been employed to solve complex knowledge discovery tasks in many important fields of human society, 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 [11, 12]. 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. Here, while we use simple examples in Figure 1 - 4 to illustrate the these challenges, our goal over time is to support a rich set of patterns, including clusters, outliers and association rules. **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). But 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 by a single user or even multiple users, 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, not to mention conduct collaborative visual analytics. 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 [2, 22]. 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 [27, 14, 10], which detect useful patterns or rules in large datasets, fulfill an important role here. These techniques will be 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 [20]. Visual data mining (VDM) [9, 13] 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 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 knowledge discovery process and only apply data mining techniques as secondary method to refine and enhance what the users have already identified as interesting during their initial exploration. [11] 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 [12] introduced computational power into the sense-making process with a focus on supporting the exchange of information among team members.

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. We provide a rich set of functionalities to manage the nuggets. With them, nuggets can be maintained automatically (e.g., out-of-date nuggets can be pruned by the system) or by the users (e.g., users can attach annotations [15] to nuggets to facilitate nugget retrieval and sharing). Lastly, the well-organized nugget pool will be used to guide users' exploration in both user- and system-initiated manners.

As a general framework for analysis-guided exploration of multivariate data, NMS can be incorporated into any multivariate visualization system. To verify the feasibility of NMS, we have integrated it into XmdvTool [22], 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, which facilitates 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 sketch tools for the management and support of visual exploration based on a nugget pool.
- We implement the above techniques of NMS in XmdvTool, a freeware multivariate data visualization tool.
- 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.

The remainder of this paper is organized as follows: Section 2 introduces the Nugget Extraction. Section 3 describes the techniques used in Nugget Combination. Nugget Refinement techniques are discussed in Section 4. Sections 5 and 6 presents our ideas on Nugget Maintenance and Nugget Guided Exploration respectively. 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, typically some subpart of the whole dataset, which could be clusters, outliers, association and any other patterns. Additional attributes of a nugget, such as a name and annotations, can be attached to it as well. For the purpose of this paper, a nugget N is defined by a range query Q over a particular dataset D as well as the result of this query, namely the result dataset $Q(D)$. $N = \{D, Q, Q(D)\}$; The query Q selects all items from the dataset D for where the following constraints hold: $D.A_m = [A_m.b_l : A_m.b_h], D.A_l = [A_l.b_l : A_l.b_h], ..., D.A_n = [A_n.b_l : A_n.b_h]$; $\{D.A_m, D.A_l, ..., D.A_n\} \subseteq$ dimensions of D. $Q(D)$ projects these queried dimensions from selected items. $A_x.b_l$ and $A_x.b_h$ are the lower bound and the upper bound of the query ranges on attribute A_x, $[A_x.b_l : A_x.b_h]$ denotes a range of values "from $A_x.b_l$ to $A_x.b_h$"; $Q(D) \subseteq D$.

A more extensive range of nugget types will be considered in our future work. The concept of nuggets is independent of the display methods in multivariate visualization systems, such as Parallel Coordinates, Scatterplots and Glyphs [22]. Without loss of generality, we use Parallel Coordinates, which is a widely used method, 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. As shown in Figures 2, 3 and 4, users can specify different queries by adjusting the lower and upper bounds of the blue band (selection ranges). Users can also hide some dimensions if they are not interested in them.

2.2 Nugget Extraction Based on User Interest

Nugget extraction can be achieved by observing users' 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 [10, 5] 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, which is in fact one of the main reasons for developing visualization systems.

Similar to other systems [11, 12], users can explicitly indicate if a particular piece of information is of interest. This is done by explicitly saving the given query and labeling it by a persistent nugget name. NMS provides a rich set of functionalities to let users input, edit, and remove the nuggets as further discussed in Section 5. A non-intrusive alternative to explicit indication is implicit indication, a method found in intelligent systems [7, 4]. In NMS, nuggets can be extracted automatically by observing a user's exploration. "Visiting time" is one factor [7] used as the main indicator of a user's interest during visual analysis. NMS extracts a nugget if a user spends a long time visiting (querying over and looking at) a subpart of the dataset. For example, if a subpart shown in Figure 2 has been visited for a long time by a single user or repeatedly visited by one or more users, NMS will flag this data subset as a potential nugget. Problems that could be caused by such log mining, such as redundancy, out-of-date nuggets, and misinterpretation of users' interests, will be tackled in Sections 3 and 5.

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 by moving the sliders which control the range query boundaries, 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, because searching for nuggets of potential interest may become increasingly time-consuming. 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 [3], filtering [19] and clustering [5] of nuggets may be employed to achieve this goal.

After careful comparison, we chose clustering, which groups similar nuggets and generates representatives for each group. This is because when constructing a representative for each group, clustering techniques consider and combine the features of all the group members, while filtering and sampling techniques tend to just pick an "important" group member as their representative. Since in many cases, we can hardly tell which nugget is more important than others, constructing a representative which "speaks" for every nugget in a group makes more sense than just picking one.

3.1 Distance Metrics

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

3.1.1 Query Distance

Nuggets are defined by both queries and their results. So, naturally, nuggets that are 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 determining how to quantify the similarity of queries. The major principle utilized in previous work [23, 24] for measuring query similarity (QS) can be summarized as:

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

Here QS(A,B) represents the query similarity between Nugget A and Nugget B, and QA and QB are the qualifiers of these two queries. For example, given QA: select * from X where X.height =[5.25:5.85], QB: select * from X where X.height=[5.45:6.15], then we have $QA \cap QB = 5.85 - 5.45 = 0.40, QA \cup QB = 6.15 - 5.25 = 0.9$, QS=0.4/0.9=0.44. 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 [25].

After we've normalized the acquired query similarities (between 0-1), we can easily calculate the query distances (QD) as shown in Formula 2:

$$QD(A,B) = 1 - QS(A,B) \tag{2}$$

3.1.2 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

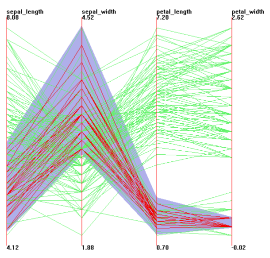


Figure 5: A nugget capturing a cluster in the "Iris" dataset

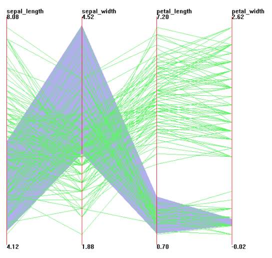


Figure 6: A nugget with no data record included

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 we must solve can be viewed as a general date analysis problem. That is, given two subsets of a multi-dimensional dataset, how can we measure the distance between them. Previous work to tackle such problems [17, 16, 8] can be classified into two main categories: statistic 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) [17]. 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. To solve this problem, 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. As shown in Figure 7, given datasets A and B whose

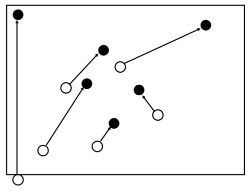


Figure 7: Trasforming A to B with moving and adding only

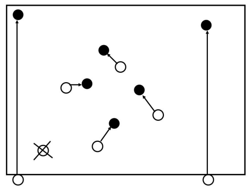


Figure 8: Transforming A to B with, moving, adding &deleting

members are represented as white and black points respectively, by using "Move" and "Add" only, 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 to "Delete", if moving it will bring too 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 $|A|$. Our heuristics for estimation are discussed in depth in [25].

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 [21] which was designed for finding minimum cost bipartite matches, provides a good approach to solve 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. Other issues that need to tackled when applying this algorithm to our case can be found in [25]

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.

3.1.3 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.

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

Since QD and DD are both normalized (between 0 to 1), ND will be normalized as well.

3.2 Nugget Clustering

Once we have learned the distances between nuggets, any generic clustering algorithm [10, 5, 27] can be applied to conduct nugget clustering. To provide a real time clustering service for our nugget pool, we employ incremental clustering algorithms [5, 27] in our system. Further discussion on the specific clustering algorithm used in our system can be found in [25]. The clustering process consolidates our nugget pool by removing redundant nuggets while keeping good representativeness.

4 Nugget Refinement

4.1 Benefits from Nugget Refinement

In this section, we'll introduce the novel concept of using data mining techniques to refine the candidate nuggets extraced 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 or 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).

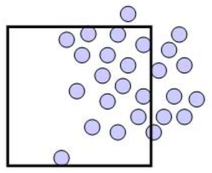


Figure 9: A nugget which captures the main body of a cluster but misses part of it

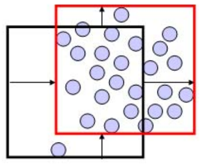


Figure 10: The refined nugget which captures the complete cluster

Nugget refinement offers two main advantages over both pure log analysis or 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. While 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. In this work, we chose density-based clustering [10] and distance-based outlier detection [14] as our sample pattern detection algorithms, which are popular algorithms in data mining field. However, the specific patterns and the refinement methods we propose below are only some of the possible ways that the nugget refinement could be carried out. Other refinement methods could equally be plugged into our framework.

4.2 Techniques for Nugget Refinement

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

4.2.1 Match phase

In this 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. We first formally define the concept of "Match".

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 (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 pattern or a partial pattern contributes to the nugget. Moreover, because a nugget is decided by a query and the results of this query (selected data), 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).

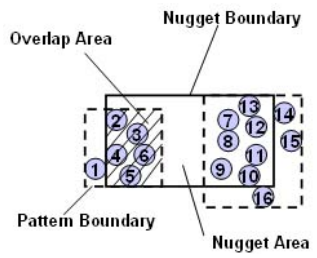


Figure 11: A nugget which captures the main bodies of two clusters

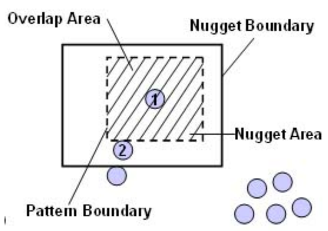


Figure 12: A nugget which includes an outlier (data point 1) and noise (data point 2)

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 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 3 different categories.

- *Clusters*

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

Where C_i's are all the cluster patterns covered or partially convered 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.

- *Outliers*

$$MatchRate(O,N) = \sum_{1 \le i \le n} CR(O_i,N) > T \quad (7)$$

Where O_i's are all the outlier patterns covered by the nugget. T is the same threshold we use in Formula 6. In this case, a nugget is matched with one or more outliers. In other words, the main components of this nugget are outliers. Here, we need to point out that although we still follow the notion of PR and CR as we did in the cluster cases, the way we calculate them is a little different. First, since an outlier pattern only has one data member, the PR for an outlier pattern is always 1. So we omit it from Formula 7. Second, the way we present pattern boundaries of outlier patterns is different also. As shown in Figure 12, the pattern area of an outlier is a (hyper) square area, where the distance from it to any boundary equals the maximum distance (among all the dimensions) between it to its nearest neighbor. All the other calculation processes remain the same as those for the cluster cases.

- *No Specific Pattern*

It is also possible that a nugget can be matched with neither clusters nor outliers. In this case, the nugget belongs to the "No Specific Pattern" category. Expanding the library of recognized patterns is an important part of our future work.

4.2.2 Refinement Phase

The match phase reveals to us what type of patterns that a user was likely searching for. 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 was 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 process becomes simple also, because we just need to make the nugget boundaries exactly the same as the pattern boundaries. In Figures 13 and 14, we show the new nuggets

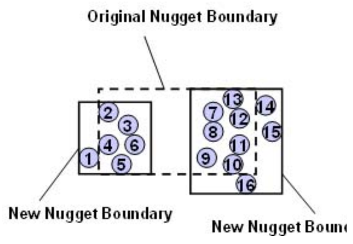


Figure 13: The refined nuggets which each capture a complete cluster

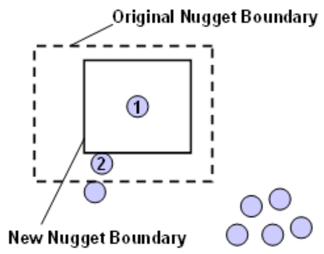


Figure 14: The refined nugget which includes an outlier only

after nugget refinement. Each now represents one pattern only.

5 Nugget Maintenance

In this section, we will discuss maintenance of the nugget pool. The nugget pool, which holds the nuggets collected during users' exploration, can be viewed as a database about users' insights about a specific dataset. Specifically, each nugget, namely the "tuple" in this database, is stored as an query specification annotated with additional attributes, as further discussed below. To faciliate users' further exploration, the nugget pool has to be built as an efficient, well-organized and user-friendly knowledge base. In our system, we build different types of indices to support quick access to nuggets. By providing different search features, nuggets can be easily located by searching on partial or complete query specifications or additonal attributes (e.g., names of nuggets). For example, a user can search for all the nuggets about the "car" dataset (a real dataset with 7 dimensions), which satisfy that the query range on dimension "cylinder" is between 4 and 8, or the names of the nuggets contain the string "high performance". NMS also provides the service to explicitly classify nuggets based on their characteristics, such as implicitly or explicitely indicated or types of patterns contained. Users can quickly narrow their search by browsing into different categories.

Besides organization of nugget pool, another important aspect of nugget maintenance is "clean up" of useless nuggets. Over the duration of the exploration, two potential hazards may leave useless nuggets in the nugget pool. 1) Out-of-date Nuggets: Some out-of-date nugget extracted early on and no longer of interest may become an unnecessary burden. 2) Misinterpreted Nuggets: Some nuggets may have been wrongly learned by misinterpreting users' interests.

To exclude these useless nuggets from our nugget pool, we introduce the concept of "vitality". Generally, the "vitality" of a nugget reflects the importance of this nugget. We use accumulated "visiting time" as its main indicator. A similar idea can be found in the literature [5, 1]. Specifically, each nugget obtains an initial "vitality" when it is extracted. This "vitality" fades as users' exploration period increases. A nugget can also gain "vitality" through two methods. 1) Being Directly Visited: If a nugget is retrieved by a user from the nugget pool, the time that this user spent on it contributes to its "vitality" increase. 2) Being Indirectly Visited: An existing nugget is indirectly visited if a similar new nugget is combined into it. This existing nugget absorbs the initial "vitality" of the new one as the increase of its own. Thus, briefly, nuggets created recently or visited frequently will have higher "vitalities", while those extracted a long time ago and never visited thereafter will have lower ones. Once the "vitality" of a nugget drops below a certain threshold, the nugget retires from the system.

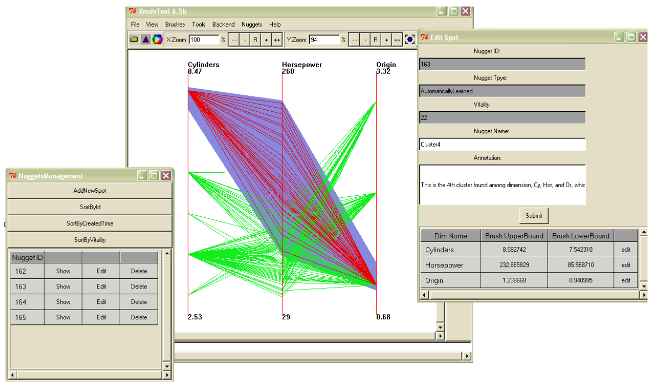
In NMS, such a natural "evolution" process of nuggets can be controlled by users. NMS allows users to cease, quicken, or slow down the "evolution" by setting different parameters, such as initial "vitality", fading rate, and increasing rate. Besides such macro-control, users can also directly manipulate any individual nugget. For example, users can mark a nugget as crucial, indicating that it should never be expired from the system. They could also directly delete some useless nuggets. Since the concept of "vitality" is to indicate the importance of nuggets, it may also provide meaningful hints for users' exploration in the nugget pool. For example, users could rank the nuggets by their "vitalities", and thus easily retrieve the "hottest" nuggets, which have received most attention.

Nugget maintenance leaves many opportunities for our future work, including: 1) How to give proper rights to multiple users working on the same nugget pool. 2) How to automatically learn and modify the parameters controlling nugget pool "evolution" during users' exploration .

6 Nugget-Guided Exploration

Nugget-guided exploration makes use of the nuggets we have learned to facilitate the knowledge discovery process. Figure 15 shows a screen shot of our prototype system. As we mentioned earlier, nuggets, as important carriers of valuable information, can be augmented with different kinds of additional attributes. Besides "vitality", names and annotations are examples of other attributes that can enrich the meaning of nuggets. For example, when exploring a dataset about "arriving passengers", a border control officer finds a nugget that represents a cluster existing in dimensions of "nationality", "arriving time", and "criminal records", she can give the nugget a meaningful name (i.e., "Suspicious Passengers Group"), and attach an annotation about her concerns to this passenger group. Such attached information will not only make it more convenient to retrieve this nugget, but also makes her nuggets shareable with other users. Meanwhile, statistical information, such as the number of data records included and average values on each dimension, can be automatically computed and attached to the nugget.

Nugget-guided exploration can be carried out in both user- and system-initiated modes. 1) User-initiated: Within this mode, users take the initiative to search and retrieve nuggets when they desire to. As mentioned earlier, NMS provides functionalities, such as sorting and querying on statistic information, keyword based search on additional attributes, to help users quickly access the nuggets of interest. Moreover, as our users are working with visualization systems, which hold the natural advantages of constructing graphic presentations to information, we could re-apply these visualization techniques to our nugget pool. For example, pixel oriented techniques [26] could be adopted to visualize our nugget pool in a condensed format, while VaR [26] could be employed to display the similarity between nuggets. With them, users could observe the whole nugget pool in a single screen and quickly learn the interrelations among nuggets. 2) System-initiated: NMS can take the initiative also when guiding users' exploration. Such guidance will be given based on watching the users' exploration. For example, when a user is querying a subpart of the dataset that is similar to one of the existing nuggets, NMS could inform the user that previous users have already found a nugget similar to what she is looking for. Other sophisticated services and HCI issues will be studied in our future work, including: How to provide hint to users about potential useful information in an as non-intrusive way as possible. How to build hierarchical structures among nuggets based on their interrelation (e.g., some nuggets may be subparts of a bigger nugget). How to guide users based on their profiles using collaborative filtering techniques [6].



Nugget ID			
162	Show	Edit	Delete
163	Show	Edit	Delete
164	Show	Edit	Delete
165	Show	Edit	Delete

Nugget ID: 163

Nugget Tpye: AutomaticallyLearned

Vitality: 22

Nugget Name: Cluster4

Annotation: This is the 4th cluster found among dimension, Cy, Hor, and Or, whic

Dim Name	Brush UpperBound	Brush LowerBound	
Cylinders	8.082742	7.542310	edit
Horsepower	232.865829	85.568710	edit
Origin	1.238668	0.940995	edit

Figure 15: A screen shot from the NMS prototype when looking for clusters hidden in the dataset

7 User Studies

In order to show the effectiveness of NMS, we have performed user studies to compare users efficiency and accuracy when solving tasks with and without the help of NMS. Also, we have observed and analyzed the stability of NMS through our user studies.

7.1 Experimental Setup

For all the user studies, we use an HP Pavilion laptop computer with 1.6MHz CPU and 512M memory. NMS is integrated in XmdvTool 7.0, which is the latest version of this multivariate visualization system [22]. **Users:** 17 subjects, all WPI students from various majors, participated in our user studies. Those in the NMS group were also given the basic idea of how NMS works and made familiar with the interfaces. Three real datasets were employed in our user studies. They are the "Iris" dataset (4 dimensions, 150 records); the "Cars" dataset (7 dimensions, 392 records); and the "Aaup" dataset (14 dimensions, 1161 records). **Tasks:** Users were asked to finish five knowledge discovery tasks. Each of the tasks requires users to study a dataset and answer a question about the dataset. These questions ranged from straightforward ones, such as, "in the car dataset, which origin has the highest average MPG?", to complex ones, such as, "How many clusters exist in the "Cars" dataset on 3 dimensions: Cylinders, Horsepower and Origin?". **Methodology:** In this user study, users were not allowed to communicate information about the user study through any other channel except NMS at any time before, during, or after the user study. This is to make sure that users can only solve the tasks based on their own exploration and the help from NMS (if available). A uniform training process was designed to give the basic idea of how NMS works and made users familiar with the interfaces of NMS. All the users were encouraged to finish the tasks as quickly and correctly as possible. No fixed time limit was inforced. Users were asked to finish the tasks in the same sequence as the tasks appeared on the task sheet.

7.2 Users' Time Efficiency

To compare users' efficiency of finishing this given set of tasks with and without help of NMS, we randomly divide the twelve users into four groups, three per group. Each user is asked to finish the same five tasks. Among these 4 groups, users of group 1 were asked to finish the tasks without NMS, while the other three groups (2- 4) were supported by NMS. Members of each group were randomly given a sequence number ranging from 1 to 3, which represents the user's sequence of solving the problems in his/her group. For example, once a user from group 1 receives the sequence number 2, he/she will be the second one in group 1 to finish the tasks. Group members were encouraged to use the functionalities of NMS (if available) to manage and share their discoveries with later users. Figure 16 shows the time used by each user and group to finish

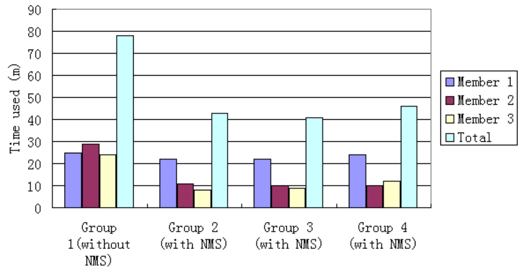


Figure 16: Comparison of users' efficiency in different groups

the tasks. As shown in Figure 16, groups 2, 3, and 4 (with NMS) spent noticeable less time (around 50 percent) than group 1 (without NMS). Such time savings due to the second and the third users, given that the first users all worked from scratch. Although NMS did facilitate their job, managing discoveries needed time. However, once the nuggets were extracted during the exploration by the

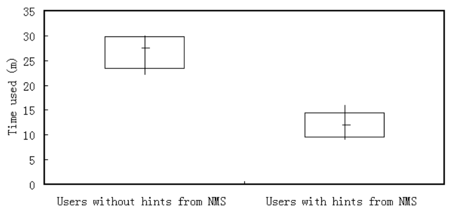


Figure 17: Comparison of users' efficiency with and without guidance from nugget pool

first users, the exploration processes of the second and the third users largely benefited from the nugget pool.

To better support our analysis, we compared the time used by six users working from scratch (three members of group 1, and three first users of each other groups) and by the other six users working with guidance of the nugget pool (the second and third users of groups 2,3 and 4). Figure 17 shows that the later six users with guidance of nugget pool were working much more efficiently. Specifically, the minimum, maximum, and the average time spent by these users are all much less than those who worked from scratch. The standard deviation is lower too.

7.3 Accuracy of Accomplished Tasks

We also studied the effect of NMS on the accuracy of the accomplished tasks. Among the five tasks we mentioned earlier, tasks 2 to 5 are straightforward problems. Users gave correct answers, although spending different amounts of time on them. However, task 1 is a common but complex knowledge discovery problem. Since not all the clusters can be easily found, in our user study the answer provided by the users varied. Figure 18 shows the number of clusters found by each user and also the number of clusters that actually exist. Two facts can be observed from Figure 18: 1) The number

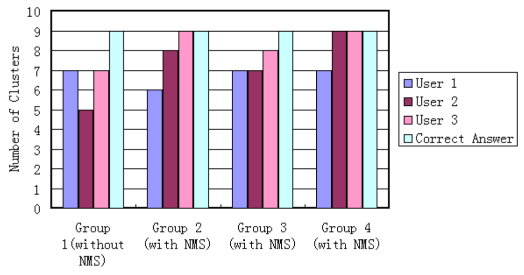


Figure 18: Comparison of users' accuracy of finishing complex task with and without help from NMS

of clusters correctly found by users working with NMS are generally closer to the number of actual clusters. 2) The later users in each group are more likely to find all existing nuggets compared to the earlier ones. These two facts show the promise of NMS indeed improving the accuracy of the tasks accomplished by the users.

7.4 Stability of NMS

Lastly, we consider NMS's stability, meaning how well it performs after long-term use. To give a preliminary evaluation of this, we asked a 7-user group to solve the five tasks mentioned earlier. We analyzed the population of the nugget pool after use by each user. The change of the nugget pool population is shown in Figure 19. The comparison of average number of nuggets identified by the 7 users for each task and the estimated number of nuggets needed for each task is shown in Figure 18. The "Estimated" number of nuggets is given based on our own experience of how many nuggets are needed for each task. From Figures 18 and 19, we observe two

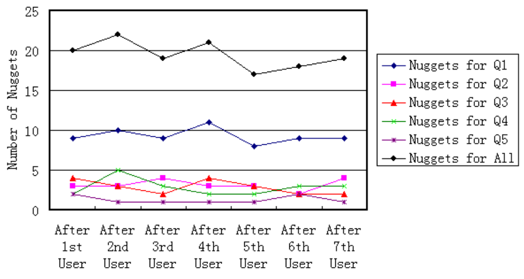


Figure 19: Evolution of nugget populations over time

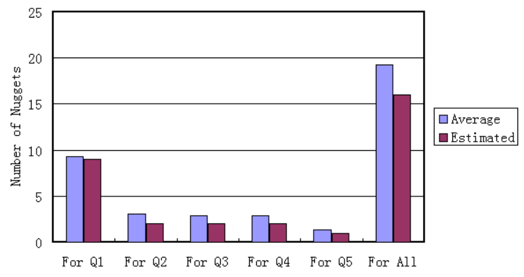


Figure 20: Comparson of numbers of nuggets generated by users and those are estimated

facts: One, the average number of nuggets formed by each user for each question is generally a little higher than estimated (Figure 20). Two, the populations of the nugget pools are relatively stable during users' exploration (Figure 19). Thus although some useless nuggets may be generated during exploration, they are usually only small portions of the whole nugget pool. Thus, this is indication that the population of the nugget pool is not likely to degenerate dramatically during a long-term use.

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 maintain the valuable information hidden in large datasets. Finally, a well-organized nugget pool can be used to guide users exploration. 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 of finishing these tasks, although more complicated tasks are needed to validate this. Lastly, NMS works in a stable manner during explorations by a sequence of users. This shows its promise of working well during long-term exploration. More comprehensive user studies which involve more users and more complex tasks will be one of the directions for our future work.

Acknowledgements: We thank Jing Yang, Zaixian Xie, Qingguang Cui, Charudatta Wad, Do Quyen Nguyen, and other members of the XMDV team at WPI for the development of XmdvTool, which formed the testbed for this work. We also thank members of DSRG at WPI for listening to talks and providing input on this work.

References

[1] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In *VLDB*, pages 81–92, 2003.

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

[3] M. R. Bolin and G. W. Meyer. A perceptually based adaptive sampling algorithm. In *SIGGRAPH*, pages 299–309, 1998.

[4] I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White. Visualization of navigation patterns on a web site using model-based clustering,. In *Proc. of ACM SIGKDD*, pages 280–284, 1998.

[5] F. Can. Incremental clustering for dynamic information processing. *ACM Trans. Inf. Syst.*, 11(2):143–164, 1993.

[6] Z. Chedrawy and S. S. R. Abidi. An item-based collaborative filtering framework featuring case based reasoning. In *IC-AI*, pages 286–292, 2005.

[7] M. Claypool, P. Le, M. Wased, and D. Brown. Implicit interest indicators. In *Intelligent User Interfaces*, pages 33–40, 2001.

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

[9] 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.

[10] 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.

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

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

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

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

[15] 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.

[16] C.-C. Pan, K.-H. Yang, and T.-L. Lee. Approximate string matching in ldap based on edit distance. In *IPDPS*, 2002.

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

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

[19] D. R. Tauritz and I. G. Sprinkhuizen-Kuyper. Adaptive information filtering algorithms. In *IDA*, pages 513–524, 1999.

[20] 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.

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

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

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

[24] 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.

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

[26] J. Yang, A. Patro, S. Huang, N. Mehta, M. O. Ward, and E. A. Rundensteiner. Value and relation display for interactive exploration of high dimensional datasets. In *InfoVis*, pages 73–80, 2004.

[27] 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.