Mining Interpretable Human Strategies: A Case Study

Xiaoli Fern, Chaitanya Komireddy, Margaret Burnett
School of EECS, Oregon State University
1148 Kelly Engineering Center
Corvallis, OR 97331
xfern, komirech, burnett@eecs.oregonstate.edu

Abstract

This paper focuses on mining the strategies of problem-solving software users by observing their actions. Our application domain is an HCI study aimed at discovering general strategies employed by software users and understanding how such strategies relate to gender and success. We cast this problem as a sequential pattern discovery problem, where user strategies are manifested as sequential patterns. Problematically, we found that the patterns discovered by standard algorithms were difficult to interpret and provided limited information about high-level strategies. To help interpret the patterns and extract general strategies, we examined multiple ways of clustering the patterns into meaningful groups, which collectively led to interesting findings about user behavior both in terms of gender differences and problem-solving success. As a real-world application of data mining techniques, our work led to the discovery of new strategic patterns that are highly correlated with user success and had not been revealed in more than nine years of manual empirical work. As a case study, our work also highlights important research directions for making data mining more assessible to non-experts and easier to apply.

1. Introduction

How can data mining be applied to better understand human behaviors? In attempting to understand how humans interact with computer systems, researchers in the Human-Computer Interaction (HCI) field often collect log data, which records user actions while using software. Often such data is manually analyzed by HCI researchers in order to understand how effective the software is supporting different users in achieving their goals. In part, this is because data about human behaviors does not seem particularly amenable to data mining efforts. For example, humans are inconsistent in their ways of approaching a task, and often introduce extraneous and irrelevant actions, resulting in data with significant noise and large variation. Further, there is often important contextual information that is not included in the log data, such as the semantics of the software environment with which the users are interacting. Interpreting data mining results from log data without such contextual information can be difficult and problematic.

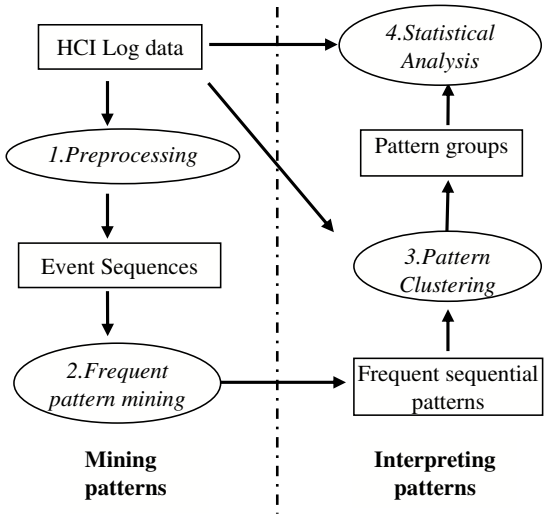


Figure 1. Our data mining process has four steps: 1. preprocessing; 2. frequent pattern mining; 3. Pattern clustering; and 4. Statistical analysis of pattern groups. Arrows represent the information flow.

In this paper, we apply data mining to a set of HCI log data collected in a particular problem-solving setting, namely users debugging spreadsheet formulas. We have the following goals. First, we wanted to automatically extract the general strategies used by software users for the problem-solving task they were performing. Second, we wanted to relate these strategies to user gender and problem-

solving success, which can then be used to help design better software that encourages the use of successful strategies and supports both genders. Finally, as a case study, we wanted to investigate the applicability of data mining techniques to this type of human behavior data, with a special focus on the interpretability of the mined results.

Figure 1 shows a summary of our overall data mining process. This process consists of two major parts. The first part is to find basic behavioral patterns from the data. The second part is to interpret these patterns, extract general strategies from them and relate them to gender and problem-solving success. Using this framework we were able to discover interesting high-level strategic patterns. Some of our main findings include: 1) Discovering patterns that match the verbalizations of users regarding strategy in an independent user study. 2) Discovering a strategic phenomenon that was hypothesized but not yet statistically varified by by HCI researchers in more than three years of manual empirical work. 3) Discovering two new strategic patterns that are highly correlated with user success and had not been revealed in more than nine years of manual empirical work.

While our application of data mining in this domain was quite successful, a significant amount of effort and data mining expertise was required. In particular, it is clear that the existing data mining tools would not have been sufficient for HCI researchers, without data mining expertise, to have made our discoveries. In this respect, our work highlighted a particularly important research direction for making data mining tools more useful to the data-mining novice. Key to our success was the consideration of a diversity of grouping mechanisms for the low-level patterns discovered by standard data mining tools. This provided insights that were not available from any single grouping. However, the process of selecting the grouping mechanisms was largely human-directed and quite tedious. This suggests that automated techniques for generating a set of diverse and potentially interesting groupings of low-level patterns is a key direction toward making data mining more assessible and easier to apply.

This paper makes the following contributions. First, we are applying data mining to a very challenging problem — identifying and understanding human strategies from noisy HCI log data. Second, a primary focus of this work is on producing interpretable results. There has been a significant amount of work devoted to the interpretability issues; however, we rarely see them applied to a real-world challenging application like ours. Third, as a case study of a pre-existing, ongoing project by seasoned HCI researchers, the lessons learned are of significant *practical* value to future applications of frequent pattern mining in the real world and suggest important research directions in data mining.

Table 1. Common actions and their meanings

Action Name	Explanation
PostFormula (PF)	Open a cell to show its content
HideFormula (HF)	Close a cell to hide its content
EditValue (EV)	Edit a value cell
EditFormula (EF)	Edit a formula cell
CheckMark (CM)	Placing CheckMark on a cell to mark its value as correct
XMark (XM)	Placing XMark on a cell to mark its value as incorrect
ArrowOn (AON)	Toggle an arrow on to show the dataflow dependency
ArrowOn (AOF)	Toggle an arrow off to hide the dataflow dependency

2 Case Study Setting

Our case study is situated in an HCI research project termed the "Gender HCI" project [2]. For this project, seasoned HCI researchers have conducted an extensive set of empirical user studies to collect in-depth data about user activity when using problem-solving software.

The problem-solving software used is a research prototype extension of spreadsheets [4, 5]. Figure 2 shows a snapshot of this prototype. This software is designed to aid users in debugging spreadsheets, providing functionalities for systematically testing a spreadsheet and giving feedback to help identify the bugs. This includes features that allow users to incrementally "check off" (Checkmark) or "X out" (Xmark) values that are correct or incorrect respectively. The software tracks the testing progress made by a user, which is displayed using varying cell border colors such that, as more testing is done, the color of a cell changes from red (light grey in Figure 2) to blue (dark grey in Figure 2). The visual feedback also includes a progress bar at the top to show the overall testedness of the spreadsheet. Finally, users can toggle arrows on and off that depict not only the dataflow relationships among cells but also the testedness status of these relationships.

```
15:43:47, TooltipShowing, CELL31567926-2332 …
15:44:12, CheckMark, CELL31567926-2332 …
15:44:57, CheckMark, CELL31567926-2332 …
```

Figure 3. An excerpt from the user action logs

The software has been instrumented to record user actions in log files. An individual user action is defined as a user's physical interaction with a debugging feature, such as placing an Xmark in a cell. In total, there are 19 ac-

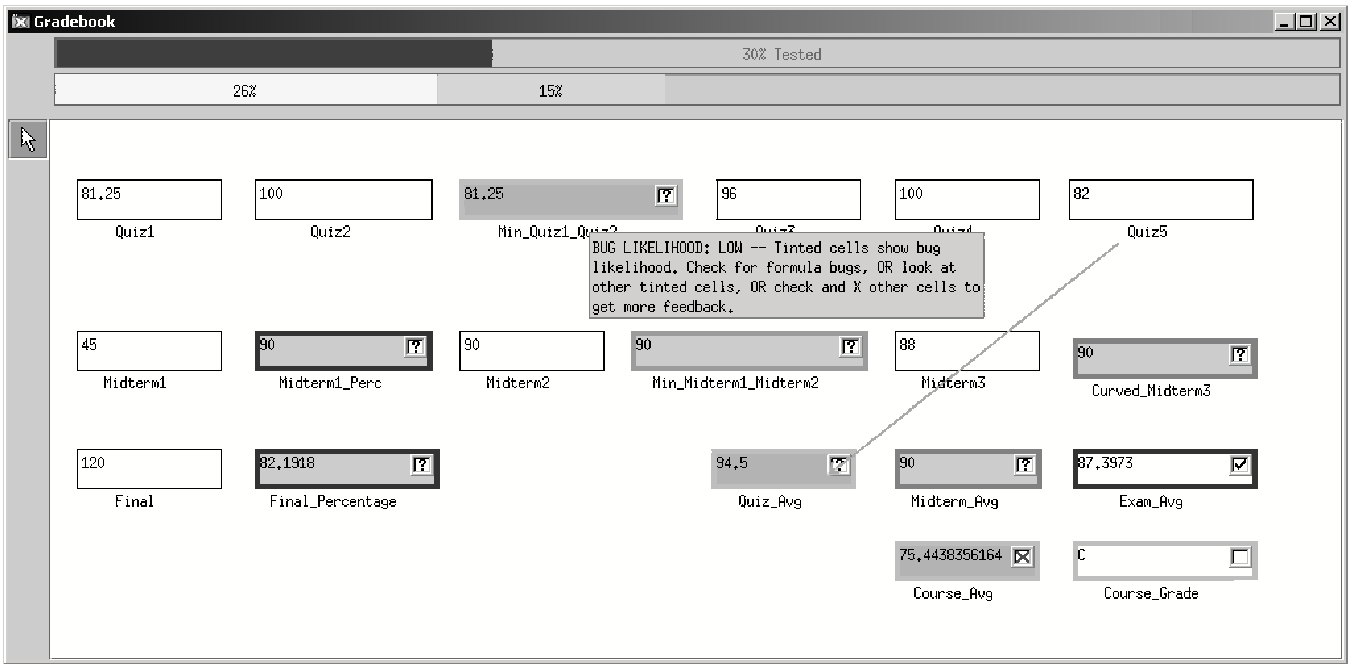


Figure 2. A snapshot of the prototype software. The user noticed an incorrect value in Course_Avg and places an Xmark. As a result, eight cells are highlighted as being possible sources for the incorrect value, with some deemed more likely (shaded darker) than others.

tions available and Table 1 shows a set of commonly used actions and their meanings. The log files contain detailed information about every user action, including a time stamp for when it was taken, on which cell it operated, and various related parameters. Figure 3 shows an excerpt of a log file. Here we only show the time stamp, the name of the action and the cell ID, omitting other information due to space limitations.

The Gender HCI project aims to find different user strategies and understand how they relate to gender and problem-solving success. Such understanding is aimed at ultimately improving software design to encourage successful strategies by users of both genders. The HCI log data collected in the Gender HCI studies is usually manually analyzed by the HCI researchers to identify interesting general behavioral trends that correspond to high level strategies. This method is necessarily somewhat restrictive, because humans have limited ability to process data of large volume and high dimension. Furthermore, bias can be introduced in the analysis due to preconcieved expectations.

This case study applies data mining to an independent, ongoing project, which is a real world application in that its processes, data collection, specifications, and goals were all established by HCI researchers, independently of any data mining considerations and without regard to data mining suitability. We use this ongoing HCI project to consider how to mine and interpret the HCI log data of human behaviors. In this paper, we focused on one particular log data set that was collected from 39 user-study participants performing a given spreadsheet debugging task. On average, the log file of each participant contained over 400 actions.

The goals of this case study were: 1) to automatically extract high level user strategies from the log data to remove human-related limitations, such that the result yields better understanding of user behavior; and 2) to examine the applicability of data mining to this challenging problem, with a special focus on the interpretability of the mined results.

3 Mining Sequential Patterns

To find strategies from the data, we need to first decide what constitutes a strategy. Typically a strategy refers to a reasoned plan for achieving a specific goal. Here we consider behavior as a surrogate for strategy. That is, we consider sequences of actions that collectively achieve a specific goal (such as deciding if a particular value is correct or not) to be evidence of an underlying strategy.Such considerations naturally led us to cast this as a sequential pattern mining problem [1, 11].

Below we describe the methods that we adopted from existing research to preprocess and mine the HCI log data. As the results below illustrate, they were not able to perform

satisfactorily on the HCI log data of human behaviors.

Preprocessing Recall that the log files contain detailed contextual information about each action the users took. In our preprocessing step all of the contextual information was removed and only the action names were retained to form the basic event sequences. This allowed us to detect behavioral trends that are general and not restricted to particular cells. For example, the log excerpt in Figure 3 translates into the simple sequence of events: (Tooltip, Checkmark, CheckMark).

Mining Sequential Patterns Sequential Pattern Mining is a general problem first introduced in the context of retail data analysis [1] and network alarm pattern analysis[11, 13]. Over the years, many different sequential pattern mining algorithms have been developed for different types of sequential data. From these techniques, we chose IPM2 [9], a method developed for mining interaction patterns, because our HCI log data share similar charateristics with the interaction trace data targeted by IPM2. Note that IPM2 may not necessarily be the best method for this task, but its technique is representative of many related methods and appears to be appropriate for our data type.

In particular, given a set of event (action) sequences, IPM2 incrementally searches for fully ordered action sequences that satisfy some user specified maximum error and minimum support criteria. The minimum support criterion specifies the minimum number of times an action sequence has to be observed in the log files for it to be considered frequent. The maximum error criterion specifies the maximum number of insertion errors allowed for pattern matching. For example, a pattern $\langle$A, B, C$\rangle$ is only considered to be present in sequence [A, E, D, B, C] if the maximum error criterion is set to 2 or larger.

In our experiments, we set the minimum support threshold to be 30, which requires a pattern to be observed at least 30 times to be considered frequent. The maximum error threshold was set to 1 to allow a single insertion. This threshold was chosen to allow some flexibility in pattern finding. Note that allowing more insertion errors can result in exponentially more patterns to be considered frequent. This is because a pattern can match a number of sequences that is exponential in the number of errors, allowing for arbitrary sequences to appear frequent.

We further limited the pattern mining algorithm to output only those patterns that were no shorter than 5 actions. This limit was set to ensure that the output patterns would be sufficiently long to provide contextual information needed to interpret the patterns. Finally, we removed those patterns that were not maximal (patterns that do not have a frequent superpattern).

Table 2. A random subsample of the found patterns. See Table 1 for action meanings.

PID	Pattern
P58	HF, CM, CM, CM, PF, HF
P149	PF, HF, CM, CM, CM, PF
P179	AON, AOF, PF, HF, PF, HF
P206	HF, CM, CM, PF, HF, PF
P273	HF, PF, EF, HF, PF, EF, HF

We applied the above mentioned procedure to the HCI log data and found a total of 289 maximal patterns of length 5 or longer. In Table 2, we show five randomly selected patterns from these 289 patterns.

Examining these patterns individually, we made the following observations.

1. There are many highly similar patterns.

   For example, P58 and P149 differ only by two actions. Note that there is no super-pattern or sub-pattern relationship between these two patterns, therefore concepts such as maximal [10] and closed [15] patterns do not provide further pruning. A key question then is whether such patterns should be considered to be equivalent. In particular, we would like to know whether they are used for the same purpose. In reality, the same strategy may result in different action sequences due to random variations among users. If we do consider P58 and P149 to represent the same general behavior, how about P206? It differs from P58 by only two actions as well. We need a principled way to address issues like this.

2. Individual patterns carry limited information.

   For instance, P179 describes the behavior of toggling on an arrow closely followed by toggling off an arrow, followed by some open- and close-cell operations. What does this specific sequence of actions tell us about the user's behavior? Hardly anything. It is difficult to reach any general understanding of user behavior from a single pattern like this. Again, we need a principled way to help us go beyond the specifics of any individual pattern and detect general trends.

The above observations led us to investigate possible ways of clustering patterns into meaningful groups. By doing so we can collectively interpret a group of similar patterns and detect from them the general behavioral trends that correspond to high level strategies.

4 Pattern Interpretation

The frequent pattern mining community has long recognized that pattern interpretability (or lack thereof) is a major bottleneck when applying pattern finding algorithms. Standard algorithms can output hundreds or thousands of patterns, prohibiting their detailed examination. Concepts such as maximal patterns [10] and closed patterns [15, 21] have been introduced to reduce pattern redundancy. However, the quantity of the patterns is only one part of the story. In many applications, individual patterns often carry limited information about the general phenomenon. For example, in gene analysis for identifying transcription factor binding sites, the same transcription factor may bind with seemingly different base sequences; two separate base sequences may jointly determine the behavior of the gene. Simply removing redundant patterns will not help in such situations. We need to extract general phenomena, whereas individual patterns are often single instantiations of such phenomena.

More recently, new techniques have emerged to compress the found patterns [19], to group the patterns to find representative ones [20], to rank and select the top-k patterns according to their significance and redundancy [18], and to provide semantic annotations of the patterns using limited contextual information [14]. We consider such techniques to be more appropriate for dealing with the above mentioned problems. Still, these techniques are designed for frequent item set patterns. In this study, we adapted the basic ideas behind these methods to apply them to the sequential pattern interpretation problem. In essence, we sought to cluster the patterns such that the patterns in each group could collectively provide some high level understanding of user strategies. Toward this goal, we examined different ways to group the 289 sequential patterns and evaluate their results based on interpretability. Below we describe the different approaches for clustering patterns into strategy-corresponding groups.

4.1 Supervised clustering of patterns

An important aspect of our data mining goal was to understand the relationship between the strategies we find and gender as well as problem-solving success. In other words, we were interested in *identifying strategies that are favored by certain user groups: in particular, female users vs. male users, and successful users vs. unsuccessful users*. One possible approach to achieving this special goal is to use supervised clustering [7, 16].

Supervised clustering include additional supervised information (such as class labels) into the clustering procedure to produce clusters that distinguish among different classes. Successful applications of supervised clustering include learning word clusters that are indiciative of document classes [16, 8] and extracting gene groups that distinguish different tissue types [7].

To apply supervised clustering, we collected for each pattern the number of times each user used it. This gave us a 39 dimensional representation of each pattern describing its usage frequency among all users. Each user was then assigned a class label. For gender analysis, we assign the class labels to be female or male, based on their background information. For success analysis, we assign the users to be successful or unsuccessful depending on the number of bugs they fixed at the end of their sessions. The supervised clustering technique introduced in [8] was then applied to find pattern groups that differentiate female users from male users or successful users from unsuccesful users respectively.

4.2 Unsupervised Clustering of Patterns

For unsupervised pattern clustering, a critical question is how to best capture the similarity among patterns. It is important to realize that there may exist many different ways for the action sequences of the same strategy to differ from or resemble one another. It is thus unlikely that one can design a single similarity measure that will capture all different possibilities. In fact, there is no reason to limit ourselves to one particular similarity measure. Different measures may reveal different underlying connections among patterns. Following this philosophy, we examined three different ways to capture the similarity among patterns.

Pattern clustering based on edit distance. In this approach, we consider the syntactic similarity among patterns. Note that patterns of similar action sequences are deemed to represent the same general behavior, only perturbed by limited amounts of extraneous and irrelavant actions. Such syntactic similarity can be captured by the edit distance measure between the two patterns, which is defined as the minimum number of action insertions, deletions or substitutions required to match one pattern with another.

We computed the pairwise edit distance measure among all 289 patterns, producing a 289×289 distance matrix. We then applied a hierarchical average link clustering algorithm to produce a dendrogram representing a hierarchy of clustering solutions. Visually inspecting the dendrogram, we decided to cluster the patterns into 37 groups. In the remaining part of the paper, we will refer to this method as the *edit distance method* for pattern clustering.

Pattern clustering based on usage profiles. Another way of judging the connection between a pair of patterns is to look into how they are used. In particular, in this approach, we created a usage profile for each pattern by looking at how frequently each pattern was used by the 39 users.

Patterns sharing similar usage profiles were then considered to be related to each other.

More specifically, similar to the supervised case, we created a 39 dimensional usage profile to represent each pattern. Each dimension is simply the number of times that the pattern was used by a particular user. We then applied K-means to the resulting 39 dimensional data set to group patterns that share similar usage profiles together. Visually inspecting the plot of the GAP statistics [17], we found 20 clusters in the data. We will refer to this method as the *usage profile method*. Note that if two patterns A and B are grouped together under the usage profile method, it suggests that users who use A a lot tend to use B a lot as well or vice versa.

Pattern clustering based on cell frequency. Finally, we looked into another aspect concerning how patterns were used. In this case, we inspected the cells that each pattern operated on. In particular, given a pattern we looked at each time that it was used, and found the cells that it operated on. For instance, if a pattern consists of five actions, every time we observed this pattern in action, the counts of the five cells that it operated on would be incremented accordingly. If a cell was operated on multiple times within these five actions, its count will be incremented multiple times. In the end, we obtained a cell frequency distribution for each pattern describing how many times this pattern operated on every cell of the spreadsheet. In total, the spreadsheet contains 25 cells. This results in a 25 dimensional representation of the patterns.

Similarly, we applied K-means to the 25 dimensional data and found 20 clusters. Note that if two patterns A and B are grouped together under this method, it suggests that cells that are touched frequently by A are also touched frequently by B and vice versa.

4.3 Statistical Testing

Having found a set of pattern groups, note that not all pattern groups necessarily correspond to interesting user strategies. To find those that are interesting to our application, i.e., the gender HCI study, we would like to relate these pattern groups to user gender and problem solving sucess. In this study, we used the two-sample unpaired t-tests [6] to identify a subset of pattern groups (stretageis) whose usages showed statistically significant differences between female and male users, and/or between successful and unsuccessful users.

Taking gender analysis as an example, we separated the users according to their gender. Given a particular pattern group in consideration, we counted how many times each user uses the patterns from that group. This gave us a count for each user. We considered the counts of the female users as one sample X (the size of the sample equals the number of female users), and the counts of the male users as another sample Y. The unpaired t-test determines whether X and Y could have the same mean assuming they are generated both from normal distributions that share the same variance. If we fail to reject the null hypothesis (X and Y have the same mean), then we considered this pattern group to be uninteresting for our study because it showed no statistically significant difference between males and females.

We tested each pattern group with respect to both gender and success and selected only those groups that are significant according to our tests for further inspection and interpretation. This allowed us to quickly zoom in to the pattern groups that are interesting to the gender HCI research.

It is important to note that pattern groups found by supervised clustering should not be tested this way due to the bias introduced in the clustering step. Because it intentionally searches for patterns to group together to achieve distinctions between males and females or between succcessful and unsuccssful users, pattern groups found this way will likely be judged as significant by statistical tests but such significance results should be disregarded.

5 Results

In this section, we present the final results of our analysis. We examine the results from both supervised and unsupervised clustering based on the interpretability of the resulting pattern groups.

5.1 Pattern Interpretation Results with Supervised Clustering

We applied the information theoretic technique developed in [8] for supervised clustering. As we mentioned in Section 4.1, we represented each pattern by its usage profile over all 39 users. Supervised clustering was performed in two different ways. In the first case, users were classified into female or male. In the second case, users were classified as successful or unsuccessful. In both cases, we clustered the patterns into 20 groups[1].

Our results for supervised clustering were disappointing — examining the resulting clusters did not reveal any general trends. The clusters appeared to contain a set of random patterns that did not seem to relate to one another. This is possibly due to the fact that supervised clustering is geared toward correctly classifying users rather than forming coherent clusters.

Also note that as mentioned in Section 4.3, because the supervised information was introduced in the clustering

[1]Varying the cluster number did not produce any noticable difference in the quality of the resulting clusters.

Table 3. A summary of the pattern groups by unsupervised clustering methods.

Method	Group	Representative Patterns	Statistical Testing Results
Edit Dist.	1	⟨ HF,PF,HF,*CM*,*CM*,*CM*,*CM*,*CM* ⟩ ⟨ PF,*CM*,*CM*,*CM*,*CM*,*CM* ⟩ ⟨ PF,HF,*CM*,*CM*,*CM*,*CM*,*CM*,*CM* ⟩	Significant differences between successful and unsuccessful users (p-value = 0.032 and 0.003 respectively) Favored by *successful users*
Edit Dist.	2	⟨ *CM*,*CM*,*CM*,*CM*,*CM*,PF,HF ⟩ ⟨ *CM*,*CM*,*CM*,*XM*,*XM* ⟩ ⟨ *CM*,*CM*,*CM*,*CM*, HF ⟩	
Edit Dist. Cell Freq.	3	⟨ HF,HF,PF,HF,PF,*EF*,HF ⟩ ⟨ HF,PF,HF,PF,PF,*EF*,HF ⟩ ⟨ HF,*EF*,HF,PF,*EF* ⟩	Significant difference between female and male users. (p-value=0.016) Favored by *female users*
Usage Freq. Cell Freq.	4	⟨ HF,PF,HF,PF,HF,HF,*CM* ⟩ ⟨ PF,PF,HF,PF,HF,*CM*,PF ⟩ ⟨ HF,PF,HF,PF,HF,PF,HF,PF,HF,*CM* ⟩	Significant difference between successful and unsuccessful users (p-value=0.017) Favored by *unsuccessful users*
Usage Freq. Cell Freq.	5	⟨ *EV*,HF,PF,*EV*,HF,*CM*,*CM*,*CM* ⟩ ⟨ PF,*EV*,HF,PF,*EV*,*CM* ⟩ ⟨ HF,PF,*EV*,HF,*CM*,*CM*,*CM*,*CM* ⟩ ⟨ CM,CM,CM,XM,XM ⟩	Significant difference between successful and unsuccessful users (p-value=0.007) Favored by *successful users*

process, it is not statistically sound to examine the statistical differences between how female and male (or successful and unsccessful) users use a particular pattern group.

In summary, while supervised clustering has been shown to be effective in generating clusters for the purpose of classification, it is not an appropriate approach for forming meaningful pattern groups that describe some general behaviors. It also introduces bias into the clusters because extra supervised information was used — making further statistical testing of these clusters inappropriate.

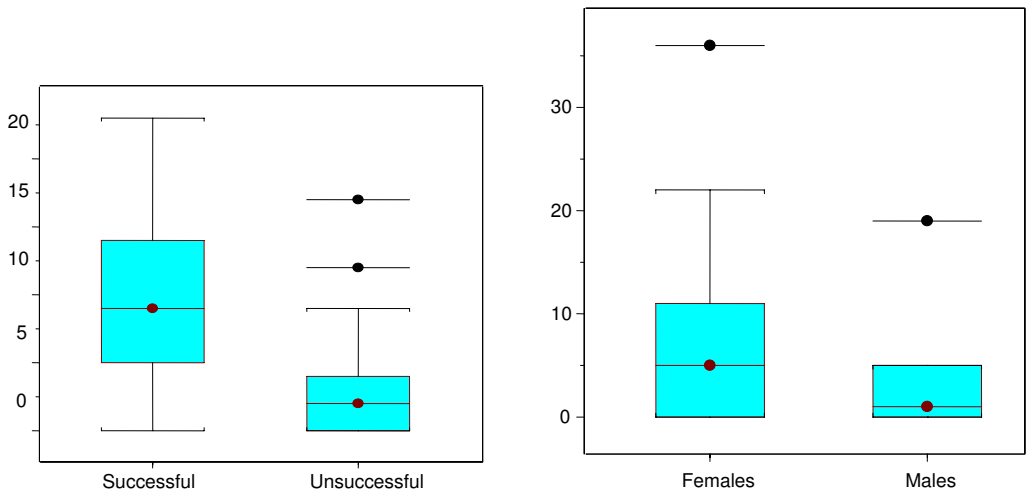
5.2 Pattern Interpretation Results with Unsupervised Clustering

Our unsupervised clustering methods produced a number of highly interesting clusters, which collectively led to insights about user strategies, relating both to user gender and to problem-solving success. In this section, we will highlight some of the most interesting findings that we obtained using unsupervised clustering to interpret the 289 patterns.

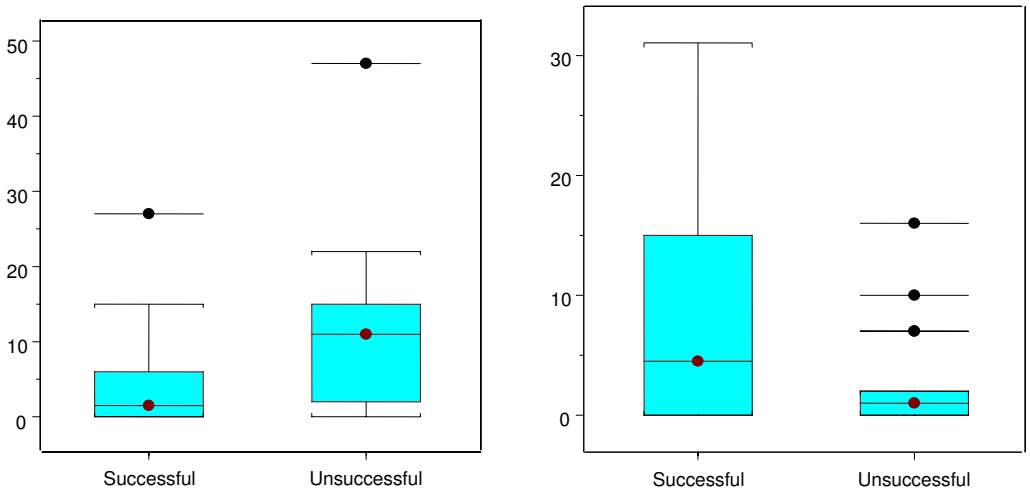
Consider the following pattern groups discovered by our unsupervised approaches.

1. Pattern Groups 1 & 2: Pattern groups 1 and 2 were both identified by the edit distance approach. We combine the discussion of these two groups together because the patterns in these two groups are similar. In particular, they can all be characterized by the behavior of *consecutively* checking off cells as being correct (CM) or incorrect (XM), i.e., a "batch" of checks made in a row (termed here the "batch-checking" strategy). As indicated in column 4 of Table 5, the statistical tests indicate both pattern groups showing a significant difference between the successful and unsuccessful user groups, with the batch-checking strategy used more by successful users. See Figure 5.2(a) for the box-plot of the group 2 usage frequencies by the successful and unsuccessful users respectively. The box-plot for group 1 is highly similar, thus omitted.

2. Pattern Group 3: Pattern group 3 was identified by the edit distance method as well as the cell frequency method. This strongly suggests that this cluster is real and not a random artifact created by the clustering algorithms. Patterns in this group are characterized by *inspecting* formula cells - PostFormula(PF) and HideFormula(HF) - followed by one or more EditFormula (EF) operations. We further inspected the cells that these patterns operate on, and found that 98% of the cells touched by these patterns are formula cells (i.e., cells that contain formulas) as opposed to value cells (i.e., cells that contain only a constant value). This suggests a strategy we call "code inspection", which involves opening and closing formula cells to inspect the code statically and making formula changes based



(a) Pattern Group 2 (b) Pattern Group 3



(c) Pattern Group 4 (d) Pattern Group 5

Figure 4. The usage frequency box-plots for different pattern groups and user groups.

on the inspection results. Interestingly, unbeknownst to us, in an independent user study [3] in which the participants were asked to describe their strategies for debugging, "code inspection" was one of the top strategies described by female participants, but not by the males. This independent finding provides further evidence of the validitity of the cluster.

3. Pattern Group 4: Pattern group 4 was again identified by two methods - the usage frequency method and the cell frequency method. The patterns in this group differ subtly from the patterns of group 3. In particular, these patterns also perform a number of formula manipulations (e.g. PF, HF). However, these manipulations were followed by one or more CheckMark (CM) operations, as opposed to EditFormula (EF) operations. This distinction is important. In fact, this group of patterns suggest a different strategy we named "to-check-list behavior", which involves visually inspecting the formula cells and then making a mark on the cells to indicate they are off the "to-check-list". An external data point regarding this cluster's validity is that this "to-check-list" strategy was explicitly mentioned by several participants in the independent user study. (Again, this information was not available to us during our analysis.) Statistical testing shows that this pattern group was used more frequently by the unsuccessful users, as indicated by column 4 of Table 5 and Figure 5.2(c).

4. Pattern Group 5: This group was again identified by two methods - the usage profile method and the cell

frequency method. The patterns in this group describe the behavior of *testing* formulas by varying the input values. (Note that testing is different from code inspection — in the former, the user evaluates values and in the latter the user evaluates the source code directly.) The testing nature of this pattern is suggested by the repeated EditValue (EV) operations accompanied by a set of CheckMark (CM) operations. We refer to this strategy as the "test-and-check" strategy. (In the independent user study, many participants explicitly described testing as a strategy.) Statistical testing indicates that it was favored by the successful users (See Table 5 and Figure 5.2(d)). Comparing this with the "to-check-list behavior", it suggests that when the Checkmark is correctly used as a marking for testing results, users see more success. This is consistent with previous HCI findings tying use of the CheckMark with successfully testing and debugging spreadsheet formulas [5].

Note that we also see some of the "batch-checking" (group 1 & 2) patterns appearing in this group. Recall that if patterns A and B are grouped together by the usage profile method, it suggests users who use pattern A a lot tend to use pattern B a lot as well and vice versa. This indicates that the batch-checking behavior is often used in combination with the "test-and-check" strategy. This provides a possible explanation as to why the batch-checking behavior is related to debugging success.

To summarize, unsupervised clustering significantly improved the interpretatiblity over supervised clustering. The resulting pattern groups revealed evidence of four different high-level strategies. There are three main points to note.

First, the match of the verbalizations in an independent user study strongly suggest that the findings of our interpretation method are not only real but also are at an appropriate level of abstraction.

Second, one of the results, namely the code inspection result (Pattern Group 3), was not yet proven. HCI researchers had begun to suspect its presence, but they had not been able to statistically show this phenomenon in more than three years of manual empirical work in the context of of gender HCI [2].

Third, two of the results are new, namely the beneficial effects of batch checking (Pattern Groups 1 and 2) and the detrimental effects of using the debugging features (CheckMarks and XMarks) for to-do list purposes (Pattern Group 5). These results had not been revealed in more than nine years of manual empirical work studying uses of these features as problem-solving devices [5].

6 Conclusion

In this paper, we described a complete data mining process applied to a set of Human Computer Interaction log data. Our goal is to identify general user strategies that are interpretable. We applied frequent sequential pattern mining as our initial step toward this task, which produced a significant number of patterns that are difficult to interpret and lack generality. This led us to explore a number of different ways to summarize/generalize beyond individual patterns that we found, including both supervised and unsupervised pattern clustering approaches. The unsupervised approaches, followed by statistical testing, successfully identified some highly interesting pattern groups that corresponded well to some strategies that have been identified by the users themselves when being asked in a separate user study[2].

As a case study, our practice led to the following understanding about applying frequent pattern mining to extract interpretable general trends from data.

- Individual patterns found by standard algorithms are difficult to interpret and they carry limited information about the general trend. This is because an individual pattern is often just one instance of a general phenomena. To understand the general trend, it often requires seeing many instances to capture what is general and go beyond the specifics of individual patterns. This suggests that, when appropriately done, grouping patterns into meaningful groups can increase the interpretability and the generality of the findings.

- To group patterns appropriately, special care must be taken to avoid introducing bias into the grouping, which is exactly what happened when we applied supervised clustering to group the patterns. Although our goal is to identify groups of patterns that are favored by female users (versus male users) or successful users (versus unsuccessful users), we are not interested in classifying users. While supervised clustering has been shown to be effective at producing good classification, it led to incoherent pattern groups that are not interpretable as general strategies.

- For unsupervised pattern clustering, there often exists a variety of contextual information that can be helpful in discerning the general trend behind a set of patterns. Using one type of contextual information (or criterion function) for clustering the patterns should not exclude the possibility of using other information for

[2]Note that this separate user study was conducted completely in parallel with our work and only after the fact did we realize that we reached a set of consistent findings.

clustering as well. We recommend leveraging different ways to group the patterns because of the following potential benefits. First, often times different methods of grouping reach consensus about some clusters, providing strong support to the validity of the results. Second, different groupings collectively may reveal insights that not available from any single grouping. This suggests that an important research direction is to develop automated or semi-automated approaches to producing a diversity of low-level pattern groupings that are potentially of interest.

For future work, we would like enrich our general framework by considering a much richer representation for the basic patterns. Current the basic patterns are simple sequential patterns that lack of ability to capture contextual information about the current state of the user and the system. We will consider using relational representations such as first-order Horn rules [12] to represent the basic patterns. We believe the interpretability challenge will remain and will apply the philosophy that we developed in this paper to summarize/generalize beyond individual rules.

References

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. of the 11th Int. Conf. on Data Engineering*, pages 3–14, 1995.

[2] L. Beckwith, M. Burnett, V. Grigoreanu, and S. Wiedenbeck. Gender hci: What about the software? *Computer*, pages 83–87, 2006.

[3] L. Beckwith, V. Grigoreanu, N. Subrahmaniyan, S. Wiedenbeck, M. Burnett, C. Cook, K. Bucht, and R. Drummond. Gender differences in end-user debugging strategies. Technical Report CS07-60-01, Oregon State University, 2007.

[4] M. Burnett, J. Atwood, R. Djang, H. Gottfried, J. Reichwein, and S. Yang. Forms/3: A first-order visual language to explore the boundaries of the spreadsheet paradigm. *Journal of Functional Programming*, 11:155–206, 2001.

[5] M. Burnett, C. Cook, and G. Rothermel. End-user software engineering. *Communications of the ACM*, pages 53–58, 2004.

[6] G. Casella and R. L. Berger. *Statistical inference*. Duxbury Press, 1990.

[7] M. Dettling and P. Buehlmann. Supervised clustering of genes. *Genome Biology*, 3, 2002.

[8] I. S. Dhillon, S. Mallela, and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classication. *Journal of Machine Learning Research(JMLR): Special Issue on Variable and Feature Selection*, 2003.

[9] M. El-Ramly, E. Stroulia, and P. Sorenson. Interaction-pattern mining: Extracting usage scenarios from run-time behavior traces. In *Proc. of the 8th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 2002)*, 2002.

[10] K. Gouda and M. Zaki. Efficiently mining maximal frequent itemsets. In *Proc. of Int. Conf. on Data Mining*, 2001.

[11] K. Hatonen, M. Klemettinen, P. Ronkainen, and H. Toivonen. Knowledge discovery from telecommunication network alarm data bases. In *Proc. of 12th Int. Conf. Data Engineering*, pages 115–122, 1996.

[12] R. Khardon. Learning action strategies for planning domains. *Artificial Intelligence*, 1999.

[13] H. Mannila, H. Toivonen, and V. A. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, pages 259–289, 1997.

[14] Q. Mei, D. Xin, H. Cheng, J. Han, and C. Zhai. Generating semantic annotations for frequent patterns with context analysis. In *Proc. of the 12th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'06)*, 2006.

[15] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *Proc. of the 7th Int. Conf. on Database Theory*, 1999.

[16] N. Slonim and N. Tishby. The power of word clusters for text classification. In *Proc. of the 23rd European Colloquium on Information Retrieval Research*, 2001.

[17] B. T. T. Hastie and J. Friedman. *Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer-Verlag, 2001.

[18] D. Xin, H. Cheng, X. Yan, and J. Han. Extracting redundancy-aware top-k patterns. In *Proc. of the 12th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'06)*, 2006.

[19] D. Xin, J. Han, X. Yan, and H. Cheng. Mining compressed frequent-pattern sets. In *Proc. of Int. Conf. on Very Large Data Bases*, 2005.

[20] X. Yan, H. Cheng, J. Han, and D. Xin. Summarizing itemset patterns: A profile-based approach. In *Proc. of 11th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2005.

[21] X. Yan, J. Han, and R. Afshar. Clospan: Mining closed sequential patterns in large datasets. In *Proc. of the 3rd SIAM International Conference on Data Mining*, 2003.