Mining Interpretable Human Strategies: A Case Study

Xiaoli Z. 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 human strategies by observing their actions. Our application domain is an HCI study aimed at discovering general strategies used by software users and understanding how such strategies relate to gender and success. We cast this 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 linked to user success and had not been revealed in more than nine years of manual empirical work. As a case study, our work highlights important research directions for making data mining more accessible to non-experts.

1. Introduction

How can data mining be applied to better understand human behaviors? To understand how humans interact with computers, 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 at supporting different users in achieving their goals. In this paper, we applied data mining to a set of HCI log data collected in a particular problem-solving setting, namely users debugging spreadsheet formulas. We had 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.

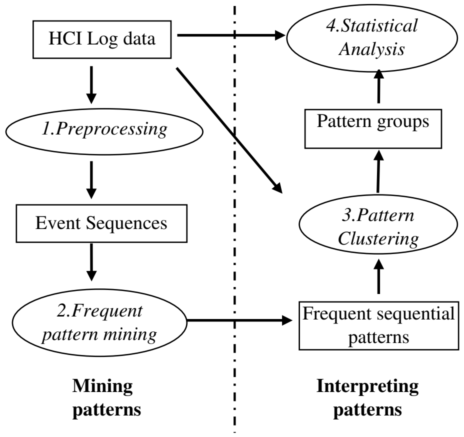


Figure 1. Our data mining process. Arrows represent the information flow.

Figure 1 summarizes our data mining process, which has two major parts. The first part finds basic behavioral patterns from the data. The second part interprets these patterns, extracts general strategies and relates them to gender and problem-solving success. Using this framework we discovered 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 verified by HCI researchers in more than three years of manual empirical work. 3) Discovering two new strategic patterns that are linked to user success and had not been revealed in more than nine years of manual empirical work.

While our application was successful, a significant amount of effort and data mining expertise was required. In particular, the existing data mining tools would not have

1550-4786/07 $25.00 © 2007 IEEE
DOI 10.1109/ICDM.2007.19

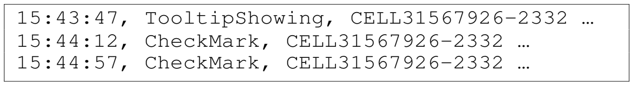
Authorized licensed use limited to: OREGON STATE UNIV. Downloaded on October 23, 2008 at 15:49 from IEEE Xplore. Restrictions apply.

been sufficient for HCI researchers, without data mining expertise, to have made our discoveries. In this respect, our work highlighted an important research direction for making data mining tools more useful to the data-mining novices. Key to our success was the use 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, selecting the grouping mechanisms was largely human-directed and quite tedious. This suggests that automated techniques for generating diverse and potentially interesting groupings of low-level patterns is a key direction toward making data mining more accessible and easier to apply.

This paper makes following contributions. First, we applied data mining to a challenging problem - identifying and understanding human strategies from noisy HCI log data. Second, its primary focus 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 real world applications and suggest important research directions in data mining.

2 Case Study Setting

Our case study is situated in an HCI research project termed the "Gender HCI" project [2]. For this project, HCI researchers have conducted extensive empirical user studies to collect data about user activity when using problem-solving software. The collected data has been and is continuingly being manually analyzed by HCI researchers.



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

Figure 2. An excerpt from the log file.

Independently from their efforts, we applied data mining to this real world application, whose 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 used this project to examine how to mine and interpret the HCI log data of human behaviors.

In particular, we focused on a research prototype extension of spreadsheets [4, 5], which is designed to aid users in debugging spreadsheets. It provides functionalities for systematically testing a spreadsheet and giving feedback to help users identify the bugs. Please refer to [9] for more information. The software has been instrumented to record user actions into log files. A user action is defined as a user's physical interaction with a debugging feature. In total, there are 19 actions available and Table 1 shows a set of commonly used actions and their meanings. The log files contain details about every user action, including a time stamp, on which cell it operated, and various related parameters. Figure 2 shows an excerpt of a log file, showing the time stamp, the action name and the cell ID. We omit other information due to space limit. The data set used in this paper 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.

Table 1. Commonly used actions

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

3 Mining Sequential Patterns

Typically a strategy refers to a reasoned plan for achieving a specific goal. Here we considered behavior as a surrogate for strategy. That is, we considered sequences of actions that collectively achieve a specific goal to be evidence of an underlying strategy. This led us to cast strategy mining as a sequential pattern mining problem [1, 11]. Below we describe our preprocess and pattern mining steps.

Preprocessing The log files contain detailed contextual information about each user action. We removed all contextual information and retained only the action names to form the action sequences. This allowed us to detect general behavioral trends not restricted to particular cells.

Mining Sequential Patterns Sequential Pattern Mining was first introduced in the context of retail data analysis [1]. Many different algorithms have been developed for various types of sequential data. From these techniques, we chose IPM2 [8], a method developed for mining interaction patterns, because our HCI log data shares similar characteristics with the interaction trace data targeted by IPM2.

In particular, given a set of action sequences, IPM2 incrementally searches for fully ordered action sequences that satisfy pre-specified max-error and min-support criteria. The min-support criterion specifies the minimum num-

Authorized licensed use limited to: OREGON STATE UNIV. Downloaded on October 23, 2008 at 15:49 from IEEE Xplore. Restrictions apply.

ber of times a pattern has to be observed to be considered frequent. The max-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 maxi-error ≥ 2.

We set min-support to 30. The max-error threshold was set to 1 to allow a single insertion. This threshold was chosen to allow some flexibility in pattern finding. We further removed the patterns shorter than 5 actions to ensure that the output patterns are sufficiently long to provide enough information for interpreting the patterns. Finally, we removed those patterns that were not maximal [10].

Table 2. A sample of the found patterns

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 procedure to the HCI log data and found 289 patterns. Table 2 shows five representative patterns from these 289 patterns. Examining these patterns individually, we made the following observations.

First, there are many highly similar patterns. For example, P58 and P149 differ only by two actions. Because there is no super-pattern or sub-pattern relationship between them, concepts such as maximal [10] and closed [14] patterns do not provide further pruning. A key question is whether such patterns should be considered equivalent. In particular, we want to know whether they were used for similar purposes. In reality, the same strategy may result in different action sequences due to random variations among users. If we do consider P58 and P149 equivalent, how about P206? It differs from P58 by only two actions as well. We need a principled way to address this issue.

Second, 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 tell us about the user's behavior? Hardly anything. It is difficult to reach a general understanding of user behavior from a single pattern like this. We need to go beyond the specifics of individual patterns and detect general trends.

These observations led us to investigate possible ways of clustering patterns into meaningful groups, which can be collectively interpreted to reveal 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 in application. Standard algorithms output large numbers of patterns, prohibiting their detailed examination. Concepts such as maximal [10] and closed patterns [14, 20] can reduce pattern redundancy. However, the quantity is only one part of the story. Many applications need to extract general phenomena, whereas individual patterns are often single instances of such phenomena. Recently, new techniques have emerged to address the interpretability issues by compressing, grouping and summarizing the found patterns [18, 19, 17, 13]. We consider such techniques more appropriate for dealing with our problems. Still, they are designed for frequent item set patterns. We adapted the basic ideas behind these methods and applied them to sequential patterns. In essence, we saught to cluster the patterns into groups such that each group collectively provides some high level understanding of user strategies. Below we present how we achieved this goal using unsupervised clustering. [1]

4.1 Unsupervised Clustering of Patterns

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

Pattern clustering based on edit distance. This approach considers the syntactic similarity among patterns. Here patterns of similar action sequences are deemed to represent the same general behavior, only perturbed by limited amounts of extraneous and irrelevant actions. Such syntactic similarity can be captured by edit distance. We computed the pairwise edit distance among all 289 patterns, producing a 289 × 289 distance matrix. We then applied hierarchical average link clustering 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 remainder 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. Specifically, we created a 39 dimensional usage profile to represent each pattern. Each di-

[1] We also examined supervised clustering [7] for finding groups of patterns that were used differently by different user groups. Please refer to our technical report [9] for details on this.

Authorized licensed use limited to: OREGON STATE UNIV. Downloaded on October 23, 2008 at 15:49 from IEEE Xplore. Restrictions apply.

Table 3. A summary of the pattern groups.

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*

mension is 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 [16], 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 and vice versa.

Pattern clustering based on cell frequency. Finally, we looked into another aspect concerning how patterns were used. Here we inspected the cells on which each pattern operated. In particular, given a pattern we looked at each time that it was used, and found the cells on which it operated. For instance, if a pattern consists of five actions, every time we observed this pattern, the counts of the five cells on which it operated were incremented accordingly. If a cell was operated on multiple times within the pattern, its count was incremented multiple times. In the end, we obtained a cell frequency distribution for each pattern describing how many times the 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.

4.2 Statistical Testing

Not all pattern groups necessarily correspond to interesting user strategies. To find those interesting to our goal, we related these pattern groups to user gender and problem solving success. We used the unpaired t-test [6] to identify a subset of pattern groups 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 based on their gender. Given a pattern group in consideration, we counted how many times each user uses the patterns from that group, giving a count for each user. We considered the counts of the females as one sample X (the size of the sample equals the number of female users), and the counts of the males as another sample Y. Applying unpaired t-test at 5% significance level, if we fail to reject the null hypothesis (X and Y have the same mean), the pattern group is deemed uninteresting because it showed no statistically significant difference between males and females.

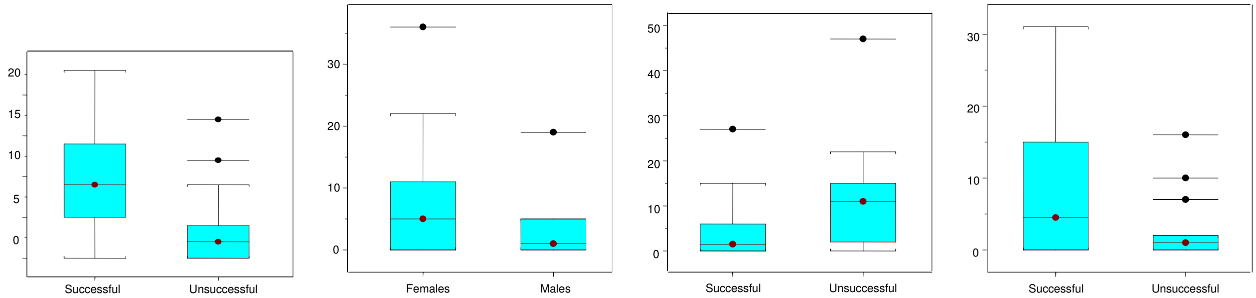
We tested each pattern group with respect to both gender and success and selected only those that are significant according to our tests for further inspection and interpretation. This allowed us to quickly zoom in to the pattern groups that are interesting to the gender HCI research.

5 Pattern Interpretation Results

Our 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. Table 3 summarizes some of the most interesting groups we found.

Pattern Groups 1 & 2: Pattern groups 1 and 2 were both identified by the edit distance approach. We discuss them together because the patterns in these two groups are similar. In particular, they can all be characterized by the behav-

Authorized licensed use limited to: OREGON STATE UNIV. Downloaded on October 23, 2008 at 15:49 from IEEE Xplore. Restrictions apply.



(a) Pattern Group 2 (b) Pattern Group 3 (c) Pattern Group 4 (d) Pattern Group 5

Figure 3. The usage frequency box-plots for different pattern groups and user groups.

ior 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 3, 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 3(a) for the box-plot of the group 2 usage frequencies by the successful and unsuccessful users respectively. The group 1 plot is highly similar, thus omitted.

Pattern Group 3: Pattern group 3 was identified by both the edit distance method and the cell frequency methods. This 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 them are formula cells (i.e., cells containing formulas) as opposed to value cells (i.e., cells containing constant values). 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 on the inspection results. Interestingly, in an independent user study [3] in which the participants were asked to describe their debugging strategies, "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 validity of the cluster.

Pattern Group 4: Pattern group 4 was identified by both the usage profile and the cell frequency methods. 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. (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 3 and Figure 3(c).

Pattern Group 5: This group was again identified by both the usage profile and the cell frequency method. Patterns in this group describe the behavior of *testing* formulas by varying the inputs. (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.) 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 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 3 and Figure 3(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].

Summary of results: Unsupervised clustering significantly improved the interpretability over individual patterns. The resulting pattern groups revealed evidence of four different high-level strategies. There are three main points to note: 1) the match of the verbalizations in an independent user study strongly suggests that our findings are not only real but also are at an appropriate abstraction level; 2) the code inspection result (Group 3), was not yet proven. HCI researchers had begun to suspect its presence, but had not been able to statistically show it in more than three years of manual empirical work in the context of of gender HCI [2]; 3) two of the results are new, namely the beneficial effects of batch checking (Groups 1 and 2) and the detrimental effects of using the debugging features (CheckMarks and XMarks) for to-do list purposes (Group 5). These results

Authorized licensed use limited to: OREGON STATE UNIV. Downloaded on October 23, 2008 at 15:49 from IEEE Xplore. Restrictions apply.

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

This paper described a complete data mining process applied to Human-Computer Interaction data. Our goal was to identify interpretable human strategies. We applied sequential pattern mining as our initial step, which produced a large number of patterns that were difficult to interpret and lacked generality. This led us to explore a number of different ways to summarize/generalize beyond individual patterns via clustering, followed by statistical testing. We successfully identified some highly interesting pattern groups that corresponded well to strategies that have been identified by the users themselves when interviewed in a separate user study.

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

First, 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 an instance of a general phenomenon. To understand the overall trend, we need to see many instances to capture what is general and go beyond the specifics of individual patterns. This suggests that grouping patterns into meaningful groups can increase the interpretability and the generality of the findings.

Second, 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 clustering as well. We recommend leveraging different ways to group the patterns because of the following potential benefits: 1) often times different methods of grouping reach consensus about some clusters, providing strong support for the validity of the results; 2) different groupings collectively may reveal insights 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.

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] 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.

[9] X. Fern, C. Komireddy, and M. Burnett. Mining interpretable human strategies: A case study. Technical Report CS07-30-02, Oregon State University, 2007.

[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] 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.

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

[19] 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.

[20] 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.

Authorized licensed use limited to: OREGON STATE UNIV. Downloaded on October 23, 2008 at 15:49 from IEEE Xplore. Restrictions apply.