FIsViz: A Frequent Itemset Visualizer

Carson Kai-Sang Leung[*], Pourang P. Irani, and Christopher L. Carmichael

The University of Manitoba, Winnipeg, MB, Canada
{kleung, irani, umcarmi1}@cs.umanitoba.ca

Abstract. Since its introduction, frequent itemset mining has been the subject of numerous studies. However, most of them return frequent itemsets in the form of textual lists. The common cliché that "a picture is worth a thousand words" advocates that visual representation can enhance user understanding of the inherent relations in a collection of objects such as frequent itemsets. Many visualization systems have been developed to visualize raw data or mining results. However, most of these systems were not designed for visualizing frequent itemsets. In this paper, we propose a *frequent itemset visualizer* (*FIsViz*). FIsViz provides many useful features so that users can effectively see and obtain implicit, previously unknown, and potentially useful information that is embedded in data of various real-life applications.

1 Introduction

Frequent itemset mining [1,10,11,12,13] plays an essential role in the mining of various patterns (e.g., association rules, correlation, sequences, episodes, maximal itemsets, closed itemsets) and is in demand for many real-life applications. Mined frequent itemsets can answer many questions (examples of which are shown in Fig. 1) that help users make important decisions. Hence, numerous frequent itemset mining algorithms have been proposed over the last decade. However, most of them return a collection of frequent itemsets in *textual form* (e.g., a very long unsorted list of frequent itemsets). As a result, users may not easily see the useful information that is embedded in the data.

To assist users in gaining insight into massive amounts of data or information, researchers have considered many visualization techniques [7,16]. Visualization systems like Spotfire [2], VisDB [8] and Polaris [17] have been developed for visualizing data but *not* the mining results. For systems that visualize the mining results, the focus has been mainly on results such as clusters [9,15], decision trees [3], social networks [4] or association rules [5,6]—instead of frequent itemsets. Showing a collection of frequent itemsets in *graphical form* can help users understand the nature of the information and show the relations embedded in the data.

Recently, researchers have shown interests in visualizing frequent itemsets. For instance, Munzner et al. [14] presented a visualization system called PowerSetViewer (PSV), which provides users with guaranteed visibility of frequent itemsets. However, PSV does not show the relationship between related itemsets (e.g., not easy to know that itemsets {apples, bananas} and {apples, bananas, cherries} are related—the former is a

[*] Corresponding author.

T. Washio et al. (Eds.): PAKDD 2008, LNAI 5012, pp. 644–652, 2008.

Q1. Store managers may want to find answers to the following questions:
- **(a)** What kinds of fruits (e.g., {apples, bananas}) are frequently purchased by customers?
- **(b)** How frequently items are purchased *individually* (e.g., 70% of customers purchased apples)?
- **(c)** How frequently items are purchased *together* (e.g., only 30% of customers purchased apples and bananas together)?
- **(d)** What items are frequently purchased together with cherries (e.g., {apples, bananas, cherries, dates})?
- **(e)** Which itemset has the highest cardinality (e.g., a basket containing 30 different kinds of fruits)?
- **(f)** Which is the most frequently purchased 3-itemset (e.g., {apples, bananas, cherries})?

Q2. University administrators may want to know which popular elective courses (e.g., {Astronomy 101, Biology 102, Chemistry 100}) are taken by students?

Q3. Internet providers may want to figure out what Webpages are frequently browsed by Internet users in a single session?

Q4. Bookstore owners may want to know which books are also bought by customers who bought a particular data mining book?

Fig. 1. Sample questions answered by frequent itemset mining

subset of the latter). Yang [18] also developed a system that can visualize frequent itemsets. However, his system was primarily designed to visualize association rules, and it does not scale very well in assisting users to immediately see certain patterns among a very large number of items/itemsets.

Hence, some natural questions to ask are: Can we design a system that explicitly shows relationships among frequent itemsets? Can we help users find satisfactory answers to important questions that could lead to critical business decisions?

To this end, we present a visualizer to enhance the data mining process of the user by providing answers to some important business questions. The **key contribution** of our work is a novel interactive system, called *FIsViz*, for *visualizing frequent itemsets*. This visualizer provides users with clear and explicit depictions about frequent itemsets that are embedded in the data of interest. Hence, FIsViz enables users—at a glance—to infer patterns and answers to many questions (e.g., Q1-Q4 in Fig. 1); it also provides interactive features for constrained and interactive mining. Moreover, with FIsViz, users can efficiently find closed itemsets and can easily formulate association rules from the displayed frequent itemsets.

This paper is organized as follows. Next section describes related work. In Section 3, we introduce our FIsViz and describe its design as well as features. Section 4 shows evaluation results. Then, we briefly discuss, in Section 5, the scalability of FIsViz with respect to large datasets. Finally, conclusions are presented in Section 6.

2 Related Work

Yang [18] designed a system mainly to visualize association rules—but can also be used to visualize frequent itemsets—in a two-dimensional space consisting of many vertical axes. In the system, all domain items are sorted according to their frequencies and are evenly distributed along each vertical axis. A k-itemset is then represented by a curve that extends from one vertical axis to another connecting k such axes. The thickness of the curve indicates the frequency (or support) of such an itemset. However, such a representation suffers from the following problems: (i) The use of thickness only shows *relative* (but not *exact*) frequency of itemsets. Comparing the thickness of curves is not easy. (ii) Since items are sorted and *evenly* distributed along the axes, users only know some items are more frequent than the others, but cannot get a sense of how these items are related to each other in terms of their exact frequencies (e.g., whether item a is twice as frequent as, or just slightly more frequent than, item b). (iii) Although Yang's system is able to show both association rules and frequent itemsets, his system does not provide users with many interactive features, which are necessary if a large graph containing many items to be displayed.

PowerSetViewer (PSV) [14] is designed specifically for displaying frequent itemsets in the context of the powerset universe. With PSV, frequent itemsets are first grouped together based on cardinality (each represented by a different background color) in a two-dimensional grid; itemsets of the same cardinality are then mapped into grid squares. When the number of k-itemsets exceeds the number of allocated grid squares, PSV maps several frequent itemsets into one square. A square is highlighted if it contains at least one frequent itemset. This provides users with *guaranteed visibility* of itemsets. While PSV is truly designed for visualizing frequent itemsets, it also suffers from the following problems: (i) As a highlighted grid square may contain many frequent itemsets, it is not easy to find out which one or more itemsets (among all the itemsets represented by such a square) are frequent. (ii) PSV does not tell the *exact* frequencies of frequent itemsets. (iii) It is difficult to grasp the relationships between two related itemsets (e.g., $\{a, b\}$ is a subset of $\{a, b, c, d\}$).

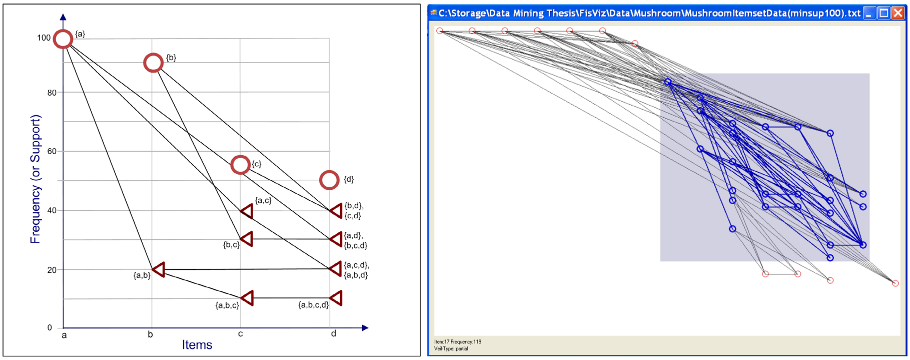
3 FIsViz: Our Proposed System for Visualizing Frequent Itemsets

In this section, we show basic representation and demonstrate features of our proposed *frequent itemset visualizer* (*FIsViz*).

3.1 Basic Representation of FIsViz

FIsViz shows frequent k-itemsets in a two-dimensional space. The x-axis shows the n domain items, which are arranged in non-ascending frequency order (by default) on the x-axis. The y-axis, which can be in *logarithmic-scale* or *normal-scale*, shows the frequencies of itemsets. A connecting edge between two items suggests that the two items appear together in the dataset. In this way, a non-singleton itemset (e.g., {apples, bananas, cherries}) is represented by a polyline (series of consecutive edges) ended with a left-pointing triangle. Each singleton itemset (e.g., {apples}) is represented by a circle. See Fig. 2(a) for a snapshot of the basic representation of FIsViz. Observations (Fig. 3) of this snapshot reveal the following *properties* associated with this basic representation of FIsViz:

1. FIsViz provides users with a quick intuitive overview about the frequency of each individual domain item (indicated by a circle) with frequency clearly indicated by its y-position.



(a) Basic representation of FIsViz **(b)** Visualization of itemsets from mushroom data

Fig. 2. Snapshots of our proposed FIsViz

1. Items a and b frequently occur individually (with $sup(a)$=100% and $sup(b)$=90%), but their combination $\{a, b\}$ does not occur that frequently (with $sup(\{a, b\})$=20%).
2. The leftmost item is a (which has the highest frequency) and the rightmost item is d (which has the lowest frequency). Moreover, $sup(a)$=100% $\geq sup(b) \geq sup(c) \geq sup(d)$=50%.
3. $sup(\{a, b\}$=20%, $sup(\{a, b, c, d\})$=10% and $sup(\{c, d\})$=40%. Knowing this information, users can easily obtain the support, confidence and lift of association rule $\{a, b\}\Rightarrow\{c, d\}$ using $sup(\{a, b, c, d\})$, $\frac{sup(\{a,b,c,d\})}{sup(\{a,b\})}$ and $\frac{sup(\{a,b,c,d\})}{sup(\{a,b\}) \times sup(\{c,d\})}$ respectively. Moreover, observing that $sup(\{a, b\}) = sup(\{a, b, d\})$, users can easily determine that $conf(\{a, b\}\Rightarrow\{d\})$=100%.
4. When moving along the polyline representing $\{a, b, c, d\}$, itemset $\{a, b\}$ appears to the left of $\{a, b, c\}$ (as the former is a *prefix* of the latter). Similarly, $\{a, b, c, d\}$ appears to the right of $\{a, b, c\}$ (as the former is an *extension* of the latter). Moreover, $sup(\{a, b\})$=20% $\geq sup(\{a, b, c\}$=10% $\geq sup(\{a, b, c, d\})$=10%.
5. All *subsets* of $\{a, c, d\}$ appear to the left and above $\{a, c, d\}$.
6. All *supersets* of $\{a, c\}$ appear to the right and below $\{a, c\}$.
7. $\{a, c\}$ is a *closed itemset*, but $\{b, c\}$ is *not* because $sup(\{b, c\}) = sup(\{b, c, d\})$.

Fig. 3. Observations on Fig. 2(a)

2. The most frequently occurring item (which with the highest frequency) appears on the left side and the least frequently occurring one appears on the right side.
3. Each k-itemset (where $k > 1$) is represented by a polyline, and its frequency is the frequency of the right-end item node of the polyline. The frequency is clearly indicated by the y-position of left-pointing triangle.
4. All prefixes of any k-itemset α appear on the left of α along the polyline that represents α, whereas all extensions of α appears on the right of α along such a polyline. Moreover, due to the Apriori property [1], it is guaranteed that the frequency of any prefix of $\alpha \geq$ the frequency of $\alpha \geq$ the frequency of any extension of α. When one moves along the polyline from right to left, the frequencies of prefixes of α are non-decreasing. Furthermore, users can see *how the frequency of α changes when*

(i) *truncating some items to form a prefix or* (ii) *appending some items to form an extension*.

5. All the nodes representing subsets of an itemset α appear to the left and above the node representing α. Knowing this property is useful because this reduces the search space (to only the left and above the node representing α) if one wants to search for all subsets of α.
6. Similarly, all the nodes representing supersets of an itemset α appear to the right and below the node representing α. Again, this property helps reduce the search space.
7. In addition to finding *frequent itemsets* (and their frequencies), users can also find *closed itemsets* (and their frequencies) effectively.

3.2 Features of FIsViz

Feature 1 (Query on frequency). With our FIsViz, users can easily find all *frequent items* and/or *frequent itemsets* (i.e., with frequencies exceeding the user-specified minimum frequency threshold *minsup*) by ignoring everything that lies below the "threshold line" y=*minsup* (i.e., ignoring the lower portion of the graph). To a further extent, the representation of itemsets in FIsViz can lead to effective *interactive mining*. To elaborate, with FIsViz, users can easily see what (and how many) itemsets are above a certain frequency. Based on this information, users can freely adjust *minsup* (by moving the slider—which controls *minsup*—up and down along the y-axis) and find an appropriate value for *minsup*. See Fig. 2(b), which shows itemsets with frequencies $\geq$ *minsup*. Moreover, FIsViz also provides two related features: (i) It allows users to interactively adjust *minsup* and automatically counts the number of itemsets that satisfy *minsup*. By doing so, users can easily find **top-*N* frequent itemsets**. (ii) It also allows users to pose a **range query on frequency** (by specifying both minimum and maximum frequency thresholds *minsup* and *maxsup*) and then shows all itemsets with frequencies falling within the range [*minsup*, *maxsup*].

Feature 2 (Query on cardinality). In FIsViz, itemsets of different cardinalities are drawn in different color, and itemsets with higher cardinality are drawn over those with the lower cardinality. This helps users find *closed itemsets* and *maximal itemsets*. Moreover, FIsViz also allows users to pose a **range query on cardinality** so that only those frequent itemsets with cardinality k within the user-specified range [k_{min}, k_{max}] are drawn.

Feature 3 (Query on itemsets). FIsViz also allows users to interactively select certain items of interest (e.g., promotional items in a store) and to pose the queries on itemsets. Examples of these queries include (i) "find all itemsets containing *some* of selected items", (ii) "find all itemsets containing at least *all* of the selected items", and (iii) "find all itemsets *not* containing any of the selected items". See Fig. 2(b), in which selected itemsets are highlighted.

Feature 4 (Details-on-demand). Details-on-demand consists of techniques that provide more details whenever the user requests them. The key idea is that FIsViz gives users an overview of the entire dataset and then allows users to interactively select parts of the overview for which they request more details—by hovering the mouse over different parts of the display. Specifically, FIsViz supports details-on-demand in the following ways: (i) When **the mouse hovers on an edge/polyline** connecting two nodes (say, representing items x and y), FIsViz shows a list of itemsets containing both x and y. Selecting an itemset in the list instantly highlights the specific edge it is contained in, as well as both of its connecting nodes, so that users can see where the edge starts and ends. (ii) When the **mouse hovers over a node**, FIsViz shows a list of all itemsets contained in all the edges starting or ending at this node. Selecting an itemset from the list instantly highlights the edge it is contained in. (iii) When the **mouse hovers over a pixel** in the display (even if it is not part of the graph), a small box appears showing the frequency and itemsets encoded by the mouse position. This is particularly useful when users need to see among the vast array of edges what a particular point in the display refers to.

Feature 5 (Formation of association rules). For many existing systems for visualizing association rules (which only shows the support and confidence of the rule $A \Rightarrow C$), it is not easy to obtain the frequencies of itemset A and of C. In contrast, our FIsViz displays the information needed to infer and compute these rules. For instance, one can form a rule and then compute its *support* as well as *confidence* based on the frequencies of A and C. See Observation 3 in Fig. 3. Moreover, FIsViz provides an additional benefit that users can compute other metrics such as *lift*.

Feature 6 (Ordering of domain items). By default, FIsViz arranges the domain items (on the x-axis) in non-ascending frequency order. However, FIsViz also provides users with an option to arrange items other orders. Having such an option is useful for *constrained mining*, in which users may want to arrange the items according to some constraints (e.g., put items of interest—say, promotional items—on the left and other items on the right of the screen). With this item ordering, the following property is preserved: *Frequencies of prefixes of the k-itemsets remain non-decreasing when moving from right to left.*

4 Evaluation Results

We conducted two sets of evaluation tests. In the first set, we tested functionality of our FIsViz by showing how it can be applicable in various scenarios or real-life applications. In the second set, we tested performance of our FIsViz.

4.1 Functionality Test

In the first set of evaluation tests, we compared our FIsViz with existing systems like Yang's system [18] and PSV [14]. We considered many different real-life scenarios. For each scenario, we determined whether these systems can handle the scenarios. If so, we examined how these system display the mining results. The evaluation results show that our FIsViz was effective in all these scenarios. A few samples of these scenarios are shown in Fig. 4.

4.2 Performance Test

In the performance test, we used (i) several IBM synthetic datasets [1] and (ii) some real-life databases (e.g., mushroom dataset) from UC Irvine Machine Learning

For **Q1(a)** in Fig. 1, frequently purchased fruits are itemsets with high frequency. With PSV, users may *not* be able to easily see the content of the itemsets because several itemsets may be mapped into a grid square. In contrast, our FIsViz shows all frequent itemsets by polylines, which are easily visible.

For **Q1(b)** and **Q1(c)**, Yang's system shows frequencies of itemsets, but it does not give users the *exact* frequencies of itemsets because frequencies are represented by the thickness of curves. In PSV, the brightness of a grid square shows its density (i.e., the number of itemsets that were mapped into that square) but *not* its frequency. In contrast, users can easily obtain the frequencies of itemsets from our FIsViz.

For **Q1(d)**, PSV does *not* provide the linkage or relationship between related itemsets. In contrast, our FIsViz provides users with a feature of handling queries on itemsets containing one specific item (in this scenario, cherries).

For **Q1(e)**, PSV shows itemsets with highest cardinality on the bottom of the screen. Our FIsViz allows users to query on cardinality. Hence, itemsets with highest cardinality (i.e., polylines with the most number of nodes) can be displayed.

For **Q1(f)**, with FIsViz, users can first pose a query on cardinality to find only 3-itemsets, and then picks the itemset with the highest frequency.

Fig. 4. Sample scenarios and evaluation results for the functionality test

Depository. The results produced are consistent. Fig 2(b) shows a screenshot of using the real-life mushroom dataset.

In the first experiment, we varied the size of databases. The results showed that the runtime (which includes CPU and I/Os) increased linearly with the number of transactions in the database.

In the second experiment, we varied the number of items in the domain. The results showed that the runtime increased when the number of domain items increased.

In the third experiment, we varied the user-defined frequency threshold. When the threshold increased, the number of itemsets that satisfy the threshold (i.e., itemsets to be displayed) decreased, which in turn leads to a decrease in runtime.

5 Discussion: Scalability of FIsViz

Recall that our FIsViz presents items on the x-axis. If each item is displayed by one pixel, then eventually the visualizer is limited by the number of items it can display within the user viewpoint. To overcome this limitation, we are developing the following approaches: (i) We apply *multi-resolution visualization*, with which we show the overall structure at one resolution and present details (upon the user request) at a different resolution. (ii) We span some of the displays beyond the viewpoint by carefully *embedding FIsViz with navigation facilities* (e.g., scrolling, panning) so that users can view items that are off-screen with minimum effort and without losing connectivity information from the lines in the display. (iii) We condense the large dataset by *creating taxonomies on domain items* based on their properties (e.g., item type) so that a large number of items can be coalesced onto a data point, which can then be opened for more details (or closed for fewer details) by users.

6 Conclusions

Most of frequent itemset mining studies return a collection of frequent itemsets in textual forms, which can be very long and difficult to comprehend. Since “a picture is worth a thousand words”, it is desirable to have visual systems. However, many existing visual systems were not designed to show frequent itemsets. To improve this situation, we proposed and developed a powerful *frequent itemset visualizer* (*FIsViz*), which provides users with explicit and easily-visible information among the frequent itemsets. Specifically, FIsViz gives a quick intuitive overview of all the itemsets and their frequencies (e.g., visual clues show which individual items are most frequent and how the items or itemsets are distributed); it also provides in-depth details of interesting itemsets (e.g., itemsets of a certain frequency and/or cardinality) through human interaction like mouse hover. Evaluation results showed the effectiveness of FIsViz in answering a board range of questions for real-life applications. These answers helps users in making appropriate business decisions.

Acknowledgement. This project is partially supported by NSERC (Canada) in the form of research grants.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proc. VLDB, pp. 487–499 (1994)
2. Ahlberg, C.: Spotfire: an information exploration environment. SIGMOD Record 25(4), 25–29 (1996)
3. Ankerst, M., Elsen, C., et al.: Visual classification: an interactive approach to decision tree construction. In: Proc. KDD, pp. 392–396 (1999)
4. Appan, P., Sundaram, H., Tseng, B.L.: Summarization and visualization of communication patterns in a large-scale social network. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) PAKDD 2006. LNCS (LNAI), vol. 3918, pp. 371–379. Springer, Heidelberg (2006)
5. Brunk, C., Kelly, J., Kohavi, R.: MineSet: an integrated system for data mining. In: Proc. KDD, pp. 135–138 (1997)
6. Han, J., Cercone, N.: AViz: A visualization system for discovering numeric association rules. In: Terano, T., Liu, H., Chen, A.L.P. (eds.) PAKDD 2000. LNCS (LNAI), vol. 1805, pp. 269–280. Springer, Heidelberg (2000)
7. Keim, D.A.: Information visualization and visual data mining. IEEE TVCG 8(1), 1–8 (2002)
8. Keim, D.A., Kriegel, H.-P.: Visualization techniques for mining large databases: a comparison. IEEE TKDE 8(6), 923–938 (1996)
9. Koren, Y., Harel, D.: A two-way visualization method for clustered data. In: Proc. KDD, pp. 589–594 (2003)
10. Lakshmanan, L.V.S., Leung, C.K.-S., Ng, R.T.: Efficient dynamic mining of constrained frequent sets. ACM TODS 28(4), 337–389 (2003)
11. Leung, C.K.-S., Khan, Q.I.: DSTree: A tree structure for the mining of frequent sets from data streams. In: Proc. IEEE ICDM, pp. 928–932 (2006)
12. Leung, C.K.-S., Khan, Q.I., Hoque, T.: CanTree: a tree structure for efficient incremental mining of frequent patterns. In: Proc. IEEE ICDM, pp. 274–281 (2005)

13. Leung, C.K.-S., Mateo, M.A.F., Brajczuk, D.A.: A tree-based approach for frequent pattern mining from uncertain data. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 653–661. Springer, Heidelberg (2008)
14. Munzner, T., Kong, Q., et al.: Visual mining of power sets with large alphabets. Technical report TR-2005-25, UBC, Canada (2005)
15. Pölzlbauer, G., Rauber, A., Dittenbach, M.: A vector field visualization technique for self-organizing maps. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 399–409. Springer, Heidelberg (2005)
16. Spence, R.: Information Visualization: Design for Interaction, 2nd edn. Prentice Hall, Harlow, UK (2007)
17. Stolte, C., Tang, D., Hanrahan, P.: Query, analysis, and visualization of hierarchically structured data using Polaris. In: Proc. KDD, pp. 112–122 (2002)
18. Yang, L.: Pruning and visualizing generalized association rules in parallel coordinates. IEEE TKDE 17(1), 60–70 (2005)