SubSect — An Interactive Itemset Visualization

Joey De Pauw[1][0000−0002−1417−922X](✉), Sandy Moens[1][0000−0002−7046−3022], and Bart Goethals[1,2][0000−0001−9327−9554]

[1] University of Antwerp, Belgium
{firstname.lastname}@uantwerpen.be
[2] Monash University, Australia

Abstract. Itemsets and association rules are among the most simple and intuitive patterns that are being used to explore transaction datasets. However, they lack meaning without both context and domain knowledge. Typically a user has to sift through hundreds of these patterns before finding an interesting one, losing sight of the forest for the trees. We propose a novel itemset and association rule visualization that makes it possible to inspect, assess, and compare patterns at a glance. In a case study we demonstrate its ability to facilitate a user in deriving and presenting valuable insights from a real-world dataset, which can not only save time and effort, but also reduce errors introduced by misconceptions.

Keywords: Visualization · Pattern mining · Itemsets · Association rules.

1 Introduction

Pattern mining is a commonly used technique in data exploration and data analysis [1]. In contrast to actively querying the data, pattern mining has the advantage of letting the data tell you what it looks like. Essentially, patterns such as itemsets and association rules provide an efficient way to represent local structures in the data. Most importantly, they have a summarizing property which facilitates the end user in interpreting and understanding a dataset.

Unfortunately, pattern mining alone does not suffice: typically a large number of patterns exists, even for relatively small datasets, making the process of discovering truly *interesting* patterns very tedious and strenuous for the practitioner. A transaction dataset with 20 different items for example, contains 2^{20} (more than 1 million) candidate itemsets. This is known as the pattern explosion problem. To make matters worse, *interestingness* is a subjective measure that can only be approximated by objective metrics or features [15].

In previous work this problem has been tackled for instance by sorting and filtering patterns based on different metrics [6] or by trying to minimize the number of reported patterns to the most descriptive subset [3, 17]. Another approach is to represent patterns in informative visualizations and rely on the end user to find what is interesting in their respective domain [4, 5, 9–13, 16]. Our contribution is situated in the latter context.

The final authenticated version is available online at https://doi.org/10.1007/978-3-030-65154-1_10

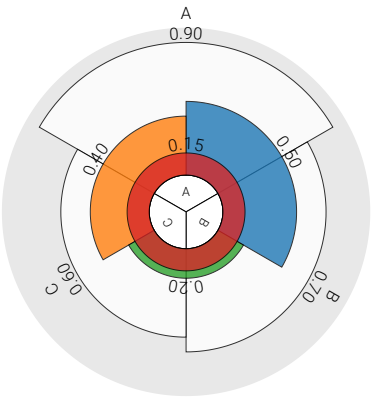


Fig. 1. Example of the visualization for an arbitrary itemset $\{A, B, C\}$.

We propose a visualization for itemsets based on the double decker plot from Hofmann et al. [11]. It exploits the monotonicity property which states that itemsets have a lower or equal support compared to the support of their subsets. An example of our visualization for the arbitrary itemset $\{A, B, C\}$ can be seen in Figure 1. To demonstrate the power of this visualization, we integrated the JavaScript-based implementation in the data mining and visualization tool SNIPER (formerly known as MIME [8]) and performed a case study.

This paper is organized as follows. In Section 2 we provide the required background in pattern mining and visualization. Section 3 describes the visualization itself with a theoretical analysis. Section 4 includes a case study where the efficiency of the visualization is verified in practice. Related work is discussed in Section 5 and finally we conclude our work in Section 6.

2 Background

2.1 Pattern Mining

Pattern mining is the process of discovering statistically relevant patterns in large datasets [7]. We focus on the mining of itemsets in a transaction database with items $\mathcal{I}$. A transaction database is a collection of subsets of $\mathcal{I}$. The *support* of an itemset is defined as the number of transactions that contain the itemset:

$$Supp(X) = |\{t \in \mathcal{D} \mid X \subseteq t\}|$$

with $\mathcal{D}$ the transaction database, t a transaction and X an itemset. Frequency is defined as the relative support:

$$Freq(X) = \frac{Supp(X)}{Supp(\emptyset)}$$

For every itemset a range of association rules can be derived by splitting it in two parts: an antecedent X and a consequent Y. An association rule is denoted

as $X \rightarrow Y$, where both X and Y are itemsets and $X \cap Y = \emptyset$. We define the *support* and *confidence* of an association rule as follows:

$$Supp(X \rightarrow Y) = Supp(X \cup Y) \qquad Conf(X \rightarrow Y) = \frac{Supp(X \rightarrow Y)}{Supp(X)}$$

Confidence is the conditional probability of a transaction containing itemset Y when X is already present. Additionally, we define *lift* as the ratio of observed support to that expected if X and Y were independent:

$$Lift(X \rightarrow Y) = \frac{Supp(X \rightarrow Y)}{Supp(X) \times Supp(Y)}$$

A frequent pattern mining algorithm such as Apriori or Eclat can be used to find all itemsets with a support higher than some user defined minimum support threshold [7]. This threshold is imposed to limit the search to patterns that are statistically relevant. With a minimum support of 1 the algorithm would simply calculate every pattern that occurs in the dataset, whereas with the minimum support threshold set to the number of transactions, the algorithm would only report patterns that occur in every transaction.

It is clear that the choice of minimum support has a big impact on the results of the pattern mining step and unfortunately there is usually no simple way to find the "right" value for this parameter. Therefore, choosing this parameter is typically an iterative and highly interactive process: a higher value may be too restrictive whereas a lower value can result in more uninteresting patterns that clutter the output. Defining the interestingness of a pattern usually requires domain knowledge which classic algorithms cannot take into account [15].

Alternatively, one can work bottom-up; whereas the first approach first mines an abundance of patterns followed by a filtering step, the bottom-up approach relies on patterns being built from the ground up by a domain expert. Hybrid solutions on the other hand try to include the user in each stage of the mining process, where for example the set of candidate itemsets can be reduced or expanded at each iteration of Apriori before the algorithm continues [18] or a framework is provided for the user to edit, combine or augment various patterns from different techniques [8].

Visualization plays a key role in any of these approaches. In the purely algorithmic approach, visualizations are mostly used in the filtering step, where a concise but informative visualization of itemsets is preferred over a plain list of itemsets. The bottom-up and hybrid approaches on the other hand use visualizations both for finding interesting combinations of items as well as for inspecting the resulting itemsets.

2.2 Visualization

The main advantage of using visualizations over textual representations is that they allow for better perceptual processing [14]. There are two phases in the theory of information processing: *perceptual processing* (seeing) and *cognitive*

processing (understanding). Perceptual processes are automatic, very fast, and mostly executed in parallel, while cognitive processes operate under conscious control of attention and are relatively slow, effortful, and sequential [14].

As an example of the power of perceptual processing one can imagine finding all itemsets that contain a specific item. In a textual notation you are limited to reading all the itemsets and remembering which ones contained the desired item. In a notation where all items are given a distinctive color it becomes possible to identify all occurrences of the item with a glance over the visualization. Likewise, many other visual variables can be used to encode information into the visualization, facilitating the user in extracting the desired information. This effect becomes even more powerful when considering comparisons between entities.

Visual variables can be categorized in *planar variables* (horizontal and vertical position) and *retinal variables* (shape, size, color, brightness, orientation and texture). A visualization can make use of these eight primitives in the visual alphabet to encode information, however it is not always desirable to use all of them. Leaving some degree of freedom is helpful when combining visualizations or annotating them.

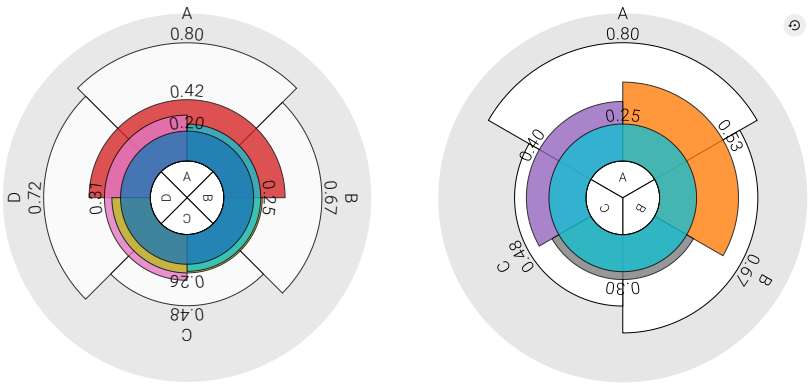
In our contribution *shape*, *size*, *color*, *brightness* and *orientation* are all used as part of the visualization to varying degrees. *Texture* and *position* are left free for annotation and for integration in other tools. The relative position of instances of the visualization is not defined, which allows it to be used in more complex layouts or potentially even in other visualization techniques.

3 SubSect

We present an efficient visualization for displaying itemsets and association rules. Its main goal is to show the most relevant information about an itemset (or a collection of itemsets) and allow for a domain expert to quickly interpret whether or not it is interesting. For this purpose, intuitiveness is very important.

Keeping the theory of visualization in mind, we define the following goals for our visualization in the context of pattern mining:

G1. The most important properties of a pattern should be semantically clear, i.e. readable on the visualization in an unambiguous way, ideally through perceptual processing.
G2. Users should be able to combine their domain knowledge with properties of itemsets to discover the most interesting patterns.
G3. A user should be able to easily compare two patterns.
G4. Ideally, more insights can be derived from the visualization or from the combination of two or more instances. For example, when the itemsets $\{A, B, C\}$ and $\{A\}$ are visible, information about the itemset $\{A, B\}$ may also be derived. Or from the set $\{A, B, C\}$ information about the rule $\{A, B\} \rightarrow \{C\}$ can be inferred.



(a) Itemset $\{A, B, C, D\}$ (b) Itemset $\{A, B, C\}$

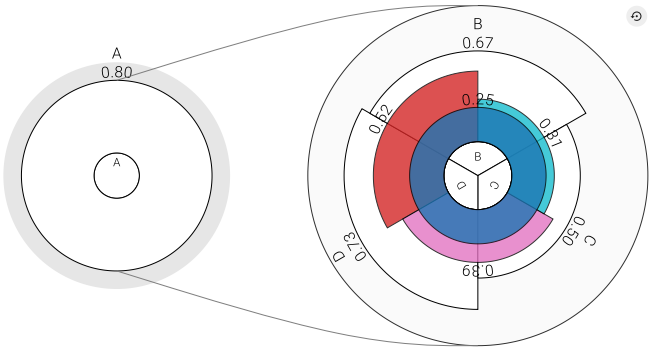
Fig. 2. The visualization for an arbitrary itemset (a) and for one of its subsets (b).

3.1 Basic Usage

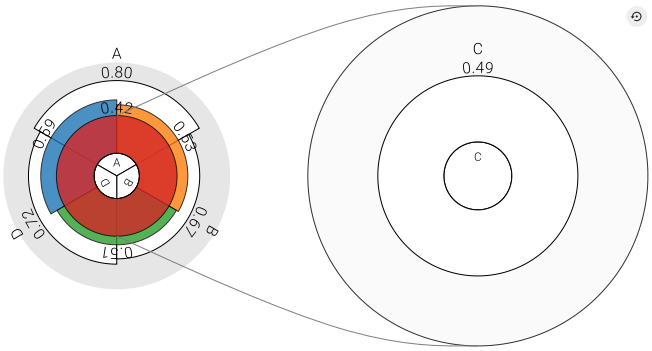
To explain how our visualization works, we first consider the example in Figure 2a. Every item in the itemset is represented in the center. The arcs around the center items show three levels of itemsets that can be formed from these items: 1) the itemset containing all k items, 2) all k-1 itemsets and 3) all singleton itemsets. For example, the blue full circle includes all four items A, B, C and D, and has a frequency of 0.2 as indicated by the label and its radius. The other segments represent subsets, like for example the cyan arc which spans items A, B and C. In correspondence with the higher frequency of this itemset (0.25), its arc also has a proportionally larger radius.

In order to reduce the overlap between arcs, we have chosen to let them span between the centers of the outer two items, rather than to have them cover 100% of the edge items. This trade-off reduces image clutter and therefore improves the scalability of our visualization, at the cost of a steeper learning curve: new users would expect the full items to be covered, but with some practice we believe that the meaning of the segments does become intuitive. The fact that same-cardinality itemsets always have the same *shape* (arc width) helps in this respect. Additionally, by hovering over the arc, label or frequency value of an itemset, all three of them are emphasized, clarifying which visual elements belong together.

Furthermore, in every image only the most interesting and informative subsets are rendered: for a k-itemset these are the k-1-itemsets and the singleton itemsets. In the left example for an itemset of 4 items, we display the 4-itemset, the 3-itemsets and the singleton itemsets. Together this combination of subsets provides the most useful information: the singleton itemsets give a global context and the k-1-itemsets place the k-itemset in a local context. Note that the arcs for singleton itemsets are always shaded in white while the others are given a unique color per itemset. This makes it easier to link multiple instances of the visualization that have items in common (G3).



(a) Itemset $\{B, C, D\}$ in $\{A\}$-conditional context.



(b) Itemset $\{C\}$ in $\{A, B, D\}$-conditional context.

Fig. 3. A more advanced example with the α-conditional view.

Finally, the visualization is equipped with three interactions to maximize usability: *dive deeper*, *α-conditional view* and *reset*. Animations like hover highlighting indicate the presence of these interactions and gradual transition animations ease the transition between "states" of the visualization, making the effect of the interactions more clear. Clicking on the cyan arc for example will *dive into* its respective itemset $\{A, B, C\}$. An animation shows that item D is removed from the center and the cyan arc becomes a full circle. Three new subsets are now visible. The result is shown in Figure 2b. Naturally this action can be repeated from the new view to *dive deeper* or the user can choose to go back to the top level with the *reset* button that just became available.

In the next section (3.2) we demonstrate a more advanced use case with the *α-conditional view*. Section 3.3 discusses the visualization from a theoretical point of view and analyzes its strengths and weaknesses.

3.2 Alpha-conditional View

For any given set of items α we define the α-conditional database as the set of transactions that contain all items in α. It provides for a natural way of thinking

about association rules. When we say *"80% of the people who buy diapers also buy beer"*, we essentially say that the itemset $\{beer\}$ has a relative support of 80% in the $\{diapers\}$-conditional database or equivalently that the association rule $\{diapers\} \rightarrow \{beer\}$ has a confidence of 80%.

Similar to the interaction for selecting an itemset to dig deeper, it is also possible to click a single item (in the center or on the outer edges) and add it to the α set or the *"scope"* as can be seen in Figure 3a. In this α-conditional view, the scope is always visible on the smaller visualization to the left. On the right-hand side, we see the remaining items and itemsets, but now with their frequencies relative to the scope.

By moving more items to the scope, it becomes clear that the scope set is rendered as another instance of the itemset visualization, i.e. with its respective subsets (see Figure 3b). This makes for a very interesting synergy, since not only the scope is visible, but also the context of what the α-conditional transactions look like. Again one can reset the visualization back to its original state with the reset button. Additionally it is possible to click items or itemsets in the left visualization to expand the scope by moving items back to the right-hand side.

3.3 Theoretical Analysis

Itemsets It is obvious that the frequency of the entire itemset, its k-1 subsets and its constituent items can be seen trivially through the labels and radii (G1). By the monotonicity property however, we can also derive some information about all the itemsets in between (G4). This is especially useful when the bounds are close together, since this provides a tighter estimate. In Figure 2a for example, we can derive that the frequency of $\{D, C\}$ must lie between 0.31 and 0.48 through the itemsets $\{A, D, C\}$ and $\{C\}$, since they are the most frequent superset and the least frequent subset respectively.

Association Rules The relation between an itemset and its subsets also implies an association rule. If two arcs are close together in terms of their radii, we know the association rule they imply will also have a high confidence (G4). Recall that the formula for confidence is $\frac{Supp(\{A,B,C,D\})}{Supp(\{A,B,C\})}$ for the rule $\{A, B, C\} \rightarrow \{D\}$. In Figure 2a we find the frequencies are 0.20 and 0.25, leading to a confidence of 0.8, which is also intuitively *"guessable"* from the difference in radius; i.e. without calculating the value, it is also simple to estimate it quite accurately from the visualization.

More importantly, we also introduced the α-conditional view to facilitate representing association rules. The frequency in an α-conditional database is equivalent to the confidence of the rule with α as antecedent and the itemset as consequent. Hence we can form any association rule from the itemset by simply moving the antecedent items to the scope and *"browsing"* to the desired itemset on the right-hand side. For example, Figure 3a shows, among others, the association rule $\{A\} \rightarrow \{B, C, D\}$ with a confidence of 0.25. We can also see that $\{A\}$ has a support of 0.8 and, from this, derive that $Supp(\{A\} \rightarrow \{B, C, D\}) = 0.8 \times 0.25 = 0.2$.

Scalability Despite our efforts to reduce overlap and clutter, it remains infeasible to render itemsets that consist of a large amount of items. For a k-itemset, $2k + 1$ arcs are rendered, which is already more favourable than an exponential amount. However, an inherent issue arises from the k-1-subsets, whose arcs need to span $k - 1$ items, resulting in an inhibiting amount of overlap for large k.

Specific solutions can be considered to facilitate the visualization of large patterns, such as combining items together in a preprocessing step or manually selecting the subsets to be rendered, for example by grouping some frequent and less interesting items in the center (see Section 6.1). In our experience however, large patterns are often of limited use: they are more complex to understand and typically either have a low frequency or contain many correlated/very frequent attributes that do not contribute to the pattern .

When the interactions between many attributes need to be studied, we opt for a collection of smaller patterns (that have attributes in common) over a single large pattern. Our visualization is better suited for this methodology of combining multiple instances together (G3).

After this brief analysis we find that our visualization already succeeds at goals G1, G3 and G4. Goal G2 relates to domain specific knowledge being integrated, which we did not discuss yet. The case study in the next section (Section 4) illustrates this concept.

4 Case Study

To demonstrate the effectiveness of our visualization we performed a case study with a real-world dataset. The main goal of this case study is to show how the visualization assists the end user in finding interesting patterns based on their domain knowledge (i.e. goal G2 from Section 3). A customer churn dataset[2] that describes information about customers of a telecom company who left within the last month was used. We chose this dataset because it is easy to understand without requiring any specific expertise or background (as for example would be the case for a financial or political dataset), yet it shows some interesting patterns. In total it contains 21 columns, describing information about 7,043 customers. Table 1 documents a subset of these attributes, i.e. the ones that appear in our examples. For every attribute we give the icon, name, description and its possible values.

First, the dataset was loaded in the research tool SNIPER[3], a web-based tool for pattern mining with a main focus on facilitating data exploration [8]. In the setup phase, we defined icons for each attribute and decided on a discretization strategy to handle numeric variables. Five equal-width buckets were used. Given the context, this choice provided a good resolution with adequate support for the individual items. After preprocessing, the resulting transaction dataset consisted of 60 unique items and 7,043 transactions.

[2] https://www.kaggle.com/blastchar/telco-customer-churn/
[3] https://bitbucket.org/sandymoens/sniper/

Table 1. Icon, name, description and possible values for each attribute in the dataset.

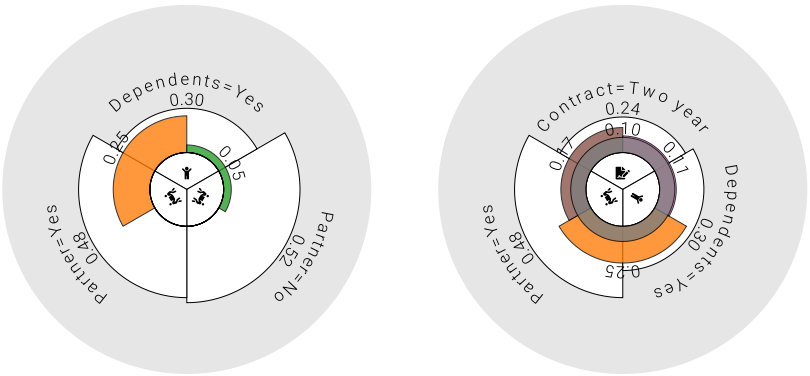
Attribute	Values
Churn	Yes, No
Indicates whether the customer left within the last month. This is the intended target variable of the dataset.	
Partner	Yes, No
Yes if the customer has a partner, *No* otherwise.	
Dependents	Yes, No
Whether or not the customer has dependents. In most cases dependents are children, students or elderly people.	
Contract	Month-to-month, One year, Two year
The contract term of the customer.	
Internet Service	Fiber optic, DSL, No
Which type of internet service the customer opted for, or *No* if none.	
Phone Service	Yes, No
Indicates if the contract includes phone service.	
Gender	Male, Female
The gender of the customer.	
Online Security	Yes, No, No internet service
Whether the customer has the online security service or *No internet service* if N/A.	
Tech Support	Yes, No, No internet service
Similar to security, this attribute indicates if the contract includes tech support.	

Then the dataset was explored with the functionality provided by SNIPER. This includes, but is not limited to, classical itemset mining (like Eclat [7]), rule mining, sorting and filtering of patterns and manually building patterns by combining them or forming them based on the insight brought by various metrics. We mainly used the latter technique to create patterns based on the conditional support and lift metrics.

The following sections each provide examples of patterns that were found in the data and how our visualization was used to find and interpret them. A live version for each example can be found on https://joeydp.github.io/SubSect/.

4.1 Example - Lift

The most straightforward use of our visualization is illustrated in this example. Suppose we are interested in the attribute *partner* and would like to investigate if there is a relation with the attribute *dependents*. Figure 4a gives a concise and intuitive representation of the information needed to compare these two



(a) Itemset {Partner=Yes, Partner=No, Dependents=Yes}

(b) Itemset {Partner=Yes, Contract=Two year, Dependents=Yes}

Fig. 4. Example of two interesting itemsets rendered in our visualization.

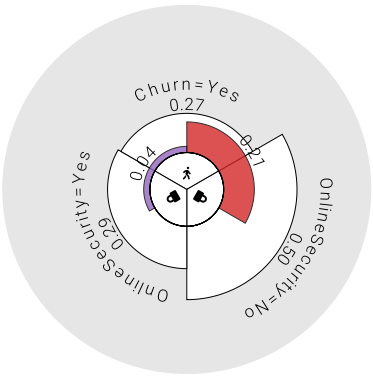
attributes. Note that we assume the meaning and possible values of these attributes are known beforehand, i.e. we know these are two boolean attributes with mutually exclusive values.

First we can see that a little over half of the customers with a partner, also have dependents ($\frac{0.25}{0.48}$). For customers without a partner this ratio is only one in ten ($\frac{0.05}{0.52}$). Without context this information is quite meaningless, so we compare it to the expected distribution of dependents, which is 30% for the overall dataset. Now it becomes clear that customers with a partner have a higher chance of also having dependents and inversely the chance is lower for customers without a partner. In the other direction we can see that five out of six customers with dependents also have a partner and the remaining one out of six do not.

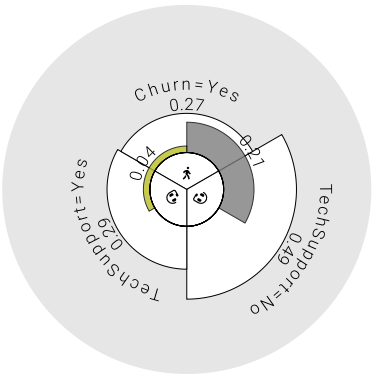
Both of the previously described patterns show association rules with relatively high confidence. The advantage of using this visualization is that the lift, i.e. the support divided by the expected support if the variables were independent, can also easily be derived. This is a good example to illustrate why the single itemsets and k-1 itemsets were selected to be visualized: the local and global context synergize to allow the end user to derive new information.

The second example in Figure 4b shows the presence of an association rule with a very high confidence, albeit with a relatively low support. That is 11% of the customers have a *two year contract* and *dependents*, and 10% have those two and a *partner*, leading to he association rule $\{Dependents, Two\ year\ contract\} \rightarrow \{Partner\}$ with a confidence of around 91%, which is intuitive given the domain knowledge behind these attributes. It seems logical that customers with a long term contract and with dependents would be more likely to have a partner.

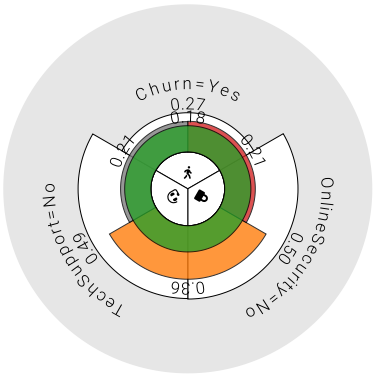
Perhaps more interesting is the fact that there is relatively little overlap between the two association rules that constitute the previous one. We find that only 25 in 30 people with dependents also have a partner ($\approx$ 83%) and



(a) Itemset {Churn=Yes, Online Security=No, Online Security=Yes}



(b) Itemset {Churn=Yes, Tech Support=No, Tech Support=Yes}



(c) Itemset {Churn=Yes, Tech Support=No, Online Security=No}

Fig. 5. Example to show how multiple instances of the visualization can be used together and how the dependence between attributes can be derived.

that only $\frac{17}{24} \approx 71\%$ of the people with a two year contract have a partner. In other words, both variables "contribute" to the association rule in the sense that without either one, the confidence would drop. This information can all be derived intuitively from our visualization.

4.2 Example - Independence

For this second example we investigate some variables that relate to *churn*. Other than the obvious variables like *tenure* and *contract* type, we also found that *online security* (Figure 5a) and *tech support* (Figure 5b) have a high impact on churn. In both cases not taking the service increases churn rate to around 42%. It is clear from the visualizations that these patterns have almost exactly

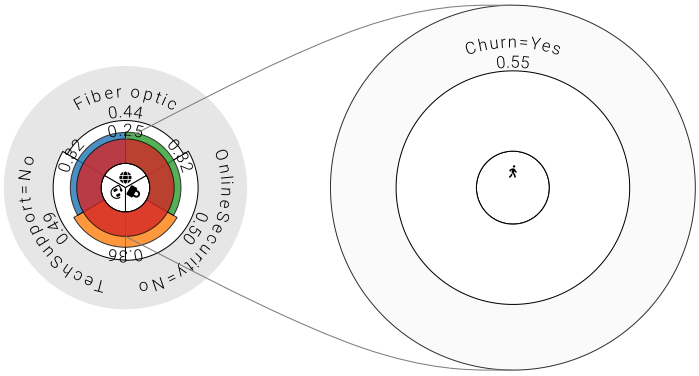


Fig. 6. Association rule {Internet Service=Fiber Optic, Tech Support=No, Online Security=No}→{Churn=Yes}.

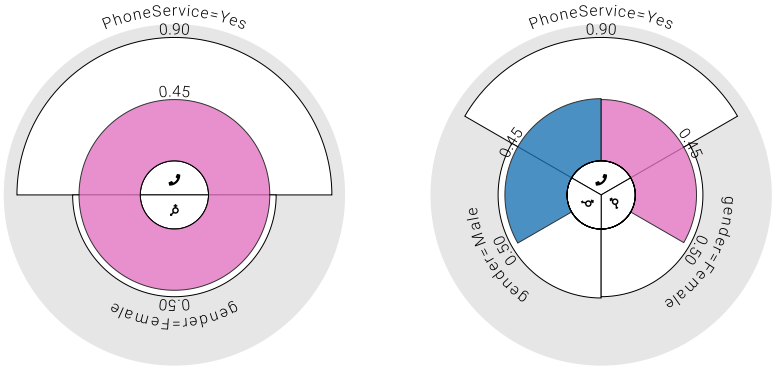
the same distribution of customers. One would assume that these services are highly dependent, such that you can only enlist for security if you also take tech support and vice versa, as would be the case if they were part of a plan.

However by combining these variables in one visualization (Figure 5c), we find that only about 72% ($\frac{0.36}{0.5}$) of the customers that don't have the security service also don't have tech support. Similarly the same relation holds for customers that don't have tech support. Again only about 72% of them didn't take the security service. In other words, the variables are less dependent than expected and hence the combination of the two also leads to a higher churn rate (50%) than either of them achieved independently (about 42% each).

Another unexpected variable that increases churn rate we found is *Fiber optic internet*. Using our visualization it is possible to play with these variables and create the desired association rules. For example an interesting task could be to find a subset of customers with a specific size, that has the highest chance of leaving. This would allow the telecom provider to invest its limited resources to counter churn in a more targeted strategy.

Figure 6 shows how this can be achieved with the help of our visualization. Items that increase churn can be added to the context, such that their interactions with other items can immediately be seen. At the same time the size of the target group and the churn rate remain visible. In this example we demonstrate a rule that selects 25% of the population with an elevated churn rate of 55%, which is quite remarkable considering that the most obvious and least influenceable attributes (*tenure* and *contract*) were not even included.

The telecom company can use this information to investigate why these unexpected variables have an impact on the churn rate. For example, offering free tech support might lead to less customers opting out and consequently to a potentially larger profit. This is however only speculation. An alternative explanation could be that customers who choose to pay for tech support tend to be prefer stability and are less prone to change between providers.



(a) Itemset {Gender=Female, Phone Service=Yes}

(b) Itemset {Gender=Male, Gender=Female, Phone Service=Yes}

Fig. 7. Two examples to illustrate the importance of taking context and meaning into account when evaluating patterns.

4.3 Example - Context

In this example, the importance of taking the context of a pattern into account is shown. Figure 7a displays what on first sight might appear to be a very interesting rule $\{Gender = Female\} \rightarrow \{Phone\ Service = Yes\}$ with a confidence of 90%. However with a closer inspection it becomes clear that 90% of the customers has *phone service*, independent of their gender, making this rule meaningless.

The second example (Figure 7b) shows a pattern where an attribute is divided equally over two values. In a different context, this might be an interesting pattern. For example when comparing adolescents with the elderly, where we might *expect* a difference. In this case however, we know that phone service should not be biased towards a specific gender and we can deduce that it is not an interesting pattern.

Alternatively, one can sort or filter the association rules behind these patterns on how much their *lift* deviates from one. Since in this case the variables are independent, the association rules have a lift of one and would hence be filtered out or given a low rank. Under the expected independence assumption we would miss these patterns, where in fact they can be interesting when the domain implies an expected dependency. That is to say, both the context of a pattern and the larger context of the meaning of its attributes contribute to its *interestingness*.

5 Related Work

In the literature we can classify visualization based on whether they support itemsets, association rules or both. Some techniques focus on representing single

itemsets [11] where others try to visualize the entire dataset at once [4, 5, 10, 12, 13, 16]. Naturally, each visualization has its strengths and weaknesses and hence repeated use of different methods can lead to deeper insights into the data [9]. We give an overview of related work in pattern visualization and remark on how these techniques relate to our visualization.

Single Itemset/Association Rule Hofmann et al. discuss a double decker plot based on mosaic plots [11]. The idea is to visualize a single association rule with one item as consequent and provide metrics from which its interestingness can be assessed. In this visualization it is easy to verify that all items contribute to the rule, which likely indicates an interesting pattern. It is however rather limited in the amount of information that is visible or can be derived interactively. For example, because all possible subsets are rendered, the support also ends up scattered over the figure. The combination of a circular layout and our choice of subsets ensures that all segments are continuous in our visualization.

Circular Two similar circular plots have been proposed earlier. However they differ from our visualization in that they both display the entire transaction database as a dissection in frequent itemsets. The first one by Dubois et al., called icVAT [4], has itemsets that radiate inward based on their support. Colors are used to show the cardinality of each itemset and the distance to the center (or radius) represents the support of each itemset.

In the second study by Keim et al. (FP-Viz [12]), items are layered to form itemsets and their support is indicated using a color scale and the width (or angle) of its segment. The main difference between these techniques is that icVAT has a fixed width and varying radius, where FP-Viz uses a fixed radius with varying width. Despite arguing that they make the link between items more clear, it remains difficult to see how different itemsets relate to each other since the same item can occur multiple times in different places, contrary to our technique.

Graph Based Ertek and Demiriz propose a straightforward graph based visualization [5]. Nodes represent items and special itemset nodes, with edges to the individual items, indicate itemsets. This distinction however makes it difficult to see interactions between itemsets and subsets, which is something our visualization excels in.

Leung et al. propose a different graph based method where nodes represent itemsets and they are organized according to their items and frequency [13]. This approach already depicts the interactions between itemsets and the differences in frequency more clearly. An issue with seeing the global picture remains however, since there is often no perceivable link between itemsets that share an item. In other words, it is difficult to trace itemsets to their sub- or supersets.

Bothorel et al. propose a graph based visualization with a circular layout to organize the nodes [2]. Itemsets are linked to their subsets and nodes that are linked also tend to be placed closer together. This visualization provides a

good overview of what the data looks like and where the patterns may be, but patterns can no longer be discerned at a large scale.

Matrix/Table Based Hahsler and Karpienko describe a grouped matrix representation in their study from 2017 [10]. This hierarchical approach enables the user to get an overview of the data at an abstract level and also to dive deeper to inspect more specific phenomena. Contrary to most techniques that try to visualize the entire dataset, this one is well equipped to handle the scaling problem. Its most prominent downside is that it projects antecedents and consequents on different axes, making it difficult to find interactions between them.

VisAR by Techapichetvanich and Datta [16] is a table based technique for visualizing association rules. It aims to provide a good overview that is efficient to query. Indeed, their technique lists association rules in a concise way with a clear and singular meaning. Since the visualization behaves like a table or list, it avoids screen clutter and occlusion. There is however no link between antecedent and consequent items in this visualization, which makes it hard to find interactions between rules. Furthermore long lists become impractical to use as well due to the increasing distance between items and between association rules.

6 Conclusions

A novel visualization technique for itemsets and association rules was introduced and analyzed from a visualization-theoretical perspective. Its functionalities and interactions were explained, making its application in data mining clear. The technique can be used as a concise representation for itemsets where the local and global context is immediately clear. In addition, the α-conditional view provides for an intuitive way to query interesting subsets of the data or to represent association rules. Furthermore the provided interactions allow an end user to actively query the visualization and extract valuable insights.

In the accompanying case study with a real-world dataset we demonstrated that a variety of patterns can be visualized and further understood with our technique. Users can combine their domain knowledge and expectations with the properties of itemsets and association rules to extract interesting information from the data. Furthermore, multiple instances of the visualization can be used together to describe complex relations.

Finally, we situated our technique in a broader context of itemset and association rule visualization techniques, remarking on differences and similarities. This thorough study of related work also supports our premise that the proposed visualization is effectively new in the field.

6.1 Future Work

We limited the features of our visualization to only the fundamental concepts of itemsets. No features for specific use cases were included, which of course leads to the benefit of making it usable in most contexts. However, as mentioned in

Section 2.2, some degrees of freedom were left for annotation and integration. It would be interesting to see how the visualization can be expanded upon to provide functionality for specific use cases. For example in our case study with a clear target variable (*churn*), one could add a tooltip on hover that displays the fraction of items in the subset where $Churn = Yes$. If desired, this information could even be visualized by partially filling the segments with a fixed color.

Furthermore, to limit clutter we opted to keep the visualization sober and concise. More complex features could be integrated to wager this conciseness for more visible information. One idea is to also show relevant complementing itemsets, perhaps as an arc that radiates in from the outside. Similarly, additional arcs can be rendered to show the expected frequency of itemsets under certain independence assumptions, which would visualize *lift* more explicitly. Another potential addition would be to reserve space in the center of the circle for "common" items. These items could be selected interactively to limit the number of rendered itemsets to the ones containing these items, which would reduce occlusion. Finally, a user study could be performed to validate the effectiveness of our visualization.

Acknowledgements

This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

References

1. Aggarwal, C.C., Bhuiyan, M.A., Al Hasan, M.: Frequent pattern mining algorithms: A survey. In: Frequent pattern mining, pp. 19–64. Springer (2014)
2. Bothorel, G., Serrurier, M., Hurter, C.: Visualization of frequent itemsets with nested circular layout and bundling algorithm. In: International Symposium on Visual Computing. pp. 396–405. Springer (2013)
3. Calders, T., Goethals, B.: Non-derivable itemset mining. Data Mining and Knowledge Discovery 14(1), 171–206 (2007)
4. Dubois, P.M., Han, Z., Jiang, F., Leung, C.K.: An interactive circular visual analytic tool for visualization of web data. In: 2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI). pp. 709–712. IEEE (2016)
5. Ertek, G., Demiriz, A.: A framework for visualizing association mining results. In: International Symposium on Computer and Information Sciences. pp. 593–602. Springer (2006)
6. Geng, L., Hamilton, H.J.: Interestingness measures for data mining: A survey. ACM Computing Surveys (CSUR) 38(3), 9 (2006)
7. Goethals, B.: Survey on frequent pattern mining. Univ. of Helsinki 19, 840–852 (2003)
8. Goethals, B., Moens, S., Vreeken, J.: Mime: a framework for interactive visual pattern mining. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 757–760. ACM (2011)
9. Hahsler, M.: arulesViz: Interactive Visualization of Association Rules with R. The R Journal 9(2), 163–175 (2017), https://doi.org/10.32614/RJ-2017-047

10. Hahsler, M., Karpienko, R.: Visualizing association rules in hierarchical groups. Journal of Business Economics 87(3), 317–335 (2017)
11. Hofmann, H., Siebes, A.P., Wilhelm, A.F.: Visualizing association rules with interactive mosaic plots. In: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 227–235. ACM (2000)
12. Keim, D.A., Schneidewind, J., Sips, M.: FP-Viz: Visual frequent pattern mining. In: InfoVis (2005)
13. Leung, C.K.S., Irani, P.P., Carmichael, C.L.: Fisviz: a frequent itemset visualizer. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 644–652. Springer (2008)
14. Moody, D.: The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering. IEEE Transactions on software engineering 35(6), 756–779 (2009)
15. Tan, P.N., Kumar, V., Srivastava, J.: Selecting the right interestingness measure for association patterns. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 32–41. ACM (2002)
16. Techapichetvanich, K., Datta, A.: VisAR: A new technique for visualizing mined association rules. In: International Conference on Advanced Data Mining and Applications. pp. 88–95. Springer (2005)
17. Vreeken, J., Van Leeuwen, M., Siebes, A.: Krimp: mining itemsets that compress. Data Mining and Knowledge Discovery 23(1), 169–214 (2011)
18. Yamamoto, C.H., Oliveira, M.C.F., Rezende, S.O.: Including the user in the knowledge discovery loop: Interactive itemset-driven rule extraction. In: Proceedings of the 2008 ACM symposium on Applied computing. pp. 1212–1217 (2008)