Msc Thesis

An Experimental Analysis of the Pattern Explosion

Author: Vincent Menger

January 2015

Utrecht University

ICA-3230643

Supervisor: prof. dr. Arno Siebes
Second supervisor: dr. Ad Feelders

Abstract

Although Frequent Itemset Mining is a classical Data Mining technique, the causes of the *pattern explosion* – one of its major challenges – have never been thoroughly researched. We perform an experimental analysis of the causes of the pattern explosion. Several experiments are performed on five selected datasets. The experiments show that similar transactions usually support similar patterns, similar patterns however do not necessarily describe similar data. In the first case the correlation is strong, yet in the second case only a weak correlation exists. We furthermore show that it is possible in many patterns to swap items for other particular items without influencing the data that is described much. This shows that in many cases, there is little interaction between the items and at least not all of their relations are significant.

1 Introduction

In the field of Data Mining, Frequent Itemset Mining is a popular method for finding interesting relations between attribute values in a dataset. The method considers a transactional database from which it mines all itemsets that are frequent according to some user specified threshold. The first approaches, for instance by Agrawal et al [1], mainly applied this technique to the analysis of market basket data, for example for product placement or marketing decisions. Since then, many other applications have been thought of. It is used in diverse topics of research such as intrusion or fraud detection, click stream data analysis, and several topics within bioinformatics such as genome analysis and drug design. Frequent Itemset Mining usually serves as a first step for learning association rules, which illustrate something about the data in the form of intuitive if-then rules. Some variations such as Frequent Tree Mining or Frequent Graph Mining also exist, but we will only concern ourselves with the problem of Frequent Itemset Mining.

1.1 Problem statement

The classical problem considers a set of items $\mathcal{I} = \{i_1, ..., i_n\}$, and a multiset $\mathcal{D} = \{t_1, ..., t_m\}$, of which each transaction t_i is a subset of $\mathcal{I}$. A transaction is said to support a pattern, if all items in the pattern are also in the transaction. The support of some pattern p, denoted by $\sigma(p)$, are those transactions in $\mathcal{D}$ that support the pattern. The mining task is then to find all patterns p that have $|\sigma(p)| \geq \texttt{minsup}$, where `minsup` is some user defined threshold. For high values of `minsup`, this usually yields few patterns that were already well known in the first place. For low values however a myriad of patterns is to be expected, far more than can be inspected by a human. It is not unusual for the number of patterns to exceed the number of tuples in the database, even by orders of magnitude. The latter case has also been referred to as the *pattern explosion*. The topic of efficiently mining all frequent itemsets is well researched, as is compressing the number of patterns to an amount that is inspectable by a human. Some assumptions have been made why this pattern explosion is happening, for instance that groups of similar patterns may largely describe the same data. Yet there appears to exist no actual research into its causes. Such research would be interesting both for gaining insight into the process of Frequent Itemset Mining, and in relation to the various methods that currently exist to regulate the number of patterns. In this thesis we attempt an experimental analysis of the causes of the pattern explosion.

1.2 Experiment setup

For our analysis we employ various datasets that are further described in Section 2. Mining the patterns from these datasets is done using A Fast APRIORI Implementation [2]. For all other experiments we use own our software, written in Java version 1.7.0.65.

In section 2, we will describe the datasets that are experimented on using some descriptive statistics, and elaborate on the mining process. In section 3 some basic properties of the patterns and the transactions will be discussed, such as the distribution of support and the prior probabilities of the items. In section 4 we will investigate how much similarity can be found both in the dataset and in the set of patterns. Section 5 will cover the sensitivity of patterns and transactions to changes in items. Finally, in section 6 we will discuss the (in)dependence of items, and demonstrate how the items in patterns can be swapped without affecting the support much. Because this type of experimental research generates many tables and figures, those can be found in respectively Sections 8 and 9.

2 Datasets

For our analysis we use the five datasets listed below, obtained from the FIMI Dataset Repository.

1. **Accidents** – Traffic accidents in Belgium, the items correspond to circumstances in which the accidents occurred.
2. **Chess** – A dataset on the King and Rook vs King and Pawn chess endgame, in which the items are specific moves on the board.
3. **Kosarak** – Click stream data of an Hungarian news portal, no further information provided.
4. **Mushroom** – A dataset on 23 types of mushroom, of which several physical properties are captured in the items.
5. **Retail** – Data of a Belgian retail store, where the transactions are actual cash register transactions and the items are store products.

In Table 1 some additional statistics are listed. The Chess and Mushroom datasets are relatively small with both under 10^4 transactions, the other three datasets have more transactions. In another aspect the Kosarak and Retail datasets are quite similar, both in number of items and in density. The other three datasets have notably less items and a much higher density. There is no particular reason to use specifically these datasets, five datasets seems like a reasonable number to perform some experiments using the limited available computational resources, and still be able to say something general about the problem we are facing.

2.1 Mining the itemsets

We use A Fast APRIORI Implementation [2] to mine the patterns from these datasets, for several different settings of the `minsup` parameter. That the pattern explosion indeed occurs can be verified in Figure 1. On the horizontal axis the relative support is plotted, and on the vertical axis the log of the number of patterns is shown. The fact that these graphs are more or less straight lines suggests that the relation between support and the number of patterns is exponential. Especially for the Kosarak and Retail datasets, the number of patterns grows very rapidly. For the remainder of the analysis, in our experiments we use at most five different values for `minsup` so that the number of patterns spans several orders of magnitude. These supports used and the number of patterns obtained are listed in Table 2. The lowest value for `minsup` was chosen so that the patterns can still be mined in a reasonable amount of time. In general, we will present our results for every dataset, though for the sake of brevity not always for every support setting. Usually we are mainly interested in the lowest value of `minsup`, because the pattern explosion pertains to large numbers of patterns. If the results for different support settings show interesting or otherwise deviating behaviour, we will also present these.

3 Distribution of support

From Table 2 we can see how many patterns are mined, for the associated value of the `minsup` parameter. Many patterns however will have support that is higher than the `minsup` value, but we are not exactly sure how much higher, and for how many of the patterns this holds. To gain some first insights into the problem we would therefore like to know how the support of the patterns is distributed. To this end we simply take the supports of all patterns, sort them in descending order, and divide them in 100 percentiles. For each dataset, the results for respectively the lowest and the highest value of `minsup` are displayed in Figures 3 – 4. For the low value of `minsup` in Figure 3 it can be seen that in all cases, the first percentiles have a higher support than the rest of the percentiles, the support then slowly converges to the `minsup` limit. It is interesting to note that a lot of patterns are supported by a similar number of transactions, especially for the Retail and Kosarak datasets. Likewise, for the other datasets, after approximately the first 30% of the patterns the support does not change much at all. The fact that a lot of patterns describe about the same amount of data may be a first indication that they may also describe similar data. For the high values of `minsup` in Figure 4 we can see that the distribution does not change much as well. The main difference is that the higher value of `minsup` causes only patterns that are supported by a large fraction of the data to remain, as indicated by the different values on the vertical axis.

This exact experiment can be repeated from the perspective of transactions, using a 'reversed' notion of support: the number of patterns that a transaction supports. For transactions, there is no `minsup` equivalent, so the reverse support could span a much bigger range. It is possible for a transaction to support every pattern, or to support no pattern at all. The results, again respectively for for the lowest and highest value of `minsup` are displayed in Figure 5 – 6. In Figure 5 it is interesting to note that for the Accidents and Chess datasets the graph starts at 1, which means that there are transactions in these databases that support nearly every pattern, even when the number of patterns is of at least the same order of magnitude as the number of transactions. In both cases, the support drops linearly with the percentiles. This generally also holds true for the mushroom dataset, although the descent is not very smooth. The Kosarak and Retail datasets show different behaviour: they have a small percentage of transactions that support many patterns, the other transactions support way fewer patterns. As can be seen in Figure 6, changing the `minsup` parameter again shows few differences: the values on the vertical axis once again change because there are fewer patterns, and the area under the curve increases slightly, but the overall distribution remains very similar.

3.1 Distribution of items

Another interesting aspect of the datasets is the distribution of the items, in a similar way as in the paragraphs above. This is a property of a dataset and not of the patterns, so the patterns are not relevant here. For each item, we find the fraction of transactions in which the item occurs. Again we sort the values in descending order and divide them into 100 percentiles. The results are displayed in Figure 2. Again we see a clear distinction between the two datasets with low density and the other three datasets. The Accidents, Chess and Mushroom datasets have few items, of which some occur in almost all transactions. For the Kosarak and Retail datasets, even the most common items occur at most once per 50 transactions. For four of the datasets, the Chess dataset being exempt, it holds true that there is a small number of items that occur frequently among the transactions, and a large number of items that occur with lower frequency that is nearly constant. Based on these graphs, we could make a distinction between common items and uncommon items, which makes sense if we keep in mind the nature of the data. The Chess dataset is an exception to this, the relative support of each item is almost linear with its percentile score.

4 Similarity of patterns

As mentioned in the introduction, one of the most common assumptions about the pattern explosion is that groups of patterns essentially describe the same transactions. The fact that at least some patterns must describe the same transactions is obvious from the fact that there are more patterns than transactions. To what extent and in what way patterns overlap is still an open question. To investigate this, we introduce the following measure for similarity of patterns:

$$s(p_1, p_2) = \frac{|\sigma(p_1) \cap \sigma(p_2)|}{|\sigma(p_1) \cup \sigma(p_2)|} \qquad \text{(Pattern Similarity)}$$

This measure is also known as the Jaccard Distance on sets, in this case applied to the support of the patterns. We use it to measure the similarity of two patterns based on the transactions they describe. If they describe exactly the same data the similarity is 1, if their support sets have no transaction in common at all their similarity is 0. If the assumption mentioned above is correct, we should be able to find that many patterns are similar according to this measure.

Ideally, we would measure the similarity for each combination of patterns, but unfortunately this is computationally not feasible for any of the datasets. By using sampling we can diminish the computational resources that are needed. Our initial approach is to sample a small number of patterns and measure the similarity to all other patterns. This is feasible for some of the datasets, but unfortunately still too demanding for some support settings. Our only option is to revert to sampling a relatively small number of patterns, and computing the similarity between all those patterns. Some experimenting on the datasets that do allow computing the similarity between all patterns shows that sampling is a very reasonable approach, resulting in very similar graphs.

The results of this experiment are displayed in Figure 7. The experiment is repeated for all of the five datasets and for each of the five settings for support, thus creating 25 grids. For each grid, the patterns are placed both on the vertical and horizontal axis. We sample 50 patterns, uniformly according to support, so that the support of the samples has the same distribution as the patterns that are sampled from. In this way we can attest whether there is any interesting relation between pattern similarity and support. The samples with the highest support are placed in the top left corner. From the diagonal it can be verified that dark cells correspond to a high similarity, since $s(p,p) = 1$.

For both the Accidents and the Chess datasets, we can see that patterns describe the at least partially the same data as all other patterns, even for low values of `minsup`. Furthermore, the uniformity of the grid shows that the similarity of patterns has no relation to the support of those patterns at all. If we look at the grids for higher values of `minsup` we can note that the grids become darker, i.e. the patterns become more similar. This makes sense when we take into account that since these patterns have higher support, they describe more data, and are thus also more likely to describe similar data. The Retail dataset on the bottom row shows entirely different behaviour from these two datasets: hardly any patterns are similar. Although this can partly be explained by noting that for the Retail dataset a much lower value of `minsup` is needed to obtain a number of patterns of the same order of magnitude as for the other datasets, we would still expect more similarity when increasing the minimum support. The results for the Mushroom and Kosarak datasets seem to fit best with the Accidents and Chess datasets. Although the grids are not so uniform, they clearly show that there is some similarity among the patterns. The patterns that emerge in the grids of the Mushroom dataset seem to be random artifacts of the dataset itself, at least no deeper meaning has been found. For the grids for the Kosarak dataset it can be noted that there is a strong relation between similarity and support. For the patterns with high support, there is no similarity at all. The similarity only occurs below a certain threshold value of `minsup`, which is why there is no similarity in the rightmost grid: only patterns with high support remain.

Based on these visualizations, the assumption that groups of patterns describe similar data seems to be largely justifiable. In any case for the dense Accidents, Chess datasets, and at least partially for the Mushroom and Kosarak datasets. For the Retail dataset we can say that there is hardly any similarity at all, which is definitely not in line with the assumption – something different must be going on.

4.1 Similarity of transactions

The above experiment can be repeated from the perspective of transactions. Let $\alpha(t)$ denote the set of patterns that are supported by t, then we can measure the similarity of transactions with:

$$s(t_1, t_2) = \frac{|\alpha(t_1) \cap \alpha(t_2)|}{|\alpha(t_1) \cup \alpha(t_2)|} \tag{Transaction Similarity}$$

The conditions of the experiment are equal to the conditions of the experiment above: we sample 50 transactions and compute the similarity between all of them. The results are displayed in Figure 8, they should be read the same way as in the experiment from the above paragraphs. For all datasets, one important characteristic is that the similarity of transactions is not uniform, but related to the number of patterns that a transaction supports. We can see that the transactions that support many patterns are usually more similar to each other than the transactions that support few patterns. The Accident and Chess datasets have many transactions that support a similar set of patterns, and again the similarity increases when the minimum support is increased. This is due to the fact that for high values of `minsup`, each transaction supports fewer patterns, which makes it more likely to share a part of its support with other transactions. For the Retail and Kosarak datasets very similar behaviour is visible as in the previous experiment. In both cases, the transactions are not very similar at all. The grids of the Mushroom dataset has some characteristics of both the dense and the sparse datasets. Once again we can observe some seemingly random patterns. It is interesting to note that among the transactions certain clusters of transactions appear, where large sets of transactions all support roughly the same patterns, but support hardly any patterns at all. These clusters of transactions may help explain the seemingly random artifacts that were visible in the Pattern Similarity grids.

No very sharp conclusion can be drawn from these plots, unless it is that the similarity among transactions is different among different datasets. We could state that dense datasets are more likely to have similar transactions, but should also disclose that the Mushroom dataset – which higher density than the Accidents dataset – does not have more similar transactions than the Accidents dataset.

5 Sensitivity to changes in items

In Section 4 we have discussed patterns in terms of the data they describe, but they can also be viewed in terms of their items. We saw that in most datasets there is similarity in the data patterns describe, it would be interesting to find out how this relates to the patterns on a syntactical level. We again use the same measure for this purpose, along with a new measure:

$$s(p_1, p_2) = \frac{|p_1 \cap p_2|}{|p_1 \cup p_2|} \tag{Item similarity}$$

This measure can be applied to both patterns and transactions, and measures the similarity in terms of the items in a pattern or transaction. In particular, it would be interesting to see what the relation between the Item Similarity and the Pattern Similarity is. We therefore run the following experiment: for some random pattern r, compute both the Pattern Similarity and the Item similarity to all other patterns. We then sort these data points by the Item Similarity and put them in a 100 bins. This results in 100 data points, which can be plotted as in Figures 9 – 13. For reference the same thing is done for the Transaction Similarity. The left column contains the result for the Transaction Similarity, the right column contains the result for the Pattern Similarity.

Unfortunately we can only show some of the plots based on a single random pattern or transaction, because incorporating hundreds of plots is simply not possible. The plots that are shown are representative of the results and are also chosen to illustrate some interesting features best. More data is available however, and shows similar results. If we look at the left column with the Transaction Similarity plots, we can clearly see that there is a strong correlation between the Transaction Similarity and the Item Similarity. The type of relation varies among the datasets and even among the different support settings, but we are not so much interested in the nature of this relation – we merely observe that they are usually correlated. For the Pattern Similarity in the right column however, this relation is not very strong. Although there exists some relation, because the data points are not uniformly distributed over the space, the correlation is not nearly as strong as with the Transaction Similarity. We have also included the plots of the Chess datasets, which serves as a counterexample to the previous statement: both for the transactions and the patterns, the plots seem to be roughly equally correlated.

This result may seem surprising at first. If we take a transaction and make a slight change to the items in the transaction, the set of patterns that it supports does not change all that much. Yet if we take a pattern, and make a slight change to the items, the support can change a lot more. The main reason for this appears to be the fact that the average number of items per transaction is usually larger than the average number of items per patter. Especially when the support is high, most patterns consist of only a few items. It seems natural to assume that a perturbation in the items will have greater effect if there are few items present in the first place.

In our goal of gaining more insight in the pattern explosion, this experiment reveals some interesting new fact. We have previously discussed the hypothesis that the pattern explosion occurs because similar patterns describe similar data. This experiment refutes a part of this assumption. If we look at any of the plots from Figures 9 – 13, we can see that for any value on the Item Distance axis, the corresponding points show great variance in Pattern Distance, and vice versa. Although our experiments from Section 3 have already shown that there is in most cases at least some overlap in the data described by the patterns, this experiment tells us that these patterns themselves don't need to be alike.

6 Independence of items

We have already seen that frequent itemset mining usually results in a large number of frequent itemsets, the only constraint for an itemset to be called frequent however is its how often it occurs in the data. The mining algorithm simply finds all those itemsets, without considering whether their items constitute any interesting relationship. The number of itemsets must to some extent be caused by coincidences in the data, the mining process however fails to consider this. To find out to what extent the items are statistically correlated, using a statistical test would be appropriate. One possibility would be a common χ^2 test, another option would be to use the more conservative Fisher's exact test as suggested by [7]. We investigate both options. Assuming independence among all items, we investigate which items co-occur in such a way that there is a statistical basis ($\alpha = 0.05$) to reject the independence assumption. Because we will be testing many hypotheses, one for each pair of items, we use the Bonferroni correction to ensure that the probability that at least one hypothesis is wrongly rejected is at most α. This is done by dividing α by the number of hypotheses tested. Not every dataset has the same number of items, so to ensure that the same number of hypotheses is tested for each dataset we sample a fixed number of items. We choose 75 items, because this is the smallest number of items, in the chess dataset. Note that the Bonferroni method is the most conservative method for regulating the familywise error rate, the outcomes are therefore on the low side. They should not be interpreted as definitive evidence but rather as an indication how strongly the items are correlated.

The results are displayed in Table 3. For both tests, the fraction of pairs of items for which the independence assumption is rejected is listed. The first thing to note is that the two statistical tests do not differ much. It is also notable that the two sparse datasets, Kosarak and Retail are very weakly correlated. The other three datasets are more strongly correlated, most notably the Mushroom dataset. As stated above, these numbers merely serve as an indication how strongly the data is correlated. It is clear that not all Frequent Itemsets can consist of strongly correlated items, if these statistical tests point out that in some cases only a very small part of the items are related at all. In the next paragraph we will further explore the dependence relations between items.

6.1 Swapping itemsets

If indeed many of the items are independent, it should be possible to remove certain items from itemsets and to replace them by certain others, while the support remains roughly the same. The challenge of course is to find out what items to swap from the patterns. First, consider some random itemset R and some subset of its items $S \subseteq R$. Furthermore, consider the closure of the subpattern $c(S)$. If we identify the other itemsets which have the same closure as S, it is known that we can swap any of those subpatterns for S without affecting the support of R. It would be interesting to loosen the constraint of identifying the itemsets that have the same closure, to finding itemsets that have similar closures. For this purpose we can use the same Pattern Similarity metric as before, since closures are itemsets as well.

The experiment we devise starts by taking a random frequent itemset R. We then proceed by identifying the set $I^- = \{i_1^-, ..., i_n^-\}$, containing n frequent subpatterns of the random itemset R, these are the items that will be swapped out of the random itemset. For each $i_k^- \in I^-$, we determine the closure $c(i_k^-)$, and then find $I_k^+ = \{\ i^+ \mid s(c(i_k^-), c(i^+)) \geq 0.9\ \}$, i.e. the set of patterns of which the closure has a similarity of at least 0.9 according to the Pattern Similarity measure. Note that each subpattern i_k^- that is swapped out is associated with a set I_k^+ of which all items can be swapped back in. Finally the swapping takes place, by subsequently removing the i_k^- from R and adding all $i^+ \in I_k^+$, which results in a number of new patterns. The patterns are then filtered so that a set of unique patterns remains. The process is repeated for a number of random patterns. Since we iterate over all patterns when identifying the subpatterns of the random itemset, then iterate over the transactions to find which closures are similar, subsequently iterate over all patterns that need to be swapped back, and finally repeat all of this for a number of random patterns, the computational burden of this task is quite high. We therefore restrict the subpatterns in I^- to itemsets of at most 3 items, and use 50 random patterns. Along with precomputing the closures, this restrains the problem sufficiently to be able to execute it.

Some sample output, in this case for the Accidents dataset, is shown below. In this snippet it can already be seen that the support after swapping itemsets does not differ much from the support of the original random pattern.

```
Random pattern [17, 18, 21, 29, 43, 63], support = 146311
This pattern contains 41 subpatterns of size ≤ 3

    Contains subpattern [29, 43] which has closure [29, 43]
        Swapping in pattern [16, 29, 43], results in [16, 17, 18, 21, 29, 43, 63]
        which has support 142725
        Swapping in pattern [12, 29, 43], results in [12, 17, 18, 21, 29, 43, 63]
        which has support 146231
        (...)
    Contains pattern [18, 63] which has closure [18, 63]
```

```
        Swapping in pattern [16, 17, 63], results in [16, 17, 21, 29, 43, 63] which
        has support 143072
        Swapping in pattern [12, 16, 63], results in [12, 16, 17, 21, 29, 43, 63]
        which has support 142971
        (...)
    (...)
```

After running the experiment for all datasets, we summarize the results averaged over 50 random patterns in Table 4. For all datasets the similarity of the closures is set to at least 0.9, except for the Retail dataset. For that case the threshold is lowered to 0.7 since no new patterns were found initially. In the second column labeled I^- we can see how many subpatterns the random pattern R on average contained. In the next column we can see how many unique new patterns the swapping resulted in. The third and fourth column respectively show the average and the standard deviation of the support after swapping, both normalized by the support of the random pattern R. The third column clearly shows that the support after swapping is on average very close to the support of the original random pattern, within 0.5%. That this is not achieved by averaging out the supports can be verified by looking at the standard deviation in column four, which is at most 3.56%. There is difference in the amount of unique new patterns that is found, this is lower for the Accidents and Retail datasets – even though the threshold for closure similarity is already lowered for the latter. Lowering the threshold even further would most likely increase the number of unique new patterns found, but at the cost of a greater standard deviation.

This experiment gives us an interesting view on the statistical significance of itemsets which seems to lie at the core of the pattern explosion. The fact that for each random pattern, several items can be replaced by items without influencing the support much shows that there is very little interaction between the replaced items and the rest of the pattern. If we look at the sample output, for example where `[29, 43]` is replaced by `[16, 29, 43]`, the new pattern is again frequent, but adding the item `16` does not expose any new interesting relation since the data that is described largely remains the same. If we combine the data of the distribution of items in Figure 2 with the data about statistical correlation between items in Table 3, we can see that many items have roughly the same support, and that there are plenty of items that have no statistical correlation. In other words, it is possible to add a number of different items to an existing itemset, to obtain a number of new itemsets that have roughly the same support but do not add anything new to the set of mined patterns. The most interesting itemsets are therefore the smallest itemsets, whereas the larger itemsets merely repeat the information that is already present in the smaller itemsets.

The number of unique patterns that are obtained in this way varies among the datasets, in the lowest case at least 10^2 patterns that describe the same data are found. The other datasets have at least 10^3 of those patterns. Lowering the threshold for similarity between closures would most likely result in an even larger number of new patterns. Supposing we could find a way to map all those patterns that describe the same data onto one pattern, the experiment suggests a reduction in the number of patterns of three orders of magnitude would be possible.

7 Conclusion

This thesis is an experimental investigation into the causes of the pattern explosion, a phenomenon that occurs when the `minsup` parameter in Frequent Itemset Mining is lowered. The hypothesis that the pattern explosion occurs because similar patterns describe similar data was found to be partially true. Although there is certainly overlap in the data that is described by the patterns, it is often the case that two similar patterns have very different support. The main contributor to the pattern explosion is the combination of the fact that many items have similar support, and that many items are statistically independent. This makes it possible to replace items in a pattern with other items without affecting the support much. Although these neighbouring patterns are also frequent, they do not show any interesting new relations in the data.

References

[1] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, June 1993.

[2] Ferenc Bodon. A fast apriori implementation. In *In Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, 2003.

[3] Toon Calders and Bart Goethals. Non-derivable itemset mining. *Data Mining and Knowledge Discovery*, 14(1):171–206, 2007.

[4] Adam Kirsch, Michael Mitzenmacher, Andrea Pietracaprina, Geppino Pucci, Eli Upfal, and Fabio Vandin. An efficient rigorous approach for identifying statistically significant frequent itemsets. In *Proceedings of the Twenty-eighth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '09, pages 117–126, New York, NY, USA, 2009. ACM.

[5] ArnoJ. Knobbe and EricK.Y. Ho. Pattern teams. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Knowledge Discovery in Databases: PKDD 2006*, volume 4213 of *Lecture Notes in Computer Science*, pages 577–584. Springer Berlin Heidelberg, 2006.

[6] P. Palmerini. Statistical properties of transactional databases. In *In SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 515–519. ACM, 2004.

[7] Geoffrey I. Webb. Self-sufficient itemsets: An approach to screening potentially interesting associations between items. *ACM Trans. Knowl. Discov. Data*, 4(1):3:1–3:20, January 2010.

8 Tables

Dataset	# Transactions	# Items	Density
Accidents	340183	468	0.072
Chess	3196	75	0.493
Kosarak	990002	41270	1.963E-4
Mushroom	8124	119	0.193
Retail	88162	16470	6.257E-4

Table 1: The number of transactions, number of items and density of each dataset

Accidents		Chess		Kosarak		Mushroom		Retail	
.3	149545	.5	1272932	.00105	542510	0.1	574431	0.0001	240852
.4	32528	.6	254944	.0014	303605	0.109	175061	0.00014	113119
.5	8057	.7	48731	.0018	134508	0.13	111799	0.0003	38152
.6	2074	.8	8227	.002	39464	0.25	5545	0.0007	12418
.7	529	.9	622	.004	2522	0.4	565	0.002	2691

Table 2: For each of the datasets, the relative support (left column) and number of patterns obtained (right column)

	χ^2	Fisher's exact test
Accidents	0.163	0.137
Chess	0.386	0.387
Kosarak	0.009	0.007
Mushroom	0.580	0.624
Retail	0.006	0.001

Table 3: For each of the datasets, the amount of correlation among the items, both for the χ^2 test and Fisher's exact test

Dataset	I^-	Unique new patterns	Average normalized support	Average normalized standard deviation of support
Accidents	66.88	394.48	0.99636	0.02412
Chess	107.18	4492.18	0.99748	0.0356
Kosarak	112.76	3146.6	0.99876	0.0187
Mushroom	102.32	2071	1.00166	0.03084
Retail	21.54	123.98	1.0065	0.03186

Table 4: Results of the swapping experiment

Dataset	# Transactions	# Items	Density
Connect	67557	129	0.3333
Pumsb	49046	2113	0.0350
T40I10D100K	100000	942	0.0420
T10I4D100K	100000	870	0.0116

Table 5: The number of transactions, items and the density of the four added datasets

Dataset	I^-	Unique new patterns	Average normalized support	Average normalized standard deviation of support
Connect	112.45	25035	0.986	0.036
Pumsb	92.038	14498.308	0.995	0.026
T40I10D100K	136.62	9118.98	1.005	0.012
T10I4D100K	21.96	349.24	1.027	0.091

Table 6: The results of the swapping experiment for the four new datasets

9 Figures

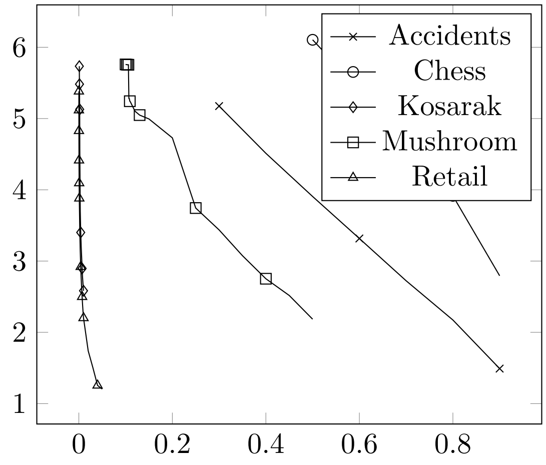


Figure 1: Support vs number of patterns on log scale

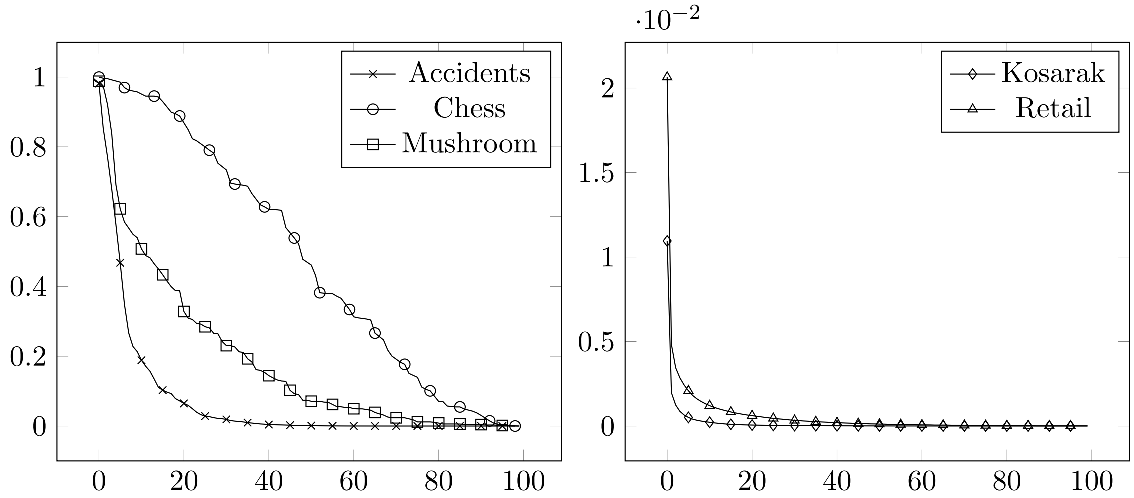


Figure 2: Prior counts of the items

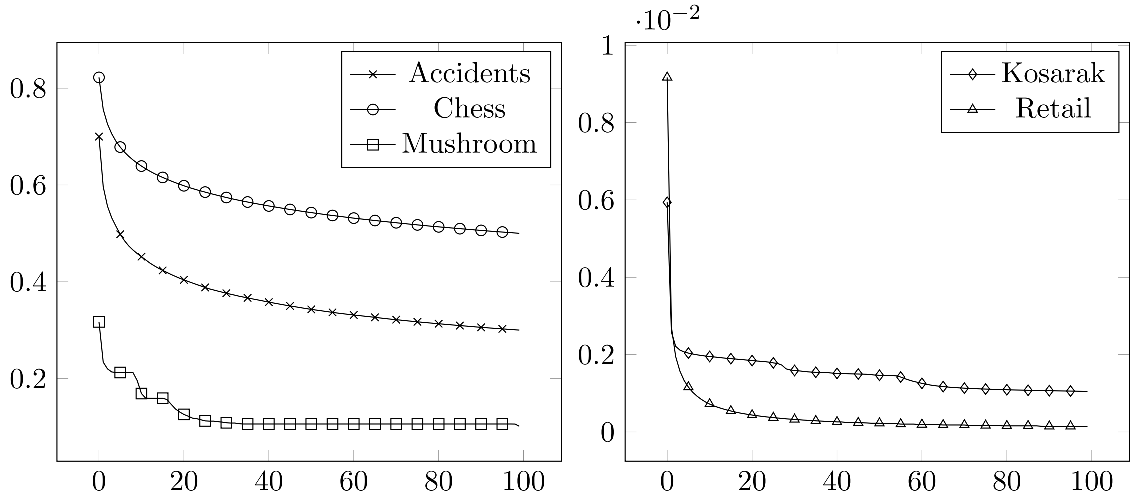


Figure 3: Distribution of support $\sigma(p)$ for the lowest value of `minsup`

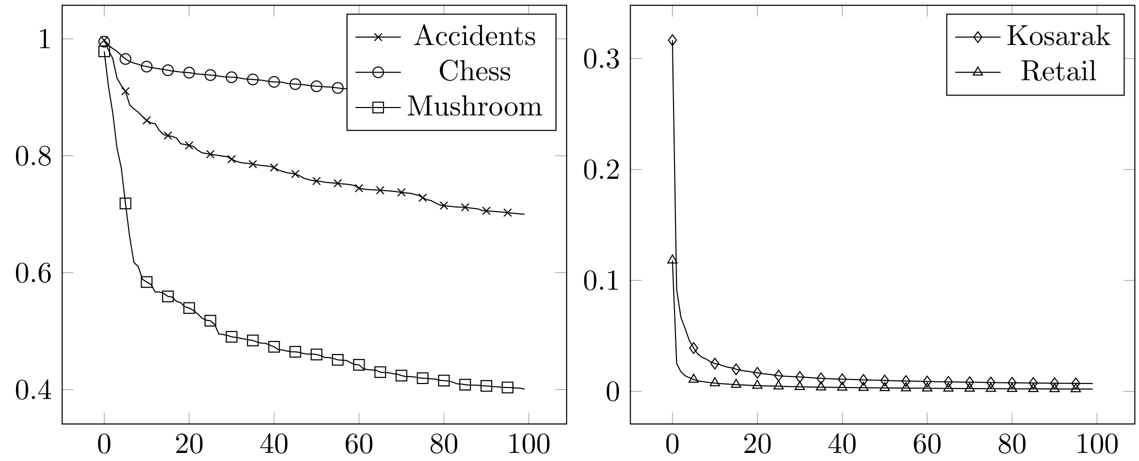


Figure 4: Distribution of support $\sigma(p)$ for the highest value of `minsup`

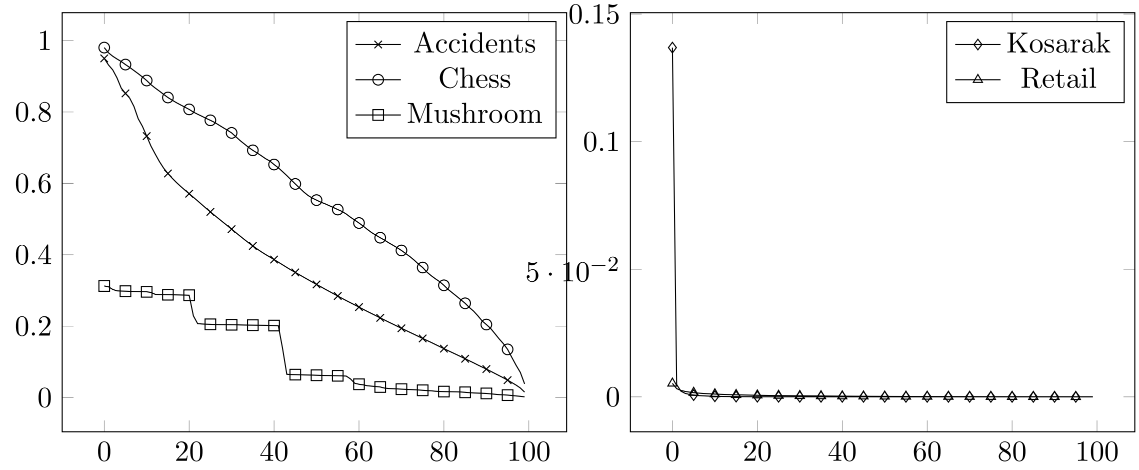


Figure 5: Distribution of reverse support $\alpha(t)$ for lowest value of `minsup`

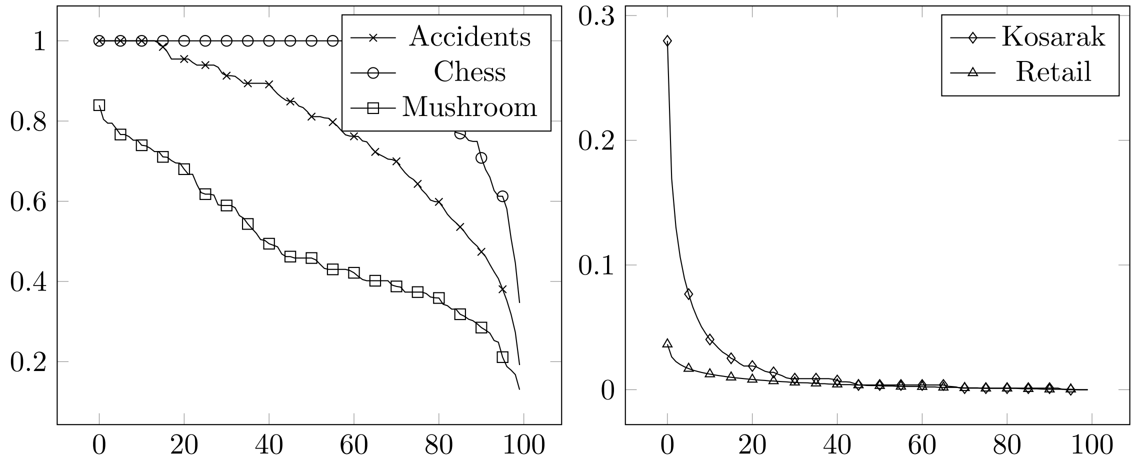


Figure 6: Distribution of reverse support $\alpha(t)$ for highest value of `minsup`

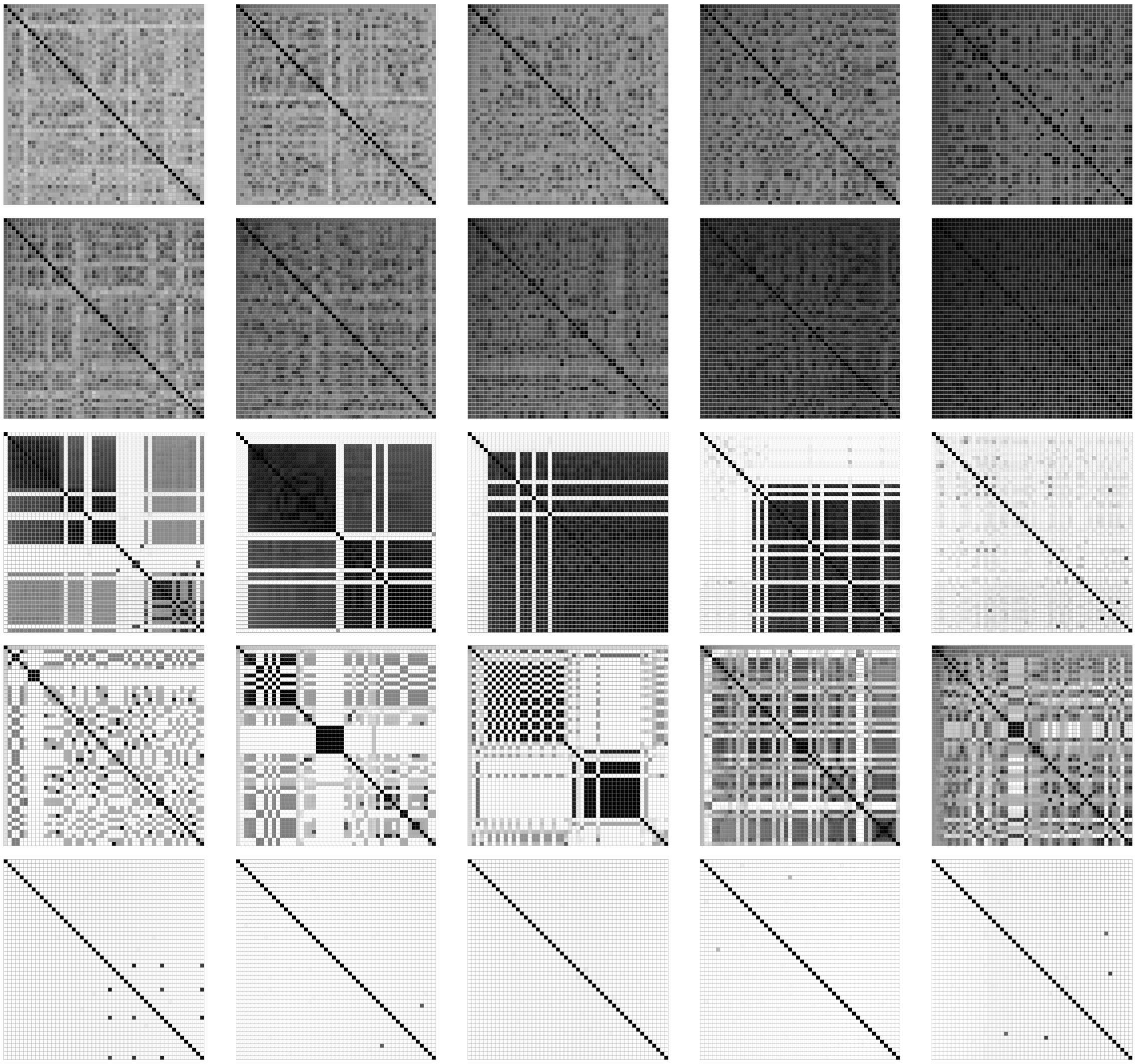


Figure 7: Pattern grids for each of the five datasets, from top to bottom: Accidents, Chess, Kosarak, Mushroom, Retail. Low support on the left, high support on the right.

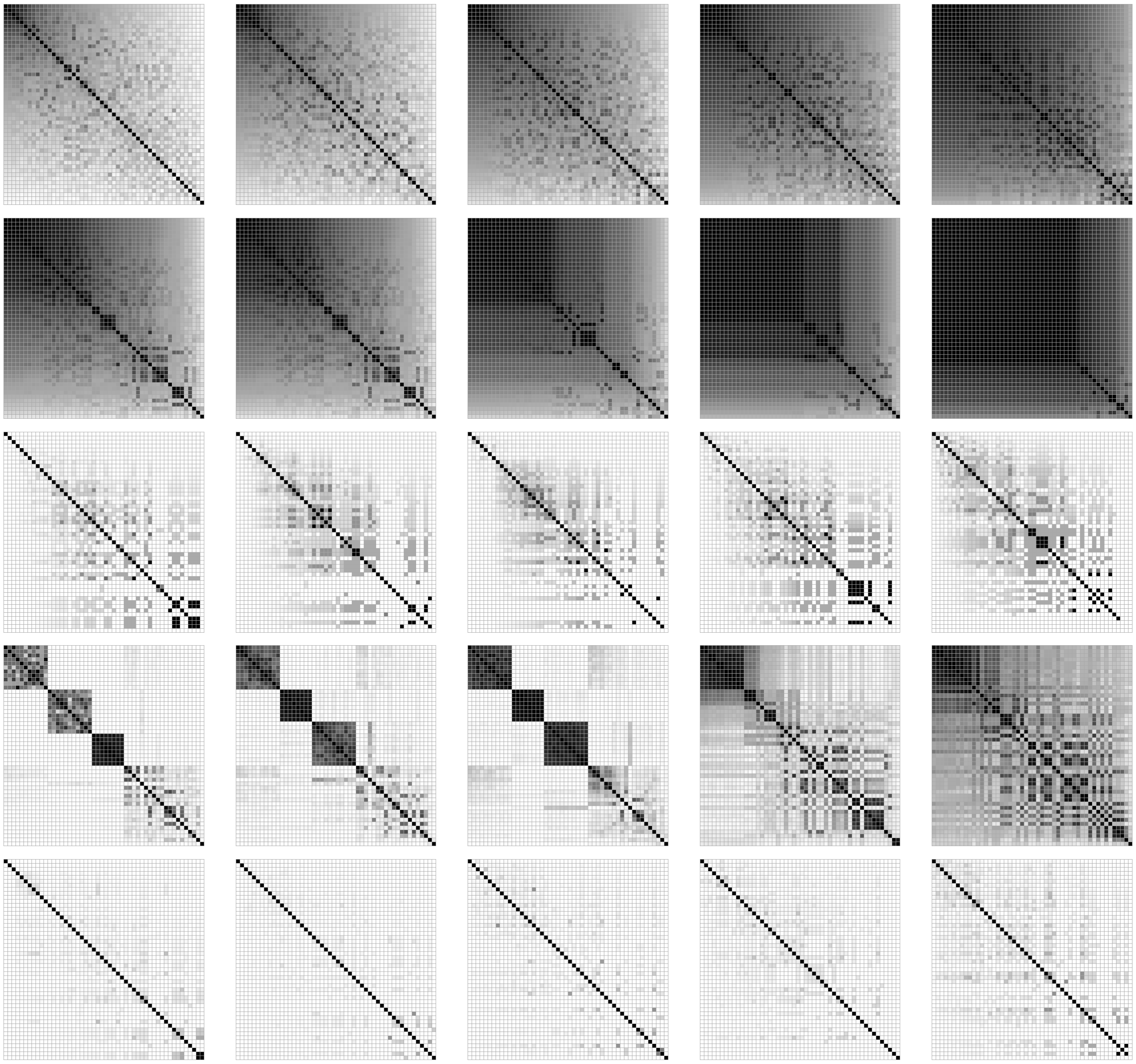
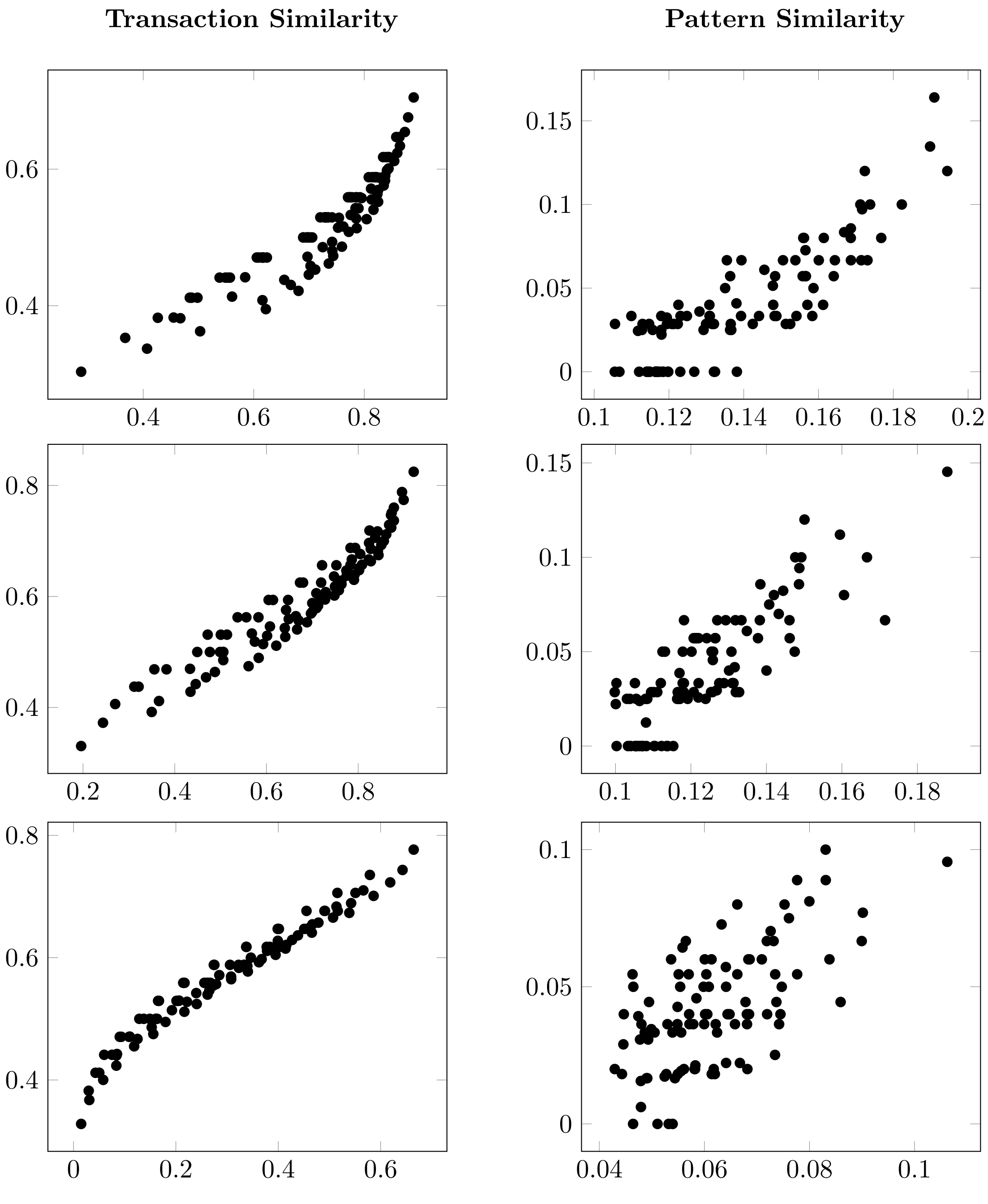


Figure 8: Transaction grids for each of the five datasets, from top to bottom: Accidents, Chess, Kosarak, Mushroom, Retail. Low support on the left, high support on the right.


Transaction Similarity

Pattern Similarity

Figure 9: Item Similarity (horizontal axis) vs Pattern and Transaction Similarity (vertical axis) for the Accidents dataset

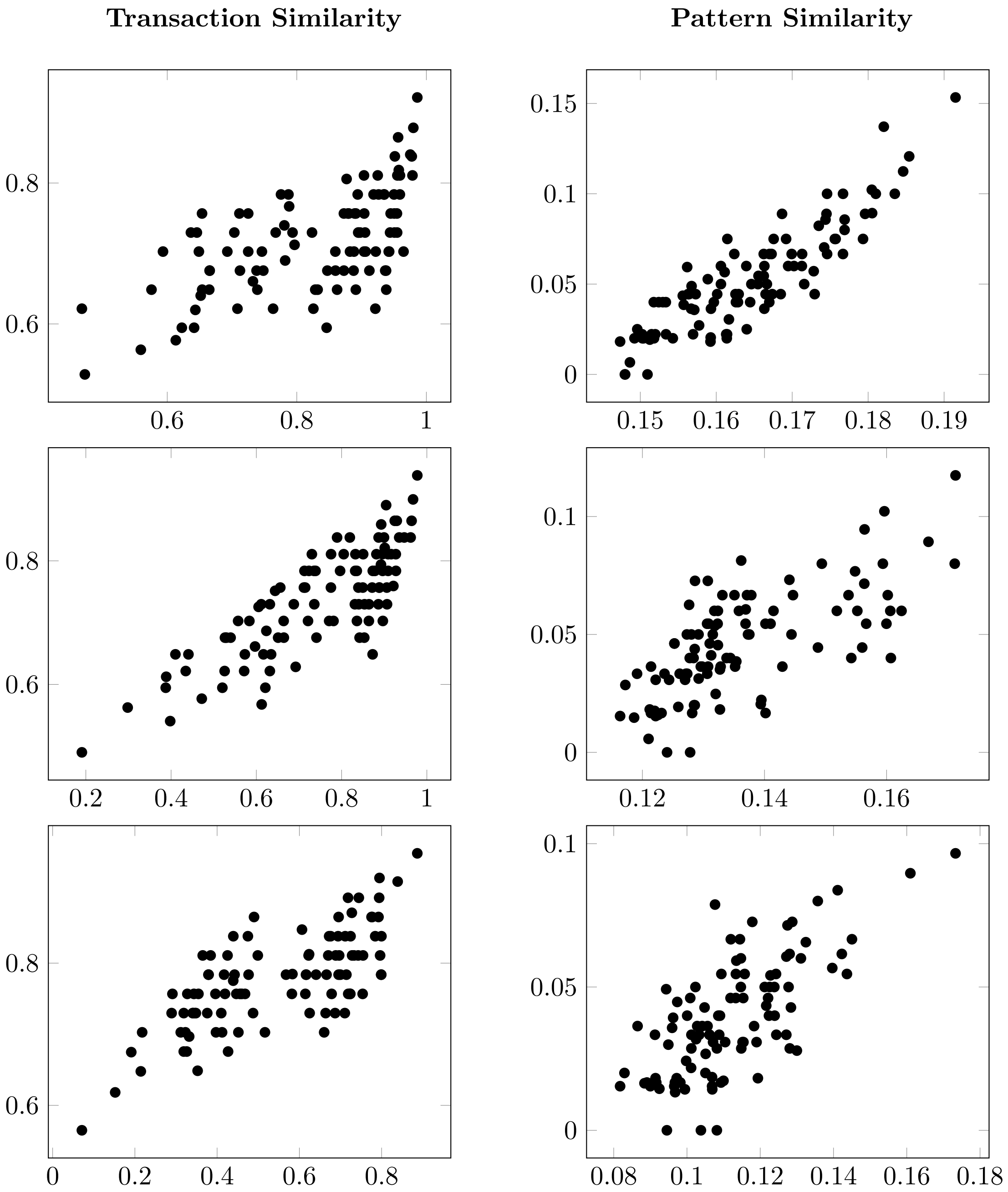


Figure 10: Item Similarity (horizontal axis) vs Pattern and Transaction Similarity (vertical axis) for the Chess dataset

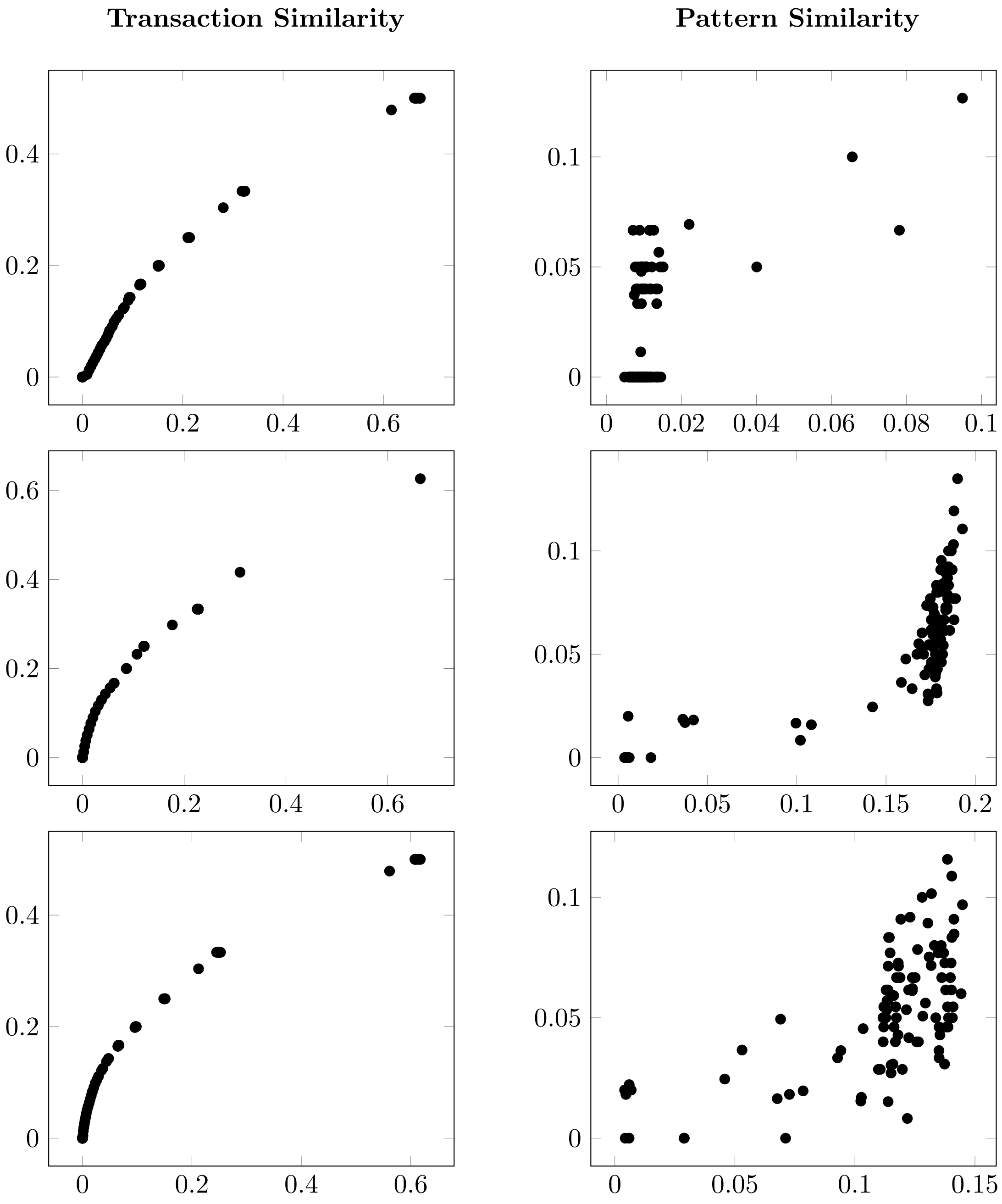


Figure 11: Item Similarity (horizontal axis) vs Pattern and Transaction Similarity (vertical axis) for the Kosarak dataset

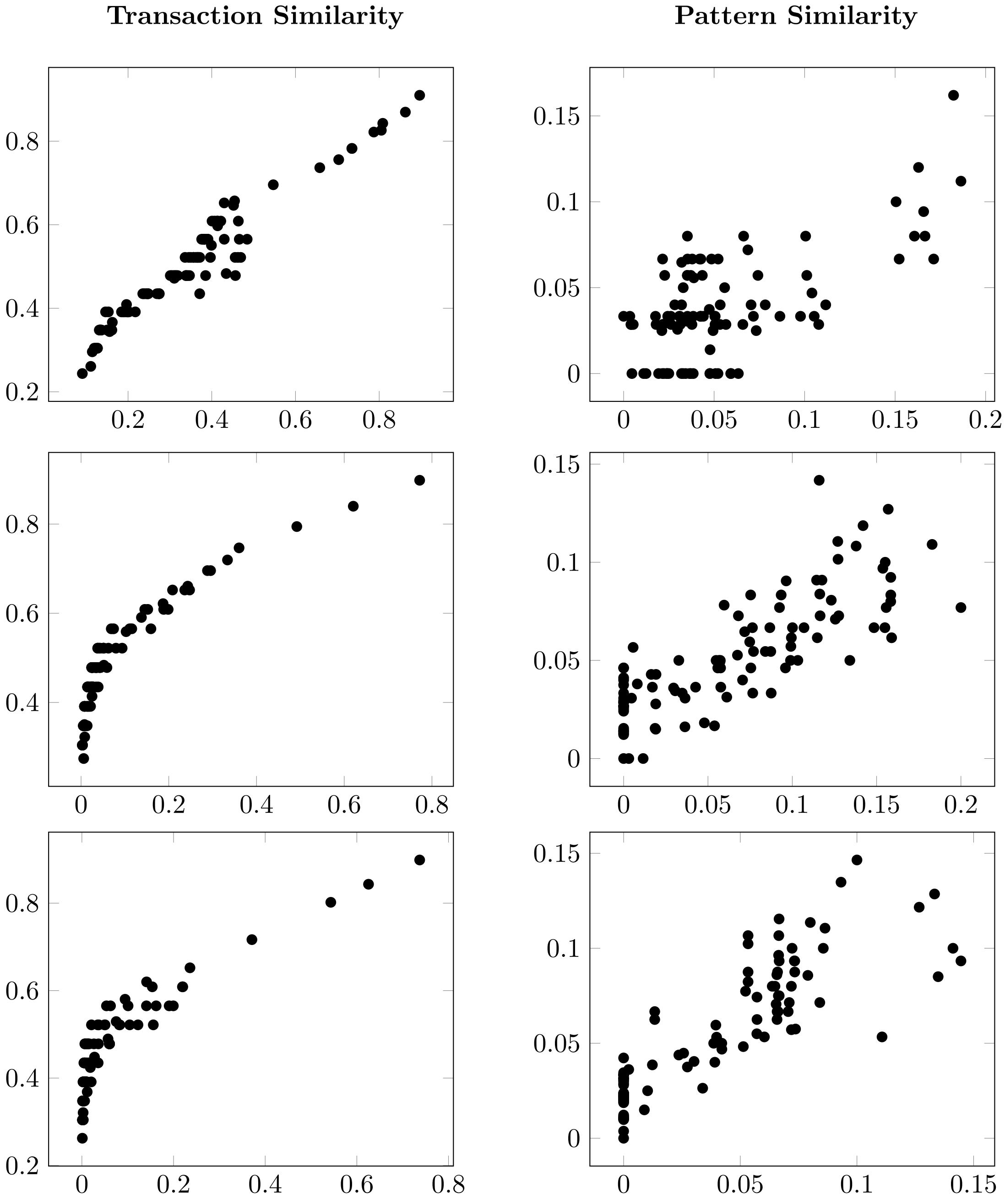


Figure 12: Item Similarity (horizontal axis) vs Pattern and Transaction Similarity (vertical axis) for the Mushroom dataset

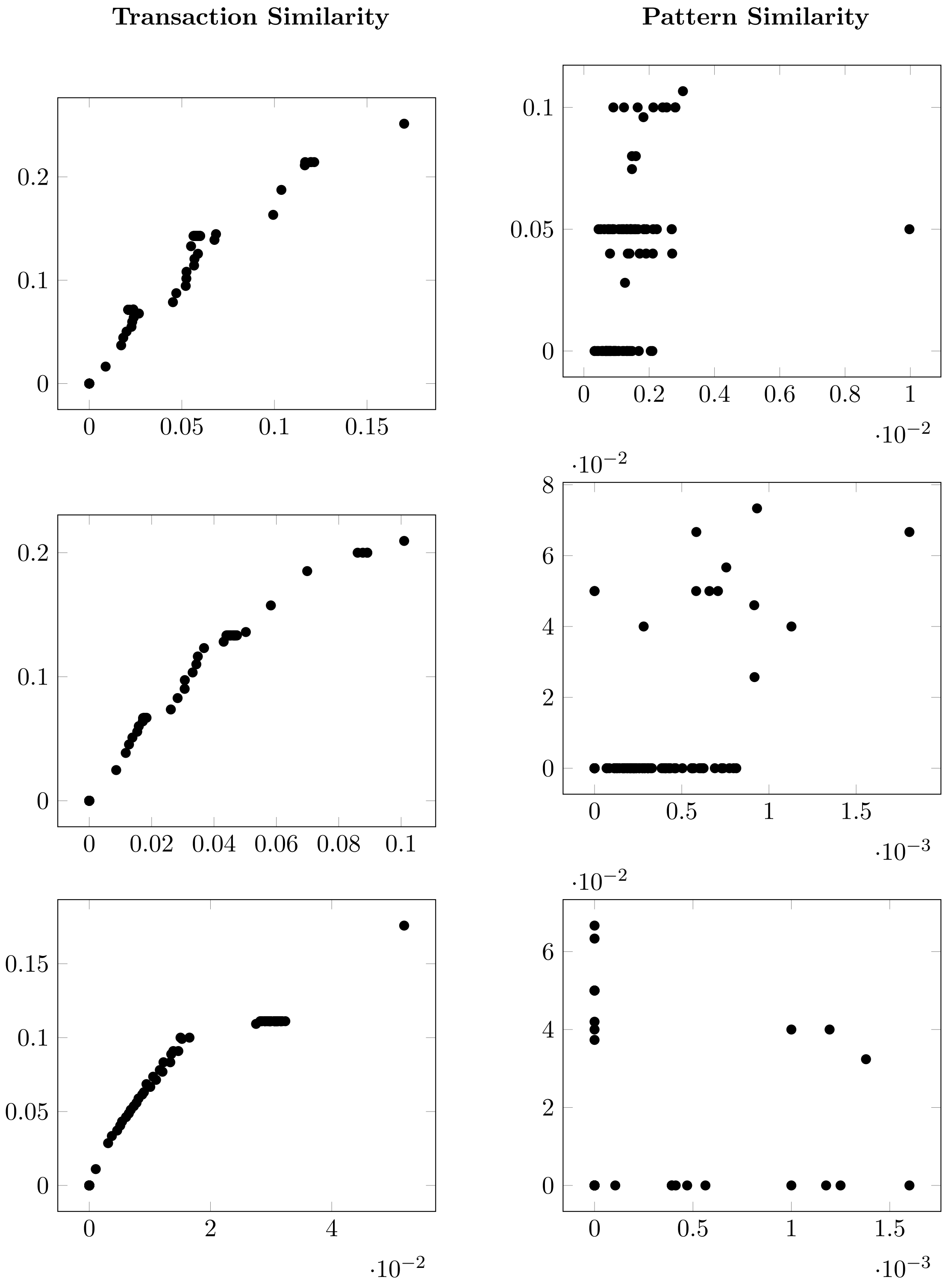


Figure 13: Item Similarity (horizontal axis) vs Pattern and Transaction Similarity (vertical axis) for the Retail dataset