Statistical Properties of Transactional Databases *

P. Palmerini
ISTI-CNR
Pisa, Italy
paolo.palmerini@isti.cnr.it

S. Orlando
Università Ca' Foscari
Venice, Italy
orlando@unive.it

R. Perego
ISTI-CNR
Pisa, Italy
raffaele.perego@isti.cnr.it

ABSTRACT

Most of the complexity of common data mining tasks is due to the unknown amount of information contained in the data being mined. The more patterns and correlations are contained in such data, the more resources are needed to extract them. This is confirmed by the fact that in general there is not a single *best* algorithm for a given data mining task on any possible kind of input dataset. Rather, in order to achieve good performances, strategies and optimizations have to be adopted according to the dataset specific characteristics. For example one typical distinction in transactional databases is between sparse and dense datasets. In this paper we consider Frequent Set Counting as a case study for data mining algorithms. We propose a statistical analysis of the properties of transactional datasets that allows for a characterization of the dataset complexity. We show how such characterization can be used in many fields, from performance prediction to optimization.

1. INTRODUCTION

Mining association rules in databases has received a great deal of attention in the last decade [6] [4]. This is due both to the straight applicability of the knowledge extracted, but also to the challenging performance issues posed by the problem complexity. If we limit ourselves to the most computational expensive part of the association mining problem, namely Frequent Set Counting (FSC), we can find tenths of algorithms proposed in the last years [1] [9] [11] [5] [7] [8]. They all adopt different strategies depending on the problem parameter values (namely the minimum support threshold s) and on the input dataset characteristics. This last feature turns out to be the most difficult to control.

One important property of the input dataset, is its *density*. The notion of density, although not yet formally defined in the literature, plays a crucial role in determining the best strategy for solving the FSC problem. A dataset is said to be dense, when most transactions tend to be similar among them: they have about the same length and contain mostly the same items. When dealing with such datasets, several optimizations can be very effective, like using compressed data structures both for the dataset and for itemsets representation. Conversely, for sparse datasets where transactions differ a lot one from another, in general it might be useful to apply some pruning technique in order to get rid of useless items and transactions. The *apriori* knowledge of a dataset density or sparsity can provide useful hints for global strategy decision in FSC algorithms. Another important issue is to determine the range of support threshold within which a *relevant* number of frequent itemsets will be found. When nothing is known about the dataset, the only option is to start mining with a high support value and then decrease the threshold until we find a number of frequent itemsets that fulfill our requirements. It would be nice if we could have an idea of the dataset behavior for all possible supports, before starting the actual - potentially expensive - computation.

In a recent work [3] Goethals *et al.* have analytically found a tight upper bound for the number of candidates that can be generated during the steps of a level-wise algorithm for FSC. From the knowledge of the number of frequent patterns found at step k, it is possible to know an upper bound for the candidates that will be generated at step $k+1$. This permits to estimate with a good level of accuracy the maximal pattern length, i.e. how many steps will be performed by the algorithm. Such knowledge is used by the authors to postpone the actual counting of candidates as much as possible, thereby limiting the number of database scans, without the risk of a combinatorial explosion in the number of candidates. Rather than focusing on the characteristics of level-wise FSC algorithms, namely candidate generation, the problem we want to address in this paper is that of finding a good characterization of a dataset complexity, in terms of the number of patterns satisfying a given support threshold, and of its density, in order to apply adequate optimization strategies.

In [10], Zaki *et al.* study the length distribution of frequent and maximal frequent itemsets for synthetic and real datasets. This characterization allows them to devise a procedure for the generation of benchmarking datasets that reproduce the length distribution of frequent patterns. To the best of our knowledge, there are no previous work addressed at a characterization of a dataset density and at how this characterization can be used for optimization purposes.

In this paper, we will try to answer the questions of whether a dataset is dense or sparse and which is the support range of potential interest for a given dataset. We introduce a

*We would like to acknowledge F. Silvestri and prof. M. J. Zaki for continuous and useful discussions.

SAC '04, March 14-17, 2004, Nicosia, Cyprus

macroscopic property of datasets - that we call H - whose behavior can be used to characterize the dataset density, thereby allowing for specific optimizations to be effectively applied. Such quantity allows also to easily build a model from which it is possible to estimate the number of frequent patterns contained in the dataset for any value of s, without executing the FSC algorithm. In Section 2 we introduce the definition of H and formalize the notion of density. In Section 3 we show how H can be effectively used in FSC algorithms for strategy decision and support range determination.

2. DEFINITIONS

We introduce a formal definition of the notion of dataset density, i.e. of how much the transactions inside the dataset resemble one with another. Intuitively, the more dense a dataset is, more its transactions will differ only for very few items.

The following two limit cases give an idea of the intuitive meaning of density. The maximum density that a dataset can have corresponds to all the transactions being identical. On the other hand, minimum density corresponds to each transaction containing only one single item and each item appearing in only one transaction. In Figure 1 we represent the dataset in binary format, where to each of the N transactions we associate a row in a matrix whose elements are 0 or 1 according to whether the corresponding item - ranging from 1 to M - is present or not in the transaction. We see that maximum density - Fig. 1 (a) - corresponds to a matrix whose elements are all 1, and minimum density - Fig. 1 (b) - to the identity matrix, with ones only along the main diagonal. In the following we will respectively refer to these two extreme datasets as $\mathcal{D}$ (for dense) and $\mathcal{S}$ (for sparse).

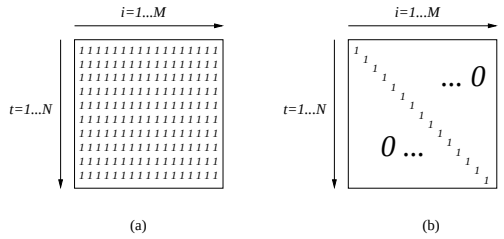

Figure 1: Limit cases of maximum (a) and minimum (b) density for transactional datasets in binary format.

Real datasets of course in general exhibits an intermediate behavior with respect to the limit cases. In Figure 2 we plotted the bitmap representation[1] of some real datasets and a synthetic one.

From this simple representation it is already possible to isolate some qualitative feature of a dataset. It is for example evident that BMS_View_1 and the synthetic dataset (Figure 2 (c) and (d) respectively) are more sparse than the others. Nevertheless, while the synthetic dataset shows a regular structure, with an almost uniform distribution of ones in the bitmap, BMS_View_1 exhibits a more complicated internal structure, with two extremely long transactions and a few items appearing in almost all transactions (two almost full rows and columns respectively).

[1]The bitmap is obtained by evaluating the number of occurrences of each distinct item in a subset of N/M transactions and assigning a level of gray proportional to such count.

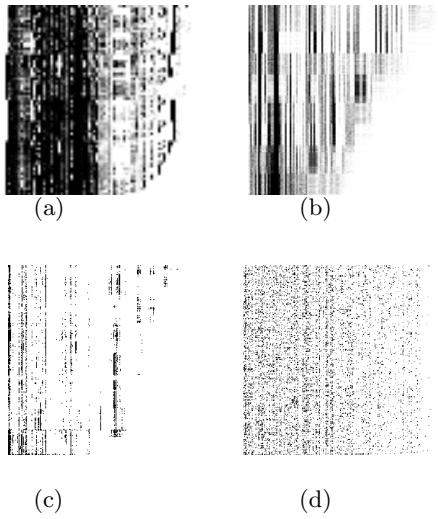

Figure 2: Bitmap representation of datasets: (a) chess; (b) mushroom; (c) BMS_View_1; (d) T25I10D10K. All images have been rescaled to the same squared size.

We would like to be able to characterize datasets with a measurable quantity from which it is possible to tell how dense a dataset is, i.e. if it is closer to $\mathcal{D}$ or to $\mathcal{S}$. The simplest choice we can make, is to consider the fraction of 1's in the dataset matrix, which is 1 for $\mathcal{D}$ and $1/M$ for $\mathcal{S}$. Such definition is not sufficient to capture all the interesting dataset features from the point of view of FSC, since two transactions with the same number of items will bring the same contribution to the overall density, regardless of how they actually resemble one another.

A more interesting behavior is exhibited by the average support of frequent items, plotted in Figure 3. For almost

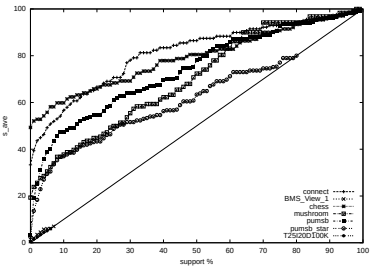

Figure 3: Average support

all values of the support threshold - along the x axis - dense datasets maintain an average support of frequent items that is sharply higher than the threshold, i.e. curves corresponding to dense datasets reside well above the $y = x$ line. On the other hand, for sparse dataset only a few items have support high enough to pass the threshold filter, even for low values of the threshold. We can use this qualitative analysis two define two classes of datasets. Dense datasets are characterized by an average support for frequent items that is *much* higher than the minimum support threshold for almost all supports. How *much* higher is a question that remains unsolved at this level of analysis.

Although we can qualitatively classify a dataset into sparse

vs. dense categories using the average support of frequent items, we still have the same problem stated above: transaction similarity is not taken into account and therefore we have little or no chance at all to have any hints on the frequent patterns contained in the dataset. To have a more accurate estimate of transaction similarity, we should measure the amount of information contained in each transaction. The information relevant to the FSC problem is the collection of itemsets contained in a transaction. Our idea is to consider the dataset as a transaction source and measure the entropy of the signal - i.e. the transactions - produced by such source. For a given length k, we define the following quantity:

$$H_k(s) = -\sum_{i=1}^{\binom{M}{k}} [\![p_i > s]\!]\, p_i \log p_i \qquad (1)$$

where M is the total number of distinct items in the dataset and p_i is the probability of observing itemset i, of length k, in the dataset, s is the minimum support threshold. The truth function $[\![expr]\!]$ which equals 1 if $expr$ is TRUE and 0 otherwise, is used to select only the frequent itemsets. The probabilities p_i are the normalized frequencies: $p_i = \frac{f_i}{\sum_j f_j}$ with f_i being equal to the number of transaction where the itemset appears, divided by the number N of transactions in the dataset.

The intuitive idea behind Equation 1 is that of considering the information contained a transaction source. By selecting only frequent itemsets, we consider the minimum support threshold influence on the problem complexity: the lower the minimum support, the harder the mining process.

Definition 1 holds for any itemset length k. Measuring $H_k(s)$ for all possible k corresponds to running an FSC algorithm. We therefore ask if it is possible to capture useful information on the dataset properties only considering small values for k.

We begin considering H_1, i.e. the entropy of single items, which is the simplest H_k to be evaluated. We consider the two limit cases $\mathcal{S}$ and $\mathcal{D}$ with no minimum support filter applied. In the first dataset $f_i = 1/M, \forall i$ so $p_i = 1/M, \forall i$ and $H_1(\mathcal{S}) = \log(M)$. For the dense dataset, $f_i = 1, \forall i$ so $p_i = 1/M \forall i$ and again $H_1(\mathcal{D}) = \log(M)$. Therefore using H_1 we are not able to distinguish between the two limit cases which are interesting for our purposes. This is due to the fact that with H_1 we cannot differentiate between the contribution of a transaction of, say, n_t items and n_t transactions each one with only one of the items of the first transaction. For example a dataset composed by one transaction $D = \{\{1, 2, 3\}\}$ and a dataset composed by three transactions $D' = \{\{1\}, \{2\}, \{3\}\}$ would have the same H_1.

The problem is that in H_1 we are not considering any correlation among items, i.e. we have no notion of transaction. The simplest level of correlation we can consider is that of single items correlations. We therefore go one step further and consider $k = 2$. Now for the sparse dataset $\mathcal{S}$ we have $f_i = p_i = 0, \forall i$, since the itemsets we consider are the $M(M-1)/2$ pairs of M items. So $H_2(\mathcal{S}) = 0$. On the other hand, for $\mathcal{D}$ we have $f_i = 1, p_i = 2/(M(M-1)) \forall i$ and $H_2(\mathcal{D}) = \log(M(M-1)/2)$. Using H_2 it is possible to distinguish between $\mathcal{D}$ and $\mathcal{S}$. Of course also H_2 fails to fully characterize a dataset. The following two datasets: $D = \{\{1, 2, 3, 4\}\}$ and $D' = \{\{1, 2\}, \{3, 4\}\}$ are again indistinguishable looking only at the pair occurrences, as in H_2, and H_3 should be considered instead.

In Figure 4 we plotted the measured value of $H_2(s)$ for different supports and different datasets. We used publicly available datasets [2] which are often referred to as *de-facto* standards for the benchmarking of FSC algorithms. It is therefore possible to identify two different qualitative behaviors. A subset of all the datasets considered, while increasing s, only a little variation in H_2 is observed. For another subset of the datasets considered this is not true. This second group exhibits a rapid decay of H while increasing s.

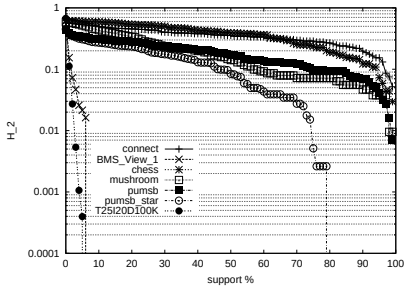

Figure 4: Entropy of known datasets.

It is important to notice that the variation of H_2 is significant more than its absolute value. In other words in order to compare the characteristics of two datasets, it is necessary to evaluate $H_2(s)$ for several supports.

We can conclude this part of our analysis by stating that datasets whose pair-entropy remains almost constant (in logarithmic scale) will be characterized by a higher density. We call such datasets *dense* and classify them in the $\mathcal{D}$ class. Conversely, the other class will be the one of *sparse* $\mathcal{S}$ datasets.

We conclude this section with an observation on the computational cost of such measure. The evaluation of H_1 only involves single items frequnecies, therefore a single dataset scan is required while the amount of memory is of $O(M)$. TO evaluate H_2 it is necessary to know the pair frequencies, which implies a further dataset scan or, if enough memory is available - $O(M^2)$ - everything can be evaluated in the first scan.

3. USING ENTROPY

We show in this section how the measure of H can effectively be used to improve the performance of FSC algorithms (Sec. 3.1) or to give hints on the potentially interesting support range for a given dataset (Sec. 3.2) or, finally, to estimate the quality of a sampling algorithm (Sec. 3.3).

3.1 Strategy decisions

Knowledge of a dataset density can be of great importance for improving the performance of an FSC algorithm. Different optimization strategies can be adopted according to whether the dataset is known to be dense or sparse. We embedded such strategies in our FSC algorithm [8] which dynamically adapts its behavior according to the dataset statistical properties. The results obtained show that its possible and effective to apply specific optimizations depending on the dataset features. In the case of sparse datasets, it is useful to prune the dataset and remove infrequent items

[2] `http://kdd.ics.uci.edu`

Dataset	$s_{min} \sim s_{max}$ (%)
connect	$45 \sim 90$
chess	$15 \sim 70$
mushroom	$1 \sim 20$
pumsb	$60 \sim 95$
pumsb_star	$25 \sim 60$
T25I20D100K	$0.55 \sim 1$
BMS_View_1	$0.06 \sim 0.4$

Table 1: Support range for each dataset in Figure 5. Ten support values were considered within each range.

and short transactions. When mining dense dataset, on the other hand, pruning can turn out to be too expensive with respect to the benefits adduced. Compact data structures can rather be adopted in order to increase locality in itemset support counting and to reduced the amount of memory required to store them. Since these strategies are already discussed elsewhere [8], although we did not use the same statistical analysis proposed in this paper, we are not going to enter into any further detail here.

3.2 Support range determination

One interesting feature of H is that it allows to determine the support range of interest for a given dataset. When applying an FSC algorithm to an unknown dataset, one option is to starts with a very high support threshold and then keep running the FSC algorithm while lowering the threshold, until a satisfactory number of frequent patterns is found. This approach has the strong limitation that we do not know in advance how long can the computation last, even for high minimum supports, and, most importantly, it is not possible to determine how many patterns the FSC algorithm will find for a given support until we run it. Since H is related to the correlation among transactions, one could think that its variation can be related to the variation in the number of patterns found. In fact, this conjecture is confirmed by experimental verification.

We considered the total number of frequent itemsets found for different datasets and supports, and measured the corresponding value of $H_2(s)$. For each dataset we considered 10 different support values in different ranges, reported in Table 1.

We found that the logarithm of the total number of frequent patterns is linearly correlated with $H_2(s)$. In Fig. 5 (a) a linear fit obtained by a minimum square regression, is superimposed to each data series.

Therefore, if we wanted to know how many frequent patterns are contained in a dataset for a given support threshold, we could run the FSC algorithm with a few values of high supports (corresponding to small execution times), and then extrapolate the unknown number of frequent patterns for the requested support. Conversely, from the same linear regression, we could determine which is the support that will produce a given number of frequent patterns. In Figure 5 (b) we plot the average error on the estimate of the number of frequent itemsets using an increasing number of points from the plot in Figure 5 (a). For each dataset, we performed a linear fit taking a variable number of points from the curves in Fig. 5 (a), starting from the highest supports, i.e lowest values for $H(s)$. Then we evaluated the average error obtained when estimating the number of frequent patterns for the rest of the points in the plot. More precisely, if we have n different values of $H(s)$ (in our case $n = 10$), and the n corresponding values for the total number of frequent patterns $|F|$, we perform the linear fit for j ($j \geq 3$) of such $(H, \log(|F|))$ pairs. From the fit parameters we can estimate the values of the remaining $n-j-1$ number of frequent patterns $\overline{|F|}$. The error of the estimate, as a function of j, is defined as:

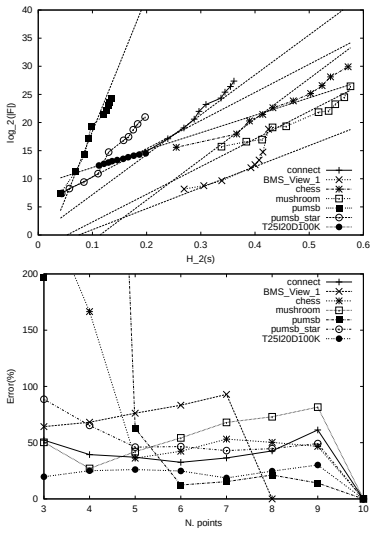

Figure 5: Total number of frequent patterns found versus the entropy (a) and the error obtained on the estimate of the total number of candidates (b)

$$error(j) = \frac{1}{n-j-1} \sum_{i=j+1}^{n-1} \frac{|F|_i - \overline{|F|}_i}{|F|_i} \cdot 100\% \qquad (2)$$

From Figure 5 (b) we can see that for most datasets already for four points the errors are lower than 40%, which permits to have a reasonable confidence when predicting the total number of frequent itemsets. In particular, by running the FSC algorithm with four values of s - that can be chosen in order to minimize the execution time - we are able to predict the total number of frequent patterns found for any value of s, within a 40% confidence.

3.3 Sampling

A common problem that arises in datamining when dealing with huge amount of data is that of obtaining an approximate result by applying an algorithm on a sample of the input dataset. In this case it is of interest to determine how accurate is the knowledge extracted from the sample. Several sampling algorithms have been proposed with variable level of accuracy [2] [12], yet the problem remains of determining whether a sample is a good representative of the entire dataset or not, without running the mining algorithm.

We argue that a sample whose entropy, as given by Eq. 1, is similar to the entropy of the entire dataset, will more likely produce the same amount of frequent itemsets. We show how the results from a set of experiments on synthetic and real datasets, confirm this conjecture. We define:

$$\Delta O_s = \sum_{k}^{k_{max}} \frac{(|F_k| - |F_k^s|)}{\sum_{k}^{k_{max}} |F_k|} * 100\% \quad (3)$$

where k_{max} is the maximum between the maximal length of frequent itemsets in the sample and in the real dataset, F_k is the set of frequent itemsets of length k in the entire dataset and F_r^s is the same in the sample.

In terms of metrics 3, a good sample will have $\Delta O_s = 0$. In a series of tests, we show how the quality of a sample can be actually correlated with a variation in the entropy. In Figure 6 we plotted ΔO versus the relative variation of entropy ΔH, for 200 random samples of six datasets. A linear fit is superimposed to the experimental data, showing the effective correlation between the two quantities.

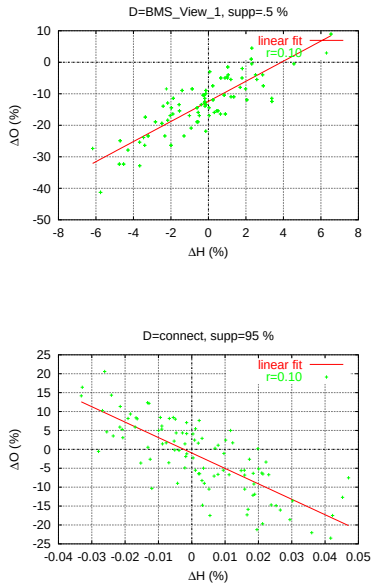

Figure 6: Correlation between variation in the output and variation in the entropy. Points refer to different random samples of the same dataset.

This result suggests that the entropy of a dataset is a good measure of the relevant statistical properties of a dataset. Even without running the complete FSC algorithm, we can use entropy to measure how representative a sample is. An interesting problem would be of finding an entropy preserving sampling algorithm.

4. CONCLUSIONS

Most of the computational complexity of Data Mining computations comes from the unknown amount of information to be extracted from the input datasets and to properties of the dataset which remain unknown until the mining algorithm is fully executed. This is the case for the well known dense/sparse classification of transactional datasets.

As a consequence, unpredictable execution times of the DM algorithms make it difficult to find general performance results. Moreover, the analyst is often forced to an explorative approach in the search of the support range of interest for each dataset.

We introduced a statistical property of a dataset, called H, that allows to formally define the notion of density. Since H is related to the order present in the dataset, we also demonstrated how it is possible to use such quantity for other purposes like determine the support range of interest for a given datasetor estimate the quality of a sample obtained by a sampling algorithm.

5. REFERENCES

[1] Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules. In *Proc. of the 20th Int'l Conference on Very Large Databases*, Santiago, Chile, September 1994.

[2] Bin Chen, Peter Haas, and Peter Scheuermann. A new two-phase sampling based algorithm for discovering association rules. In *Proceedings ACM-SIGKDD Conference*, Edmonton, Canada, July 2002.

[3] F. Geerts, B. Goethals, and J. Van den Bussche. A tight upper bound on the number of candidate patterns. In N. Cercone, T.Y. Lin, and X. Wu, editors, *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 155–162. IEEE Computer Society, 2001.

[4] Bart Goethals. *Efficient Frequent Itemset Mining.* PhD thesis, Limburg University, Belgium, 2003.

[5] J. Han, J. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 1–12, Dallas, Texas, USA, 2000.

[6] Hipp, J. and Güntzer, U. and Nakhaeizadeh, G. Algorithms for Association Rule Mining – A General Survey and Comparison. *SIGKDD Explorations*, 2(1):58–64, June 2000.

[7] J. Liu, Y. Pan, K. Wang, and J. Han. Mining Frequent Item Sets by Opportunistic Projection. In *Proc. 2002 Int. Conf. on Knowledge Discovery in Databases (KDD'02), Edmonton, Canada*, 2002.

[8] S. Orlando, P. Palmerini, R. Perego, and F. Silvestri. Adaptive and resource-awere mining of frequent sets. In *Proceedings of IEEE International Conference on Data Mining*, 2002.

[9] J. S. Park, M.-S. Chen, and P. S. Yu. An Effective Hash Based Algorithm for Mining Association Rules. In *Proc. of the 1995 ACM SIGMOD Int. Conf. on Management of Data*, pages 175–186, 1995.

[10] G. Ramesh, W. A. Maniatty, and M. J. Zaki. Feasible itemset distributions in data mining: Theory and application. In *22nd ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, June 2003.

[11] M. J. Zaki and K. Gouda. Fast Vertical Mining Using Diffsets. In *9th Int. Conf. on Knowledge Discovery and Data Mining, Washington, DC*, 2003.

[12] M. J. Zaki, S. Parthasarathy, W. Li, and Ogihara M. Evaluation of sampling for data mining of association rules. In *7th International Workshop on Research Issues in Data Engineering.*