Market Basket Analysis of Retail Data: Supervised Learning Approach

Gabriel Kronberger and Michael Affenzeller

Heuristic and Evolutionary Algorithms Laboratory
School of Informatics, Communications and Media
Upper Austria University of Applied Sciences, Campus Hagenberg
Softwarepark 11, 4232 Hagenberg, Austria
{gabriel.kronberger,michael.affenzeller}@fh-hagenberg.at

Abstract. In this work we discuss a supervised learning approach for identification of frequent itemsets and association rules from transactional data. This task is typically encountered in market basket analysis, where the goal is to find subsets of products that are frequently purchased in combination.
In this work we compare the traditional approach and the supervised learning approach to find association rules in a real-world retail data set using two well known algorithm, namely Apriori and PRIM.

1 Introduction

The aim of market basket analysis is to identify sets of products that are purchased frequently together. This information is relevant because it can be used to optimize shelf space, or to plan and control targeted marketing campaigns. Market basket analysis is especially relevant for e-commerce. In this area there is a practically unlimited shelf space and the set of offered goods and products can often be changed easily. In addition, targeted marketing campaigns can be implemented more easily online than in the real world. In the recent years the potential of market basket analysis and related data mining approaches has been fully recognized. Currently most of the large online retailers use market basket analysis and recommender systems to improve their volume of sales.

In this paper we revisit the idea of using a supervised learning approach for market basket analysis introduced and discussed in [4], [6]. First we give a formal description of the problem and introduce the terminology. Then we describe the supervised learning approach for the problem and in the subsequent sections we compare the two different approaches using a real-world retail dataset.

2 Formalization

Generally, the goal of market basket analysis is the identification of frequent itemsets (sets of products) in groups (baskets or transactions). Given a set of N transactions $T = (t_n)_{1..N}$, and a set of K items $I = (i_k)_{1..K}$, where $t_n \subset I, t_n \neq$

$\emptyset$, the goal is to find itemsets $\mathcal{I} \subset I$ that frequently occur in all transactions t_n. The support of an itemset $\mathcal{I}$ shown in Equation 1, is the number of transactions $t_j \subset T$ which contain all items in X. The frequency of an itemset $\mathcal{I}$ shown in Equation 2, is the probability that the itemset occurs in a transaction.

$$\text{support}(\mathcal{I}, T) = |\{t_j | t_j \in T, t_j \subset \mathcal{I}\}| \tag{1}$$

$$\text{frequency}(\mathcal{I}, T) = \frac{\text{support}(\mathcal{I}, T)}{|T|} = \Pr(\mathcal{I}) \tag{2}$$

Usually, the primary goal is not the identification of frequent itemsets but the identification of association rules. An aAssociation rule $X \Rightarrow Y$, where $X \subset I, Y \subset I, X \cap Y = \emptyset$ defines a set of products Y in the consequent that are frequently purchased together with the products in the antedecent X. An example for an association rule is: $\{\text{rum}, \text{mint}\} \Rightarrow \{\text{limes}\}$. Two frequently used metrics for association rules are the confidence of a rule $X \Rightarrow Y$ shown in Equation 3 and the lift shown in Equation 4. The lift of rule $X \Rightarrow Y$ is the relative number of observations of both itemsets X and Y in transactions T, relative to the expected number of observations if X and Y are independent.

$$\text{confidence}(X \Rightarrow Y, T) = \frac{\text{support}(X \cup Y, T)}{\text{support}(X, T)} = \Pr(Y|X) \tag{3}$$

$$\text{lift}(X \Rightarrow Y, T) = \frac{\text{confidence}(X \cup Y, T)}{\text{support}(Y, T)} = \frac{\Pr(X \wedge Y)}{\Pr(X)\Pr(Y)} \tag{4}$$

Traditionally, rules with high confidence and at the same time large support are considered as most interesting, because these rules concern a large fraction of all transactions and thus, any action taken to boost sales of items in these rules will have a large impact on the overall sales volume. However, it is reasonable to consider other metrics for interesting items, e.g. it could be interesting to search for rules which include highly priced items in the consequent.

The most well known algorithm for identification frequent itemsets and association rules is the Apriori algorithm [1]. It is considered as one of the top ten algorithms in data mining [8]. The main advantage of Apriori is that it scales to very large data bases with millions of items and transactions. Mining data sets of this size is non-trivial because it is not possible to keep occurrence counts of all frequent item pairs in memory. Apriori accomplishes this by minimizing the passes over the whole data set and keeping only a small fraction of all possible itemsets in memory. A heuristic is used to determine which itemsets to keep. Other well known algorithms are Eclat [9] and FP-growth [5] which also scale to large databases. A good survey of algorithms for frequent itemset mining is given in [2].

3 Supervised Learning Approach to Frequent Itemset Mining

The problem of identification of association rules can be transformed into a supervised learning problem as suggested in [6]. The potential advantage of the

transformation is that the well-developed supervised learning methodology and algorithms can be applied to market basket analysis. This was not easily possible in the beginning of research on frequent itemset mining because of the computational complexity, however *"[...] the increased computational requirements are becoming much less of a burden as increased resource become routinely available"* [6]. In this paper we pursue this line of argumentation and apply Aprioi and a supervised learning algorithm for bump hunting, namely the patient rule induction method (PRIM) [4], on a real-world retail data set for frequent itemset mining, and compare the results produced by both algorithms.

Given a matrix of N observations of K variables $X = (x_{i,j})_{i=1..N,j=1..K}$ and a vector of labels $Y = (y_i)_{1..N}$ for each observation, the goal of supervised learning is to find a function $f(x_1, ...x_K)$, that maps input values $(x_j)_{j=1..K}$ to labels. The two most frequently occurred supervised learning tasks are classification and regression. For classification tasks the label values y_i are discrete and usually nominal (e.g. positive/negative, malignant/benign, black/white). In contrast, in regression the label values are continuous. The challenge of supervised learning is that a function $f(x_1, ..., x_k)$ must be found that is accurate on the set of observations available for learning (or training), but more importantly is also generalizable to new observations.

Frequent itemset mining can be reformulated to a supervised learning problem by transforming the set of transactions T into an incidence matrix B that associates transactions t_i and items i_j.

$$B = (b_{i,j})_{i=1..N,j=1..K},\ b_{i,j} = \begin{cases} 1, i_j \in t_i \\ 0, \text{otherwise} \end{cases} \tag{5}$$

The incidence matrix contains the observed values for the input variables in the supervised learning problem. Each observation stands for one transaction and each variable stands for one possible item. Additionally, the target label vector Y must be defined. One possible scenario is to search a classifier to classify actually observed transactions and random transactions. This can be done by sampling another set of random transactions T_0 and incidence matrix B_0, where the probability of observing a given item is independent of the other items in the transaction. For classification the two incidence matrices B, B_0 are combined and the label vector is set to 1 for all observations in B and to 0 for all observations in B_0.

$$X = \begin{pmatrix} B \\ B_0 \end{pmatrix}, Y = \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \end{pmatrix} \tag{6}$$

If one is interested in association rules $X \Rightarrow Y$ for a particular given subset of items Y, namely searching the antecedent X for a predetermined consequent Y, then it is not strictly necessary to generate a random set of transactions. Instead, the label vector Y can be generated as shown in Equation 7.

$$y_i = \begin{cases} 1, & Y \subset t_i \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

In this situation, searching for a classifier means to identify the factors of the input matrix which increase the chance of observing the antecedent, which can be likened to the goal of association rules mining for predefined items. One notable difference is that in such a classifier it is principally possible to encode that a given item must not be present in a transaction, to increase the chance of observing the antecedent.

3.1 Bump Hunting

However, in association rule mining the primary concern is not to find a full classifier but to find regions of the input space which are more densely populated than other regions ("bump hunting"). In terms of frequent itemset mining an itemset is a rectangular box in the input space and the frequency of the itemset is the number of observations within the box over all observations. Thus, in the supervised learning approach of frequent itemset mining the goal is to find rectangular boxes in the K-dimensional space defined by X that contain large fractions of all observations. If X contains only binary elements, as is the case in market basket analysis, the for each dimension only three different boxes are possible: $[0, 1]$, $[0]$, or $[1]$.

One algorithm that can be used for this supervised learning task is the patient rule induction method (PRIM) [4]. PRIM is a heuristic method that greedily reduces the size of the box, which initially covers the whole input space, until either a lower threshold for the mean value of y in the box, or for the support (mass of the box) for the itemset, represented by the box, is reached. The mean values of boxes are calculated over the y values of all observations in the box. PRIM follows a two stage approach to find boxes. First the size of the box is reduced by *peeling*, and subsequently the size of the box is increased again by pasting. In the peeling stage the box is reduced in one dimension in each step, so that the reduction in fitness is minimal. In the pasting stage the size of the box is increased in one dimension in each step, so that the increase in fitness is maximized. The step sizes for size reductions in the peeling and pasting stage are important parameters of the algorithm controlling the patience of the algorithm. With small step sizes the convergence of the algorithm is slower, however, for large step sizes the danger of converging to a bad local optimum is higher.

4 Comparison of Apriori and PRIM

Apriori and PRIM are two rather different algorithms, which have been defined for different problem types and with different design goals. Thus, for certain problems it is more natural to choose Apriori over PRIM while for other problems PRIM is the better choice.

Apriori is specialized for huge databases of transactions with a relatively small number of items in each transaction (sparse incidence matrices). Apriori can only handle binary input values, either a transaction contains a given item or not. It is not possible to handle continuous values (e.g. multiple instances of

the same product), or discrete nominal values (e.g. variants of a given product, an item from a group of products). The only way to handle such situations is through introduction of virtual items for sub-ranges of continuous variables or subsets of discrete values. Aprior is deterministic and finds all rules with support larger than the given lower threshold. A problem of Apriori is that the computational requirements of the algorithm grow exponentially with a decreasing lower support threshold. Thus, it is infeasible to search for very specialized rules with have small support even though they could be interesting as they have large confidence or concern very valuable items. Finding interesting rules in the large set of possible rules produced by Apriori is again a data mining challenge. Often a large subset of rules produced by the algorithm are trivial or well known.

In contrast, PRIM is specialized for data sets of continuous variables that fit entirely into RAM. The peeling and pasting phases of PRIM are well defined for continuous input spaces, however, the algorithm also works for boolean and nominal variables. PRIM is a heuristic but deterministic algorithm and produces a sequence of models with gradually decreasing support. Because it is uses a greedy heuristic approach it is not guaranteed that the algorithm finds the optimal box. Before each peeling and pasting step PRIM must calculate the difference of the box mean for each size reduction or increment over all columns of the input matrix X, thus it does not scale to problems with many dimensions (=large number of items). The number of observations is not that critical as long as all observations fit into memory because the algorithm iterates many times over all observations. An advantage of PRIM is that it can also be used to search for rules with small support as it only optimizes one candidate box and does not track all possible boxes in memory. However, the problem of rules with small support is that the confidence of such rules cannot be estimated accurately because of the small data sample. This leads to the well known issue of overfitting. The large confidence of a rule with small support but large confidence on the training set could be a statistical fluke and such rules are not necessarily generalizable to new observations.

To summarize, Apriori is the better choice for sparsely populated input matrices with binary elements, while PRIM is better suited for densely populated matrices with continuous variables.

5 Experiments

The software used for the comparison is R, the free software environment for statistical computing and graphics [7] (version 2.13.0). We used an implementation of the Apriori algorithm by Christian Borgelt wrapped in the R package *arules* (version 1.6-0) and the implementation of PRIM provided in the R package *prim* (version 1.11.0).

The dataset used in the experiments is the *retail* dataset published by Tom Brijs [3][1]. The dataset is relatively small and contains 88163 receipts from 5133

[1] The data set and all scripts can be downloaded from `http://dev.heuristiclab.com/AdditionalMaterial/`

customers collected over approximately five months from a supermarket store in Belgium carrying 16470 stock keeping units (SKU). The retail dataset is frequently used in benchmarks of market basket analysis algorithms. Figure 1 shows the a graphical plot of an excerpt of the incidence matrix for the retail data set.

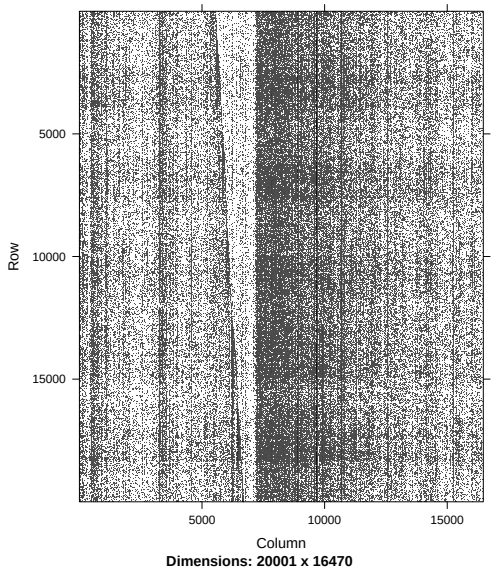

Fig. 1. Dot plot of incidence matrix for the retail dataset. The transactions are shown on the y-axis, the items are given on the x-axis.

As seen in Figure 1 some items shown on the x-axis are purchased very frequently while other items are purchased only seldom, interestingly there is also a seasonal variation that can be seen by variations in the density of regions over the transactions shown on the y-axis.

6 Results

First of all we imported the retail data set and executed Apriori to generate all rules with a support threshold of 0.01 and a confidence threshold of 0.6. The result is a set of 84 rules, where the top ten rules by support are given in table 1.

Next we apply the PRIM algorithm on the same data set searching for association rules with the consequent "39". Because of the large number of items in the retail data set it is infeasible to try the algorithm on the full data set. Instead we prepared incidence matrices containing only items with a frequency larger than 5%, 1% and 0.5% in order to reduce the number of dimensions. While this

id	lhs	rhs	support	confidence	lift
1	48	⇒ 39	0.33	0.69	1.20
2	41	⇒ 39	0.12	0.76	1.32
3	38	⇒ 39	0.11	0.66	1.15
4	41	⇒ 48	0.10	0.60	1.26
5	41,48	⇒ 39	0.08	0.81	1.42
6	39,41	⇒ 48	0.08	0.64	1.35
7	38,48	⇒ 39	0.06	0.76	1.33
8	32,48	⇒ 39	0.06	0.67	1.16
9	32,39	⇒ 48	0.06	0.63	1.33
10	38,41	⇒ 39	0.03	0.78	1.36

Table 1. Top ten association rules identified by Apriori for the retail data set sorted by support.

reduction helps significantly to decrease the runtime of the algorithm this also means that rules with smaller support than the threshold cannot be identified. However, since we are mostly concerned about rules with large support this is no issue. For PRIM we used a support threshold of 10% and a step size of 0.5 for peeling and pasting. The resulting boxes are shown in Table 2 the rule for the 0.5% frequency subset is not shown because it is rather long. Interestingly the algorithm produced tendentially rules that state that a set of given items must not be present in a transaction. Another relevant observation is that the rules identified by PRIM have very small lift, which means that the frequency of observations including item 39 in the box is almost the same as the overall frequency of item 39 in all observations. Thus, the information gain from the rule is only small. It should be noted that the most frequent items in all three subsets are 32, 38, 39, 41, 48, and 65. The first rule in Table 2 states that it doesn't matter which items out of this set occur in the transaction as long as item number 32 does not occur the frequency of transactions matching this rule is 57%.

freq threshold	lhs	rhs	support	confidence	lift
5%	!32	⇒ 39	0.82	0.57	1
1%	!31, !32, !101, !117, !123, !301, !548, !592, !1004	⇒ 39	0.74	0.58	1.01

Table 2. Boxes identified by PRIM for the retail data set.

7 Summary

In this paper we discussed the task of frequent itemset mining and the related task of identifying association rules typically encountered in market basket analysis. We discussed the supervised learning approach of frequent itemset mining

and compared a bump hunting algorithm (PRIM) to the well known Apriori algorithm, which is specialized for this task. In summary, both algorithms are principally suited for frequent item mining, however, Apriori is especially tuned for large data bases and binary variables, in contrast PRIM can be applied easily to data sets with continuous variables and also allows to search for itemsets with small support.

Acknowledgments This work mainly reflects research work done within the Josef Ressel-center for heuristic optimization "Heureka!" at the Upper Austria University of Applied Sciences, Campus Hagenberg. The center "Heureka!" is supported by the Austrian Research Promotion Agency (FFG) on behalf of the Austrian Federal Ministry of Economy, Family and Youth (BMWFJ).

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proceedings of the 20th VLDB conference. pp. 487–499 (1994)
2. Bodon, F.: A survey on frequent itemset mining. Tech. rep., Budapest University of Technology and Economic (2006)
3. Brijs, T., Swinnen, G., Vanhoof, K., Wets, G.: Using association rules for product assortment decisions: A case study. In: Knowledge Discovery and Data Mining. pp. 254–260 (1999)
4. Friedman, J.H., Fisher, N.I.: Bump hunting in high-dimensional data. Statistics and Computing 9, 123–143 (1999)
5. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Mining and Knowledge Discovery 8, 53–87 (2004)
6. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning - Data Mining, Inference, and Prediction. Springer (2009), second Edition
7. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2011), `http://www.R-project.org`, ISBN 3-900051-07-0
8. Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z.H., Steinbach, M., Hand, D.J., Steinberg, D.: Top 10 algorithms in data mining. Knowledge Information Systems 14, 1–37 (2007)
9. Zaki, M.J.: Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engineering 12(3), 372–390 (2000)