Mining Optimal Item Packages using Mixed Integer Programming

N R Achuthan[1] Raj P. Gopalan[2] Amit Rudra[3]

[1]Department of Mathematics and Statistics
Curtin University of Technology, Kent St, Bentley WA 6102, Australia
archi@maths.curtin.edu.au
[2]Department of Computing
Curtin University of Technology, Kent St, Bentley WA 6102, Australia
raj@cs.curtin.edu.au
[3]School of Information Systems
Curtin University of Technology, Kent St, Bentley WA 6102, Australia
Amit.Rudra@cbs.curtin.edu.au

Abstract. Traditional methods for discovering frequent patterns from large databases are based on attributing equal weights to all items of the database. In the real world, managerial decisions are based on economic values attached to the item sets. In this paper, we introduce the concept of the value based frequent item packages problems. Furthermore, we provide a mixed integer linear programming (MILP) model for value based optimization problem in the context of transaction data. The problem discussed in this paper is to find an optimal set of item packages (or item sets making up the whole transaction) that returns maximum profit to the organization under some limited resources. The specification of this problem opens the way for applying existing and new MILP solution techniques to deal with a number of practical decision problems. The model has been implemented and tested with real life retail data. The test results are reported in the paper.

1. Introduction

With the proliferation of data available to an organization from day to day operations, the prospect of finding hidden nuggets of knowledge has greatly increased [19]. Traditional inventory systems help a retailer to keep track of what items are required to be stocked and when to replenish specific items. The issue these days is not just replenishing the stock on the shelves but also to group them according to their perceived association with items that attract the attention of the customer. Using past sales of frequent items and the association among them can be determined efficiently by current algorithms. The methods for finding the frequent patterns involve different types of partial enumeration schemes where all items are given equal importance. However, in most business environments, items are associated with varying values of price, cost, and profit. So, the relative importance of items differs significantly. Kleinberg et al. [1] noted that frequent patterns and association rules extracted from real life data would be of use to business organizations only if they are addressed within the microeconomic context of the business. Brijs et al. [2] suggest that patterns in the data are interesting only to the extent to which they can be used in the decision making process of the enterprise. For example, the management of a supermarket may be interested in selecting packages of items that generate the maximum profit and requires physical storage space within certain limits. Another example is finding association rules where the items are most profitable or have the lowest margin.

Many such real-world problems can be expressed as optimization problems that maximize or minimize a real valued function. In this paper, we will focus on one such optimization problem in the context of transaction data and refer to it as **value based frequent item packages problem**. A package consists of items that are usually sold together. The aim is to find a set of items that can be sold as part of various packages to realize the maximum profit overall for the business.

Data mining research in the last decade has produced several efficient algorithms for association rule mining [3] [4] [5], with potential applications in financial data analysis, retail industry, telecommunications industry, and biomedical data analysis. However, literature on the use of these algorithms to solve real-world problems is limited [2]. Ali et al [6] reported the application of association rules to reducing fall-out in the processing of telecommunication service orders. They also used the technique to study associations between medical tests on patients. Viveros et al [7] applied data mining to health insurance data to discover unexpected relationships between services provided by physicians and to detect overpayments.

Most of the data mining algorithms developed for transaction data give equal importance for all the items. However, in a real business, not all the items are of equal value and many management decisions are made based on the money value associated with the items. The value may be in terms of the profit made or cost incurred or any other utility function defined on the items. Recent works by Aumann and Lindel [18] and Webb [17] discuss the quantitative aspects of association rules and tackle the problem using a rule based approach. More recently, Brij et al [2] developed a zero-one mathematical programming model for determining a subset of frequent item sets that account for total maximum profit from a pre-specified collection of frequent item sets with certain restrictions on the items selected. They used this model for the market basket analysis of a supermarket. Demiriz and Bennett [8] have successfully used similar optimization approaches for semi-supervised learning.

Mathematical programming has been applied as the basis for developing some of the traditional techniques of data mining such as classification, feature selection, support vector machines, and regression [9] [8]. However, these techniques do not address the value based business decision problems arising in the context of data mining and knowledge discovery. To the best of our knowledge, except for [2], mathematical modeling approach to classes of real world decision problems that integrate patterns discovered by data mining has not been reported so far. In this paper, we address this relatively unexplored research area and propose a new mathematical model for some classes of the value based frequent item packages problem. We contend that frequently occurring and profitable baskets are of greater importance to the retailer than subsets of transactions. The items that occur in a transaction can be packaged together or alternatively sold as individual items. We consider the expected minimum revenue, minimum and maximum number of items in the optimal item packages, and storage constraint pertinent to a real life retailer.

The structure of the rest of this paper is as follows: In Section 2, we define relevant terms used in transaction data, frequent item sets and association rule mining. In Section 3, we consider a general version of an optimal item packages problem and present an integer linear programming formulation for the same. In Section 4, we illustrate the model by a sample profit optimal item packages problem; provide its MILP formulation and the result of processing it using commercial mathematical programming software (CPLEX). Finally, we conclude our paper in Section 5 providing pointers for further work.

2. Transaction Data - Notations and Definitions

Transaction data refers to information about transactions such as the purchases in a store, each purchase described by a transaction ID, customer ID, date of purchase, and a list of items and their prices. A web transaction log is another example in which each transaction may denote a user id, web page and time of access.

Let T denote the total number of transactions. Let I = {1, 2, …, N} denote the set of all potential items that may be included in any transaction and more precisely the items included in the t^{th} transaction may be denoted by I_t , a subset of I, where t ranges from 1 to T.

The **support** s of a subset X of the set I of items, is the percentage of transactions in which X occurs. A set of items X is a **frequent item set** if its support s is greater than or equal to a minimum support threshold specified by the user. An **association rule** is of the form **X ⇒ Y**, where X and Y are frequent item sets that do not have any item in common. We say that **X ⇒ Y** has **support** s if s% of transactions includes all the items in X and Y, and **confidence** c if c% of transactions containing the items of X also contains the items of Y. A valid association rule is one where the support s and the confidence c are above user-defined thresholds for support and confidence respectively. Association rules [10] [11] [12] identify the presence of any significant correlations in a given data set.

3. Optimal Item Packages Problem

Brijs et al [2] considered a market basket analysis problem for finding an optimal set of frequent item sets that returns the maximum profit and proposed a mixed integer linear programming (MILP) formulation of their problem. Their model proposed maximizing the profit function of frequent item set X, i.e.

$\max \sum_{X \in L}^{k} M(X) * P_X - \sum_{i \in L} Cost_i * Q_i$, where L is the set of all frequent item sets X; M(X) is gross sales margin generated by X; and $P_X, Q_i \in \{0,1\}$ are decision variables; subject to $\sum_{i \in L} \mathrm{Q_i} = ItemMax$, where i is a basic item and *ItemMax* is the maximum threshold set for the number of items in X.

We generalize their problem to include different types of resource restrictions and develop an integer linear programming formulation for the same. The **Optimal Item Packages Problem** (**OIPP)** is to choose a set of frequent item sets or what we term as item packages, so as to maximize the total net profit subject to conditions on maximum storage space for selected items and minimum total revenue from the selected frequent item sets. Our formulation of the problem is much more flexible compared to Brij et al's [2], as the model can adapt to not only different resource restrictions but also to various bounds on the number of items in the final selection list. For example, it can specify the minimum and maximum number of elements in the final solution.

3.1 Motivation for OIPP

Often, in many real-life businesses, a transaction consists of a specific set of items as a package, facilitating a purchase. In such instances, both the number of items and the particular items forming the package are fixed. For example, while buying a car, a customer's choice may be made easier by having a number of fixed packages offered by the supplier. In some other businesses, it may not make sense to separate any item from a given package; e.g. medical procedures, travel packages etc.

Alternatively, a vendor may be interested in finding out from previous sales as to which, if any, set of items exist that could be offered as a package. This packaging of items (or products) could potentially offer him certain amount of profit under a number of resource constraints. For instance, the resource constraints could be available stocking space, budget (minimum cost or maximum profit), quantity (that needs to be sold) etc. He may be further interested in doing a sensitivity analysis as to how far the resources can be stretched while the given solution remains optimal. Again, in another instance, the vendor may like to see how a change in a certain resource affects his profitability (for example, if he is able to organize a little more space for storage or invest a little more money). For a travel bureau, a constraint could be time-oriented resources (like, a travel consultant's time),

OIPP:

For a given database $\{a_{it} : 1 \le i \le N , 1 \le t \le T\}$, let $\{X_j : 1 \le j \le k\}$ be a pre-specified list of k frequent item sets. Let f_j and n_j respectively denote the number of transactions that exactly include X_j (i.e. $f_j = |\{t: I_t = X_j\}|$) and the number of items in X_j , $1 \le j \le k$. Let p_j denote the revenue made by the frequent item set X_j whenever X_j forms a transaction. Let c_i denote the cost incurred (per unit) while selecting item i, $1 \le i \le N$. Let s_i denote the storage space (in appropriate units) required per unit for item i whenever the item is selected. Furthermore, let S denote the total available storage space. Find, a subset $\hat{\mathrm{I}}$ of $\{i: 1 \le i \le N\}$ and a subset F of the set of frequent item sets $\{X_j : 1 \le j \le k\}$ such that they satisfy the following properties:

1. The number of items in $\hat{I}$ is bounded below and above by positive integers N_L and N_U respectively;
2. A frequent item set X_j is selected in F if and only if X_j is covered by $\hat{I}$, that is, $X_j \subseteq \hat{I}$;
3. The total storage space required for the selected items of $\hat{I}$ does not exceed the available space of S units;
4. The total revenue made by frequent item sets of F is at least Minrev ;
5. The net profit (total revenue – the total cost) is maximized.

We now provide an MILP model for the OIPP described above. Let y_i denote the 0-1 decision variable that assumes value 1 whenever item i is chosen. Let z_j denote the 0-1 decision variable that assumes value 1 whenever the frequent item set X_j is covered by the set of selected items, that is, by the set of items $\{ i: y_i = 1\}$.

(6) Lower and upper bound constraints: $N_L \le \sum_{i=1}^{N} y_i \le N_U$

(7) Occurrence constraint of X_j: $\sum_{i \in X_j} y_i - n_j z_j \ge 0, \qquad 1 \le j \le k$

(8) Item storage space constraint: $\sum_{j=1}^{k} \left(\sum_{i=1}^{N} b_{ij} s_i \right) f_j z_j \le S$

(9) Lower bound constraint on revenue: $\sum_{j=1}^{k} p_j f_j z_j \ge$ Minrev

(10) Restrictions on variables: $y_i = 0$ or 1, $z_j = 0$ or 1

(11) Objective function: Maximize $\sum_{j=1}^{k} p_j f_j z_j - \sum_{j=1}^{k} \left(\sum_{i=1}^{N} b_{ij} c_i \right) f_j z_j$

In this value based frequent item set problem the input information regarding X_1, …, X_k , p_j, f_j , s_i and c_i must be extracted through data mining of frequent item sets. For the above model (6) – (11), the constraints and the objective function may be validated as follows:

Let the set of selected items to cover all the selected frequent item sets be denoted by

$\hat{I} = \{ i: y_i = 1\}$. It is easy to see that $|\hat{I}| = \sum_{i=1}^{N} y_i$ and the constraint (6) provides the lower and upper bound restrictions on this number. The number of items common to the set $\hat{I}$ and the frequent item set X_j is given by $\sum_{i \in X_j} y_i$. Whenever $\sum_{i \in X_j} y_i = |X_j| = n_j$, the set $\hat{I}$ covers the frequent item set X_j . The constraint (7) ensures that the decision variable z_j is 1 if and only if the frequent item set X_j is covered by $\hat{I}$. In this case note that $F=\{ X_j : z_j = 1\}$ is the collection of frequent item sets selected. The storage space required by an item of the selected item set X_j is $\sum_{j=1}^{k} b_{ij}\, s_i\, f_j z_j$, where b_{ij} is a known constant that takes value 1 or 0 according as item i is in item set X_j or not for $1 \le i \le N$, and $1 \le j \le k$. The constraint (8) expresses the upper bound restriction on the available storage space viz. S. The contribution made by the frequent item set X_j to the profit may be expressed as $p_j f_j z_j$ where $z_j = 1$ if and only if X_j is covered by the set $\hat{I}$. The constraint (9) ensures a minimum revenue contribution from the set of all covered frequent item sets. The constraints of (10) express the 0-1 restrictions of the decision variables y_i and z_j. The objective function in (11) maximizes the total profit contribution expressed as the total net revenue.

4. Experimental Results

To verify our MILP formulation of the OIPP, we implemented and experimentally tested our model with real life market transaction data obtained from a Belgian retail store [16]. The dataset (retail.txt) stores five months of transaction data collected over four separate periods.

Retail data characteristics:

Total number of transactions	88,163
Item ID range	1- 16470
Number of items (N)	3,151
Total number of customers	5,133
Average basket size	13
Data collection period	5 months total (in four separate periods)

For further details of the data refer to [16]. Since not all characteristics of the data are publicly available (presumed to be confidential), we supplemented them with values for such fields as storage space required per item (s_i), revenue from selling item package X_j (p_j) and cost attributed to item i (c_i).

4.1 Data preparation stages

As discussed in section 3, before building the MILP model of the market data we need to know the data characteristics. Therefore, the data is preprocessed using the following steps to prepare it for input to the mathematical programming software:

1. Each transaction record is organized as an ascending sequence of item Ids;
2. A count of the number of items (n_j) in each transaction is inserted as the first field of the record;
3. The records in the database are then sorted in ascending order according to the count of items and then by the item Ids as minor keys;
4. Finally, identical transactions are counted to obtain their frequencies (f_j);
5. This final dataset is fed to a program (createLP) which builds the MILP model corresponding to the current problem.

This model is then submitted to a mathematical programming application to be solved as an MILP with binary integer variables (y_i's and z_j's).
[We used C++ programs (for steps 1 – 5 above) to process the input retail market basket dataset and produced the output in appropriate format. As our MILP formulation assumes data mining activities as a pre-step, discussions regarding the preprocessing done by these programs are unnecessary.]

4.2 Sample optimal package selection problem

To help explain our methodology, we use an example problem and work it through the different stages of finding the optimal profit from the given dataset. Consider the following dataset consisting of 5 sales transactions involving 7 items.

$X_1 = \{7\}$, $X_2 = \{1, 2\}$, $X_3 = \{5, 6\}$, $X_4 = \{12, 13\}$ and $X_5 = \{2, 6, 12, 13\}$

Table 1 below, shows the characteristics of various items (sp_i – selling price, $prof_i$ – profit per unit); while Table 2 presents the details of each item package.

Item	1	2	5	6	7	12	13
s_i	0.2	0.3	0.25	0.3	0.15	0.3	0.2
c_i	2.5	3.1	4.5	3.7	2.1	3.5	2.5
$s.p._i$	3.2	3.9	6.7	4.9	2.6	4.0	3.1
$prof_i$	0.7	0.8	2.2	1.2	0.5	0.5	0.6

Table 1. Characteristics of items in the sample dataset

Count of items (n_j)	Number of transactions (f_j)	Item package (X_j)	Package revenue ($p_j * f_j$)	Package storage ($\sum s_i * f_j$)	Package cost ($\sum c_i * f_j$)
1	200	7	520	30	420
2	231	1 2	1640.1	115.5	1293.6
2	34	5 6	394.4	18.7	278.8
2	341	12 13	2421	170.5	2046
6	11	2 6 12 13	174.9	12.1	140.8

Table 2. Processed sample dataset for creating the MILP model

The first column shows the number of items in the packages viz. n_j; the second shows the frequency (f_j); while the third shows the individual items that make up each item package. The last three columns show the computed aggregates for each package.

The createLP program, outlined in step 5 above, processes the formatted dataset (steps 1-4) and produces the corresponding MILP model (Fig. 1) to the sample dataset. This model is then solved using CPLEX, a commercial package for solving the all kind of linear programs. Fig. 2 presents the output from the package.

```
\Problem name: sample.lp

 Maximize

  100z1 + 346.5z2 + 115.6z3 + 375.1z4 + 34.1z5

 Subject To
 -1z1 +y7 >= 0
 -2z2 +y1 +y2 >= 0                                  Max.storage
 -2z3 +y5 +y6 >= 0                                  constraint
 -2z4 +y13 +y12 >= 0
 -4z5 +y2 +y6 +y12 +y13 >= 0

  30z1 + 115.5z2 + 18.7z3 + 170.5z4 + 12.1z5 <= 100

  520z1 + 1640.1z2 + 394.4z3 + 2421.1z4 + 174.9z5 >= 600  <-- Min. revenue
                                                              constraint

  y1 +y2 +y5 +y6 +y7 +y12 +y13 >= 5                 } lower & upper
  y1 +y2 +y5 +y6 +y7 +y12 +y13 <= 10                }  bounds

 Binaries
  z1 z2 z3 z4 z5
  y1 y2 y5 y6 y7 y12 y13

 End
```

Fig. 1. Sample problem `sample.lp`

```
Integer optimal solution:  Objective =   2.4970000000e+002
Solution time =    0.03 sec.  Iterations = 0  Nodes = 0

CPLEX> dis sol var -
Variable Name           Solution Value
z1                            1.000000
z3                            1.000000
z5                            1.000000
y7                            1.000000
y2                            1.000000
y5                            1.000000
y6                            1.000000
y13                           1.000000
y12                           1.000000

All other variables in the range 1-12 are zero.
```

Fig. 2. Solution of sample MILP using CPLEX

We notice (Fig. 2) that the optimal value i.e. the maximal profit, obtained under the given constraints of 100 units of storage space and satisfying the minimum revenue of \$600 is \$249.70. The three best item packages to stock are X_1, X_3 and X_5 which correspond to the binary decision variables z_1, z_3 and z_5 respectively. Further, the particular items in the optimal set to store are 7, 2, 5, 6, 12 and 13 (corresponding to the decision variables y_7 y_2 y_5 y_6 y_{12} y_{13}. The remaining item packages and items do not participate in the optimal solution.

We now present the results from the retail dataset as described at the beginning of the section. Given a certain maximum storage space, the retailer might like to find out the optimum profit (and item packages) against a maximum number of items to be put on the shelves. He might also be curious as to how the profit varies if he is able to acquire more storage space. To show how easily this can be achieved using our MILP formulation, we varied the values for S, the maximum storage space parameter, from 1000 to 4000 and varied the upper limit for the number of items to be shelved i.e. N_U from 20 to 500. The resulting MILP was then submitted to CPLEX 9.0 to calculate the value of the net profit function (z). Table 3 shows the effect of changing the maximum number of items (N_U) has on the objective.

4000									
Nu	20	50	100	150	200	300	400	450	500
profit	22,697	27,996	32,323	34,646	36,320	39,281	39,384	39,381	39,384
time	0.24	0.24	0.23	0.26	0.23	0.24	0.22	0.12	0.11

3000									
Nu	20	50	100	150	200	300	400	450	500
profit	22,697	27,996	32,323	34,646	36,320	39,281	39,281	39,328	39,328
time	0.25	0.22	0.23	0.26	0.22	0.24	0.22	0.35	0.34

2000									
Nu	20	50	100	150	200	300	400	450	500
profit	22,697	27,996	32,323	34,646	36,320	38,315	38,423	38,423	38,423
time	0.25	0.22	0.23	0.26	0.21	0.22	0.23	0.23	0.23

1000									
Nu	20	50	100	150	200	300	400	450	500
profit	22,697	27,996	32,323	34,556	35,564	35,636	35,636	35,636	35,636
time	0.25	0.22	0.23	0.38	0.41	0.23	0.23	0.23	0.23

Table. 3. Comparative values of the profit function as the number of items to be stocked and storage available are varied.

We then chart (Figure 3) the observations to visualize the effects of max. storage and N_u on the value of the objective, z. We observe that while increasing the number of items does increase the net profit quite substantially, after a certain stage the rate or amount of change in the same is not significant, eventually peaking and remaining so in spite of increasing resources (storage space or number of items stored). This observation could be of value to the retailer as he can clearly visualize the expected changes in profit by changing certain parameters as need be. Similarly, one can study the effect of varying the limits of other resources and study their effects on the profitability function.

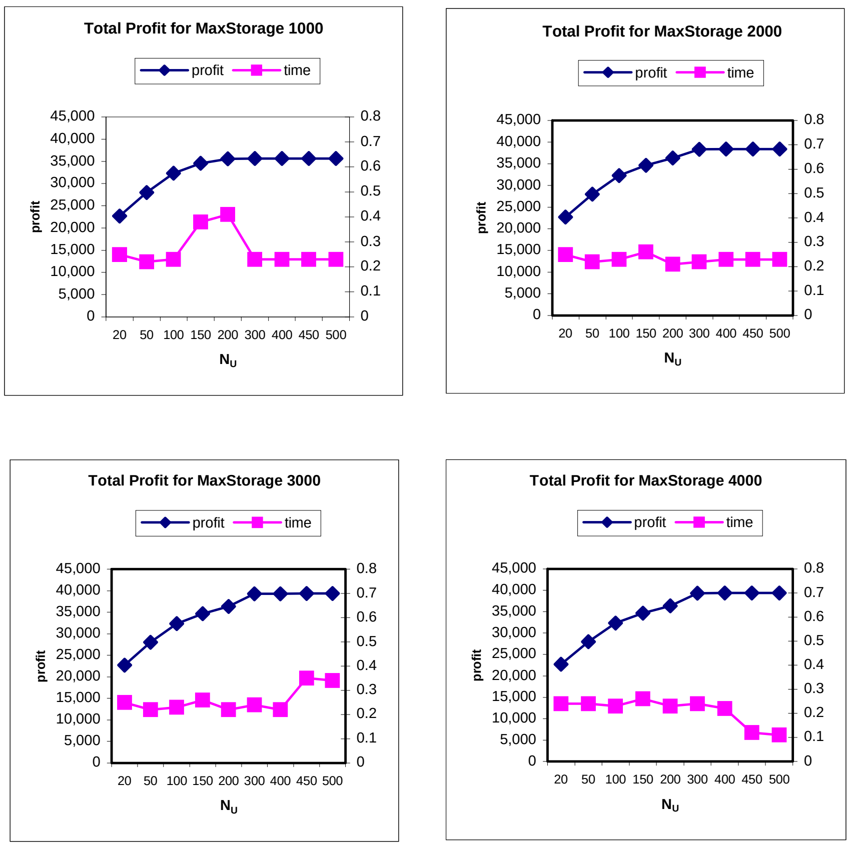

Fig. 3 Effect of varying max storage and max number of items on the objective

For our experiments, we used an AMD Athlon XP2100 PC with a CPU clock of 2.1 GHz having 512 MB of RAM running Windows 2000. Our experiments show very encouraging results as all of them are achieved in a sub-second response time. This proves that our method of solving such problems is very much viable.

5. Conclusions

In this paper, we have introduced a general class of problems called the value based optimal item package problem that can support real world business decisions using data mining. The solutions to these problems require the combination of mathematical modeling with data mining and knowledge discovery from large transaction data. We formulated a generic problem using the mixed integer linear programming model and implemented it using real life transactional data from a retail store. Our specification provides scope for using a large number of methodologies available in the literature to solve the value based frequent item set problems.

It is well known that the general integer linear programming problem is NP hard. In addition, in many practical applications of the frequent item set problem, the parameters like N, the number of items and T, the number of transactions in the data base may be very large. When N and T are not very large, we can use some of the standard commercial software products such as CPLEX to solve the model proposed in this paper. Furthermore, future research can be focused on developing specially designed branch and cut algorithms [13] [14] [15], branch and price algorithms and/or efficient heuristics and probabilistic methods to solve our MILP formulations of these models. When N and T are large, the future research can explore the possibility of solving these models restricted to some random samples drawn from the database and developing methods of estimating the required information.

References

1. Kleinberg, J., Papadimitriou, C., Raghavan, P.: A Microeconomic View of Data Mining. Data Mining and Knowledge Discovery. **2** (1998) 311-324
2. Brijs, T., Swinnen, G., Vanhoof, K, Wets, G.: Building an Association Rules Framework to Improve Product Assortment Decisions. Data Mining and Knowledge Discovery. **8** (2004) 7-23
3. Gopalan, R.P., Sucahyo, Y.G.: High Performance Frequent Patterns Extraction using Compressed FP-Tree. Proceedings of SIAM International Workshop on High Performance and Distributed Mining (HPDM04), Orlando, USA (2004)
4. Han, J., Pei, J., Yin, Y.: Mining Frequent Patterns without Candidate Generation. Proceedings of ACM SIGMOD, Dallas, TX (2000)
5. Liu, J., Pan, Y., Wang, K., Han, J.: Mining Frequent Item Sets by Opportunistic Projection. Proceedings of ACM SIGKDD, Edmonton, Alberta, Canada (2002)
6. Ali, K., Manganaris, S., Srikant, R.: Partial Classification using Association Rules. Proceedings of KDD-97, Newport Beach, California (1997)
7. Viveros, M.S., Nearhos, J.P., Rothman, M.J.: Applying Data Mining Techniques to a Health Insurance Information System. Proceedings of VLDB-96, Bombay, India, 1996.
8. Demiriz, A., Bennett, K.P.: Optimization Approaches to Semi-Supervised Learning. In Complementarity: Applications, Algorithms and Extensions. Kluwer Academic Publishers, Boston (2001) 121-141
9. Bradley, P., Gehrke, J., Ramakrishnan, R., Srikant, R.: Scaling Mining Algorithms to Large Databases. Communications of the ACM. **45** (2002) 38-43
10. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R.: Advances in Knowledge Discovery and Data Mining. MIT Press, Cambridge, MA (1996)

11. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, San Francisco (2001)
12. Hand, D., Mannila, H., Smyth, P.: Principles of Data Mining. MIT Press, Cambridge, MA (2001)
13. Achuthan, N.R., Caccetta, L., Hill, S.P.: A New Subtour Elimination Constraint for the Vehicle Routing Problem. E.J.O.R. **91** (1996) 573-586
14. Achuthan, N.R., Caccetta, L., Hill, S.P.: Capacitated Vehicle Routing Problem: Some New Cutting Planes. Asia-Pacific Journal of Operational Research. **15** (1998) 109-123
15. Achuthan, N.R., Caccetta, L., Hill, S.P.: An Improved Branch and Cut Algorithm for the Capacitated Vehicle Routing Problem. Transportation Science. **37** (2003) 153-169.
16 Brijs T., Swinnen G., Vanhoof K., and Wets G. The Use of Association Rules for Product Assortment Decisions: A Case Study, in: Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining, San Diego (USA), August 15-18, (1999) 254-260.
17. Webb, G. Discovering Associations with Numeric Variables. Proceedings of the Knowledge Discovery in Databases (KDD 01), San Francisco (USA), (2001) 383-388.
18. Aumann, Y., Lindell, Y. A Statistical Theory for Quantitative Association Rules. Proceedings of the Knowledge Discovery in Databases (KDD 99), San Francisco (USA), (1999) 262-270.
19. Marakas, G. M. Modern Data Warehousing, Mining and Visualization. Prentice Hall, Upper Saddle River, New Jersey (USA). (2003).