CLASSIFICATION AND TARGET GROUP SELECTION BASED UPON FREQUENT PATTERNS

WIM PIJLS AND ROB POTHARST

ERIM REPORT SERIES *RESEARCH IN MANAGEMENT*	
ERIM Report Series reference number	ERS-2000-40-LIS
Publication status / version	October 2000
Number of pages	11
Email address first author	Pijls@few.eur.nl
Address	Erasmus Research Institute of Management (ERIM) Rotterdam School of Management / Faculteit Bedrijfskunde Erasmus Universiteit Rotterdam PoBox 1738 3000 DR Rotterdam, The Netherlands Phone: # 31-(0) 10-408 1182 Fax: # 31-(0) 10-408 9640 Email: info@erim.eur.nl Internet: www.erim.eur.nl

Bibliographic data and classifications of all the ERIM reports are also available on the ERIM website:
www.erim.eur.nl

ERASMUS **R**ESEARCH **I**NSTITUTE OF **M**ANAGEMENT

REPORT SERIES
RESEARCH IN MANAGEMENT

BIBLIOGRAPHIC DATA AND CLASSIFICATIONS

Abstract	In this technical report, two new algorithms based upon frequent patterns are proposed. One algorithm is a classification method. The other one is an algorithm for target group selection. In both algorithms, first of all, the collection of frequent patterns in the training set is constructed. Choosing an appropriate data structure allows us to keep the full collection of frequent patterns in memory. The classification method utilizes directly this collection. Target group selection is a known problem in direct marketing. Our selection algorithm is based upon the collection of frequent patterns.	
Library of Congress Classification (LCC)	5001-6182	Business
	5201-5982	Business Science
	QA1-43	Mathematics: General
Journal of Economic Literature (JEL)	M	Business Administration and Business Economics
	M 11 R 4	Production Management Transportation Systems
	C 6	Mathematical Methods and Programming
European Business Schools Library Group (EBSLG)	85 A	Business General
	260 K 240 B	Logistics Information Systems Management
	250 A	Mathematics
Gemeenschappelijke Onderwerpsontsluiting (GOO)		
Classification GOO	85.00	Bedrijfskunde, Organisatiekunde: algemeen
	85.34 85.20	Logistiek management Bestuurlijke informatie, informatieverzorging
	31.80	Toepassingen van de wiskunde
Keywords GOO	Bedrijfskunde / Bedrijfseconomie	
	Bedrijfsprocessen, logistiek, management informatiesystemen	
	Algoritmen, Classificatietheorie, Doelgroepen	
Free keywords	Classification, Target Group Selection, Association rules, Frequent Item Sets	
Other information	The first part of this technical report has been published as W. Pijls and R. Potharst, Classification based upon Frequent Patterns, in: D. Lukose and G.J. Williams (eds.), Proceedings of the Symposium on the Application of Artificial Intelligence in Industry, pp. 87-94. Deakin University, Geelong, 2000.	

Classification and Target Group Selection based upon Frequent Patterns

Wim Pijls Rob Potharst

Erasmus University,
P.O.Box 1738, 3000 DR Rotterdam

Abstract

In this technical report[1], two new algorithms based upon frequent patterns are proposed. One algorithm is a classification method. The other one is an algorithm for target group selection. In both algorithms, first of all, the collection of frequent patterns in the training set is constructed. Choosing an appropriate data structure allows us to keep the full collection of frequent patterns in memory. The classification method utilizes directly this collection. Target group selection is a known problem in direct marketing. Our selection algorithm is based upon the collection of frequent patterns.

1 Introduction

Machine learning and datamining are major issues in the present-day AI research. An almost classical issue in this field is classification. Several classification methods have been developed over the past two decades. We mention, amongst others, neural networks, decision trees, genetic algorithms etc. Some techniques are based upon frequent patterns, viz. rough sets, logical analysis of data, association rules. In this technical report we propose a new classification algorithm based upon frequent patterns. A second algorithm proposed in this technical report is also based upon frequent patterns and deals with target group selection in direct marketing.

[1] The first part of this technical report has been published as [8].

The first step in the algorithms is the construction of the collection of frequent patterns. Without any refining operation, those frequent patterns are used for classification and selection respectively. Thus, we might say that the new algorithms are brute-force datamining methods using a raw collection of patterns. Such an approach appears to be feasible, which is not surprising given today's high values of speed and memory size. We will compare our classifier with other existing classifiers.
The outline of this technical report is as follows. Section 2 recalls some definitions and facts about frequent patterns. The first new algorithm, a classification method is presented in Section 3. We also design a variant algorithm which actually performs a relaxation of the classification procedure. The algorithms are illustrated by experimental results. In Section 4 we show that there exist clear relationships between our method on the one hand and three other methods on the other, viz. classifying based on associations, rough sets and logical analysis of data. Section 5 shows a new method for target group selection. Again, experimental results are included.

2 Frequent patterns

In this section, we discuss some definitions and facts on frequent patterns. A data set is a set of *records* or *cases*. Each record consists of an n-tuple (n is fixed) of discrete values. The positions in a record correspond to attributes. Suppose that we have $n = 4$ and the four attributes are: income, married, children, creditworthy. An example of a record in a dataset of customers is *(5, yes, 3, yes)*, which means that the customer related to this record has income group equal to 5, is married, has 3 children and is creditworthy. As mentioned above, we require discrete values. If an attribute has continuous values, discretization is required. One attribute is appointed to be the target attribute. The values which are taken by the target attribute are called *classes* or *class labels*.
Suppose that a data set D has n attributes $a_1, a_2, \ldots, a_n$, where a_n is the target attribute. A pattern (also called an item set in some literature) is defined as a series of m ($m \leq n-1$) equalities of the form $(a_{i_1} = v_{i_1}, a_{i_2} = v_{i_2}, \ldots, a_{i_m} = v_{i_m})$, with a_{i_k} a non-target attribute and v_{i_k} a value that can be taken by attribute a_{i_k}, $1 \leq k \leq m$. A record R is a *supporter* of a pattern $P = (a_{i_1} = v_{i_1}, a_{i_2} = v_{i_2}, \ldots, a_{i_m} = v_{i_m})$, if R matches pattern P, i.e., attribute a_{i_k} occurs in R and has value v_{i_k} for each k, $1 \leq k \leq m$. The *support*

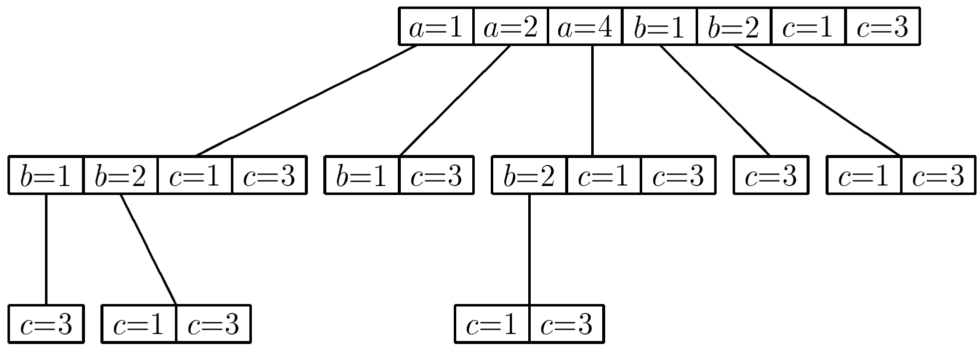

Figure 1: An example of a trie (without support counts).

of a pattern P denoted by $supp(P)$ is equal to the number of its supporters. Given a threshold or minimal support (denoted by $minsup$) a pattern is called frequent if $supp(P) \geq minsup$. A convenient data structure viz. a trie can be used to capture the full collection of frequent patterns. A trie is able to store all frequent items in a compact way. An example of a trie is shown in Figure 1. This trie is related to a data set which has three attributes a, b and c, and target attribute d. The value range for a is the integer set $1, 2, 3, 4$; the value range for b is $1, 2$ and that for c is $1, 2, 3$. Each path (not necessarily ending at a leaf) in the trie represents a frequent pattern and vice versa. For instance, $(a = 1, b = 2, c = 1)$ denotes a path as well as as a frequent pattern. Any path corresponding to a subset of $(a = 1, b = 2, c = 1)$ is also a frequent pattern. Hence, the paths $(a = 1, b = 2)$, $(a = 1, c = 1)$ and $(b = 2, c = 1)$ also correspond to frequent patterns and hence, are included in the trie. The patterns $(a = 3)$ and $(c = 2)$ are assumed to be infrequent. Consequently, each pattern containing any of these two patterns is also infrequent.

As mentioned before, $supp(P)$ is equal to the number of the supporters of P. For a given class c and a pattern P, the class support of c in P, denoted by $supp_c(P)$, is equal to the number of supporters for P with class label c. The confidence of a class c, also called the relative support, in a pattern P is defined as $conf_c(P) = supp_c(P)/supp(P)$. The class of a pattern P, denoted by $class(P)$, is defined as a class c for which $supp_c(P)$ and hence also $conf_c(P)$ is maximal. For $c_0 = class(P)$, the value $conf_{c_0}(P)$ is also denoted by $conf(P)$. The following example illustrates the definitions. Consider the trie in Figure 1. Suppose that target attribute d has class labels 0 and 1. Suppose that pattern $(a = 1, b = 2)$ is supported by 30 records, of which there are 20 in class 1 and 10 in class 0. Then we have $supp(P) = 30$, $supp_1(P) = 20$, $supp_0(P) = 10$, $conf_1(P) = 2/3$, $conf_0(P) = 1/3$. Since class label 1 has the greatest support, we have $class(P) = 1$ and $conf(P) = 2/3$.

3 A classifier based upon frequent patterns

PatMat We now present our new algorithm, called PatMat. To build a classifier we need a data set D which acts as training set. The collection of frequent patterns in D is constructed and is stored into a trie T. Any algorithm designed to find frequent patterns is appropriate. Apriori is the best-known algorithm in this area. We assume that $supp(P)$ and also $supp_c(P)$ for every c is stored into the cell at the end of path P. To calculate these quantities, only a slight extension of Apriori is needed. The classification of a record R proceeds as follows. Each path or equivalently each frequent pattern P is examined for being a supporter of R. The pattern P for which $conf(P)$ is maximal is selected and its class value $class(P)$ determines the class for R. The following code reflects the above classification process.

(1) **function** $classifying(R)$
(2) $bestconf = \max\{conf(P) \mid P$ a frequent pattern supported by $R\}$;
(3) P_0 is a frequent pattern such that $bestconf = conf(P_0)$;
(4) c_0 is a class such that $conf(P_0) = conf_{c_0}(P_0) = bestconf$;
(5) **return** c_0, P_0;

In order to determine the value of $bestconf$ (see line (2)), the trie needs to be traversed. Hence, for any record P that is being classified, a trie traversal is performed. Clearly, some parts of the trie may be discarded. If R does not support a pattern corresponding to a cell P, then R does not support any extension of P either. In case cell P is visited, the subtrie below P (if any) may be discarded.
Theoretically, it might happen that there is no frequent pattern supported by R (see line (2)). However, it can be shown that the occurrence of such an event is very improbable. In fact, in all our experiments this did not occur a single time. If we want the algorithm to cover all possible cases, we could assign a default class to such a record R, for instance the most frequent class. From the fact that the confidences are relative frequencies approximating pattern probabilities, it can be seen that the algorithm is an approximation

to the Bayes classifier under the assumption that the training set has proportions that reflect the class probabilities. By restricting attention to *frequent* patterns we try to enhance the quality of the approximation.

We conducted several experiments with PatMat. We will give a more detailed description of those experiments later. We have noticed many collisions in line (3), mostly with $bestconf = 100\%$. A collision means that patterns P and P' are found with $bestconf = conf(P) = conf(P')$ or equivalently, pattern-class combinations (P, c) and (P', c') are found with $bestconf = conf_c(P) = conf_{c'}(P')$. We formulated the following criteria for solving collisions.

- if $conf_c(P_i) = conf_{c'}(P_j)$ and $supp_c(P_i) > supp_{c'}(P_j)$ for any classes c and c', then P_i takes precedence over P_j;
- if the previous rule still leaves a collision, then a longer pattern (i.e., a higher number of attributes) precedes a shorter one.

Applying these criteria the number of collisions nearly vanishes. Once a pattern P_0 is chosen, usually a unique value c_0 can be determined in line (4). A major advantage of the new method over other methods is its transparency. Note, that the classification function not only returns c_0, the appointed class, but also P_0, the pattern that caused c_0 to be picked as the appointed class. Thus, it is possible to give a clear account for the target label assigned to a test record. A record is put into a class, since it meets a certain pattern and many training records meeting that pattern belong to the same class. Thus, an instance could be classified as class A, because it has value 3 for attribute 2, and value 1 for attribute 4, and 97% of such instances in the training set belong to class A.

A variant algorithm We have mentioned that line (3) in the code of the procedure *classifying* gives rise to a lot of collisions. In addition, we noticed that many so-called *near-collisions* occurred. A *near-collision* means that a combination (P, c) can be found with $bestconf \approx conf_c(P)$ or more precisely $conf_c(P) > 0.9 * bestconf$. Because of the great number of collisions, we decided to apply multiple classification: a record may be classified into multiple classes, or put another way, a record may take multiple class labels. In order to take the collisions into account, the following code was used.

(1) **function** *multiple-classifying*(R)
(2) *bestconf*$=\max\{$*conf*$(P) \mid P$ a frequent pattern supported by $R\}$;
(3) $\mathcal{C} = \{c \mid$ $conf_c(P) > 0.9 * bestconf$ for any frequent P supported by $R\}$;
(4) **return** $\mathcal{C}$;

A record R is considered to be classified wrongly, if the class label of R does not occur in $\mathcal{C}$, the set resulting from the call *multiple-classifying*(R). As will be shown in the next paragraph, the variant algorithm has a considerably lower error rate.

Experiments In this section we will describe some experiments. The new algorithms, PatMat and its variant, were implemented under the Borland C++ 5.02 environment. These algorithms were compared with C4.5, a famous decision tree classifier. For our experiments we actually used the Java implementation of C4.5, Release 8 from [9]. We used eight data sets from the UCI ML Repository[6]. If a data set contained continuous attributes, these were discretized using the method in [4]. The three algorithms under consideration were applied to each of the eight data sets. In order to compare the performance of the algorithms, we measured the error rate of each classification run. The error rate is defined as the number of wrongly classified records divided by the total number of classified records. The difference between the error rate of PatMat and that of the variant is a measure for the number of collisions during PatMat. A test run for a combination of a data set and an algorithm consists of the following steps. A random selection is made containing 2/3 of the records in the data set. This set acts as the training set. The remaining 1/3 of the records is used as the test set. The error rate was measured in each run. This action was carried out ten times. The average error rate and the standard deviation were computed over these ten runs. The outcomes are displayed in Figure 2. For finding frequent patterns in PatMat and its variant, *minsup*$= 1\%$ is chosen. Figure 2 shows that PatMat performs slightly better than C4.5. Remarkably, we noticed that over 90% over the records with *bestconf*$= 100\%$ and almost all records were classified with *bestconf*$\geq 90\%$. None of the test runs took longer than a few seconds, although some of the data sets consist of thousands of cases.

	C4.5		PatMat		variant	
	aver	stddev	aver	stddev	aver	stddev
Austra	15.3	3.14	14.5	2.05	3.1	0.81
Breast-w	5.3	2.03	4.6	1.66	1.5	0.63
Diabetes	21.4	3.41	21.4	1.73	17.7	2.23
German	28.2	2.03	26.9	1.63	17.0	2.30
Glass	30.8	6.33	28.1	5.30	20.9	4.00
Led7	27.6	1.26	28.0	1.02	21.8	1.58
Lymph	25.6	4.77	18.9	5.55	2.8	2.08
Heart	24.4	2.85	18.6	2,52	1.5	1.34
Pima	22.3	1.43	21.0	2.21	16.5	2.00

Figure 2: Classification results

4 Comparison with other methods

Several algorithms in the machine learning work with a list of rules of the form $P \Rightarrow c$ where P is a pattern and c a class. In contrast with those algorithms, we do not construct a list, but we utilize the complete trie of frequent patterns. Closest to our algorithm is the CBA method in [5]. An intricate and cumbersome method is proposed to extract rules from the collection of frequent item sets. For many data sets, this results into a list of ten thousands of rules.

Similar classification rules based on discrete-valued datasets can also be generated using the Rough Sets (RS) method introduced by Z. Pawlak[7]. It can be easily seen that all rules generated by the Rough Set method have 100% confidence, in our terminology. It can be shown that PatMat classifies more cases with 100% confidence than RS can classify at all. Moreover, PatMat also has rules available that do not match a 100% confidence rule; in those cases classical RS cannot classify or has to resort to a default rule. Thus, our method seems more appropriate than RS to discover non-deterministic, noisy associations rather than clear-cut functional dependencies. Only recently, some generalized models of RS have been introduced, in which exact dependencies are replaced by approximate dependencies, repairing this deficiency in a RS manner, see e.g. [10].

The Logical Analysis of Data (LAD) method developed by a group around P. Hammer[2] is very similar to the Rough Set method, but has the further drawback that it works only for dichotomous attributes. The patterns found

by LAD form a subset of the frequent patterns that constitute our trie.
Like k-nearest-neighbor and IGTree[3], the proposed algorithm is a lazy learning method. However, unlike those methods, our trie is not a lossless compression of the training set. Generalization takes place by restricting the attention to *frequent* patterns.

5 Target group selection

In addition to classification problems, a variant of our algorithm may also be used for target group selection problems, such as arise in direct mailing. Suppose, in a direct mailing campaign, we want to cover in our target group as many individuals as possible with a certain hidden characteristic, e.g. buyers of a certain commodity or service we are selling. In a training set at our disposal, the hidden characteristic of the individuals is known. In addition, for each individual the values on a number of attributes are available. How should we, on the basis of the known attribute values, assemble a fixed size target group such that it will contain as many future buyers as possible? Such a target group selection problem can be modeled as a two class problem, where class = 1 if an individual has the hidden characteristic and class = 0 otherwise. We should note first of all, that this is not a classification problem, since in all subsets of individuals the hit density (i.e. the relative frequency of class 1 individuals) can be very low. Thus, every individual would be classified as class 0. In the following algorithm for the target group selection problem, which we named PatSelect, we circumvent this difficulty by choosing subsets with maximal hit density.

PatSelect Like in PatMat, we use the training set D to form a collection of frequent patterns stored in a trie T. Separately, we have a data set S from which to select a number N of records that should contain as many target class c objects as possible. The selection process we propose proceeds as follows. For each record R in the data set S, we consider all patterns P in T that are supported by R. Among those patterns, we choose the one with the highest value of $conf_c(P)$. This value is called $recordconf(R)$. Finally, we select the group of N records R with the highest values of $recordconf(R)$. The following code reflects this process.

procedure *selection*(*class c, size N*)
for every record R in the data set S **do**
    recordconf(R) = $\max\{conf_c(P) | R$ supports $P\}$;
sort all records R according to decreasing *recordconf*(R);
select topmost N records;

To test our selection method PatSelect we performed several experiments both on direct marketing data and on the data sets we tested PatMat on. The direct mailing data set was taken from the CoIL website[2] which recently threw the Challenge2000 contest for data miners. The test set contained 4000 records, of which 238 were caravan owners. The problem consisted in selecting from the test set a group of 800 records containing as many caravan owners as possible. While the best contestants scored between 110 and 121 caravan owners, PatSelect scored 112. Again, a competitive result.
The next table shows the results of applying PatSelect to the data sets of Section 3. In each case, we took the least frequent class in the data set as our target class. Subsequently, the data set was randomly split to form a training set (2/3) and a test set (1/3). The size of the target group was taken to be the expected number of target class members in the test set. Ideally, a selection procedure could then select all target class members. In the table is recorded the percentage of target class members that were selected by PatSelect in the group, and in a separate column, the percentage of expected target class members in the selected group, if the group would have been chosen randomly. For each data set this was done 10 times, each time splitting the data randomly. The averages and standard deviations over these 10 repeats are recorded in Figure 3. As can be seen from the table, in all cases a significant improvement can be attained over random selection, while in some cases (like the Breast-w data) the improvement is spectacular.

6 Concluding remarks

We have presented two new methods based on frequent patterns, one for classification and one for target group selection. Both methods appear to perform adequately. The main advantage of both methods above the existing methods is their transparency: it is very easy to state the reason why an

[2]http://www.dcs.napier.ac.uk/coil/

	random		PatSelect	
	aver	stddev	aver	stddev
Austra	43.5	2.4	67.6	4.9
Breast-w	33.9	2.4	91.1	3.7
German	29.1	1.6	48.7	4.2
Glass	6.3	1.8	50.9	21.4
Led7	8,7	0.6	65,0	2.5
Lymph	36.9	3.4	41.4	10.6
Heart	44.6	4.8	57.7	5.5
Pima-Indian	34.0	2.1	64.8	3.3

Figure 3: Target Group Selection results.

object is classified as it is, or why an individual is selected in the mailing group. It is argued that our classification method generalizes the Rough Sets and LAD methods, making it more appropriate for noisy data and large data sets. In a forthcoming paper we intend to compare our target selection method with existing methods like CHAID and regression methods.

References

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A.I. Verkamo, *Fast Discovery of Association Rules*, Chapter 12 in: U.M Fayyad et al. (eds.), Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, pp. 307-328, 1996.

[2] E. Boros, T. Ibaraki, E. Mayoraz, P.Hammer, A. Kogan and I. Muchnik, *An Implementation of Logical Analysis of Data*, Technical Report, RRR22-96, RUTCOR-Rutgers University, forthcoming in: IEEE Transactions on Knowledge and Data Engineering.

[3] W. Daelemans, A. van den Bosch and A. Weijters, *IGTree: using trees for compression and classification in lazy learning algorithms,* in: Artificial Intelligence Review 11, pp. 407-423, 1997.

[4] U. M. Fayyad and K.B. Irani, *Multi-interval discretization of continuous-valued attributes for classification learning,* IJCAI-93, pp. 1022-1027.

[5] Bin Liu, Wynn Hsu and Yiming Ma, *Integrating Classification and Association Rule Mining,* in: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), New York, 1998.

[6] C.J. Merz and P. Murphy, *UCI machine learning repository*, http://www.cs.uci.edu/~mlearn/MLRepository.html

[7] Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht etc., 1991.

[8] W. Pijls and R. Potharst, *Classification based upon Frequent Patterns*, in: D. Lukose and G.J. Williams (eds), Proceedings of the Symposium on the Application of Artificial Intelligence in Industry, pp.87–94. Deakin University, Geelong, 2000.

[9] Ian H. Witten and Eibe Frank, *Data Mining, Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann Publishers, 2000.

[10] W. Ziarko, *Variable Precision Rough Set Model*, J. of Computer and System Sciences, Vol. 46, pp. 39–59, 1993.

ERASMUS **R**ESEARCH **I**NSTITUTE OF **M**ANAGEMENT

REPORT SERIES
RESEARCH IN MANAGEMENT

Publications in the Report Series Research* in Management

Impact of the Employee Communication and Perceived External Prestige on Organizational Identification
Ale Smidts, Cees B.M. van Riel & Ad Th.H. Pruyn
ERS-2000-01-MKT

Critical Complexities, from marginal paradigms to learning networks
Slawomir Magala
ERS-2000-02-ORG

Forecasting Market Shares from Models for Sales
Dennis Fok & Philip Hans Franses
ERS-2000-03-MKT

A Greedy Heuristic for a Three-Level Multi-Period Single-Sourcing Problem
H. Edwin Romeijn & Dolores Romero Morales
ERS-2000-04-LIS

Integer Constraints for Train Series Connections
Rob A. Zuidwijk & Leo G. Kroon
ERS-2000-05-LIS

Competitive Exception Learning Using Fuzzy Frequency Distribution
W-M. van den Bergh & J. van den Berg
ERS-2000-06-LIS

Start-Up Capital: Differences Between Male and Female Entrepreneurs, 'Does Gender Matter?'
Ingrid Verheul & Roy Thurik
ERS-2000-07-STR

The Effect of Relational Constructs on Relationship Performance: Does Duration Matter?
Peter C. Verhoef, Philip Hans Franses & Janny C. Hoekstra
ERS-2000-08-MKT

Marketing Cooperatives and Financial Structure: a Transaction Costs Economics Analysis
George W.J. Hendrikse & Cees P. Veerman
ERS-2000-09-ORG

---

* ERIM Research Programs:
LIS Business Processes, Logistics and Information Systems
ORG Organizing for Performance
MKT Decision Making in Marketing Management
F&A Financial Decision Making and Accounting
STR Strategic Renewal and the Dynamics of Firms, Networks and Industries

A Marketing Co-operative as a System of Attributes: A case study of VTN/The Greenery International BV,
Jos Bijman, George Hendrikse & Cees Veerman
ERS-2000-10-ORG

Evaluating Style Analysis
Frans A. De Roon, Theo E. Nijman & Jenke R. Ter Horst
ERS-2000-11-F&A

From Skews to a Skewed-t: Modelling option-implied returns by a skewed Student-t
Cyriel de Jong & Ronald Huisman
ERS-2000-12-F&A

Marketing Co-operatives: An Incomplete Contracting Perspective
George W.J. Hendrikse & Cees P. Veerman
ERS-2000-13– ORG

Models and Algorithms for Integration of Vehicle and Crew Scheduling
Richard Freling, Dennis Huisman & Albert P.M. Wagelmans
ERS-2000-14-LIS

Ownership Structure in Agrifood Chains: The Marketing Cooperative
George W.J. Hendrikse & W.J.J. (Jos) Bijman
ERS-2000-15-ORG

Managing Knowledge in a Distributed Decision Making Context: The Way Forward for Decision Support Systems
Sajda Qureshi & Vlatka Hlupic
ERS-2000-16-LIS

Organizational Change and Vested Interests
George W.J. Hendrikse
ERS-2000-17-ORG

Strategies, Uncertainty and Performance of Small Business Startups
Marco van Gelderen, Michael Frese & Roy Thurik
ERS-2000-18-STR

Creation of Managerial Capabilities through Managerial Knowledge Integration: a Competence-Based Perspective
Frans A.J. van den Bosch & Raymond van Wijk
ERS-2000-19-STR

Adaptiveness in Virtual Teams: Organisational Challenges and Research Direction
Sajda Qureshi & Doug Vogel
ERS-2000-20-LIS

Currency Hedging for International Stock Portfolios: A General Approach
Frans A. de Roon, Theo E. Nijman & Bas J.M. Werker
ERS-2000-21-F&A

Transition Processes towards Internal Networks: Differential Paces of Change and Effects on Knowledge Flows at Rabobank Group
Raymond A. van Wijk & Frans A.J. van den Bosch
ERS-2000-22-STR

Assessment of Sustainable Development: a Novel Approach using Fuzzy Set Theory
A.M.G. Cornelissen, J. van den Berg, W.J. Koops, M. Grossman & H.M.J. Udo
ERS-2000-23-LIS

Creating the N-Form Corporation as a Managerial Competence
Raymond vanWijk & Frans A.J. van den Bosch
ERS-2000-24-STR

Competition and Market Dynamics on the Russian Deposits Market
Piet-Hein Admiraal & Martin A. Carree
ERS-2000-25-STR

Interest and Hazard Rates of Russian Saving Banks
Martin A. Carree
ERS-2000-26-STR

The Evolution of the Russian Saving Bank Sector during the Transition Era
Martin A. Carree
ERS-2000-27-STR

Is Polder-Type Governance Good for You? Laissez-Faire Intervention, Wage Restraint, And Dutch Steel
Hans Schenk
ERS-2000-28-ORG

Foundations of a Theory of Social Forms
László Pólos, Michael T. Hannan & Glenn R. Carroll
ERS-2000-29-ORG

Reasoning with partial Knowledge
László Pólos & Michael T. Hannan
ERS-2000-30-ORG

Applying an Integrated Approach to Vehicle and Crew Scheduling in Practice
Richard Freling, Dennis Huisman & Albert P.M. Wagelmans
ERS-2000-31-LIS

Informants in Organizational Marketing Research: How Many, Who, and How to Aggregate Response?
Gerrit H. van Bruggen, Gary L. Lilien & Manish Kacker
ERS-2000-32-MKT

The Powerful Triangle of Marketing Data, Managerial Judgment, and Marketing Management Support Systems
Gerrit H. van Bruggen, Ale Smidts & Berend Wierenga
ERS-2000-33-MKT

The Strawberry Growth Underneath the Nettle: The Emergence of Entrepreneurs in China
Barbara Krug & Lászlo Pólós
ERS-2000-34-ORG

Consumer Perception and Evaluation of Waiting Time: A Field Experiment
Gerrit Antonides, Peter C. Verhoef & Marcel van Aalst
ERS-2000-35-MKT

Trading Virtual Legacies
Slawomir Magala
ERS-2000-36-ORG

Broker Positions in Task-Specific Knowledge Networks: Effects on Perceived Performance and Role Stressors in an Account Management System
David Dekker, Frans Stokman & Philip Hans Franses
ERS-2000-37-MKT

An NPV and AC analysis of a stochastic inventory system with joint manufacturing and remanufacturing
Erwin van der Laan
ERS-2000-38-LIS

Generalizing Refinement Operators to Learn Prenex Conjunctive Normal Forms
Shan-Hwei Nienhuys-Cheng, Wim Van Laer, Jan N Ramon & Luc De Raedt
ERS-2000-39-LIS