Automated Dimensionality Reduction of Data Warehouses

Mark Last
Department of Computer Science and Engineering,
University of South Florida
4202 E. Fowler Avenue, ENB 118
Tampa, FL 33620 USA
mlast@csee.usf.edu

Oded Maimon
Department of Industrial Engineering
Tel-Aviv University, Tel Aviv 69978
Israel
maimon@eng.tau.ac.il

Abstract

A data warehouse is designed to consolidate and maintain all attributes that are relevant for the analysis processes. Due to the rapid increase in the size of the modern operational systems, it becomes neither practical, nor necessary to load and maintain in the data warehouse every operational attribute. This paper presents a novel methodology for automated selection of the most relevant independent attributes in a data warehouse. The method is based on the information-theoretic approach to knowledge discovery in databases. Attributes are selected by a stepwise forward procedure aimed at minimizing the uncertainty in the values of key performance indicators (KPI's). Each selected attribute is assigned a score, expressing its degree of relevance. Using the method does not require any prior expertise in the domain of the data and it can be equally applied to nominal and ordinal attributes. An attribute will be included in a data warehouse schema, if it is found as relevant to at least one KPI. We demonstrate the applicability of the method by reducing the dimensionality of a direct marketing database.

1 Introduction

A data warehouse is defined by Inmon (1994) as a "subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision making." Each organization has its own *key performance indicators* (KPI), which are used by management to monitor the organization's performance. Thus, a manufacturing company may measure its performance by throughput and cost, a KPI of a service company is the mean time to handle a service call, etc. When designing a data warehouse, we assume that these KPI's depend on some non-key attributes (called *classifying attributes*) in data warehouse relations (Schouten, 1999). The form and the strength of these dependencies are often the subject of data analysis on a data warehouse.

Since the operational systems and the data warehouses are built for different purposes (see Inmon, 1994), some attributes that are essential to the operational system, may be completely irrelevant to the performance measures of a company and thus excluded from its data warehouse. This fact is emphasized by several authors (like Gupta, 1997), but usually their assumption is that the data warehouse designers are knowledgeable enough to choose the right set of attributes. Though this assumption may be correct to certain extent in most data warehousing projects, the process of reducing the warehouse dimensionality, can be supported by an automated feature selection procedure that can quickly examine numerous potential dependencies, leading to automatic elimination of all irrelevant and redundant attributes.

There is another advantage in reducing the warehouse dimensionality. According to (Elder and Pregibon, 1996), large number of attributes constitutes a seriously obstacle to efficiency of most data mining algorithms, which may be applied to the detail data in a data warehouse. Such popular methods as k-nearest neighbors, decision trees, and neural networks do not scale well in the presence of numerous features. Moreover, some algorithms may be confused by irrelevant or noisy attributes and construct poor prediction models. A successful choice of features provided to a data mining tool can increase its accuracy, save the computation time, and simplify its results.

John *et al.* (1994) distinguishes between two models of selecting a "good" set of features under some objective function. The *feature filter* model assumes filtering the features *before* applying a data mining algorithm, while the *wrapper* model uses the data mining algorithm itself to evaluate the features. The possible search strategies in the space of feature subsets include *backward elimination* and *forward selection*. The performance criterion of the wrapper model in (John *et al.*, 1994) is the prediction

Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2000)
Stockholm, Sweden, June 5-6, 2000
(M. Jeusfeld, H. Shu, M. Staudt, G. Vossen, eds.)

http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-28/

accuracy of a data mining algorithm, estimated by *n*-fold cross validation.

A new book on feature selection by Liu and Motoda (1998) suggests a unified model of the feature selection process. Their model includes four parts: feature generation, feature evaluation, stopping criteria, and testing. In addition to the "classic" evaluation measures (accuracy, information, distance, and dependence) that can be used for removing irrelevant features, they mention important *consistency* measures (e.g., *inconsistency rate)*, required to find a *minimum* set of relevant features. By decreasing the inconsistency rate of data, both irrelevant and redundant features are removed. However, as indicated by Liu and Motoda (1998), consistency measures are only suitable for selecting discrete features.

An enhanced greedy algorithm, based on the wrapper model, is presented by Caruana and Freitag (1994). Again, the metric used is the generalization performance of the learning algorithm (its accuracy over the validation data set), which increases the computation time of the entire process.

An information-theoretic method for selecting relevant features is presented by Almuallim and Dietterich (1992). In their Mutual-Information-Greedy (MIG) Algorithm defined for Boolean noise-free features, the feature is selected if it leads to the minimum conditional entropy of the classification attribute. Since the data is assumed being noise-free, no significance testing is required (any non-zero entropy is significant). The above assumptions leave the MIG algorithm at quite a distance from most practical problems of reducing data warehouse dimensionality.

Kira and Rendell (1992) have suggested an efficient feature selection algorithm, called Relief, which evaluates each attribute by its ability to distinguish among instances that are near each other. Their selection criterion, the *feature relevance*, is applicable to numeric and nominal attributes. The greatest limitation of Relief is its inability to identify redundant features within a set of relevant features. Consequently, the set of features selected by Relief may not be optimal.

To sum-up this section, the backward elimination strategy is very inefficient for reducing dimensionality of real-world data warehouses, which may have hundreds and thousands of original attributes. On the other hand, the forward selection wrapper methods are highly expensive in terms of the computational effort. The filter algorithms are computationally cheaper, but they usually fail to remove all redundant features. In the next section, we are describing the information-theoretic method of feature selection in data warehouses, based on the knowledge discovery procedure introduced by us in (Maimon, Kandel, and Last, 1999). The procedure integrates feature selection with a highly scalable data mining algorithm, leading to elimination of both irrelevant and redundant features. The method description is followed by a case study of feature selection in a direct marketing database. We conclude the paper with discussing the potential enhancements of the proposed approach.

2 Information-Theoretic Method of Feature Selection

The method selects features by constructing *information-theoretic connectionist networks*, which represent interactions between the classifying features and each dependent attribute (a key performance indicator). The method is based on the extended relational data model, described in sub-section 2.1. In sub-section 2.2, we present the main steps of the feature selection procedure. Extraction of functional dependencies from the constructed networks is covered by sub-section 2.3. Finally, in sub-section 2.4, we evaluate the computational complexity of the algorithm.

2.1 Extended Relational Data Model

We use the following standard notation of the relational data model (see Korth and Silberschatz, 1991):

1) R - a relation schema including N attributes ($N \geq 2$). A relation is a part of the operational database.

2) A_i - an attribute No. i. $R = (A_1, ..., A_N)$.

3) D_i - the domain of an attribute A_i. We assume that each domain is a set of M_i discrete values. $\forall i: M_i \geq 2$, finite. For numeric attributes, the domain is a set of adjacent intervals. The discretization of classifying continuous attributes is performed automatically in the process of feature selection (see next sub-section). Continuous KPI's may be discretized manually into pre-defined intervals.

4) V_{ij}- a value No. j of domain D_i. Consequently, $D_i = (V_{i1},..., V_{iMi})$. For discretized numeric attributes, each value represents an interval between two continuous values.

5) r - a relation instance (table) of the relation schema R.

6) n - number of tuples (records) in a relation r ($n \geq 2$).

7) $t_k[A_i]$ - value of an attribute No. i in a tuple (record) No. k. $\forall k,i: \quad t_k[A_i] \in D_i$.

To find the set of classifying attributes in a relation of the operational database, we make the following partition of the relation schema:

1) O - a subset of *target* (dependent) attributes ($O \subset R$, $|O| \geq 1$). These attributes represent the key performance indicators (KPI's) of an organization. Such attributes are referred as *facts* in the data warehouse logical model (Inmon, 1994). Each connectionist network is aimed at predicting the values of a KPI, based on the values of *input* attributes (see below).

2) C - a subset of *candidate input* attributes ($C \subset R$, $|C| \geq 1$). This is a subset of attributes (features), which *can be* related to the *target* attributes.

3) I_i - a subset of *input* attributes (classifying features) selected by the algorithm as related to the target attribute i ($\forall i$: $I_i \subset C$). These attributes are going to be loaded as *dimensions* into the data warehouse.

Assumptions:

1) $\forall i : I_i \cap O = \varnothing$ (An attribute cannot be both an input and a target).

2) $\forall i : I_i \cup O \subseteq R$ (Some attributes may be neither input, nor target). For example, the attributes used as primary keys in the operational database (like a social security number) may be completely useless for a data warehouse, which has primary keys of its own. Sometimes, (e.g., in health care) the identifying attributes are removed from the data warehouse to preserve the privacy of the historic records.

2.2 Feature Selection Procedure

The main steps of the feature selection procedure are given below.

Step1 - Obtain the relation schema (name, type, and domain size of each attribute) and the schema partition into a subset of *candidate input* and a subset of *target* attribute (see the extended relational model in sub-section 2.1 above).

Step 2 - Read the relation tuples (records) from the operational database. Tuples with illegal or missing target values are ignored by the algorithm.

Step 2.1- Encode missing values of candidate input attributes in a pre-determined form.

Step 2.2- Discretize each continuous attribute by maximizing its mutual information with the target attribute. Mutual information (see Cover, 1991) is defined as a decrease in the uncertainty of one attribute, given the value of another attribute.

Step 3 - Enter minimum significance level for splitting a network node (default = 0.1%). This significance level is used by the likelihood ratio test (see Step 4.2.1.2.3 below).

Step 4 - Repeat for every target attribute (KPI) *i*:

Step 4.1 - Initialize the information-theoretic network (one hidden layer including the root node associated with all tuples, no input attributes, one target layer for values of the target attribute). A new network is built for every target attribute. An example of the initial network structure for a three-valued target attribute is shown in Figure 1.

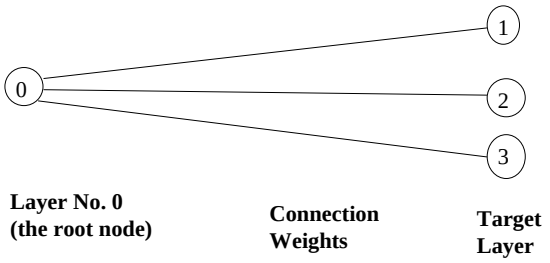

Figure 1: Information-Theoretic Connectionist Network - Initial Structure

Step 4.2 - Repeat for the maximum number of hidden layers (default = number of candidate input attributes).

Step 4.2.1 - Repeat for every candidate input attribute *i'* which is still not an input attribute:

Step 4.2.1.1 - Initialize to zero the degrees of freedom, the estimated conditional mutual information, and the likelihood-ratio statistic of the candidate input attribute and the target attribute, given the final hidden layer of nodes. Conditional mutual information (Cover, 1991) measures the net decrease in the entropy (uncertainty) of the dependent attribute due to adding information about each new classifying attribute. If the target is completely independent of an attribute, the conditional mutual information, given that attribute is zero.

Step 4.2.1.2 - Repeat for every node of the final hidden layer:

Step 4.2.1.2.1 - Calculate the estimated conditional mutual information of the candidate input attribute and the target attribute, given the node.

Step 4.2.1.2.2 - Calculate the likelihood-ratio statistic of the candidate input attribute and the target attribute, given the node.

Step 4.2.1.2.3 - If the likelihood-ratio statistic is significant, mark the node as "splitted" and increment the conditional mutual information of the

candidate input attribute and the target attribute, given the final hidden layer of nodes; else mark the node as "unsplitted".

Step 4.2.1.2.4 - Go to next node.

Step 4.2.1.3 - Go to next candidate input attribute.

Step 4.2.2 - Find the candidate input attribute maximizing the estimated conditional mutual information. According to the information theory, this attribute is the best predictor of the target, given the values of the other input attributes.

Step 4.2.3 - If the maximum estimated conditional mutual information is greater than zero:

- Make the best candidate attribute an input attribute.
- Define a new layer of hidden nodes for a Cartesian product of splitted hidden nodes of the previous layer and values of the best candidate attribute.
- Record the tuples associated with every node of the new layer (a new node is defined if there is at least one tuple associated with it).

Else **stop** the search and output the subset I_i of *input* attributes (classifying features) associated with the target attribute *i*.

Step 4.3 - Go to next target attribute.

Step 5 - Define the set *I* of selected attributes (dimensions) as the *union* of sets of input attributes with respect to every target attribute by:

$$I = \bigcup_{i=1}^{|O|} I_i$$

Step 6 - **End**.

In Figure 2, a structure of a two-layered network (based on two selected input attributes) is shown. The first input attribute has three values, represented by nodes no. 1,2, and 3 in the first layer, but only nodes no. 1 and 3 are splitted due to the statistical significance testing in Step *4.2.1.2* above. The second layer has four nodes standing for the combinations of two values of the second input attribute with two splitted nodes of the first layer. Like in Figure 2, the target attribute has three values, represented by three nodes in the target layer.

Details of the network construction procedure for a single target attribute are provided in (Maimon et al., 1999). The application of the algorithm to design of data warehouses is presented here for the first time.

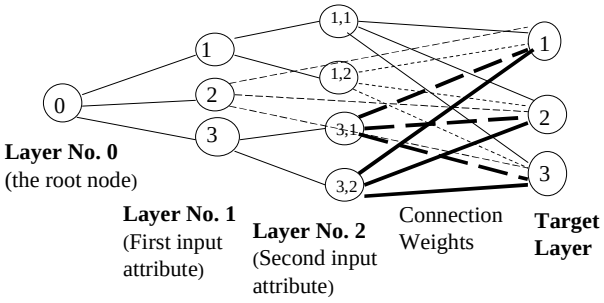

Figure 2: Information-Theoretic Connectionist Network - Two-Layered Structure

2.3 Extracting Functional Dependencies

Each connection between a terminal (unsplitted / final layer) node and a target node in the information-theoretic network represents an association rule between a conjunction of input attribute-values and a target value. Due to the inherent noisiness of the real-world data, this rule can be considered a *weak* (probabilistic) functional dependency as opposed to deterministic functional dependencies of primary and alternate keys (Schouten, 1999). In the relational model (see Korth and Silberschatz, 1991), a functional dependency between sets of attributes *X* and *Y* means that the values of the *Y* component of a tuple are *determined by* the values of the *X* component. On the contrast, an association rule between *X* and *Y* has a more limited meaning: given values of *X*, we can estimate the *probability distribution* of *Y*.

An information-theoretic weight of each rule is given by:

$$w_z^{ij} = P(V_{ij};z) \bullet \log \frac{P(V_{ij} / z)}{P(V_{ij})}$$

Where

$P(V_{ij};z)$ - an estimated joint probability of the target value V_{ij} and the node *z*.

$P(V_{ij}/z)$ - an estimated conditional (*a posteriori*) probability of the target value V_{ij}, given the node *z*.

$P(V_{ij})$ - an estimated unconditional (*a priori*) probability of the target value V_{ij}.

The connection weights express the mutual information between hidden and target nodes. A connection weight is positive if the conditional probability of a target attribute value, given the node, is higher than its unconditional probability and negative otherwise. A weight close to zero means that the target attribute value is almost independent of the node value. This means that each rule having a positive connection weight can be interpreted as *if node, then target value*. Accordingly, a negative weight refers to a rule of the form *if node, then* ***not*** *target value*.

The most informative rules can be found by ranking the information-theoretic connection weights (w_z^{ij}) in decreasing order. Both the rules having the highest positive and the lowest negative weights are of potential interest to a user. The sum of connection weights at all unsplitted and final layer nodes is equal to the estimated mutual information between a set of input attributes and a target attribute (see the definition of mutual information in Cover, 1991). According to the well-known Pareto principle, a small number of informative rules are expected to explain a major part of the total mutual information, which agrees with the experimental results presented in the next section.

2.4 Computational Complexity

To calculate the computational complexity of the feature selection procedure, we are using the following notation:

n - total number of tuples in a training data set

$|C|$ - total number of candidate input attributes

p - portion of significant input attributes, selected by the search procedure

m - number of hidden layers (input attributes), $m \le |C|$

$|O|$ - total number of target attributes

M_C - maximum domain size of a candidate input attribute

M_T - maximum domain size of a target attribute

The computational "bottleneck" of the algorithm is calculating the estimated conditional mutual information $MI(A_{i'}; A_i / z)$ of the candidate input attribute $A_{i'}$ and the target attribute A_i, given a hidden node z. Since each node of m-th hidden layer represents a conjunction of values of m input attributes, the total number of nodes at a layer No. m is apparently bounded by $(M_C)^m$. However, we restrict defining a new node (see step 4.2.3 above) by the requirement that there is at least one tuple associated with it. Thus, the total number of nodes at any hidden layer cannot exceed the total number of tuples (n). In most cases the number of nodes will be much smaller than n, due to tuples having identical values and the statistical significance requirement of the likelihood-ratio test.

The calculation of $MI(A_{i'}; A_i / z)$ is performed at each hidden layer of every target attribute for all candidate input attributes at that layer. The summation members of $MI(A_{i'}; A_i / z)$ refer to a Cartesian product of values of a candidate input attribute and a target attribute. The number of hidden layers is equal to $p|C|$. This implies that the total number of calculations is bounded by:

$$|O| \bullet \sum_{m=0}^{p|C|} n \bullet M_T \bullet M_C \bullet (|C| - m) \le \frac{|O| \bullet n \bullet M_T \bullet M_C \bullet |C|^2 \bullet p \bullet (2-p)}{2}$$

Thus, the feature selection algorithm can be applied to large-scale databases in a time, which is linear in the number of records and quadratic polynomial in the number of candidate input attributes. It is also directly proportional to the number of target attributes (key performance indicators).

3 Case Study: Direct Marketing Database

3.1 Background and Objectives

The original source of data is the Paralyzed Veterans of America (PVA), a non-profit organization that provides programs and services for US veterans with spinal cord injuries or disease. With an in-house database of over 13 million donors, PVA is also one of the largest direct mail fundraisers in the US. The data set presents the results of one of PVA's recent fund raising appeals. This mailing was sent to 3.5 million PVA donors who were on the PVA database as of June 1997. Everyone included in this mailing had donated at least once to PVA.

One group that is of particular interest to PVA is "Lapsed" donors. These individuals donated to PVA 13 to 24 months ago. They represent an important group to PVA, since the longer someone goes without donating, the less likely they will be to give again. Therefore, recapture of these former donors is a critical aspect of PVA's fund raising efforts. The data set to be analyzed includes all lapsed donors, who received the mailing (responders and non-responders). The total dollar amount of gift is given for each responder. The attributes extracted from the database can be used for understanding the behavior of both the most and the least profitable individuals. This important

insight may lead to more successful promotion campaigns in the future.

The dataset of lapsed donors, extracted from the PVA database, is publicly available on the UCI KDD Archive [http://kdd.ics.uci.edu] as KDD Cup 1998 Data. It has been originally used for the Second International Knowledge Discovery and Data Mining Tools Competition in 1998.

3.2 Database Characteristics

The special characteristics of the Direct Marketing database include the following:

- *Dimensionality*. The data set to be used contains 95,412 tuples, including 5.1 % (4,843 cases) of responders to the mailing. Each tuple has 481 attributes, namely one key, 478 input and 2 target attributes.
- *Input Attributes*. There are 76 character (nominal) and 402 numeric (continuous) attributes. These attributes include donor demographic data (as collected by PVA and third-party data sources), the promotion / donation history, and the characteristics of donors neighborhood, as collected from the 1990 US Census. For privacy reasons, no identifying data (like the donor name, address, etc.) has been included in the data set.
- *Target Attributes*. There are two target attributes in each tuple: the binary indicator for response (the attribute **TARGET_B**) and the dollar amount of the donation (the attribute **TARGET_D**). Naturally, the donation amount is zero for all non-responders. Both can be considered as key performance indicators (KPI's) for the fund raising organization.
- *Data Quality*. As indicated in the database documentation, some of the fields in the analysis file may contain data entry and/or formatting errors.
- *Missing Values*. Most input attributes contain a certain amount of missing values, which should be inferred from known values at the pre-processing stage.

3.3 Data Pre-processing

The pre-processing tasks included the following:

- *Attribute decoding.* The original attributes, presenting donor promotion history and status, contain codes, where each byte has a different meaning (e.g., recency, frequency and amount of donation). These codes have been decoded by splitting each encoded attribute into several separate attributes. The decoding operation has increased the total number of attributes from 481 to 518.
- *Missing values.* Missing values have been replaced with the mode (the most frequent value of an attribute). As recommended by the data documentation, attributes containing 99.5 and more missing values have been omitted from the analysis.
- *Rare values.* All values of nominal attributes that occur less than 100 times in the data set have been encoded as "Other".
- *Transformation by division.* To decrease the number of distinct values for large scale continuous attributes, the values of some attributes have been divided by a constant factor (10, 100, etc.) and then rounded off to the nearest integer number.
- *Discretization of Target Attribute.* The continuous target attribute TARGET_D has been discretized to equal width intervals of $10 donation each. The total number of discretization intervals has been 20, covering the attribute range between $0 and $200.
- *Discretization of Input Attributes.* Numeric attributes have been discretized by using the significance level of 99%. For 191 attributes (out of 404), no statistically significant partition has been found and these attributes have been omitted from the further stages of the analysis. The remaining 213 attributes have been left as candidate input attributes.

3.4 Feature Selection

The process of feature selection in the Direct Marketing database requires building separate networks for two KPI's: **TARGET_B** (the binary indicator for response) and **TARGET_D** (the dollar amount of the donation). However, when we have applied the dimensionality reduction procedure of sub-section 2.2 above to the first target attribute (TARGET_B), no significant input attributes have been found. In other words, no candidate input attribute, presenting in the database, is relevant to the fact of somebody responding to the mailing. Thus, we proceed with the results of the information-theoretic network built for the second target attribute (TARGET_D) only.

The dimensionality reduction procedure of sub-section 2.2 above has been applied to all tuples of responders to the raising appeal (4,843). The algorithm has selected five significant input attributes, which are about 1% of the original candidate input attributes (478). Thus, the resulting information-theoretic network includes five hidden layers only. The selected attributes and their information-theoretic scores are presented in Table 1 below.

Table 1 shows three information-theoretic measures of association between the input attributes and the target attribute: Mutual Information, Conditional Mutual Information, and Conditional Entropy. All these parameters are based on the notion of *Entropy* (see Cover, 1991), which represents the uncertainty of a random variable. The entropy is measured in *bits*. Information on input attributes, associated with the target, can decrease the uncertainty and the resulting entropy of the target.

The column "Mutual Information" shows the cumulative association between a subset of input attributes, selected up to a given iteration inclusively, and the target attribute. The next column, "Conditional MI (Mutual Information)" shows the net decrease in the entropy of the target attribute "Target_D" due to adding each input attribute. The last column ("Conditional Entropy") is equal to the difference between the unconditional entropy of Target_D (1.916 bits) and the estimated mutual information.

As you can see from the description column, the first four attributes selected represent the person's donation history, while the fifth significant input attribute characterizes the donor neighborhood. The most significant attribute LASTGIFT contributes alone about 90% of total mutual information. From the viewpoint of interestingness, the last attribute (TPE13) seems to be the most unexpected one. It is rather unreasonable that an average human expert in donations and direct mailing would pick this attribute out of the list of more than 500 attributes.

If the PVA organization is designing a data warehouse for supporting its fundraising activities, the results of the feature selection algorithm can cause a dramatic reduction in the amount of stored data. Given that an average attribute (field) requires four bytes of memory, one donor tuple (record) of 478 input attributes takes about 2k bytes. Since PVA has a donor database of about 13 million records, dimensionality reduction of 99% means saving about **25 GB** of computer storage *at the time of data creation* (1997). Perhaps, PVA could be interested in storing additional information in its data warehouse, based on its business expertise, but still the above results suggest that the majority of operational data is not needed for loading into the data warehouse: that data is either redundant, or completely irrelevant to PVA's fundraising performance.

Table 1: Direct marketing database – dimensionality reduction procedure

Iteration	Attribute Number	Attribute Name	Attribute Description	Mutual Information	Conditional MI	Conditional Entropy
1	**318**	**LASTGIFT**	**Dollar amount of most recent gift**	**0.6976**	**0.6976**	**1.2188**
2	316	MAXRAMNT	Dollar amount of largest gift to date	0.7599	0.0624	1.1564
3	314	MINRAMNT	Dollar amount of smallest gift to date	0.7671	0.0072	1.1492
4	319	LASTDATE	Date associated with the most recent gift	0.7711	0.004	1.1452
5	233	TPE13	Percent Traveling 15 - 59 Minutes to Work	0.7737	0.0025	1.1427

3.5 Functional Dependencies

The connections between input and target nodes in the information-theoretic network, constructed in the previous sub-section, have been used to extract disjunctive association rules between the significant input attributes and the target attribute (Target_D). The total of 935 positive and 85 negative rules have been extracted. The rules have been scored by the information-theoretic weights of their connections (see sub-section 2.3 above).

In Table 2 below, we are presenting only the rules having the highest connection weights in the network. Most rules in this table indicate that there is a direct relationship between the donation history and the actual amount of donation (TARGET_D). Thus, rule no. 2 says that if the last gift is between $20.5 and $28, then the actual donation is expected to be between $20 and $29 (almost in the same range). A similar interpretation can be given to rule no. 3: those who were poor donors for the last time are expected to preserve their behavior. Rule no. 1 is a slight but interesting exception: it says that those who donated about $20 for the last time are expected to *increase* their donation by as much as 50%.

Table 2: Direct marketing database – highest positive connection weights

Rule	LASTGIFT	MAXRAMNT	MINRAMNT	TARGET_D	weight	Cum. Weight	Cum. Percent
1	$19 - $20.25			$20 - $29	0.1164	0.1164	11.9%
2	$20.5 - $28			$20 - $29	0.0859	0.2023	20.7%
3	$1 - $5			$0 - $9	0.0841	0.2864	29.3%
4	$12 - $14			$10 - $19	0.0596	0.346	35.4%
5	$1 - $5	$7 - $9		$0 - $9	0.0534	0.3994	40.9%
6	$1 - $5	$10 - $11		$0 - $9	0.0402	0.4396	45.0%
7	$6 - $7	$7 - $9		$0 - $9	0.0392	0.4788	49.0%
8	$10 - $11	$10 - $11		$10 - $19	0.0377	0.5165	52.9%
9	$20.5 - $28			$30 - $39	0.0307	0.5472	56.0%
10	$15 - $15.5	$15 - $15.5	$4.68 - $8.99	$10 - $19	0.0293	0.5765	59.0%

4 Conclusions

The paper presents a method for automated selection of attributes in a data warehouse. The method is based on the information-theoretic approach to knowledge discovery in databases (Maimon et al., 1999). It does not require any prior knowledge about the business domain, can eliminate both irrelevant and redundant features, and is applicable to any type of data (nominal, numeric, etc.). While no automated procedure can fully replace a human expert, the decisions of data warehouse designers can certainly be supported by a feature selection system presented here. As shown by the direct marketing example, integrating the automated feature selection in the design process can lead to a significant dimensionality reduction of data warehouse.

Developing automated approaches to the design of data warehouses requires further investigation of several issues. For example, automated methods need to be developed for determining the set of transformations to be applied to the original attributes. A cost associated with each attribute (e.g., cost of third-party data) can be integrated in the feature selection procedure. The data model (see sub-section 2.1 above) can be further extended to include candidate input attributes from multiple relations (via foreign keys). Another important problem is detecting dynamic changes in the weak functional dependencies and updating the data warehouse structure accordingly.

References

H. Almuallim and T. G. Dietterich, *Efficient Algorithms for Identifying Relevant Features*, Proc. of 9th Canadian Conf. on AI, pages 38-45, Morgan Kaufmann, 1992.

S.D. Bay, *The UCI KDD Archive [http://kdd.ics.uci.edu]*, Irvine, CA: University of California, Department of Information and Computer Science,1999.

R. Caruana and D. Freitag, *Greedy Attribute Selection*, Proc. of 11th Conf. On Machine Learning, pages 28-36, Morgan Kaufmann, 1994.

T. M. Cover, *Elements of Information Theory*, Wiley, New York, 1991.

J.F. Elder IV and D. Pregibon, *A Statistical Perspective on Knowledge Discovery in Databases*, In *Advances in Knowledge Discovery and Data Mining*, U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. AAAI/MIT Press, Menlo Park, CA, pp. 83-113, 1996.

V. R. Gupta, *An Introduction to Data Warehousing*, System Services Corporation, Chicago, Illinois, 1997.

W.H. Inmon and R.D. Hackathorn, *Using the Data Warehouse*, John Wiley & Sons, New York, 1994.

G. H. John, R. Kohavi, and K. Pfleger, *Irrelevant Features and the Subset Selection Problem*, Proc. of the 11th Int'l Conf. on Machine Learning, pages 121-129, Morgan Kaufmann, 1994.

K. Kira and L.A. Rendell, *The Feature Selection Problem: Traditional Methods and a New Algorithm*, Proc. of AAAI'92, pages 129-134, 1992.

H.F. Korth and A. Silberschatz, *Database System Concepts*, McGraw-Hill, Inc., New York, 1991.

H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer, Boston, 1998.

O. Maimon, A. Kandel, and M. Last, *Information-Theoretic Fuzzy Approach to Knowledge Discovery in Databases*. In *Advances in Soft Computing - Engineering Design and Manufacturing*, R. Roy, T. Furuhashi and P.K. Chawdhry, Eds. Springer-Verlag, London, pp. 315-326, 1999.

H. Schouten, *Analysis and Design of Data Warehouses*, Proc. International Workshop on Design and Management of Data Warehouses (DMDW'99), Heidelberg, Germany, 1999.