Boolean Function Complementation Based Algorithm for Data Discretization

Grzegorz Borowik

Institute of Telecommunications, Warsaw University of Technology, Warsaw, Poland
G.Borowik@tele.pw.edu.pl

Abstract. This paper presents a fast algorithm for discretization of decision tables. An important novelty of the proposed solution is the application of the original algorithm of Boolean function complementation, which is a basic procedure of the field of logic synthesis, in the process of discretizing the data. This procedure has already been used by the author to calculate reducts of decision tables, where the time of calculation has been significantly reduced. It yields the idea of using the algorithm of complementation in the process of discretization. The algorithm has been generalized for the discretization of inconsistent decision tables and is used in the processing of numerical data from various fields of technology, especially for multimedia data.

Keywords: discretization, quantization, data mining, Boolean function complementation, telecommunications, biomedical engineering.

1 Introduction

The largest branch of data mining, widely known as knowledge discovery in databases, is a rapidly growing discipline of computer science with a wide range of applications, including telecommunications, biomedical engineering, banking, etc., and especially the processing of multimedia data.

One of the major applications of data mining algorithms in telecommunications is anomaly detection in telecommunications networks and systems. Since the decision of anomaly detection is based on a combination of decision rules generated by the algorithm for the training data, the algorithm is the standard procedure for machine learning. The system creates a knowledge base containing patterns of analyzed anomalies. Then, using the algorithm of decision-making and classification, it creates a set of decision rules classifying the current data. A characteristic example of training data is the database for e-mail classification [14], which contains 58,042 records represented by 64 attributes, for which the objective of the algorithm is to obtain decision rules classifying data according to the following conditions: y_spam, n_spam, other, etc.

Another application of data mining algorithms is to support medical diagnosis of various diseases. Then the main task of the algorithm is the induction of decision rules on the basis of the medical research results from the database of many patients. The decision rules induced (also called classifiers) allow diagnosis of new

R. Moreno-Díaz et al. (Eds.): EUROCAST 2013, Part II, LNCS 8112, pp. 218–225, 2013.

patients. A typical example of a database and its analysis is the Breast Cancer Wisconsin Database (source: Dr. William H. Wolberg, University of Wisconsin Hospital, Madison, Wisconsin, USA) where the diagnosis of breast cancer for a new patient is supported by the database of nine attributes and collected for 699 patients [9, 15]. Another example is the analysis of the Pima Indians Diabetes Database of eight attributes and 768 female patients (source: National Institute of Diabetes and Digestive and Kidney Diseases, Maryland, USA), where the diagnostic binary-valued decision attribute investigates whether the patient shows signs of diabetes according to World Health Organization criteria [13, 15].

The most common use of data mining algorithms combining various fields of application is processing of multimedia data. This is particularly evident in biomedical engineering, where data collected for hundreds of variables describe medical parameters / measurements of patients and thus there is a need to process large collections of multimedia data. For example, in the paper [5] the classification of patients with Alzheimer's disease was described. It was carried out basing on brain imaging magnetic resonance imaging (MRI), since it is very important to use a non-invasive method to obtain images inside the objects.

A significant difficulty in implementing such decision-making systems is determined by efficient discretization of numeric data. For example, the attributes of the Pima Indians Diabetes Database include: number of times pregnant, plasma glucose concentration a 2 hours in an oral glucose tolerance test, diastolic blood pressure (mm Hg), triceps skin fold thickness (mm) 2-hour serum insulin (μU/ml) body mass index (weight in kg / (height in m)^2) diabetes pedigree function, age (years), and class variable (0 or 1). Most of these features are numeric, so for a proper analysis of this database it is necessary to discretize/quantize the data. A similar problem we face in the classification of electronic mails where records characterizing various network parameters used to perform anomaly analysis are often given as numeric values.

The primary method of data discretization works by determining the ranges of numeric data which ultimately represent discrete attributes. Thus, the initial ranges yielded from a proposed set of cuts are then analyzed in order to obtain a minimum set of cuts differentiating objects of distinguished decision classes. Usually the selection of a minimum set of cuts is made by a transformation of Boolean formula of a conjunction normal form (CNF) into a disjunction normal form (DNF) [7]. As it is known, this transformation can be reduced to the problem of searching of all prime implicants of Boolean formula, which is the issue of the non-polynomial complexity [6, 10] and one of the bottlenecks of the rough set theory [7], e.g. the transformation proposed in [12] is an ineffective Boolean function minimization procedure. Moreover, recently expanding databases, both in the number of instances as well as in the number of attributes, noticeably cut down the effectiveness of the existing data mining algorithms.

An important novelty of the proposed solution in this paper is the application of Boolean function complementation algorithm to the transformation of CNF into DNF. The algorithm has already been used by the author to calculate reducts of decision tables where time of calculation has been significantly reduced [3]. It yields the idea of using the algorithm in the process of discretization. This application is possible due to the fact that the Boolean expression is a monotonic CNF which can be represented by a binary matrix. Thus, the transformation process of CNF into DNF can be reduced to the calculation of minimum column covers of this matrix [1,8].

The paper has also resolved the problem of discretization of the inconsistent decision-making systems, expanding the idea presented in [7].

The structure of the paper is as follows: the second chapter presents the concepts and definitions, the third chapter presents an algorithm of discretization for consistent decision-making systems by applying logic synthesis procedure, i.e. fast algorithm Boolean function complementation, then the problem is generalized for inconsistent decision-making systems; the paper ends with a summary.

2 Preliminaries

Let $\mathcal{A} = (U, A \cup \{d\})$ be a decision system, where $U = \{u_1, u_2, \ldots, u_n\}$ is a set of objects, $A = \{a_1, \ldots, a_m\}$ a set of condition attributes, d – decision attribute. Values of attributes are determined by a function from the set U to V_a, where V_a is a domain of $a \in A$. Then, a function ρ maps the product $U \times A$ into the value set V_a. By $\rho(u, a)$, where $u \in U$, $a \in A$, we denote the value of the attribute a for an object u. We assume that values of each conditional attribute belong to a fixed interval of real numbers $\rho(u, a) \in V_a = [l_a; r_a] \subset \mathcal{R}$, decision d is discrete and mapping $U \to d$ is unambiguous (decision system is consistent).

A pair $(a, c_{a(k)})$, where $a \in A$, $c_{a(k)} \in V_a$, which is a real interval $[l; r]$, we call a *cut* on a domain V_a. Then,

$$P_a = \{[c_{a0}; c_{a1}), [c_{a1}; c_{a2}), \ldots, [c_{at}; c_{a(t+1)}]\},$$

we call a *partition* of V_a into t subintervals, where $l = c_{a0} < c_{a1} < \ldots < c_{at} < c_{a(t+1)} = r$ and $[c_{a0}; c_{a1}) \cup [c_{a1}; c_{a2}) \cup \ldots \cup [c_{at}; c_{a(t+1)}] = [l; r]$. Cuts are uniquely defined for each attribute value range and the number of cuts is denoted by $t(a)$.

A set of cuts / partitions $P = \bigcup_{a \in A} P_a$ for a decision system $\mathcal{A} = (U, A \cup \{d\})$ defines a new discrete decision system $\mathcal{A}^P = (U, A^P \cup \{d\})$, where set of attributes $A^P = \{a^P : a \in A\}$ and $\rho(u, a^P) = k$, iff $\rho(u, a) \in [c_{ak}; c_{a(k+1)})$, $u \in U$ and $k \in \{0, \ldots, t(a)\}$.

3 Decision System Discretization Algorithm

Let $\mathcal{A}$ be a decision system given in Table 1 where attribute value domains are as follows: $\rho(u_t, a) \in [1; 4]$, $\rho(u_t, b) \in [0; 2]$.

Discretization of the decision system lies in construction of P_a partitions for each attribute domain V_a. Then, the real value of the attribute is converted into subinterval consisting given attribute value.

In the first step of the construction we propose a set of cuts determined by ordered attribute values and different from these values. We assume that

Table 1. Example of continuous decision system

$\mathcal{A}$	a	b	d
u_1	2.6	1.5	0
u_2	2.0	0.25	0
u_3	1.6	1.0	1
u_4	2.8	0.5	1
u_5	2.8	1.0	0
u_6	3.2	1.5	1
u_7	1.8	0.4	0
u_8	2.6	0.5	1

subinterval corresponds to only one point, for example the arithmetic mean. Thus, we obtain the following set of cuts:

$$c_{a1} = (a, 1.7), c_{a2} = (a, 1.9), c_{a3} = (a, 2.3), c_{a4} = (a, 2.7), c_{a5} = (a, 3.0),$$
$$c_{b1} = (b, 0.325), c_{b2} = (b, 0.45), c_{b3} = (b, 0.75), c_{b4} = (b, 1.25).$$

It may be noted that a single cut defines a new binary conditional attribute; e.g. for attribute a and cut $(a; 1.9)$ we assume '0' when $\rho(u_t, a) < 1.9$, otherwise we assume '1'. In other words, objects located on different sides of the $\rho = 1.9$ are distinguished by this cut. Hence, boundary cuts have been omitted since they do not distinguish any values.

In the second step we obtain a minimal set of cuts. Let C be a set of proposed cuts, i.e. $C = \{c_{a1}, c_{a2}, c_{a3}, c_{a4}, c_{a5}, c_{b1}, c_{b2}, c_{b3}, c_{b4}\}$. Let $\chi(u_p, u_q)$ be a discernibility function constructed according to the given set of cuts C and a pair of objects (u_p, u_q) belonging to different decision classes; e.g. to distinguish object u_1 from u_3 we use the cut c_{a1} or c_{a2} or c_{a3} or c_{b4}. Then:

$$\begin{aligned}
\chi(u_1, u_3) &= c_{a1} \vee c_{a2} \vee c_{a3} \vee c_{b4} \\
\chi(u_1, u_4) &= c_{a4} \vee c_{b3} \vee c_{b4} \\
\chi(u_1, u_6) &= c_{a4} \vee c_{a5} \\
\chi(u_1, u_8) &= c_{b3} \vee c_{b4} \\
\chi(u_2, u_3) &= c_{a1} \vee c_{a2} \vee c_{b1} \vee c_{b2} \vee c_{b3} \\
&\vdots \\
\chi(u_7, u_8) &= c_{a2} \vee c_{a3} \vee c_{b2}
\end{aligned}$$

Thus, to distinguish each pair of objects of different decisions we create a Boolean expression which is a conjunction of the above formulas. Transforming the resulting Boolean formula, i.e. a product of sums into a sum of products we obtain all the minimal sets of cuts. In other words, each prime implicant of

Table 2.

$\mathcal{A}$	a	b	d
u_1	1	1	0
u_2	0	0	0
u_3	0	1	1
u_4	1	0	1
u_5	1	1	0
u_6	2	1	1
u_7	0	0	0
u_8	1	0	1

Table 3.

$\mathcal{A}$	a	b	d
$\{u_1, u_5\}$	1	1	0
$\{u_2, u_7\}$	0	0	0
u_3	0	1	1
$\{u_4, u_8\}$	1	0	1
u_6	2	1	1

the constructed Boolean formula corresponds to a minimal set of cuts. After the transformation of the above formula, we obtain:

$$c_{a3}c_{a5}c_{b3} \vee c_{a4}c_{b2}c_{b4} \vee c_{a2}c_{a5}c_{b1}c_{b3} \vee c_{a1}c_{a5}c_{b2}c_{b3} \vee c_{a2}c_{a5}c_{b2}c_{b3}$$
$$\vee\ c_{a3}c_{a4}c_{b3}c_{b4} \vee c_{a1}c_{a2}c_{a4}c_{b1}c_{b4} \vee c_{a1}c_{a3}c_{a4}c_{b1}c_{b4} \vee c_{a2}c_{a4}c_{b1}c_{b3}c_{b4}$$

Finally, taking as an example the first set of cuts, i.e. cuts belonging to the first product $\{c_{a3}, c_{a5}, c_{b3}\}$ and encoding corresponding subintervals:

$$P_a = \{[1; 2.3), [2.3; 3.0), [3.0; 4]\} = \{0, 1, 2\}$$
$$P_b = \{[0; 0.75), [0.75; 2]\} = \{0, 1\}$$

we obtain a discrete decision system which is shown in Table 2. Removing redundant rows we acquire the form of Table 3.

3.1 Efficient Algorithm of Boolean Function Complementation

The method proposed yields a discernibility function which is a monotonic Boolean formula in CNF. The simplification of the discernibility function is carried out by transforming the CNF into DNF. Such a transformation is of non-polynomial computational complexity and therefore it is important to use efficient algorithms which can handle this task.

An interesting approach proposed by the author is based on the fast complementation algorithm [3]. The key strength of the algorithm lies in Shannon expansion procedure of monotone function f. Then,

$$f = \overline{x}_j f_{\overline{x}_j} + f_{x_j} \tag{1}$$

This procedure is fundamental in the field of logic synthesis, however it can successfully be applied in the field of data mining.

Proposed approach benefits from the transformation (2), i.e. double complementation of a Boolean function.

$$\prod_k \sum_l x_{kl} = \overline{\overline{\prod_k \sum_l x_{kl}}} = \overline{\sum_k \prod_l \overline{x_{kl}}} \tag{2}$$

Given that the discernibility function f_M (conjunction of χ formulas) representing the CNF is unate (monotone), it can be transformed into the F form (first complementation) and then considered as a binary matrix M (Fig. 1). In fact, the task of searching the complement of function F, i.e. $\overline{F}$, can be reduced to the concept of searching of a column cover C of the binary matrix M (second complementation).

Theorem [4]. Each row i of C, the binary matrix complement of M, corresponds to a column cover L of M, where $j \in L$, iff $C_{ij} = 1$.

The approach presented significantly accelerates calculations and has already been used by the author to calculate reducts of decision tables. As a result, the time of calculation has been significantly reduced. An efficient representation of the algorithm in computational memory allows the authors to achieve results that cannot be calculated using published methods and systems. Some of the results and detailed theory can be found in [1–3]. Hence the idea of applying the algorithm of complementation in the process of discretization.

Example. Lets consider the discernibility function f_M as follows:

$$f_M = (x_2 + x_3 + x_4)(x_1 + x_2)(x_3 + x_4)(x_2 + x_3 + x_5).$$

Performing the multiplication and applying absorption law we obtain:

$$f_M = x_2x_3 + x_2x_4 + x_1x_3 + x_1x_4x_5.$$

The same result can be obtained performing the mentioned approach, i.e. double complementation of the function f_M. Then,

$$F = \overline{f}_M = \overline{x}_2\overline{x}_3\overline{x}_4 + \overline{x}_1\overline{x}_2 + \overline{x}_3\overline{x}_4 + \overline{x}_2\overline{x}_3\overline{x}_5,$$

and finally, applying Shannon expansion procedure, we calculate $\overline{F}$.

The illustrative diagram of the method has been shown in Fig. 1 and the full scheme of complementation of F using Shannon expansion in [3].

$$F = \begin{bmatrix} - & 0 & 0 & 0 & - \\ 0 & 0 & - & - & - \\ - & - & 0 & 0 & - \\ - & 0 & 0 & - & 0 \end{bmatrix} \longrightarrow M = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

$$\downarrow$$

$$\overline{F} = \begin{bmatrix} - & 1 & 1 & - & - \\ - & 1 & - & 1 & - \\ 1 & - & 1 & - & - \\ 1 & - & - & 1 & 1 \end{bmatrix} \longleftarrow C = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Fig. 1. Diagram of the proposed algorithm

3.2 Discretization Algorithm of Inconsistent Decision System

The presented algorithm of discretization also works when we deal with an inconsistent decision system.

We can make the system of Table 1 inconsistent by adding u_9, which is a copy of an object u_8 with decision equal '0' (Table 4). For this new decision system we can propose exactly the same set of cuts C, as for the system from Table 1. Calculating discernibility function $\chi(u_p, u_q)$ for each pair of objects of different decision classes we have: $\chi(u_8, u_9) = \emptyset$. Then, we assume that the conjunction of all $\chi(u_p, u_q)$ does not include $\chi(u_8, u_9)$. In general, we remove all empty functions χ, which we proceed similarly for all contradictions in the decision table. This is equivalent to the assignment of all inconsistent pairs of objects to new decision classes. Then, for decision system from Table 4 we obtain a form presented in Table 5.

Such an approach is consistent with the theory of rough sets [11] proposed by Z. Pawlak. Then, the lower approximation of objects with respect to the decision d is $\{u_1, \ldots, u_7\}$, while the upper approximation is the set of all objects.

Finally, encoding partitions:

$$P_a = \{[1; 2.3), [2.3; 2.7), [2.7; 4]\} = \{0, 1, 2\}$$
$$P_b = \{[0; 0.75), [0.75; 1.25), [1.25; 2]\} = \{0, 1, 2\}$$

we obtain discrete decision system, which after removing redundant rows takes the form of Table 6. It should be noted that the objects u_8 and u_9 remained inconsistent.

Table 4.

$\mathcal{A}$	a	b	d
u_1	2.6	1.5	0
u_2	2.0	0.25	0
u_3	1.6	1.0	1
u_4	2.8	0.5	1
u_5	2.8	1.0	0
u_6	3.2	1.5	1
u_7	1.8	0.4	0
u_8	2.6	0.5	1
u_9	2.6	0.5	0

Table 5.

$\mathcal{A}$	a	b	d
u_1	2.6	1.5	0
u_2	2.0	0.25	0
u_3	1.6	1.0	1
u_4	2.8	0.5	1
u_5	2.8	1.0	0
u_6	3.2	1.5	1
u_7	1.8	0.4	0
$\{u_8, u_9\}$	2.6	0.5	{0,1}

Table 6.

$\mathcal{A}$	a	b	d
u_1	1	2	0
$\{u_2, u_7\}$	0	0	0
u_3	0	1	1
u_4	2	0	1
u_5	2	1	0
u_6	2	2	1
u_8	1	0	1
u_9	1	0	0

4 Summary

The key idea of this paper is to use the complement of Boolean function method from logic synthesis in the field of data mining. Although the methods outlined in this paper are known, many logic synthesis methods have not been previously used or have rarely been used in the field of data mining. It is mainly due to the lack of knowledge of methods and algorithms of logic synthesis and therefore they are skipped and not used by specialists of data mining. However, they may have significant impact on the acceleration of the calculations [1–3, 8].

Acknowledgements. This work was partly supported by the Foundation for the Development of Radiocommunications and Multimedia Technologies.

References

1. Borowik, G.: Data mining approach for decision and classification systems using logic synthesis algorithms. In: Klempous, R., Nikodem, J., Jacak, W., Chaczko, Z. (eds.) Advanced Methods and Applications in Computational Intelligence. Topics in Intelligent Engineering and Informatics, vol. 6, pp. 3–23. Springer International Publishing (2014), doi:10.1007/978-3-319-01436-4_1
2. Borowik, G., Łuba, T.: Fast algorithm of attribute reduction based on the complementation of boolean function. In: Klempous, R., Nikodem, J., Jacak, W., Chaczko, Z. (eds.) Advanced Methods and Applications in Computational Intelligence. Topics in Intelligent Engineering and Informatics, vol. 6, pp. 25–41. Springer International Publishing (2014), doi:10.1007/978-3-319-01436-4_2
3. Borowik, G., Łuba, T., Zydek, D.: Features reduction using logic minimization techniques. International Journal of Electronics and Telecommunications 58(1), 71–76 (2012)
4. Brayton, R.K., Hachtel, G.D., McMullen, C.T., Sangiovanni-Vincentelli, A.: Logic Minimization Algorithms for VLSI Synthesis. Kluwer Academic Publishers (1984)
5. Cuingnet, R., Gerardin, E., Tessieras, J., Auzias, G., Lehericy, S., Habert, M.O., Chupin, M., Benali, H., Colliot, O.: Automatic classification of patients with Alzheimer's disease from structural MRI: A comparison of ten methods using the ADNI database. NeuroImage 56(2), 766–781 (2011), doi:10.1016/j.neuroimage.2010.06.013
6. Dasgupta, S., Papadimitriou, C.H., Vazirani, U.V.: Algorithms. McGraw-Hill (2008)
7. Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough sets: A tutorial (1999)
8. Łuba, T., Rybnik, J.: Rough sets and some aspects in logic synthesis. In: Słowiński, R. (ed.) Intelligent Decision Support – Handbook of Application and Advances of the Rough Sets Theory. Kluwer Academic Publishers (1992)
9. Mangasarian, O.L., Wolberg, W.H.: Cancer diagnosis via linear programming. SIAM News 23(5), 1–18 (1990)
10. Papadimitriou, C.H.: Computational complexity. Academic Internet Publ. (2007)
11. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers (1991)
12. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Słowiński, R. (ed.) Intelligent Decision Support – Handbook of Application and Advances of the Rough Sets Theory. Kluwer Academic Publishers (1992)
13. Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., Johannes, R.S.: Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In: Proceedings of the Symposium on Computer Applications and Medical Care, pp. 261–265. IEEE Computer Society Press (1988)
14. Žádník, M., Michlovský, Z.: Is Spam Visible in Flow-Level Statistics? Tech. rep., CESNET National Research and Education Network (2009), `http://www.fit.vutbr.cz/research/view_pub.php?id=9277`
15. UC Irvine Machine Learning Repository, `http://archive.ics.uci.edu/ml/`