Tolerance Based Templates for Information Systems

Piotr Synak[1,2] and Dominik Ślęzak[2,3,1]

[1] Polish-Japanese Institute of Information Technology
Koszykowa 86, 02-008 Warsaw, Poland
[2] Infobright Inc.
218 Adelaide St. W, Toronto, ON, M5H 1W8 Canada
[3] Department of Computer Science, University of Regina
3737 Wascana Parkway, Regina, SK, S4S 0A2 Canada

Abstract. We discuss generalisations of the basic notion of a template defined over information systems using indiscernibility relation. Generalisations refer to the practical need of operating with more compound descriptors, over both symbolic and numeric attributes, as well as to a more entire extension from equivalence to tolerance relations between objects. In this paper, we briefly show that the heuristic algorithms known from literature to search for templates in their classical indiscernibility-based form, can be easily adapted to the case of tolerance relations.

Keywords: Information Systems, Templates, Tolerance Relations.

1 Introduction

Information systems, studied in particular within the framework of the theory of rough sets [13, 4], provide an efficient means for data representation and analysis. Basing on attribute-based indiscernibility relation between objects (records, rows), we can express various types of *patterns* satisfied, partially satisfied, or simply enough frequently occurring, within the real-world data tables.

Among very diversified definitions of patterns, *templates* are ones of the most basic constructions, based on conjunctions of single-attribute *descriptors* of various levels of complexity and generalisation [8, 9, 11], widely applied in data mining [1, 3, 2]. In this particular paper, we discuss – besides providing a number of illustrative examples of templates and their generalisations – the need of extension of already known algorithmic framework for the template extraction onto the case of information systems with tolerance (similarity) relation instead of standard indiscernibility. Although tolerances have been already widely studied for information systems [5, 16, 20], we are the first to consider them in the context of efficient representation and search of generalised templates. In this way, we make an important step towards a wider application of templates in real-world data mining and knowledge discovery problems.

The paper is organised as follows: In Section 2 we recall basic notions related to information systems and indiscernibility relations. In Section 3, we introduce

the notion of a template for the most standard case of information systems. In Section 4, we generalise templates by building them from more sophisticated descriptors, definable for both symbolic and numeric attributes. In Section 5, we discuss examples of tolerance relations in information systems with weakened requirements for indiscernibility relations. In Section 6, we consider the corresponding tolerance based templates. In Section 7, our main contribution in this paper, we explain how the most popular algorithms for efficient template generation can be extended towards extraction of tolerance based templates from data. Finally, in Section 8, we conclude the paper.

2 Preliminaries

In the paper, we use the notation of the theory of rough sets [13, 4]. In particular, by $\mathbb{A} = (U, A)$ we denote an *information system* [12, 14] with the universe U of *objects* and the attribute set A. Each *attribute* $a \in A$ is a function $a : U \to V_a$, where V_a is the *value set* of a. For a given set of attributes $B \subseteq A$, we define the *indiscernibility relation* $IND(B)$ on the universe U that partitions U into classes of indiscernible objects. We say that objects x and y are *indiscernible* with respect to B if and only if $a(x) = a(y)$ for each $a \in B$.

The values of attributes for a given object $x \in U$ form an elementary pattern generated by $\mathbb{A}$, where an *elementary pattern* (or *information signature*) $Inf_B(x)$ is a set $\{(a, a(x)) : a \in B\}$ of attribute-value pairs over $B \subseteq A$ consistent with a given object x. By

$$INF(A) = \{Inf_B(x) : x \in U, B \subseteq A\} \tag{1}$$

we denote the set of all signatures generated by A.

3 Templates in Information Systems

One of the main tasks of data mining process is searching for patterns in data [1, 3, 2]. There are several kinds of patterns considered in the literature. In the paper we consider one kind of such patterns, which we call templates, that are defined by descriptors over the space of attributes of some information system.

Let $\mathbb{A} = (U, A)$ be an information system and $|A| = m$. A *template* T of $\mathbb{A}$ is any propositional formula $\bigwedge(a_i = v_i)$, where $a_i \in A$, $a_i \neq a_j$ for $i \neq j$, and $v_i \in V_{a_i}$ [8, 9, 11]. Assuming $A = \{a_1, \ldots, a_m\}$ one can represent any template

$$T = (a_{i_1} = v_{i_1}) \wedge \ldots \wedge (a_{i_k} = v_{i_k}) \tag{2}$$

by the sequence $[u_1, \ldots, u_m]$ where on position p is either v_p if $p = i_1 \ldots i_k$ or "*" (don't care symbol) otherwise. We say that an object $x \in U$ *satisfies* the descriptor $a = v$ if $a(x) = v$. An object x satisfies (matches) the template T if it satisfies all the descriptors of T (i.e., if $x \in \|T\|_{\mathbb{A}}$). For any template T by $length(T)$ we denote the number of different descriptors $a = v$ occurring in T

and by $fitness_{\mathbb{A}}(T)$ we denote its *fitness*, i.e., the number of objects from the universe U satisfying T. If T consists of one descriptor $a = v$ only we also write $n_{\mathbb{A}}(a, v)$ (or $n(a, v)$) instead of $fitness_{\mathbb{A}}(T)$. By the *quality* of template T we often understand the number $fitness_{\mathbb{A}}(T) \times length(T)$. If s is an integer then by $Template_{\mathbb{A}}(s)$ we denote the set of all templates with fitness not less than s.

4 Generalised Templates

The idea of a template may be extended to so called *generalised templates*, i.e., templates of the form

$$GT = (a_{i_1} = v_{i_1} \vee \ldots \vee a_{i_1} = v_{i_n}) \wedge \ldots \wedge (a_{j_k} = v_{j_1} \vee \ldots \vee a_{j_k} = v_{j_m}). \quad (3)$$

The main difference is that instead of onevalue we have manyvalued descriptors. We say that an object x *satisfies* a generalised descriptor $a = v_1 \vee \ldots \vee a = v_m$ if the value of a on x belongs to the set $\{v_1, \ldots, v_m\}$. An object x satisfies the generalised template GT if it satisfies all the descriptors of GT. Another extension of this idea is based on templates with numeric descriptors, i.e.,

$$a \in [v_{1_1}, v_{1_2}] \vee \ldots \vee [v_{m_1}, v_{m_2}]. \quad (4)$$

where, in particular, we can have $m = 1$. In the case of generalised template GT one may modify the length of a descriptor from GT by

$$l(a) = \begin{cases} 1/k & \text{if } a \text{ occured in a template} \\ 0 & \text{otherwise} \end{cases}, \quad (5)$$

where the number k is equal to length of the generalised descriptor of a. By the *quality* of a generalised descriptor of a we mean the product of $l(a)$ and the number of matching objects. Using the function l one can easily modify fitness and length functions of generalised templates. By $fitness_{\mathbb{A}}(GT)$ of GT we understand the number of objects satisfying GT. The length of GT we define by $length(GT) = \sum_{a \in A} l(a)$. The quality of template GT is defined by $fitness_{\mathbb{A}}(GT) \times length(GT)$.

5 Tolerance Relations

The indiscernibility relation – the main driving tool of rough sets – in some cases can be insufficient to deal with data, for example, when we have real value attributes. In this case every object can potentially differ from the others. The equivalence classes of indiscernibility relation can be then one- or few-element sets and thus very specific. The standard rough set approach [13] can be extended by assuming any type of binary relation instead of the equivalence relation (see, e.g., [5, 16, 20]). In this paper, we consider reflexive and symmetric relations, usually called tolerance relations. Formally, a relation $\tau \subseteq U \times U$ is a *tolerance* relation on the set U iff

1. (reflexivity) $\forall_{x \in U}\ (x,x) \in \tau$
2. (symmetry) $\forall_{x,y \in U}\ \{(x,y) \in \tau \Rightarrow (y,x) \in \tau\}$

Any tolerance relation can extend the notion of indiscernibility of objects to their similarity. For a given information system $\mathbb{A} = (U, A)$, where $A = (a_1, \ldots, a_k)$, any object is characterised by its signature, i.e., a vector of attribute values. Similarity defined on attributes can be thus easily expressed as similarity between objects. Suppose we are given a tolerance $\tau_A \subseteq INF(A) \times INF(A)$. Then,

$$\forall_{x,y \in U} \{(x,y) \in \tau \Leftrightarrow (Inf_A(x), Inf_A(y)) \in \tau_A\}. \tag{6}$$

For any object $x \in U$ we can define a *tolerance class* with respect to τ:

$$[x]_\tau = \{y \in U : (x,y) \in \tau\} \tag{7}$$

Let us emphasise that because tolerance relation is not transitive then one object may belong to two or more different tolerance classes.

There are several classes of tolerance relations considered in the literature (see, e.g., [17, 16, 11, 6]). Any class is characterised by a first order formula and some parameters which are tuned up in the optimisation process. One class of relations is based on some *similarity measures* defined on the attributes: $\delta_a : V_a \times V_a \to \mathbb{R}^+ \cup \{0\}$. Different measures can be used for different attributes. In the case of real attributes we can consider, for example, a distance between particular values. Symbolic attributes can be equipped with some preference order [17]. Sometimes, there is also a structure of attribute values given making it possible to define a similarity measure.

Once the similarity measures are defined for each attribute there can be formulated several tolerance relations. Let us list here only a few examples [11]:

1. $(x,y) \in \tau_1(\varepsilon) \Leftrightarrow \max_{a_i \in A}\{\delta_{a_i}(x,y)\} \leq \varepsilon$
2. $(x,y) \in \tau_2(\varepsilon_1, \ldots, \varepsilon_k) \Leftrightarrow \forall_{a_i \in A}[\delta_{a_i}(x,y) \leq \varepsilon_i]$
3. $(x,y) \in \tau_3(w_1, \ldots, w_k, w) \Leftrightarrow \sum_{a_i \in A} w_i \cdot \delta_{a_i}(x,y) + w \leq 0$
4. $(x,y) \in \tau_4(w) \Leftrightarrow \prod_{a_i \in A} \delta_{a_i}(x,y) \leq w$

One can also consider *local tolerance relations* defined for each attribute (thus, defined on attribute's domain set). Let $\tau_a \subseteq V_a \times V_a$ be a tolerance relation defined for attribute a. We define a tolerance relation for objects $x, y \in U$ as follows:

$$(x,y) \in \tau \Leftrightarrow \forall_{a_i \in A}(a_i(x), a_i(y)) \in \tau_{a_i}. \tag{8}$$

6 Tolerance Based Templates

Let us assume that we are given an information system $\mathbb{A} = (U, A)$. Any template T for $\mathbb{A}$ determines some pattern in data by means of all the descriptors constituting T. More precisely, the pattern is determined by attributes forming descriptors and objects satisfying them. In the case when for each attribute $a \in A$ there is defined some local tolerance relation τ_a we can extend the semantics of

a pattern by modifying definition of descriptor satisfaction by means of $(\tau_a)_{a \in A}$. We can assume that given descriptor is satisfied by an object if value from the descriptor is in the same tolerance class as the corresponding value of object. Thus, we say that an object $x \in U$ *satisfies* the descriptor $a = v$ *relatively to* τ_a if $(a(x), v) \in \tau_a$. An object x satisfies (matches) the template T if it satisfies all the descriptors of T relatively to the corresponding tolerance relations.

The same idea can be extended to generalised templates. A generalised descriptor is satisfied by an object relatively to tolerance relation if any value from descriptor is in the same tolerance class as the corresponding value of object. In the case of more complex types of descriptors (e.g, first order logic formulas) we can reason in an analogous way.

Let us emphasise that tolerance based templates may be extremely helpful when we have many different values in data, e.g., real attributes, and there are no straightforward patterns of values. Usually, in such cases some discretisation methods were required to be applied before template generation process. Taking advantage from tolerance relations makes the solution more fitted to data and more accurate.

7 Tolerance Based Templates Generation

In this section we are going to show that most of the known methods of templates generation (see [7, 10, 9, 18, 11, 6]) can be applied also to the case of tolerance based templates.

One of the methods is a greedy algorithm (see *Max I* and *Max II* method in [11]) iteratively adding optimal descriptors to initially empty template. The optimality of a descriptor is measured by its fitness, i.e., the number of objects from the universe U satisfying it (see Section 3). In the tolerance based case the method remains the same but the fitness of a descriptor is measured by number of objects that satisfy this descriptor relatively to the corresponding relation.

Another method is a heuristic based on random search of objects with respect to the attached weights (*Object weight* algorithm). Any found set of objects defines some template. The weights attached to objects are based on their average similarity to all the other objects. In this method the similarity of objects is based on the equality of values. In the tolerance based case this equality can be changed to the membership to the same tolerance class.

In the *Attribute weight* method the weights are attached to the attributes as well as to their values and the random process tries to add iteratively descriptors to the initially empty set (or remove descriptors from the template generated from some selected object). In this method after some descriptor is chosen it is tested against the fitness improvement. In the tolerance based case the descriptor satisfied by the higher number of objects relatively to the corresponding tolerance relation will have higher chance to be chosen.

Finally, the genetic algorithm used for templates generation (see [9, 19]) is based on finding an optimal permutation of attributes that for a selected base object x generates the best template. An important factor of template quality

measure is a fitness so analogously as in previous examples this method can be adopted to the tolerance based case.

8 Conclusions

We generalised the notion of a template onto the case of information systems equipped with tolerance, instead of standard indiscernibility equivalence relation. We provided examples illustrating the need of such generalisation, referring to symbolic and numeric attributes, as well as to similarity functions defined both locally, for single attributes, and globally – for attribute sets. Then, we discussed possibilities of extension of the known algorithms extracting templates from data, which would enable to deal efficiently also with tolerance based patterns.

Acknowledgements

The research has been supported by the grants 8 T11C 025 19 and 3 T11C 007 28 from the Ministry of Scientific Research and Information Technology of the Republic of Poland and by the Research Center at the Polish-Japanese Institute of Information Technology, Warsaw, Poland.

References

1. Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. The AAAI Press/The MIT Press, Cambridge, MA, 1996.
2. J. H. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, Heidelberg, Germany, 2001.
3. Willi Kloesgen and Jan Żytkow, editors. *Handbook of Knowledge Discovery and Data Mining*. Oxford University Press, Oxford, UK, 2002.
4. Jan Komorowski, Lech Polkowski, and Andrzej Skowron. Rough sets: A tutorial. In Sankar K. Pal and Andrzej Skowron, editors, *Rough Fuzzy Hybridization: A New Trend in Decision-Making*, pages 3–98. Springer-Verlag, Singapore, 1999.
5. Krzysztof Krawiec, Roman Słowiński, and Daniel Vanderpooten. Learning decision rules from similarity based rough approximations. In Polkowski and Skowron [15], pages 37–54.
6. Sinh Hoa Nguyen. *Regularity Analysis and Its Applications in Data Mining*. PhD thesis, Warsaw University, Warsaw, Poland, 2000.
7. Sinh Hoa Nguyen, Tuan Trung Nguyen, Lech Polkowski, Andrzej Skowron, Piotr Synak, and Jakub Wróblewski. Decision rules for large data tables. In *Computational Engineering in Systems Applications CESA*, pages 942–947, Lille, France, July 9-12 1996.
8. Sinh Hoa Nguyen, Tuan Trung Nguyen, and Piotr Synak. Knowledge discovery by rough set methods. In *International Conference on Information System Analysis and Synthesis ISAS*, pages 26–33, Orlando, FL, July 22-26 1996.

9. Sinh Hoa Nguyen, Lech Polkowski, Andrzej Skowron, Piotr Synak, and Jakub Wróblewski. Searching for approximate description of decision classes. In *Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery RSFD*, pages 153–161, Tokyo, Japan, November 6-8 1996.
10. Sinh Hoa Nguyen, Andrzej Skowron, and Piotr Synak. Rough sets in data mining: Approximate description of decision classes. In *Fourth European Congress on Intelligent Techniques and Soft Computing EUFIT*, pages 149–153, Aachen, Germany, September 2-5 1996. Verlag Mainz.
11. Sinh Hoa Nguyen, Andrzej Skowron, and Piotr Synak. Discovery of data patterns with applications to decomposition and classification problems. In Lech Polkowski and Andrzej Skowron, editors, *Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems*, volume 19 of *Studies in Fuzziness and Soft Computing*, chapter 4, pages 55–97. Physica-Verlag, Heidelberg, Germany, 1998.
12. Zdzisław Pawlak. Information systems - theoretical foundations. *Information Systems*, 6:205–218, 1981.
13. Zdzisław Pawlak. Rough sets. *International Journal of Computer and Information Sciences*, 11:341–356, 1982.
14. Zdzisław Pawlak. *Rough Sets: Theoretical Aspects of Reasoning about Data*, volume 9 of *D: System Theory, Knowledge Engineering and Problem Solving*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991.
15. Lech Polkowski and Andrzej Skowron, editors. *Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems*, volume 19 of *Studies in Fuzziness and Soft Computing*. Physica-Verlag, Heidelberg, Germany, 1998.
16. Andrzej Skowron and Jarosław Stepaniuk. Tolerance approximation spaces. *Fundamenta Informaticae*, 27(2-3):245–253, 1996.
17. Roman Słowiński, Salvatore Greco, and Benedetto Matarazzo. Rough set analysis of preference-ordered data. In James J. Alpigini, James F. Peters, Andrzej Skowron, and Ning Zhong, editors, *Third International Conference on Rough Sets and Current Trends in Computing RSCTC*, volume 2475 of *Lecture Notes in Artificial Intelligence*, pages 44–59, Malvern, PA, October 14-16 2002. Springer-Verlag.
18. Piotr Synak. Methods of approximate reasoning in searching for rough dependencies. Master's thesis, Warsaw University, Warsaw, Poland, 1996. In Polish.
19. Jakub Wróblewski. Genetic algorithms in decomposition and classification problem. In Polkowski and Skowron [15], chapter 24, pages 471–487.
20. Yiyu Yao, S.K. Michael Wong, and Tsau Young Lin. A review of rough set models. In Tsau Young Lin and N. Cercone, editors, *Rough Sets and Data Mining. Analysis of Imprecise Data*, pages 47–75. Kluwer Academic Publishers, Boston, MA, USA, 1997.