Dealing with not Fully Described Objects in Decision Support Systems: Alternative Approaches

[1] *Valery N. Yudin <yudin@ispras.ru>*
1,2 Leonid E. Karpov <mak@ispras.ru>
1 Ivannikov Institute for System Programming of the Russian Academy of Sciences
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia
2 Lomonosov Moscow State University,
GSP-1, Leninskie Gory, Moscow, 119991, Russia

Abstract. Not fully described objects may be seen in many different areas and applications – from medicine and up to space apparatus control. Starting to de-sign and develop a decision support system, which should work with not fully described objects, we may choose between alternatives. One of two approaches compared in this article is based on logical deduction according to a priory specified rules. Here the "IF-THEN" productions are intensively used. Another approach, which is often called case-based, assumes the presence of case base filled by real and/or artificial (model) cases. This second approach does not insist on rules and object models, but it is much closer to the mental model used when humans are thinking.

Keywords: decision support system; rule-based approach; case-based reasoning; case base; classes of equivalence; objects; features; measure of closeness.

DOI: 10.15514/ISPRAS-2018-30(1)-9

For citation: Yudin V.N., Karpov L.E. Dealing with not Fully Described Objects in Decision Support Systems: Alternative Approaches. Trudy ISP RAN/Proc. ISP RAS, vol. 30, issue 1, 2018, pp. 127-136. DOI: 10.15514/ISPRAS-2018-30(1)-9

1. Introduction

There are many practically used decision support programming systems (DSS) nowadays (see examples in [1-3]). In these systems the most difficult for analysis and decision making are the situations when the interacted objects have informal characteristics, that means that it is difficult to discover main factors influencing the objects and relations actual for those objects. Establishing the exact object behavior model is often not possible because of lack of knowledge about the objects themselves and about the environment in which they are functioning and interacting. Nevertheless, operations with such objects are often even more

important than operations with well-formalized objects. Some methods for operating with not fully-described objects are not enough productive.

2. Rules and cases

There exists a very important example of such an object – it is a human organ-ism. The application area for which human organism is of principle interest is medicine in general and medicinal diagnostics in particular. Programming DSS used for physician's decision support for diagnostics and cure choosing are often based on mathematical methods. It is easy to see two alternatives in dealing with the objects mentioned above: rule-based and case-based deduction.

In production rule approach, knowledge is represented with the set of several rules formulated in the "IF ... – THEN ..." form. These rules are used to de-duct the conclusion from input data. Such a method of reasoning is an example of a direct logical deduction.

Decision trees are often used in medicine. They represent a particular case of decision trees when the conclusion starts with checking some sign assigned to the tree root, and continues by moving along the tree to its leaves where the different decisions are located. Decision trees with single entry named tree root are often treated as a particular case of deduction rules.

Rule-based deduction makes it possible to incorporate knowledge into the system with the help of descriptive logic. Rules are following each other in a definite order, which is helpful for understanding them, but in fact does not create any order relation.

Rule-based models are much less structured and reflect the order in general action flow to smaller extent than many other approaches. They are really helpful in situations with relatively small set of limits applied to active actions, and where, consequently, a small amount of rules may define a very comprehensive scheme of interactions of component parts of integrated system.

Conditions in rule notation are in fact rule premises. They consist of one or several pairs "attribute – value" with logical "AND", "OR", "NOT" connectives. The conclusion denotes some fact or some instruction for definite action, which should be fulfilled according to the rule. Logical deduction mechanism is looking for those rules that include the facts entered, and then actualize the appropriate rules. The rule becomes active if the fact entered corresponds with the rule condition. In such a case, the active rule conclusion becomes a fact too. When all the activated rules respond, the final conclusion may be proved or rejected.

In practice, one can see a reverse logical deduction where the reasoning starts from the assumption made for possible final conclusion and moves towards the facts that may confirm the hypothesis.

Case-based reasoning (see [4-5]) is focused on knowledge about previous situations, or cases, stored in "case base". Decision made in conditions that are treated as similar to some situation met earlier, after being adapted is applied to the current

case. This method in conjunction with differential set (a set of potentially suitable decisions) is optimal for not-fully described case if there is a lack of time and resources. The case itself, if it was announced as similar, is the basis of the decision.

Decision making modeling human reasoning is practically used in many different areas of human activity. There exist a broad spectrum of possible applications, and control of poorly formalized objects is among them. We may treat the human organism as an object under control. It cannot be described with relatively simple mathematical model; thus, case-based reasoning methodology for physician's DSS may be considered as having very good prospects. Case-based reasoning is a method of looking for similar problem situations, which takes into account previous experience of problem solving. Instead of searching the solution from the very beginning, an attempt is done to use the solution found earlier in similar situation, and then adapting it to the changing situation of the current case. After this current case is processed, it (and its solution) is added to the case base, from where it can be used later.

Each case may consist of the situation description with the problem to be solved included, and the list of actions that were used to solve the appropriate problem. The solution may have the form of the previous case or of suggested typical example of problem solving method.

Accumulated collection of cases, which can be replenished with modeled or met in practice cases, forms the so called "case base". System built on this principles is in fact self-learning: the more cases are stored in the case base, the wider are the limits of their possible values, the higher is the probability to find "the most suitable" case, and therefore, the higher is the quality of the final solution.

3. Metrics and measures

Most part of existing approaches to building case-based reasoning DSS is focusing on only one aspect: choosing the most suitable cases. The basis of all the approaches to case selection is the method of estimation of similarity of previous cases and the current case. In these systems the metrics is defined in feature space. The point corresponding to the current case is then defined in this space, and, according to the chosen metrics, the closest case point for the current case is defined. Depending on case feature types, different metrics may be chosen: Chebyshev, Euclidean, Hamming, Mahalanobis, Manhattan, Akritean, Minkowski, Zhuravlyov (see [6-8]) distances and many others. Still there are situations when no metrics can be introduced. In these cases, measure of closeness is used instead of metrics. In its turn, measure of closeness may be defined by different ways, for example, in the form of case selection rule.

Structuring of a case set is also very useful. Different methods, in particular Data Mining approach, make it possible to clarify hidden knowledge about the application area. Classes of case equivalence may be established basing on various techniques: with the help of expert knowledge, or using the learning samples, or by clustering the case base. Breaking the case base into equivalence classes is the way

of speeding-up case searching process: cases belonging to some class are announced similar by the definition. Unfortunately, this measure of closeness is not absolutely adequate for estimating interrelations between current case and previous cases, especially when this current case falls into equivalence classes' intersection.

Image recognition problems often assume that object descriptions are founded on the set of features and signs, and this set is common for objects of all the classes. In other words, classes of equivalence and investigated objects may be located in unified feature space and have the dimensions. Industrial applications often break this condition. Not only real objects, but also classes' descriptions may have their own unique feature space. As an example, in medicine each disease (every disease may be treated as a separate class of equivalence) may be characterized with its own set of significant features. In addition, a case under investigation may have feature set that is absolutely different with those feature sets that were initially entered into the system. Relations between current case and classes of equivalence may be retrieved with the help of classes' projections on object feature set. Not fully described case may fall into the projection, to which it does not belong only because of lack of the feature that may prevent it from ambiguity.

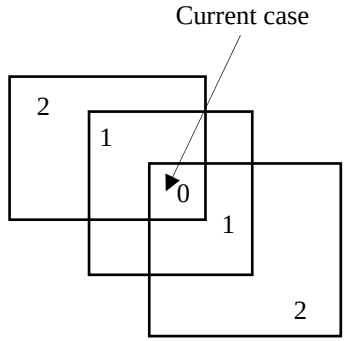

Fig. 1. Estimating the measure of closeness between the current case and previous cases

Estimating of the current case assumes the comparison of the corresponding point of feature space with spatial location of classes' projections ([9]). Analogues are the cases that belong to the class, to which current case belongs, these cases are considered as the most closest to that current case. If current case falls into classes' intersection, then the analogues belonging to the same intersection will be treated as close ones. Depending on the complexity of intersection, we may divide all analogues into separate groups (fig. 1). Analogues from the inner part of intersection are naturally considered as more close to the current case than analogues that are belonging only to one class. And, of course, analogues of the highest rank are located in the intersection of all the classes from the differential set.

Now the definition of offered measure of closeness can be finally formulated. *It is the distance between the current case and previous case evaluated as the difference of the number of classes that include the current case, and the number of classes that include the previous case.*

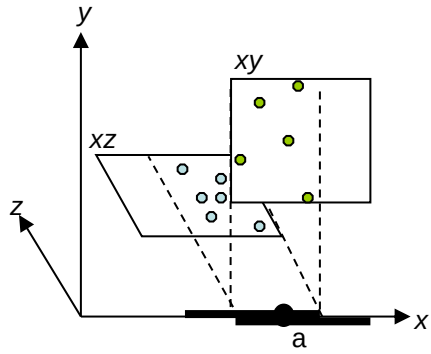

Fig 2. Using analogues for defining missing features of the current case

Initial case selection may not bring any tangible result. For example, the presence of a single "high temperature" feature will give a plenty of analogues. The physician should either agree that this poor set of features will not help to solve the problem or should expand feature set.

Analogues themselves contain the information about the features that should be retrieved. On fig. 2 one-dimensional current case $x=a$ falls into projection of classes *xy* and *xz*. To be compared with analogues from *xy* class, it lacks the *y* feature, and for *xz* class – it needs *z* feature to be added.

Thus, case selection cycle may be split into stages:

1. searching analogues for current case;
2. estimating the validity of the selected set (done "manually"). If "yes", the selection succeeds. If "no", execute the next stage;
3. building a ranked list of additional features in order to differentiate classes (these features may be found in previous cases);
4. making an attempt to retrieve additional features (done "manually"). Some features can never be exposed. If this is the case, the cycle stops with negative result.

4. Classes and features

The method described here was implemented in the programming DSS designed and developed in the Institute for System Programming of Russian academy of

sciences (ISP RAS) in cooperation with specialists from Moscow Regional Research and Clinical Institute ("MONIKI") (see [9-13]).

Physician is working in terms of "*disease*", "*treatment strategy*", "*patient*", "*sign*". The system provides access to the case base and automatically converts these terms into corresponding system notions "*class*", "*object*", "*feature*".

In some countries, there exists hands-on experience to carry out minimal but absolutely needed set of research actions before the diagnosis is stated. In our country physicians often set the diagnosis in much more comprehensive conditions, when not all the signs are known. The research actions are usually directed from very simple to more complex and expensive, starting from complaints and external examination and moving towards laboratory and instrumental research. From one point of view, if the diagnosis is rather clear from very first stages, there is no need to fulfill expensive analysis. From an-other side, the final solution may be delayed after making additional investigations, or wait till new symptoms appear. If feature set allows choice among several diseases, all of them are included into differential set.

Before starting the estimation process, physician needs not only to estimate the sufficiency of feature set, but also select those features that are linked with the differential set. In order not to overload the estimation process by extra features, the system automatically selects the needed signs, and if the set appeared is not full, will show still absent features. The final solution either to estimate partially filled set, or to continue with additional research, should be made by physician.

When there is a lack of time or resources, it may be impossible to expose all the absent features. Some new notions should be introduced, for instance, "*persistent feature combination*" and "*feature rank within its class according to the degree of its informativeness*". To define the priority of feature retrieving, additional selection criteria of new features searching should be defined according to frequency of feature used in the application, to object category, to feature availability.

Obviously, not all the features that are mentioned in the class description have the same informativeness. For example, in medicine there exist (pathognomonic) symptoms, that have absolute diagnostic value (markers of cancer, infarction, different types of hepatitis) and make it possible to determine the particular disease. Not only in medicine, but in much more general case, ideal features have the highest level of informativeness. They identify their classes unambiguously and can never be met in other classes. However, even in medicine ideal features sometimes can't be discovered while diagnosing the appropriate disease or its particular stage or form, or while a particular. If no ideal feature is found, some other features that are typical for the hypothesized disease should be investigated. There are features that appear in some classes with a much greater probability than that of their occurrence in other classes. These features are called controlling or causal features (bilirubin, hepatic enzymes in hepatitis). To get the final solution, we must not be focusing on one such feature, it should be considered in a combination with some other features. There also exist the third type of features – attendant features. They do not

characterize classes (in medicine, for example, some symptoms may accompany the disease: high temperature or erythrocyte sedimentation rate, and so on). The presence of these features may be treated as a necessary but not sufficient condition of belonging to the class. Their role in differentiating class is negligible. In summary, it may be stated that the features of the classes in the case base are ranked from the ideal to causal and then to attendant. Only ideal features may reliably identify the state of the object. If there is an ideal feature for some class, even if there are a number of attendant features, the fact of belonging of an object to a certain class can be stated only with some probability.

5. Advantages and drawbacks

In order to exactly define the current case class, it's always possible to operate with one single sign. Possible relations with other features should be taken into account. For example, a consistently observed combination of symptoms, defined in medicine as a syndrome, has a special diagnostic value. It is very close to Data Mining method known as association analysis. This method is very useful and often successful in processing of classes' descriptions in the case base. It helps to retrieve consistent feature combinations and ranks. Despite the fact that medicine is a precedent science, a lot of case examples in literature are built in terms of decision trees. Still, the lack of controlling (determining) feature may block the entry point into the production rule-based mechanism (for example, into the decision tree or into one of its nodes).

The DSS for transplantology designed and developed in ISP RAS and MONIKI is the concretization of general conceptual case-based complex objects control system. Usually, information support for physicians in similar DSS is limited to the "waiting list" card catalogues. Such kind of DSS are intended to provide initial selection of "donor-recipient" pair. All the problems of pair survival are beyond the competence of many practically used DSS. This point significantly differs them from the designed system. In fact, the main problem is in choosing of right tactics of patient support that should be followed after surgery transplantation operation. Many input parameters and contradictory factors should be taken into account. The role of DSS is dramatically increasing. Only automated systems may help the physician to solve conflicting problems. Human organism as an object under control can't be described with relatively simple mathematical model, that's why case-based reasoning methods for DSS may be considered as very promising.

6. Conclusion

Production rule-based model provides ease of perception and modification, simple mechanism of deduction, but has a number of disadvantages: dissimilarity with human mental structures of knowledge representation, uncertainty in rule interrelations and others.

The main advantage of case-based reasoning is simplicity and ease of implementation, while the drawback is in its inability to create models and rules that can generalize previous experience. One of the main problems of this method is the difficulties with correct selection of appropriate cases, which rests on the assessment of the similarity of the precedent and the current case. However, in certain circumstances, particularly when there is the need to work with not fully described objects, case-based reasoning method has notable advantage over other approaches.

Acknowledgements

Authors are really thankful to Russian Foundation for Basic Research that is supporting their work in 18-07-01211_a project.

References

[1]. Hillary Don. Decision making in critical care. University of California School of Medicine, San Francisco, California, B.C. Decker Inc., The C.V. Mosby company, 1995.
[2]. Chike. F. Oduoza. Decision support system based on effective knowledge management framework to process customer order enquiry. Decision Support Systems, Editor Chiang S. Jao, pp. 406, January 2010, INTECH, Croatia.
[3]. Prabhu Murugesan, Senthil Prabhu Natarajan, Lakshmi Karthigeyan. Clinical decision support systems. Computer Sciences Corporation, 2014.
[4]. Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications, 7(1):39-59, 1994.
[5]. Klaus-Dieter Althof, Eric Auriol, Ralph Barlette, and Michel Manago. A Review of Indus-trial Case-Based Reasoning Tools. AI Intelligence, 1995.
[6]. Greg Grudic and Jane Mulligan. Outdoor Path Labeling Using Polynomial Mahalanobis Distance, Robotics: Science and Systems II, 2006.
[7]. Elena Deza and Michel Marie Deza. Encyclopedia of Distances. Springer. 2009.
[8]. Zhuravlyov Yu. I. Selected scientific works, Magistr [Master], Moscow, 1998 (in Russian).
[9]. Yudin V.N. Measure of closeness in Case-based Reasoning System. Dokladyi 12-y Vserossiyskoy konferentsii Matematicheskie Metodyi Raspoznavaniya Obrazov (MMRO-12) [Proc. 12-th All-Russia Conf. on Mathematical Methods in Pattern Recognition], MAKS Press, Moscow, 2005, pp. 241-244 (in Russian).
[10]. Karpov L.E., Yudin V.N.. Integration of data Mining and Case-Based Reasoning methods in medical diagnostics and treatment choosing, Sbornik dokladov 13-j Vserossijskoj konferentsii Matematicheskie metody raspoznavaniya obrazov [Proc. of 13-th All-Russian conference Math. methods of pattern recognition], October 2007, MAKS Press, 2007, pp. 589-591, ISBN 978-5-317-02060-6, http://www.mmro.ru/files/mmro13.pdf (in Russian).
[11]. Yudin V.N., Karpov L.E., Vatazin A. V. Data mining and case-based reasoning methods in physician's decision making support software system, Almanah klinicheskoy meditsinyi [Almanac of Clinical Medicine], Moscow, 2008, v. 17, part 1, pp. 266-269, ISSN 2072-0505 (in Russian).

[12]. Yudin V.N., Karpov L.E., Vatazin A.V. Cure process as an adaptive control of human organism in software system “Physician’s Partner”. Almanah klinicheskoy meditsinyi [Almanac of Clinical Medicine], Moscow, 2008, v. 17, part 1, pp. 262-265, ISSN 2072-0505 (in Russian).

[13]. Valery Yudin, Leonid Karpov. The Case-Based Software System for Physician’s Decision Support. Lecture Notes in Computer Science Sublibrary: SL 3, Springer Verlag, Berlin, Heidelberg, 2010, pp. 78-85

Работа с неполностью описанными объектами в системах поддержки принятия решений: альтернативные подходы

[1] *В.Н. Юдин <yudin@ispras.ru>*

[1,2] *Л.Е. Карпов <mak@ispras.ru>*

[1] *Институт системного программирования им. В. П. Иванникова РАН*
Москва, 109004, ул. Солженицына, д. 25

[2] *Московский Государственный университет имени М. В. Ломоносова*
Москва, 119991, ГСП-1, Ленинские горы, д. 1

Аннотация. Неполностью описанные объекты могут встретиться в самых разных предметных областях и приложениях - от медицины до управления космическими кораблями. Начиная работу по проектированию и разработке системы поддержки принятия решений, которая должна работать с неполностью описанными объектами, необходимо предварительно выбрать один из альтернативных подходов. В основе одного из двух подходов, рассматриваемых в этой статье, находится логический вывод по заранее зафиксированным правилам. В таком подходе интенсивно используются продукции вида "ЕСЛИ-ТО". Другой подход, который часто называется прецедентным, предполагает наличие базы прецедентов, наполненной реальными и/или искусственными (модельными) случаями. Для этого второго подхода правила и модели объектов не являются необходимыми, но он значительно ближе к модели принятия решений, которая используется человеком в процессе мышления.

Ключевые слова: неполностью описанные объекты; поддержка принятия решений; логический вывод на основе порождающих правил; деревья решений; вывод на основе прецедентов; база прецедентов; метрика случаев; мера близости случаев; дифференциальный ряд случая; информативность признаков случая.

DOI: 10.15514/ISPRAS-2018-30(1)-9

Для цитирования: Юдин В.Н., Карпов Л.Е. Работа с неполностью описанными объектами в системах поддержки принятия решений: альтернативные подходы. Труды ИСП РАН, том 30, вып. 1, 2018 г., стр. 127-136. 10.15514/ISPRAS-2018-30(1)-9

Список литературы

[1] Hillary Don. Decision making in critical care. University of California School of Medicine San Francisco, California, B. C. Decker Inc., The C. V. Mosby company, 1995.

[2] Chike. F. Oduoza. Decision support system based on effective knowledge management framework to process customer order enquiry. Decision Support Systems, Editor Chiang S. Jao, ISBN 978-953-7619-64-0, pp. 406, January 2010, INTECH, Croatia.

[3] Prabhu Murugesan, Senthil Prabhu Natarajan, Lakshmi Karthigeyan. Clinical decision support systems. Computer Sciences Corporation, 2014.

[4] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodologi-cal variations, and system approaches. AI Communications, 7(1):39-59, 1994.

[5] Klaus-Dieter Althof, Eric Auriol, Ralph Barlette, and Michel Manago. A Review of Indus-trial Case-Based Reasoning Tools. AI Intelligence, 1995.

[6] Greg Grudic and Jane Mulligan. Outdoor Path Labeling Using Polynomial Mahalanobis Distance, Robotics: Science and Systems II, 2006.

[7] Elena Deza and Michel Marie Deza. Encyclopedia of Distances. Springer. 2009.

[8] Журавлев Ю. И. Избранные труды. М., Магистр, 1998.

[9] Юдин В. Н., Мера близости в системе вывода на основе прецедентов. Математические методы распознавания образов. Сборник докладов 12-й Всероссийской конференции, стр. 241-244, М., МАКС ПРЕСС, 2005

[10] Карпов Л. Е., Юдин В. Н., Интеграция методов добычи данных и вывода по прецедентам в медицинской диагностике и выборе лечения. Математические методы распознавания образов. Сборник докладов 13-й Всероссийской конференции, стр. 589-591

[11] Юдин В. Н., Карпов Л. Е., Ватазин А. В. Методы интеллектуального анализа данных и вывода по прецедентам в программной системе поддержки врачебных решений. Альманах клинической медицины, МОНИКИ, Москва, 2008 г., стр. 266-269

[12] Юдин В. Н., Карпов Л. Е., Ватазин А. В. Процесс лечения как адаптивное управление человеческим организмом в программной системе "Спутник врача". Альманах клинической медицины, т. XVII в двух частях, ч. 1. МОНИКИ, Москва, 2008 г., стр. 262-265

[13] Valery Yudin, Leonid Karpov. "The Case-Based Software System for Physician's Decision Support". Lecture Notes in Computer Science Sublibrary: SL 3, Springer Verlag, Berlin, Heidelberg, 2010, pp. 78-85