Liet. matem. rink, **45**, *spec. nr.*, 2005, 459–464

Making Virtual Learning Environment more intelligent: the problem of software agent's mental state recognition

Dalia BAZIUKAITĖ
e-mail: dalia@ik.ku.lt

Abstract. Intelligent decision making process, which is performed according to a learner in the Virtual Learning Environment (VLE), leads to the problem of solving several rather complex tasks. Two of them are most of interest. First, we need to train a software agent to recognize its mental state; and second, we want agent to apply optimal strategy to teach learners when it is in some mental state. The second issue we have discussed in [2]. We propose the agent's ability to recognise its mental state could be based on the classification result to a particular learner. Classification used to be done should be based on the discovered groups in the database data and classification rules that prescribe given learner to the one of the available clusters. Conceptual clustering seems to be the suitable technique able to provide a solution for the problem raised.

Keywords: agent's state, clustering, conceptual clustering, machine learning, virtual learning environment.

1. Introduction

In earlier works [1, 2] we define an intelligent Virtual Learning Environment as a learning environment, where the role of teacher is prescribed to the system itself. This means, it has the collection of tools that enable to simulate teacher's decisions made according to the learner; and the learner is equipped with tools that allow him/her to adopt the learning environment to their needs. In the intelligent VLE the role of teacher is prescribed to the agent, who makes decisions how to teach the learner depending on the experience presented to the agent. This idea needs to be supported by the tools able to bring into the life an opportunity to manage the behavior of VLE in an intelligent way. We reffer to the algorithms of Machine Learning (ML) in two cases. First, we need a tool for deciding what actions are better to perform for an agent, when it is beiing in some mental state. This issue is mainly discussed in our earlier work, which is presented in [2]. And second, we need a tool, able to discover groups in the available student data, collected in VLE. These groups build the base for the agent's mental state recognition.

Let we define an information system as a tetrad of S, Q, V and f like in [7]:

$$IS =< S, Q, V, f >, \tag{1.1}$$

where S is a finite set of objects $S = \{e_1, e_2, \ldots, e_m\}$; Q is a finite set of features $F = \{F_1, F_2, \ldots, F_n\}$; $V = \bigcup V_{Fj}$ is a set of feature values and V_{Fj} is a domain of feature $F_j \in Q$; $f\colon S \times Q \to V$ is an information function (IF) such that $f(e_i, F_j) \in V_{Fj}$ for every $F_j \in Q$, $e_i \in S$. Each set S is also called a learning set.

Suppose we are interested in features

$$F_1 = selftest_1, F_2 = selftest_2, \ldots, F_{n-1} = selftest_{n-1}, F_n = class. \tag{1.2}$$

The domains of features could be defined as $V_{Fj} = \{weak, medium, good, perfect\}$, for every feature $j = 1 \ldots (n-1)$ and

$$V_{F_n} = \{beginer, preintermidiate, intermediate, advanced\}, \tag{1.3}$$

for so called decision attribute. An example of information function would be

$$f(e_1, F_2) = good. \tag{1.4}$$

The goal is to discover the class conditions in terms of the example's features and their values. Coming from this, the class c_i is defined as

$$\big(example \in S | condition_i(example)\big) \to c_i. \tag{1.5}$$

This is the general hypothesis that we want an algorithm to generate as a solution of our problem.

There exist ML algorithms that generate one rule (or hypothesis) about the training data and there are such that are able to generate several. In general, the latter may perform better on new unseen data [6, 7, 8].

2. Clustering or conceptual clustering?

The data, we are going to examine, is results expressing the learning productivity of human learners. Actually, this data set expresses nothing else, but the knowledge they have in the curriculum, as is defined in [1]. The data in the data set can be numerical, nominal or either both. Moreover, coming from our idea of intelligent VLE, we need to find rules, that will allow the grouping of learners and make agent be able to apply them when the new data on a new (or current) learner arise. This generally expresses the problem of agent's mental state recognition. We suppose, that agent will be able to distinguish the mental state it currently is in, according to the result of the discriminative rules application on the data concirning the given learner.

Clustering is a prime example of unsupervised learning in ML [7]. We should note that were is a difference between clustering and conceptual clustering. In ML the notion of conceptual clustering is used to distinguish it from typical clustering. Clustering is best suited for handling numerical data only, where conceptual clustering can also deal with nominal data [6, 8]. It consists of two basic steps. The first is *clustering*, which finds clusters in a given data set, and the second is a *characterisation*, which generates a concept description for each cluster found by clustering.

We could think about several ways of solving the problem raised in this article. *Why not to apply the basic cluster analysis methods in combination with discriminative rules*, for instance the well known k-means clusterisation method for discovering groups in given data?

In case of *k-means* or other clustering method we can deal only with numerical data. But in data set, which actually expresses the results of student learning productivity,

the algorithm has to deal with nominal data, due to the information function defined in equation (1.4). Another problem is that at some starting point or even later were could be not enough clusters in data as we wish to define. The number of clusters has to vary dinamically and in case of k-means [8, 7] this number is predefined. Yet more, we face with the problem of incomplete data set. In our case we get it, when we treat the columns with quiz results that have been not taken yet by a student. Normally, in such column of data set stands the value "null". This has a meaning "the quiz has not been taken yet". In case of clustering such a value has to be treated as "zero", which is far away from his essential meaning. To avoid this we think about conceptual clustering algorithms, for instance ITERATE, which is developed by Biswas and Weinberg [3, 4].

This algorithm has four steps [3]: 1) order the data sequence based on a anchored dissimilarilty ordering (ADO) [4, 5] scheme, 2) generate a hierarchical concept tree using the partition score measure, 3) choose a representative set of concepts from the hierarchy to create an initial class partition, 4) consider objects one by one and based on category match measure redistribute objects to the most similar class. Repeat step no. 4 until no objects change class.

In the ADO ordering algorithm the object chosen to be the next in the order is the one that maximizes the sum of Manhattan distance between it and the previous n objects in the order. The Manhattan distance measured in perpendicular system of axes between two points $p_1(x_1, y_1)$ and $p_2(x_2, y_2)$ is defined by

$$MA = |x_1 - x_2| + |y_1 - y_2|. \tag{2.1}$$

The Manhattan distance between two objects defined by nominal-valued attributes is simply the number of diferences in the attribute-value pairs and differs from the distance measure used in k-means. The size, n, is user defined, and empirically corresponds to the actual number of classes expected in the data. Partition score is the utility of a partition structure made up of K classes and is defined by [5]

$$PS = \frac{\sum_{k=1}^{K} CU_k}{K}. \tag{2.2}$$

This means, that PS is average category utility over K classes. Category utility of a class C_k is defined as [5]

$$CU_k = P(C_k) = \sum_i \sum_j P(A_i = V_{ij}|C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2. \tag{2.3}$$

The $P(A_i = V_{ij})$ is the probability of feature A_i taking on value V_{ij} and $P(A_i = V_{ij}|C_k)$ is the conditional probability of $A_i = V_{ij}$ in class C_k. This represents an increase in the number of feature values that can be correctly guessed for class $C_k(P(A_i = V_{ij}|C_k))$, over the expected number of correct guesses, given that no class information is available $[P(A_i = V_{ij})^2]$.

Data objects are reassigned to optimize a chosen criterion function, usually the mean square error. With nominal-valued data, the match between an object d and a

class k is defined as a probabilistic similarity measure called the *category match measure* [4, 5]

$$CM_{dk} = P(C_k) \sum_{i,j \in \{A_i\}_d} \left(P(A_i = V_{ij} | C_k)^2 - P(A_i = V_{ij})^2 \right). \tag{2.4}$$

The class C_k is defined in terms of the conditional probability distribution of all feature values for the class. Also, the category match measure assumes that a data object has only one value per attribute (represented by $j \in \{A_i\}_d$ in equation (2.4)). Category match measures the increase in expected predictability of class C_k for the attribute values present in data object d.

The ITERATE can produce the rule model hierarchy by running the algorithm recursively on the obtained rules [4, 5, 3].

3. Results on data expressing student learning productivity

To illustrate the proposition we stated above, that "were could be not enough clusters in data as we wish to define", we use so called cluster silhouettes. Cluster silhouette displays a measure of how close each point in one cluster is to points in the neighboring clusters. This measure gets values from $+1$ and indicates points that are very distant from neighboring clusters, to -1, which indicates points that are probably assigned to the wrong cluster. Value 0 indicates points that are not distinctly in one cluster or another [9].

The Fig. 1 demonstrates possible situations in real student data that represent their learning productivity in different courses. This data set was collected during the semester and concerns with courses that were supported through the VLE of Klaipeda University. As is defined in [2], we propose four possible states for so called Teacher agent. Agent, in order to recognise itself beiing in some state, should classify the given learner to one of the appropriate clusters. But these four clusters, coming from the nature of student data, are not always persistent. For instance, part A of Fig. 1 shows that the structure of student data is such that only two clusters (y axis) would be the best choice to classify current data. Silhouettes values (x axis) are near to 1 in both clusters. The part B illustrates the ideal situation in given data. There are four possible clusters and all points in them have values equal to 1. The part D shows that depending on the current collection of student results it could sometimes be crutial mistake to classify data exactly into four clusters. Here, i.e., we see that half of points in the second cluster have negative values and are probably assigned to the wrong cluster.

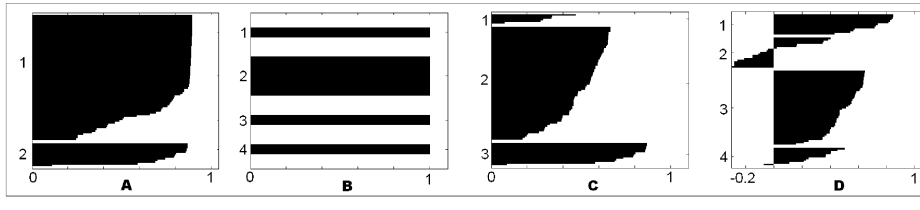

Fig. 1. The silhouettes values for student data expressing learning productivity.

Real data is stored in several data tables and the algorithm needs to get input in a form where each row is a i-th learner and every column is a j-th quiz in a k-th cource. For this, before running the conceptual clustering algorithm the appropriate virtual (in terms of databases) data table has to be formed. The pseudo-code expressing algorithmic steps in order to achieve results on agent sate recognition is given below.

Let we define the sets $C = \{c_1, c_2, \ldots, c_n\}$, $L = \{l_1, l_2, \ldots, l_m\}$ and $Q = \{q_{1c_1}, q_{2c_1}, \ldots, q_{rc_1}, q_{1c_2}, \ldots, q_{pc_2}, q_{1c_n}, \ldots, q_{kc_n}\}$, where the C stands for the set of available courses, L denotes the set of learners and the Q is the set of quizes available in each course c_i, where $i = 1 \ldots n$. According to the ideas in [1], every learner is free to select his/her own curriculum. In order to check is the selected curriculum best suited to the particular learner, system performs verification (gives a quiz).

Algorithm. Consider some $c_j \in C$.

INPUT: the row of learner's l_i results on quizes q_{sc_j}, where $s = 1 \ldots r$.

IF there exist $l_i \in L$ in a c_j,
such that l_i has finished $q_{sc_j} \in Q$ THEN
BEGIN
CHECK is the $count(l_i) > 1$
if TRUE, then
BEGIN
apply concept rules to l_i
ITERATE and GET RULES describing the concepts
END
else THINK the l_i is a beginner
END
ENDIF

OUTPUT: definition of learner l_i according to the concepts, and rules describing the concepts.

According to the algorithm above, every learner l_i is prescribed to some cluster depending on the result the learner achieves in some quiz q_{sc_j} (or set of quizes) that are available in a given course c_i. The prescribtion is done having in mind the concepts of clusters on data excluding this current [particular learner] result. The concepts that describe clusters considering newly arrived data are changed afterwards and should be applied in the next step.

4. Conclusions

The task of software agent's mental state recognition reduces to the solution of the conceptual clustering task and application of formed rules model to the newly arriving data. For such purpose, given data tables with results expressing the advances of students have to be transformed in to the form suitable for the input of the algorithm. The applicable algorithm has to be incremental, because there are some cases known, where desired number of cluster is unpersistent. The number of clusters vary and together with this number the rules, which prescribe new data subset to the appropriate cluster, changes.

References

1. D. Baziukaitė, Multiagent system engineering application modeling curriculum setup sub-system, in: R. Krištolaitis (Ed.), *9-oji magistrantų ir doktorantų konferencija*, Vytauto Didžiojo universitetas, Kaunas (2004), pp. 172–178.
2. D. Baziukaitė, Making virtual learning environment more intelligent: application of Markov decision process, *Liet. matem. rink.*, **44** (spec. nr.), 797–801 (2004).
3. G. Biswas, J. B. Weinberg, L. Cen, Iterate: A conceptual clustering method for knowlege discovery in database, in: B. Braunschweig and R. Day (Eds.), *Artifical Intelligence in the Petroleum Industry*, Editions Technip, Paris (1995), pp. 111–139.
4. G. Biswas, J.B. Weinberg, C. Li, *ITERATE*: *A Conceptual Clustering Method for Knowledge Discovery in Databases*, Dept. of Computer Science, Vanderbilt University, USA (1996).
5. G. Biswas, J.B. Weinberg, D.H. Fisher, ITERATE: a conceptual clustering algorithm for data mining, *IEEE Transactions on systems, man, and cybernetics*, **28**(2), 219–230 (1998).
6. K. Cios, D.K. Wedding, N. Liu, CLIP3: Cover learning using integer programming, *Kybernetes*, **26**(5), 513–536 (1997).
7. K. Cios, W. Pedrycz, R. Swiniarski, *Data Mining Methods for Knowledge Discovery*, Cluwer Academic Publishers (2000).
8. A.K. Pankaj, M.H. Nabil, *k*-means projective clustering, in: *PODS*, Paris, France (2004), pp. 155–165.
9. *MatLAB Language Reference* (2005).

REZIUMĖ

D. Baziukaitė. Intelektinės virtualios mokymo(si) aplinkos: programinio agento būsenos atpažinimo problema

Sprendimo priėmimo procesas, kuris atliekamas besimokančiojo atžvilgiu Virtualioje Mokymo(si) Aplinkoje (VMA), reikalauja kelių pakankamai sudėtingų uždavinių sprendimo. Mus labiausiai domina du. Pirmasis – siekama apmokyti programinį agentą atpažinti būseną, kurioje jis yra duotoju momentu. Antrasis – trokštama, kad agentas priimdamas sprendimus vadovautųsi optimalia strategija, kuri priimtina žinant, kad agentas yra tam tikroje būsenoje. Antroji problema jau kartą buvo aptarta [2]. Pasiūlytas agento būsenos atpažinimo uždavinio sprendimas remiasi klasifikavimo rezultatu konkretaus besimokančiojo atžvilgiu. Klasifikavimas atliekamas duomenyse rastų grupių atžvilgiu taikant klasifikavimo taisykles, kurios priskiria duotąjį besimokantįjį į kurį nors vieną galimą klasterį. Straipsnyje aptartas konceptualaus klasterizavimo metodas, kuris pateikia tinkamas priemones iškeltai problemai spręsti.