A Similarity Evaluation Technique for Cooperative Problem Solving with a Group of Agents

Seppo Puuronen[1] and Vagan Terziyan[2]

[1] University of Jyvaskyla, P.O.Box 35, FIN-40351 Jyvaskyla, Finland
sepi@jytko.jyu.fi
[2] Kharkov State Technical University of Radioelectronics,
14 Lenin Avenue, 310166 Kharkov, Ukraine
vagan@kture.cit-ua.net

Abstract. Evaluations of distances or similarity measurements are very important in cooperative problem solving with multiple agents. Distance between problems is used by agents to recognize nearest solved problems for any new problem, distance between solutions is necessary to compare and evaluate solutions made by different agents, distance between agents is useful to evaluate weights of all agents to be able to integrate them by weighted voting. The goal of this paper is to develop similarity evaluation technique to be used for cooperative problem solving based on opinions of several agents. Virtual training environment used for this goal is represented by predicates that define relationships within three sets: problems, solutions, and agents. Every agent selects a solution for each problem by giving its vote: yes, no, and no-op. We derive internal relations and appropriate similarity values between any pair of subsets of the same type taken from the three sets: problems, solutions, and agents, and also external similarity relations for any two subsets of different types.

1 Introduction

Cooperative problem solving is the process of finding previously unknown and potentially interesting solutions in large collaborative virtual environments [2]. Numerous cooperative problem solving methods have recently been developed based on intelligent information agents’ framework. In many cases it is necessary to solve the problem of evaluation and selection of the most appropriate agent or a group of the most appropriate agents for every new problem. Often the agent selection is done statically without analyzing each particular problem. If the method selection is done dynamically taking into account characteristics of each problem, then cooperative problem solving usually gives better results.

We use an assumption that each agent has its competence area inside the virtual environment. The problem is then to try to estimate these competence areas in a way that helps to select the best agent for every problem. From this point of view cooperative problem solving with a set of available methods has much in common with the multiple expertise problem or with the problem of combining multiple classifiers in machine learning, data mining and knowledge discovery [4,5]. All three groups of problems solve the task of taking submissions from several sources and selecting the best one separately for every new case. In [15] we have suggested a voting-type technique and recursive statistical analysis to handle knowledge obtained from multiple medical experts. In [16] we presented a meta-statistical tool to manage different statistical techniques used in knowledge discovery.

Evaluations of distances or similarity measurements are very important in cooperative problem solving with group of agents. Distance between problems is used to recognize solved problems, which are nearest neighbors for some new problem. Such distance, for example, is widely used to integrate multiple classifiers [13]. Distance between solutions is necessary when an agent learns based on virtual training environment to evaluate its individual solution relatively to previously made cooperative solutions (the same Cross-Validation Majority approach [7,9] is used to learn multiple classifiers). This distance helps to find out the areas of competence for agents (like in [8] to find out competence areas for every individual classifier from an ensemble). Distance between agents is useful for example to evaluate weights of every agent to be able to integrate solution results by weighted voting (this is also has analogy with the same classifiers' integration technique [1]).

There are many approaches to define distance between any two problems based on their numerical or semantic closeness. For example the semantic closeness between terms is a measure of how closely terms are related in the solution schema [14]. Distance metric used by Rada et al. [11] represents the conceptual distance between concepts. Rada et al. uses only the path length to determine this conceptual distance, with no consideration of node or link characteristics. Distance is measured as the length of the path representing the traversal from the first term to the second. Rocha [12] has suggested a method to “fuzzify” conversation theory, by calculating continuously varying conceptual distances between nodes in an entailment mesh, on the basis of the number of linked nodes they share. In order to measure the distance between two concepts in a mind, Jorgensen measures a distance between two concepts, which he calls *psy* [6]. It has been suggested to assign an arbitrary distance of *n* units to the separation between two concepts such as "Concept *A*" and "Concept *B*" and then ask a subject to tell us how far other concepts (*C* and *D*) are from each other in these units. Problem-based learning techniques typically handle continuous and linear input values well, but often do not handle nominal input attributes appropriately. To compute the similarity between two solutions also a probabilistic metric of the PEBLS algorithm [3] can be applied. The distance d_i between two solutions v_1 and v_2 for certain problem is:

$$d(v_1, v_2) = \sum_{i=1}^{k} \left(\frac{C_{1i}}{C_1} - \frac{C_{2i}}{C_2} \right)^2 ,$$

where C_1 and C_2 are the numbers of problems in the virtual training environment with selected solutions v_1 and v_2, C_{1i} and C_{2i} are the numbers of problems from the *i*-th group of problems, where the solutions v_1 and v_2 were selected, and *k* is the number of groups of problems.

For example, let us assume that there are three groups of problems collected in a virtual training environment: “Search of scientific information”, “Search of commercial information”, and “Search of traveling information”. All groups have the same attribute: “Selection of a search machine”. Two possible solutions among others for this problem attribute are *Yahoo* and *Infoseek*. Let us assume that *Yahoo* was selected 2 times to search scientific information, 8 times to search commercial information, and 4 times to search traveling information. For the same aims *Infoseek* was selected 6, 9, and 1 times respectively. Thus we have the following values in our example:

$$v_1 =< Yahoo >,\ v_2 =< Infoseek >, \text{ and}$$

$$k=3,\quad C_1=14,\ C_2=16,\ C_{11}=2\ ,\ C_{12}=8,\ C_{13}=4,\ C_{21}=6\ ,\ C_{22}=9,\ C_{33}=3.$$

The distance between the two solutions *Yahoo* and *Infoseek* in the example is calculated as follows:

$$d(v_1, v_2) = (\frac{2}{14} - \frac{6}{16})^2 + (\frac{8}{14} - \frac{9}{16})^2 + (\frac{4}{14} - \frac{1}{16})^2 \approx 0{,}1038 \cdot$$

The value difference metric was designed by Wilson and Martinez [17] to find reasonable distance values between nominal attribute values, but it largely ignores continuous attributes, requiring discretization to map continuous values into nominal values. As it was mentioned in the Wilson and Martinez review [17] there are many learning systems that depend upon a good distance function to be successful. A variety of distance functions are available for such uses, including the Minkowsky, Mahalanobis, Camberra, Chebychev, Quadratic, Correlation, and Chi-square distance metrics; the Context-Similarity measure; the Contrast Model; hyperrectangle distance functions and others [17].

The present paper deals with the cooperative problem solving with virtual training environment and a group of agents. The agents make a virtual training environment that is then used for learning purposes. For virtual training environment representation, we use predicates that connect problems, solutions and agents. Each agent defines its selection concerning each problem-solution pair by a voting-type system supporting or resisting the use of a solution to solve the problem. The agent also has the option to refuse to vote in favor of either possibility.

We apply our formalisms developed in [10] to define general framework of similarity evaluation between problems, solutions and agents to be used in the cooperative problem solving with group of agents. In chapter 2 we present the basic notation used throughout the paper and the problems to be discussed. The next chapter deals with finding the most supported relations among the agents. In chapter 4 we discuss the discovery of similarity relations between problems, solutions, and agents. We end with short conclusions in the last chapter.

2 Notation and Problems

In this chapter we present the basic notation used throughout the paper and describe briefly the main problems discussed in the paper.

Virtual training environment of a group of agents is represented by a quadruple:

$$< D,C,S,P >,$$

where D is the set of the problems D_1, D_2,..., D_n in the virtual training environment; C is the set of the solutions C_1, C_2,..., C_m, that are used to solve the problems; S is the set of the agents S_1, S_2,..., S_r, which select solutions to solve the problem; and P is the set of semantic predicates that define relationships between D, C, S as follows:

$$P(D_i,C_j,S_k) = \begin{cases} 1, \mathit{if\ the\ agent}\ S_k\ \mathit{selects\ solution}\ C_j \\ \quad \mathit{to\ solve\ the\ \ problem}\ D_i; \\ -1, \mathit{if}\ S_k\ \mathit{refuses\ to\ select}\ C_j \\ \quad \mathit{to\ solve}\ D_i; \\ 0, \mathit{if}\ S_k\ \mathit{does\ not\ select\ or\ refuse} \\ \quad \mathit{to\ select}\ C_j\ \mathit{to\ solve}\ D_i. \end{cases}$$

We will consider two groups of problems that deal with processing virtual training environment based on a group of agents.

1. *Binary relations between the elements of (sub)sets of C and D; of S and C; and of S and C.* The value of the relation between each pair (C_j,D_i) of elements shows the support among all the agents for selection (or refusal to select) of the solution C_j to solve the problem D_i. This is called the total support. The value of the relation between each pair (S_k,D_i) of elements shows the total support which the agent S_k receives selecting (or refusing to select) all the solutions from C to solve the problem D_i. The value of the relation between each pair (S_k,C_j) of elements shows the total support, which the agent S_k receives selecting (or refusing to select) the solution C_j to solve all the problems from D. We will refer to this first group of deriving relations as *deriving external similarity values*.

2. *Binary relations between two subsets of D; two subsets of C; and two subsets of S.* The value of the relation between each pair (D_s,D_t) of the two problems from D shows the support for the neighborhood (“similarity”) of these problems via the solutions, via the agents, or via both of these. The value of the relation between each pair (C_s,C_t) of the two solutions from C shows the support for the nearness (“similarity”) of these solutions via the problems, via the agents, or via both of these. The value of the relation between each pair (S_s,S_t) of the two agents from S shows the support for the likeness (“similarity”) of these agents via the problems, via the solutions, or via both of these. We will refer to this second group of deriving relations as *deriving internal similarity values*.

3 Deriving External Similarity Values

In this chapter, we define how the total support for binary relations is formed. We consider relations between any pair of subsets taken from different sets D, C, or S. We introduce how the values describing this total support are standardized to the closed interval [0,1]. We study an evaluation of the quality of the agents. Then we describe the threshold value for the relations and we conclude with an example.

3.1 Total Support of Binary Relations

The total support of the binary relations *DC*, *CD*, *DS*, *SD*, *CS* and *SC* is formed using the following formulas:

$$DC_{i,j} = CD_{j,i} = \sum_{k}^{r} P(D_i, C_j, S_k), \forall D_i \in D, \forall C_j \in C;$$

$$SC_{k,j} = CS_{j,k} = \sum_{i}^{n} DC_{i,j} \cdot P(D_i, C_j, S_k), \forall S_k \in S, \forall C_j \in C;$$

$$SD_{k,i} = DS_{i,k} = \sum_{j}^{m} DC_{i,j} \cdot P(D_i, C_j, S_k), \forall S_k \in S, \forall D_i \in D.$$

The definition of the value of the relation between each pair (C_j,D_i) of the elements of the sets *C* and *D* sums up the total support among all the agents for selection (or refusal to select) of the solution C_j to solve the problem D_i. If, for example, three agents select the solution C_j to solve the problem D_i, then $DC_{i,j}=3$.

The definitions of the value for each pair (S_k,D_i) of the elements of the sets *S* and *D* and for each pair (S_k,C_j) of the elements of the sets *S* and *C* use the total support calculated above. The value of the relation (S_k,C_j) represents the total support that the agent S_k obtains selecting (or rejecting) the solution C_j to solve all the problems.

The value of the relation (S_k,D_i) represents the total support that the agent S_k receives selecting (or refusing to select) all the solutions to solve the problem D_i.

3.2 Standardizing Total Support of Binary Relations

The goal of standardizing external relations is to make the appropriate similarity values to be within the closed interval [0,1]. For this we use simple basic scheme:

$$\text{standardizing value} = [\text{value}] = \frac{\text{value - min(value)}}{\text{max(value) - min(value)}}.$$

From the definitions presented in 3.1 it follows that the minimum and maximum values of total support in each relation are:

$$\max_{i,j} DC_{i,j} = \max_{j,i} CD_{j,i} = r; \quad \min_{i,j} DC_{i,j} = \min_{j,i} CD_{j,i} = -r; \quad \max_{k,j} SC_{k,j} = \max_{j,k} CS_{j,k} = n \cdot r;$$

$$\min_{k,j} SC_{k,j} = \min_{j,k} CS_{j,k} = n \cdot (2 - r); \quad \max_{k,i} SD_{k,i} = \max_{i,k} DS_{i,k} = m \cdot r;$$

$$\min_{k,i} SD_{k,i} = \min_{i,k} DS_{i,k} = m \cdot (2 - r).$$

The transformation to the standardized values can be made using the following formulas (notice that we use brackets around the name of the standardized support array to distinguish it from the array with the basic support values):

$$[DC]_{i,j} = [CD]_{j,i} = \frac{DC_{i,j} + r}{2 \cdot r}; \qquad [SC]_{k,j} = [CS]_{j,k} = \frac{SC_{k,j} + n \cdot (r-2)}{2 \cdot n \cdot (r-1)};$$

$$[SD]_{k,i} = [DS]_{i,k} = \frac{SD_{k,i} + m \cdot (r-2)}{2 \cdot m \cdot (r-1)}.$$

3.3 Quality Evaluation

The quality of each agent from the support point of view is calculated using the standardized total support values derived in 3.2. For each agent S_k, we define a quality value $Q^D(S_k)$ that measures the abilities of the agent in the area of problems, and a quality value $Q^C(S_k)$ that measures the abilities of the agent in the area of solutions:

$$Q^D(S_k) = \frac{1}{n}\cdot\sum_{i}^{n}[SD]_{k,i}\,;\; Q^C(S_k) = \frac{1}{m}\cdot\sum_{j}^{m}[SC]_{k,j}\,.$$

Theorem: $Q^D(S_k) \equiv Q^C(S_k)$.

Proof:

$$Q^D(S_k) = \frac{1}{n}\cdot\sum_{i}^{n}[SD]_{k,i} = \frac{1}{n}\cdot\sum_{i}^{n}\frac{SD_{k,i}+m\cdot(r-2)}{2\cdot m\cdot(r-1)} = \frac{1}{n}\cdot\sum_{i}^{n}\frac{\sum_{j}^{m}(DC_{i,j}\cdot P(D_i,C_j,S_k))+m\cdot(r-2)}{2\cdot m\cdot(r-1)} =$$

$$= \frac{1}{2\cdot n\cdot m\cdot(r-1)}\cdot\Big(\sum_{i}^{n}\sum_{j}^{m}(DC_{i,j}\cdot P(D_i,C_j,S_k))+\sum_{i}^{n}m\cdot(r-2)\Big) =$$

$$= \frac{1}{2\cdot n\cdot m\cdot(r-1)}\cdot\Big(\sum_{i}^{n}\sum_{j}^{m}(DC_{i,j}\cdot P(D_i,C_j,S_k))+n\cdot m\cdot(r-2)\Big) =$$

$$= \frac{1}{2\cdot m\cdot n\cdot(r-1)}\cdot\Big(\sum_{j}^{m}\sum_{i}^{n}(DC_{i,j}\cdot P(D_i,C_j,S_k))+m\cdot n\cdot(r-2)\Big) =$$

$$= \frac{1}{2\cdot m\cdot n\cdot(r-1)}\cdot\Big(\sum_{j}^{m}\sum_{i}^{n}(DC_{i,j}\cdot P(D_i,C_j,S_k))+\sum_{j}^{m}n\cdot(r-2)\Big) =$$

$$= \frac{1}{m}\cdot\sum_{j}^{m}\frac{\sum_{i}^{n}(DC_{i,j}\cdot P(D_i,C_j,S_k))+n\cdot(r-2)}{2\cdot n\cdot(r-1)} = \frac{1}{m}\cdot\sum_{j}^{m}\frac{SC_{k,j}+n\cdot(r-2)}{2\cdot n\cdot(r-1)} = \frac{1}{m}\cdot\sum_{j}^{m}[SC]_{k,j} = Q^C(S_k)\cdot$$

This theorem shows that the evaluation of an agent competence (ranking, weighting) does not depend on the competence area “virtual world of problems” or “conceptual world of solutions” because both competence values are always equal.

3.4 Selecting Relations Using Threshold Value

There are situations where it is reasonable to pick out the most supported relations as a cooperative result of the agents. We use a threshold value as a for calculating the cutting points used to select the appropriate relations. These cutting values are applied to the standardized support arrays. First we select the threshold value *T* that belongs to the closed interval [0,1] and then we calculate the cutting points and apply them to the standardized values of relations as follows:

$$\forall T(T\in[0,1])\forall[A]_{s,t}([A]_{s,t}\in[A]\in\{[CD];[DC];\ldots;[DS]\})\exists$$

$$\exists \tilde{T}((\tilde{T} = \frac{T\cdot\sigma^2_{[A]}+\mu_{[A]}}{\sigma^2_{[A]}+1})\& \begin{cases} ([A]_{s,t} \ge \tilde{T})) & \Rightarrow [A]^T_{s,t} = 1; \\ ([A]_{s,t} \le 2\cdot\mu_{[A]} - \tilde{T})) & \Rightarrow [A]^T_{s,t} = -1; \\ (2\cdot\mu_{[A]} - \tilde{T} < [A]_{s,t} < \tilde{T})) & \Rightarrow [A]^T_{s,t} = 0, \end{cases}$$

where: $\mu_{[A]}$ is the average and $\sigma^2_{[A]}$ is the standard deviation of the values of $[A]$;

We will use the operator $[A]^{\mathrm{T}}$ for selection of the relations according to the threshold value T. This operator takes into account the distribution of the values.

3.5 Example

Let us suppose that four agents have to solve three problems related to the search of information in WWW using keywords and search machines available. The agents should define their selection of appropriate search machine for every search problem. The final problem is to obtain a cooperative solution of all the agents.

We assume in this example that the solutions are search machines to be selected (they are listed in Table 1), the agents with hypothetical names are presented in Table 2, the problems to be searched are defined by appropriate keywords (Fig. 1).

Table 1. *C*-set in search example

Solutions - search machines	Notation
AltaVista	C_1
Excite	C_2
Infoseek	C_3
Lycos	C_4
Yahoo	C_5

Table 2. *S*-set in search example

Agents	Notation
Fox	S_1
Wolf	S_2
Cat	S_3
Hare	S_4

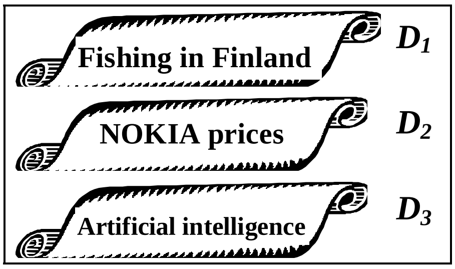

Fig. 1. *D*-set in search example: search problems with keywords

The predicates of the solution results of the agents are given in the summary table contained in Table 3.

Table 3. Selections made by agents in the example

D_1						D_2					D_3				
P	C_1	C_2	C_3	C_4	C_5	C_1	C_2	C_3	C_4	C_5	C_1	C_2	C_3	C_4	C_5
S_1	1	-1	-1	0	-1	-1	0	-1	0	1	1	0	1	-1	0
S_2	0	-1	0	1	-1	1	-1	-1	0	0	0	1	0	-1	1
S_3	0	0	-1	1	0	1	-1	0	1	1	-1	-1	1	-1	1
S_4	1	-1	0	0	1	-1	0	0	1	0	-1	-1	1	-1	1

Please notice that the values used in this example are hypothetical ones without any connection to real agents or search machines. For example, the agent *Wolf* prefers to select *Lycos* to find information about “Fishing in Finland” and it refuses to select *Excite* or *Yahoo* for the same problem. Also for this problem, *Wolf* does not use or refuse to use the *AltaVista* or *Infoseek*.

We obtain the total support values of the relations presented in Table 4a-f.

Table 4. Total support values of binary relations

a)

SC	C_1	C_2	C_3	C_4	C_5
S_1	1	3	7	4	3
S_2	0	4	2	6	4
S_3	1	3	5	8	5
S_4	3	4	3	6	2

b)

SD	D_1	D_2	D_3
S_1	8	4	6
S_2	6	4	6
S_3	4	6	12
S_4	4	2	12

c)

CS	S_1	S_2	S_3	S_4
C_1	1	0	1	3
C_2	3	4	3	4
C_3	7	2	5	3
C_4	4	6	8	6
C_5	3	4	5	2

d)

CD	D_1	D_2	D_3
C_1	2	0	-1
C_2	-3	-2	-1
C_3	-2	-2	3
C_4	2	2	-4
C_5	-1	2	3

e)

DC	C_1	C_2	C_3	C_4	C_5
D_1	2	-3	-2	2	-1
D_2	0	-2	-2	2	2
D_3	-1	-1	3	-4	3

f)

DS	S_1	S_2	S_3	S_4
D_1	8	6	4	4
D_2	4	4	6	2
D_3	6	6	12	12

The standardized support values of the relations are presented in Table 5a-f.

Table 5. Standardized support values of the relations in the example

a)

[SC]	C_1	C_2	C_3	C_4	C_5
S_1	0.39	0.5	0.72	0.56	0.5
S_2	0.33	0.56	0.44	0.67	0.56
S_3	0.39	0.5	0.61	0.78	0.61
S_4	0.5	0.56	0.5	0.67	0.44

b)

[SD]	D_1	D_2	D_3
S_1	0.6	0.47	0.53
S_2	0.53	0.47	0.53
S_3	0.47	0.53	0.73
S_4	0.47	0.4	0.73

c)

[CS]	S_1	S_2	S_3	S_4
C_1	0.39	0.33	0.39	0.5
C_2	0.5	0.56	0.5	0.56
C_3	0.72	0.44	0.61	0.5
C_4	0.56	0.67	0.78	0.67
C_5	0.5	0.56	0.61	0.44

d)

[CD]	D_1	D_2	D_3
C_1	0.75	0.5	0.375
C_2	0.125	0.25	0.375
C_3	0.25	0.25	0.875
C_4	0.75	0.75	0
C_5	0.375	0.75	0.875

e)

[DC]	C_1	C_2	C_3	C_4	C_5
D_1	0.75	0.125	0.25	0.75	0.375
D_2	0.5	0.25	0.25	0.75	0.75
D_3	0.375	0.375	0.875	0	0.875

f)

[DS]	S_1	S_2	S_3	S_4
D_1	0.6	0.53	0.47	0.47
D_2	0.47	0.47	0.53	0.4
D_3	0.53	0.53	0.73	0.73

For example, agents *Cat* (S_3) and *Hare* (S_4) have the highest standardized support value 0.73 among their colleagues concerning “Artificial Intelligence” search problem (D_3) as can be seen from the array [*SD*].

When we use threshold value T=0.75 to select the relations then one of them $[DC]^{0.75}$ gives us the cooperative solution as shown in Fig. 2.

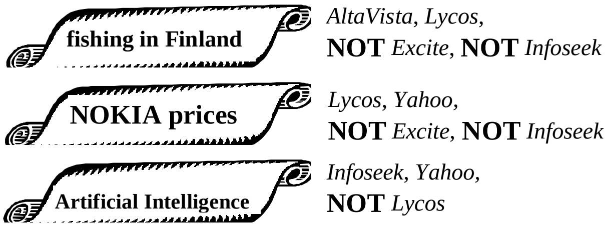

Fig. 2. Cooperative solution of the search machines’ selection

The arrays $[SC]^{0.75}$ and $[SD]^{0.75}$ describe the agents “competence” in the subject area and in the search machines’ area from the support point of view. For example, the selection proposals obtained from the agent *Fox* (S_1) should be accepted if they concern search machines *Infoseek* (C_3) and *Lycos* (C_4) or search problems related to

“Fishing in Finland” (D_1) and “Artificial Intelligence” (D_3), and these proposals should be rejected if they concern *AltaVista* (C_1) or “NOKIA Prices” (D_2). In some cases it seems to be possible to accept selection proposals from the agent *Fox* if they concern *Excite* (C_2) and *Yahoo* (C_5). All four agents are expected to give an acceptable selection concerning “Artificial Intelligence” related search and only suggestion of the agent *Cat* (S_3) can be accepted if it concerns “NOKIA Prices” search.

4 Deriving Internal Similarity Values

In this chapter we define internal similarity values between any two subsets taken from one set: agents, solutions, or problems.

An internal similarity value is based on an internal relation between any two subsets taken from the same set *D*, *C*, or *S*. An internal relation is derived using as an intermediate set one of the other sets (Fig. 3a) or both of the other sets (Fig. 3b).

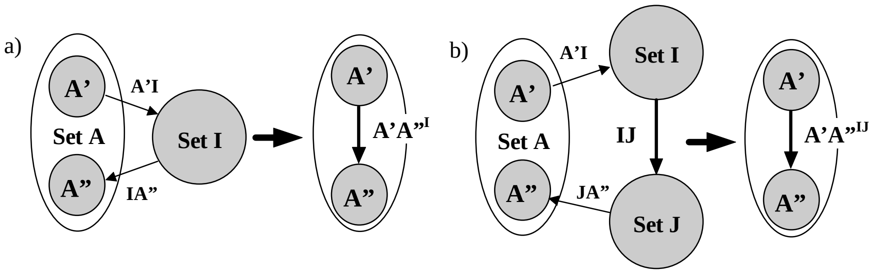

Fig. 3. Internal similarity values via one or two intermediate sets

We refer to the relation, which has one intermediate set *I*, as *I*-based relation, and we refer to the relation with two intermediate sets *I* and *J* as *IJ*-based relation.

In the following definitions we need parts of the relations *SD*, *DS*, *SC*, *CS*, *DC*, and *CD*. These parts are formed from the original relations by taking the appropriate subsets of values. For example, when we have two subsets *S’* and *S’’* of the set S, then *S’D* and *DS’’* are:

$$\forall S'_{k'} \in S, \forall D_i \in D \exists S_k ((S_k \in S) \& (S_k = S'_{k'}) \& (S'D_{k',i} = SD_{k,i}));$$

$$\forall S''_{k''} \in S, \forall D_i \in D \exists S_k ((S_k \in S) \& (S_k = S''_{k''}) \& (DS''_{i,k''} = DS_{i,k})).$$

Internal similarity values between any two subsets of the set *S* are derived using as an intermediate set the set of problems, the set of solutions, or both of these.

D-based similarity. Let *S’* and *S’’* be subsets of the set *S*. We define the *D*-based internal similarity value between *S’* and *S’’* as a value of the relation $S'S''^{D}$ obtained by the following rule ($\times$ means the multiplication of the appropriate matrixes):

$$\forall S' \subset S, \forall S'' \subset S \Rightarrow S'S''^{D} = S'D \times DS'',$$

To obtain standardized values we assume that $r \geq 2$ (there exist at least two agents). Then we calculate *min* and *max* values, which are:

$$\min_{k',k''} S'S''^{D}_{k',k''} = \min_{k'',k'} S''S'^{D}_{k'',k'} = \sum_{i}^{n} (\min_{k,i} SD_{k,i} \cdot \max_{i,k} DS_{i,k}) = n \cdot m^2 \cdot r \cdot (2-r);$$

$$\max_{k',k''} S'S''^{D}_{k',k''} = \max_{k'',k'} S''S'^{D}_{k'',k'} = \sum_{i}^{n} (\max_{k,i} SD_{k,i} \cdot \max_{i,k} DS_{i,k}) = n \cdot m^2 \cdot r^2.$$

The standardized values of D-based internal similarity between agents are:

$$\left[S'S''\right]^{D}_{k',k''} = \left[S''S'\right]^{D}_{k'',k'} = \frac{S'S''^{D}_{k',k''} + n \cdot m^2 \cdot r \cdot (r-2)}{2 \cdot n \cdot m^2 \cdot r \cdot (r-1)}, \forall S', S'' \subset S.$$

C-based similarity. We define the C-based internal similarity between any two subsets S' and S'' of the set S as value of the relation $S'S''^{C}$ obtained as follows:

$$\forall S' \subset S, \forall S'' \subset S \Rightarrow S'S''^{C} = S'C \times CS''.$$

DC- based similarity and *CD- based similarity*. We define these relations as:

$$\forall S' \subset S, \forall S'' \subset S \Rightarrow S'S''^{DC} = S'D \times DC \times CS'' \text{ and}$$
$$\forall S' \subset S, \forall S'' \subset S \Rightarrow S'S''^{CD} = S'C \times CD \times DS''.$$

Internal similarity values between any two subsets of solutions are defined in a similar way as those between agents. We represent here only formulas for calculating main values of the appropriate relations.

S-based, D-based, DS- and SD-based similarity relation. We define these relations between any two subsets C' and C'' of the set C as follows:

$$\forall C' \subset C, \forall C'' \subset C \Rightarrow C'C''^{S} = C'S \times SC''; \ \forall C' \subset C, \forall C'' \subset C \Rightarrow C'C''^{D} = C'D \times DC'';$$
$$\forall C', C'' \subset C \Rightarrow C'C''^{DS} = C'C''^{SD} = C'D \times DS \times SC'' = C'S \times SD \times DC''.$$

We define the internal similarity between problems in a similar way as the previous ones between agents and solutions.

S-based, C-based, CS- and SC-based similarity relation. We define these relations between any two subsets D' and D'' of the set D as follows:

$$\forall D' \subset D, \forall D'' \subset D \Rightarrow D'D''^{S} = D'S \times SD''; \ \forall D' \subset D, \forall D'' \subset D \Rightarrow D'D''^{C} = D'C \times CD'';$$
$$\forall D', D'' \subset D \Rightarrow D'D''^{CS} = D'D''^{SC} = D'C \times CS \times SD'' = D'S \times SC \times CD''.$$

5 Conclusion

The goal of this paper was to develop formal similarity evaluation framework to be used in cooperative problem solving with group of agents. We represent a virtual training environment by predicates defining relations between the elements of three sets: problems, solutions, and agents. This representation is directly applicable, for example, when training environment is collected using a three-value voting technique. Each agent submits his opinion of the use of each solution to solve each problem. Each agent can accept the use of the solution to solve the problem, refuse such use or be indifferent. The results of voting are collected into a basic predicate array. Discussion was given to methods of deriving the total support of each binary similarity relation. This represents the most supported solution result that can, for example, be used to evaluate the agents. We also discussed relations between elements taken from the same set: problems, solutions, or agents. This is used, for example, to divide agents into groups of similar competence relatively to the problem environment.

References

1. Bauer, E., Kohavi, R.: An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. Machine Learning, Vol. 33 (1998).
2. Benford, S., Bowers, J., Fahlén, L., Greenhalgh, C., Snowdon, D.: User Embodiment in Collaborative Virtual Environments. In: Proceedings of CHI95, ACM Press (1995).
3. Cost, S., Salzberg, S.: A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features. Machine Learning, Vol. 10, No. 1 (1993) 57-78.
4. Dietterich, T.G.: Machine Learning Research: Four Current Directions. AI Magazine, Vol. 18, No. 4 (1997) 97-136.
5. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R.: Advances in Knowledge Discovery and Data Mining. AAAI/ MIT Press (1997).
6. Jorgensen P.: P Jorgensen's COM515 Project Page (1996) Available in WWW: http:// jorg2.cit.bufallo.edu/COM515/ project.html.
7. Kohavi, R.: A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In: Proceedings of IJCAI'95 (1995).
8. Koppel, M., Engelson, S.P.: Integrating Multiple Classifiers by Finding their Areas of Expertise. In: AAAI-96 Workshop On Integrating Multiple Learning Models (1996) 53-58.
9. Merz, C.: Dynamical Selection of Learning Algorithms. In: D. Fisher, H.-J.Lenz (Eds.), Learning from Data, Artificial Intelligence and Statistics, Springer Verlag, NY (1996).
10. Puuronen, S., Terziyan, V.: The Voting-type Technique in the Refinement of Multiple Expert Knowledge. In: Sprague, R. H., (Ed.), Proceedings of the Thirtieth Hawaii International Conference on System Sciences, Vol. V, IEEE CS Press (1997) 287-296.
11. Rada R., Mili H., Bicknell E., Blettner M.: Development and Application of a Metric on Semantic Nets, IEEE Tr. on Systems, Man, and Cybernetics, Vol. 19, No. 1 (1989) 17-30.
12. Rocha L., Luis M.: Fuzzification of Conversation Theory, In: F. Heylighen (ed.), Principia Cybernetica Conference, Free University of Brussels (1991).
13. Skalak, D.B.: Combining Nearest Neighbor Classifiers. Ph.D. Thesis, Dept. of Computer Science, University of Massachusetts, Amherst, MA (1997).
14. Tailor C., Tudhope D., Semantic Closeness and Classification Schema Based Hypermedia Access, In: Proceedings of the 3-rd International Conference on Electronic Library and Visual Information Research (ELVIRA'96), Milton, Keynes (1996).
15. Terziyan, V., Tsymbal, A., Puuronen, S.: The Decision Support System for Telemedicine Based on Multiple Expertise. International Journal of Medical Informatics, Vol. 49, No. 2 (1998) 217-229.
16. Terziyan, V., Tsymbal, A., Tkachuk, A., Puuronen, S.: Intelligent Medical Diagnostics System Based on Integration of Statistical Methods. In: Informatica Medica Slovenica, Journal of Slovenian Society of Medical Informatics,Vol.3, Ns. 1,2,3 (1996) 109-114.
17. Wilson D., Martinez T., Improved Heterogeneous Distance Functions, Journal of Artificial Intelligence Research, Vol. 6 (1997) 1-34.