ECAI 2008
M. Ghallab et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-891-5-881

A probabilistic trust model for semantic peer-to-peer systems

Nguyen Gia-Hien[1] and **Chatalic Philippe**[2] and **Rousset Marie-Christine**[1]

1 Preliminaries and illustrative example

We consider a network of semantic peers $\mathcal{P} = (P_i)_{i=1..n}$. Each peer P_i uses its own ontology, expressed on its own vocabulary V_i, for describing and structuring its knowledge as well as for annotating its resources. A class $C \in V_i$ of a peer P_i is referred by $P_i{:}C$ or simply by C when no confusion is possible. Peers are connected each other by means of *mappings*, corresponding to logical constraints linking classes of different peers. Users ask queries to one of the peers, using the vocabulary of this peer. When processing a query, the reasoning propagates from one peer to other peers thanks to those mappings. The mappings are exploited during information retrieval or query answering for query reformulation between peers.

For example, let us consider a semantic P2P system sharing movies based on semantic annotations, where P_1 organizes his video resources according to their genres (*Suspense*, *Action*, *Animation*), and P_2 organizes his films based on the actors playing in the movies (Bruce Willis, Jolie). While having different views for classifying movies, P_1 and P_2 can establish some mappings between their two classifications. For example, they can agree that the class *BruceWillis* of P_2 (denoted by $P_2{:}BruceWillis$) is more specific than the class *Action* of P_1 (denoted by $P_1{:}Action$). It will result into the mapping $P_2{:}BruceWillis \sqsubseteq P_1{:}Action$. Similarly, P_1 and another peer P_3 can have established the mapping $P_1{:}Action \sqcap P_1{:}Suspense \sqsubseteq P_3{:}Thriller$ between their two classifications, in order to state that the category named $Thriller$ by P_3 is more general than what P_1 classifies as both $Action$ and $Suspense$. As a result, the movies that are classified by P_1 as $Suspense$ and by P_2 as $BruceWillis$ are returned as answers to the query $Thriller$ asked by the user at the peer P_3.

We assume that each resource r returned as an answer to some query is associated with a label $L(r) = C_{i1} \dots C_{iL}$ corresponding to its logical justification. $L(r)$ is a set of classes of the vocabularies of (possibly different) peers known to annotate the resource r and supposed to characterize a *sufficient condition* for r to be an answer. Any other resource annotated in the same way is thus equally supposed to be an alternative answer to the query. We also assume that the classes used in labels are *independent* in the sense that for any two classes of a justification, none of them is a subclass of the other. This important assumption means that for a returned answer, the only classes that appear in its justifications are those corresponding to most specific classes of the network.

Finally we assume that the user, when querying a peer P_i, is randomly asked to evaluate some of the returned answers as *satisfying* or *not satisfying* and to store the result of this evaluation in a local *observation database* $\mathcal{O}_i$. Each evaluation is recorded into $\mathcal{O}_i$ as a pair $S.L$ or $\overline{S}.L$, where S (resp. $\overline{S}$) denotes the user satisfaction (resp. unsatisfaction) and L is the label of the evaluated resource.

Definition 1 (Observation relevant to a label L) *Let $\mathcal{O}_i$ be the set of observations of a peer P_i and L be a label. An observation of $\mathcal{O}_i$ is said to be* relevant to L *if and only if its label contains all classes of L. The number of satisfying and unsatisfying observations of P_i that are relevant to L are respectively denoted by:*
$O_i^+(L) = |\{S.L' \in \mathcal{O}_i / L \subseteq L'\}|$
$O_i^-(L) = |\{\overline{S}.L' \in \mathcal{O}_i / L \subseteq L'\}|$

These two numbers summarize the past experience of the peer P_i relevant to the label L, i.e. of the evaluated resources justified by at least the classes of L.

For instance, suppose that Peter is the user querying the peer P_1. After a number of answers have been evaluated, Peter's past experience may be summarized as in table 1.

Label (L)	$O_1^+(L)$	$O_1^-(L)$
$P_2 : MyActionFilms$	30	6
$P_2 : MyCartoons$	3	15
$P_4 : ScienceFiction$	14	14
$P_5 : Italian\ \ P_5 : Western$	0	6
$P_6 : AnimalsDocum$	8	2
$P_7 : JeanRenoir$	22	11
$P_8 : Bollywood$	6	35

Table 1. Summary of Peter's observations at P_1

Among all the resources evaluated by Peter and annotated with the class $MyActionFilms$ of the peer P_2, 30 have been considered as satisfactory and 6 as not satisfactory. For the same peer P_2, only 3 out of 18 evaluated resources tagged by $MyCartoons$ were positive. Similarly all evaluated resources annotated with both $Italian$ and $Western$ by P_5, obtained negative feedbacks.

Each peer P_i can progressively update its observation database $\mathcal{O}_i$, as new answers are evaluated, and refine the trust it has towards answers justified by the different observed labels. The level of trust can vary according to the justification.

2 Bayesian model and estimation of trust

Given a label L, let X_{iL} be the binary random variable defined on the set of resources annotated by L as follows:

$$X_{iL}(r) = \begin{cases} 1 & \text{if the resource } r \text{ is satisfying for } P_i \\ 0 & \text{otherwise} \end{cases}$$

[1]University of Grenoble, LIG, France, email: gia-hien.nguyen@imag.fr, marie-christine.rousset@imag.fr
[2]Univ. Paris-Sud, LRI, France, email: philippe.chatalic@lri.fr

We define the trust of a peer P_i towards a label L as the probability that the random variable X_{iL} is equal to 1, given the observations resulting from the past experiences of P_i.

Definition 2 (Trust of a peer towards a label L) *Let $\mathcal{O}_i$ be the set of observations of a peer P_i and L be a label, the trust $T(P_i, L)$ of P_i towards L is defined as follows:*

$$T(P_i, L) = Pr(X_{iL} = 1|\mathcal{O}_i)$$

The following theorem provides a way to estimate the trust $T(P_i, L)$ of a peer P_i towards a label L, and the associated error of estimation.

Theorem 1 *Let $\mathcal{O}_i$ be the set of observations of a peer P_i and L be a label. After $O_i^+(L)$ satisfying and $O_i^-(L)$ unsatisfying observations relevant to L have been performed, $T(P_i, L)$ can be estimated to*

$$\frac{1 + O_i^+(L)}{2 + O_i^+(L) + O_i^-(L)}$$

with a standard deviation of

$$\sqrt{\frac{(1 + O_i^+(L)) \times (1 + O_i^-(L))}{(2 + O_i^+(L) + O_i^-(L))^2 \times (3 + O_i^+(L) + O_i^-(L))}}$$

It follows from a well known result (e.g., [3],page 336) in probabilities of the application of the Bayes rule to random variables following a Bernoulli distribution the parameter of which is unknown.

Table 2 summarizes the estimations with their associated standard deviation obtained by applying Theorem 1 to the Peter's observations summarized in Table 1.

Label (L)	Estimated trust of P_1 towards L	Standard deviation
$P_2 : MyActionFilms$	0.815	0.062
$P_2 : MyCartoons$	0.2	0.087
$P_4 : ScienceFiction$	0.5	0.089
$P_5 : Italian\ \ P_5 : Western$	0.125	0.11
$P_6 : AnimalsDocum$	0.75	0.12
$P_7 : JeanRenoir$	0.657	0.079
$P_8 : Bollywood$	0.162	0.055

Table 2. Estimated trust of P_1 towards the labels of Table 1

3 Propagation of trust

When the observation database does not contain enough observations relevant to a label for computing trust with a good precision, we have to use some propagation mechanism to compensate for the lack of local relevant observations.

Instead of propagating trust between peers, our approach consists in *propagating the pairs of numbers used for computing trust*. Propagating two numbers instead of one does not represent a significant overhead. Yet, it has the significant advantage of providing a well-founded way to compute a joint trust using the same Bayesian model as the one presented in section 2.

Instead of using an ad-hoc aggregation function for combining local coefficients of trust, the numbers $O_{i1}^+(L) \dots O_{il}^+(L)$ (respectively $O_{i1}^-(L) \dots O_{il}^-(L)$) coming from solicited peers $P_{i1} \dots P_{il}$ are cumulated to compute the joint trust of the subset $P_{i1} \dots P_{il}$ towards L, by applying the formula of Theorem 1.

Different strategies are possible to gather on the querying peer the relevant information from the solicited peer's observations.

- The *lazy* strategy consists in waiting for getting some answer justified by a label L and then asking one or several trusted neighbors for their direct feedbacks about the label L. Since it applies after the obtention of answers, such a strategy can be used as a post-precessing and does not require to change the query evaluation mechanism itself. As a consequence it can be applied to different kinds of semantic P2P systems, provided they are able to justify answers by means of such labels (e.g. sets of independant semantic annotations).
- The *greedy* strategy consists in collecting the direct feedbacks likely to be relevant (i.e., concerning the classes in the annotation being built) during the query processing. It thus requires some adaptation of the query answering algorithm. In a system like SOMEWHERE [1], the DECA algorithm [2] is first used to infer, from the ontologies and mappings, all the possible reformulations (i.e. rewritings) of the initial query into conjunctions of extensional classes (i.e. containers of instances) $C_1, \dots, C_n$. Each instance in $C_1 \sqcap \dots \sqcap C_n$ is then produced as an answer, $C_1 \dots C_n$ being the semantic annotation justifying it. The DECA algorithm can be slightly modified in order to convey, when transmitting back rewritings from a queried peer P to the querying peer P', those feedbacks likely to be relevant. When a rewriting $C_j \sqcap \dots \sqcap C_m$ is transmitted from P to P' within a message, P uses that message to convey its direct observations $(O^+(L), O^-(L))$ for all labels L containing the classes of the rewriting. By construction, those classes will be part of the annotation of an answer. Therefore, observations relevant to these classes may be relevant for computing (if needed) the joint trust towards the labels annotating answers returned to the peer the initial query is issued from. Note that this strategy leads to combining feedbacks from the very peers that have contributed to obtain an answer. Those peers may thus be considered as naturally relevant for obtaining appropriate feedbacks. However, such sets of peers are determined at query time and may vary according to the query and the returned answer.

4 Perspectives

One of the objectives of reputation systems is the detection and handling of malicious agents in an electronic environment. In a P2P system, a peer can be malicious by providing to other peers virus-affected resources, or by simply lying when reporting its feedbacks about others. In our model, when a peer has enough direct experiences, it does not have to rely on other peers and thus avoid malicious peers. When it has to rely on observations of other peers for estimating its trust towards a label, it is reasonable to assume that the number of malicious peers is small. Therefore, it is possible to either increase the number of peers to solicit to get observations (in order to decrease the impact of wrong observations coming from few peers) or to discard the peers the observations of which change a lot the joint trust (they are likely to be malicious).

REFERENCES

[1] P. Adjiman, Philippe Chatalic, François Goasdoué, Marie-Christine Rousset, and Laurent Simon, 'Somewhere in the semantic web.', in *PPSWR*, pp. 1–16, (2005).

[2] Philippe Adjiman, Philippe Chatalic, François Goasdoué, Marie-Christine Rousset, and Laurent Simon, 'Distributed reasoning in a peer-to-peer setting: Application to the semantic web', *Journal of Artificial Intelligence Research*, **25**, 269–314, (2006).

[3] Morris H.DeGroot and Mark J.Schervish, *Probability and Statistics*, Addison Wesley, 2002.