Multiagent Cooperative Learning of User Preferences

Adorjan Kiss and Joël Quinqueton

LIRMM, Multiagent Team, 161 rue Ada, F-34392 Montpellier Cedex, France
{kiss,jq}@lirmm.fr,
WWW home page: http://www.lirmm.fr/~kiss

Abstract. We present in this paper a Machine Learning method designed to predict preference knowledge in a multi-agent context. In the first part we show some theoretical properties of our learning scheme and then present an application of it to a corporate knowledge management system.

1 Introduction

In this paper we will present some attempts to design a Machine Learning method to predict preference knowledge in a multi-agent context. Here we define preference knowledge as knowledge about a preference between elements of a set.

For instance, the documents found by a search engine on the web are ordered according to a preference function computed from the user request. Thus, they can be considered as ordered according to a preference relation.

The framework that gave birth to this work is a joint research project, CoMMA[1], dedicated to corporate memory management in an intranet. The main objective of the project is to implement and test a Corporate Memory management framework integrating several emerging technologies in order to optimize its maintenance and ease the search inside it and the use of its content by members of the organization.

The main challenge is to create a coherent system that relies upon several promising new technologies which are in the middle of their struggle to become standards:

- Multi-agent architecture: it is well suited to the heterogeneity of the Corporate Memory; its flexibility eases the system maintenance and keeps the rhythm with the dynamics and evolution of the Corporate Memory; cooperating and adaptive agents assure a better working together with the user in his pursuit to more effectively achieve his goals. The FIPA standard, supported by the CoMMA project, offers the specifications for interoperable intelligent multi-agent systems.

[1] This work was supported by the CoMMA (Corporate Memory Management through Agents) project [Con00] funded by the European Commission under Grant IST-1999-12217, which started beginning of February 2000.

- XML: is a standard recommended by the World Wide Web Consortium intended to offer a human and machine understandable description language: a good choice if it is important to ensure an easy maintenance, and seamless flow through various information processing systems that are expected to evolve in time.
- RDF/RDFS another W3C recommendation, that creates a semantic level on the top of XML formal description. RDF annotations allow having an integrated, global view of the Corporate Memory keeping untouched (in terms of storage, maintenance) the heterogeneous and distributed nature of the actual info sources. RDF also allows us to create a common ontology to represent the enterprise model. The ontological commitment, a fundamental choice in our approach to design the Corporate Memory, is motivated by our belief that the community of corporate stakeholders is sharing some common global views of the world that needs to be unified and formalized (RDFS) to form the basis of the entire information system.
- Machine Learning Techniques make the system adaptive to the user, and come even more naturally due to the previous choices, as presented in the following section.

2 The use of Preference knowledge

Here we define preference knowledge as knowledge about a preference between elements of a set. Such knowledge can be stated in various forms: a numerical value assigned for each item, a total ordering relation, a partial ordering relation or even a preordering of the set.

Logical models have been proposed to deal with such knowledge, some dealing directly with the comparison abilities [Sch96] and others inspired from discrete linear-time temporal logics [SR99].

It is generally admitted that a preference stands for an order that maybe partial, even a preorder, but that it is often convenient to represent it by a linear extension (which is a total order) or a numeric value compatible with the known orderings.

Then, in terms of Machine Learning, different strategies may be used, depending on the form of the preference knowledge.

2.1 Numeric labelling

A numeric labeling, i.e. a mapping of our set of examples into a set of real numbers, is a convenient way to summarize a preference relation. Some Machine Learning methods are available to learn numerical variables [Gas89,Bre96b,Bre96a].

Generally, the methods for learning to predict a numerical variable v measure the quality of a predictive rule R by the standard deviation $\sigma^2(v)$ of the value of the variable among the set of objects verifying the concept R to be tested.

$$Q(R,v) = \frac{1}{|R(x)true|} \sum_{x}^{R(x)true} (v(x) - \bar{v})^2$$

The lower $Q(R, v)$ is, the better R is to predict v, because the mean value of v can be used with less error. With such criteria, any learning method will lead to grouping several neighbour values around their mean. Then, the learnt rules will not be very different from rules learnt from examples roughly rated with a finite set of values.

2.2 The order relation

By definition, a binary relation, which we note $\triangleleft$, is an order if it has the following properties:

- reflexive $x \triangleleft x$,
- transitive: if $x \triangleleft y$ and $y \triangleleft z$, then $x \triangleleft z$,
- antisymmetric: if $x \triangleleft y$ and $y \triangleleft x$, then $x = y$.

Then, we can imagine to learn the binary relation by learning each of its elements, that is, learn on each couple of objects (a, b) such that $a \triangleleft b$. Then, let us summarize the suitable properties of such a learning set for this approach to work correctly.

First, if (a, b) with $a \triangleleft b$ is an example, then (b, a) is a counter-example. Then, what happens to (a, a)? We can see that they would be both examples and counter-examples, then it is better to consider the strict order relation, and eliminate diagonal elements.

With these hypotheses, the description of an example is made of 2 parts: the attributes which are modified between a and b, and those which keep the same value. We can notice here that these attributes are the same as those involved in the sorting tree of our examples.

Then, our method appears to be "half lazy", in comparison with lazy learning methods, like kNN or LWR [Aha92]. Our learned knowledge is partly explicit, but in the classification step, we need to compare a new instance with several elements of the learning set (maybe in a dichotomic way) to put it in the right place.

2.3 Statistical evaluation criteria

Usually in Machine Learning, particularly for building of decision trees, the learned classification rules are evaluated by their similarity to the desired classification.

We can use the same principle here, and we have two possible families of criteria. If we can compute a rank for each element, the similarity is computed by measuring the rank correlation to the expected ranking. Otherwise, each pair must be given: then we use a pairwise comparison between the expected order and the learnt order.

Several measures of similatity between 2 different orderings of the same data have been proposed. In each case, one has to deal with tied elements.

The Spearman rank order correlation The Spearman rank order correlation r_s is a correlational measure that is used when both variables are ordinal. The traditional formula for calculating the Spearman rank-order correlation is

$$Corr(r, r') = 1 - \frac{6\sum_{i=1}^{i=n}(r_i - r'_i)^2}{n(n^2-1)}$$

where r and r' are the ranks to compare of paired ranks. When there are tied cases they should be assigned the mean of their ranks. The mean of the ranks from $p+1$ to $p+n$ is $\frac{1}{n}(\frac{(n+p)(n+p+1)}{2} - \frac{p(p+1)}{2})$, which become after simplification $\frac{n+2p+1}{2}$.

The Kendall pairwise $\boldsymbol{\tau}$ criterion When we have to compare each pair of data, they can be classified as either tied (T), concordant (P), or discordant (Q).

The best measure for this case is Kendall's τ_b which takes into account a correction for tied pairs. Its formula is

$$\tau_b = \frac{P - Q}{\sqrt{(P + Q + T_x)(P + Q + T_y)}}$$

where T_x is the number of pairs tied on X but not Y, and T_y is the number of pairs tied on Y but not X.

2.4 Verifying the consistency of the method

In the case we perform a pairwise learning of the order relation, we can notice that a fundamental property, the transitivity, can be guaranteed by the learning process itself, as we show below for a version space method [Mit97].

We can check that, if, for 3 examples the transitivity holds, then it is not necessary to add the 3rd pair as example to learn the relation :

- let $(a, b) = (a_1 \dots a_n, b_1 \dots b_n)$ and $(b, c) = (b_1 \dots b_n, c_1 \dots c_n)$. Then, S is of the form $L \wedge R$, with the left part L as a generalisation of both a and b, and the right part R of both b and c. Then, as L is a generalisation of a and R of c, S is a generalisation of (a, c).
- with the same conventions, G has a disjunctive form whose elements reject all the examples, then, if we represent any of its elements as $L \wedge R$. If (b,a) and (c,b) are rejected, it means that L rejects b or R rejects a, and L rejects c or R rejects b. But G must also be a generalisation of S.

Of course, this is only a scheme of the proof, and is, strictly speaking, only available for version-space-like learning. In a more general case, like decision tree learning, we can only make the hypothesis that it is true. We concluded that we could learn directly a sorting rule (in a greedy way, like decision trees) and evaluate the obtained rule with the τ criteria defined in section2.3.

Let us now describe more widely our application.

3 Using preference in Knowledge Management

This section aims to present and discuss how our work on preference learning fits into CoMMA project's Knowledge Management System [Con00].

3.1 Getting the user profile

One of the advantages of an enterprise that should be exploited by such a corporate information management system is that the users (i.e. the employees) can be known (their domains of interest/competence, their current activities/tasks). This can be especially useful in some cases where users are likely to be overwhelmed by the quantity of information to process and navigate themselves through (new employees during accommodation, technology monitoring scientists) who would appreciate personalized automated help in their process of information retrieval. Nevertheless, using Machine Learning to reach this goal can present some challenges. Some generic solutions are presented in [WPB01].

3.2 Using semantic annotations

Secondly, we have "human and machine understandable" semantic information upon the corporate knowledge offered by the RDF formalization, based upon an "enterprise ontology" (RDF schema).

The combination of these two sources of information can provide a rich ground to infer knowledge about the users' probable/possible preferences. This combination is made possible due to the fact that we use the same RDF standard for formalizing the user profile; the same base ontology for the enterprise and user models.

It can be imagined that the info combined from these sources form sets of attributes that will be used as input for an ML mechanism.

In order to set up such a ML mechanism, there are two main tasks to complete:

1. Getting and formalizing the information to be decomposed as attributes to feed the ML mechanism.
2. Defining the ML methodology to process this info

3.3 Collecting the information to create a set of most meaningful attributes

We will need to answer the following question: *Why does a user prefer a document?*

In our attempt to give an example of some possible answers, we are gradually going deeper and deeper into details in case of complex answers: *The document is interesting.*

- Because it has been stated so:

- By the user himself (the user has already seen the document, and "told" the system, that he is interested in)
- By someone else (someone, maybe "close" to the user, wanted to share a favorable opinion about a document)

– Because it concerns a topic close to the user's *interest fields*:
  - by the relation with the user:
    - ∗ Personal interest fields
    - ∗ Professional interest fields (known by his role in the enterprise)
  - by the way they are obtained:
    - ∗ Declared interest fields (the user has stated his interest documents concerning a topic)
    - ∗ Implied interest fields (the user is included in a community of interest which is close to a topic)

The second question, that introduces the notion of temporality into the preference: Why does a user prefer a document at a given moment?

In other words, to make the difference from the first question: *Why does a user prefer a document at a given moment, and does not prefer it at another moment?*

– The document is interesting only if seen the first time (or the first few times)
– It is interesting during a certain period (when the user performs a certain activity, etc.)

These answers are just some samples, one can think of many other possible reasons. Though, we realize that it is a very important to find the right questions and answers, that include the majority of possible situations. Indeed, getting the right questions and answers and translating them into quantifiable attributes, and making sure that the highest number of possible situations are observed is a key to the success of such a learning mechanism, that may even outclass in importance the chosen learning technique.

Nevertheless, we will present our approach in the Comma project to choose some typical answers and attributes, but we keep more focused on the second issue: the preference learning methodology.

3.4 Learning in the CoMMA system

After a short presentation of the design of the CoMMA system, this section presents the multiagent interaction in which Machine Learning is performed in CoMMA.

The chosen MAS consists of a society of coarse-grained agents, that fulfill in general multiple roles, and are organized in a small number of functional sub-societies. The MAS architecture was designed in order to optimize task-division, flexibility and robustness of the system, and network layout (extensibility, scalability, traffic optimization).

For the implementation of the prototype system, the Jade agent platform was chosen, which is an Open Source Project developed by project partners, University of Parma and CSELT. Jade is a FIPA compliant agent platform, implemented in Java, and has also the advantages of a wide opening towards Internet and the Web, interoperability with other MAS-s, and future systems.

In the current status of implementation, the CoMMA system will help the user in three main tasks:

- insertion and RDF annotation of documents,
- search of existing documents, and
- autonomous document delivery in a push fashion to provide her/him with information about new interesting documents.

We have already experimented such an architecture in Network Supervision [EDQ96], with Machine Learning abilities [QEN97]. Here, for the Machine Learning part, we choose to use the Weka open source Java library [WF99], which enables us to experiment various learning methods and frameworks.

3.5 The learning agent

The first context to assess preference learning was chosen to be the document retrieval scenario, via semantic annotations. The search engine used for document retrieval in the CoMMA system is an inference engine called CORESE [CDH00] developed by INRIA, one of the partners of the project. CORESE uses Conceptual Graphs and combines the advantages of using the RDF language for expressing and exchanging metadata, and the query and inference mechanisms available in CG formalism. In order to produce inferences, CORESE exploits the common aspects between CG and RDF: it defined a mapping from annotation statements (RDF triples) to Conceptual Graphs and vice-versa.

One of the shortcomings of such a query retrieval engine is that there is no standard method to sort the information returned, such as keyword frequency in keyword-based search engines. The returned data set must be post-processed, filtered and sorted to present the user with the relevant information. Here comes the aid offered by our ML mechanism.

In the CoMMA system, information that feeds the ML comes from several sources: The document sub-society (the annotations accompanying a query response), the user sub-society (user monitoring and explicit user feedback), and ontology sub-society (to help getting the "meaning" of the results). And of course the user profile. Therefore, the learning behavior was "awarded" to the User Profile Manager agent, which belongs to the connection dedicated sub-society, and performs notably a role of middleman between agents. This decision was justified also by network traffic optimization reasons, especially because in reaction to a user action (query), several interactions can be triggered between agents of different roles.

For example, during a query retrieval, the main interactions are as described in the following diagram.

In this scenario, the role of the ML component starts when the query answers are collected and transmitted to the profile magager agent. First, the

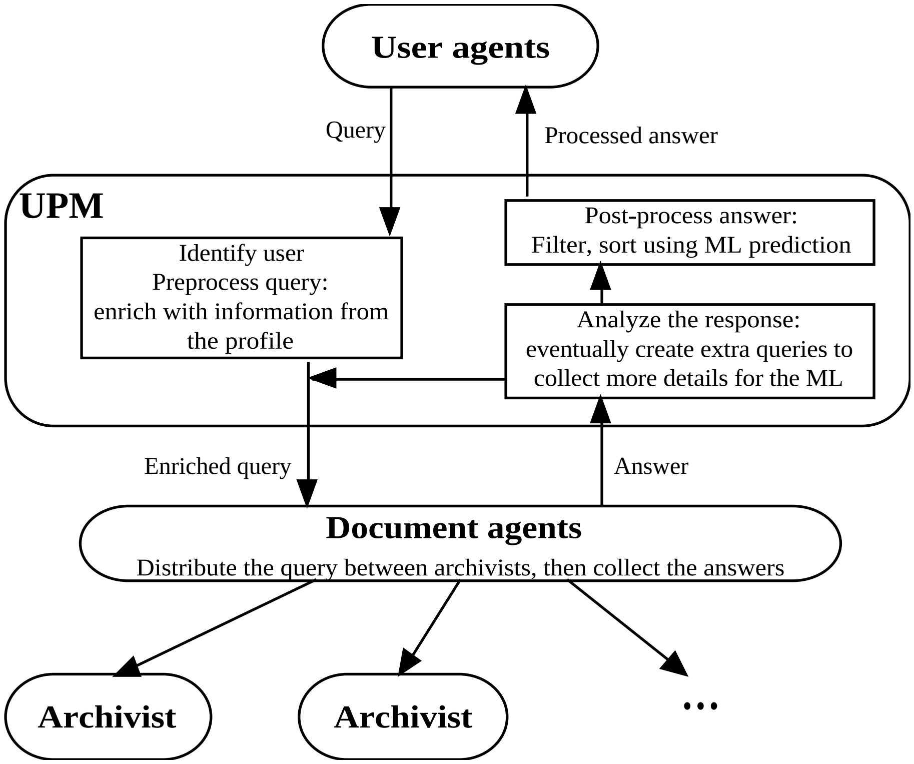

Fig. 1. The main interactions in CoMMA during a query

semantic annotations are extracted and analysed. They are combined with the relevant fields from the user profile in the attempt to compute the values of the input attributes (such as the ones listed in the section about attributes). In case the system finds that there could be more semantic annotations related to the documents retrieved that could help in computing the input attributes, a supplementary query can be formulated to retrieve them before going on with processing and forwarding the answer.

The second step is to use the learnt preference measures to rank the documents and sort the answers list to be returned by these predictions.

3.6 The learning cycle

The goal of the ML component is to produce a set of rules that will be used to produce predictions about user preferences. It can be supposed that the system starts with a set of predefined rules, that will be gradually improved during the process of adaptation to users (method known as theory refinement). Otherwise, the system may start with an empty knowledge base, and will be undergo a training period, to accumulate sufficient knowledge to allow its deployment.

In our approach, user adaptability comes from both implicit observation of the user (user monitoring subsystem), and explicit user feedback.

We use a "predict-or-learn" type protocol, that is, when the input is coming from query answers, the system tries to use its knowledge to predict, otherwise, when the input comes from the user in the form of negative feedback, the system tries to update its rules set.

3.7 A sample set of attributes

A sample set of attributes we used to create instances from the answers we gave as examples to the question of preference is listed in the followings. These attributes may not be the most relevant ones, or adapted to any case, we only tried to make it diverse and for most of them restricted the scope as much as possible for the sake of simplicity of our prototype.

Is the document related to the user's role We suppose that the users, depending on their roles, will be assigned certain interest fields, that will be recorded in their profiles. Then, the documents can have a generic $Concerns$ property, that associates them with such topics. In this scenario, the notions like $Topic$, $Concerns$, $Interestedby$, etc are defined as concepts in the ontology that constitutes the basis of the enterprise model.

As an observation, the notion *topic* or $interestfield$ is used in a general sense to categorise documents.

Then we can extract this information from the user profile and document annotation and combine them to create an instance attribute. We can define this attribute as taking a binary value.

In a complete implementation it is likely that this relationship can be seen as more complex and eventually be split into several attributes, and/or may take a wider range of values.

Is the document related to the user's COINs (communities of interests) This attribute reflects a further differenciation we have made in interest topics when answering the preference question: topics can be assigned or chosen by the user; inherited or inferred by the system, etc.

In the same conditions as for the attribute above, the attribute will take a binary value, and in case we define relationships, we can use the same procedure as above.

User experience at the company Since in our project addressing the New Employee scenario is particularly focused upon, we considered important making the system behave differently towards new versus experienced employees. We have segmented the range of values so that this may take values such as: $lessthan1week$, $lessthan1month$, $lessthan2months$, ... all these regarding the novice user, and $morethan3months$ (or simply emphexpert, or whatever the opposite for *NE* is). A tip for implementation, if such a segmentation is desired, is to define these intervals in the enterprise ontology, so it can adapt to its specific needs.

The following subset of attributes are related to *user monitoring*.

In our example we have imagined a simple user monitoring scenario, that supposes tracking each consultation of a document by the user, and building a navigation history (or consultation trace).

Document last seen Usually it is important if a document was seen before or not, and if it was then how long before. After extraction from the user's navigation history it should also be discretised into several intervals: ¡1hour, ¡1day, ¡1week, ... , never.

Average return frequency In certain cases the user may return more often to a document, in other cases an information can only present interest when first seen. This attribute may also tell something about the situation for the current case. The value will also result from processing the navigation history, and it would probably be enough to use some rough intervals (like once, small, large, etc.)

Document category touch frequency It might present an interest to extend the above attribute for categories the document belongs to. In this case the value will be processed the same way as for the previous attribute. In case a specific strategy for creating implicit communities of interest is envisaged, it should be checked if there are possible conflicts with the use of this attribute.

The next attributes contain specific information about the particular document being analysed:

The user's rating for the document For some specific documents, the user may explicitly wish to manifest his interest or non-interest. In this case the user profile should allow storing this information. It is a general vote for the document, and does not have the temporal aspect implied by the output of this classifier.

Public ratings Sometimes a user would like to share his opinion about a document with others. In this case it must be foreseen in the enterprise model so that it may be stored in the form of an annotation, that can be used also by our classifier. A method should be formalised to allow storing information about people possibly interested about this opinion.

This was a list we used in our first trials. Once again we make the remark that the list of attributes should be open, and checking the completeness and exhaustiveness of it has to be an important and ongoing task. In other words, watching that the factors that contribute to a document being seen as more or less relevant in a situation have been well captured, and that there are no other decisive factors that were omitted, or can not be represented. Because in either case, the system might make major mistakes in certain situations, whatever learning algorithm was used.

3.8 Document ranking and sorting

In our case, the first goal is to sort documents in a query response by the order of predicted user preference. The two principal strategies that can be used to achieve that, are: grouping documents using numeric labeling or learning the order relation (or the sorting rule).

For the first trial of our system, we choose to learn a rough labelling in a finite set of classes (namely 5), to classify the documents retrieved by the system after a user query. Then, we clearly fall in the first kind of strategy, whose goal is not to obtain a fine grained ranking, but only a coarse grained rating.

This is equivalent to the use of numeric labeling, associating numeric values from a finite set to categories of documents representing their importance. But the drawback is that the system will not be aware of the semantic of the order relation. And there is also a risk, that in case of large number of items to classify, many of them will fall into the same class, and there will be no further means to differentiate them.

On the other hand, if an order relation is used (either by pairwise learning or by learning the sorting rules), we will have no distance measure to separate the values. That is, to give an idea about for instance "how much a document is more important then another".

Then, a perspective for document retrieval systems of this kind can be to use both methods in a complementary way:

1. in serial-coupling (one method used to pre-process, the second to post-process)
2. in parallel (eventually in distinct agents) then putting together the results and solving conflicts.

In order to evaluate the contribution of learning to the efficiency of the retrieval system, we have designed an experimental protocol. It consists on evaluating the learning step by comparing the use trace with and without learning, in the various scenarii taken in account in the CoMMA system.

4 Conclusion

In the CoMMA project, the Machine Learning and user adaptability component is one of the main features. In this paper we presented the advances that we have made in this domain, especially focussing on the learning of preference data for document retrieval.

We proposed a specific method to learn preference data, and a framework to experiment and evaluate it. The main choice we focus on does not only present the usefulness of Machine Learning, but also tries to overcome some of the limitations of semantic information retrieval systems.

In our opinion, the choice we made here will give interesting results during the first trial we planned in the project, then allows an experimental evaluation through feedback from the user.

The implementation is currently well advanced, and we begin to have some experimental results.

References

[Aha92] D. Aha. Tolerating noisy, irrelevant, and novel attributes in instance-based learning algorithms. *International Journal of Man Machine Studies*, 36(2):267–216, 1992.

[Bre96a] L. Breiman. Bagging predictors. *Machine Learning*, 24:123, 1996.

[Bre96b] L. Breiman. Stacked regression. *Machine Learning*, 24:49, 1996.

[CDH00] Olivier Corby, Rose Dieng, and C. Hébert. A Conceptual Graph Model for W3C Resource Description Framework. In *the 8th International Conference on Conceptual Structures (ICCS'00)*, number LNCS 1867 in Lecture Notes in Artificial Intelligence, Darmstadt, Germany, 2000. Springer Verlag, Springer Verlag.

[Con00] CoMMA Consortium. Corporate Memory Management through Agents. In *E-Work and E-Business conference, Madrid*, October 2000.

[EDQ96] Babak Esfandiari, Gilles Deflandres, and Joël Quinqueton. An interface agent for network supervision. In *Intelligent Agents for Telecommunication Applications*, Budapest, Hungary, 1996. ECAI'96 Workshop IATA, IOS Press.

[Gas89] O. Gascuel. A conceptual regression method. In Katharina Morik, editor, *EWSL-89, 4th European Working Session on Learning*, pages 81–90, Montpellier, France, Décembre 1989. Jean Sallantin and Joel Quinqueton, CRIM, Pitman, Morgan Kaufman.

[Mit97] Tom M. Mitchell. *Machine Learning*. Mac Graw Hill, 1997.

[QEN97] Joël Quinqueton, Babak Esfandiari, and Richard Nock. Chronicle learning and agent oriented techniques for network management and supervision. In Dominique Gaïti, editor, *Intelligent Networks and Intelligence in Networks*. Chapman & Hall, September 1997.

[Sch96] P-Y. Schobbens. A comparative logic for preferences. In Pierre-Yves Schobbens, editor, *Working Notes of 3rd ModelAge Workshop: Formal Models of Agents*, Sesimbra, Portugal, January 1996.

[SR99] P.-Y. Schobbens and J.-F. Raskin. The logic of "initially" and "next": Complete axiomatization and complexity. *Information Processing Letters*, 69(5):221–225, March 1999.

[WF99] Ian H. Witten and Eibe Frank. *Data Mining: practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann, 1999.

[WPB01] G. I. Webb, M. J. Pazzani, and D. Billsus. Machine learning for user modeling. *User Modeling and User-Adapted Interaction (UMUAI)*, 11, 2001.