An Information Theoretic Approach to Ontology-based Interest Matching

Waikit Koh and Lik Mui

Laboratory for Computer Science
Clinical Decision Making Group
Massachusetts Institute of Technology
waikit@mit.edu and lmui@mit.edu

Abstract

Designing a general algorithm for interest matching is a major challenge in building online community and agent-based communication networks. This paper presents an information theoretic concept-matching approach to measure degrees of similarity among users. Kullback-Leiber distance is used as a measure of similarity on users represented by concept hierarchy. Preliminary sensitivity analysis shows that KL distance has more interesting properties and is more noise tolerant than keyword-overlap approaches. A multi-agent system has also been built to deploy the interest-matching algorithm.

1 Introduction

With the emergence of online communities on the Internet, software-mediated social interactions are becoming an important field of research. Within an online community, history of a user's online behavior can be analyzed and matched against other users to provide collaborative sanctioning and recommendation services to tailor and enhance the online experience [Mui, *et al*.].

In this paper, the process of finding similar users based on data from logged behavior is called *interest matching*. In the context of online societies, interest matching can help locate and cluster similar users so as to facilitate relevant knowledge and resource sharing. The ability to find users that share similar interests in online services, ranging from music-sharing networks[1], such as *Napster* and *Freenet*, to online billboards[2], such as *eGroups* and *SixDegrees*, greatly enhances the effectiveness of such online communities.

1.2 Present Approaches to Interest Matching

Many existing approaches to Interest Matching uses the keyword overlap as a measure of similarity. The cosine similarity measure extends the keyword overlap to accommodate non-binary weights associated with each keyword. Examples of such systems are Systems, such as Yenta and Retsina matching-making systems, approach interest matching from a keyword-matching viewpoint [Foner et al; K. Sycara et al].

Another interesting variant of the keyword overlap approach is the instance overlap approach. Instead of using representative keywords, each user is represented by the ratings on a set of items that he or she has posted. *Net Perceptions*, the company founded on the foundations of *GroupLens*, provides a good example: users who give similar ratings to an article tend to rate related articles similarly [P. Resnick et al]. *Amazon's* recommendation service[3] examines items bought by *Amazon*'s users; if two users have the history of buying similar items, the system infers that the user would be interested in an item bought by the other user. *Firefly Networks* (bought by *Microsoft*) and *LikeMinds* also use such similar approaches in their own domains.

The keyword overlap strategy in matching users has enjoyed relative success in the industry. However, such keyword-matching strategies ignore relations among keywords, *i.e.*, semantics. By ignoring the semantics behind the keywords, keyword-matching algorithms have difficultyextrapolating beyond the base set of keywords to predict users' interest in related concepts not captured by the same set of keywords. In this paper, algorithms that consider the relations between keywords are termed concept-matching algorithms.

1.3 Contribution

[1] Examples: Napster, http://www.napster.com; *Freenet*, http://freenet.sourceforge.net

[2] Examples: *eGroups*, http://www.egroups.com; *SixDegrees*, http://www.sixdegrees.com

[3] http://www.amazon.com/

This paper proposes an information-theoretic approach to compare interests of users. The interests of each user are captured in a weighted ontology of keywords. Each keyword is interpreted as an encoding of its sub-ontological specification consistent with information theory framework. Concepts of entropy are used to derive measures of similarities among the ontologies, thus users they represent. In addition, this paper proposes a learning algorithm to construct and modify individual ontologies that represent each user's interest.

2 Background

2.1 Theory

Ontology[4]

An ontology is "a specification of a conceptualization" [T. Gruber]. It is an explicit representation of the concepts and relations among them in a domain of interest. In agentized systems, an ontology can be used to represent what the agent "knows" about the external world. The ontology provides an environment or framework based on which the agent executes its actions. This proposal approaches interest matching from an engineer's perspective. A discussion of the validity in using ontology to represent a user's interests can be found in Section 3.

Ontologies may take many forms. In this paper, an ontology is expressed in a tree-hierarchy of concepts. In general, tree-representations of ontologies are usually poly-trees. However, for the purpose of simplicity, in this paper, the tree representation is assumed to be singly connected and that that all child nodes of a node are mutually exclusive. Concepts in the hierarchy represent the subject areas that the user is interested in. To facilitate ontology-exchange between agents, an ontology can be encoded in the DARPA Agent Markup Language (DAML). Figure 1 illustrates a visualization of this sample ontology.

Personalizing the ontology

The root of the tree represents the interests of the user. Subsequent sub-trees represent classifications of interests of the user. Each parent node is related to a set of children nodes. A directed edge from the parent node to a child node represents a (possibly exclusive) sub-concept. For example, in Figure 1, *Seafood* and *Poultry* are both sub-categories of the more general concept of *Food*. However, in general, every user is to adopt the standard ontology, there must be a way to *personalize the ontology* to describe each user. For each user, each node has a weight attribute to represent the importance of the concept. In this ontology, given the context of *Food*, the user tends to be more interested in *Seafood* rather than *Poultry*.

The weights in the ontology are determined by observing the behavior of the user. History of the user's online readings and explicit relevance feedback are excellent sources for determining the values of the weights. (Approaches to determine the weights are discussed in Section 3.)

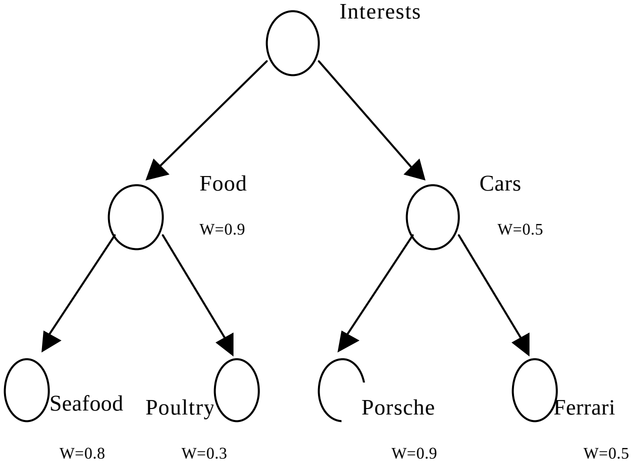

Figure 1: an example of an ontology used

The resulting ontology can be used to infer or predict the user's interest. For example, given the topic of *Food*, what is the probability that the user is referring to the topic *Seafood* versus *Poultry*? Normalizing the weights of the children nodes for *Food* provides a good first order estimate for its conditional probability distribution.

$$P_A(Seafood \mid Food) = \frac{W_{seafood}}{W_{seafood} + W_{poultry}}$$

$$P_A(Poultry \mid Food) = \frac{W_{poultry}}{W_{seafood} + W_{poultry}}$$

where the suffix A indicates that this ontology belongs to the user with id A.

The conditional probability of each directed edge from a parent to a child node represents the likelihood that given the context of the parent node the user is referring to that particular child node. The probability is in essence the weight of the child node normalized with respect to the sum of the weights of all the other children nodes:

[4] An introduction to ontology can be found at: http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

$$P_x(i \mid parent) = \frac{W_i}{\sum_{\forall child} W_i}$$

x: user
i: the ith child node of the parent node
W$_j$: the weight of the jth child node

Information Theory

Entropy, *H(p)*, of a random variable is a measure of the uncertainty of a random variable, *x* and is defined as

$$H(p) = -\sum_{\forall x} p(x) \log_2 p(x)$$

A common interpretation of this entropy measure is as an estimate on the number of binary questions required to determine the value of the random variable *x*.

The Kullback-Leiber distance (or relative entropy) is a related measure that measures the increase in uncertainty of a random variable if its true probability distribution, *p*, is not known. In particular, if probability distribution *q* is used to approximate the random variable, the resultant increase in uncertainty is

$$D(p \| q) = \sum_{\forall x} p(x) \log_2 \frac{p(x)}{q(x)}$$

3 Interesting Matching using Ontologies

An ontology of a user, as described in this paper, can be viewed as a probabilistic description of his/her interests. As such, the question of interest matching can then be reframed as follows:

> *Given the probabilistic descriptions of a user's interests (captured in his/her ontology), what is the error if this ontology is used to describe another user's interests?*

The following paragraphs clarify and expand upon this question to develop an interest-matching algorithm based on concepts in information theory.

The challenge of applying entropy measure to an ontology is the hierarchical nature of an ontology. For this paper, each user is represented by a standard ontology ontology with conditional probabilities on all the edges. Entropy of an ontology can then be defined in terms of the probability distribution of the tree. The root node is views as the random variable representing the user's interests. The leaf nodes of the tree are considered as outcomes of the user's interests. The intermediate nodes are outcomes of the random variable at varying levels of granularity. Since the ontology is a singly connected tree (therefore acyclic) and subnodes are mutually exclusive, there are no loops in the ontology.

For example, using the ontology in the previous section, the probability a query about the user ends up with the answer *Seafood,* is the *P(User, Food, Seafood)*. The joint probability can be expressed into a decision tree of dependencies:

$$P(User, Food, Seafood) = P(User)P(Food \mid User)P(Seafood \mid Food)$$

The following notation is adopted:

$$P(a,b,c,d,......., y,z) = P(a)P(b \mid a)P(c \mid b)P(d \mid c)........P(z \mid y)$$

This means, "what is the total probability of a query traversing through all the nodes from node a, b, c, … till z in a tree structure shown in Figure 2?"

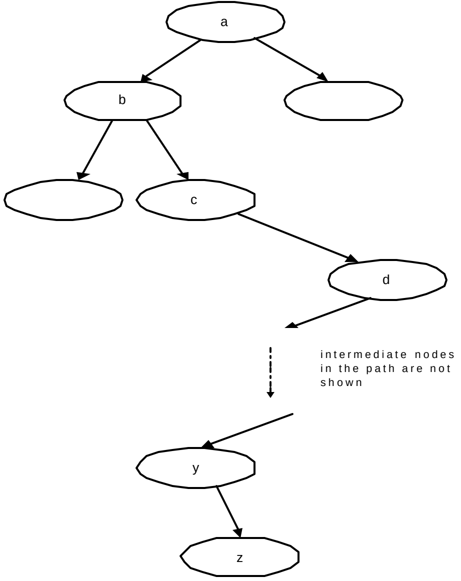

Figure 2: A path in the ontology

A simple observation can be made to derive an entropy relation between a parent and its immediate children node. Intuitively, the entropy of a sub-tree of the ontology comprises of two components:
1) The first component is the entropy of viewing the immediate children nodes of the root node of the sub-tree as a (conditional) probability distribution of that root node. 2) The second component is due to the fact the immediate children nodes themselves can possibly be the root node of its own sub-tree. As a result, each children node also has an entropy measure associated with it. Therefore the second contribution to the total entropy of the sub-tree is the weighted sum of the entropy associated with each child

node. Each weight is the conditional probability of a child node given the root node.

A simple derivation [Hyvarinen] can shown that

$$H'(r) = H(r) + \sum_{\forall k} p(k \mid r) H'(k)$$

where r is the root node of the sub-tree, H(node) being first component component and the weighted sum being second component and k is an immediate child node of r.

A recursive application of this relation on an ontology finds the entropy of the ontology.

In parallel to finding the entropy in a hierarchical fashion, the relative entropy of two ontologies can be found by applying the following relation recursively, where *r* is a common node into the two ontologies, and k is an immediate child node or *r*

$$D'(r) = D(r) + \sum_{\forall k} p(k \mid r) D'(k)$$

This formulae recursively applied on the ontology will result in the relative entropy of the two ontologies.

3.1 Ontology Matching and Relative Entropy

KL distance can be applied on the ontology of two users to measure their similarity. Interpreted within the framework of information theory, using the probability distribution of the ontology of user 1, how many extra questions do we ask about second user to predict his/her interests? If the ontologies match exactly, no extra questions are needed. If none of the nodes in the ontology overlap, no number of questions can determine the interest of the second user.

The interest-matching algorithm can be summarized as composing the following steps:

1. For each ontology, treat distinct paths from the root node to leaf nodes in the ontology (from the root node to a leaf) as possible values of a random variable.
2. Each path has an associated probability, which is the joint probability distribution of all the nodes in the path. The joint probability distribution is estimated as the product of all the conditional probabilities of the nodes in the path.
3. With the probability distribution from the two ontologies, determine the relative entropy between the two ontologies' distributions. This is a measure of similarity between the two users.
4. For a pre-defined relative entropy threshold, all users with their relative entropy similarity measures less than this threshold are considered to have matching interest.

Due to the recursive dependencies of computing the relative entropy of two ontologies, the computational complexity of the interest-matching algorithm grows with O(n) where n is the depth of the ontology.

4 Discussion

This information-theoretic ontology-matching algorithm goes beyond keyword-matching algorithms. By considering the conceptual relations between lexicons, we propose that the algorithm can perform ontology matching on closely related, but yet disjoint concepts.

In order to test the effectiveness of the algorithm, the algorithm is to be deployed in a peer-to-peer network as an interest-matching module. The effectiveness of a peer-to-peer network can be greatly enhanced if users can locate other users that are similar to them. For example, in a knowledge exchange network, it is important to find similar users to obtain relevant documents and resources. Initially, the ontology of each user is to be handcrafted by the users of the network. Once the effectiveness of the algorithm is determined and verified, the ontology construction is automated through the use of template ontologies (such as the Open Directory Project) and personalized using relevance feedback such as that returned by personalized search engines.

4.1 Learning the ontology of a user

Finding a suitable database of the user's past behavior and parsing it into the ontology that represents the user's interest is an important issue. Since this algorithm is to be deployed in an online community, data will be collected based on the user's online behavior.

Building an ontology requires a database. Suitable databases can range from the resources of the user (such as papers and documents in the user's possession) to the frequency of visits to a categorized website. Building individual ontologies is a very difficult problem. A way to circumvent the difficulty is to use a standard ontology for all users and attempt to classify each data in the user's database into the ontology. For example, the *Open Directory Project*[5] or *ODP* is a hierarchy of categories. Editors review websites and insert each URL into a particular category in the ODP. Since editors represent a large sample of Internet users, it can be viewed as a general standard ontology of subjects recognized by users in general. Using the standard ontology, the websites the user visits can be classified and entered into the standard ontology to personalize it. A form of weight for

[5] http://www.dmoz.org/

each category can then be derived: if a user frequents websites in that category or an *instance* of that *class*, it can be viewed that the user will also be interested in other instances of the class. With the weights, the formalism derived in Section 2 can be used to perform comparisons between interests of different users.

5 Simulations

5.1 Sensitivity Analysis

In order to compare the behavior of the ontology-based approach to the keyword overlap measures, two ontologies, exactly the same at time 0, are mutated by randomly selecting or unselecting nodes in the ontology. After each mutation, the KL distance is measured. The keyword-overlap measure is obtained by treating each node as an independent feature in a feature vector.

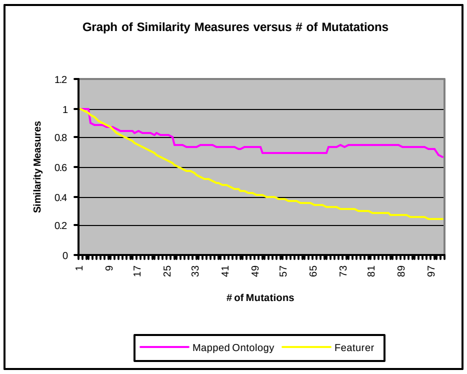

Figure 3: Graphs of similarity measures

The mapped ontology graph maps KL distance into the [0, 1] space to compare to the keyword overlap measure. The keyword overlap measure decreases in smoothly as more nodes are no longer coincidental. KL distance, however, have a strong stochastic behavior with the same general trend. The strong upswings and downswings correspond to mutations are different levels of the ontology, a feature ignored by the keyword overlap measure. Another interesting feature of the KL distance is the initial tolerance to the mutations. For the first few mutations, there is minimal change in similarity. As such, the KL distance has a stronger noise tolerance than the keyword overlap measure.

To accentuate the differences, the next graph plots the ratio of the gradients of the KL distance and the keyword overlap measure. The sharp spikes in the graph correspond to the swings in Figure 3. They isolate mutations that do not affect the KL distance strongly. Each spike corresponds to a mutation that results in a sharp change to the KL distance measure. As shown in the both graphs, the keyword overlap measure does not take into account of the structure of the ontology.

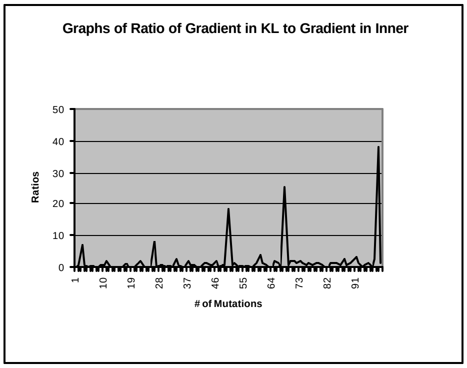

Figure 4: Graphs of similarity measures

5.2 The CSNET: a multi-agent system

Preliminary work in deploying the interest-matching algorithm has taken the form of agents in the peer-to-peer network. Each agent is a program connected to the network and is capable of automated discovery of other agents in the network. In addition to discovery protocol, agents can communicate with other agents in the network via message passing. In the context of our research, the network of agent is simulated in a prototype environment called Collaborative Sanctioning Network (or CSNet). In addition to discovery protocols, each agent also contains an ontology that represents the user’s interest. The structure of the agent-program can be divided into three functional modules: Profile; ReqHandler and Scheduler.

Profile

The profile module encapsulates information about the user and whom the user knows in the network. Running the Profile module requires two configuration files that contain configuration information pertaining to the user and the neighbors in the network that the user may know

ReqHandler

The RequestHandler module (or ReqHandler) consists of a set of handlers to handle incoming messages the agent may receive during operation. The Figure below shows the MDD of the ReqHandler module.

Upon starting up the ReqHandler module, the Listener daemon starts listening to the network at a port. Upon receiving a request from the network, the message is passed to the RequestMap, which extracts the header from the message to determine which type of handler should be created to handle the message. There is a one-to-one correspondence to the message type and handler type.

Scheduler

The ReqHandler module is essentially a set of message handlers. The Scheduler modules, in comparison, can be regarded as a set of schedules that are started up during program initiation. Each schedule probes the network of agents for new agents and regular updates on the ontological status of each agent.

Initial Results

The initial prototype of the system is a graphical display of the agents in the network and their interaction.

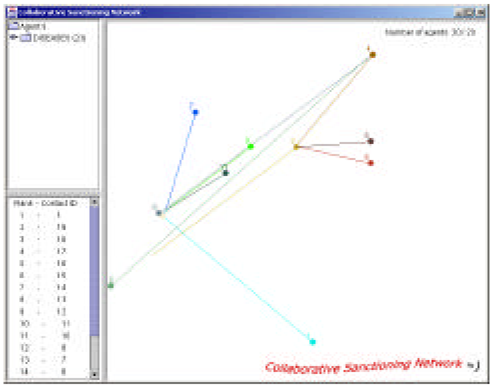
Collaborative Sanctioning Network
Rank - Contact ID
Collaborative Sanctioning Network

Figure 5: Initial display

Each node on the right display represents an agent, with their ID on the top left of each node. On the top right is a tree representation of the ontology of an agent selected. The bottom left is a sorted rank of other agents whose similarity to itself in the network as determined by the interest-matching algorithm. Agents are introduced into the network at a regular time interval. Each agent probes a range of network address to find other agents and start discovery protocols to find other agents in the network. For each agent, a line is drawn for itself to the agent it finds to be the most similar to. Since the relative entropy (the measure for similarity) is not symmetrical, ($D(p\|q)$ is not the same as $D(q\|p)$), this produces a directed graph in the network. As more agents are added to the network, the edges in the graph will change over time.

In order to accentuate the similarities and differences of the agents in the display, each node will pull the most similar node towards itself and repel the most dissimilar node away from it. The next figure shows the program after sufficient time is given for the positions of the nodes to stabilize.

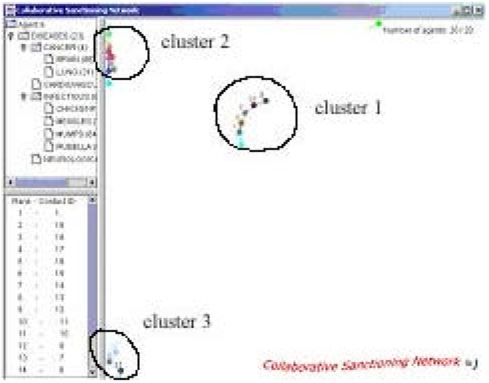
Collaborative Sanctioning Network
cluster 2
cluster 1
cluster 3
Rank - Contact ID
Collaborative Sanctioning Network

Figure 6: Display after position has stabilized

As shown, clusters of nodes are formed. The clusters can be interpreted as groups of users whose ontology are the most similar to those in the group. This grouping mechanism can form a basis for future analysis of future online communities.

6 Conclusion

This paper has proposed an interest-matching algorithm by applying concepts in information theory to compare ontologies for similarity. By viewing the nodes in ontology as a probability distribution of interests of the user, an ontology of a user can be used to predict his or her similarity of interest to the others in the online community. KL distance measures the accuracy of the prediction by comparing the actual ontology of the user to be predicted and the ontology used to predict the interests of the user. Simulations results show that KL distance have more interesting properties and is more noise tolerant than the usual keyword-overlap approach. The effectiveness of the ontology-matching algorithm is to be determined by deploying it in various instances of online communities. This paper also presented a prototype multi-agent simulation environment, which implemented the interest-matching algorithm presented in this paper.

7 References

[L Foner et al] L. Foner. Yenta. *A Multi-Agent Referral Based Matchmaking System*. Proceedings of the First International Conference on the Practical Application of<E-395> Intelligent Agents and Multi-Agent Technology, pp. 245-261, London, UK, April<E-387> 1996. <E-18>

[L. P. Hyvarinen] *Information for Systems Engineers*. Springer Verlag New York. 1970

[P. Maes and M. Metral] *http://agents.www.media.mit.edu/groups/agents/projects/* MIT Media Lab Software Agents group Project HOMR (Firefly)

[L Mui et al] L Mui. P Solovitz. *Collaborative Sanctioning*. Autonomous Agents Conference 2001. To appear

[P Resnick et al] P. Resnick, N Iacovou, M. Suchak, P. Bergstrom, J. Riedl. *GroupLens: An Open Architecture for Collaborative Filtering of Netnews*. Internal Research Report, MIT Center for Coordination Science, March 1994.

[K Sycara et al] K. Sycara, J. Lu, M.Klusch, S. Widoff. *Matchmaking among Heterogeneous Agents on the Internet*. In Proceedings of the AAAI Spring Symposium on Intelligent Agents in Cyberspace, Stanford, USA, 1999