Incremental Non-Unanimous Concept Reformation through Queried Object Classification

Nima Mirbakhsh, Arman Didandeh
Department of Computer Science
Institute for Advanced Studies in Basic Sciences,
Zanjan, Iran,
{*nmirbakhsh,a_didandeh*}@*iasbs.ac.ir*

Mohsen Afsharchi
Department of Electrical and Computer Engineering
University of Zanjan, Zanjan, Iran
afsharchim@znu.ac.ir

***Abstract*—Given a set of observations or new information, agents should be able to update their understandings of the world. As a part of any agents' world ontology, concepts need to evolve in time. In this paper we present a new representation for non-unanimous concepts based on the combination of feature-values and their probabilities. This representation leads us to incrementally evolve the concepts upon facing with new observations or information. As agents' access to the knowledge structure of the peer agents is limited due to the high cost of communication, we enabled our agents to use any queried object and update the previously calculated probability of every feature-value combination based on the probability of that object being an instance of a concept.**

***Keywords*-Non-unanimous Concepts, Concept learning, Concept Reformation**

I. Introduction

Agent collaboration has always been a must to form any multi agent system while any agent has partial access to the environment information. While communication does not always have to involve the use of a language, for many purposes utilizing a language is a very convenience way to convey information between agents. In addition to this common language, a common semantics is also necessary for communicating agents to interact and understand each other. Ontology research community tries to address issues arisen from violation or relaxation of any of the above two requirements. [17]

Several attempts have been made to form an approach in a multi agent system, when an agent can find no or not sufficient inner information when it happens to meet a situation. A recent approach is to let the agents have their individualized ontologies and provide them with learning mechanisms to learn the concepts they need during communication [10] [7] [3] [6]. While almost all authors have looked at one agent teaching one other agent, in [1] we presented a general framework that allows an agent to learn a concept from a group of teacher agents. At first glance, learning from a group of agents instead of a single agent only seems to add potential problems, namely that the teachers might not agree on some aspects of a concept to learn, so that it is up to the learning agent to decide on these aspects on its own. In [2] we presented an extension to the definition of a concept in an ontology that allows an agent to simultaneously communicate with a group of agents that might have different understandings of some concepts. We also provided a way to learn such non-unanimous concepts by using a method for learning concepts from a group of teachers. The general idea of non-unanimous concepts is to use the teachers to identify the core of a concept that everyone agrees on and the remaining shall be what at least some of the teachers think belongs to the concept. The learning agent also decides what belongs to the concept for itself and whenever it needs to communicate with a group of other agents and needs to be precise, it makes use of these three concept aspects by providing additional example objects for what might be misunderstood.

In many situations agents encounter new information which is compatible with their knowledge structure (i.e. ontology concepts) and have some explicit information that can help the agent to make the definition of a concept more concrete. This new information could come to the agent from different sources such as the queries from other peer agents. In this paper we concentrate on some scenarios in which a concept has been formed and upon arriving some new information it tries to make its understanding of some certain concept current. At first it seems that this agent can keep the object in its memory which is a kind of update for ontology concepts. Not only in many situations this keeping of objects is not feasible(e.g. web agents), but also for each object, agents having non-unanimous concepts should consult other agents to decide which boundary to put the object in. The general method in our approach enables agents to have a modified representation of non-unanimous concepts based on probabilistic interpretation of features. Based on this definition, agents can update their understanding of non-unanimous concept c_k by using Bayes conditioning. This type of reformation of understanding based on new observations(e.g. a queried object o) and information should take place *incrementally* despite of even having irrelevant and incomplete information. The process is simply explained in Figure 1.

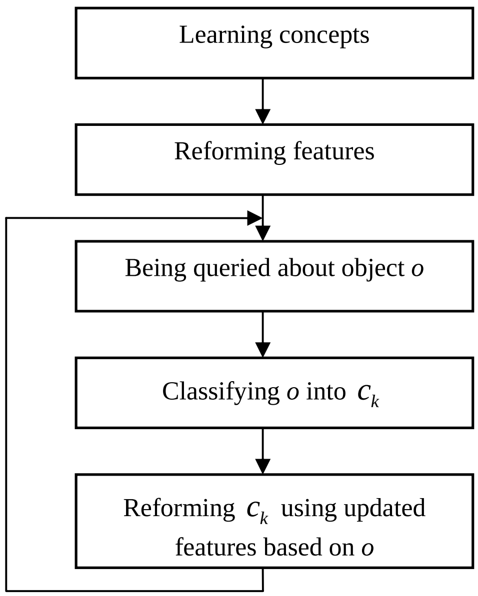

Figure 1. General concept reformation process

The structure of this paper is as follows: In the following section we give definitions for the basic concepts that we use throughout this paper. In Section III we define our non-unanimous concepts based on the feature-value combinations with probabilities. Section IV discuss our new approach to the concept reformation which results in updating agents understanding of the world. In Section V we present the result of our experiments which somehow shows the usefulness of our approach. Finally we review the related works and conclude the discussion.

II. Basic Definitions

In this section, we provide some basic definitions around ontologies, ordinary and non-unanimous concepts on which we will build our approach in the following sections.

A. Ontologies and Ordinary Concepts

A formal definition for ontology has been presented in [7] in which an *ontology* has been defined as a structure $\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R)$. C and R are two disjoint sets with members of C being called *concept identifiers* and members of R are *relation identifiers*. $\leq_C$ is a partial order on C called *concept hierarchy* or *taxonomy* and $\leq_R$ is a partial order on R, named *relation hierarchy*.

$\sigma : R \rightarrow C^+$ is a function providing the argument concepts for a relation such that $|\sigma(r_1)| = |\sigma(r_2)|$ for every r_1, $r_2 \in R$ with $r_1 \leq_R r_2$ and for every projection π_i $(1 \leq i \leq |\sigma(r_1)|)$ of the vectors $\sigma(r_1)$ and $\sigma(r_2)$ we have $\pi_i(\sigma(r_1)) \leq_C \pi_i(\sigma(r_2))$. If $c_1 \leq_C c_2$ for c_1, $c_2 \in C$, then c_1 is called a *subconcept* of c_2 and c_2 is a *superconcept* of c_1. Obviously, the relation $\leq_C$ is supposed to be connected with how concepts are defined. In the literature, taxonomies are often build using the subset relation, i.e. we have

$c_i \leq_C c_j$ iff for all $o \in c_i$ we have $o \in c_j$.

This definition of $\leq_C$ produces a partial order on C as defined above and we will use this definition in the following for the ontologies that our agents use.

Concepts often are seen as collections of objects that share certain *feature* instantiations. In this work, for an ontology $\mathcal{O}$ we assume that we have a set of features $\mathcal{F} = \{f_1, ..., f_n\}$ and for each feature f_i we have its domain $D_i = \{v_{i1}, ..., v_{im_i}\}$ that defines the possible values the feature can have. Then an object $o = ([f_1 = v_1], ..., [f_n = v_n])$ is characterized by its values for each of the features (often one feature is the identifying name of an object and then each object has a unique feature combination). By $\mathcal{U}$ we denote the set of all (possible) objects. In machine learning, often every subset of $\mathcal{U}$ is considered as a concept. In this work we want to be able to characterize a concept by using feature values. Therefore, a *symbolic concept* c_k is denoted by $c_k([f_1 = V_1], ..., [f_n = V_n])$ where $V_i = \{v'_{i1}, ..., v'_{ij_i}\} \subseteq D_i$ (if $V_i = D_i$ then we often omit the entry for f_i). An object $o = ([f_1 = v_1], ..., [f_n = v_n])$ is *covered* by a concept c_k, if for all i we have $v_i \in V_i$. In an ontology according to the definition above, we assign a concept identifier to each symbolic concept that we want to represent in our ontology.

B. Non-unanimous Concepts

In order to allow to express the range of possible misunderstandings about a concept, instead of representing a concept by one feature set as in Section II-A, we use 3 such feature sets, which means that we essentially use 3 concepts:

$c = (c_{core}, c_{own}, c_{periphery})$.

These three "normal" concepts provide us with two boundaries and the agent's own definition of the particular concept (represented by c_{own}).

Figure 2 gives a graphical representation of the trio of "old" concepts that we use to represent a *non-unanimous concept*. The inner boundary c_{core} is intended to provide the agent with a concept definition that represents all objects for which there is no doubt among all agents that they belong into the concept, so obviously c_{core} covers the *core* of the concept. The outer boundary $c_{periphery}$ covers all objects that ever might be considered to belong to the concept, which means that all objects not covered by $c_{periphery}$ for sure are not in the concept c. So, essentially $c_{periphery}$ defines the extend of the *periphery* of the concept.

If we want to use non-unanimous concepts within ontologies, then most of what we defined in Section II-A does not have to be changed. The only potential problem is $\leq_C$, since obviously there is always the chance that the peripheries of two concepts might overlap (given that the objects in $c_{periphery}$ - c_{core} are somewhat questionable

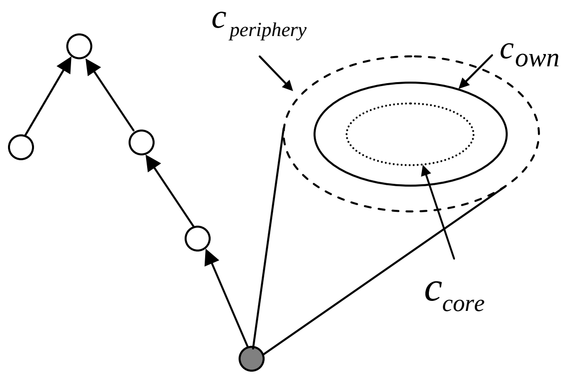

Figure 2. A Non-unanimous Ontology Concept

with regard to really representing the concept and the objects in $c_{periphery}$ - c_{own} are not covered by the concept in the point of view of the agent). If we have to provide the equivalent of $\leq_C$ for non-unanimous concepts, then we will use the relation $\leq_{C_{nu}}$, which we define as

$(c_{core_1}, c_{own_1}, c_{periphery_1}) \leq_{C_{nu}} (c_{core_2}, c_{own_2}, c_{periphery_2})$,
iff for all $o \in c_{own_1}$ we have $o \in c_{own_2}$.

This makes sense, since c_{own} represents what an agent thinks is the concept.

Finally, let us take a look at the representation of a non-unanimous concept c_{nu} using features. All the three concepts $c_{core_{nu}}$, $c_{own_{nu}}$ and $c_{periphery_{nu}}$ naturally have a presentation as feature value sets according to Section II-A. For a feature f_i this means that we have now three value sets, namely $V_{i_{core}}$, $V_{i_{own}}$ and $V_{i_{periphery}}$, with $V_{i_{core}} \subseteq V_{i_{own}} \subseteq V_{i_{periphery}} \subseteq D_i$. So, associated with potential misunderstandings in communication will be certain feature values for some of the features that an agent uses in its ontology.

III. Feature Based Definition of Non-Unanimous Concepts

Learning of new concepts in [1] is happening from scratch which means there was no place for the concept in agents' ontology before the learning process started. Due to the lack of having any predefined feature-value set to represent the concept, the learner agent starts to collect the objects from differen teacher agents. To preserve the opinion of different agents regarding a specific concept, the learner agent draws three boundaries for the concept which has originated from the conflict resolution mechanism. Although objects and the boundaries are a good way of representing non-unanimity of agents, it does not help agents to incrementally update their understanding of a concept in time. That is because every agent in its life time encounters with some new objects and it is very unlikely to ask other agents about the object if it has some concepts describing the object in its ontology. On the other hand agents usually want to take opportunity and learn from objects they have been queried about.

To be able to incrementally update the understanding of an agent about a specific concept,which in fact is concept reformation, we should have a different representation of the non-unanimous concepts which is defined by a set of features and their corresponding values while preserving the boundaries from the object based definition. As we want to update agents' understanding of concepts, this representation allows us to address a method to change some properties of feature and their values (i.e. probability) and switch to the incremental learning.

Borrowing the main idea from COBWEB [13], we define for any feature f_i and value v_{ij} in boundary bu, a conditional probability $P(f_i = v_{ij} | c_{bu_k})$ which means the probability of feature f_i having value v_{ij} in boundary bu of concept c_k. As we will see, this probabilistic definition of concept will help us in incremental reformation of concepts' boundaries. To compute these probabilities we use objects that has been collected by the agent during the first time learning of concept c_k.

We base our calculation on the fact that the frequency of any feature f_i having value v_{ij} could vary from concept to concept. Also in one specific concept the frequency of f_i having value v_{ij} varies in three different boundaries. This assumption can help agents to distinguish between concepts. Therefore we assign for any combination of features and values for a specific concept c_k and different boundaries, the probability $P(f_i = v_{ij} | c_{bu_k})$ as follows:

$$P(f_i = v_{ij} | c_{bu_k}) = \frac{|\{(f_i, v_{ij}) | f_i = v_{ij} \wedge v_{ij} \in V_i\}|}{|objects \in c_{bu_k}|} \qquad (1)$$

This means we count the number of occurrences of $f_i = v_{ij}$ in the set of all objects in boundary bu and divide it by the whole number of variations of f_i in boundary bu (which forms the space of all objects of that certain concept in its regarding boundary) to come up with a probability for $P(f_i = v_{ij} | c_{bu_k})$. It is important to notice that we do not eliminate the collected objects in c_k; we just derive another definition based on features and probabilities to help agents keep their understandings of any concept updated.

After the process of non-unanimous concept redefinition based on the features, we have for each boundary of $c_{core}, c_{own}, c_{periphery}$ probabilities assigned to every combinations of features and values.

IV. Incremental Reformation of Non-Unanimous Concepts

As stated in section I, in many situations agents' access to the domain knowledge or the knowledge structure of the peer agents is limited. Assuming an environment of collaborating agents and in many circumstances, agents query each other to learn a concept or an object. While the process of learning in the side of learner agent is well elaborated in multiagent context [10] [7] [3], it has been mainly neglected that if the queried agent can learn from the query. In this section we present a general method of incrementally updating the understanding of an agent regarding a specific non-unanimous concept using a very small piece of information called *"queried object"*.

The model we presented in Section III to represent non-unanimous concepts based on features, allows us to compute the conditional probability of feature-values given the concept (i.e. $P(f_i = v_{ij}/c_k)$) which is called *predictability*. Using these values and assuming that features are independent we can calculate the probability of an object o being an instance of concept c_k as the following:

$$P(c_k|o) = P(c_k) \prod_{f_i} P(f_i = v_{ij}/c_k) \qquad (2)$$

Using equation (1) the queried agent can decide which concept is more likely to be the object instantiated from, as the higher probability shows which concept is representing the object. After deciding about the object and the concept representing it, the queried agent has the chance to update its understanding using the objects features. Conditioning is the generally agreed-upon method for updating probability distribution when one agent learns a new object [16]. Considering the conditional probability of the concept given the feature-value (i.e. $P(c_k/o)$) which is called *predictiveness*, we can write the update equation as the follows:

$$P_{new}(f_i = v_{ij}/c_{bu_k}) = P_{old}(f_i = v_{ij}/c_{bu_k}) + f_M(c_{bu_k}, o, P_{old}, thr_{bu_k}) \qquad (3)$$

where P_{new} means any updated probability after gent being queried about a object and modified the feature-value combination probability.

The function f_M calculates the modification value in scale of three elements: $P(c_{bu_k}/o)$, $P_{old}(f_i = v_{ij}/c_{bu_k})$ and thr_{bu_k} which the latter is a threshold chosen due to the application domain and is responsible for deciding whether the amount of likelihood for the object being an instance of the regarding class. This procedure chooses to increase or decrease the value for $P_{old}(f_i = v_{ij}/c_{bu_k})$ in a sign function which returns values -1, 1 for decrease or increase needs. Then it computes a weight for this modification task and applies this change to the old value of $P(f_i = v_{ij}/c_{bu_k})$ to reach to $P_{new}(f_i = v_{ij}/c_{bu_k})$. This comes as follows:

$$\begin{gathered} f_M(c_{bu_k}, o, P_{old}, thr_{bu_k}) = \\ sgn(P(c_{bu_k}/o) - thr_{bu_k}) \times \alpha \times P(c_{bu_k}/o) \times \\ (1 - P_{old}(f_i = v_{ij}/c_{bu_k})) \times \\ P_{old}(f_i = v_{ij}/c_{bu_k}) \qquad (4) \end{gathered}$$

where α is some small inhibiter value to prohibit harsh changes in probability values.

Without loss of generality we can extend our solution of updating probabilities of ordinary concepts to non-unanimous concept. As we have the probabilities for three different areas of concepts, we can simply calculate the new probabilities of $P_{new}(c_{core_k}/f_i = v_{ij})$, $P_{new}(c_{own_k}/f_i = v_{ij})$ and $P_{new}(c_{periphery_k}/f_i = v_{ij})$.

Algorithm 1 Probabilistic incrementally update of agent understanding regarding a specific concept

Observe an object o through a query
for all concept c_i in agent's ontology **do**
  $k = argmax_i \; P(c_i) \prod_{f_i} P(f_i = v_{ij}/c_i)$
end for
for all Boundary bu in $core$,own and $periphery$ of c_k **do**
  $P(c_{bu_k}/o) = P(o/c_{bu_k}) \times P(c_{bu_k})$
  for all f_i and possible value v_{ij} **do**
    $P_{new}(f_i = v_{ij}/c_{bu_k}) = P_{old}(f_i = v_{ij}/c_{bu_k}) + sgn(P(c_{bu_k}/o) - thr_{bu_k}) \times \alpha \times P(c_{bu_k}/o) \times (1 - P_{old}(f_i = v_{ij}/c_{bu_k})) \times P_{old}(f_i = v_{ij}/c_{bu_k})$
  end for
end for

This process is summarized in Algorithm 1. After executing this process we have new probabilities representing the concept c_k.

V. Experimental Results

To study the usefulness of our approach to reformation of agents' understanding of the concepts, we use the same application domain as [1] which is quite popular in ontology learning community. As our concentration is on updating non-unanimous concepts of one agent's ontology, we assume that there is a learner agent $\mathcal{A}g$ that has learnt a non-unanimous concept and is being queried about a new object to classify it. The queried object is not in the boundary of non objects but is classified as an instance of a specific concept.

A. The University Units and Courses Domain

The university units and courses domain consists of files describing the courses offered by Cornell University, the University of Washington and the University of Michigan, together with ontologies for each of the three universities describing their organizational structure (see [4] and [9]). The objects of this domain are the course files that consist of a course identifier, a plain text course description and the prerequisites of a course. All in all, there are 19061 courses among the three universities and each university's ontology has at least 166 concepts on top of their courses. These concepts are the academic units to which the courses of the particular university belong, which naturally contains all the departments and faculties of the university.

Table I
EXAMPLES OF COURSES (OBJECTS) IN c_{core}, c_{own}, AND $c_{periphery}$ FOR CONCEPT *Computer Science* OF $\mathcal{A}g$

Border	Computer Science
c_{core}	Computer Programming I *Design and Analysis of Algorithms II* *Computer Science Research Seminar* *Introduction to Artificial Intelligence* *Computer Networks* *Introduction to Computer Organization* *Computer Architecture* *Foundations of Computer Science* *Interactive Computer Graphics*
c_{own}	Applied Logic *Theory of Computing* *Computer System Performance* *Computer Game Design and Development* *Parallel Computing* *Computational Geometry* *Formal Models in Computer Science*
$c_{periphery}$	Computational Molecular Biology *Computational Tools & Methods for Finance* *Computers and Society* *Intelligent Transportation Systems* *Introduction to Logic Design* *Reliable Computing Systems*

B. Concept Evolution: An Example

As stated before, we assume that the learner agent $\mathcal{A}g$ has learned a non-unanimous concept. Table I presents some insight into courses(i.e. objects) in concept *computer science* which has been learnt by $\mathcal{A}g$. Recalling from [2], we should mention that the examples for c_{own} are examples that are only in c_{own} and not in c_{core} and the example courses for $c_{periphery}$ are only in $c_{periphery}$ and not in c_{own}. Computer Science has 188 courses in c_{core}, 505 in c_{own} and 565 in $c_{periphery}$. As it is obvious non-unanimity is high in this concept(i.e. lot of objects in $c_{periphery}$ that are not in c_{core}) and that is because Computer Science is not a program that many universities are agreed upon.

Since the objects of our domain are essentially text files, the features that our agents use need to express properties of these files, respectively of the texts in them. A first obvious choice for a feature would be the occurrence of a particular key word (resulting in a Boolean feature) or the number of occurrences of the particular key word. Unfortunately, this idea can not be directly applied, because the texts used to describe courses that are taught by a particular unit usually do not contain any single key words that are common to all courses of this unit and that do not also occur very often in course descriptions of other units that are not parent units to the particular unit. Therefore we extended this idea to use *or-combinations* of key words to create Boolean features. So, the feature for the key word subset {*picture*, *photo*,*figure*} ($f_{picture,photo,figure}$) is true for a text t, if either *picture* or *photo* or *figure* occur in t. Moreover a major characteristic, or difficulty in the systems that use textual documents is the high dimensionality of the key word space. This unique set of key words that occurs in documents (i.e. objects) can be tens or hundreds of thousands of terms for even a moderated-size object collection. While random selection of words is a simple solution, it is not intelligent. Therefore for every particular concept learning process and in a pre-processing phase in every teacher agent, we use some techniques from the information retrieval domain to reduce the key word dimensionality. Our automatic key word selection methods include the removal of non-informative words according to a set of objects statistics, and the construction of new features which combine lower level key words into higher level orthogonal dimensions. We use $\mathcal{X}^2$ statistics (see [12]) to select key words that are basically creating the feature sets for our agents regarding every concept which is being learned. The second column(i.e. features) in Table II shows some of the main features making the *Computer Science*.

It is already mentioned in section III that we compute the prior probability of features based on the frequency of occurrences of them both in supporting objects(i.e. positive examples) and counter supporting objects (i.e. negative example) as every agent can create a set of negative examples for any concept (for more information see [1]). For our application domain and the specific type of features that we have defined, we modified the solution in [16] and calculate the probabilities as follows:

$$P(f_i|c_k) = \frac{n_i+1}{n_p+|T|} \qquad (5)$$

Where if we define S_p as a single vector created by key words occurring in objects in each boundary then n_i is the sum of the number of times each keyword f_i occurs in S_p. n_p is the total number of distinct key words in S_p and T A vector of collection of all key words that occur in the whole objects.

A preliminary result of our experiment shows that as we move from inner boundaries out, the probability of some features are decreasing. For instance it is obvious that $f_{web,algorithm,program}$ is a unanimous feature among agents and as we exclude objects that are in c_{core} from c_{own} and $c_{periphery}$ the probability of an object being in these boundaries decreases. The story is different for $f_{plasma,signal,circuit}$ which usually represents *Electrical Engineering* concept rather than *Computer Science*. The probability for this feature is increasing as we move outward and that is because it is a feature that has been originated from a non-unanimous object(i.e. objects such as "fotonic"). Hence it is unlikely to be in c_{core} boundary but the probability of being in c_{own} or $c_{periphery}$ is higher.

To see how our approach to incrementally evolving agent understanding about an agent works, we presented two object in two different rounds to $\mathcal{A}g$ as query objects and calculate the posterior probability based on the theory we have developed in Section IV. The first object we have presented was a course descriptor text file which was

Table II
SAMPLE FEATURES DESCRIBING CONCEPT *computer science* THEIR PRIOR PROBABILITIES AND PROBABILITIES AFTER TWO ROUNDS OF PROBABILITY UPDATING

Border	features	prior	round #1	round #2
c_{core}	$f_{web,algorithm,program}$	0.721	0.781	0.780
	$f_{design,processor,reliability}$	0.650	0.661	0.669
	$f_{performance,central,protocols}$	0.433	0.452	0.451
	$f_{proof,cryptography,undecidability}$	0.513	0.507	0.505
	$f_{power,image,electronic}$	0.303	0.297	0.301
	$f_{plasma,signal,circuit}$	0.257	0.246	0.251
c_{own}	$f_{web,algorithm,program}$	0.513	0.521	0.518
	$f_{design,processor,reliability}$	0.461	0.470	0.475
	$f_{performance,central,protocols}$	0.392	0.393	0.388
	$f_{proof,cryptography,undecidability}$	0.480	0.472	0.469
	$f_{power,image,electronic}$	0.419	0.416	0.425
	$f_{plasma,signal,circuit}$	0.473	0.461	0.467
$c_{periphery}$	$f_{web,algorithm,program}$	0.370	0.390	0.388
	$f_{design,processor,reliability}$	0.400	0.407	0.413
	$f_{performance,central,protocols}$	0.376	0.379	0.374
	$f_{proof,cryptography,undecidability}$	0.498	0.492	0.491
	$f_{power,image,electronic}$	0.721	0.700	0.711
	$f_{plasma,signal,circuit}$	0.619	0.601	0.610

describing a course entitled ”Semantic Web”. Our classifier algorithm based on equation (2) in Section IV classified it as an instance of *Computer Science* with probability 0.82. As it can be seen in the third column of Table II the presence of some of the keywords of the object in the features(i.e. web, program) modified the probability values for different features. For example it incremented the probability for the $f_{web,algorithm,program}$ which is very likely to be positively affected by this object. As another example the role of $f_{power,image,electronic}$ as a feature describing the *Computer Science* gets weakened because it does not get support from the keywords in the ”Semantic Web” object. The behavior of the method is almost the same for all boundaries and features.

In the second round, we presented another course descriptor as a query called ”Power Electronic”. Obviously this object contains some keywords describing the *Electronic Engineering* concept. As it is expected, this object should provide some support to $f_{power,image,electronic}$ and perhaps $f_{plasma,signal,circuit}$ and in the opposite way should weaken other features such as $f_{web,algorithm,program}$ that has been shaped from keywords in *Computer Science* objects. The very exciting point here is that because $\mathcal{A}g$ classified ”Power Electronic” with probability 0.36 as an instance of *computer science* (in fact it should not be considered as a candidate to update *Computer Science* but we presented it to see the behavior of our method) its effect on probability of features are not bold. It can be seen that the probability of $f_{power,image,electronic}$ increases from 0.297 to 0.301, from 0.416 to 0.425 and from 0.700 to 0.711 in c_{core},c_{own} and $c_{periphery}$ respectively. For $f_{proof,cryptography,undecidability}$ this effect is much weaker as we see a very small decrease from 0.507 to 0.505, from 0.472 to 0.469 and from 0.492 to

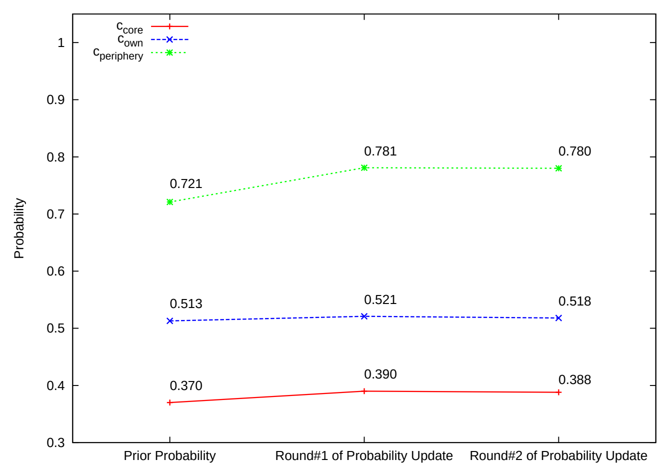

Figure 3. Updating the probability of feature $f_{web,algorithm,program}$

0.491. Figure 3 shows the behavior of our method regarding $f_{web,algorithm,program}$ when it updates probabilities in two rounds of learning from two distinct queried objects.

C. Assessing the Newly-Formed Concept Efficiency

An important question to evaluate our concept is how efficient it is to have an agent learn a new concept. To assess the efficiency of the newly formed concept we conduct another experiment to see how the concept with newly formed probabilities could classify the other objects in the environment. First we collected some positive and negative examples that can be seen by our agent $\mathcal{A}g$. This assures us that agent will be tested against every probable object in the world. To make it clear, we should mention that we repeated our experiment with different percent of objects

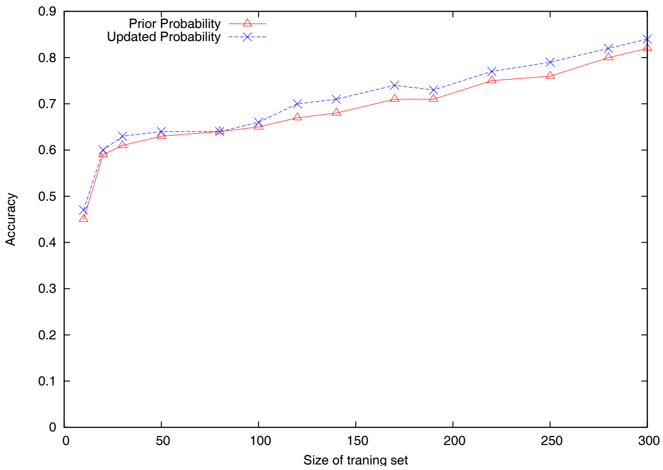

Figure 4. Two different set of features and their classification accuracy

in c_{own} of the *Computer Science* non-unanimous concept. That means for each subset of objects we calculated the probabilities as we stated in Section III and classified the test objects using this subset. In the next round we added some more objects and did the process again. After presenting a new object we updated the probabilities as it is discussed in Section IV. Then we ran the same process of classification. The result is shown in Figure 4. Since there is not really a total agreement on what courses constitute *Computer Science* (beyond a certain strong core, naturally) there is no possibility to achieve 100 percent accuracy. The picture shows a slight increase in the classification accuracy of the $\mathcal{A}g$ when its probabilities updated as it is seen in round #2 of Table II. That is because the reformation of the concept c_k is toward classifying better objects regarding the "Semantic Web" which is quite likely.

VI. Related Works

The task of inducing a concept hierarchy in an incremental manner is known as incremental concept formation or simply concept formation and is a fundamental process of human learning. In this article we presented and analyzed incremental algorithms for updating agents' understanding of a concept. The main distinguishing characteristics of this method with respect to other well-known concept formation methods such as UNIMEM [11], COBWEB [13] and CLASSIT [8] are that the concept here is non-unanimous and in fact the process is updating the feature probabilities rather than learning a concept from scratch. The resulting concept does not depend on the order in which the instances are acquired. A good overview of work on concept formation is found in [5].

On the side of ontologies, there are quite some works that look into so-called *fuzzy ontologies*. But as the name suggests, they see ontologies as fuzzy sets, i.e. sets where membership in the set is not a yes-no decision, but allows different degrees. This is, for example, stated in [15]. [14] sees a fuzzy ontology as a fuzzy formal context in which each concept has a fuzzy relation with other concepts. We do not see how these can incrementally evolve the non-unanimous concepts.

VII. Conclusion

In this paper we provided a new definition for non-unanimous concepts based on feature values and their regarding probabilities. We also represented a new method of incremental non-unanimous concept reformation. Our approach tries to shape any concept in its form of trio non-unanimity by evolving the concept upon facing with new observations or information. The modifications are to affect the ontology of agents in a multi agent system. This method was tested on a popular ontological application domain. Examples of implementing the approach on the problem domain and its effect on updating the probabilities are provided. At the end, an accuracy comparison is presented to access the efficiency of the newly formed concept and to evaluate the whole approach.

References

[1] M. Afsharchi, B.H. Far, J. Denzinger: Ontology-Guided Learning to Improve Communication between Groups of Agents, Proc. AAMAS 2006, Hakodate, 2006, pp. 923–930.

[2] M. Afsharchi, B.H. Far, J. Denzinger: Learning Non-Unanimous Ontology Concepts to Communicate with Groups of Agents, Proc. IAT 2006, Hong Kong, 2006, pp. 211–217.

[3] J. van Diggelen, R.J. Beun, F. Dignum, R.M. van Eijk, J.-J.Ch. Meyer: ANEMONE: An Effective Minimal Ontology Negotiation Environment, Proc. AAMAS 2006, Hakodate, 2006, pp. 899–906.

[4] Illinois Semantic Integration Archive. http://anhai.cs.uiuc.edu/archive/, as seen on Jan 30, 2005.

[5] D.H.Fisher,M.J. Pazzani and P. Langley,(Ed.): Concept Formation: Knowledge and Experience in Unsupervised Learning. San Mateo, CA, (1991,) Morgan Kaufmann.

[6] S. Sen, P.P. Kar: Sharing a concept, AAAI Tech Report SS-02-02, Stanford, 2002.

[7] G. Stumme: Using Ontologies and Formal Concept Analysis for Organizing Business Knowledge, in Becker, Knackstedt (eds.): Wissensmanagement mit Referenzmodellen – Konzepte für die Anwendungssystem- und Organisationsgestaltung, Physica, 2002, pp. 163–174.

[8] J.H. Gennari,P. Langley,D. Fisher: Models of Incremental Concept Formation. In J. Carbonell (Eds.), Machine Learning: Paradigms and Methods, 1990, (pp. 11-62). Amsterdam, The Netherlands: MIT Press.

[9] University of Michigan academic units. http://www.umich.edu/units.html, as seen on Jan 30, 2005.

[10] A.B. Williams: Learning to Share Meaning in a Multi Agent System, JAAMAS 8(2), 2004, pp. 165–193.

[11] M. Lebowitz: Experiments with Incremental Concept Formation: UNIMEM. Machine Learning, 2,1987, 103-138

[12] Y. Yang and J. O. Pedersen: A comparative study on feature selection in text categorization, Proceedings of 14th International Conference on Machine Learning (ICML-97),412–420,1997

[13] D. Fisher: Knowledge Acquisition via Incremental Conceptual Clustering. Machine Learning, 2, (1987), 139-172

[14] Q.T. Tho, S.C. Hui, A.C.M. Fong, T.H. Cao: Automatic Fuzzy Ontology Generation for Semantic Web, IEEE Trans. Knowl. Data Eng. 18(6), 2006, pp. 842–856.

[15] C.S. Lee, Z.W. Jian, L.K. Huang: A fuzzy ontology and its application to news summarization, IEEE Trans. Systems, Man, Cybernetics, Part B 35(5), 2005, pp. 859–880.

[16] T. M. Mitchell: Machine Learning, McGraw-Hill Science/Engineering/Math; 1997; Chap.6, P.182.

[17] T. R. Gruber: The role of common ontology in achieving sharable, reusable knowledge bases. Proceedings of the Second Iniernanonal Conference on Principles of Knowledge Representation and Reasoning, pages 601–602, 1991.