Clouds: A Module for Automatic Learning of Concept Maps

Francisco Câmara Pereira and Amílcar Cardoso

{camara, amilcar}@dei.uc.pt
Centro de Informática e Sistemas da Universidade de Coimbra (CISUC)
Polo II – Pinhal de Marrocos
3030 Coimbra

Abstract. There are currently several interesting works on interactive concept map construction. This simple representation of knowledge - the concept maps - is widely accepted as a promising device for helping in complex tasks such as planning and learning. Moreover, several psychologists (mainly from the constructivist stream) argue that the use of concept maps in teaching can bring relevant improvements in students. Nevertheless, as far as we know, these tools for interactive construction of concept map diagrams have a passive role in the sense that their main concerns rely upon interface and generality. If a Machine Learning based module was added to such frameworks, the computer could have an active role in participating in the concept map construction.

This paper presents Clouds, a module that uses Inductive Learning methods to help a user build her own concept maps. It uses each new entry on the map as an input for the learning algorithms, which can be used later for suggesting new concepts and relations.

1. Introduction

A concept map is a very simple diagrammatic representation of the way concepts relate to each other in a domain. It consists on a directed graph in which arcs are relations and nodes are concepts. Although elementary, this representation has already been studied and applied in education with success [1]. Given its value and applicability, several commercial concept map construction tools are already available (like Inspiration [2] or Decision Explorer [3]), which are very interesting and diverse programs that help a user organise and represent graphically his/her own concept maps. Their behaviour is nevertheless essentially passive in the sense that there is no initiative in suggesting concepts and relations to the user. If we get a mechanism for the apprehension of the way one relates concepts, then we will be able to predict and have an active role in the interaction.

Integrated in a MSc project [4], we developed a system, named Clouds, that uses Machine Learning techniques to help a user to build concept maps. It uses each new entry (in the form of new relations between concepts) as an input to two Inductive Learning algorithms. Both aim to understand what characterises each relation. Then, as the interaction goes on, it starts applying the learnt knowledge to ask and suggest for new concepts and relations. This is an Artificial Intelligence based module that we believe can be used for support in other applications that depend in structures similar to concept maps.

1. Using Inductive Learning Algorithms in Concept Map Construction

Since the user enters information in the form of relations between concepts (e.g. *eat(monkey, banana), property(sun, yellow)*), it makes sense to focus learning on getting what characterises each relation, both in terms of its arguments and of the context that surrounds it. We use two Inductive Learning algorithms to extract each of these features. The first algorithm program aims at finding, for each relation, for pairs of categories that it typically links (e.g. "trees *typically* produce fruit"). In order to do it, Clouds relies on a taxonomic isa-list to find generalizations and specializations. The algorithm is very simple: each time it receives an observation, it calculates the isa-lists involved and joins the current hypothesis with the new observation. The leftmost intersections of both lists yields a generalization.

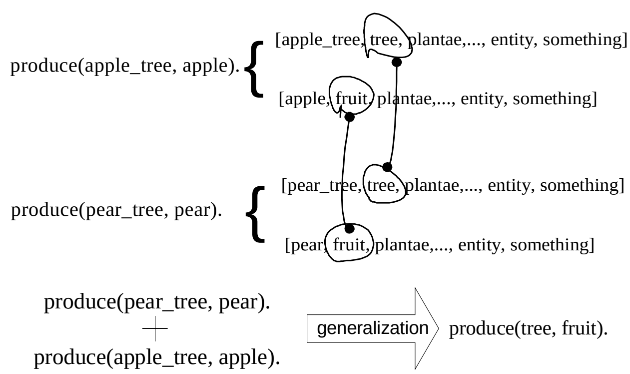

Figure 2 – Intersecting the lists of "produce(pear_tree, pear)" and "produce(apple_tree, apple)", and selecting the leftmost elements (tree and fruit), yields a generalization "produce(tree, fruit)".

The specialization occurs when a negative example is given. In this case, Clouds searches down in the tree for the most general specializations that "avoid" this new observation. The result of this, conversely to generalization, is to split the space into new hypothesis. This divide and conquer results in a number of binary predicates that represents the pairs

of categories of the arguments that cover the positive examples and avoid the negative ones.
The second algorithm is based on Inductive Logic Programming [5]. We implemented a relation explanation generator that concentrates on the context of each argument.

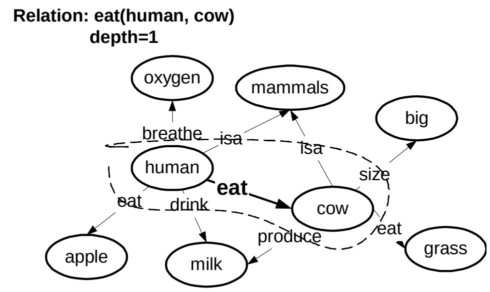

Figure 3 – The relation "eat(human, cow)" and its context

This method enables Clouds to understand each relation in terms of what usually characterizes its context, opening the way for applying deduction and abduction.
The result of this algorithm has the form of Prolog expressions, as in the example shown in fig. 4

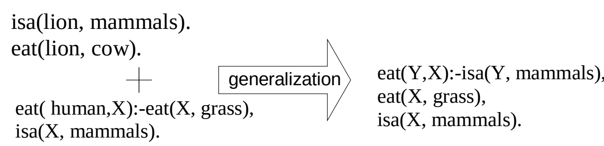

Figure 4 – Generalization of the first argument of eat/2 based on observations "isa(lion, mammals" and "eat(lion, cow)"

We refer the reader to [5] to know more about this algorithm.

References

1. Novak, J.D. and Gowin, D.B.; Learning How To Learn. New York, Camb. University Press. 1984.
2. Inspiraton. Software Developed by *Inspiration Software Inc.* http://www.inspiration.com
3. Decision Explorer. Developed by *Banxia Software*. http://www.banxia.com/demain.html
4. Pereira, F. C. *Construção Interactiva de Mapas Conceptuais*, M.Sc. Dissertation, University of Coimbra, Portugal, 2000
5. S. Muggleton and C. Feng (1990) Efficient Induction of Logic Programs. In Proceedings of the First Conference on Algorithmic Learning Theory, Tokyo, Ohmsa Publishers. Reprinted by Ohmsa Springer Verlag.