MODELLING UNCERTAINTY IN AGENT PROGRAMMING

Johan Kwisthout   Mehdi Dastani

ICS, Utrecht University, P.O.Box 80.089, 3508 TB Utrecht

Abstract

Existing cognitive agent programming languages that are based on the BDI model employ logical representation and reasoning for implementing the beliefs of agents. In these programming languages, the beliefs are assumed to be certain, i.e. an implemented agent can believe a proposition or not. These programming languages fail to capture the underlying uncertainty of the agent's beliefs which is essential for many real world agent applications. We introduce Dempster-Shafer theory as a convenient method to model uncertainty in agent's beliefs.

1 Mapping agent beliefs to Dempster-Shafer sets

In Dempster-Shafer theory[4], a *frame of discernment* Ω is defined as the set of all hypotheses in a certain domain. On the power set 2^Ω, a *mass function* $m(X)$ is defined for every $X \subseteq \Omega$, with $m(X) \geq 0$ and $\sum_{X \subseteq \Omega} m(X) = 1$. If there is no information available with respect to Ω, $m(\Omega) = 1$, and $m(X) = 0$ for every subset of Ω. A *simple support function* is a special case of a mass function, where the evidence supports one set of hypotheses A, and zero mass value is assigned to any subset of Ω other than A. Mass functions can be combined using *Dempster's Rule of Combination*. This combination rule for mass function m_1 and m_2 is denoted as $m_1 \oplus m_2$ and is defined, for $X, Y, Z \subseteq \Omega$, as:

$$m_1 \oplus m_2(X) = \frac{\sum_{Y \cap Z = X} m_1(Y) \cdot m_2(Z)}{\sum_{Y \cap Z \neq \emptyset} m_1(Y) \cdot m_2(Z)} \qquad \textit{and}$$

$$m_1 \oplus m_2(\emptyset) = 0$$

There is an intuitive mapping between agent beliefs, in the form of logical formulae, and Dempster-Shafer sets of hypotheses: a logical formula φ can be represented by a simple support function m_φ, that supports the maximum set of hypotheses that are models of φ, i.e. hypotheses in which φ is true. The extent to which φ is believed to be true can be represented by the mass value of that particular set of hypotheses. A belief base, i.e., a set of logical formulae, can then be represented by the combination of the mass functions that represent the individual belief formulae.

2 Computational complexity

Because n belief formulae lead to a mass function of 2^n combinations, keeping a mass function in memory and updating it when the belief base changes, will lead to a combinatorial explosion in both processing time and memory requirements: Dempsters's Rule of Combination is #P-Complete, as Orponen [3] showed. Wilson [5] has provided a number of algorithms to overcome this problem using for example Monte Carlo methods, and Barnett [1, 2] has shown, that the calculation of the combination is linear if only singleton subsets are used or if the subsets are atomic with respect to the evidence. Both restrictions, however, are problematic if we want to map a logical belief base to a mass function.

We propose an alternative solution to this problem by demanding that only simple support functions are used to model agent beliefs, and that the frame of discernment Ω is such that all combinations of simple support functions related to these beliefs have a model in Ω, i.e., there are no inconsistent beliefs in the belief base. In that case, Dempsters Rule of combination can be simplified to $m_\phi \oplus m_\psi(X) = m_\phi(Y) \cdot m_\psi(Z)$ where $Y \cap Z = X$. Using this simplified rule and given the fact that any mass function only supports one single piece of evidence, it is possible to calculate the mass value of a proposition in linear time, without needing to calculate the combination of all mass functions.

3 Updating and querying the belief base

Since we can derive a mass function from a given belief base, we can just add a belief formula and its probability to this belief base to update the belief base, if the belief base does not yet contain this formula. If the belief base already contains this belief formula, we can update it with a simple combination since Dempster's rule of combination is associative. We can query the belief base by testing if a certain proposition can be derived from the mass function that represents the belief base. For this, we use the Belief and Plausibility functions as defined in [4]. The former denotes the total mass assigned to a set of hypotheses and its subsets while the latter denotes the total mass that is *not* assigned to the complement of that set. Informally, these functions can be seen as a lower and upper limit on the probability of a certain proposition.

4 Conclusion

Particulary appealing in Dempster-Shafer theory is the ability to model *ignorance* as well as uncertainty. Nevertheless, the combinatorial explosion of the combination rule is a serious disadvantage of this theory for practical applications like agent programs. We have investigated a possible mapping of Dempster-Shafer sets to a belief base, which is represented by logical formulae in an agent programming language. With restrictions on the mass functions and on the frame of discernment, Dempster-Shafer theory is a convenient way to model uncertainty in agent beliefs such that computational complexity can be controlled.

References

[1] J.A. Barnett. Computational methods for a mathematical theory of evidence. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 868–875, 1981.

[2] J.A. Barnett. Calculating Dempster-Shafer plausibility. *IEEE transactions on pattern analysis and machine intelligence*, 13:599–603, 1991.

[3] P. Orponen. Dempster's rule of combination is # p-complete. *Artificial Intelligence*, 44:245–253, 1990.

[4] G. Shafer. *A mathematical theory of evidence.* Princeton Univ. Press, Princeton, NJ, 1976.

[5] N. Wilson. Algorithms for dempster-shafer theory. In D.M. Gabbay and P. Smets, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 5: Algorithms, pages 421–475. Kluwer Academic Publishers, 2000.