KEEPER and Protégé: An Elicitation Environment for Bayesian Inference Tools

Mike Pool, Jeffrey Aikin

Information Extraction and Transport, Inc.
1911 North Fort Myer Dr., Suite 600
Arlington, VA 22209
{mpool,jaikin}@iet.com

ABSTRACT

In this presentation we will discuss the role of Protégé in a system designed for eliciting and reasoning with probabilistic models. Information Extraction and Transport, Inc. (IET) is developing the Knowledge Elicitation Environment for Probabilistic Event and Entity Relation (KEEPER) system, a tool for eliciting, storing, updating and implementing probabilistic relational models.[1] A key feature of this tool, and focus of this presentation, is the KEEPER's ability to elicit probabilistic relational models (PRMs or RPMs) from different sources including subject matter experts. The KEEPER elicitation component implements a single ontology for purposes of constraining and guiding elicitation and for purposes of providing the semantic bedrock for the reintegration of diverse sources and learning from diverse sources. This ontology guides and constrains the probabilistic models created by users. The KEEPER system also implements a first-ordering reasoning tool to support querying and learning and to facilitate the implementation of the PRMs in actual data scenarios.

The KEEPER ontology, or tactical modeling language (TML), is stored in Protégé and acts as the domain language within which all KEEPER knowledge is represented. The TML can be extended by users, but all terms used in the KEEPER knowledge base must be defined within the TML. However, in this presentation we focus on our utilization of Protégé as a tool for the elicitation of PRMs and the implementation of those models in IET's suite of uncertainty reasoning tools, Quiddity*Suite.[2]

We will address three distinct issues:

- Implementing Probabilistic relational models in the Protégé environment
- Elicitation extensions required for PRM-specific elicitation extensions
- Communication between Protégé and IET's Quiddity*Suite of probabilistic reasoning tools.

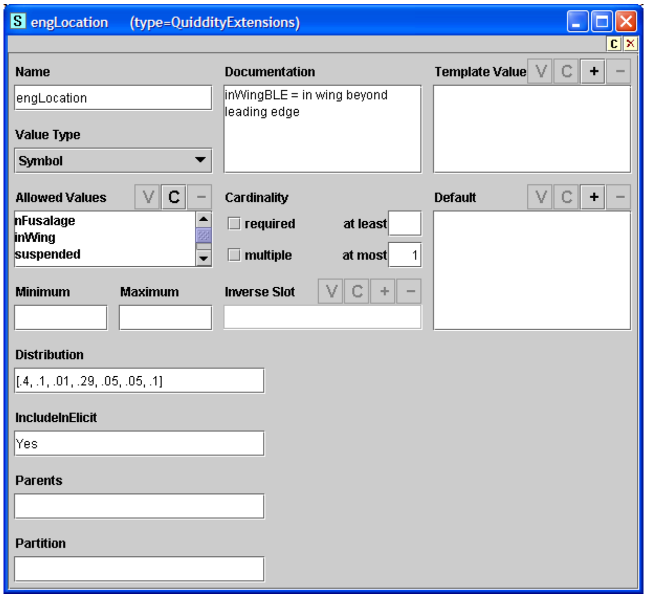
Protégé and PRMs

First, we consider how the Protégé frame-based representation environment lends itself to the representation of PRMs. PRMs are type-level representations that represent how the properties of an entity depend probabilistically on properties of other related entities and explicitly represent the relationships of the entities so involved. The models are at the class or type level and are instantiated for particular sets of entities and the relations between them. Upon instantiation, they encode a Bayesian network and probabilistic reasoning tools can be used to reason about properties of the objects instantiated.

The frame-based representation in Protégé can be adapted to represent the information required for a full PRM. This requires the ability to include relations between relations, i.e., causal links. It also requires the ability to represent "slot chains" that indicate the complex relationships between distinct objects that may be relevant in a particular reasoning situation. For example, we need to able to represent that the property 'gpa' on instances of Class 'Student' are dependent on the value of the 'salary' relation of the person bearing the relation 'professor' to them, i.e., '(parent Student.gpa, professor.salary)'. In addition to such information users must also be able to represent conditional probability distributions and the partitioning of continuous variables. We discuss how we have used metaslots to facilitate the elicitation of information salient to a probabilistic relational model in Protégé. The extensions that we have created allow users to represent PRMs in Protégé. Furthermore, the TML implemented in Protégé ontology ensures that the PRMs created at different time or by different sources implement the same vocabulary and semantics as those created by other users. This is relevant when attempting to constrain elicitation according to the knowledge representation implemented in a particular data source(s). It is also relevant to attempts to ensure that different models elicited from different SMEs are more or less interoperable.

[1] For definition of PRM (RPM) see Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach", second edition, 2003, pp. 519-21., or L. Getoor, N. Friedman, D. Koller, A. Pfeffer, "Learning Probabilistic Relational Models", in Proceedings of the 16th IJCAI, pp. 1300-1307, Stockholm, Sweden, 1999, Morgan Kaufman.

[2] See www.quiddity.com.


engLocation (type=QuiddityExtensions)

Name: engLocation

Value Type: Symbol

Documentation: inWingBLE = in wing beyond leading edge

Template Value

Allowed Values:
nFusalage
inWing
suspended

Cardinality
□ required at least
□ multiple at most 1

Default

Minimum

Maximum

Inverse Slot

Distribution: [.4, .1, .01, .29, .05, .05, .1]

IncludeInElicit: Yes

Parents

Partition

PRM Elicitation Supplements

While the Protégé tool provides an excellent interface for creating and extending ontologies, and while the Protégé metaslots allow us to extend Protégé to create and store probabilistic relational models in Q*M, some of the essential components of a PRM are difficult to elicit from users without the aid of specialized elicitation tools.

For example, users may want to adjust the domain or the range of value types of a particular relation (slot) for a particular PRM or be able to indicate a particular sort of partitioning of the space. While the "age" relation/slot on the class "Person" may be of type 'integer' or 'float', a user creating an advertising PRM may find it useful to partition the values according to relevant demographics. Also, users may require a means to be more or less specific about the relevant possible values of a relation. While a particular variable may have n possible values, only m (m < n) may be relevant to the user's PRM and it should be possible to reduce the complexity of the resultant PRM by grouping the other m-n values under the general value "Other".

Furthermore, the possible number of permutation and combinations of possible causal relations in any PRM involving more than a small handful of relations is suggestive of the need for a graphical interface in which users can easily specify all causal links between relations of interest. This is complicated by the fact that users must not only specify the relations between which causal links should exist, but also must specify the relevant relations between the objects holding those values. So, for example, if the age of a person influences the kind of car s/he is likely to buy, it does not suffice for the user to simply specify a link between Car.type and Person.age, they must also specify that that link exists between a given person and a given car if and only if the person in question bears the relation "owner" to the car in question. Furthermore, a large number of causal relations in any PRM require a tool to generate the associated conditional probability tables. We will discuss these challenges and the tools used to meet these elicitation requirements as well as the means by which they are integrated with Protégé and KEEPER.

Reasoning with PRMs

Suppose that we use our elicitation system to create two distinct PRMs. The first is a general PRM associating ages of persons with the kinds of vehicles that they own while the second associates car types with likelihoods of different kinds of vehicle malfunctions.

Using these PRMs in a reasoning environment requires passing the PRMs from Protégé into IET's own PRM syntax, i.e., Quiddity*Modeler, instantiating them according to the requirements of actual reasoning scenarios and using IET's probabilistic inference tools. Integration within the Quiddity environment enables implementation of IET's Java Symbolic probabilistic inference algorithms[3] as well as IET's tool for execution and hypothesis management techniques. So, for example, if we have an instance of person with age '32' and some car owned by that person, we can instantiate our first PRM and our reasoning tools will thereby allow us to draw conclusions about the type of the car. There are two means for implementing this migration at present, one is direct translation from the Protégé and a second is from an intermediary knowledge repository. We discuss the translation directly from Protégé to the Q*M and the translation to the knowledge repository.

In addition to using single PRMs directly, we also consider the challenge of determining how and when to use and integrate sets of distinct PRMs stored in our library of PRMs. The Protégé-based TML is used to index the resulting library of PRMs and reasoning chains across different PRMs. We show how we are using this indexing to help us identify the salient PRMs when we are attempting to reason from one property about which we have evidence, e.g., a person's age, in one PRM while requiring information about a property in another PRM, e.g., the likelihood that the person's car will have a transmission failure. Our ontology allows us to reason across sets of PRMs to identify subsets relevant to a particular reasoning situation.

[3] See Li, Z. and Bruce D'Ambrosio, "Efficient inference in Bayes networks as a combinatorial optimization problem", *International Journal of Approximate Reasoning*, Vol.11, 1994, pp 55-81. and Takikawa, M. and B. D'Ambrosio, "Multiplicative factorization of noisy-max", *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, 1999, pp. 622-630.