A Framework for Ontologically-Grounded Probabilistic Matching [??]

Rita Sharma [a,*] David Poole [b] Clinton Smyth [a]

[a] *Georeference Online Ltd.*

[b] *Department of Computer Science, University of British Columbia*

Abstract

In all scientific disciplines there are multiple competing and complementary theories that have been, and are being, developed. There are also observational data about which the theories can potentially make predictions. To enable semantic inter-operation between the data and the theories, we need ontologies to define the vocabulary used in them. For example, in the domain of minerals exploration, research geologists spend careers developing models of where to find particular minerals. Similarly, geological surveys publish geological descriptions of their jurisdictions as well as instances of mineral occurrences. The community is starting to develop standardized ontologies to enable consistent use of vocabulary and the semantic inter-operation between the model descriptions and the instance descriptions. This paper describes a framework for representing instances and theories using these ontologies, and describes ontologically-mediated probabilistic matching between instances and theories. We give an example of our matcher in the geology domain, where the problem is to determine what minerals can be expected at a location, or which locations may be expected to contain particular minerals. This is challenging as models and instances are built asynchronously, and they are described in terms of individuals and properties at varied levels of abstraction and detail. This paper shows, given a model, an instance, and a role assignment that specifies which individuals correspond to each other, how to construct a Bayesian network that can compute the probability that the instance matches the model.

Key words:
Probabilistic reasoning, ontologies, Bayesian networks, scientific theories, models, instances, individuals, relational models

* Corresponding Author.

Email addresses: rsharma@cs.ubc.ca (Rita Sharma), poole@cs.ubc.ca (David Poole), cpsmyth@msn.com (Clinton Smyth).

URLs: http://www.cs.ubc.ca/spider/rsharma/ (Rita Sharma), http://www.cs.ubc.ca/spider/poole/ (David Poole), http://www.georeferenceonline.com/ (Clinton Smyth).

1 Introduction

We are interested in decision making and probabilistic reasoning in complex scientific domains [Poole et al., 2008] in which both scientific theories (or hypotheses) and data pertinent to them are available in computer-readable form. We want to make probabilistic predictions in these domains [Schumm, 1991; Jaynes, 2003; Howson and Urbach, 2006] and incorporate rich ontologies [Fox et al., 2006] to allow for semantic interoperability between the theories and the data about which they make predictions. Semantic interoperability between theories and data is a prerequisite for making predictions from theories based on collections of data. For computer-based systems, it also makes possible the provision, in human-readable form, of explanations of conclusions reached by the computer system.

This paper shows how we combine probabilities and ontologies with these goals in a pragmatic way. We want to use the available relevant data and represent the sort of theories that scientists publish, building on ontologies that are being developed by the scientific communities. We have built applications to make predictions about where to best search for particular minerals or where different sorts of landslides are more likely to occur. These systems contain multiple models that make predictions that can be tested against existing data. These predictions can be, for example, a basis of deciding where further exploration is required or an input to the computation of insurance premiums, in the case of landslides. The domains we consider in this paper are characterized by having multiple individuals that are described at various levels of abstraction and detail.

There are many examples of Bayesian approaches to the geological domains [Demoulin and Chung, 2007; Chung and Fabbri, 2005]. Fuzzy sets [Dewitte and Demoulin, 2008] have also been broadly applied to geological domains and other domains addressed by geographic information systems [Robinson et al., 2003]. All these applications take as values for some, if not all, of the parameters used in the Bayesian or Fuzzy calculations, words from scientific classification systems or taxonomies, for example, rock or soil type and geomorphological class. However, what, exactly, is meant by these words, and the relationships that may exist between them (as, for example, in the sub-class relationship between "igneous rock" and "granite") is ignored by these calculations, and may be a significant source of error in the results they produce. The often-hidden complexity inherent in these classification systems is well-described by Arnold [2006] with respect to soil classification systems.

Modelling uncertainty and the use of ontologies have both been recognized as important, but seem to work in different realms. For example, in the 2004 review book "Geographic Information Science and Mountain Geomorphology" [Bishop and Schroder, 2004], there is a chapter entitled "A science of topography: From qualitative ontology to digital representations" by Mark and Smith and another

chapter entitled "Artificial Intelligence in the study of mountain landscapes" by Moody and Katz. The former talks about ontologies, but not uncertainty. The latter, while it reviews fuzzy sets, neural networks, genetic algorithms and other techniques applied to practical geomorphological problem-solving, makes no mention of the ontological framework offered in the aforementioned chapter, and how this framework allows for more rigorous management of the classification language inputs to the techniques reviewed.

There is body of work in combining ontologies and uncertainty, such as PR-OWL [da Costa et al., 2005], OntoBayes [Yang and Calmet, 2005], BayesOWL [Ding et al., 2006] and P-SHIF(D) and P-SHOIN-D [Lukasiewicz, 2008]. All of these add uncertainty to the ontology. For our domain, we think that the appropriate methodology is to use the standard (non-probabilistic) ontologies that are currently being developed, and allow the theories that use the ontologies to make probabilistic predictions. This is for two reasons. Pragmatically, developing ontologies is difficult, and we want to be able to use standard ontologies and inter-operate with the data sets that refer to these ontologies. We also think there is a fundamental difference between definitions of vocabulary—that do not make empirical predictions—and theories that make empirical predictions. The former we call ontologies, and the latter theories. See [Poole et al., 2008] for more discussion of these issues.

A consequence of our approach is that probabilities (for theory property values) and truth status values (true or false flags for instance property values) are additional data elements carried within all our descriptions. How we work with these data elements is a major focus of this paper.

This paper, and the system it describes, is part of our work to bridge the gap, highlighted above, that appears to exist between Bayesian approaches to problem-solving in the earth sciences and the ontological frameworks that are being built in these sciences to assist with managing the complexities of their science languages.

There are three main components of our system:

- An ontology that defines the vocabulary used.
- A set of instances, which are descriptions of things in the world. For example, an instance could be a particular rock outcrop, a volcano that used to exist, or apartment #103 at 555 Short St. These have properties, but also have relationships to other individuals that are important to know about.
- A set of models (or theories) that make probabilistic predictions. For example, we may have a model of what rocks are likely to contain gold, a model of where landslides may occur, or a model of an apartment that Sue would be content to live in.

Intuitively, the models correspond to (conditional) probability statements, and the instances correspond to observations that are conditioned on. The instances and models are described in terms of individuals (objects, things) and their properties.

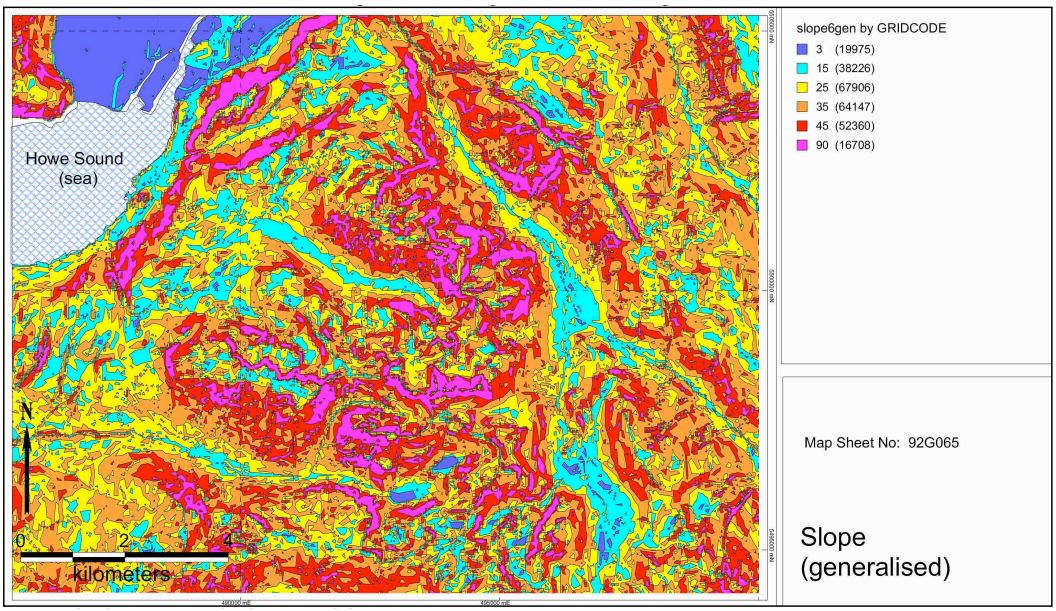

Fig. 1. Slope input to the landslide application

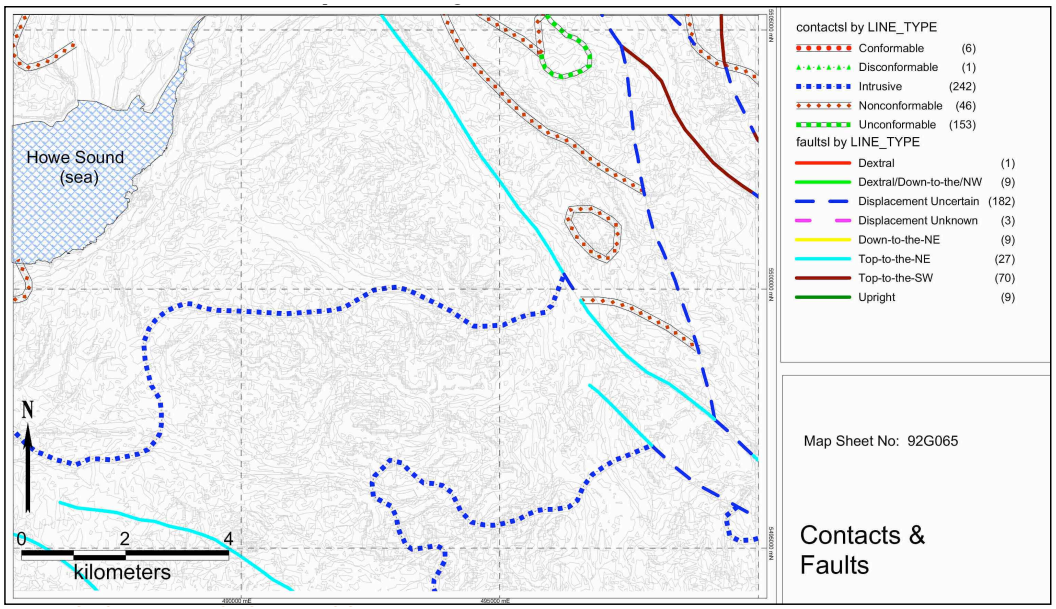

Fig. 2. Structure input to the landslide application

The ontology is used to define the vocabulary so that the terminology can be used consistently in the models and the instances. We consider models as scientific theories that make probabilistic predictions and can be tested according to how well they fit the data. We use the terms "theory" and "model" interchangeably. There can be multiple competing and complementary theories that use the same vocabulary.

Example 1.1 One of the applications we are building, HazardMatch, is for predicting landslide susceptibility [Jackson, Jr. et al., 2008]. Some of the inputs to the system are shown in Figures 1 and 2. Figure 1 shows the slope information for a

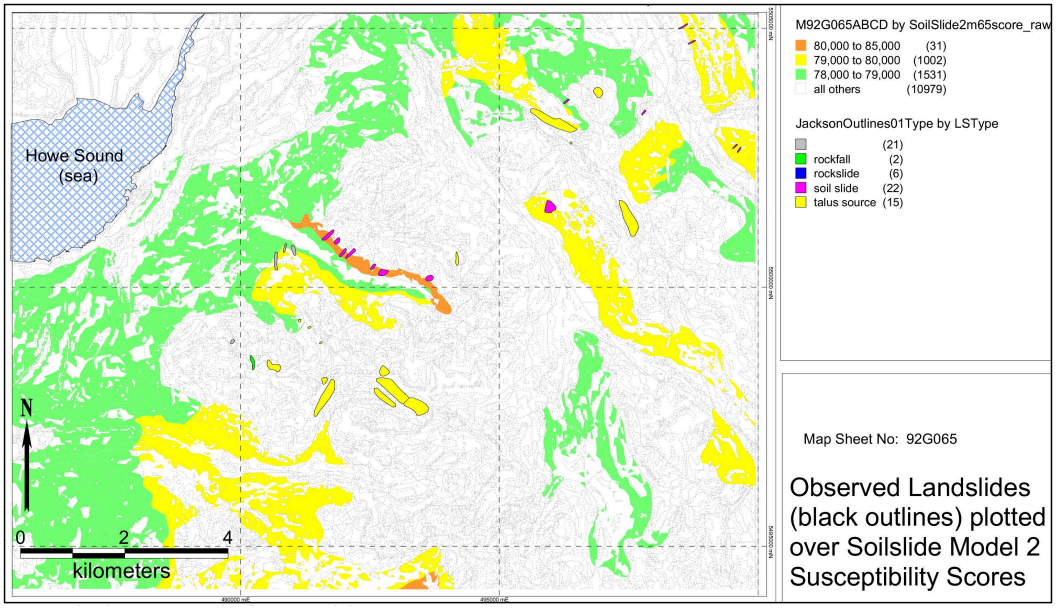

Fig. 3. The output of one of the models matched

part of British Columbia near the Sea to Sky Highway. Figure 2 shows the contacts and faults for the same area.

Each polygon in the maps is an "instance". Each instance is represented in terms of individuals; here the individuals include the surface soil, underlying bedrock, contacts, faults, rivers, etc. These individuals can have properties, that include the relationship to other individuals, e.g., the existence and type of surrounding faults.

We also represent models of different types of landslides. These are created from the literature on landslides and critiqued by experts. In HazardMatch, we work with tens of thousands of spatial instances (polygons) described using standard taxonomies of environmental modelling such as rock type, geomorphology and geological age. To date, we have worked with approximately ten models of landslide hazards which we compare with the spatial instances.

Figure 3 shows the output of one of the models ("Soilslide Model 2") matched against the instances for the same area as Figures 1 and 2. A soil slide is a kind of landslide. The colored regions are the predictions of the soil slide model (in a log-probability scale). The black-outlined regions are the observed soil slides. There are similar predictions from the other models.

This output is for a previous version of our matcher [Poole and Smyth, 2005] which used the likelihood of the model for each instance. The models used a coarse 5-valued scale (always, usually, sometimes, rarely, never), and the predictions were based on the kappa-calculus [Spohn, 1988; Pearl, 1989; Darwiche and Goldszmidt, 1994]. In our work on evaluating and refining the system, we found two main problems. First, the inputs could not be tuned with sufficient clarity. Second, the models

did not take prior information into account. In particular, users often thought that some features should be made more important as they are more diagnostic. How diagnostic a feature is depends on its prior probability, so we wanted to make prior information explicit. To solve these problems, we had to base the system on more rigorous foundations. We wanted to do this while keeping the system manageable: being able to explain the system to the user is a major criteria for acceptance of the system.

Example 1.2 In another application for modelling mineral occurrences, MineMatch describes more than 25,000 instances of mineral occurrences using various taxonomies, including the British Geological Survey Rock Classification scheme [1] and the Micronex taxonomy of Minerals [2] . We also work with more than 100 deposit type models, including those described by the US Geological Survey [3] and the British Columbia Geological Survey [4] . We treat these as probabilistic models even though they are stated in qualitative terms.

Note that we are not considering the problem of taxonomy alignment; the ontologies we use are (designed to be) about disjoint sets of concepts, and inter-operate without confusion.

Using published models to make predictions for particular locations is challenging for a number of reasons, including:

- The models and the instances are described by different people at various levels of abstraction (using more or less general terms) and detail (in terms of parts and sub-parts or holistically). Descriptions of mineral deposits or geological regions are recorded at varied levels of abstraction and detail because some areas have been explored in more detail than others, and the people describing the instances have different backgrounds and goals. There are some models that people spend careers developing, described in great detail for those parts that the modeler cares about. Other models are less well developed, and described only in general terms. Because the instance and model descriptions are generated by different people according to their needs, the levels of abstraction and detail cannot be expected to match. We do, however, need to make decisions based on all of the information available.
- The ontologies used to define the vocabulary for the models and the instances are large and under development. We cannot wait until the ontologies have stabilized to start using them, particularly as the use of the ontologies suggests where they need to be improved.
- The models are positive, in that people only specify positive models of the phenomenon being modeled, not negative models. For example, people publish mod-

[1] http://www.bgs.ac.uk/bgsrcs/
[2] http://micronex.golinfo.com
[3] http://minerals.cr.usgs.gov/team/depmod.html
[4] http://www.em.gov.bc.ca/Mining/Geolsurv/

els of where gold can be found, but do not give models of where gold cannot be found. The models are not exhaustive, in that they define probabilities for a limited number, but not all contexts. For example, a model of where to find gold does not specify whether gold is expected when the conditions of the model are false. There may be contexts where no models are applicable.
- The models are neither disjoint nor covering. Often the models are refinements of each other, and they do not cover all of the cases.

We expect many other domains to have these characteristics.

There are two tasks that we consider in this paper:

- given an instance, determine which models best fits it. This would be used, for example, by someone who has the mineral rights on a piece of land and wants to know what mineral deposits may be there, based on the description of the property.
- given a model, determine which instances best match the model. This would be used by someone who has a model of where, say, gold can be found, and wants to find which of many pieces of land is most likely to contain gold, based on this model.

MineMatch is similar in its goals to the Prospector expert system [Hart, 1975], but builds on the developments in probabilistic reasoning and ontologies of the last 30 years. In previous work [Smyth and Poole, 2004; Poole and Smyth, 2005], we described models using qualitative probabilities, based on the kappa calculus [Spohn, 1988; Pearl, 1989; Darwiche and Goldszmidt, 1994], which measures uncertainty in degree of "surprise". This work was extended by Lukasiewicz and Schellhase [2007] to allow for conditional dependencies. In this paper, we develop an approach based on probability for making decisions. An earlier version of this paper [Sharma et al., 2007], which glosses over many of the details discussed in this paper, discussed how to construct a Bayesian network dynamically during the matching process for computing the posterior probability of a match.

This paper takes a different perspective on combining ontology and probabilistic reasoning than many other recent proposals. Koller et al. [1997] propose giving probabilities over class relationships in a description logic. They do not consider relations amongst individuals that are described using the ontology of the description logic. Ding and Peng [2004] proposed an extension to OWL for representing particular Bayesian networks. They provide a means of translating an ontology implementing the set constructors of OWL into a Bayesian network and are concerned explicitly with set or class membership rather than relationships between attributes. da Costa et al. [2005] proposed a probabilistic ontology language PR-OWL to augment standard ontologies with probabilistic information about the domain. The probabilistic information includes structural information such as conditional independence as well as numerical information such as error rates or sensor

error. PR-OWL is used to model the uncertainty in the observations (e.g., sensor error, error rates), which we are not currently modelling in our framework.

We do not assume that the ontologies include uncertainty about properties and relations. Ontologies are created and maintained by communities, which (hopefully) can agree on vocabulary. However, communities should not agree on probabilities, as the (posterior) probability depends on the prior and the data. People may have different priors and, even if they have access to the same data, the data grows as time progresses. The ontology should have a longer life than one data set; we don't want to update an ontology after each new dataset, and then be required to map between these different ontologies.

We use standardized representations whenever appropriate. In particular, we use OWL as the standard representation for ontologies. We use our own representations for instances and models, as there are no standard languages that are adequate for expressing what we need to (our statements could be reified into RDF, but it would not make the paper more readable).

2 Inputs

The inputs to our matcher are ontologies, instances, models, and what we call supermodels. We describe each in turn.

2.1 Ontologies

In philosophy, *ontology* is the study of what exists. In AI, an *ontology* [Smith, 2003] is a specification of the meaning of the symbols (or of the data) in an information system. We assume that the ontologies are represented in formal representation systems that can be processed by computers.

We adopt OWL [McGuinness and van Harmelen, 2004] to represent ontologies. OWL represents the world in terms of individuals, classes and properties. OWL is built on the Resource Description Framework (RDF) [Manola and Miller, 2004], a language for individual-property-value triples. In this paper, we write a triple using a standard mathematical notation: $\langle i, p, v \rangle$ represents that individual i has value v on property p. RDF represents such triples in XML.

The *owl:Thing* class is a pre-defined class that is the most general class in OWL; everything, whether it is an individual, class or property is an element of *owl:Thing*. The *owl:Class* is the set of all OWL classes; its elements are the classes. Things in OWL have types. The type of a thing is expressed using the *rdf:type* property. The triple:

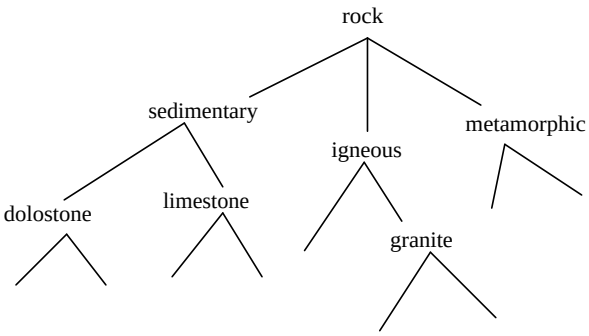

Fig. 4. Part of a taxonomic hierarchy of rock types.

$\langle i, \text{rdf:type}, c \rangle$

means individual i is in class c, where c is an *owl:Class*.

Properties define relationships between individuals and values. The values are either other individuals or datatype values. Each property has a domain and a range. Class d is the domain of property p means that any individual with property p must be of type d. Class r is the range of p means that all values of property p must be members of class r. The range of a property could be a primitive datatype or a class. In OWL, a property is a datatype property if its range is a primitive datatype. Otherwise, it is an object property. We will not consider datatype properties in the rest of this paper, as each datatype becomes a special case that will complicate the discussion.

We distinguish enumerated individuals, which are those individuals that need to be shared between models or instances, e.g., colours, sizes, minerals, days of the week. We use the class *EnumClass* to be the class of enumerated individuals. Subclasses of *EnumClass* are called *enumerated classes*. Enumerated classes will be used to define the domains of discrete random variables. An object property whose range is an enumerated class, is an *enumerated property*. An object property whose range is not an enumerated class is called an *entity property*.

For this paper, we assume that the classes form a tree structure, a taxonomic hierarchy, where the children of a class (its immediate subclasses) are mutually disjoint. This is achieved by using the *owl:disjointWith* property.

Figure 4 shows an example of a taxonomic hierarchy. An igneous rock is a kind of rock. A granite is a kind of igneous rock. In this figure, *rock* is the topmost class.

Example 2.1 Our example mineral deposit ontology consists of four disjoint enumerated classes: *geneticSetting*, *weatheringDegree*, *colour*, and *age* and three disjoint object classes: *mineralDeposit*, *rock*, and *mineral*. A sub-tree of the British Geological Survey Rock Classification scheme [Gillespie and Styles, 1999] is shown in Figure 4.

Property	***Domain***	***Range***	***other properties***
hasHostRock	*mineralDeposit*	*rock*	
hasGeneticSetting	*mineralDeposit*	*geneticSetting*	*functional*
hasAge	*mineralDeposit* $\cup$ *mineral*	*age*	*functional*
hasMineral	*rock*	*mineral*	
hasWeatheringDeg	*rock*	*weatheringDegree*	*functional*
hasColour	*mineral*	*colour*	*functional*

Fig. 5. Properties in the Mineral Deposit Example

The enumerated classes consist of the following enumerated individuals:

colour = {*clear, white, pink, blue,...*}
geneticSetting = {*greenStoneBelt,oceanRidge,...*}
weatheringDegree = {*weathered,unweathered*}
age = {*proterozoic, archean, palaeozoic, cainozoic,...*}

The actual enumerated classes we use are much more complicated, and have a hierarchical structure. For this paper, we only use the values specified here.

The mineral deposit ontology has two entity properties: *hasHostRock*, *hasMineral*, and four enumerated properties: *hasGeneticSetting*, *hasWeatheringDeg*, *hasAge*, and *hasColour*. The domains, ranges and whether the properties are functional are summarized in the table of Figure 5.

2.2 Instances

An instance is a thing in the world we are reasoning about (the real world at some time, some temporally extended world, or even some imaginary world). In the geological domain, an instance is often a particular location that someone has identified as being interesting. It is important to distinguish an instance from its description. While a description may be at a high level of abstraction, the instance itself isn't.

An instance is described in terms of a set of related individuals (the instance individuals). One of these individuals is the designated top-level individual. For example, in an instance describing a deposit, the individuals are the mineral deposit, its rocks, their minerals etc. The designated top-level individual is the mineral deposit.

An instance individual is described by its values on various properties. This can include its relationship to other individuals (e.g., its parts). We do not only want to state positive facts, but also negative facts such as that a mineral deposit does not

contain a metamorphic rock, or that a mineral is red colour but is not a pink (without enumerating all of the non-pink red colours). We represent instance descriptions with the quadruples of the form:

$\langle$*ind, P, value, truthvalue*$\rangle$

where P is a property, *ind* is an individual in the domain of P, *truthvalue* is either *present* or *absent*, and *value* is an individual in the range of P.

For this paper, we assume that each individual has relationship to exactly one parent individual (each individual is the *value* of only one tuple, and there are no cycles in this relationship between individuals). The top-level individual is not the value of any property. We also assume that the individuals are all distinct.

A description of an instance means a conjunction in first-order logic with limited quantification. Enumerated individuals are logical constants. Object individuals are variables in the translation. We assume an ordering for the tuples, where the tuple with v as a value is before any other tuple containing v. The tuples can then be interpreted as follows:

- A tuple $\langle o, P, v, present \rangle$, where P is an enumerated property, means the atom $P(o, v)$.
- A tuple $\langle o, P, v, absent \rangle$, where P is an enumerated property, means $\neg P(o, v)$.
- A tuple $\langle o, P, v, present \rangle$, where P is an entity property, means $\exists v\ P(o, v) \wedge v \neq i_1 \wedge \cdots \wedge v \neq i_k$, where all following tuples (which, by the ordering assumed, includes those containing v) are in the scope of the existential quantification, and $i_1 \ldots i_k$ are the previously defined individuals (the top-level individual and those variables whose scope this is in).
- The tuple $\langle o, P, v, absent \rangle$, where P is an entity property, means $\neg\exists v\ P(o, v) \wedge v \neq i_1 \wedge \cdots \wedge v \neq i_k$, where the other tuples containing v are in the scope of the existential quantification and $i_1 \ldots i_k$ are the previously defined individuals.

The only free variable in the translation is the top-level individual.

Example 2.2 To say that mineral deposit *mindep1* has a granitic rock, but does not have a sedimentary rock we could write:

$\langle$*mindep1, hasHostRock, rock1, present*$\rangle$
$\langle$*rock1, rdf:type, granite, present*$\rangle$
$\langle$*mindep1, hasHostRock, rock2, absent*$\rangle$
$\langle$*rock2, rdf:type, sedimentary, present*$\rangle$.

This means the first-order formula:

$$\exists rock1\ hasHostRock(mindep1, rock1)$$
$$\wedge type(rock1, granite) \wedge rock1 \neq mindep1$$
$$\wedge \neg\exists rock2\ (hasHostRock(mindep1, rock2) \wedge rock2 \neq rock1 \wedge rock1 \neq mindep1$$
$$\wedge type(rock2, sedimentary))$$

The first two tuples of this example together specify that mineral deposit *mindep1* has a granitic rock. Note that if tuple ⟨*rock1, rdf:type, granite*,present⟩ is not specified, we can infer that the type of *rock1* is *rock*, which is the range of *hasHostRock*. The last two tuples together specify that there is no sedimentary rock in *mindep1*. However, it doesn't preclude the existence of a second rock as long as it is not a sedimentary rock.

We can use *absent* to specify the number of individuals of particular type, as is shown in the following example.

Example 2.3 To specify that there are at least two rocks in *mindep1*, we explicitly specify the presence of two rocks. We do not specify anything about a third rock. Thus, we can specify the tuples:

⟨*mindep1, hasHostRock, rock3, present*⟩
⟨*rock3, rdf:type, igneous, present*⟩
⟨*mindep1, hasHostRock, rock4, present*⟩
⟨*rock4, rdf:type, sedimentary, present*⟩

To state that there are exactly two igneous rocks in *mindep1*, we state that two igneous rocks exist, but there doesn't exist a third igneous rock. Thus, we write as follows:

⟨*mindep1, hasHostRock, rock3, present*⟩
⟨*rock3, rdf:type, igneous, present*⟩
⟨*mindep1, hasHostRock, rock4, present*⟩
⟨*rock4, rdf:type, igneous, present*⟩
⟨*mindep1, hasHostRock, rock5, absent*⟩
⟨*rock5, rdf:type, igneous, present*⟩

Example 2.4 To state that mineral deposit *mindep1* has an igneous rock *rock3* that is not a granite, we could write:

⟨*mindep1, hasHostRock, rock3, present*⟩
⟨*rock3, rdf:type, igneous, present*⟩
⟨*rock3, rdf:type, granite, absent*⟩

We can draw the tuples as a semantic network [Quillian, 1968]. The nodes represent individuals, classes and data types. The arcs are labelled with properties and truth values. The arcs represent quadruples, labelled with the properties and the

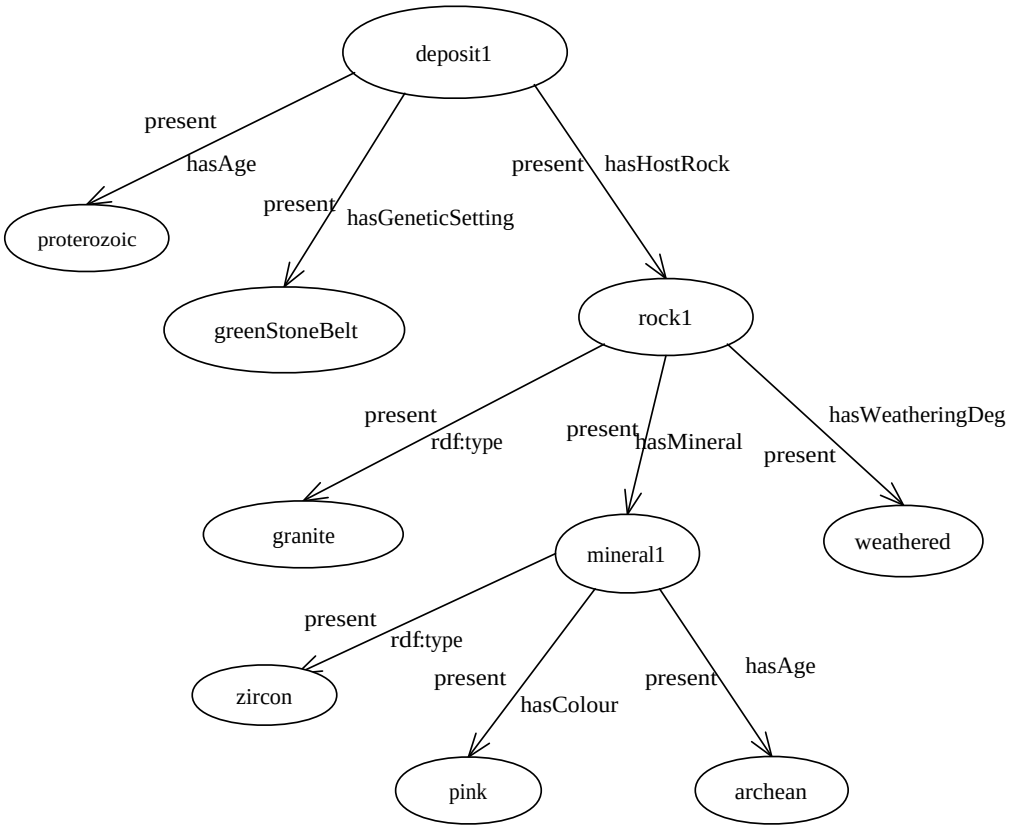

Fig. 6. A Semantic network representation of deposit instance *deposit1*.

truth values. The semantic network representation of deposit instance *deposit1* is shown in Figure 6. The arc connecting individuals *deposit1* and *rock1*, represents quadruple $\langle$*deposit1, hasHostRock, rock1, present*$\rangle$.

2.3 Models

A model specifies a probability distribution over instances. A model can be used to make a prediction about the instance given the evidence provided by the description of the instance.

The standard Bayesian view is that a model is used to determine the probability of a prediction given the evidence. Here the evidence is the description of the instance. Suppose M is a set of mutually exclusive and covering models:

$$
\begin{aligned}
&P(prediction|instance)\\
&= \sum_{model \in M} P(prediction|model, instance) P(model|instance)\\
&= \sum_{model \in M} P(prediction|model, instance) \frac{P(instance|model)P(model)}{P(instance)}\\
&= \sum_{model \in M} P(prediction|model, instance) \frac{P(instance|model)}{P(instance)} P(model)
\end{aligned}
$$

and often we assume that $P(prediction|model, instance) = P(prediction|model)$, i.e., that the prediction is independent of the instance given the model.

Our models are complicated in a number of ways:

- The models that scientists (in our domains, at least) publish are not exclusive and do not cover all of the cases.
- The models that are published are typically not detailed enough to give a prediction for each instance.
- To make a prediction from a model, we need to identify a "match" between the individuals in the model and the individuals in the instance. The models typically specify roles, and we need to identify which individual specified to exist in an instance fills each role in the model (or if there is no individual in the instance to fulfill a role).
- We want an explanation as to why a prediction is reasonable. As people have to act based on the advice of the system, they need to be convinced that the answers produced are reasonable.

The contribution of a model is given by the product of three terms. If any of the terms is close to zero, their product will be close to zero. We approximate the sum above by only considering the top models; those where the product is highest. This means that we only need to consider the models that have a non-trivial prior, predict an item of interest, and predict the instance best. A conclusion based on model averaging is difficult to explain to a user, but if the average is dominated by a few models, a user can understand such an explanation.

The probability of an instance will be the product of a number of terms (using the chain rule of probability, and some independence assumptions). The models that experts give us are only partly specified. Some of the elements of the product will be of the form $P(instance\_feature|model)/P(instance\_feature)$, we assume that any conditional probability that the model does not specify is equal to the prior probability. Thus the probability given the model divided by the prior is 1 for these features, and so doesn't affect the product. We need the prior probabilities for the predictions specified in a model. The priors are traditionally computed by summing over a set of exclusive and covering hypotheses, but, as we don't have a set of exclusive and covering hypotheses, we assume that the prior probabilities are specifies as part of a supermodel (see Section 2.4).

Models describe probabilities of the existence and properties of individuals. Each model has a designated top-level individual that it is nominally about. A model can also refer to other individuals that are related (perhaps indirectly) to the top-level individual.

As an example, deposit model $depModelA$, which we will use as an ongoing example, is a simplified model describing a particular type of deposit. The model predicts that an instance that matched the model likely has the age that is Proterozoic, and contains an igneous rock with zircon in the deposit. Note that the actual models we use are more complicated than this; this model was constructed to show the issues.

A model specifies the probability of property values associated with individuals. For enumerated properties, we have the probability of the different values. For object properties, we need to reason about the probability of the existence of objects. These properties typically specify the role of an object, so we want the probability of the existence of an object that fills a role. Poole [2007] argued that, to satisfy the clarity principle (that all propositions are well defined), a model has to be clear about what doesn't exist when existence is false, and, when there is more than one object that exists, a model has to be clear about which object it is referring to.

A model has two sorts of statements. One is to state the existence of an object in the world. The second is to specify what is expected to be true about an object that exists.

A model is described in terms of quadruples. A quadruple is of the form:

$$\langle ind, P, value, p \rangle$$

where P is a property, ind is an individual in the domain of P, p is a probability, and $value$ is in the range of P. In particular, $value$ is:

- an enumerated individual , if P is an enumerated property
- an object individual, if P is an entity property
- a class, if P is the *rdf:type* property

Similarly to instances, we assume that each individual, apart from the designated top-level individual, appears as the value in exactly one quadruple. We also assume that graph induced by the $\langle ind, value \rangle$ pairs is acyclic. Thus an individual can only be in the model if it is connected to the top-level individual.

Example 2.5 To state that mineral deposit model $depModelA$ has a rock that fills a particular role with probability 0.93, and that the rock that fills the role is definitely an igneous rock, we write:

$\langle$*depModelA, hasHostRock, rockB, 0.93*$\rangle$
$\langle$*rockB , rdf:type, igneous, 1.0* $\rangle$

The probability 0.93 represents the probability of the existence of a rock that fills the role. If it exists we call it *rockB*.

In some situations, we would like to have a probability distribution over which class an individual is a member of. We can use *rdf:type* property to represent the uncertainty over the types of an individual.

Example 2.6 Consider stating that mineral deposit model $depModelA$, with probability 0.77, has a rock that fills a particular role. A rock in that role, when it exists, is of type igneous with probability 0.85. To state this we write:

$\langle$*depModelA, hasHostRock, rockA, 0.77*$\rangle$
$\langle$*rockA , rdf:type, igneous, 0.85*$\rangle$

The probability 0.77 represents the probability of the existence of a rock that fills the role of *rockA*. If such a rock exists, it is of type igneous with probability 0.85. Note that rockA, if it exists, is of type rock, which is the range of hasHostRock, with probability 1.0.

Suppose a model M specifies the tuples

$\langle ind, \text{rdf:type}, C_l, p_1 \rangle$
$\ldots$
$\langle ind, \text{rdf:type}, C_n, p_n \rangle$

for some individual *ind*. Let $\mathscr{C}_{ind} = \{C_l, \ldots, C_n\}$. We say that C_i is a *highest subclass* of C_j in $\mathscr{C}_{ind}$ if $\nexists C_s \in \mathscr{C}_{ind}$ such that $C_i \subset C_s$ and $C_s \subset C_j$, where $\subset$ is the (strict) subclass relation as specified in the ontology. There are three constraints on the probability values $p_1, \ldots, p_n$:

- If $C_i \subseteq C_j$, then $p_i \leq p_j$.
- Suppose $C_1, \ldots, C_k$ are the highest subclasses of C_j then $p_j \geq \sum_{i=1}^{k} p_i$. The constraint holds because of the disjointness of subclasses.
- If $C_1, \ldots, C_k$ are *all* of the immediate subclasses of C_j then $p_j = \sum_{i=1}^{k} p_i$.

A functional enumerated property has only one value for each individual. However, it is possible that we may not know what that value is. A model specifies a probability distribution over the values. Thus, for functional enumerated properties, the model quadruples must follow the constraint: Let P be a functional enumerated property and suppose that the model has k quadruples: $\langle ind, P, val_1, p_1 \rangle$, $\ldots$, $\langle ind, P, val_k, p_k \rangle$. Then, $\sum_{i=1}^{k} p_i \leq 1$. We need $\sum_{i=1}^{k} p_i = 1$ if $\{val_1, \ldots, val_k\}$ is the whole enumerated class. That is, we do not require that models are complete and specify the probabilities for each value of a property, but, if they are complete, the probabilities must sum to 1.

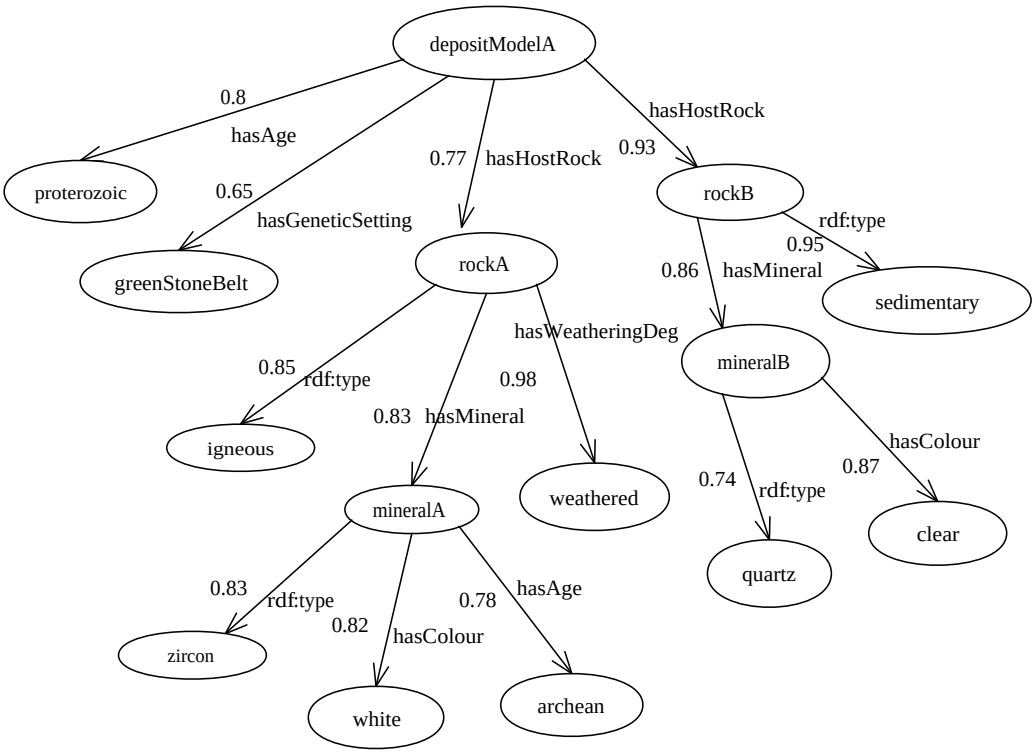

Fig. 7. A Semantic network representation of mineral deposit model *depModelA*.

Example 2.7 A semantic network representation of mineral deposit model *depModelA* is shown in Figure 7. The nodes represent individuals, classes and data types. The arcs represent quadruples and are labelled with properties and probabilities. For example, the arc connecting individuals *depModelA* and *rockA*, represents quadruple $\langle$*depModelA, hasHostRock, rockA, 0.77*$\rangle$.

2.4 Supermodel

The role of the supermodel is to provide prior probabilities. In standard Bayesian reasoning, the prior probability of an observation is computed by summing over all models. However, we do not assume that our models are disjoint or covering, and so need an extra specification of the prior probabilities.

A supermodel provides the following prior (background) information:

- The prior distribution of classes, i.e., the prior probability that an individual is of type C, for any class C
- For each enumerated property P, and for each value $v \in range(P)$, the prior probability that an individual has value v for property P.

Consider the tree hierarchy of the classes in the ontology. To specify the prior distribution of classes, for each class C_k that has children (immediate subclasses), the

supermodel contains $P(C_j|C_k)$ for each child C_j of C_k. The prior probability of the root class in the tree hierarchy is 1. Given these conditional probabilities, we can compute the prior probability of any class (type) in a recursive manner by multiplying the probability of class given its immediate super-class and the probability of its immediate super-class. Thus, for each immediate subclass C_j of C_k

$$P(C_j) = P(C_j|C_k) \times P(C_k)$$

Note that the complexity of computing the probability of C_j, $P(C_j)$, is linear in depth of C_j in the hierarchy and otherwise is not a function of the hierarchy's size.

To specify the prior on properties, for each enumerated property P, with domain cl, for each value $v \in range(P)$, we specify the prior probability that an individual of type cl has value v for property P using quadruples of the form:

$$\langle cl, P, v, p \rangle$$

where p is the prior probability. Note that we add cl as an argument to the quadruple, even though it could be inferred from the property, to allow for future versions that have different priors on subclasses of cl.

If P is a functional enumerated property, where the range of P is the set $\{v_1, \ldots, v_n\}$, there are n quadruples $\langle cl, P, v_1, p_1 \rangle$, $\ldots$, $\langle cl, P, v_n, p_n \rangle$ in the supermodel such that $\sum_i p_i = 1$. Non-functional properties do not have the constraint of summing to 1.

Example 2.8 Consider defining the prior probabilities for the mineral deposit ontology as given in Example 2.1. Some of the probabilities that supermodel specifies are as follows:

In the *Rock* hierarchy, assume we have:

$P(Rock) = 1$
$P(igneous|Rock) = 0.45$
$P(sedimentary|Rock) = 0.2$
$P(granite|igneous) = 0.3$

In the mineral hierarchy, assume we have:

$P(sulphide|mineral) = 0.2$
$P(silicates|mineral) = 0.5$
$P(oxides|mineral) = 0.3$
$P(quartz|silicates) = 0.6$
$P(zircon|silicates) = 0.4$

Assume that the enumerated classes have the following prior distributions [5] :

[5] For the sake of keeping the example simple, we assume that the domains of the properties

$\langle mineralDeposit, hasAge, proterozoic, 0.35\rangle$
$\langle mineralDeposit, hasAge, archean, 0.05\rangle$
$\langle mineralDeposit, hasAge, cainozoic, 0.35\rangle$
$\langle mineralDeposit, hasAge, palaeozoic, 0.25\rangle$

$\langle mineralDeposit, hasGeneticSetting, greenStoneBelt, 0.4\rangle$
$\langle mineralDeposit, hasGeneticSetting, oceanRidge, 0.6\rangle$

$\langle rock, hasWeatheringDeg, weathered, 0.7\rangle$
$\langle rock, hasWeatheringDeg, unweathered, 0.3\rangle$

$\langle mineral, hasColour, white, 0.4\rangle$
$\langle mineral, hasColour, blue, 0.2\rangle$
$\langle mineral, hasColour, clear, 0.1\rangle$
$\langle mineral, hasColour, pink, 0.3\rangle$

$\langle mineral, hasAge, proterozoic, 0.3\rangle$
$\langle mineral, hasAge, archean, 0.2\rangle$
$\langle mineral, hasAge, cainozoic, 0.35\rangle$
$\langle mineral, hasAge, palaeozoic, 0.15\rangle$

2.5 Semantics of Models

A possible world is a complete description of a set of related individuals. It includes all property values for these individuals, at the lowest level of detail. For example, in a possible world, an individual cannot be a rock without being a specific sort of rock.

A particular model and the supermodel defines a probability measure over a set of possible worlds. We only need to specify the measure enough to be able to select the set of possible worlds in which an instance is true. In particular, the possible worlds need to describe the objects that are related to the designated top-level individual of an instance. As described in Section 2.2, the meaning of an instance is a logical formula with a single free variable corresponding to the top-level individual. The measure over possible worlds must be able to give a probability of description with respect to the designated top-level individual.

For this paper, we make strong independence assumptions. The only probabilistic dependencies are those that are entailed by the logical dependency. An object only

in Example 2.1 are complete.

has a property when the object exists. Thus the propositions that specify that an object has a property must be conditioned on the existence of the object.

To look-up the probability associated with a tuple, we first see if it is specified in the model and, if so, use that value, otherwise we use the value from the supermodel. Existence probabilities only come from the model.

We will define the semantics of models in terms of a first-order semantic tree [Poole, 2007], which is like a familiar event tree, but allows for splits on first-order formulae (taking into account the scoping of variables). Each world gets filtered down the tree (to a unique position). The probability at any node is the measure of the worlds that get filtered to that node. We need to define the measure well enough so that the measure of the set of possible worlds that satisfy an instance can be determined. We can stop filtering an instance description at a node if all future splits won't change the truth value of the instance description. As part of the description we allow for equality between model individuals and instance individuals; the model individual is said to match the instance individual. At the top node the top-level individuals match.

We assume a total ordering of the tuples in the model so that the tuple that contains an individual as the value comes before any other tuples that contain the individual.

At each node in the semantic tree, we split on the top-most tuple in the total ordering that is applicable (we will define recursively what is applicable; initially all tuples are applicable). Suppose $\langle ind, P, value, p\rangle$ is the next applicable tuple. There are two cases:

- P is an enumerated property. In this case, $value$ must be an enumerated individual. The set of possible worlds filtered this node are divided into two: those where the $P(ind) = value$ is true and those where it is false. The probability masses of these sets of worlds are divided in the ratio $p : 1 - p$.
- P is an entity property. In this case $value$ must be an object individual. We first split on whether there exists an individual that is in relation P to ind, i.e., whether $\exists value\ P(ind, value)$. This split divides the probability masses of the worlds in the ratio $p : 1 - p$. The tuples that contain $value$ are only applicable in the worlds where existence is true; that is we ignore the other tuples what contain $value$ in the sub-tree where the existence is false. For the branch where the existence is true, we assume that any object in the instance for which $P(ind, value)$ is true is, a priori, equally likely to match $value$. We then split on the equality between $value$ and the corresponding instance variables (with a uniform prior probability).

Thus, for an entity property, out of the worlds where existence is true, each individual that satisfies the existence has equal prior probability of being the match. The probabilities of the match then depends on the how well the instance individual matches the object individual.

We now can define the probability in a standard way: the probability of any instance is the measure of the worlds in which the instance is true. Note that this averages over role assignments: the prior probability of an individual filling a role is equal, but the posterior probability can change.

3 Probabilistic Matching

The matcher is used in two modes:

- In instance-to-models matching, one instance is compared to multiple models. Finding the most likely models for the instance can be used to determine what is the most likely mineral or landslide to be at the particular location described by the instance.
- In model-to-instances matching, one model is compared to multiple instances. This can be used to find the location(s) that are most likely to have landslides or contain particular minerals.

The basic problem is to determine the probability of the instance given the model. The model specifies probabilities over roles of the top-level individual, and its related individuals. To determine the conditional probability, we need to determine which individuals specified to exist in the instance fill the roles specified in the instance. We call this correspondence a *match*. We write $M_i \sim I_j$ to specify the proposition that instance individual I_j fills the role specified by the model individual M_i. We use the same notation to give the match for the top-level individuals, M_t and I_t of M and I respectively. We want to determine the posterior probability of $M_t \sim I_t$ given I's description, M's description, the domain ontology, and the supermodel. This is the probability the top-level individual of the instance fills the role that the model is modeling.

3.1 Computing the probability of a model individual's type

One of the sub-tasks in the algorithm below is to compute the probability that a model individual is in some given class. This requires using probabilities from both the model and the supermodel.

Suppose M_i is an individual in a model M. Class C_n is exceptional for M_i if M contains a tuple $\langle M_i, \text{rdf:type}, C_n, p_n \rangle$. We call C_n an exceptional class when M_i is clear from context.

The probability that M_i is of type C_k, denoted by p_k can be calculated as follows:

- If $\langle M_i, \text{rdf:type}, C_k, p_k \rangle \in M$, return p_k

- else if C_k doesn't have any subclasses that are exceptional for M_i, p_k can be computed from the probability of lowest super-class C_n of C_k that is exceptional for M_i. Let $C_s^1,\ldots,C_s^j$ be the highest subclasses of C_n that are exceptional for M_i. Then, $(p_n - \sum_{C_s^i} p_s^i)$ is the probability that M_i is of type C_n but not of types $C_s^1,\ldots,C_s^j$. Suppose $P(C_n)$ denotes the prior probability that M_i is of type C_n. Then, $\frac{P(C_k)}{P(C_n)-\sum_{C_s^i} P(C_s^i)}$ is the prior probability that M_i is of type C_k, given that M_i is of type C_n but not of types $C_s^1,\ldots,C_s^j$. Then, the probability that M_i is of type C_k, p_k, is:

$$p_k = (p_n - \sum_{C_s^i} p_s^i) \times \frac{P(C_k)}{P(C_n) - \sum_{C_s^i} P(C_s^i)} \tag{1}$$

- otherwise, p_k can be derived, recursively, from the children of C_k in T_r

$$p_k = \sum_{\forall C_i \in children(C_k)} p_i \tag{2}$$

Example 3.1 Suppose we have the supermodel of Example 2.8, and the model of Figure 7. We use the probabilities of the supermodel, as modified by the model. Consider computing the probability that rock *rockA* is of type sedimentary. The model specifies that rock *rockA* is of type igneous with probability 0.85. We use the probabilities of the supermodel, taking into account this constraint. That is, treat the probability of the model as constraints, otherwise using the ratios between probabilities in the supermodel.

Suppose p_{sed} denotes the probability *rockA* is of type *sedimentary*. Then,

$$\begin{aligned} p_{sed} &= (1-0.85) \times \frac{P(sedimentary)}{P(Rock) - P(igneous)} \\ &= (1-0.85) \times \frac{0.2}{(1-0.45)} \\ &= 0.0545 \end{aligned}$$

The rock *rockA* is of type sedimentary with probability 0.0545.

3.2 Role Assignment

To determine the probability of a match between a model and an instance, we need to know which model individuals correspond to which instance individuals. The correspondence between the model individuals and the instance individuals is called a *role assignment*. We use $M_k = I_j$ to mean that model individual M_k corresponds to instance individual I_j and $M_k = \perp$ to mean that model individual M_k does

not corresponds to any instance individual. A role assignment is a list of correspondence statements of the forms $M_k = I_j$, $M_k = \perp$, that we define recursively on the structure of the model, such that:

- $M_t = I_t$, where M_t and I_t are top-level individuals
- if $M_k = I_j$ is in the role assignment (so that, in particular, $M_k \neq \perp$), and M_k has children, the role assignment include assignments of the form $M_c = \perp$ or $M_c = I_c$ where M_c is a child of M_k and I_c is a child of I_j such that:
  - If $M_c = I_c$ then M_c and I_c must be of compatible types.
  - Each child M_c of M_k appears exactly once in the list and each child I_c of I_j appears at most once.
  - $M_c = \perp$ cannot appear if there is a child I_c of I_j that is of a compatible type to M_c and is not assigned to another model individual.

Example 3.2 In matching mineral deposit model $depModelA$ shown in Figure 7 with deposit $deposit1$ shown in Figure 6, there are two legal role assignments:

- $\Re 1 = \{depModelA = deposit1, rockA = rock1, rockB = \perp, mineralA = mineral1\}$
- $\Re 2 = \{depModelA = deposit1, rockA = \perp, rockB = rock1, mineralB = mineral1\}$

3.3 Individual Matching

Intuitively the model individuals represent roles. A particular role assignment specifies which individuals can be considered in the roles, but does not specify how well these individuals fill the roles. If M_k is a model individual and I_j is an instance individual, such that $M_k = I_j$ is in the role assignment, then, $M_k \sim I_j$ represents the proposition that I_j fills the role that M_j represents.

Example 3.3 Consider matching mineral deposit model $depModelA$ as shown in Figure 7 with mineral deposit $deposit1$ as shown in Figure 6, and the role assignment $\{depModelA = deposit1, rockA = rock1, rockB = \perp, mineralA = mineral1\}$. The match $mineralA \sim mineral1$ represents the proposition that $mineral1$ fills the role of $mineralA$. The probability that it fills the role depends on its colour and age. The match $rockA \sim rock1$ represents the proposition that $rock1$ fills the role of $rockA$. The match $depModelA \sim deposit1$ represents the proposition that the role of $depModelA$ is met by $deposit1$. This is the proposition that we want to compute the probability of.

4 Construction of Bayesian network

The high level algorithms for both tasks of the matcher (model-to-instances and instance-to-models matching) are shown in Figures 8 and 9 respectively. For both

Input: O: Ontology, S: Supermodel, M: model, and I_{set}: set of instances
Output: ranking of instances
  $M_t \leftarrow$ the top-level individual of M
  for each $I \in I_{set}$ **do**
    $I_t \leftarrow$ the top-level individual of I
    for each role assignment $\mathfrak{R}$ **do**
      Compute $P(M_t \sim I_t|\mathfrak{R})$
    end for
    Let $p_I = \max_{\mathfrak{R}} P(M_t \sim I_t|\mathfrak{R})$
  end for
  Return $\{\langle I, p_I\rangle : I \in I_{set}\}$ ordered by p_I

Fig. 8. Algorithm for model-to-instances matching

Input: O: Ontology, S: Supermodel, M_{set}: set of models, and I: instance
Output: ranking of models
  $I_t \leftarrow$ the top-level individual of I
  for each $M \in M_{set}$ **do**
    $M_t \leftarrow$ the top-level individual of M
    for each role assignment $\mathfrak{R}$ **do**
      Compute $P(M_t \sim I_t|\mathfrak{R})$
    end for
    Let $p_M = \max_{\mathfrak{R}} P(M_t \sim I_t|\mathfrak{R})$
  end for
  Return $\{\langle M, p_M\rangle : M \in M_{set}\}$ ordered by p_M

Fig. 9. Algorithm for instance-to-models matching

tasks of the matcher, we need to compute the posterior probability of $M_t \sim I_t$ for the role assignment $\mathfrak{R}$, $P(M_t \sim I_t|\mathfrak{R})$. Given instance ($I$), model ($M$), role assignment ($\mathfrak{R}$) , ontology ($O$) and supermodel ($S$), in this section we show how to construct a Bayesian network. We compute the probability $P(M_t \sim I_t|\mathfrak{R})$ from the constructed Bayesian network.

The model description defines a Bayesian network, given I, $\mathfrak{R}$, O and S. We can construct the Bayes net dynamically during the matching process. There are two phases. In the first phase we construct the graph structure and in the second phase we construct the conditional probability tables. There are five kinds of random variables in the Bayesian network:

K1: For each model individual M_k, if $M_k = I_j \in \mathfrak{R}$, and for each functional enumerated property P specified in the model, a random variable, which we write as $\langle M_k, P\rangle$, corresponds to individual-property pair. The domain (or values) of $\langle M_k, P\rangle$ is the range of P.

K2: For each model individual M_k, if $M_k = I_j \in \mathfrak{R}$, and for each non-functional enumerated property P specified in the model, for each value V in the range of P, a Boolean random variable, which we write as $\langle M_k, P, V\rangle$, corresponds to

Input: O: ontology, $\mathfrak{R}$: role assignment, M : model, and I: instance
Output: G: graph structure of Bayes net

Construct Nodes:

for each correspondence statement r of $\mathfrak{R}$ **do**
  if r is of the form $M_k = \perp$ **then**
    construct Boolean random variable $\langle M_k = \perp \rangle$
  else
    r is of the form $M_k = I_j$
    construct a Boolean random variable $\langle M_k \sim I_j \rangle$
    for each functional enumerated property, P, of M_k **do**
      construct random variable $\langle M_k, P \rangle$
      $values(\langle M_k, P \rangle) \leftarrow range(P)$
    end for
    for each non-functional enumerated property, P, of M_k **do**
      for each value $V \in range(P)$ **do**
        construct Boolean random variable $\langle M_k, P, V \rangle$
      end for
    end for
    if $\langle M_k, \text{rdf:type}, t_i, p_i \rangle \in M$ **then**
      construct a random variable $\langle M_k, \text{rdf:type} \rangle$
      create the values of $\langle M_k, \text{rdf:type} \rangle$
    end if
  end if
end for
Let N be the set of all the constructed random variables

Construct Arcs:

for each constructed random variable n of N **do**
  if n is of the form $\langle M_k \sim I_j \rangle$ or $\langle M_k = \perp \rangle$ **then**
    $parent(n) \leftarrow \langle M_p \sim I_t \rangle$ such that $\langle M_p, prop, M_k, prob \rangle \in M, \langle M_p \sim I_t \rangle \in N$
  else if n is of the form $\langle M_k, P \rangle$ or $\langle M_k, P, V \rangle$ **then**
    $parent(n) \leftarrow \langle M_k \sim I_j \rangle$ such that $\langle M_k \sim I_j \rangle \in N$
  end if
end for
Return graph structure G

Fig. 10. Algorithm for constructing the graph structure of the Bayes net

individual-property-value pair.

K3: For each model individual M_k, if $\langle M_k, \text{rdf:type}, C, p \rangle \in M$, a random variable, which we write as $\langle M_k, \text{rdf:type} \rangle$. The variable $\langle M_k, \text{rdf:type} \rangle$ is a hierarchically structured variable [Sharma and Poole, 2005]. The values that $\langle M_k, \text{rdf:type} \rangle$ can take are hierarchically structured into an abstraction tree of classes (tree hierar-

chy of the types of M_k).

K4: For each correspondence statement $M_k = I_j \in \mathfrak{R}$, a Boolean random variable, which we write as $\langle M_k \sim I_j \rangle$. The Boolean random variable $\langle M_k \sim I_j \rangle$ represents that model individual M_k matches with instance individual I_j.

K5: For each correspondence statement $M_k = \perp \in \mathfrak{R}$, a Boolean random variable, which we write as $\langle M_k = \perp \rangle$. The Boolean random variable $\langle M_k = \perp \rangle$ represents that model individual M_k doesn't match with any instance individual.

The algorithm for constructing the graph structure of BN is shown in Figure 10.

The values of $\langle M_k, \text{rdf:type} \rangle$ are hierarchically structured into an abstraction tree of classes that can be the type of M_k. We create the domain of $\langle M_k, \text{rdf:type} \rangle$ with only few values that are necessary for computing the posterior probability of match. The creation of the values of $\langle M_k, \text{rdf:type} \rangle$ is discussed in Section 4.1.

Example 4.1 Consider matching the mineral deposit model *depModelA* as shown in Figure 7 with the instance *deposit1* as shown in Figure 6. For the role assignment $\mathfrak{R}1$ of Example 3.2, the constructed Bayesian network is shown in Figure 11. The domains of the variables $\langle depModelA, hasGeneticSetting \rangle$, $\langle depModelA, hasAge \rangle$, $\langle mineralA, hasColour \rangle$, and $\langle mineralA, hasAge \rangle$ are given below:

$$domain(\langle depModelA, hasAge \rangle) = \{palaeozoic, archean, cainozeic\}$$
$$domain(\langle depModelA, hasGeneticSetting \rangle) = \{greenStoneBelt, oceanRidge\}$$
$$domain(\langle mineralA, hasColour \rangle) = \{white, blue, pink, clear\}$$
$$domain(\langle depModelA, hasAge \rangle) = \{palaeozoic, archean, cainozeic\}$$

The computation of the values of $\langle rockA, \text{rdf:type} \rangle$ is discussed in Example 4.2.

4.1 Creating the values of $\langle M_k, rdf{:}type \rangle$

The variable $\langle M_k, \text{rdf:type} \rangle$ is a hierarchically structured variable [Sharma and Poole, 2005]. The types of M_k are hierarchically structured into an abstraction tree of classes. For efficient inference in Bayesian network, we can compute the domain of $\langle M_k, \text{rdf:type} \rangle$, given the model and instance descriptions, with few values that are necessary to compute the posterior probability of match [Sharma and Poole, 2005].

Suppose model individual M_k corresponds to instance individual I_j and I_j is of type C_p but not of types $C^1_{ab}, \ldots, C^k_{ab}$. The instance description provides observation for the constructed Bayesian network. Thus, the values that are true for $\langle M_k, \text{rdf:type} \rangle$ are C_p but not $C^1_{ab}, \ldots, C^k_{ab}$. The observations (the type of I_j) divides the tree hierarchy of M_k's types into three regions $R1$, $R2$, and $R3$:

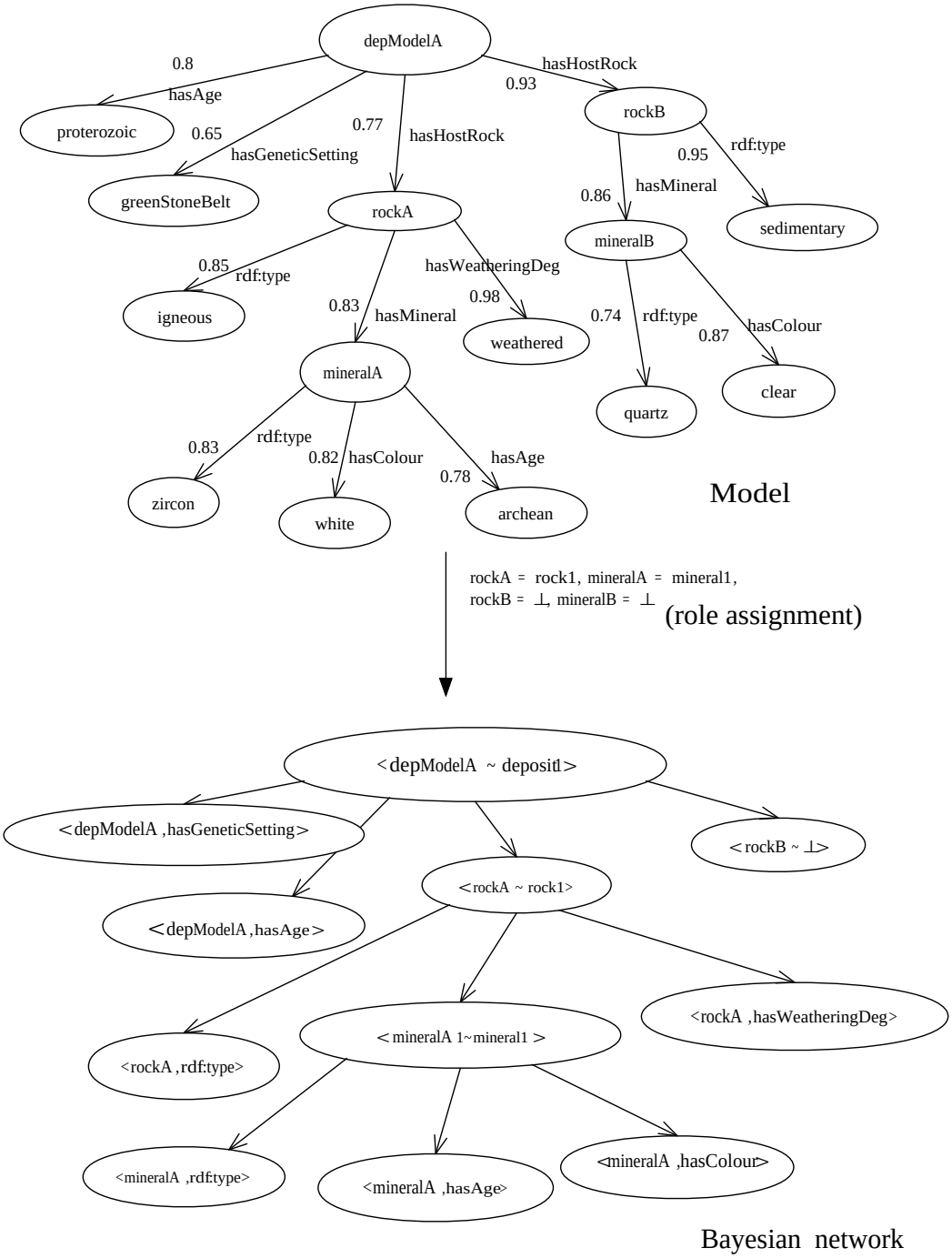

Fig. 11. A Bayesian network defined by the semantic network shown in Figure 7 for the role assignment: $depModelA = deposit1$, $rockA = rock1$, $rockB = \bot$, $mineralA = mineral1$

R1: consists of C_p and its super-classes, i.e., the classes that are true for the observation.

R2: consists of subclasses of C_p that are not $C_{ab}^1, \ldots, C_{ab}^k$, i.e., the classes that we know nothing about.

R3: the rest of the classes from the tree hierarchy of M_k's types, i.e., it includes the

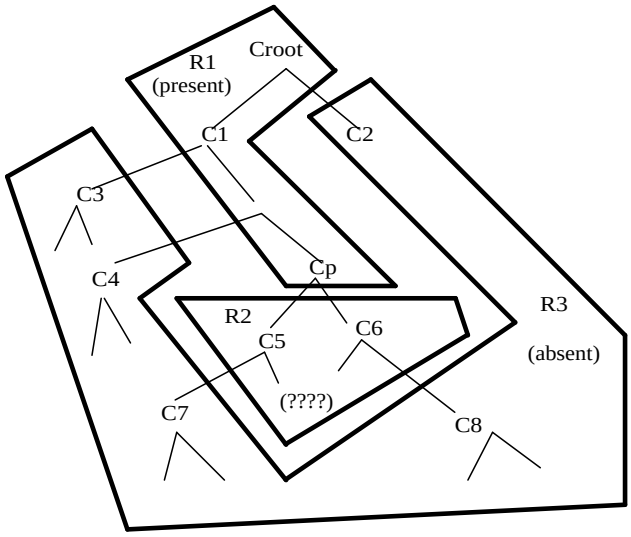

Fig. 12. Observing that Cp is present and $C7$ and $C8$ are absent divide the abstraction hierarchy of M_k's types into three regions R1, R2, and R3.

classes that we know are false (or absent).

For example, suppose I_j is of type C_p but not of types $C7$ and $C8$ as shown in Figure 12. The regions $R1$, $R2$, and $R3$ are shown by the bold lines in Figure 12.

We do not need to distinguish between the values of $\langle M_k, \text{rdf:type} \rangle$ that we know are false. Then, we can consider that $\langle M_k, \text{rdf:type} \rangle$ has two values: one represents all the values that are false and another represents all the values we know nothing about. Suppose (v_{false}) represents all the values that are false and $(v_{notknow})$ represents all the values we know nothing about. Thus,

v_{false} = "$(C_{root} - C_p) \cup (\text{absent observation})$"
$v_{notknow}$ = "$C_p - (\text{absent observation})$"
$domain(\langle M_k, \text{rdf:type} \rangle) = \{v_{notknow}, v_{false}\}$

If M_k has exceptional classes in region $R2$, we need to partition the values represented by $v_{notknow}$ into subsets based on the exceptional classes of M_k that are in region $R2$. Let V_{ex} be the union of C_p and the set of the exceptional classes of M_k that are in region $R2$.

We can construct the domain of $\langle M_k, \text{rdf:type} \rangle$ by values v^a that corresponds to each exceptional class $C_k \in V_{ex}$ as discussed by Sharma and Poole [2005]. We compute the value v^a for each exceptional class $C_k \in V_{ex}$ as follows:

- v^a = "C_k - {absent classes that are subclasses of C_k}", if C_k doesn't have exceptional strict subclasses in V_{ex}
- Otherwise, v^a = "$C_k - C_1 - \cdots - C_m$ - {absent classes that are subclasses of C_k}" where $C_1, \ldots, C_m$ are the highest exceptional strict subclasses of C_k in V_{ex}.

Let $v_1^a, \ldots, v_k^a$ be the abstract values that represent non-empty sets and corresponds to exceptional classes $C_k \in V_{ex}$. Then, the domain of $\langle M_k, \text{rdf:type} \rangle$ is:

$$domain(\langle M_k, \text{rdf:type} \rangle) = \left\{ v_1^a, \ldots, v_k^a, v_{false} \right\}$$

The variable $\langle M_k, \text{rdf:type} \rangle$ has only those values that are necessary for computing the posterior probability of match. Please refer to [Sharma and Poole, 2005] for the details about the computing the values of a hierarchically structured variable.

Example 4.2 Consider determining the domain of variable $\langle rockA, \text{rdf:type} \rangle$ as shown in Figure 11. The individual *rockA* corresponds to instance individual *rock1*, which is of type *granite*. Given the observation that *rock1* is of type granite, the domain of $\langle rockA, \text{rdf:type} \rangle$ contains only two values: "granite" and "rock − granite".

$$domain(\langle rockA, \text{rdf:type} \rangle) = \{\text{“granite”, “rock} - \text{granite”}\}$$

4.2 Construct tables

After constructing the graph structure of the Bayes net, we construct the conditional probability tables (CPTs) for each node in the Bayesian network, given the model (M) and supermodel (S). The CPTs for each type of variable is computed as follows:

K1: For random variable $\langle M_k, P \rangle$, where P is a functional enumerated property, we compute $P(\langle M_k, P \rangle = V_i | \langle M_k \sim I_j \rangle)$ as follows:

$$P(\langle M_k, P \rangle = V_i | \langle M_k \sim I_j \rangle = true) = \begin{cases} p_i & \text{if } \langle M_k, P, V_i, p_i \rangle \in M \\ p_r^i \times (1 - \sum_j p_j) & \text{otherwise} \end{cases}$$
$$P(\langle M_k, P \rangle = V_i | \langle M_k \sim I_j \rangle = false) = p_r^i$$

where p_r^i is the prior probability that M_k has value V_i for property P. The probability p_r^i is defined by the supermodel. We take p_r^i from quadruple $\langle cl, P, V_i, p_r^i \rangle \in S$, such that $domain(P) = cl$.

Example 4.3 Consider computing the conditional probability of node $\langle depModelA, hasAge \rangle$ as shown in Figure 11 conditioned on node $\langle depModelA \sim deposit1 \rangle$. The conditional probability $P(\langle depModelA, hasAge \rangle = large | \langle depModelA \sim deposit1 \rangle)$ is:

$$P(\langle depModelA, hasAge \rangle = proterozoic | \langle depModelA \sim deposit1 \rangle = true) = 0.8$$
$$P(\langle depModelA, hasAge \rangle = proterozoic | \langle depModelA \sim deposit1 \rangle = false) = 0.35$$

The value 0.8 is defined by the mineral deposit model $depModelA$ as shown in Figure 7. The value 0.35 is defined by the supermodel as shown in Example 2.8.

K2: Boolean node $\langle M_k, P, V \rangle$, where P is a non-functional enumerated property. The conditional probability $P(\langle M_k, P, V \rangle \,|\, \langle M_k \sim I_j \rangle)$ is given by:

$$P(\langle M_k, P, V \rangle = true \,|\, \langle M_k \sim I_j \rangle = true) = \begin{cases} p & \langle M_k, P, V, p \rangle \in M \\ p_r & \text{otherwise} \end{cases}$$
$$P(\langle M_k, P, V \rangle = true \,|\, \langle M_k \sim I_j \rangle = false) = p_r$$

where p_r is the prior probability that M_k has value V for property P. The probability p_r is defined by the supermodel (S). We take p_r from quadruple $\langle cl, P, V, p_r \rangle \in S$, such that $domain(P) = cl$.

K3: For random variable $\langle M_k, \text{rdf:type} \rangle$, we compute $P(\langle M_k, \text{rdf:type} \rangle = v \,|\, \langle M_k \sim I_j \rangle)$ for each value v of $\langle M_k, \text{rdf:type} \rangle$. As discussed in Section 4.1, value v of $\langle M_k, \text{rdf:type} \rangle$ represents a set difference. Suppose $v =$ "$C_n - C_m^1 - \cdots - C_m^k$". Then,

$$P(\langle M_k, \text{rdf:type} \rangle = v \,|\, \langle M_k \sim I_j \rangle = true)$$
$$= p_n - \sum_{j=1,k} p_m^j$$

where p_n is the probability that M_k is of type C_n. We can compute the probability p_n using equations (1) and (2) as discussed in Section 3.1.

$$P(\langle M_k, \text{rdf:type} \rangle = v \,|\, \langle M_k \sim I_j \rangle = false)$$
$$= P(C_n) - \sum_{j=1,k} P(C_m^k)$$

where $P(C_j)$ is the prior probability that M_k is of type C_j. We can compute $P(C_j)$ by multiplying the probabilities up in the abstraction hierarchy as discussed in Section 2.4.

Example 4.4 Consider computing the conditional probability of Boolean node $\langle rockA, \text{rdf:type} \rangle$ as shown in Figure 11 conditioned on its parent node $\langle rockA \sim rock1 \rangle$. As shown in Example 4.2 the variable $\langle rockA, \text{rdf:type} \rangle$ has two values "granite", and "rock $-$ granite". Then,

$$P(\langle rockA, \text{rdf:type} \rangle = \text{"granite"} \,|\, \langle rockA \sim rock1 \rangle = true)$$
$$= 0.85 \times 0.3$$
$$= 0.255$$

$$P(\langle rockA, \text{rdf:type} \rangle = \text{"rock} - \text{granite"} \,|\, \langle rockA \sim rock1 \rangle = true)$$
$$= 1.0 - 0.255$$
$$= 0.745$$

K4: Boolean node $\langle M_k \sim I_j \rangle$. We compute the conditional probability $P(\langle M_k \sim I_j \rangle \,|\, \langle M_p \sim I_n \rangle)$ as follows:

$$P(\langle M_k \sim I_j \rangle = true | \langle M_p \sim I_n \rangle = true) = p$$
$$P(\langle M_k \sim I_j \rangle = true | \langle M_p \sim I_n \rangle = false) = P(\langle M_k \sim I_j \rangle = true)$$

where p is associated with quadruple $\langle M_p, prop, M_k, p \rangle \in M$.

K5: Boolean node $\langle M_k = \perp \rangle$. We compute the conditional probability table $P(\langle M_k = \perp \rangle \,|\, \langle M_p \sim I_n \rangle)$ as follows:

$$P(\langle M_k = \perp \rangle = true | \langle M_p \sim I_n \rangle = true) = 1 - p$$
$$P(\langle M_k = \perp \rangle = true | \langle M_p \sim I_n \rangle = false) = 1 - P(\langle M_k \sim I_j \rangle = true)$$

where p is associated with quadruple $\langle M_p, prop, M_k, p \rangle \in M$.

The probability $P((M_k \sim I_j) = true)$, in cases **K4** and **K5**, represents the prior probability that model individual M_k is matching with instance individual I_j. The computation of $P(\langle M_k \sim I_j \rangle = true)$ is discussed in Section 4.3.

4.3 Computation of $\mathbf{P(\langle M_k \sim I_j \rangle = true)}$

The cases **K4** and **K5** of Section 4.2 require the computation of probability $P(\langle M_k \sim I_j \rangle = true)$. In this section we show how to compute it.

Consider computing the prior probability $P(\langle rockA \sim rock1 \rangle = true)$ in the Bayes net as shown in Figure 11. Suppose *rockA* in mineral deposit model $depModelA$ represents a role of rock in a mineral deposit. Then, *mineralA* represents a mineral we would expect to be in a rock that fills that role. As shown in Figure 7, this mineral has certain properties. It is zircon, white and archean. The prior probability of having such a mineral is actually constrained by the mineral's properties. That is, the prior probability $P(\langle mineralA \sim mineral1 \rangle = true)$ is constrained by propositions: $\langle mineralA, hasAge \rangle = archean$, $\langle mineralA, \text{rdf:type} \rangle = zircon$, and $\langle mineralA, hasColour \rangle = white$.

$$\begin{aligned} &P(\langle mineralA, hasAge \rangle = archean) \\ &= P(archean | \langle mineralA \sim mineral1 \rangle = true) \times P(\langle mineralA \sim mineral1 \rangle = true) \\ &\quad + P(archean | \langle mineralA \sim mineral1 \rangle = false) \times P(\langle mineralA \sim mineral1 \rangle = false) \end{aligned}$$

So,

$$\begin{aligned} &P(\langle mineralA, hasAge \rangle = archean) \\ &\geq P(archean | \langle mineralA \sim mineral1 \rangle = true) \times P(\langle mineralA \sim mineral1 \rangle = true) \end{aligned}$$

Thus, as long as $P(archean | \langle mineralA \sim mineral1 \rangle = true) \neq 0$,

$$P(\langle mineralA \sim mineral1 \rangle = true) \leq \frac{P(archean)}{P(archean|\langle mineralA \sim mineral1 \rangle = true)}$$
$$P(\langle mineralA \sim mineral1 \rangle = true) \leq \frac{0.2}{0.78}$$

Similarly,

$$P(\langle mineralA \sim mineral1 \rangle = true) \leq \frac{P(white)}{P(white|\langle mineralA \sim mineral1 \rangle = true)}$$
$$P(\langle mineralA \sim mineral1 \rangle = true) \leq \frac{0.4}{0.82}$$

$$P(\langle mineralA \sim mineral1 \rangle = true) \leq \frac{P(zircon)}{P(zircon|\langle mineralA \sim mineral1 \rangle = true)}$$
$$P(\langle mineralA \sim mineral1 \rangle = true) \leq \frac{0.2}{0.83}$$

Thus,

$$P(\langle mineralA \sim mineral1 \rangle = true) \leq min\left\{\frac{0.4}{0.82}, \frac{0.2}{0.78}, \frac{0.2}{0.83}\right\}$$
$$P(\langle mineralA \sim mineral1 \rangle = true) \leq 0.241$$

Any value which is less than 0.241 can be used for $P(\langle mineralA \sim mineral1 \rangle = true)$. However, if a mineral that we need in a deposit of interest doesn't have any other properties that can make $P(\langle mineralA \sim mineral1 \rangle = true)$ less than 0.241, we can consider $P(\langle mineralA \sim mineral1 \rangle = true) = 0.241$. For our implementation, we assume that all of the constraints are given and we consider $P(\langle mineralA \sim mineral1 \rangle = true) = 0.2$.

In general, we can constrain the prior probability of any $\langle M_k \sim I_j \rangle$ node by all of its direct children in the constructed Bayesian network. Let $A_1, \ldots, A_n$ be the direct children of $\langle M_k \sim I_j \rangle$. Then,

$$P(\langle M_k \sim I_j \rangle = true) \leq p_{min}$$

where

$$p_{min} = min_{A_i} \frac{P(A_i)}{P(A_i|\langle M_k \sim I_j \rangle = true)}$$

Note that $P(A_i)$ may be given or it is computed recursively from its children. Any value less than or equal to p_{min} can be taken for $P(\langle M_k \sim I_j \rangle = true)$.

4.4 Computation of $P(M_t \sim I_t | \mathfrak{R}, observation)$

After constructing a Bayesian network, given a role assignment $\mathfrak{R}$, from the semantic network, we want to compute the posterior probability of $M_t \sim I_t$, i.e, $P(M_t \sim I_t | observation, \mathfrak{R})$. The *observation* is the instance I's description.

To insert the evidence in the constructed Bayesian network, we need to map the instance description to the evidence for the constructed Bayesian network. An instance description is a set of triples and quadruples of the forms: $\langle I_j, \text{rdf:type}, C \rangle$, $\langle I_j, P, V, present \rangle$, and $\langle I_j, P, V, absent \rangle$. We map the instance description to the evidence for the constructed Bayesian network as follows:

- a quadruple of the forms $\langle I_j, P, V, present \rangle$ and $\langle I_j, P, V, absent \rangle$, if P is non-functional enumerated property, provides observation for random variable $\langle M_k, P, V \rangle$, if $M_k = I_j \in \mathfrak{R}$
  - the quadruple $\langle I_j, P, V, present \rangle$ provides observation: $\langle M_k, P, V \rangle = true$
  - the quadruple $\langle I_j, P, V, absent \rangle$ provides observation: $\langle M_k, P, V \rangle = false$
- a quadruple of the forms $\langle I_j, P, V, present \rangle$ and $\langle I_j, P, V, absent \rangle$, if P is functional enumerated property, provides observation for random variable $\langle M_k, P \rangle$, if $M_k = I_j \in \mathfrak{R}$
  - the quadruple $\langle I_j, P, V, present \rangle$ provides observation: $\langle M_k, P \rangle = V$
  - the quadruple $\langle I_j, P, V, absent \rangle$ provides observation: $\langle M_k, P \rangle \neq V$
- the quadruples of types $\langle I_p, P, I_j, absent \rangle$, if P is an entity property, provides observations: $\forall M_k$ s.t. $type(M_k) \subseteq type(I_j)$, $\langle M_k \sim \perp \rangle = true$
- the quadruples of the form $\langle I_j, \text{rdf:type}, C \rangle$, and $\langle I_j, \text{rdf:type}, C_1, absent \rangle$ provides observation for $\langle M_k, \text{rdf:type} \rangle$ variable, if $M_k = I_j \in \mathfrak{R}$. The evidence for $\langle M_k, \text{rdf:type} \rangle$ is the disjunction of all those values of $\langle M_k, \text{rdf:type} \rangle$ that are true for I_j.

We can compute the posterior probability of match, $P(M_t \sim I_t | \mathfrak{R}, observation)$, from the constructed Bayesian network using any standard inference algorithms, e.g., VE [Zhang and Poole, 1994]. The random variable $\langle M_t \sim I_t \rangle$ in the constructed BN is the query variable.

Example 4.5 Consider computing the conditional probability of matching mineral deposit model $depModelA$ with deposit $deposit1$, $P(\langle depModelA \sim deposit1 \rangle | observation, \mathfrak{R})$, for the role assignment $\mathfrak{R}$. The role assignment $\mathfrak{R}$ consists of statements: $depModelA = deposit1$, $rockA = rock1$, $mineralA = mineral1$, $rockB = \perp$. The Bayesian network constructed for matching $depModelA$ with $deposit1$ for the role assignment $\mathfrak{R}$ is shown in Figure 11. We can compute $P(\langle depModelA \sim deposit1 \rangle | observation, \mathfrak{R})$ from this Bayesian network. The *observation* is the description of deposit $deposit1$. The *observation* is:

$\langle depModelA, hasSize \rangle = proterozoic$
$\langle depModelA, hasgeneticSetting \rangle = greenStoneBelt$

$\langle rockA, \text{rdf:type} \rangle =$ ‘‘Granite”
$\langle rockA, hasWeatheringDeg \rangle = w$eathered
$\langle mineralA, hasColour \rangle = pink$
$\langle mineralA, hasAge \rangle = archean$
$\langle mineralA, \text{rdf:type} \rangle = zircon$

We use the VE algorithm to compute $P(\langle depModelA \sim deposit1 \rangle | observation, \mathfrak{R})$. The conditional probability $P(\langle depModelA \sim deposit1 \rangle | observation, \mathfrak{R})$ is given below:

$P(\langle depModelA \sim deposit1 \rangle = true | observation, \mathfrak{R}) = 0.81$
$P(\langle depModelA \sim deposit1 \rangle = false | observation, \mathfrak{R}) = 0.19$

In the algorithms of Figures 8 and 9, to compute the probability of a match between a model and an instance, we need to maximize the probability of a match over all possible role assignments and choose the role assignment that maximizes it. However, when there are many individuals the number of role assignments are too many for maximizing over all role assignments. We, therefore, do a greedy search for the best role assignment for the children of a node. We commit the best match for a single model-instance individual role assignment, before considering the other role assignments.

5 Evaluation

It may seem that we should evaluate the current system by simply testing the predictions against real data (as in Figure 3). However, while this may test the reliability of the theory being used, it may not provide an evaluation of our framework, or even our system. For example, Figure 3 is an evaluation of Soilslide Model 2 (and our representation of it) as much as an evaluation of HazardMatch itself.

Our framework is meant to evaluate multiple theories, good, bad and in between. The fact that some of the theories are poor predictors of the data should not be seen as discrediting of our approach but a vindication of it. MineMatch, HazardMatch and other applications of our matching system will be most successful when we can say that some theory does not actually work very well, and we can convince the authors of the theory that we have a faithful representation of their theory, and based on the evidence, the theory does not perform well. In this way, others can know that the theory doesn’t work well, and hopefully the authors of the theory will refine or abandon the theory.

This should be seen as an instance of what is known as the cycle of perception [Mackworth, 1978; Neisser, 1976]. Our matcher is one part of a closed loop of (1) theory (model) specification, (2) data preparation, (3) matching, (4) results evalu-

ation, and (5) theory refinement, which is equivalent to (1). This closed loop for HazardMatch is documented by [Jackson, Jr. et al., 2008].

We make strong independence assumptions in this work, mainly because these assumptions are adequate to represent current published theories [6] . However, we expect that, when people start writing theories using a more formal representation, they will want to have more than the naive Bayes assumption that underlies this work. Lukasiewicz and Schellhase [2007] present a way to extend our previous work to allow conditional probabilities. We expect that we, and others, will extend this work to include more sophisticated modelling abilities including distributions over real quantities (e.g., slopes or geological time), probabilistic dependencies, and designing ontologies with their integration into probabilistic predictions in mind [Poole et al., 2009].

6 Conclusion

In this paper we have described a framework for decision making in rich domains, where we can describe the world at multiple levels of abstraction and detail and have probabilistic models at different levels of abstraction and detail, and are able to use them to make decisions. We are building knowledge-based decision tools in various domains such as mineral exploration and hazard mapping, where we need to have probabilistic reasoning and rich ontologies.

This paper only solves part of the problem. The assumption that the type of the individuals are from taxonomic hierarchies is not generally applicable. In some cases, we may need to represent the types of the individuals by restriction on some of their properties. In this case we need to model the inter-dependencies between the properties. These are ongoing research topics that build on the foundations given in this paper.

References

Arnold, R.W. (2006). Soil survey and soil classification. In S. Grunwald (Ed.), *Environmental Soil-Landscape Modeling*. Taylor and Francis, New York.

Bishop, M.P. and Schroder, J.F. (Eds.) (2004). *Geographic Information Science and Mountain Geomorphology*. Springer, Berlin.

[6] Current published theories are written in natural language which is difficult to translate into a formal representation. They do specify the existence of objects, but do not have complicated conditional statements beyond the statement of the condition in which the theory is applicable.

Chung, C. and Fabbri, A.G. (2005). Systematic procedures of landslide hazard mapping for risk assessment using spatial prediction models. In T. Glade, M. Anderson, and M. Crozier (Eds.), *Landslide Hazard and Risk*. John Wiley and Sons, New York.

da Costa, P.C.G., Laskey, K.B., and Laskey, K.J. (2005). PR-OWL: A Bayesian ontology language for the semantic web. In *Proceedings of the ISWC Workshop on Uncertainty Reasoning for the Semantic Web*. Galway, Ireland. URL `http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-173/`.

Darwiche, A. and Goldszmidt, M. (1994). On the relation between kappa calculus and probabilistic reasoning. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-94)*, pp. 145–153. Morgan Kaufmann.

Demoulin, A. and Chung, C. (2007). Mapping landslide susceptibility from small datasets: A case study in the Pays de Herve (E Belgium). *Geomorphology*, 89(3-4): 391–404.

Dewitte, O. and Demoulin, A. (2008). Combining high-resolution spatial data in landslidemapping: a fuzzy-set-based approach in W Belgium. Geophysical Research Abstracts 10, EGU2008-A-04579, SRef-ID: 1607-7962/gra/EGU2008-A-04579, EGU General Assembly.

Ding, Z. and Peng, Y. (2004). A probabilistic extension to ontology language OWL. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04)*.

Ding, Z., Peng, Y., and Pan, R. (2006). Bayesowl: Uncertainty modeling in semantic web ontologies. In Z. Ma (Ed.), *Soft Computing in Ontologies and Semantic Web*, volume 204 of *Studies in Fuzziness and Soft Computing*. Springer.

Fox, P., McGuinness, D., Middleton, D., Cinquini, L., Darnell, J., Garcia, J., West, P., Benedict, J., and Solomon, S. (2006). Semantically-enabled large-scale science data repositories. In *5th International Semantic Web Conference (ISWC06)*, volume 4273 of *Lecture Notes in Computer Science*, pp. 792–805. Springer-Verlag. URL `http://www.ksl.stanford.edu/KSL_Abstracts/KSL-06-19.html`.

Gillespie, M.R. and Styles, M.T. (1999). BGS rock classification scheme, volume 1: Classification of igneous rocks. Research Report (2nd edition) RR 99-06, British Geological Survey. URL `http://www.bgs.ac.uk/bgsrcs/`.

Hart, P. (1975). Progress on a computer-based consultant. In *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 831–841.

Howson, C. and Urbach, P. (2006). *Scientific Reasoning: the Bayesian Approach*. Open Court, Chicago, Illinois, 3rd edition.

Jackson, Jr., L.E., Smyth, C.P., and Poole, D. (2008). Hazardmatch: an application of artificial intelligence to landslide susceptibility mapping, Howe Sound area, British Columbia. In *4th Canadian Conference on Geohazards*.

Jaynes, E.T. (2003). *Probability Theory: The Logic of Science*. Cambridge University Press. URL `http://omega.albany.edu:8008/JaynesBook.html`.

Koller, D., Levy, A., and Pfeffer, A. (1997). P-classic: A tractable probabilistic description logic. In *Proceedings of 14th National Conference on Artificial In-*

telligence, pp. 390–397.

Lukasiewicz, T. (2008). Expressive probabilistic description logics. *Artificial Intelligence*, 172(6-7): 852–883.

Lukasiewicz, T. and Schellhase, J. (2007). Variable-strength conditional preferences for ranking objects in ontologies. *Journal Web Semantics*, 5(3): 180–194.

Mackworth, A.K. (1978). Vision research strategy: Black magic, metaphors, mechanisms, miniworlds and maps. In E. Riseman and A. Hanson (Eds.), *Computer Vision Systems*, pp. 53–59. Academic Press.

Manola, F. and Miller, E. (2004). *RDF Primer*. W3C Recommendation 10 February 2004. URL `http://www.w3.org/TR/rdf-primer/`.

McGuinness, D. and van Harmelen, F. (2004). OWL web ontology language overview. *W3C Recommendation 10 February 2004, W3C*.

Neisser, U. (1976). *Cognition and Reality*. Freeman, San Francisco, CA. URL `http://huwi.org/2.php`.

Pearl, J. (1989). Probabilistic semantics for nonmonotonic reasoning: A survey. In R.J. Brachman, H.J. Levesque, and R. Reiter (Eds.), *Proc. Frst International Conf. on Principles of Knowledge Representation and Reasoning*, pp. 505–516. Toronto.

Poole, D. and Smyth, C. (2005). Type uncertainty in ontologically-grounded qualitative probabilistic matching. In *Eighth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU-2005)*, pp. 763–774.

Poole, D. (2007). Logical generative models for probabilistic reasoning about existence, roles and identity. In *22nd AAAI Conference on AI (AAAI-07)*. URL `http://www.cs.ubc.ca/spider/poole/papers/AAAI07-Poole.pdf`.

Poole, D., Smyth, C., and Sharma, R. (2008). Semantic science: Ontologies, data and probabilistic theories. In P.C. da Costa, C. d'Amato, N. Fanizzi, K.B. Laskey, K. Laskey, T. Lukasiewicz, M. Nickles, and M. Pool (Eds.), *Uncertainty Reasoning for the Semantic Web I*, LNAI/LNCS. Springer. URL `http://www.cs.ubc.ca/spider/poole/papers/SemSciChapter2008.pdf`.

Poole, D., Smyth, C., and Sharma, R. (2009). Ontology design for scientific theories that make probabilistic predictions. *IEEE Intelligent Systems*, pp. 27–36. URL `http://www2.computer.org/portal/web/computingnow/2009/0209/x1poo`.

Quillian, M. (1968). Semantic memory. In M. Minsky (Ed.), *Semantic Information Processing*, pp. 227–270. MIT Press, Cambridge, MA.

Robinson, V.B., Petry, F.E., and Cobb, M.A. (2003). Special issue on incorporating fuzzy sets in geographic information systems. *Transactions in GIS*, 7(1).

Schumm, S.A. (1991). *A Scientific Approach to Earth Science: Ten Ways to be Wrong*. Cambridge University Press.

Sharma, R. and Poole, D. (2005). Probabilistic reasoning with hierarchically structured variables. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, pp. 1391–1397.

Sharma, R., Poole, D., and Smyth, C. (2007). A system for ontologically-grounded probabilistic matching. In *Proceedings of the Fifth UAI Bayesian Modeling Ap-*

plications Workshop (UAI-AW 2007).

Smith, B. (2003). Ontology. In L. Floridi (Ed.), *Blackwell Guide to the Philosophy of Computing and Information*, pp. 155–166. Oxford:Blackwell.

Smyth, C. and Poole, D. (2004). Qualitative probabilistic matching with hierarchical descriptions. In *Proceedings of Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR-2004)*.

Spohn, W. (1988). A general non-probabilistic theory of inductive reasoning. In *Proceedings of the Fourth Annual Conference on Uncertainty in Artificial Intelligence (UAI-88)*, pp. 149–158.

Yang, Y. and Calmet, J. (2005). Ontobayes: An ontology-driven uncertainty model. In *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, volume 1, pp. 457–463.

Zhang, N. and Poole, D. (1994). A simple approach to Bayesian network computation. In *Proc. of the 10th Candian Conference on Artificial Intelligence*, pp. 171–178.