Representation and Reasoning with Hierarchically Structured Variables

Rita Sharma

rsharma@cs.ubc.ca

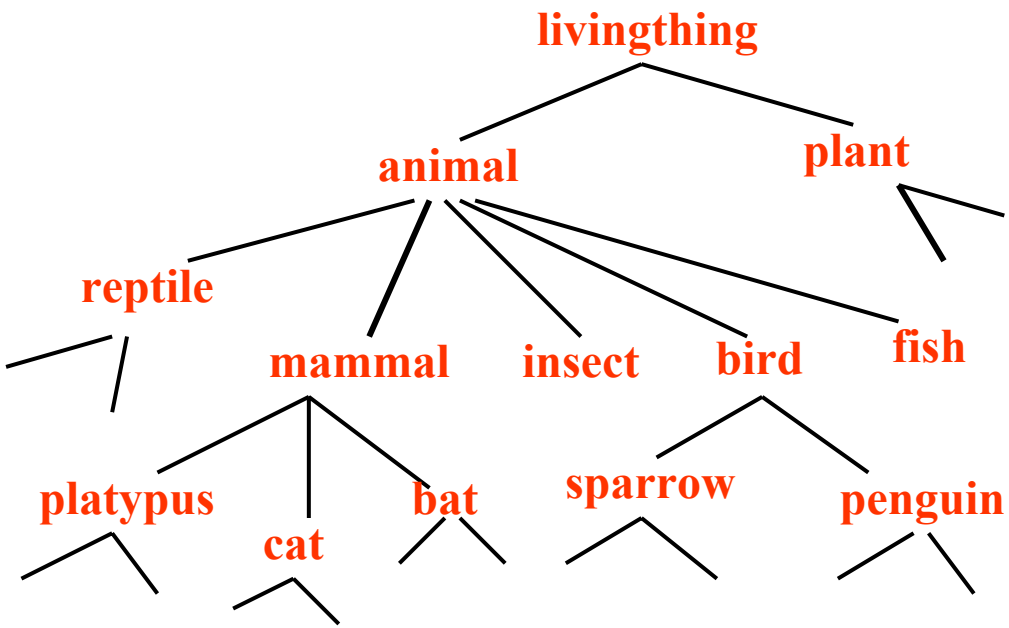
Supervisor: David Poole

Dept. of Computer Science
University of British Columbia
Vancouver, Canada

Introduction

Taxonomical concept hierarchies are an important part of RDF and OWL ontologies used on the semantic web and other hierarchies, e.g., spatial hierarchies.

For example, subsumption hierarchies based on the *subClassOf* or *part Of* properties are widely used.



Taxonomic Hierarchy of Living Things

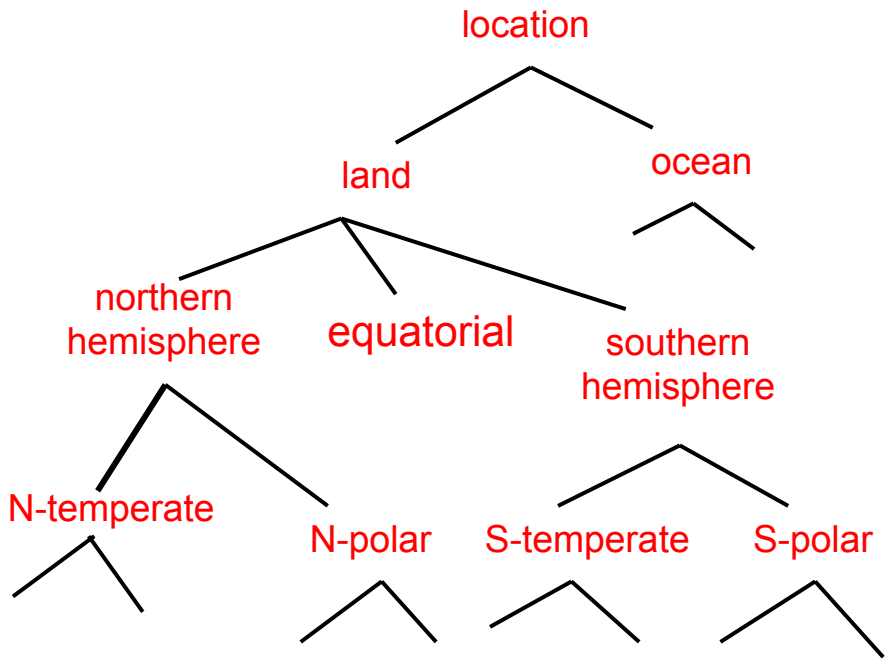
Problem Description

How to represent and exploit taxonomic hierarchies in probabilistic reasoning

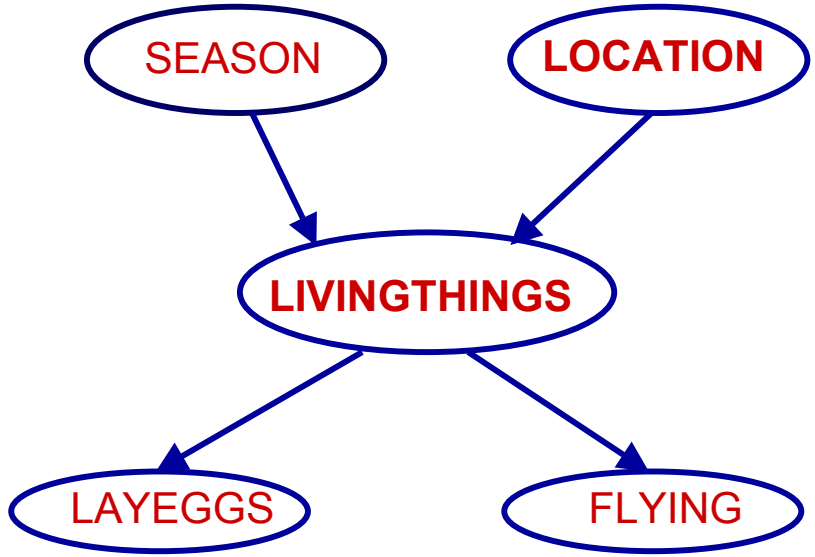
We look at two related problems in Bayesian networks that have both simple and hierarchical variables

- Compact representation for the conditional probabilities
- How to exploit this representation in probabilistic reasoning for computational gain

Example:



taxonomic hierarchy of location



Bayesian Network

- LOCATION and LIVINGTHINGS are hierarchical variables
- FLYING, LAYEGGS, and SEASON are simple variables

Related Work

(Pearl 1988, Poh and Fehling 1993, Heckerman 1990) combine the Belief network and taxonomic hierarchies. In these works the Bayesian network is restricted to a diagnostic network form.

In more recent work Koller and Pfeffer(1998) combine the frame representation systems and Bayesian networks for representing complex structured domain. Their framework uses *isA*, and *partOf* hierarchy in an object oriented way.

The work by Koller and Pfeffer is different than ours, they are looking on different aspect of the problem, concentrating multiple objects. However we are considering when the domain of a random variable is taxonomically structured.

Conditional Probabilities of Bayesian Networks that have Hierarchical Variables

To represent the CPTs compactly we utilise the structure provided by the abstraction hierarchies

There are two main issues:

- specifying the probability for hierarchical variables
- specifying how variables are conditioned on hierarchical variables

Probability for Hierarchical Variable

- The probability of a hierarchical variable can be specify in a top-down manner.
- Each internal class specify a probability distribution over its immediate subclasses conditioned on parents of the hierarchical variable.

e.g., $P(\text{animal} \mid \text{livingthing}) = 0.4$

$P(\text{plant} \mid \text{livingthing}) = 0.6$

In this representation the probability of any class can be computed in a recursive manner as follows:

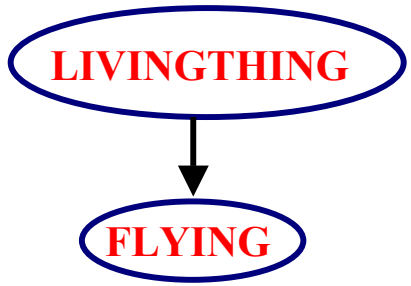
$P(C_j) = P(C_j|C_k)\, P(C_k)$, where C_k is the superclass of C_j

The root class has probability 1 as it represents the set of all values.

$P(\text{sparrow}) = P(\text{sparrow}|\text{bird}) \times P(\text{bird}|\text{animal}) \times P(\text{animal}|\text{livingthing})$

Hierarchical Variable is the parent of a simple variable

We define the default distribution over FLYING for few classes in the hierarchy of living things.

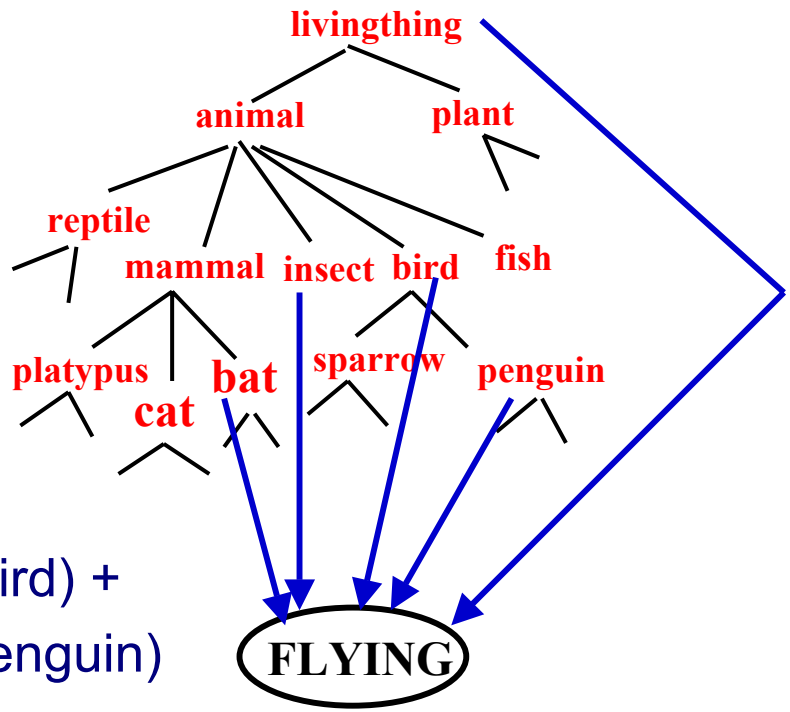


e.g., to define P(FLYING|LIVINGTHINGS) we need to define only 5 distributions:

P_d(FLYING|livingthings), P_d(FLYING|insect)

P_d(FLYING|bird), P_d(FLYING|bat)

P_d(FLYING|penguin)



- We use inheritance

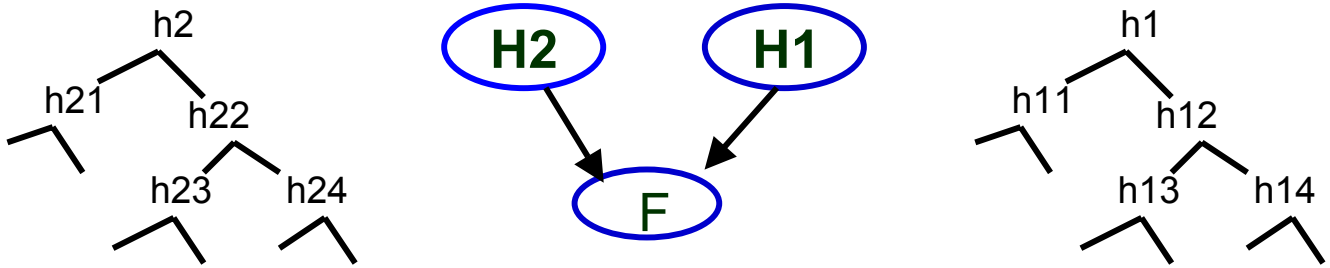
  then P(flying|sparrow) = P_d(flying|bird)

P(flying|bird) = P(sparrow|bird)×P_d(flying|bird) +
P(penguin|bird)×P_d(flying|penguin)

Multiple Inheritance

Multiple inheritance can arise when a variable has more than one hierarchical parents

Example:



Suppose to specify P(F|**H1**Λ**H2**) we have defined default distributions:

$P_d(F|h23 \Lambda h12)$ and $P_d(F|h22 \Lambda h14)$

then $P(F|h23 \Lambda h14)$ can be inherited either from

$P_d(F|h23 \Lambda h12)$ or $P_d(F|h22 \Lambda h14)$

- We explicitly disallow multiple inheritance
- The user need to provide distribution $P_d(F|h23 \Lambda h14)$

Exploiting Structure Provided by Abstraction Hierarchies

Idea: given some evidence and a query construct a flat Bayesian Network by abstracting the hierarchical variables to simple variables

Abstraction of Hierarchical Variable H

An abstract value of H is either

- a class in the tree hierarchy
- a non-empty set of classes that are not ancestors of one another

An abstraction level (L_H) of H is a set of abstract values of H that are mutually exclusive and exhaustive.

An abstraction of H is a simple variable H^a with domain L_H

Construction of Flat Bayesian Network

The algorithm consist of two phases:

Phase1: Abstract

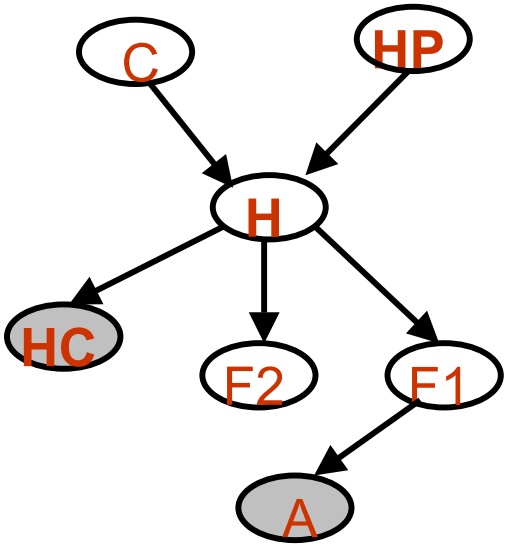
- we traverse the network from leafs upwards
- we prune a variable that isn't queried or observed and doesn't have any children
- compute the abstraction H^a for each relevant hierarchical variable **H**
  - find the relevant exceptional classes of **H** for each child of **H**
  - The domain of H^a is the set of abstract values for each exceptional class

Phase 2: Construct Tables

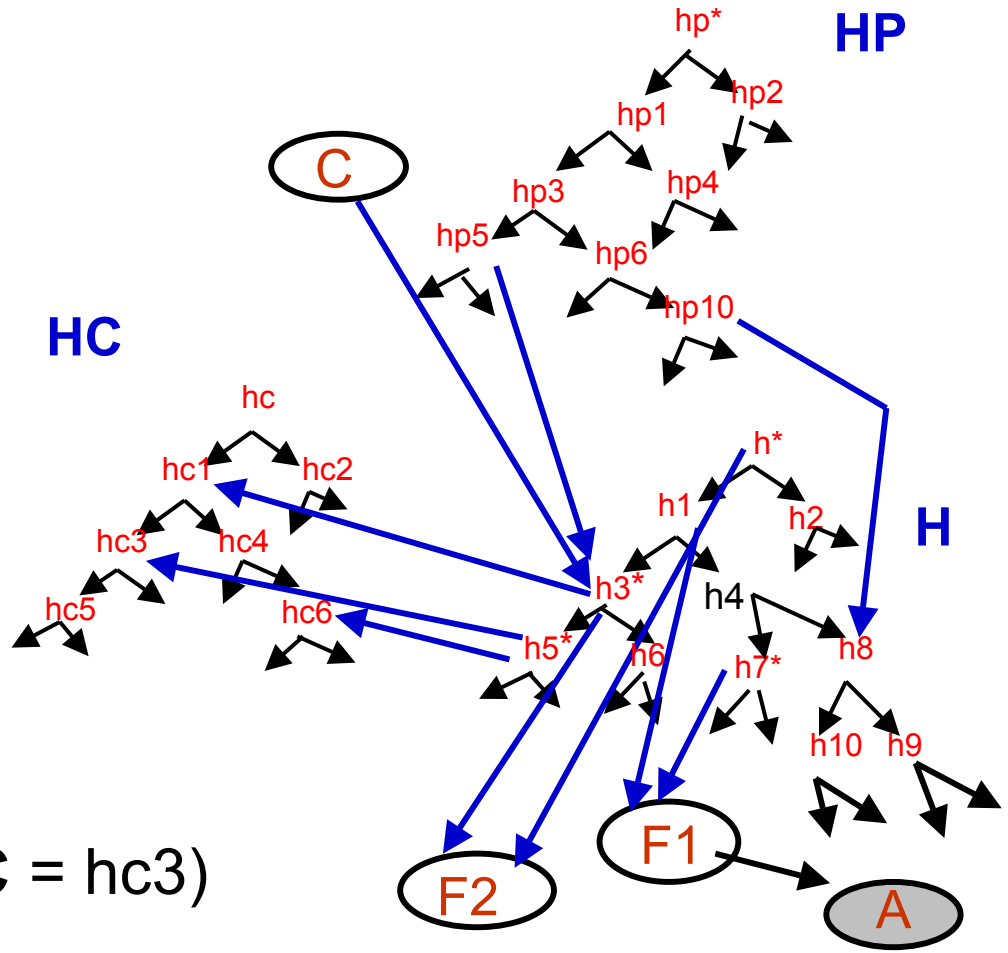
The algorithm constructs the CPT for the following two cases:

- Xa is abstracted variable
- Xa is not abstracted variable but some of its parents are abstracted

Example :



Bayesian Network



Bayesian Network with tree hierarchies

query P(**H** = h10|A = a1Λ**HC** = hc3)

- F2 is pruned
- hierarchical variables **H**, **HP**, and **HC** are abstracted

Abstraction of HC (Observed variable, with no children)

We observed **HC** = hc3

- All the superclasses of hc3 are also true.
- Any class that is neither an ancestor or descendant of hc3 is false.

We can abstract **HC** by a simple variable HC^a

$Val(HC^a) = \{hc3, hc \wedge \sim hc3\}$

Computation of $P(HC^a = hc3|H^a)$:

$$P(HC^a = hc3|H^a) = Pd(hc3|hc1 \wedge H^a) \times Pd(hc1|hc \wedge H^a)$$

We can compute the right hand side after abstracting **H**

Abstraction of H

The classes that we need are

- those that are exceptional for F1 and HC^a (h, h1, h3, h7)
- necessary to answer the query (h110)

We can abstract H by simple variable H^a that has domain

$Val(H^a) = \{h3, h7, h10, h1 \wedge \sim h3 \wedge \sim h7 \wedge \sim h10, h \wedge \sim h1\}$

- values h3 and h7 are there because we get evidence for these values from HC^a and F1
- value h10 is the value we will eventually query
- The last two values cover the different cases that have the same default probabilities

The domain of H^a is a minimal set of values that preserve the distinctions needed and is adequate to answer the query

Conditional Probability Tables

We create the values of H^a such that we can have simple CPTs for its children

$P(F1|H^a)$

H^a	$P(F1|H^a)$
h3	$P_d(F1|h1)$
h7	$P_d(F1|h7)$
h10	$P_d(F1|h1)$
h1Λ~h3 Λ~h7Λ~h10	$P_d(F1|h1)$
hΛ~h1	$P_d(F1|h)$

$P(HC^a = hc3|H^a)$

H^a	$P(HC^a = hc3|H^a)$
h3	$P_d(hc3|hc1\Lambda h3)\ P_d(hc1|hc\ \Lambda h3)$
h7	$P_d(hc3|hc1\Lambda h3)\ P_d(hc1|hc\ \Lambda h3)$
h10	$P_d(hc3|hc1\Lambda h3)\ P_d(hc1|hc\Lambda h3)$
h1 Λ~h3Λ~h7Λ~h10	$P_d(hc3|hc1\ \ \Lambda h3)\ P_d(hc1|hc\Lambda h3)$
h Λ~h1	$P_d(hc3|hc1\ \ \Lambda h3)\ P_d(hc1|hc\Lambda h3)$

Computational Complexity

- The size of the flat Bayesian network is independent of the size of the hierarchies.

  It depends on how many classes in the hierarchy are exceptional with respect to their children that have observed descendants.

- The running time to construct the flat Bayesian network depends on :
  - depth of the exceptional classes
  - number of exceptional classes.

Future Thesis Work

- Evaluation of the proposed approach on some realistic data
- What are the trade offs ?
- Exploring the relationship with Context specific Independence