Causal Structure Estimation by Prolog

Tatsuo Otsu
Hokkaido University, Department of Behavioral System Science
N.10 W.7, Kita-ku
Sapporo 060-0810, Japan
E-Mail otsu@let.hokudai.ac.jp

Recent advances of statistical methodology have increased the need for manipulating complex models. Especially graphical modeling requires representation of complicated structures. The ability of Prolog, which is a logic programming language for symbolic manipulation, for inference on graphical models is shown. Function of pattern matching and indeterminate computing mechanism support easy implementation of enumerating conditional independence.

1. DAG Representation

A DAG (Directed Acyclic Graph) structure defines a class of probabilistic distribution that has the form $\prod_i f(X_i|pa_i)$, where pa_i shows the parents nodes of X_i. List structure plays an important role in DAG representation. In standard Prolog syntax, a list with four elements is represented as `[a,b,c,d]`. Decomposition of a list into the head and the rest is expressed as `[a,b,c,d] = [Head|Tail]`, where the symbols that start with uppercase letter show *variables*. The equality symbol shows *unification*. If this expression is evaluated, `Head` is unified to `a`, and `Tail` is unified to `[b,c,d]`. These functions enable simple pattern-matching description. Suppose that DAG G has three nodes `a, b`, and `c`. And that `b` and `c` are the children of `a`, and `c` is the child of `b`. This structure is represented by a list `[a-[b,c],b-[c],c-[]]`.

2. Path Search by Prolog

Using Prolog's indeterminate computation mechanism, we can search the directed paths in a DAG with above representation. The following short code can search directed paths between `Start` and `End` in a DAG G.

```
sg_path(Start,End,G, Path) :-  sg_path0(Start,End,G,[Start],Path).
sg_path0(Start,End,G,_,[Start,End]) :-  sg_edge(Start,End,G).
sg_path0(Start,End,G,Path1,[Start|Path2]) :-
        sg_edge(Start,Next,G), Next \== End, \+ member(Next,Path1),
        sg_path0(Next,End,G,[Next|Path1],Path2).
```

A builtin function, 'findall', enumerates possible solutions of a predicate.

A slight modification of the path search predicates enables d-separation identification. The definition of $d-$separation is a topological feature of the nodes in a DAG. If and only if X and Y are d-separated by Z, X and Y are conditionally independent as to Z in the DAG compatible distributions. A similar method to the path enumeration enables to find d-connected paths of nodes. The complete enumeration of d-connected paths by Prolog makes d-separation detection possible.

3. Classification of Equivalent DAG Structures

If differenent DAGs G and G' specify the same distributions, they are called *statistically equivalent*. A DAG usually has several statistically equivalent model structures. If and only if G and G' have the same skeleton (undirected path structure) and the same *v-structure*s, which are edge patterns of $X \rightarrow Y \leftarrow Z$ with no edge between X and Z, they are statistically equivalent (Pearl,2000; Spites et al. 1993). The pattern generation ability of Prolog enables to determine the equivalent classes of DAG structures. The enumeration procedure has two parts. The first is classification of topologically equivalent skeletons. The second is the $v-structure$ identification. The former is the more computationally burdensome process. A sophisticated native mode compiler (SICStus Prolog 3) on small (and old) UNIX WS (SUN SS5) took a few seconds for complete classification of five nodes DAG structures. This simple enumeration procedure takes exponentially increasing iterations as the number of nodes. The classification of DAGs of connected six nodes with eight paths took more than six hours. They are classified into 22 undirected path structures. Automatic generation of intervention effect formula (Pearl,1995) is also possible.

REFERENCES

Pearl,J. (1995). Causal diagrams for empirical research (with discussions). *Biometrika*, **82**, 669-710.

Pearl,J. (2000). *Causality: Models, Reasoning and Inference*, Cambridge UP.

Spirtes,P., Glymour,C. & Scheines,R. (1993). *Causation, Prediction, and Search, Springer Lecture Notes in Statistics 81*, Springer.

Sterling,L. & Shapiro,E. (1994). *The Art of Prolog, 2nd ed.*, MIT Press.

RESUME

Le progrès récent des méthodologies statistiques nous demande de manipuler des modèles plus complexes. Ce sont surtout les modélisations en graphes qui nécessitent une représentation de structures graphiques compliquées. Nous démontrons ici l'efficacité de Prolog, un langage de programmation en logique, dans l'inférence de ces modélisations. Il facilite l'énumération de l'indépendance conditionnelle grâce à sa fonction de pattern-matching et son système de calcul indéterminé.