Some Techniques of Data Mining in Biology Part 3

Marco Valtorta
mgv@cs.sc.edu

Department of Computer Science and Engineering
University of South Carolina

March 23, 2000

Overview

A Method for Implementing a Probabilistic Model as a Relational Database

Modeling Biological Data and Structure with Probabilistic Networks (This part was not covered today and will be covered in the March 30 meeting)

Why Implement Probabilistic Models as Relational Databases?

- Highly optimized database machines

- Tighter integration of data mining operations and DBMSs. (Piatezky-Shapiro, *IEEE-IS*, 14, 6 (Nov. 1999): "Data mining will become just another service provided by a good database system.")

- Paper reviews hypergraph model of Bayesian networks (used by Shafer and Shenoy)

- Common computational core with some relational database operations: triangulation, running intersection property, perfect elimination ordering, non-serial dynamic programming.

Relational Data Models

We consider the property of acyclicity.

Acyclicity is a desirable property for relational data models. Reference [1] (by Beeri, Fagin, Maier, and Yannakakis) in Wong, Butz, and Xiang's paper explains why. For example, it is the case that, if a database schema is acyclic, then joins can be computed in an especially simple way, i.e., the schema has a "monotone, sequential join expression."

Hypergraphs and Hypertrees

- Hypergraph $\mathcal{H}$: a "set of subsets"

- Hypertrees have twigs and branches. (This is the terminology used by Shafer and Shenoy.)

- Hypertrees are α-acyclic hypergraphs.

Key insight: Hyperedges of $\mathcal{H}$ are cliques in undirected graph G. The hypergraph is a hypertree iff there is a perfect elimination sequence for G iff G is triangulated.

A factored probability distribution "can be viewed as a product on a hypergraph $\mathcal{H}$" (p.557).

- Configuration (of a subset of variables)

- Projection of a configuration (not so named in paper)

- Concatenation of configurations

- Marginalization: $\phi_k^{\downarrow h}$ is the marginal of function ϕ whose domain is the set of variables k on h, where $h \subseteq k$.

Each function ϕ_h in a factored probability distribution is expressed as a relation Φ_h (by extending the configuration of arguments with the probability mass). The number of attributes in Φ_h is $1 + \#h$.

The product of two functions has a corresponding *product join* of the corresponding relations, which is computed by the three steps on p.558. The product join of Φ_h and Φ_k is written $\Phi_h \otimes \Phi_k$. Note that the first (and most difficult) step involves the computation of the natural join $\Phi_h \bowtie \Phi_k$, while the third step involves a projection.

Marginalization can also be defined by a two-step construction. The first (and most difficult) step involves a projection.

A procedure to compute $\Phi^{\downarrow k}$, for $k \in \mathcal{H}$ is given on p. 559.

The procedure is a translation of the junction tree method (without DistributeEvidence). Evidence is collected to the virtual root $h_1 = k$, using a hypertree construction ordering

$$\{h_1, h_2, \ldots, h_k\}$$

as elimination sequence.

To implement a probabilistic inference systems using a relational DBMS, Wong et al. start by assuming that the hypergraph induced by the potentials is a hypertree, i.e., that the married graph of the Bayesian network is already triangulated.

First, potentials are transformed to relations, as already described. Then, we choose a hypertree construction ordering (defined by its branching function $b(i)$).

The CollectEvidence part has already been described.

The DistributeEvidence formulas are described on p. 561. Note the division by the "old" marginal on the separator variables, resulting in $\Phi^{\downarrow h_i} = \Phi^i_{h_i} \otimes ((\Phi^i_{h_i})^{\downarrow h_i \cap h_{b(i)}})^{-1} \otimes (\Phi^{\downarrow h_{b(i)}})^{\downarrow h_i \cap h_{b(i)}}$.

Some highlights:

- Lemmas 1 and 2 have analogs in Jensen's 1996 book.

- Theorem 1 states (in the language of hypertrees) that the joint probability over all variables is the product of the clique tables divided by the product of the separator tables. The key formulas stating this are at the bottom of p.559 (second column). The last formula shows that *generalized join* involves a division by the separator potential.

- Section 4.2 deals with the operations necessary to compute requests for arbitrary conditional probabilities, rather than just marginals on a single variable.

Problem: Learning Probabilistic Models from Relational Data

Can we exploit the structure of a relational database schema (possibly acyclic) when learning a Bayesian network?

One issue is that it seems that computer scientists are *not* taught to design acyclic database schemata.

Items for Possible Future Work

1. In EM, one may exploit the indexing facilities of the DBMS when computing expectations (FVJ).

2. A BN may be viewed as an extension of a DB where some attributes only have expectations (KGO).

3. There is a paper by Getoor and Friedman on this topic (Helge Langseth). It is: Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. "Learning Probabilistic Relational Models." *Proceedings of IJCAI-99*, pp.1300-1307.

4. The DB structure tells you that some attributes are non-influential (e.g., keys determine stg.). Conditional independence and context-specific decompositions within tables may be found by exploring the DB schema (Anders Madsen).

References

Wong, S.K.M., C.J. Butz, and Y. Xiang. "A Method for Implementing a Probabilistic Model as a Relational Database." *Proceedings of UAI-95*, pp.556–564.