A logical way to find high probability pedigrees

James Cussens, University of York

Graphical Models and Genetic Applications
University of Warwick, 2009-04-16

Outline

The problem

Given

- A set G of possible pedigrees;
- a prior over pedigrees $p(g)$;
- observed marker data x_o;
- and an assumption of Mendelian segregation

Find

- $\arg\max_{g \in G} P(g|x_o)$

Why logic?

- There are a great many hard (i.e. non-probabilistic) constraints between the elements of pedigrees, ordered genotypes and unordered genotypes.
- Logic provides a useful way of representing them.
- Logic-based optimisers can exploit them.
- Can incorporate probabilities via weighted logical formulae.

All variables are binary

- Due to the logical approach all variables are binary.
- There will be four disjoint collections of binary variables to encode:
  1. the pedigree (g)
  2. the unobserved ordered genotypes (y)
  3. the observed and unobserved unordered genotypes ($x = (x_h, x_o)$)
  4. and the (possibly observed) auxilliary variables giving e.g. relative age information (z).
- An exponential-family distribution will be defined for $P(g, x, y, z)$.

Outline

Pedigree and auxiliary variables

Pedigree variables $\text{father}(\text{bob}, \text{alice})$, $\text{mother}(\text{alice}, \text{rob})$, ...

Auxiliary variables $\text{older}(\text{bob}, \text{alice})$, ...

Each such variable has two values: TRUE (1) or FALSE (0).
There are many constraints, for example:

- $\forall X, Y : \text{father}(X, Y) \rightarrow \text{older}(X, Y)$,
  $\forall X, Y, Z : \text{older}(X, Y) \wedge \text{older}(Y, Z) \rightarrow \text{older}(X, Z)$,...
- $\forall X, Y, Z : X \neq Y \rightarrow \neg\text{father}(X, Z) \vee \neg\text{father}(Y, Z)$, ...

First-order and propositional logic

A universally-quantified first-order formula like:

$$\forall X, Y : \mathrm{father}(X, Y) \rightarrow \mathrm{older}(X, Y)$$

is a compact representation for all its 'ground instances':

- $\mathrm{father}(\mathit{bob}, \mathit{alice}) \rightarrow \mathrm{older}(\mathit{bob}, \mathit{alice})$
- $\mathrm{father}(\mathit{bob}, \mathit{tom}) \rightarrow \mathrm{older}(\mathit{bob}, \mathit{tom})$
- ...

Since we have only finitely many people and alleles we can replace each first-order formula by its set of ground instances.

Ordered genotype variables

Ordered genotype variables $\mathrm{pat}(\mathrm{bob}, \mathrm{a2})$, $\mathrm{mat}(\mathrm{alice}, \mathrm{a4})$, $\ldots$

- pat and mat are functional relations:
  $\forall X, A, B : A \neq B \rightarrow \neg\mathrm{mat}(X, A) \vee \neg\mathrm{mat}(X, B)$,
  $\forall X : \exists A : \mathrm{pat}(X, A)$, $\ldots$
- Homozygous inheritance:
  $\forall X, Y, A : \mathrm{pat}(X, A) \wedge \mathrm{mat}(X, A) \wedge \mathrm{father}(X, Y) \rightarrow \mathrm{pat}(Y, A)$.

Unordered genotype variables

Unordered genotype variables $\text{genotype}(\text{bob}, \text{a1}, \text{a2}), \ldots$

- $\forall X, A, B : \text{genotype}(X, A, B) \leftrightarrow$
  $(\text{pat}(X, A) \wedge \text{mat}(X, B)) \vee (\text{mat}(X, A) \wedge \text{pat}(X, B))$

Possible worlds

- Recall that each *ground atomic formula (atom)*, such as $\mathrm{father}(\mathrm{bob}, \mathrm{alice})$ or $\mathrm{genotype}(\mathrm{bob}, \mathrm{a1}, \mathrm{a2})$ is a binary variable with values: TRUE and FALSE.
- A full joint instantiation of truth-values to all atoms in a given language is known as a *possible world*.
- The rules rule out possible worlds which cannot represent a feasible segregation network.
- We can use *weighted rules* to define a suitable distribution over the surviving worlds.

An example possible world

As is typical only true ground atoms are listed:

```
father(m1,m2)      pa(m2,a1)
older(m1,m2)       pa(f1,a3)
mother(f1,m1)      ma(m1,a1)
older(f1,m1)       ma(m2,a2)
mother(f1,m2)      ma(f1,a1)
older(f1,m2)       genotype(m1,a1,a1)
pa(m1,a1)          genotype(m2,a1,a2)
                   genotype(f1,a1,a3)
```

Weighted rules

A weighted first-order rule like:

$$30 : \forall X, Y, Z : \mathrm{mother}(X, Y) \wedge \mathrm{father}(Y, Z) \rightarrow \neg \mathrm{mother}(X, Z)$$

represents its set of ground instances:

- $30 : \mathrm{mother}(\mathrm{f1}, \mathrm{m1}) \wedge \mathrm{father}(\mathrm{m1}, \mathrm{m2}) \rightarrow \neg \mathrm{mother}(\mathrm{f1}, \mathrm{m2})$
- $30 : \mathrm{mother}(\mathrm{f1}, \mathrm{m2}) \wedge \mathrm{father}(\mathrm{m2}, \mathrm{m1}) \rightarrow \neg \mathrm{mother}(\mathrm{f1}, \mathrm{m1})$
- ...

Each such ground instance is either true or false in any possible world, and so is a binary 'feature' of that world.

An exponential-family distribution

Let m be a possible world satisfying all hard constraints then:

$$P(m) = Z_w^{-1} \exp\left(\sum_{f \in \mathrm{GF}} w_f I(m \models f)\right)$$

where the sum is over all ground weighted formulae, w_f is the weight and $m \models f$ iff formula f is true in world m.

Markov logic

Let $\{(w_i, F_i)\}_i$ be a collection of weighted formulae (clauses) and let m be a possible world, then a *Markov logic network* (Richardson & Domingos, MLJ 05) defines the following exponential-family distribution:

$$P(m) = Z_w^{-1} \exp\left(\sum_i w_i n_i(m)\right)$$

where $n_i(m)$ is the number of true 'instances' of formula F_i in world m.
Hard rules have infinite weight.

Penalty for heterozygosity

Assuming Mendelian segregation

$$-\log 0.5 : \forall X, Y, A, B : \neg(\mathrm{father}(X, Y) \wedge \mathrm{pat}(X, A) \wedge \mathrm{mat}(X, B) \wedge A \neq B)$$

Encoding population frequencies

$$-\log p_i : \forall Y : \exists X : \mathrm{father}(X, Y) \vee \neg \mathrm{pat}(Y, a_i)$$

$$-\log p_i : \forall Y : \exists X : \mathrm{mother}(X, Y) \vee \neg \mathrm{mat}(Y, a_i)$$

Priors on pedigrees

For example:

$$30 : \forall X, Y, Z : mother(X, Y) \wedge father(Y, Z) \rightarrow \neg mother(X, Z)$$

$$20 : \forall X, Y, Z : mother(X, Z) \wedge father(Y, Z) \rightarrow \neg related(X, Y)$$

Just a different way of writing down the prior of (Sheehan & Egeland, 2007).

Incorporating evidence

- Just add in the appropriate ground atoms, e.g.
  - $\mathrm{genotype}(\mathrm{bob}, \mathrm{a1}, \mathrm{a2})$
  - $\mathrm{father}(\mathrm{john}, \mathrm{robin})$
- thus ruling out all worlds in which these are not true.
- The intelligent approach is to 'propagate' the evidence to specialise the general-purpose logical knowledge base.

Outline

From first-order logic to propositional logic

- For any given problem we have a finite number of individuals and alleles.
- So we can (and do) generate all ground instances of all formulae (hard and soft).
- So, the first-order representation is merely a convenient way of formulating the problem.
- The *Alchemy* and *thebeast* Markov logic software do this conversion internally.
- I did it with a little Prolog program.

An simple example

With a uniform prior on pedigrees and this (unordered) genotype data:

```
genotype(m1,a1,a2)  genotype(f1,a2,a2)
genotype(m2,a1,a2)  genotype(f2,a1,a2)
genotype(m3,a2,a2)  genotype(f3,a2,a2)
genotype(m4,a1,a2)  genotype(f4,a1,a2)
genotype(m5,a1,a1)  genotype(f5,a1,a1)
mother(f1,f3)
```

Result

The exact weighted MAX-SAT solver (minimaxsat1.0, Heras *et al*) took 30 seconds to establish that this 'possible world' is the most probable:

```
father(m2,m4)   mother(f1,m1) pa(m1,a1)   ma(m1,a2)
father(m2,f1)   mother(f1,m3) pa(m2,a1)   ma(m2,a2)
father(m2,f5)   mother(f1,m4) pa(m3,a2)   ma(m3,a2)
father(m4,m1)   mother(f1,f3) pa(m4,a1)   ma(m4,a2)
father(m4,m3)   mother(f2,m2) pa(m5,a1)   ma(m5,a1)
father(m4,m5)   mother(f2,f1) pa(f1,a2)   ma(f1,a2)
father(m4,f3)   mother(f2,f4) pa(f2,a1)   ma(f2,a2)
father(m4,f4)   mother(f2,f5) pa(f3,a2)   ma(f3,a2)
                mother(f5,m5) pa(f4,a2)   ma(f4,a1)
                              pa(f5,a1)   ma(f5,a1)
```

Not there yet!

- This is a nice way of solving $\arg\max_{g,y} P(x, y|g)P(g)$, but we actually want to solve
  $\arg\max_g P(x|g)P(g) = \arg\max_g \sum_y P(x, y|g)P(g)$.
- Domingos's group (University of Washington) apparently working on this right now.

Times

Took 145s.

```
genotype(m1,a1,a2)   genotype(f1,a2,a2)
genotype(m2,a1,a2)   genotype(f2,a1,a2)
genotype(m3,a2,a2)   genotype(f3,a2,a2)
genotype(m4,a1,a2)   genotype(f4,a1,a2)
genotype(m5,a1,a1)   genotype(f5,a1,a1)
```

If a total order is added, this reduces to 0.076s.

Times

Took 15s.

```
gc(hard,[genotype(m1,a1,a1)]).
gc(hard,[genotype(m2,a1,a2)]).
gc(hard,[genotype(m3,a2,a2)]).
gc(hard,[genotype(m4,a2,a2)]).
gc(hard,[genotype(m5,a1,a1)]).
gc(hard,[genotype(f1,a2,a2)]).
gc(hard,[genotype(f2,a2,a2)]).
gc(hard,[genotype(f3,a2,a2)]).
gc(hard,[genotype(f4,a1,a1)]).
gc(hard,[genotype(f5,a1,a1)]).
```

Actual input

```
c 1 father(m1,m2)
c 2 older(m1,m2)
c gc(hard,[-father(m1,m2),older(m1,m2)]) [1617,-1,2,0]
...
p wcnf 250 2794
1617 -1 2 0
1617 -3 4 0
....
```

Actual output

```
o 4936
...
o 40
c RES: 40 145.134 50926224
s OPTIMUM FOUND
v -1 2 -3 4 5 6 -7 8 -9 10 -11 -12 -13 14 -15 16 17 18 -19
c ------------------------------------------------------------
c
c restarts              : 1
c conflicts             : 0                (0 /sec)
c decisions             : 50926224         (350891 /sec)
c propagations          : 24816129         (170988 /sec)
c inspects              : 0                (0 /sec)
c CPU time              : 145.134 s
c
```