A one–minute introduction to the `gRain` package

Søren Højsgaard
Department of Genetics and Biotechnology
Aarhus University, Denmark

May 7, 2010

Contents

1 Introduction

The `gRain` package implements propagation in [gra]phical [i]ndependence [n]etworks (hereafter abbreviated `grain`). Such networks are also known as probabilistic networks and Bayesian networks. More information about the package might be available from the webpage `http://gbi.agrsci.dk/~shd/`.

2 A worked example: chest clinic

This section reviews the chest clinic example of Lauritzen and Spiegelhalter (1988) (illustrated in Figure 1) and shows one way of specifying the model in `gRain`. Lauritzen and Spiegelhalter (1988) motivate the chest clinic example as follows:

> "Shortness–of–breath (dyspnoea) may be due to tuberculosis, lung cancer or bronchitis, or none of them, or more than one of them. A recent visit to Asia increases the chances of tuberculosis, while smoking is known to be a risk factor for both lung cancer and bronchitis. The results of a single chest X–ray do not discriminate between lung

cancer and tuberculosis, as neither does the presence or absence of dyspnoea.”

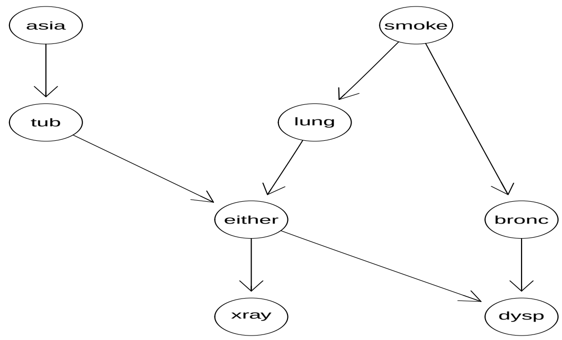

Figure 1: Chest clinic example from LS.

2.1 Building a `grain`

A Bayesian network is a special case of graphical independence networks. In this section we outline how to build a Bayesian network. The starting point is a probability distribution factorising accoring to a DAG with nodes V. Each node $v \in V$ has a set $pa(v)$ of parents and each node $v \in V$ has a finite set of states. A joint distribution over the variables V can be given as

$$p(V) = \prod_{v \in V} p(v|pa(v)) \tag{1}$$

where $p(v|pa(v))$ is a function defined on $(v, pa(v))$. This function satisfies that $\sum_{v^*} p(v = v^*|pa(v)) = 1$, i.e. that for each configuration of the parents $pa(v)$, the sum over the levels of v equals one. Hence $p(v|pa(v))$ becomes the conditional distribution of v given $pa(v)$. In practice $p(v|pa(v))$ is specified as a table called a conditional probability table or a CPT for short. Thus, a Bayesian network can be regarded as a complex stochastic model built up by putting together simple components (conditional probability distributions).

Thus the DAG in Figure 1 dictates a factorization of the joint probability function as

$$p(V) = p(\alpha)p(\sigma)p(\tau|\alpha)p(\lambda|\sigma)p(\beta|\sigma)p(\epsilon|\tau,\lambda)p(\delta|\epsilon,\beta)p(\xi|\epsilon). \tag{2}$$

In (2) we have α = asia, σ = smoker, τ = tuberculosis, λ = lung cancer, β = bronchitis, ϵ = either tuberculosis or lung cancer, δ = dyspnoea and ξ = xray. Note that ϵ is a logical variable which is true if either τ or λ are true and false otherwise.

2.2 Queries to grains

Suppose we are given the finding (evidence) that a set of variables $E \subset V$ have a specific value e^*. For example that a person has recently visited Asia and suffers from dyspnoea, i.e. $\alpha =$ yes and $\delta =$ yes.

With this finding, we are often interested in the conditional distribution $p(v|E = e^*)$ for some of the variables $v \in V \setminus E$ or in $p(U|E = e^*)$ for a set $U \subset V \setminus E$.

In the chest clinic example, interest might be in $p(\lambda|e^*)$, $p(\tau|e^*)$ and $p(\beta|e^*)$, or possibly in the joint (conditional) distribution $p(\lambda, \tau, \beta|e^*)$.

Interest might also be in calculating the probability of a specific event, e.g. the probability of seeing a specific finding, i.e. $p(E = e^*)$.

3 A one–minute version of gRain

A simple way of specifying the model for the chest clinic example is as follows.

1. Specify conditional probability tables (with values as given in Lauritzen and Spiegelhalter (1988)):

```
> yn <- c("yes", "no")
> a <- cptable(~asia, values = c(1, 99), levels = yn)
> t.a <- cptable(~tub + asia, values = c(5, 95, 1, 99), levels = yn)
> s <- cptable(~smoke, values = c(5, 5), levels = yn)
> l.s <- cptable(~lung + smoke, values = c(1, 9, 1, 99), levels = yn)
> b.s <- cptable(~bronc + smoke, values = c(6, 4, 3, 7), levels = yn)
> e.lt <- cptable(~either + lung + tub, values = c(1, 0, 1, 0, 1,
+     0, 0, 1), levels = yn)
> x.e <- cptable(~xray + either, values = c(98, 2, 5, 95), levels = yn)
> d.be <- cptable(~dysp + bronc + either, values = c(9, 1, 7, 3, 8,
+     2, 1, 9), levels = yn)
```

   Notice that the following forms are also valid specifications

```
> cptable(~tub | asia, values = c(5, 95, 1, 99), levels = yn)

vpa    : tub asia
values : 5 95 1 99
levels (tub) : yes no
normalize : TRUE smooth : 0

> cptable(c("tub", "asia"), values = c(5, 95, 1, 99), levels = yn)

vpa    : tub asia
values : 5 95 1 99
```

```
levels (tub) : yes no
normalize : TRUE smooth : 0
```

2. Create the `grain` from the conditional probability tables:

```
> plist <- compileCPT(list(a, t.a, s, l.s, b.s, e.lt, x.e, d.be))
> in1 <- grain(plist)
> in1

Independence network: Compiled: FALSE Propagated: FALSE
 Nodes: chr [1:8] "asia" "tub" "smoke" "lung" "bronc" "either" "xray" ...
```

3. The `grain` can be queried to give marginal probabilities:

```
> querygrain(in1, nodes = c("lung", "bronc"), type = "marginal")

$lung
lung
  yes    no
0.055 0.945

$bronc
bronc
 yes   no
0.45 0.55
```

   Likewise, a joint distribution can be obtained.

```
> querygrain(in1, nodes = c("lung", "bronc"), type = "joint")

     bronc
lung     yes     no
  yes 0.0315 0.0235
  no  0.4185 0.5265
```

4. Findings can be entered as:

```
> in12 <- setFinding(in1, nodes = c("asia", "dysp"), states = c("yes",
+     "yes"))
```

5. The `grain` can be queried again:

```
> querygrain(in12, nodes = c("lung", "bronc"))

$lung
lung
       yes         no
0.09952515 0.90047485
```

```
$bronc
bronc
      yes        no
0.8114021 0.1885979
```

```
> querygrain(in12, nodes = c("lung", "bronc"), type = "joint")
```

```
     bronc
lung         yes         no
  yes 0.06298076 0.03654439
  no  0.74842132 0.15205354
```

6. Zero probabilities

   Consider setting the finding

```
> in13 <- setFinding(in1, nodes = c("either", "tub"), states = c("no",
+     "yes"))
```

Under the model, this finding has zero probability;

```
> pFinding(in13)
```

```
[1] 0
```

Therefore, all conditional probabilities are (under the model) undefined;

```
> querygrain(in13, nodes = c("lung", "bronc"), type = "joint")
```

```
     bronc
lung  yes  no
  yes NaN NaN
  no  NaN NaN
```

References

Steffen Lilholt Lauritzen and David Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *J. Roy. Stat. Soc. Ser. B*, 50(2):157–224, 1988.