Graphical Models With Sparse CPTs in R

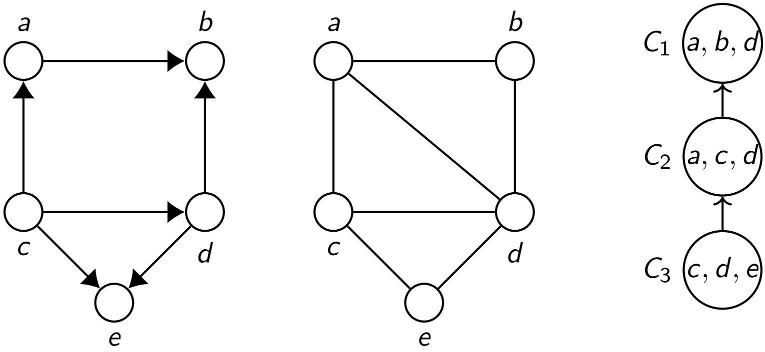
`sparta` and `jti`

Mads Lindskou[†‡]

[†] Department of Mathemaical Sciences, Aalborg University
[‡] Section of Forensic Genetics,
Department of Forensic Medicine,
Faculty of Health and Medical Sciences,
University of Copenhagen, Denmark

07 April, 2021

The Road to Junction Tree Inference



$$
\begin{aligned}
p(a,b,c,d,e) &\overset{\text{definition}}{=} p(c)p(a \mid c)p(b \mid a,d)p(d \mid c)p(e \mid c,d) \\
&\overset{\text{initialize}}{=} \phi_{C_1}(a,b,d)\phi_{C_2}(a,c,d)\phi_{C_3}(c,d,e) \\
&\overset{\text{propagate}}{=} \frac{p(a,b,d)p(a,c,d)p(c,d,e)}{p(c,d)p(a,d)}
\end{aligned}
$$

Necessary Operations

Multiplication (and division)

$$\phi_1(a, b, c)\phi_2(b, c, d) \rightarrow \phi(a, b, c, d)$$

b	b1		b2	
c	c1	c2	c1	c2
a				
a1	1	3	0	0
a2	0	4	2	0

b	b1		b2	
c	c1	c2	c1	c2
d				
d1	0	2	3	1
d2	0	0	1	0

Necessary Operations

Multiplication (and division)

$$\phi_1(a, b, c)\phi_2(b, c, d) \rightarrow \phi(a, b, c, d)$$

b	b1		b2	
c	c1	c2	c1	c2
a				
a1	1	3	0	0
a2	0	4	2	0

b	b1		b2	
c	c1	c2	c1	c2
d				
d1	0	2	3	1
d2	0	0	1	0

	b	b1		b2	
	c	c1	c2	c1	c2
a	d				
a1	d1	0	6	0	0
	d2	0	0	0	0
a2	d1	0	8	6	0
	d2	0	0	2	0

Necessary Operations

Marginalization

$$\sum_a \phi(a, b, c)$$

b	b1		b2	
c	c1	c2	c1	c2
a				
a1	1	3	0	0
a2	0	4	2	0

c	c1	c2
b		
b1	1	7
b2	2	0

This is sparta!

```r
ftable(x)
```

```
      c c1 c2
a  b
a1 b1     1  3
   b2     0  0
a2 b1     0  4
   b2     2  0
```

```r
sparta::as_sparta(x)
```

```
  a b c val
1 1 1 1   1
2 2 2 1   2
3 1 1 2   3
4 2 1 2   4
```

This is `sparta`!

Function	Description
`as_sparta`	Convert array-like object to a `sparta`
`as_cpt`	Convert `sparta` object to a CPT
`mult/div/marg`	Multiply, divide and marginalize
`slice`	Enter evidence

Evidence

Evidence Variables

Introducing evidence variables into the model enables us to calculate posterior (conditional) probabilities that may otherwise be hard to compute.

Evidence

Evidence Variables

Introducing evidence variables into the model enables us to calculate posterior (conditional) probabilities that may otherwise be hard to compute.

Absorbing Evidence

Evidence is entered into the model by setting entries to zero whenever they do not conform with the evidence.

Evidence: `c = c1`

```
      c c1 c2
a  b
a1 b1    1  3
   b2    0  0
a2 b1    0  4
   b2    2  0
```

```
  a b c val
1 1 1 1   1
2 2 2 1   2
3 1 1 2   3
4 2 1 2   4
```

Evidence: `c = c1`

```
       c c1 c2
a  b
a1 b1     1  3
   b2     0  0
a2 b1     0  4
   b2     2  0
```

```
  a b c val
1 1 1 1   1
2 2 2 1   2
3 1 1 2   3
4 2 1 2   4
```

```
       c c1 c2
a  b
a1 b1     1  0
   b2     0  0
a2 b1     0  0
   b2     2  0
```

```
  a b c val
1 1 1 1   1
2 2 2 1   2
```

jti

```
cl <- jti::cpt_list(asia2); cl
```

```
 List of CPTs
 -------------------------
  P( asia )
  P( tub | asia )
  P( smoke )
  P( lung | smoke )
  P( bronc | smoke )
  P( either | lung, tub )
  P( xray | either )
  P( dysp | bronc, either )

  <cpt_list, list>
 -------------------------
```

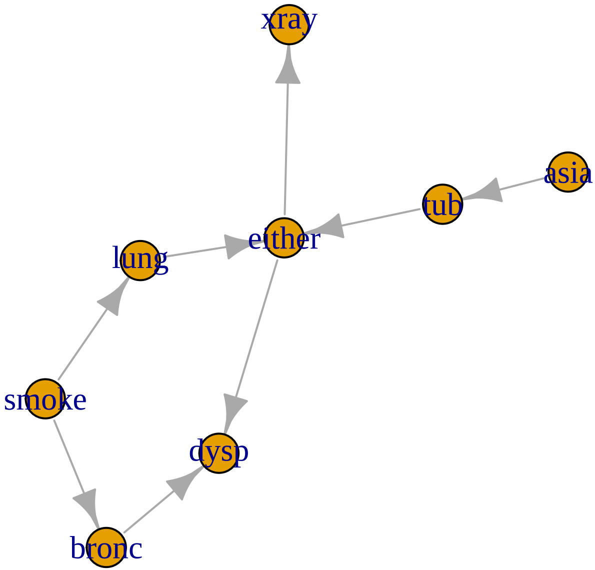
jti

```r
asia2[8]
```

```
$dysp
, , either = yes

     bronc
dysp  yes  no
  yes 0.9 0.7
  no  0.1 0.3

, , either = no

     bronc
dysp  yes  no
  yes 0.8 0.1
  no  0.2 0.9
```

jti

```
plot(jti::get_graph(cl))
```



jti

```r
cp <- jti::compile(cl, evidence = NULL); cp
```

```
Compiled network
-------------------------
 Nodes: 8
 Cliques: 6
  - max: 3
  - min: 2
  - avg: 2.67
 <charge, list>
-------------------------
```

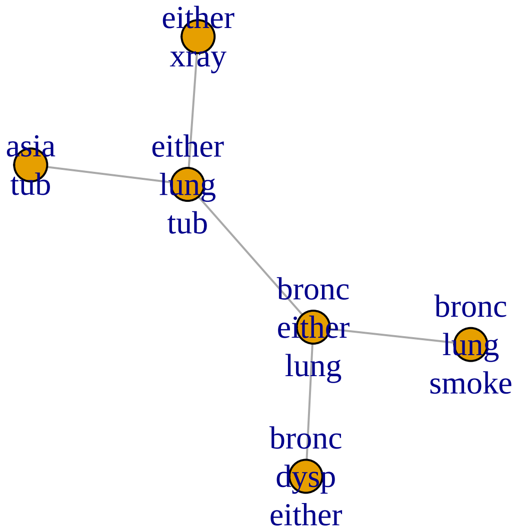
jti

```r
j <- jti::jt(cp, evidence = NULL); j
```

```
Junction Tree
-------------------------
 Propagated: full
 Flow: sum
 Nodes: 6
 Edges: 5 / 15
 Cliques: 6
  - max: 3
  - min: 2
  - avg: 2.67
 <jt, list>
-------------------------
```

jti

```r
plot(j)
```



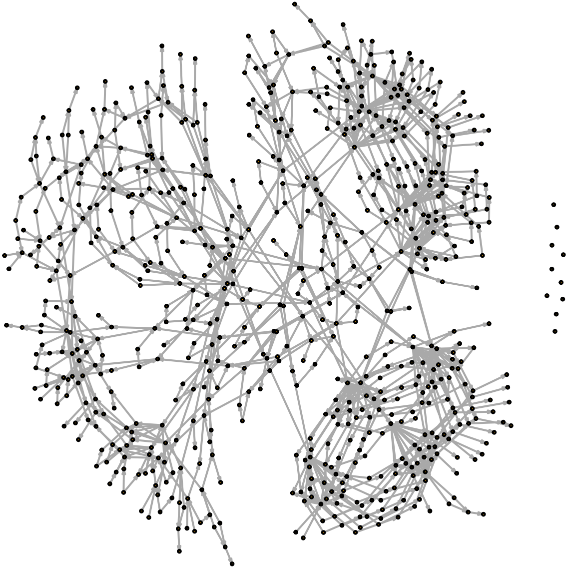
either
xray
asia
tub
either
lung
tub
bronc
either
lung
bronc
lung
smoke
bronc
dysp
either

jti

```r
jti::query_belief(j, c("smoke", "lung"), "joint")
```

```
     lung
smoke   yes    no
  yes 0.050 0.450
  no  0.005 0.495
```

link



link

```r
cl <- jti::cpt_list(jti::bnfit_to_cpts(link))
tri_min_nei  <- jti::triangulate(cl, tri = "min_nei")

# Number of cliques
length(tri_min_nei$cliques)
```

```
[1] 594
```

```r
# Amount of memory (GB) needed
sum(tri_min_nei$statespace)  * 8 / 1e9
```

```
[1] 27.37925
```

link

```r
idx_max  <- which.max(tri_min_nei$statespace)
max_vars <- tri_min_nei$cliques[[idx_max]]
max_dim  <- jti::dim_names(cl)[max_vars[1:8]]
e        <- sapply(max_dim, `[[`, 1L)
cp       <- jti::compile(cl, e, tri = "min_nei")
j        <- jti::jt(cp)
```

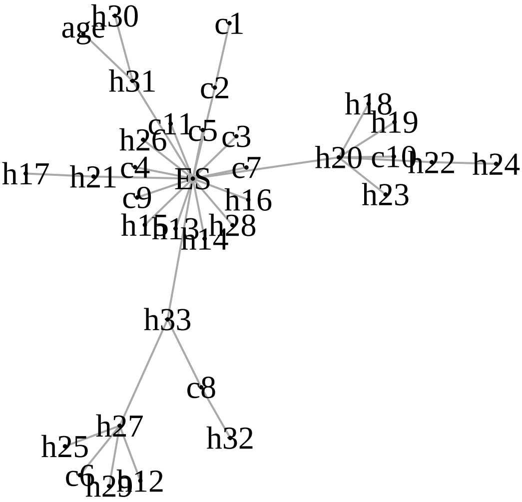
link

```
j
```

```
 Junction Tree
 -------------------------
  Propagated: full
  Flow: sum
  Nodes: 594
  Edges: 593 / 176121
  Cliques: 594
   - max: 20
   - min: 1
   - avg: 5.2
  Evidence:
   - N17_a_m: 1
   - N17_d_m: 1
   - N18_a_m: 1
   - N18_d_m: 1
   - N22_a_m: 1
```

Discrete Markov Random Fields

```r
g <- ess::fit_graph(derma, trace = FALSE)
plot(g, vertex.size = 1)
```



Discrete Markov Random Fields

```r
psor <- derma %>%
  filter(ES == "psoriasis") %>%
  select(-ES) %>%
  as_tibble()

g_psor  <- ess::fit_graph(psor, trace = FALSE)
```

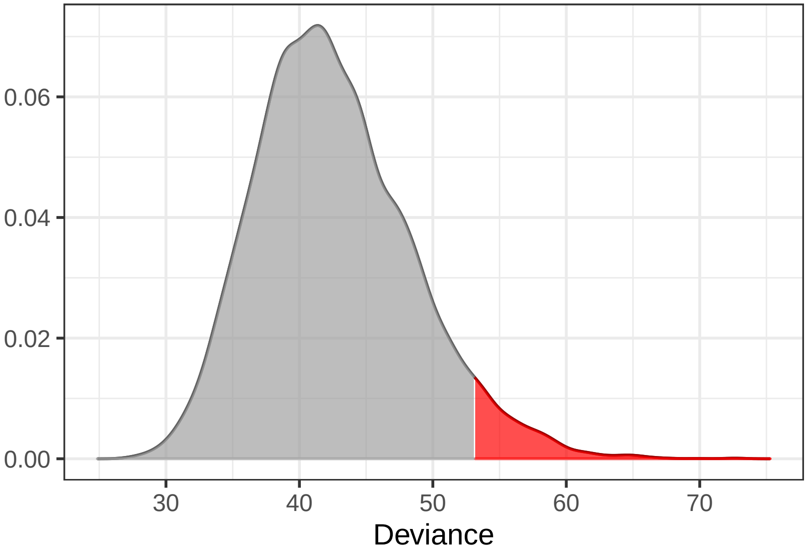
Discrete Markov Random Fields

```
m <- molic::fit_outlier(psor, g_psor); m
```

```
 ---------------------------------
  Simulations: 10000
  Variables: 34
  Observations: 111
  Estimated mean: 42.6
  Estimated variance: 33.97
 ---------------------------------
  Critical value: 53.12871
  Alpha: 0.05
  <outlier, outlier_model, list>
 ---------------------------------
```

Discrete Markov Random Fields

```
plot(m)
```



Discrete Markov Random Fields

```r
psor[which(molic::outliers(m)), ]
```

```
# A tibble: 8 x 34
  c1    c2    c3    c4    c5    c6    c7    c8    c9    c10
  <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
1 2     1     2     2     2     0     0     0     0     0
2 2     2     2     3     3     0     0     0     0     2
3 3     3     2     2     1     0     0     0     0     1
4 1     1     1     1     1     0     0     0     1     1
5 3     2     1     2     2     0     0     0     2     2
6 1     1     1     1     1     0     0     0     2     2
7 2     3     1     2     1     0     0     0     0     0
8 3     2     3     0     0     0     0     0     3     0
# ... with 21 more variables: h14 <chr>, h15 <chr>, h16 <chr>,
#   h18 <chr>, h19 <chr>, h20 <chr>, h21 <chr>, h22 <chr>, h2
#   h24 <chr>, h25 <chr>, h26 <chr>, h27 <chr>, h28 <chr>, h2
#   h30 <chr>, h31 <chr>, h32 <chr>, h33 <chr>, age <chr>
```

Software

All packages on CRAN:

- https://github.com/mlindsk/sparta
- https://github.com/mlindsk/jti
- https://github.com/mlindsk/ess
- https://github.com/mlindsk/molic