Integer programming for Bayesian network structure learning

James Cussens, University of York

Paris, 2014-04-10

(Vanilla) BN learning using integer programming

- **Have** observations of random variables V in some dataset.
- **Want** to learn an 'optimal' Bayesian Network for some *decomposable* score.

(Vanilla) BN learning using integer programming

- **Have** observations of random variables V in some dataset.
- **Want** to learn an 'optimal' Bayesian Network for some *decomposable* score.
- Can encode any graph by creating a binary IP variable $I(u \leftarrow W)$ for each BN variable $u \in V$ and each candidate parent set W.

$$\begin{tikzcd}
0 \arrow[rr] \arrow[dr] & & 1 \arrow[dl] \\
& 2 &
\end{tikzcd}$$

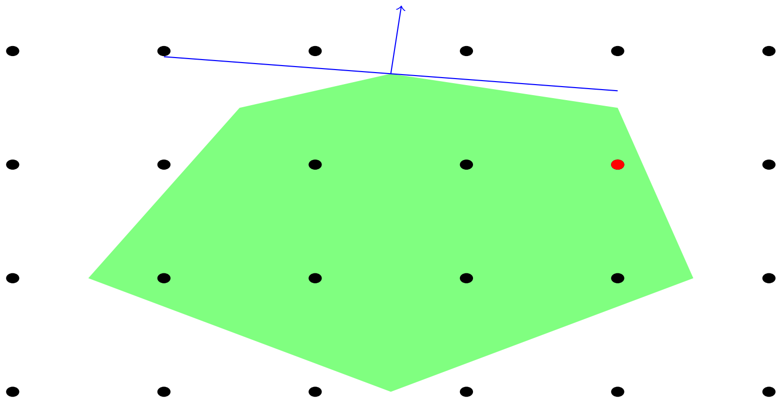
- $I(0 \leftarrow \emptyset) = 1$
- $I(1 \leftarrow \{0\}) = 1$
- $I(2 \leftarrow \{0, 1\}) = 1$
- All other IP variables zero.

(Vanilla) BN learning using integer programming

- **Have** observations of random variables V in some dataset.
- **Want** to learn an ‘optimal’ Bayesian Network for some *decomposable* score.
- Can encode any graph by creating a binary IP variable $I(u \leftarrow W)$ for each BN variable $u \in V$ and each candidate parent set W.
- Since the score is *decomposable*, each $I(u \leftarrow W)$ has an (assumed precomputed) *local score* $c(u, W)$.

Instantiate the $I(u \leftarrow W)$ to maximise:
$\sum_{u,W} c(u, W) I(u \leftarrow W)$
subject to linear constraints ensuring that the $I(u \leftarrow W)$ represent a DAG.

Solving the linear relaxation of an IP



- $x = 4, y = 2$ is the optimal integer solution.
- $x = 2.5, y = 2.8$ is the solution to the linear relaxation.
- Linear relaxation can be solved quickly, and provides an upper bound.

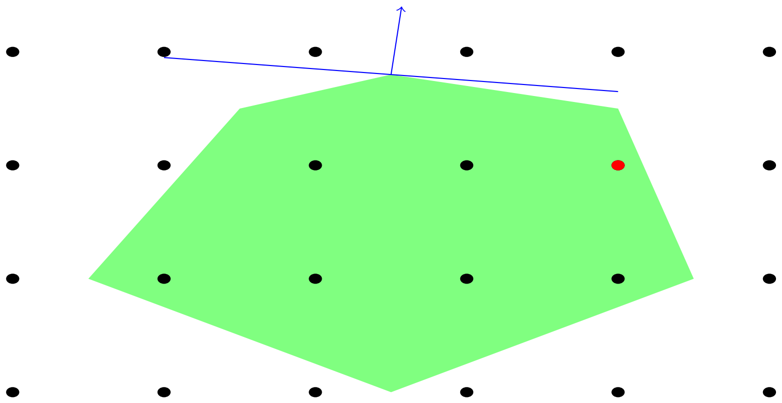
Ruling out non-DAGs with linear constraints

$$\forall u \in V : \sum_{W} I(u \leftarrow W) = 1$$

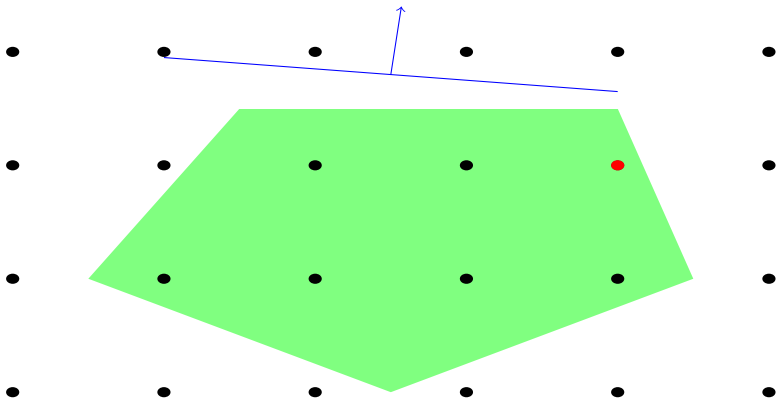
$$\text{Where } C \subseteq V : \sum_{u \in C} \sum_{W : W \cap C = \emptyset} I(u \leftarrow W) \geq 1 \tag{1}$$

- Let x^* be the solution to the linear relaxation (LP). We search for a cluster C such that x^* violates (1) and then add (1) to get a new LP.
- Repeat as long as a 'cluster cut' can be found.
- Cluster constraints introduced by Jaakkola, Sontag, Globerson and Meila (AISTATS2010) [Jaakkola et al., 2010].

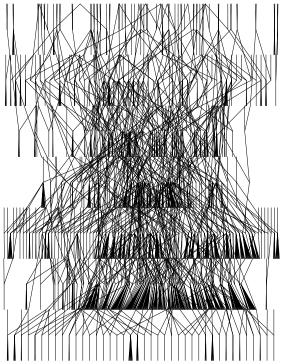
Separating the LP solution with a cutting plane



Separating the LP solution with a cutting plane



Learning very large pedigrees



- This is the structure of a real pedigree relating 1614 individuals from North West Greenland.
- Can find an optimal (maximum likelihood) pedigree from data simulated from this pedigree in between 3 and 42 minutes.
- Much faster if you assume non-founders have exactly 2 known parents.

Solving in the root node

Eskimo pedigree. 1614 BN variables. At most 2 parents. Simulated genotypes. 11934 IP variables.

```
 time |frac|cuts |   dualbound     | primalbound    |   gap
 1110s|120 | 661 | -3.162149e+04 |-4.616035e+04 |  45.98%
 1139s|118 | 669 | -3.162175e+04 |-4.616035e+04 |  45.98%
 1171s| 94 | 678 | -3.162213e+04 |-4.616035e+04 |  45.97%
 1209s| 26 | 684 | -3.162220e+04 |-4.616035e+04 |  45.97%
 1228s|103 | 685 | -3.162223e+04 |-4.616035e+04 |  45.97%
 1264s|  0 | 692 | -3.162234e+04 |-4.616035e+04 |  45.97%
*1266s|  0 |   - | -3.162234e+04 |-3.162234e+04 |   0.00%

SCIP Status        : problem is solved [optimal solution found
Solving Time (sec) : 1266.40
```

Solving after branching

Alarm. 37 BN variables. At most 3 parents. 10000 datapoints. 6473 IP variables.

```
 time | node   | left   |frac |strbr|   gap
 19.5s|      1 |      0 | 198 |  64 |   0.53%
 20.1s|      1 |      2 | 198 |  79 |   0.53%
 20.4s|      2 |      3 |  87 |  94 |   0.52%
 20.5s|      3 |      2 |   - |  94 |   0.52%
 21.2s|      4 |      3 |  18 | 172 |   0.52%
R21.3s|      5 |      2 |  90 | 172 |   0.51%
....               .....
 63.3s|   1726 |      1 |   - |1261 |   0.04%
 63.3s|   1727 |      0 |   - |1261 |   0.00%

SCIP Status        : problem is solved [optimal solution founc
Solving Time (sec) : 63.34
```

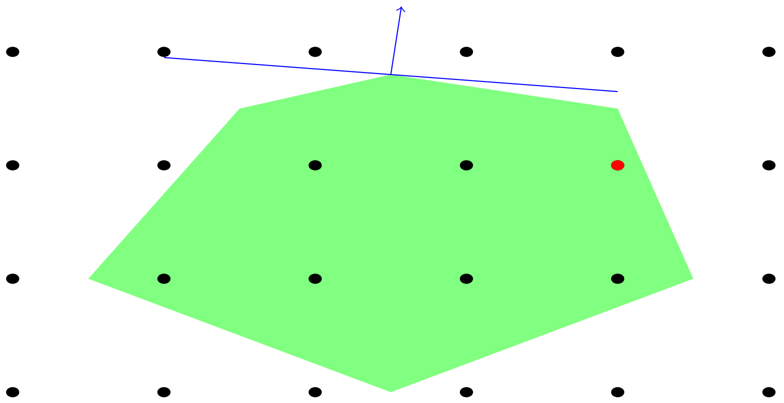
Adding constraints

- Adding in structural constraints is facilitated by the constrained optimisation approach.
- Banning/insisting on certain edges, ruling out particular immoralities, or all immoralities (all these linear).
- We use linear constraints to rule out specific BNs, allowing k-best learning
- Also allow conditional independence constraints . . .
- . . . This will be efficiently done in the next release!

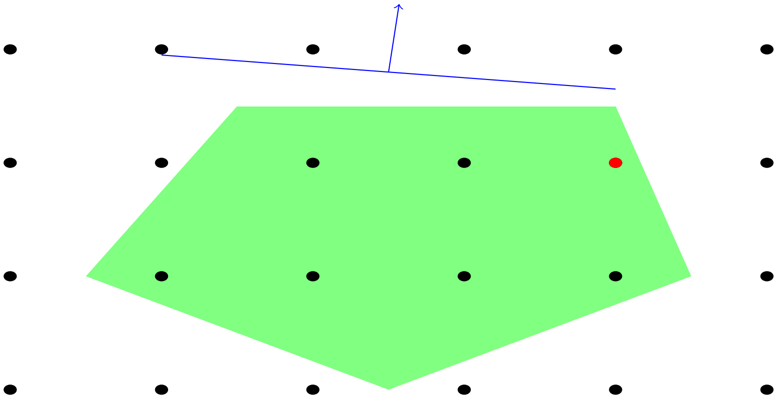
Cutting plane approach

- **Generate** a point outside the convex hull.
- **Separate** the point: find a valid inequality it violates.
- Usually this point is the solution to the current linear relaxation.
- But is this the best point to separate?

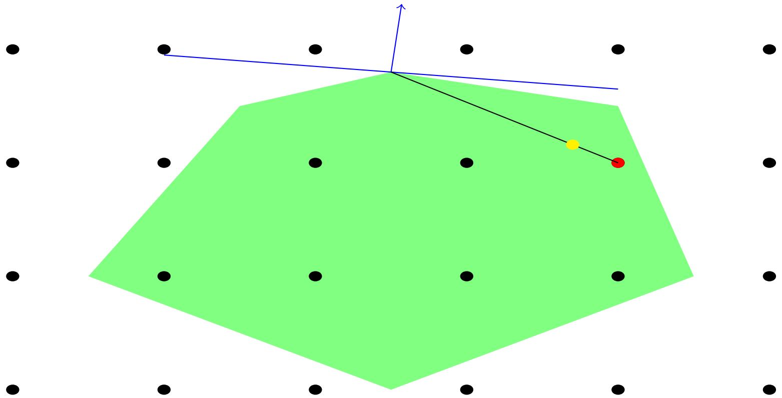
Separating the LP solution with a cutting plane



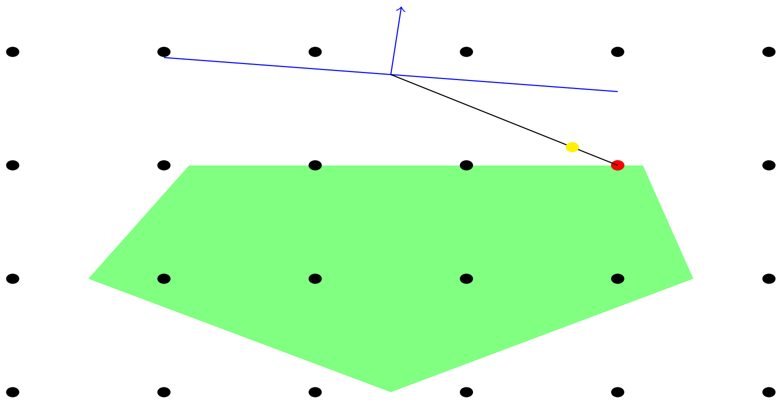
Separating the LP solution with a cutting plane



Generating a ‘close’ point to separate



Generating a ‘close’ point to separate



Exact estimation of multiple directed acyclic graphs

- Learn BNs $G^{(k)}$ for multiple related but non-identical units or ‘individuals’ $k \in \{1, 2, \ldots, K\}$.
- Improve robustness, reduce small sample bias.
- *Exchangeable learning*: Penalise structural difference of the BNs for any pair of individuals.
- *Non-exchangeable learning*: Penalise structural difference of the BNs only for *related* individuals. And learn which are related.
- Have used simulated and fMRI data.

Learning scenario

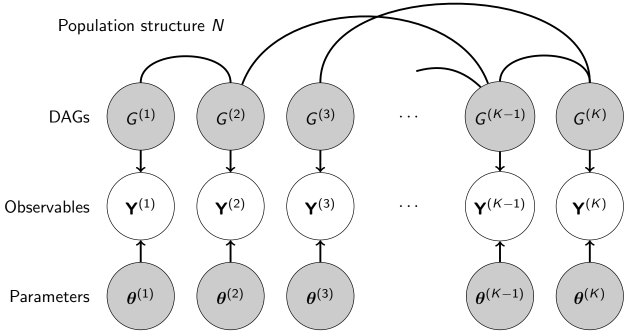


Figure : Multiple directed, acyclic graphical models (DAGs) with population structure encoded by an undirected network N. [Shaded nodes are unobserved. $G^{(1:K)}$ = data-generating graphs, $\boldsymbol{\theta}^{(1:K)}$ = data-generating parameters, $\mathbf{Y}^{(1:K)}$ = observation vectors.]

From [Oates et al., 2014]

Exchangeable learning

To decide whether individuals k, l agree on a specific edge (j, i), we introduce additional variables:

$$d^{(k,l)}(j,i) = \mathbb{I}\{j \in G_i^{(k)} \Delta G_i^{(l)}\} \quad \forall i, j, k, l \text{ with } k < l$$

Nonlinear to linear

$$\begin{array}{ccccc}
+d^{(k,l)}(j,i) & -e^{(k)}(j,i) & -e^{(l)}(j,i) & \leq & 0 \\
-d^{(k,l)}(j,i) & +e^{(k)}(j,i) & -e^{(l)}(j,i) & \leq & 0 \\
-d^{(k,l)}(j,i) & -e^{(k)}(j,i) & +e^{(l)}(j,i) & \leq & 0 \\
+d^{(k,l)}(j,i) & +e^{(k)}(j,i) & +e^{(l)}(j,i) & \leq & 2
\end{array}$$

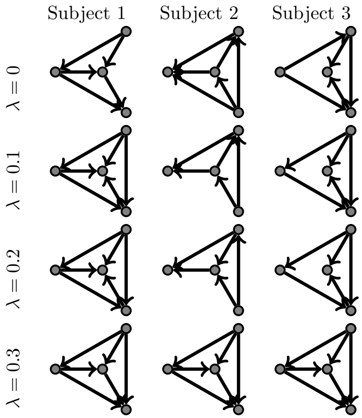
- SCIP does this linearisation for us.

Exchangeable learning

The MAP estimate $\hat{G}^{(1:K)}$ is the solution of the integer linear program

$$\begin{aligned}\hat{G}^{(1:K)} &= \arg\max_{G^{(1:K)} \in \mathcal{G}^K} \sum_{k=1}^{K} \sum_{i=1}^{P} \sum_{\pi \subseteq \{1:P\} \setminus \{i\}} s^{(k)}(i,\pi)\Pi^{(k)}(i,\pi) \\ & \quad -\lambda \sum_{(k,l)\in N} \sum_{i=1}^{P} \sum_{j=1}^{P} d^{(k,l)}(j,i) \\ & \quad \textit{subject to certain constraints}\end{aligned}$$

Exchangeable learning



Non-exchangeable learning

$$D^{(k,l)}(j,i) = \mathbb{I}\{j \in G_i^{(k)} \Delta G_i^{(l)} \text{ and } (k,l) \in N\} \quad \forall i,j,k,l.$$

Non-linear to linear

$$\begin{array}{cccccc} +D^{(k,l)}(j,i) & -E^{(k,l)} & & \leq & 0 \\ +D^{(k,l)}(j,i) & & -d^{(k,l)}(j,i) & \leq & 0 \\ -D^{(k,l)}(j,i) & +E^{(k,l)} & +d^{(k,l)}(j,i) & \leq & 1. \end{array}$$

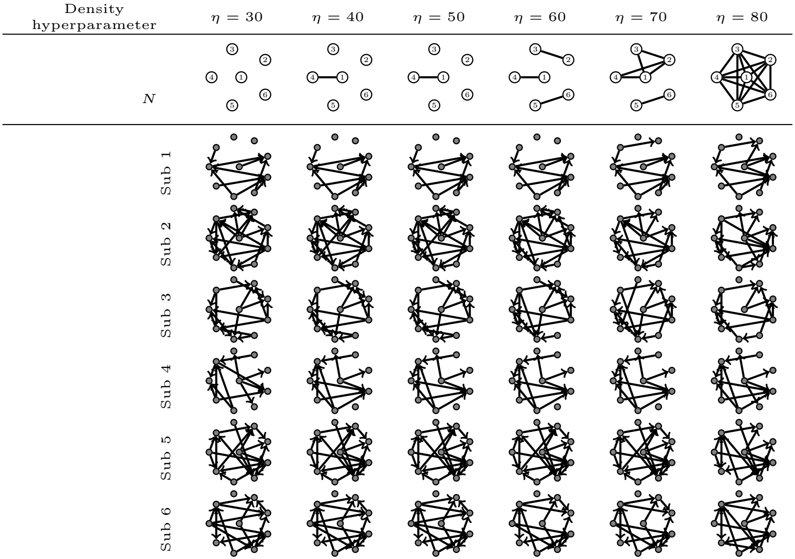
- SCIP does this linearisation for us.

Non-exchangeable learning

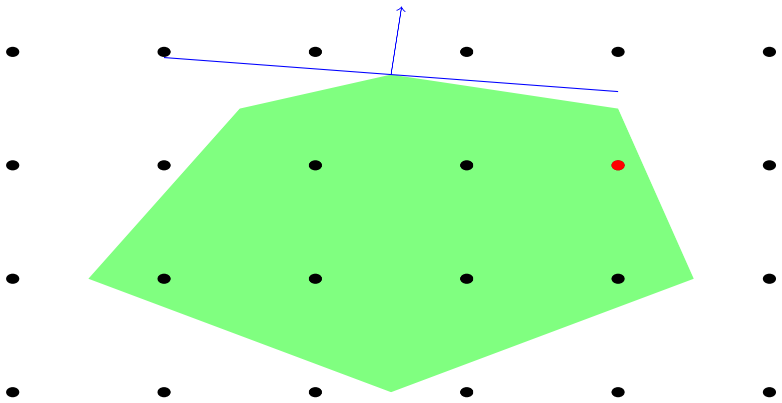
The MAP estimate $(\hat{G}^{(1:K)}, \hat{N})$ is the solution of the integer linear program

$$
\begin{aligned}
(\hat{G}^{(1:K)}, \hat{N}) \quad := \quad & \operatorname*{arg\,max}_{G^{(1:K)} \in \mathcal{G}^K, N \in \mathcal{N}} \sum_{k=1}^{K} \sum_{i=1}^{P} \sum_{\pi \subseteq \{1:P\}} s^{(k)}(i,\pi)\Pi^{(k)}(i,\pi) \\
& -\lambda \sum_{i=1}^{P} \sum_{j=1}^{P} \sum_{k=1}^{K} \sum_{l=k+1}^{K} D^{(k,l)}(j,i) + \eta \sum_{k=1}^{K} \sum_{l=k+1}^{K} E^{(k,l)} \\
& \textit{subject to certain constraints}
\end{aligned}
$$

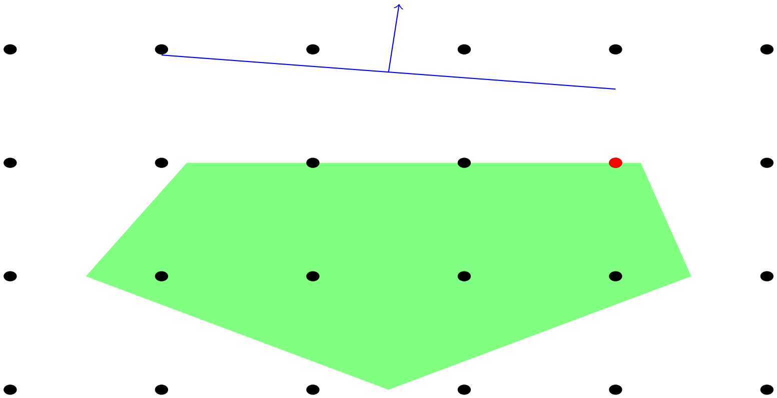
Non-exchangeable learning



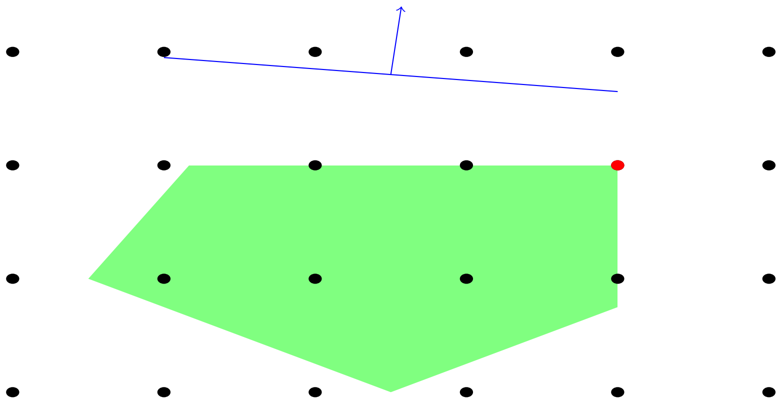
Facets: Not all cuts are equal



Facets: Not all cuts are equal



Facets: Not all cuts are equal



Finding facets

- Cluster constraints [Jaakkola et al., 2010] are facets of the convex hull of DAGs (unpublished proof)
- *k*-cluster constraints [Cussens, 2011] are also facets (unpublished proof).
- How to find more?

Generalising from small examples

- In our (full-dimensional) representation each of the 543 4-node DAGs is a point in $\mathbb{R}^{28}$, since $28 = 4 \times (2^3 - 1)$.
- Matti Järvisalo used Fukuda's `cdd` to find (in 1 second!) all 135 facets of the convex hull of these 543 points.
- These include
  - lower bounds on variables,
  - (modified) convexity constraints
  - (k)-cluster inequalities
- And many inequalities (i.e. constraints) we did not know about . . .

Facets not always easy to interpret

6 facets like this:

$$\begin{gathered}
I(a \leftarrow c) + I(a \leftarrow d) + I(a \leftarrow b, c) + I(a \leftarrow b, d) + I(a \leftarrow c, d) + 2I(a \leftarrow b, c, d) \\
+ I(b \leftarrow c) + I(b \leftarrow d) + I(b \leftarrow a, c) + I(b \leftarrow a, d) + I(b \leftarrow c, d) + 2I(b \leftarrow a, c, d) \\
+ I(c \leftarrow a) + I(c \leftarrow b) + I(c \leftarrow d) + 2I(c \leftarrow a, b) + I(c \leftarrow a, d) + I(c \leftarrow b, d) \\
+ 2I(c \leftarrow a, b, d) \\
+ I(d \leftarrow a) + I(d \leftarrow b) + I(d \leftarrow c) + 2I(d \leftarrow a, b) + I(d \leftarrow a, c) + I(d \leftarrow b, c) \\
+ 2I(d \leftarrow a, b, c) \leq 4
\end{gathered}$$

Lifting facets - an example

This facet for the convex hull of 3-node DAGs

$$I(a \leftarrow b, c) + I(b \leftarrow a, c) + I(c \leftarrow a, b) \leq 1$$

can be 'lifted' to this facet for 4-node DAGs:

$$\begin{aligned} I(a \leftarrow b, c) + I(a \leftarrow b, c, d) + I(b \leftarrow a, c) + I(b \leftarrow a, c, d) \\ + I(c \leftarrow a, b) + I(c \leftarrow a, b, d) \leq 1 \end{aligned}$$

- It is possible to lift facets *of any type* in this way (unpublished proof).
- So the 4-node DAG facets provide facets for *any* BN learning problem (with more than 4 nodes).

Not all facets are equal

$$I(a \leftarrow b,c) + I(a \leftarrow b,c,d) + I(b \leftarrow a,c) + I(b \leftarrow a,c,d) \\ + I(c \leftarrow a,b) + I(c \leftarrow a,b,d) \leq 1$$

- This facet is *score-equivalent* . . .
- . . . if two BNs are Markov equivalent then the LHS of the facet is the same for both BNs.
- 66 4-node facets are **not** score-equivalent.
- Score-equivalent facets are better since only they can help form the *optimal face*.

Facet finding in practice

- Latest version of GOBNILP, searches for (lifted) 4-node facets which separate a given LP solution.
- Some facets added into problem at the beginning.
- It does indeed help sometimes dramatically!
- And yes, the score-equivalent facets help more.

Column generation (what we're doing wrong)

- Just as one can create constraints on the fly (cutting planes) one can also create variables dynamically (column generation).
- Think of the not-currently-created variables as being initially fixed to zero.
- After solving the LP search for a parent set with a positive *reduced* local score. If we can't find one we have all the variables we need for an optimal solution.
- For the reduced local score we need (i) the (unreduced) local score (ii) dual values for all the linear constraints in which it will appear and (iii) its coefficient in each of these linear constraints.
- A lot of work but some hope of scaling up for more general-purpose BN structure learning.

Acknowledgements

Joint work with:

- Mark Bartlett (University of York)
- Milan Studený (Czech Academy of Sciences, Prague)
- Matti Järvisalo (University of Helsinki)
- Chris Oates, Jim Smith (University of Warwick)
- Sach Mukherjee (MRC Biostatistic Unit & CRUK Cambridge Institute)

Supported by the UK Medical Research Council (Project Grant G1002312)

Cussens, J. (2011).
Bayesian network learning with cutting planes.
In Cozman, F. G. and Pfeffer, A., editors, *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI 2011)*, pages 153–160, Barcelona. AUAI Press.

Jaakkola, T., Sontag, D., Globerson, A., and Meila, M. (2010).
Learning Bayesian network structure using LP relaxations.
In *Proceedings of 13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, volume 9, pages 358–365. Journal of Machine Learning Research Workshop and Conference Proceedings.

Oates, C., Smith, J., Mukherjee, S., and Cussens, J. (2014).
Exact estimation of multiple directed acyclic graphs.
arkiv:1404.1238.