On the polyhedral geometry of conditional independence relations

James Cussens, University of York

Munich, 2019-10-24

The problem to be solved

- Given data *and some known/assumed conditional (CI) independence relations* find a good DAG (Bayesian network).
- In a sense, this is easy if we view DAG learning as a *constrained optimisation* problem.
- We just tell the solver to reject any DAG not satisfying the given CI relations, and keep searching.
- But this sort of 'generate-and-test' approach is woefully inefficient.
- We need some theory to help us do better . . .

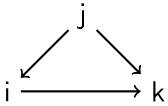
Polyhedral geometry for DAGs and Markov equivalence classes of DAGs

- We start by considering DAG learning without any CI constraints.
- And examine the geometry of a polytope that is central to *integer programming (IP)* approaches to solving this problem.
  - A key step in IP is to solve (in polynomial time) the *linear relaxation* of the original problem (where we remove all integrality constraints).
  - The solution to the linear relaxation is an optimal vertex of the polytope defined by the linear constraints of the problem.

Polyhedral geometry for DAGs and Markov equivalence classes of DAGs

- We start by considering DAG learning without any CI constraints.
- And examine the geometry of a polytope that is central to *integer programming (IP)* approaches to solving this problem.
  - A key step in IP is to solve (in polynomial time) the *linear relaxation* of the original problem (where we remove all integrality constraints).
  - The solution to the linear relaxation is an optimal vertex of the polytope defined by the linear constraints of the problem.
- This involves encoding DAGs as vectors.
- We also consider an approach where each Markov equivalence class of DAGs is encoded as a single vector.

Encoding DAGs as zero-one vectors

- To use an integer programming approach to learning DAGs from data it is necessary to encode each DAG as a vector.
- For DAG learning the most useful encoding is via *family variables*.
- This digraph: 

is this point in $\mathbb{R}^{12}$:

$x_{i\leftarrow\{\}}$	$x_{i\leftarrow\{j\}}$	$x_{i\leftarrow\{k\}}$	$x_{i\leftarrow\{j,k\}}$
0	1	0	0

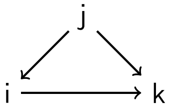
$x_{j\leftarrow\{\}}$	$x_{j\leftarrow\{i\}}$	$x_{j\leftarrow\{k\}}$	$x_{j\leftarrow\{i,k\}}$
1	0	0	0

$x_{k\leftarrow\{\}}$	$x_{k\leftarrow\{i\}}$	$x_{k\leftarrow\{j\}}$	$x_{k\leftarrow\{i,j\}}$
0	0	0	1

Most objectives (BDeu, BIC, etc) are linear functions of family variables.

Altering the encoding

- Clearly each vertex has exactly one parent set in any DAG, so we can drop the family variables indicating an empty parent set.
- If we have n nodes we end up with $n(2^{n-1} - 1)$ family variables.
- Assume this encoding from now on.



$x_{i\leftarrow\{j\}}$	$x_{i\leftarrow\{k\}}$	$x_{i\leftarrow\{j,k\}}$
1	0	0

$x_{j\leftarrow\{i\}}$	$x_{j\leftarrow\{k\}}$	$x_{j\leftarrow\{i,k\}}$
0	0	0

$x_{k\leftarrow\{i\}}$	$x_{k\leftarrow\{j\}}$	$x_{k\leftarrow\{i,j\}}$
0	0	1

The family-variable polytope

- For some fixed set of n nodes consider the set of all DAGs with those nodes.
- Each DAG corresponds to a 0-1 vector (indexed by the family variables).
- The convex hull of all these vectors is the *family-variable polytope*.
- This polytope has dimension $n(2^{n-1} - 1)$ and so our encoding is *full-dimensional*.

Facets of the family-variable polytope

- Like any polytope, the family-variable polytope can also be defined via its *facets*.
- A *face* of a polytope $P \subseteq \mathbb{R}^m$ is a set of the form

$$F := P \cap \{x \in \mathbb{R}^m \mid cx = \delta\},$$

  where $cx \leq \delta$ is a *valid inequality* for P.
- A face is *proper* if it is non-empty and properly contained in P.
- An inclusion-wise maximal proper face of P is called a *facet*.

Family-variable polytope for $n = 4$

- When $n = 4$ there are 543 DAGs.
- There are $4 \times (2^3 - 1) = 28$ family variables.
- And the family-variable polytope has 135 facets.
- 28 of the facets are defined by lower bounds on the family variables
- These lower bound facets are defined by facet-defining inequalities like this: $x_{i \leftarrow J} \geq 0$.

Some facet-defining inequalities

- Here are some other facet-defining inequalities for $n = 4$, where we assume the nodes of the DAGs are $\{a, b, c, d\}$, and write e.g. ab for $\{a, b\}$.

Even cyclic digraphs have to satisfy inequalities like this one:

$$x_{a \leftarrow b} + x_{a \leftarrow c} + x_{a \leftarrow d} + x_{a \leftarrow bc} + x_{a \leftarrow bd} + x_{a \leftarrow cd} + x_{a \leftarrow bcd} \leq 1$$

At least one of a, b and c must have no parents in $\{a, b, c\}$:

$$\begin{aligned} & x_{a \leftarrow b} + x_{a \leftarrow c} + x_{a \leftarrow bc} + x_{a \leftarrow bd} + x_{a \leftarrow cd} + x_{a \leftarrow bcd} \\ & + x_{b \leftarrow a} + x_{b \leftarrow c} + x_{b \leftarrow ac} + x_{b \leftarrow ad} + x_{b \leftarrow cd} + x_{b \leftarrow acd} \\ & + x_{c \leftarrow a} + x_{c \leftarrow b} + x_{c \leftarrow ab} + x_{c \leftarrow ad} + x_{c \leftarrow bd} + x_{c \leftarrow abd} \leq 2 \end{aligned}$$

Some more facet-defining inequalities

$$
\begin{aligned}
& x_{a\leftarrow b} + x_{a\leftarrow bc} + x_{a\leftarrow bd} + x_{a\leftarrow cd} + x_{a\leftarrow bcd} \\
& + x_{b\leftarrow a} + x_{b\leftarrow ac} + x_{b\leftarrow ad} + x_{b\leftarrow cd} + x_{b\leftarrow acd} \\
& + x_{c\leftarrow ad} + x_{c\leftarrow bd} + x_{c\leftarrow abd} \\
& + x_{d\leftarrow ac} + x_{d\leftarrow bc} + x_{d\leftarrow abc} \qquad \leq 2
\end{aligned}
$$

$$
\begin{aligned}
& x_{a\leftarrow cd} + x_{a\leftarrow bcd} \\
& + x_{b\leftarrow c} + x_{b\leftarrow ac} + x_{b\leftarrow cd} + x_{b\leftarrow acd} \\
& + x_{c\leftarrow b} + x_{c\leftarrow d} + x_{c\leftarrow ab} + x_{c\leftarrow ad} + x_{c\leftarrow bd} + 2x_{c\leftarrow abd} \\
& + x_{d\leftarrow a} + x_{d\leftarrow b} + x_{d\leftarrow c} + x_{d\leftarrow ab} + 2x_{d\leftarrow ac} + x_{d\leftarrow bc} + 2x_{d\leftarrow abc} \leq 3
\end{aligned}
$$

Empty and complete DAGs

- Recall: 28 of the facets are defined by lower bounds on the family variables: $x_{i \leftarrow J} \geq 0$.
- The vertex corresponding to the empty graph (the zero vector) is the vertex at the intersection of these 28 facets (and lies on none of the other 107 facets).
- A complete DAG, in contrast, lies on many facets.

Score equivalence

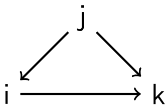
- Let $G \sim H$ denote that DAGs G and H are Markov equivalent. Let $x(G)$ and $x(H)$ be their family-variable encodings.
- A vector c is called a *score-equivalent objective* if whenever $G \sim H$ then $cx(G) = cx(H)$.
- We call a face *score-equivalent* if it is defined by a valid inequality $cx \leq \delta$ where c is a score-equivalent objective.

Theorem[CHS16]. If S is a facet of the family-variable polytope, the following conditions are equivalent:

1. S is closed under Markov equivalence.
2. S contains the whole equivalence class of complete graphs.
3. S is score equivalent.

Encoding Markov equivalence classes of DAGs as zero-one vectors

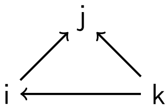
- If we want to encode each Markov equivalence classes of DAGs as a zero-one vector then we can use the *characteristic imset* encoding [SHL10].
- For any $S \subseteq N$, $|S| \geq 2$, $\mathrm{c}_G(S) = 1$ iff there is a vertex $a \in S$, such that all parents of a (in G) are also in S.
- **Fundamental fact:** G and H are Markov equivalent iff $\mathrm{c}_G = \mathrm{c}_H$.



$\mathrm{c}(ij)$	$\mathrm{c}(ik)$	$\mathrm{c}(jk)$	$\mathrm{c}(ijk)$
1	1	1	1

Encoding Markov equivalence classes of DAGs as zero-one vectors

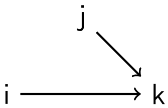
- If we want to encode each Markov equivalence classes of DAGs as a zero-one vector then we can use the *characteristic imset* encoding [SHL10].
- For any $S \subseteq N$, $|S| \geq 2$, $\mathrm{c}_G(S) = 1$ iff there is a vertex $a \in S$, such that all parents of a (in G) are also in S.
- **Fundamental fact:** G and H are Markov equivalent iff $\mathrm{c}_G = \mathrm{c}_H$.



$\mathrm{c}(ij)$	$\mathrm{c}(ik)$	$\mathrm{c}(jk)$	$\mathrm{c}(ijk)$
1	1	1	1

Encoding Markov equivalence classes of DAGs as zero-one vectors

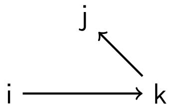
- If we want to encode each Markov equivalence classes of DAGs as a zero-one vector then we can use the *characteristic imset* encoding [SHL10].
- For any $S \subseteq N$, $|S| \geq 2$, $\mathrm{c}_G(S) = 1$ iff there is a vertex $a \in S$, such that all parents of a (in G) are also in S.
- **Fundamental fact:** G and H are Markov equivalent iff $\mathrm{c}_G = \mathrm{c}_H$.



$\mathrm{c}(ij)$	$\mathrm{c}(ik)$	$\mathrm{c}(jk)$	$\mathrm{c}(ijk)$
0	1	1	1

Encoding Markov equivalence classes of DAGs as zero-one vectors

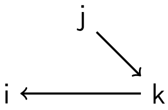
- If we want to encode each Markov equivalence classes of DAGs as a zero-one vector then we can use the *characteristic imset* encoding [SHL10].
- For any $S \subseteq N$, $|S| \geq 2$, $\mathrm{c}_G(S) = 1$ iff there is a vertex $a \in S$, such that all parents of a (in G) are also in S.
- **Fundamental fact:** G and H are Markov equivalent iff $\mathrm{c}_G = \mathrm{c}_H$.



$\mathrm{c}(ij)$	$\mathrm{c}(ik)$	$\mathrm{c}(jk)$	$\mathrm{c}(ijk)$
0	1	1	0

Encoding Markov equivalence classes of DAGs as zero-one vectors

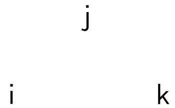
- If we want to encode each Markov equivalence classes of DAGs as a zero-one vector then we can use the *characteristic imset* encoding [SHL10].
- For any $S \subseteq N$, $|S| \geq 2$, $\mathrm{c}_G(S) = 1$ iff there is a vertex $a \in S$, such that all parents of a (in G) are also in S.
- **Fundamental fact:** G and H are Markov equivalent iff $\mathrm{c}_G = \mathrm{c}_H$.



$\mathrm{c}(ij)$	$\mathrm{c}(ik)$	$\mathrm{c}(jk)$	$\mathrm{c}(ijk)$
0	1	1	0

Encoding Markov equivalence classes of DAGs as zero-one vectors

- If we want to encode each Markov equivalence classes of DAGs as a zero-one vector then we can use the *characteristic imset* encoding [SHL10].
- For any $S \subseteq N$, $|S| \geq 2$, $\mathrm{c}_G(S) = 1$ iff there is a vertex $a \in S$, such that all parents of a (in G) are also in S.
- **Fundamental fact:** G and H are Markov equivalent iff $\mathrm{c}_G = \mathrm{c}_H$.



c(ij)	c(ik)	c(jk)	c(ijk)
0	0	0	0

A linear projection between the two representations

$$\mathrm{c}(S) = \sum_{a \in S} \sum_{B : S \setminus \{a\} \subseteq B \subseteq N \setminus \{a\}} x_{a \leftarrow B} \qquad \text{for any } S \subseteq N,\ |S| \geq 2.$$

Characteristic imset polytope

- The *characteristic imset polytope* is the convex hull of all characteristic imset vectors.
- It has dimension $2^n - n - 1$.
- The characteristic imset polytope is the image of the family-variable polytope by the linear map on the preceding slide.
- When $n = 4$, the characteristic imset polytope is of dimension 11, has 185 vertices and 154 facets.
- So it has 358 fewer vertices but 19 more facets than the family variable polytope for $n = 4$.

Matroids define facets

This score-equivalent facet-defining inequality for the family-variable polytope:

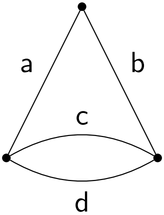
$$\begin{aligned} & x_{a\leftarrow b} + x_{a\leftarrow bc} + x_{a\leftarrow bd} + x_{a\leftarrow cd} + x_{a\leftarrow bcd} \\ & + x_{b\leftarrow a} + x_{b\leftarrow ac} + x_{b\leftarrow ad} + x_{b\leftarrow cd} + x_{b\leftarrow acd} \\ & + x_{c\leftarrow ad} + x_{c\leftarrow bd} + x_{c\leftarrow abd} \\ & + x_{d\leftarrow ac} + x_{d\leftarrow bc} + x_{d\leftarrow abc} \qquad \leq 2 \end{aligned}$$

corresponds to this facet for the characteristic imset polytope:

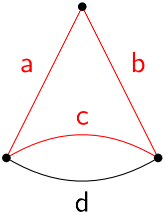
$$\mathrm{c}(abc) + \mathrm{c}(abd) + \mathrm{c}(cd) - \mathrm{c}(abcd) \leq 2$$

- And both correspond to a *matroid* with $\{a, b, c, d\}$ as the ground set and this set of *circuits*: $\{abc, abd, cd\}$.
- Every connected matroid generates a score-equivalent facet for both the family-variable and characteristic imset polytope [Stu15].

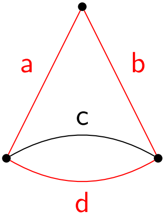
A (graphical) matroid



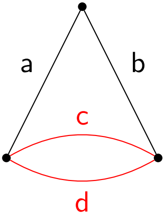
A (graphical) matroid



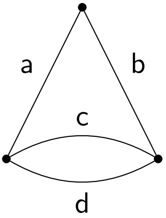
A (graphical) matroid



A (graphical) matroid



A (graphical) matroid



- *Circuits* (minimal *dependent sets*) are $\mathcal{C} = \{abc, abd, cd\}$
- *Bases* (maximal *independent sets*) are $\mathcal{B} = \{ab, ac, ad, bc, bd\}$
- *Rank* is 2
- Matroid is *connected*: every pair of elements in some circuit.

Not all matroids have a graphical representation!

Decomposable models

- If we restrict ourselves to decomposable models and encode using characteristic imsets, we get the *chordal graph polytope*.
- In the case of decomposable models we have $\mathrm{c}_G(S) = 1$ iff S is a complete set in the chordal graph.
- There is a conjecture [SC17] that the set of facets of the chordal graph polytope (for node set N) is in one-one correspondence with the set of *clutters* of subsets of N containing at least one singleton (plus the lower bound $\mathrm{c}(N) \geq 0$).
- True up to $n = 5$ (where this polytope has 822 vertices and 682 facets).
- The complete graph (saturated model) lies on all facets except the one defined by $\mathrm{c}(N) \geq 0$.

Example clutter inequalities

- From clutter $\mathcal{L} = \{\{a, b, c\}, \{d\}\}$ we get this facet-defining inequality

  $$\mathrm{c}(abc) \leq \mathrm{c}(abcd)$$

  . (monotonicity)
- For clutter $\mathcal{L} = \{\{a, b\}, \{a, c\}, \{b, c\}, \{d\}\}$ we get this facet-defining inequality

  $$\mathrm{c}(abd)+\mathrm{c}(acd)+\mathrm{c}(bcd)-2\cdot\mathrm{c}(abcd) \ \leq \ \mathrm{c}(ab)+\mathrm{c}(ac)+\mathrm{c}(bc)-2\cdot\mathrm{c}(abc)$$

  (generalised monotonicity)

Clutter inequalities and junction trees

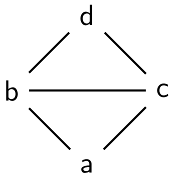
- If $\mathcal{L}$ is a clutter (containing at least one singleton $\{a\}$) then $\mathcal{L}^{\uparrow}$ denotes the *filter* of all supersets of members of $\mathcal{L}$.
- Let G be a chordal graph and let $C_1, \dots C_m$ be an ordering of its (maximal) cliques satisfying the *running intersection property* where $a \in C_1$. Let $S_2, \dots S_m$ be the separators.
- Then the clutter inequality for $\mathcal{L}$ 'says':

$$\sum_{j=1}^{m} \delta(C_j \in \mathcal{L}^{\uparrow}) - \sum_{j=2}^{m} \delta(S_j \in \mathcal{L}^{\uparrow}) \geq 1$$

$$\Leftrightarrow \quad \delta(C_1 \in \mathcal{L}^{\uparrow}) + \sum_{j=2}^{m} \delta(C_j \in \mathcal{L}^{\uparrow}) - \delta(S_j \in \mathcal{L}^{\uparrow}) \geq 1$$

Incomplete graphs and clutters

- If a chordal graph is not complete then there is a non-empty set of clutter inequality facets that it does **not** lie on.
- For example, this chordal graph:



- where $C_1 = \{a, b, c\}$, $C_2 = \{b, c, d\}$ and $S_2 = \{b, c\}$
- does not lie on the facet defined by this clutter $\mathcal{L} = \{\{a\}, \{b, c, d\}\}$.
- (If we removed the edge between, say, b and d then the resulting graph would lie on the facet.)

What to do to (facet-defining) inequalities when CI constraints are given?

- Given CI constraints, we can
  1. Require that solutions lie on certain facets, and/or
  2. Remove certain facet-defining inequalities, and/or
  3. Tighten certain facet-defining inequalities.
- A more ambitious approach (not done here) would be to characterise the polytope that arises from the convex hull of all DAGs (or MECs) satisfying the given CI relations.

Enforcing tight lower bounds

- Clearly if $A \perp B|S$ is required and $a \in A$ and $b \in B$ then we set e.g. $x_{a \leftarrow bc} = 0$.
- If $A \perp B|S$ is required and $a \in A$, $b \in B$ and $c \in S$ then we set e.g. $x_{c \leftarrow ab} = 0$.
- So we end up with a lower-dimensional polytope by requiring that solutions lie on certain facets.

Disregarding redundant inequalities

- If we have $a \perp b$ and $a \perp c$ and tighten the relevant lower bounds then this inequality:

$$
\begin{aligned}
& x_{a \leftarrow b} + x_{a \leftarrow c} + x_{a \leftarrow bc} + x_{a \leftarrow bd} + x_{a \leftarrow cd} + x_{a \leftarrow bcd} \\
+ & x_{b \leftarrow a} + x_{b \leftarrow c} + x_{b \leftarrow ac} + x_{b \leftarrow ad} + x_{b \leftarrow cd} + x_{b \leftarrow acd} \\
+ & x_{c \leftarrow a} + x_{c \leftarrow b} + x_{c \leftarrow ab} + x_{c \leftarrow ad} + x_{c \leftarrow bd} + x_{c \leftarrow abd} \leq 2
\end{aligned}
$$

- will always be satisfied (due to other inequalities) and so there is no put adding it.
- (Typically such inequalities are added as *cutting planes* so are disregarded ‘automatically’.)

Tightening (formerly) facet-defining inequalities

Inequalities like this are called *cluster constraints* [JSGM10]:

$$
\begin{aligned}
& x_{a\leftarrow b} + x_{a\leftarrow c} + x_{a\leftarrow bc} + x_{a\leftarrow bd} + x_{a\leftarrow cd} + x_{a\leftarrow bcd} \\
+ & x_{b\leftarrow a} + x_{b\leftarrow c} + x_{b\leftarrow ac} + x_{b\leftarrow ad} + x_{b\leftarrow cd} + x_{b\leftarrow acd} \\
+ & x_{c\leftarrow a} + x_{c\leftarrow b} + x_{c\leftarrow ab} + x_{c\leftarrow ad} + x_{c\leftarrow bd} + x_{c\leftarrow abd} \leq 2
\end{aligned}
$$

- This inequality corresponds to the (rank 1) matroid with these circuits: $\{ab, ac, bc\}$.
- If a DAG lies on the facet (the LHS=2) then one of $\{a, b, c\}$ is a common ancestor of the other two.
- So if it is required that, say, $a \perp b$ then we can tighten this facet-defining inequality to $\cdots \leq 1$.

Conditional independence constraints for chordal graphs

- If chordal graph G satisfies $a \perp b|S$, and
- $\mathcal{L}$ is a clutter where
  1. $\{a\} \in \mathcal{L}$
  2. $S \notin \mathcal{L}^{\uparrow}$
  3. $\{b\} \cup S \in \mathcal{L}^{\uparrow}$
- then c_G does not lie on the facet defined by $\mathcal{L}$.
- So if we require that $a \perp b|S$, then we can tighten the relevant clutter inequalities.

Conclusions

- As we add conditional independence constraints the polytope (family-variable or characteristic imset) shrinks towards the vertex that is the empty graph.
- But clearly a lot more work to be done!

James Cussens, David Haws, and Milan Studený.
Polyhedral aspects of score equivalence in Bayesian network structure learning.
Mathematical Programming, 2016.
doi:10.1007/s10107-016-1087-2.

Tommi Jaakkola, David Sontag, Amir Globerson, and Marina Meila.
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, 2010.
Journal of Machine Learning Research Workshop and Conference Proceedings.

Milan Studený and James Cussens.
Towards using the chordal graph polytope in learning decomposable models.
International Journal of Approximate Reasoning, 88:259–281, 2017.

Milan Studený, Raymond Hemmecke, and Silvia Lindner.
Characteristic imset: a simple algebraic representative of a Bayesian network structure.
In Petri Myllymäki, Teemu Roos, and Tommi Jaakkola, editors, *Proceedings of the Fifth European Workshop on Probabilistic Graphical Models (PGM-2010)*, 2010.

Milan Studený.
How matroids occur in the context of learning Bayesian network structure.
In Marina Meila and Tom Heskes, editors, *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence (UAI 2015)*, pages 832–841. AUAI Press, 2015.