Stable mixed graphs

Kayvan Sadeghi

Department of Statistics

University of Oxford

April 2012

Introduction

- Graphical Markov models use graphs to capture conditional independence statements of sets of random variables.

- Nodes of the graph correspond to random variables and edges to dependencies.

- Unobserved variables are related to Marginalisation and selection variables to conditioning.

Independence model

independence model $\mathcal{J}$ over a set V: a set of triples $\langle X, Y | Z \rangle$ where $X, Y, Z \subset V$.

$\langle X, Y | Z \rangle$ interpreted as "X is independent of Y given Z".

A graph induces an independence model by the use of a **separation criterion**.

Example: Probabilistic conditional independence:

$$\langle X, Y | Z \rangle \in \mathcal{J}_P \quad \Leftrightarrow \quad X \perp\!\!\!\perp Y \,|\, Z \quad \Leftrightarrow$$

$$f_{XYZ}(x, y, z) = \frac{f_{XZ}(x,z) f_{YZ}(y,z)}{f_Z(z)}.$$

Marginal and Conditional independence models

independence model $\mathcal{J}$ after marginalisation over M:

$$\alpha(\mathcal{J}; M, \varnothing) = \{\langle A, B \mid D\rangle \in \mathcal{J} : (A \cup B \cup D) \cap M = \varnothing\}$$

independence model after conditioning on C:

$\alpha(\mathcal{J}; \varnothing, C) = \{\langle A, B \mid D\rangle : \langle A, B \mid D \cup C\rangle \in$
$\mathcal{J}$ and $(A \cup B \cup D) \cap C = \varnothing\}$.

independence model after marginalisation over M and conditioning on C:

$\alpha(\mathcal{J}; M, C) = \{\langle A, B \mid D\rangle : \langle A, B \mid D \cup C\rangle \in$
$\mathcal{J}$ and $(A \cup B \cup D) \cap (M \cup C) = \varnothing\}$

Stability

$\mathcal{T}$: a family of graphs

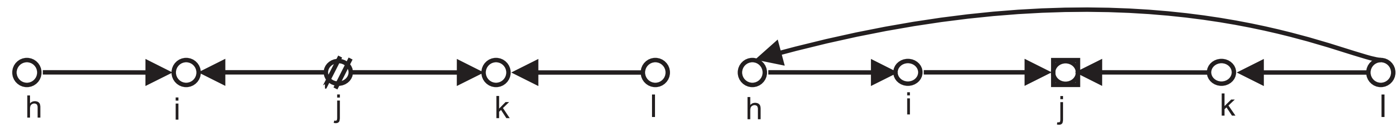
$\mathcal{J}^{\mathcal{T}} = \{\mathcal{J}^G\}_{G \in \mathcal{T}}$: a family of independence models

$\mathcal{T}$ **stable** (under marginalisation and conditioning) with respect to $\mathcal{J}^{\mathcal{T}}$:

For $G = (V, E) \in \mathcal{T}$ and $M, C \subset V$:
$\exists H \in \mathcal{T}$ s.t. $\mathcal{J}^H = \alpha(\mathcal{J}^G; M, C)$.

Stability of DAGs using d-separation

DAGs are **not stable**:



(a) (b)

(a) A DAG, which shows DAGs are not stable under marginalisation. ($\not\bigcirc \in M$.) (b) A DAG, which shows DAGs are not stable under conditioning. ($\boxed{\bigcirc} \in C$.)

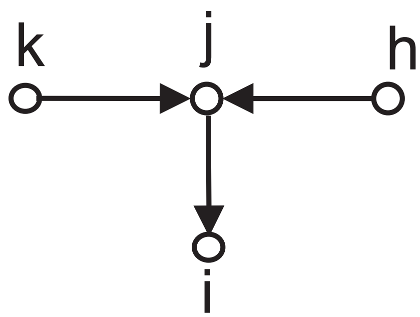
Stable Mixed graphs

- **Mixed graph** : a graph containing three types of edges denoted by arrows $\succ\!\!\rightarrow$, arcs $\prec\!\!\succ$, and lines $—$.
- Multiple edges of different types allowed, multiple edges of the same type not allowed $\Rightarrow$ Up to four edges as a multiple edge between any two nodes.
- Mixed graphs contain DAGs
- We look for **stable** subclasses of mixed graphs:
  MC (Koster 2002), Ancestral (Richardson and Spirtes 2002), Summary (Wermuth 2011), and Ribbonless graphs.

The m-separation

- **m-connecting path given C** : all its collider nodes are in $C \cup \mathrm{an}(C)$ and all its non-collider nodes are outside C.
- $A \perp_m B \mid C$ if there is no m-connecting path between A and B given C.

Example.



$j \in \mathrm{ant}(i) \Rightarrow \langle k, j, h \rangle$ m-connecting given i

$\Rightarrow k \perp_m h \mid i$ does not hold.

Ribbonless graphs

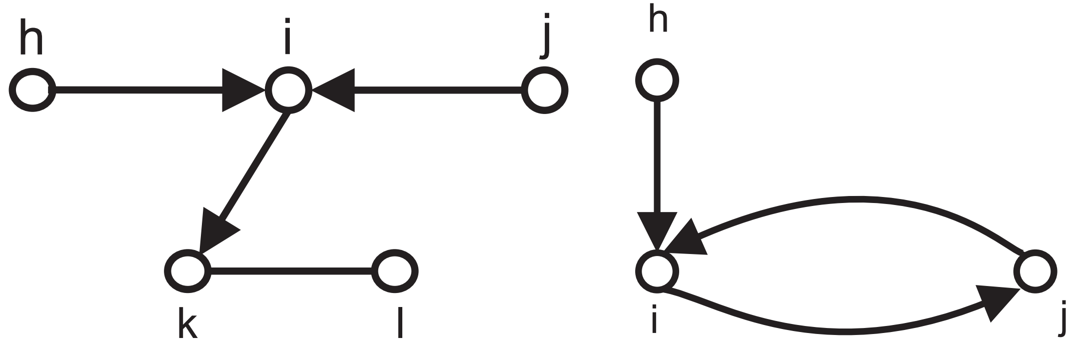
A **ribbon** is a graph containing a collider V-configuration $\langle h, i, j \rangle$ s.t.

1. no $j \leftrightarrow h$ if $h \leftrightarrow i \leftrightarrow j$; no $j \;\frac{\quad}{}\; h$ if $h \rightarrow i \leftarrow j$; no $h \rightarrow j$ if $h \rightarrow i \leftrightarrow j$;

2. i or a descendant of i is the endpoint of a line or on a direction-preserving cycle.

Ribbonless graph (RG): an LMG that does not contain ribbons.

example



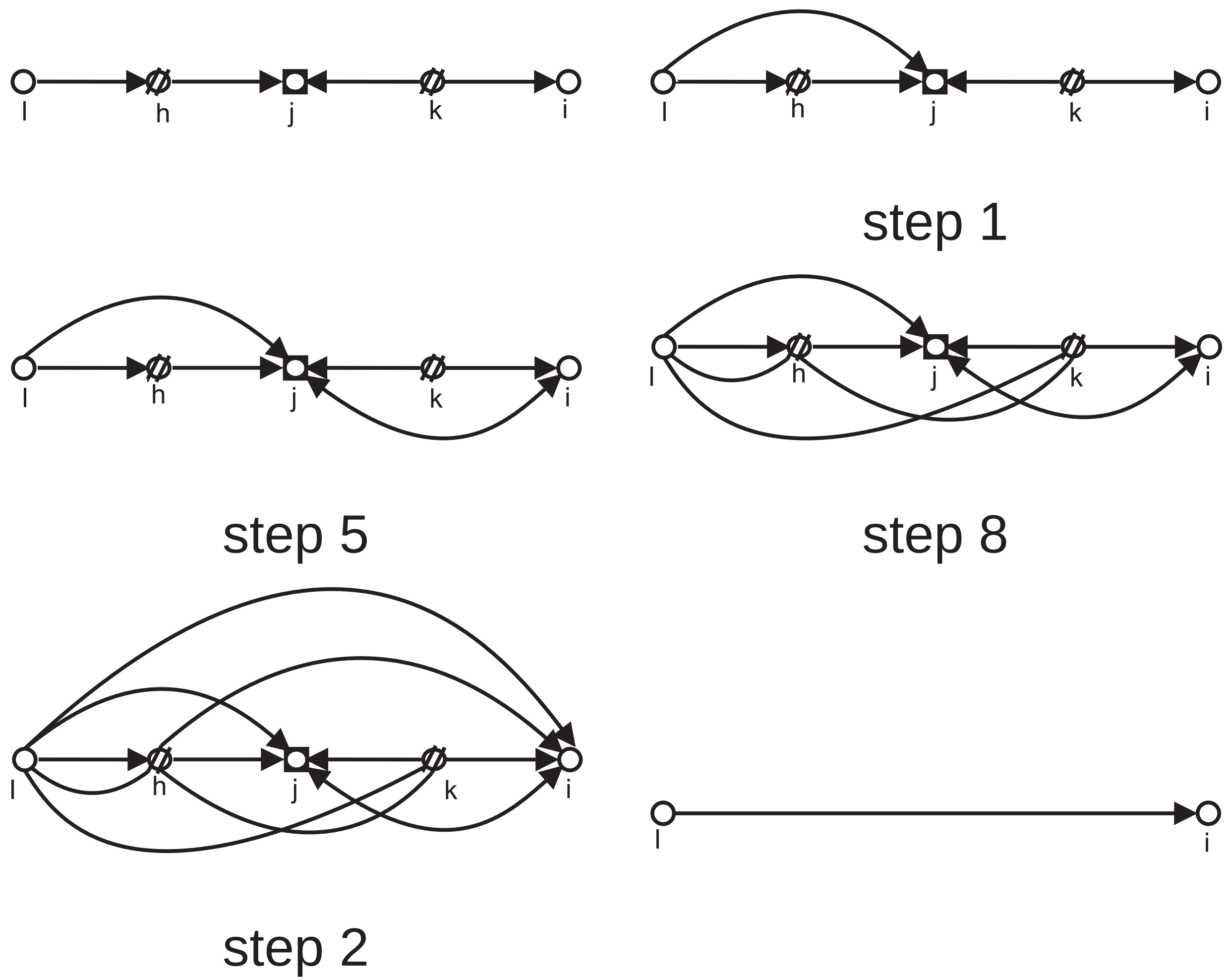
Two ribbons

Polynomial algorithm for generating RGs from DAGs or RGs

- $m \in M$: nodes to be marginalised over.
- C: nodes to be conditioned on.
- $s \in C \cup \mathrm{an}(C)$.
- We apply the following table to all V-configurations repeatedly until no other edge can be generated
- The generated graph is denoted by $\alpha_{RG}(H, M, C)$.

1	i ← m ← j	generates	i ← j
2	i ← m — j	generates	i ← j
3	i ↔ m — j	generates	i ← j
4	i ← m → j	generates	i ↔ j
5	i ← m ↔ j	generates	i ↔ j
6	i — m ← j	generates	i — j
7	i — m — j	generates	i — j
8	i ↔ s ← j	generates	i ← j
9	i ↔ s ↔ j	generates	i ↔ j
10	i → s ← j	generates	i — j

Example

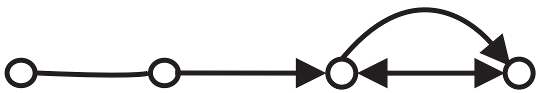


Important properties of α_{RG}

- The set of RGs is the exact image of α_{RG}.
  - RGs are probabilistic, i.e., there is a probability distribution faithful to them.
- $\alpha_{RG}(\alpha_{RG}(H, M, C), M_1, C_1) = \alpha_{RG}(H, M \cup M_1, C \cup C_1)$.
- $A \perp_m B \mid C_2$ in $\alpha_{RG}(H, M, C_1) \Leftrightarrow A \perp_m B \mid C_1 \cup C_2$ in H:
  - $\alpha(\mathcal{J}_m(H); M, C) = \mathcal{J}_m(\alpha_{RG}(H; M, C))$.
  - RGs are stable.
  - Undirected graphs and bidirected graphs are stable.

Summary graphs

A summary graph has no arrowheads pointing to lines and no direction-preserving cycles.



An SG

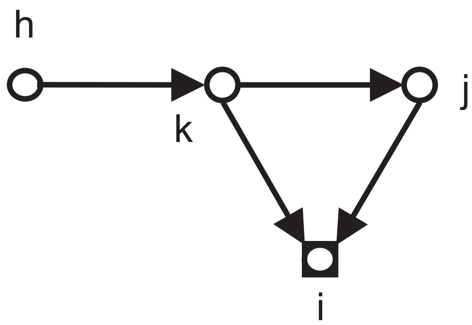
(b) An MG that is not an SG.

Polynomial algorithm for generating SGs from DAGs or SGs

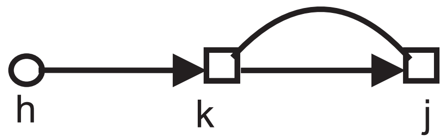
α_{SG}: Label the nodes in $\mathrm{an}(C)$.

1. Generate an RG.
2. Remove all edges (arrows or arcs) with an arrowhead pointing to a node in $\mathrm{an}(C)$, and replace these by the edge with the arrowhead removed (line or arrow).

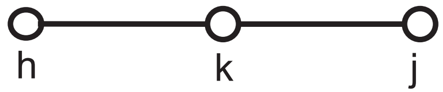
Example



$\boxed{\bigcirc} \in C$



generated RG, $\square \in \mathrm{an}(C)$



step 2

Important properties of α_{SG}

- The set of SGs is the exact image of α_{SG}.
- $\alpha_{SG}(\alpha_{SG}(H, M, C), M_1, C_1) = \alpha_{SG}(H, M \cup M_1, C \cup C_1)$.
- $A \perp_m B \mid C_2$ in $\alpha_{SG}(H, M, C_1) \Leftrightarrow A \perp_m B \mid C_1 \cup C_2$ in H:
  - $\alpha(\mathcal{J}_m(H); M, C) = \mathcal{J}_m(\alpha_{SG}(H; M, C))$.
  - SGs are stable.

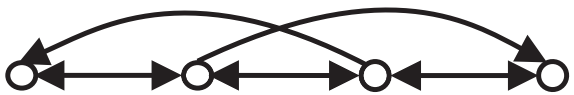
Ancestral graphs

An ancestral graph has

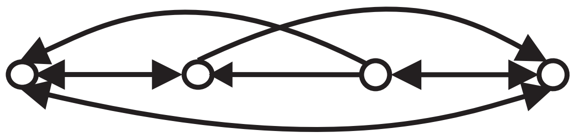
no arrowheads pointing to lines,

no direction-preserving cycles,

no bow: An arc with one endpoint that is an ancestor of the other endpoint.



An AG.



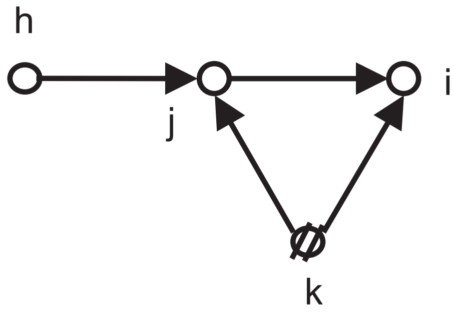
An SG that is not ancestral.

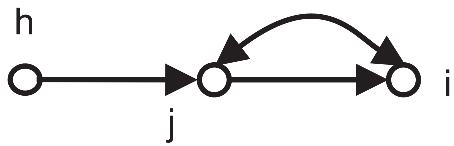
Polynomial algorithm for generating AGs from DAGs or AGs

α_{AG}:

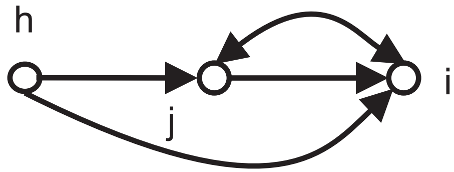
1. Generate an SG.
2. Generate $j \rightarrow i$ or $i \leftrightarrow j$ for $j \rightarrow k \leftrightarrow i$ or $j \leftrightarrow k \leftrightarrow i$ when $k \in \mathrm{an}(i)$.
3. Remove $j \leftrightarrow i$ in the case that $j \in \mathrm{an}(i)$, and replace it by $j \rightarrow i$.

Example

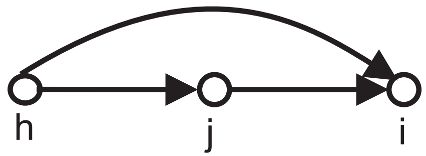




the generated SG



step 2



Important properties of α_{AG}

- The set of AGs is the exact image of α_{AG}.
- $\alpha_{AG}(\alpha_{AG}(H, M, C), M_1, C_1) = \alpha_{AG}(H, M \cup M_1, C \cup C_1)$.
- $A \perp_m B \mid C_2$ in $\alpha_{AG}(H, M, C_1) \Leftrightarrow A \perp_m B \mid C_1 \cup C_2$ in H:
  - $\alpha(\mathcal{J}_m(H); M, C) = \mathcal{J}_m(\alpha_{AG}(H; M, C))$.
  - AGs are stable.

Stable mixed graphs in R

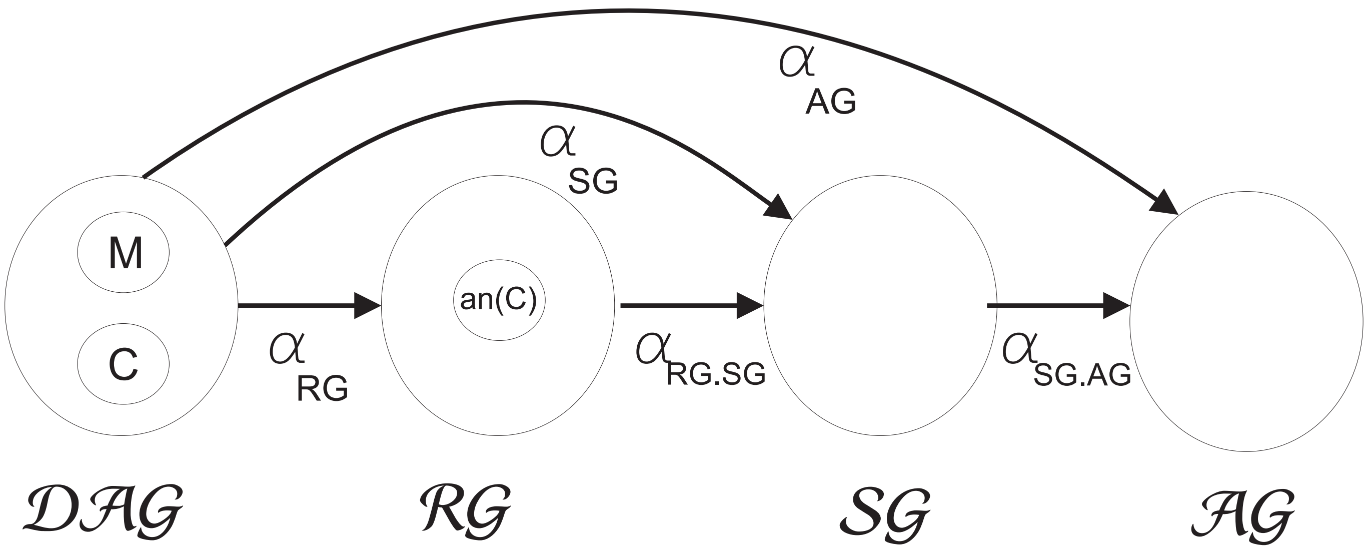
The algorithms have been implemented in R and available under the `ggm` package.

`RG` for α_{RG}

`SG` for α_{SG}

`AG` for α_{AG}

The relationship between stable mixed graphs

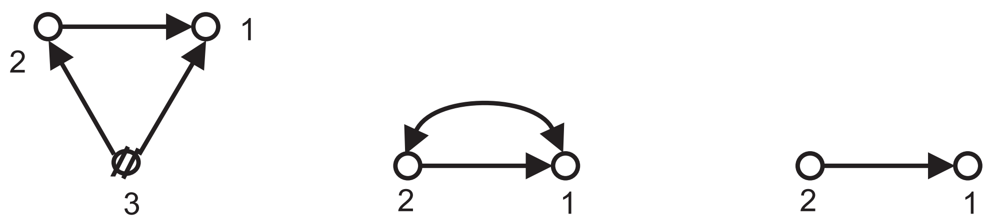


Ribbonless, summary, or ancestral graphs?

- Ancestral graphs have the simplest structure among these three types of graphs.
- Ribbonless graphs have the simplest generating algorithm.
- AGs are used when the generating DAG is not known but a set of conditional independencies is known.
- In the Gaussian case maximal AGs are identified, the models are curved exponential families, and conditional fitting algorithm for maximum likelihood estimation exists.
- By moving towards AGs we lose information:

Distortions

Summary graphs are more alerting to distortions than ancestral graphs when dealing with following the effects in multivariate regression systems after marginalisation and conditioning.



DAG   generated SG   generated AG

$$Y_1 = \beta Y_2 + \delta Y_3 + \epsilon_1, \quad Y_2 = \gamma Y_3 + \epsilon_2, \quad Y_3 = \epsilon_3,$$

$$E(Y_1 \mid Y_2) = (\beta + \delta\gamma) Y_2.$$