Inférence de réseaux

Graphes dirigés

Olivier Goudet

Université d'Angers

20 janvier 2026

Organisation générale du cours

4 séances de 2 heures de CM :

1. Cours 1 - S. Aubourg
   - Introduction aux réseaux de gènes.
2. Cours 2 - O. Goudet
   - Introduction à la causalité.
   - Notions d'indépendance entre différentes variables.
   - Graphes non dirigés.
3. Cours 3 - O. Goudet
   - Graphes dirigés.
   - Causalité paire à paire.
4. Cours 4 - O. Goudet
   - Méthodes d'inférence de réseaux utilisées en bioinformatique.

Plan du cours aujourd'hui

1. Introduction et notations
2. Hypothèses pour la causalité
3. Classes d'équivalence de Markov
4. Réseau bayésien et modèle fonctionnel causal
5. Méthodes d'inférence d'un graphe dirigé
   - Méthodes d'inférence locales à base de contraintes
   - Méthodes à base de scores de vraisemblance

Section 1

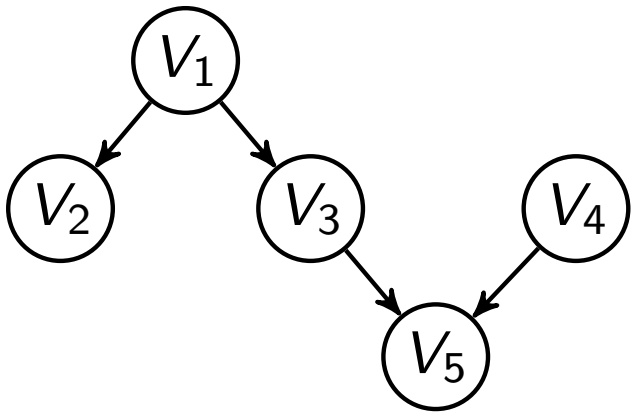
Introduction et notations

Modèles graphiques dirigés

- Dans cette partie, on considère uniquement des graphes dirigés sans cycles, appelés DAG pour *Directed Acyclic Graph*.
- Chaque modèle correspond à un graphe $\mathcal{G} = (V, E)$, avec:
  - V, l'ensemble des noeuds du graphe (chaque noeud $V_i \in V$ correspond à une variable X_i de X).
  - E, l'ensemble des liens dirigés du graphe. On note $(V_i, V_j) \in E$ une paire ordonnée qui correspond à un lien dirigé de V_i vers V_j.
  - On définit l'ensemble des parents de V_j dans le graphe $\mathcal{G}$ comme $pa_{\mathcal{G}}(V_j) = \{V_i \in V : (V_i, V_j) \in E\}$.
  - Ensemble des descendants de V_j dans $\mathcal{G}$: $de_{\mathcal{G}}(V_j) = \{V_i \in V : V_i = V_j \text{ ou } V_j \to \cdots \to V_i \in \mathcal{G}\}$.
  - Ensemble des non-descendants de V_j dans $\mathcal{G}$: $nd_{\mathcal{G}}(V_j) = V \backslash de_{\mathcal{G}}(V_j)$.
  - Ensemble des ancêtres de V_j dans $\mathcal{G}$: $an_{\mathcal{G}}(V_j) = \{V_i \in V : V_i = V_j \text{ ou } V_i \to \cdots \to V_j \in \mathcal{G}\}$.

Représentation du graphe dirigé par une matrice d'adjacence

- Un graphe dirigé avec d variables peut être représenté par une matrice binaire A représentant tous les liens dirigés entre les variables.
- $A_{ij} = 1$ si et seulement si il y a un lien dirigé du noeud V_i vers le noeud V_j dans le graphe $\mathcal{G}$, $A_{ij} = 0$ sinon.
- Remarque : pour un DAG, la matrice A peut être mise sous la forme d'une matrice triangulaire supérieure stricte (moyennant éventuellement un réindiçage des variables). Pour un DAG, on a donc toujours $A^d = 0$.



$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (1)$$

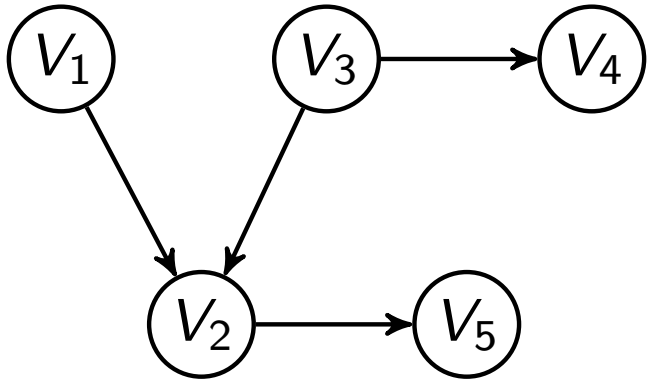
Chemins dans un graphe dirigé

- On dit que les sommets V_i et V_j sont adjacents dans le DAG $\mathcal{G} = (V, E)$ si $(V_i, V_j) \in E$ ou $(V_j, V_i) \in E$.
- Un chemin est une séquence de noeuds tels que les noeuds successifs sont adjacents.
- Si $\pi = (V_0, V_1, ..., V_k)$ est un chemin alors on dit que V_0 et V_k sont les points de terminaison du chemin π.
- Un sommet V_i qui n'est pas un point de terminaison est un *collider* de π si $V_{i-1} \rightarrow V_i \leftarrow V_{i+1}$ est un sous-chemin de π. Sinon V_i est un *non-collider* de π.

Notions de d-connexion et de d-séparation

- Deux sommets V_i et V_j dans un DAG $\mathcal{G} = (V, E)$ sont dits *d-connectés* étant donné un ensemble de variables $C \subseteq V \backslash \{V_i, V_j\}$ si $\mathcal{G}$ contient un chemin π avec les points de terminaison V_i et V_j tel que :
  - Tous les *colliders* de π sont dans $an_{\mathcal{G}}(C)$
  - Il n'y a pas de *non-collider* de π dans C.
- On dit que deux ensembles de variables disjoints $A, B \subset V$ sont *d-connectés* étant donné $C \subseteq V \backslash (A \cup B)$ s'il existe deux sommets $V_i \in A$ et $V_j \in B$ qui sont *d-connectés* étant donné C.
- Si ce n'est pas le cas, alors C *d-sépare* A et B dans le graphe $\mathcal{G}$, ce qui pourra se noter $A \perp\!\!\!\perp_{\mathcal{G}} B | C$.

Exemples de d-connexion et de d-séparation

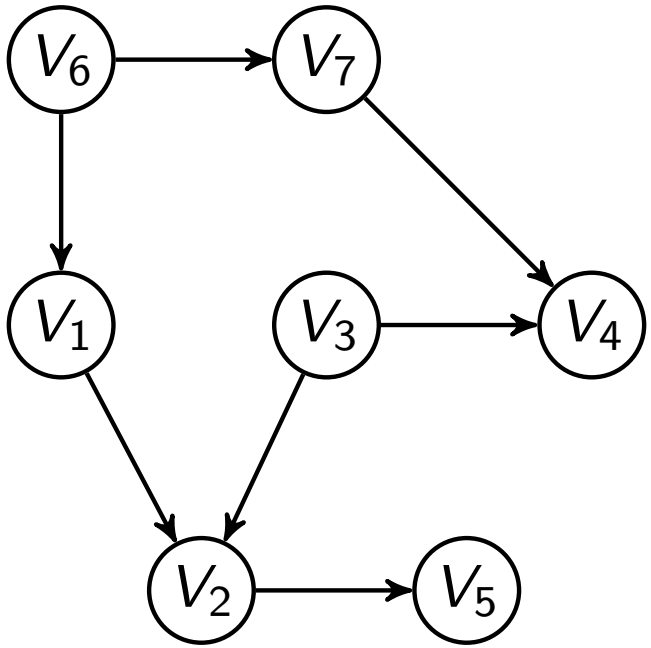


- Le sommet V_2 est un *collider* du chemin $V_1 \to V_2 \leftarrow V_3 \to V_4$, tandis que ce n'est pas un *collider* du chemin $V_1 \to V_2 \to V_5$.
- V_1 et V_4 sont d-connectés étant donné $C = \{V_2\}$, $C = \{V_5\}$ et $C = \{V_2, V_5\}$, mais d-séparés étant donnés tous les autres sous-ensembles C' de $C = \{V_2, V_3, V_5\}$.

Relation de d-separation non monotone

Attention, contrairement à la notion de séparation dans les graphes non dirigés, la d-separation dans les DAG n'est pas monotone, dans le sens où $A \perp\!\!\!\perp_{\mathcal{G}} B|C$ n'implique pas que $A \perp\!\!\!\perp_{\mathcal{G}} B|C'$ pour tout ensemble C' tel que $C \subsetneq C'$.

Exemple. Relation de d-separation non monotone.



- Ici, on a $V_6 \perp\!\!\!\perp_{\mathcal{G}} V_4 | \{V_7\}$.
- Mais $V_6 \not\perp\!\!\!\perp_{\mathcal{G}} V_4 | \{V_7, V_5\}$.
- En effet, le sommet V_5 a parmi ses ancêtres le sommet V_2 qui est un *collider* du chemin $V_6 - V_1 - V_2 - V_3 - V_4$, ce qui "connecte" maintenant les sommets V_6 et V_4 via ce chemin alternatif.

Section 2

Hypothèses pour la causalité

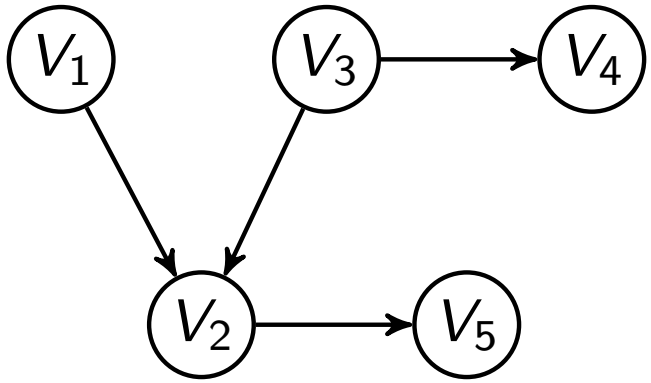
Hypothèses sur les données pour les graphes dirigés - Propriétés de Markov

- Un vecteur de variable aléatoire $X = (X_1, \ldots, X_d)$ associé aux sommets de V satisfait la propriété de Markov locale (Ld) par rapport au DAG $\mathcal{G}$ si pour tout $V_i \in V$:

$$X_i \perp\!\!\!\perp X_{nd_{\mathcal{G}}(V_i) \setminus pa_{\mathcal{G}}(V_i)} | X_{pa_{\mathcal{G}}(V_i)} \tag{2}$$

- X satisfait la propriété de Markov globale (Gd) par rapport au graphe $\mathcal{G}$ si $X_A \perp\!\!\!\perp X_B | X_C$ pour tous les triplets d'ensembles de variables disjoints $A, B, C \subset V$ tels que C *d-separe* A et B dans $\mathcal{G}$, ce qui peut se noter $A \perp\!\!\!\perp_{\mathcal{G}} B | C$.
- Si $\mathcal{G}$ est un DAG, les propriétés de Markov locale (Ld) et globale (Gd) sont équivalentes (Lauritzen et al., 1990). Voir preuve pdf annexe du cours.
- Si X satisfait la propriété de Markov globale par rapport au graphe $\mathcal{G}$ alors $\mathcal{G}$ est appelé une *independence map* de X.

Exemple de DAG et propriétés de Markov

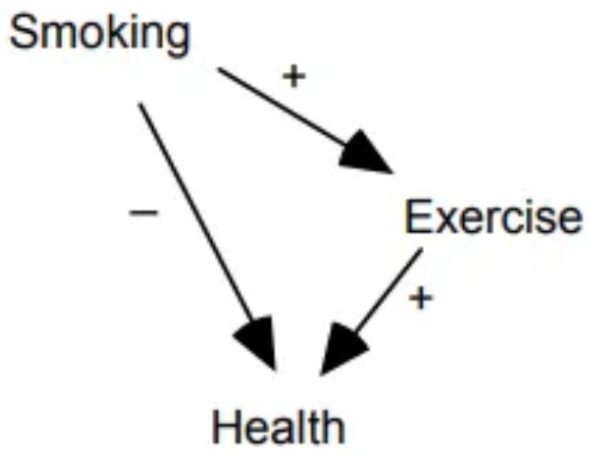


- Le noeud V_2 est un *collider* du chemin $V_1 \to V_2 \leftarrow V_3 \to V_4$, tandis que ce n'est pas un *collider* du chemin $V_1 \to V_2 \to V_5$.
- Pour le noeud V_4, la propriété de Markov locale requiert $X_4 \perp\!\!\!\perp (X_1, X_2, X_5) | X_3$.
- V_1 et V_4 sont d-connectés étant donné $C = \{V_2\}$, $C = \{V_5\}$ et $C = \{V_2, V_5\}$, mais d-séparés étant donnés tous les autres sous-ensembles C' de $C = \{V_2, V_3, V_5\}$, la propriété de Markov globale impose que $X_1 \perp\!\!\!\perp X_4 | X_{C'}$ pour tous ces sous-ensembles C' de sommets.

Hypothèses sur les données pour les graphes dirigés - *faithfullness*

- Un DAG est une *perfect map* de X si pour tout les ensembles de variables disjoints deux à deux $A, B, C \subseteq V$: $A \perp\!\!\!\perp_{\mathcal{G}} B|C$ si et seulement si $X_A \perp\!\!\!\perp X_B|X_C$.
- Une *perfect map* nécessite donc la propriété de Markov globale mais aussi son implication inverse appelé *faithfulness* : $X_A \perp\!\!\!\perp X_B|X_C \Rightarrow A \perp\!\!\!\perp_{\mathcal{G}} B|C$.
- En particulier, si C est l'ensemble vide, et les ensembles A et B sont des singletons $A = \{V_i\}$ et $A = \{V_j\}$, alors cette hypothèse donne $X_i \perp\!\!\!\perp X_j \Rightarrow V_i \perp\!\!\!\perp_{\mathcal{G}} V_j|\{\}$. Ou encore, si $X_i \perp\!\!\!\perp X_j$ alors les sommets V_i et V_j ne sont pas adjacents dans le graphe.

Exemple de cas *unfaithful*



(Scheines, R. (1997)

- On suppose que ce graphe imaginaire est le vrai graphe causal sous-jacent. Il y a deux chemins dans ce graphe qui ont un effet sur la santé à partir de la variable "Fumer" (l'un a un impact positif, l'autre négatif).
- Il peut arriver que ces deux impacts se compensent et qu'on mesure en fait empiriquement $Smoking \perp\!\!\!\perp Health$.
- Ce cas est exclu par l'hypothèse de *faithfullness*, sinon on ne peut jamais retrouver le vrai graphe causal.

Hypothèses sur les données pour les graphes dirigés - *causal sufficiency*

- Hypothèse de *causal sufficiency* : il n'existe pas de paires de variables $\{X_i, X_j\}$ de X ayant une cause commune qui n'est pas dans $X_{\setminus i,j}$.
- Cette hypothèse est souvent faite à cause d'effets confondants cachés.
- Remarque : si on fait cette hypothèse et qu'on observe que V_i et V_j sont adjacents dans le graphe, alors ou a forcément $V_i \to V_j$ ou bien $V_j \to V_i$.
- Vu différemment, si une variable "cachée" causait à la fois X_i et X_j, ces deux variables pourraient devenir dépendantes, alors qu'il n'y pas d'arc dirigé entre les deux (cf. exemple chocolat et prix Nobel du cours précédent).

Section 3

Classes d'équivalence de Markov

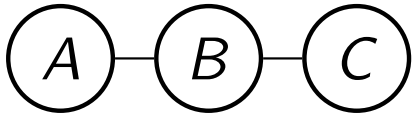
Graphes Markov équivalents

- Deux graphes $\mathcal{G}$ et $\mathcal{G}'$ sont Markov équivalents si $A \perp\!\!\!\perp_{\mathcal{G}} B | C$ est équivalent à $A \perp\!\!\!\perp_{\mathcal{G}'} B | C$.
- Deux graphes sont Markov équivalents s'ils ont le même squelette et les mêmes *v-structures* (Verma and Pearl, 1991).
- Une *v-structure* est un triplet de sommets tels que $V_i \rightarrow V_k \leftarrow V_j$ avec V_i et V_j non adjacents.

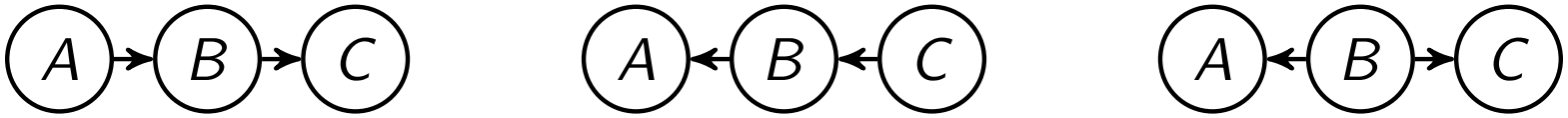
CPDAG : Completed Partially Directed Acyclic Graph

- Chaque classe d'équivalence de Markov peut être représentée par un CPDAG (*Completed Partially Directed Acyclic Graph*) qui a des liens dirigés et des liens non dirigés.
- Un CPDAG a le lien dirigé $V_i \rightarrow V_j$ si et seulement si cette arc $V_i \rightarrow V_j$ est commun à tous les DAGs de la classe d'équivalence.
- Si la classe contient un DAG avec $V_i \rightarrow V_j$ et un autre DAG avec $V_j \rightarrow V_i$ alors le CPDAG a le lien non dirigé $V_i - V_j$

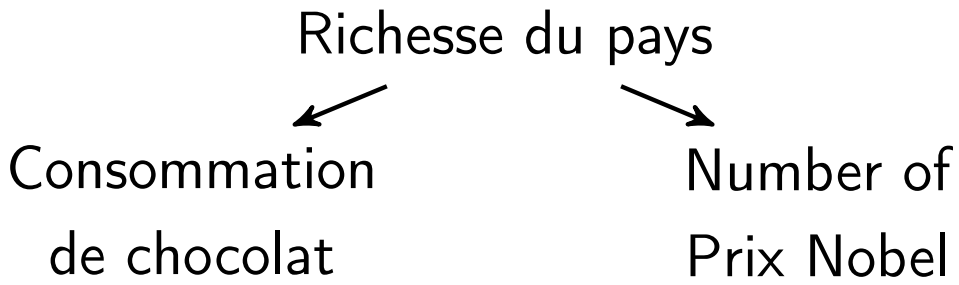
Exemple : classes d'équivalence de Markov

Squelette: 

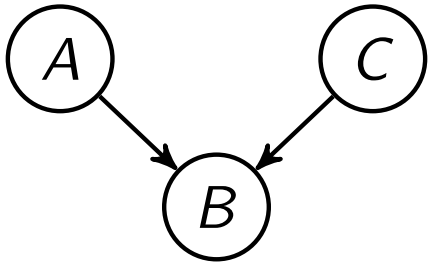
3 classes d'équivalence de Markov: $A \perp\!\!\!\perp C | B$



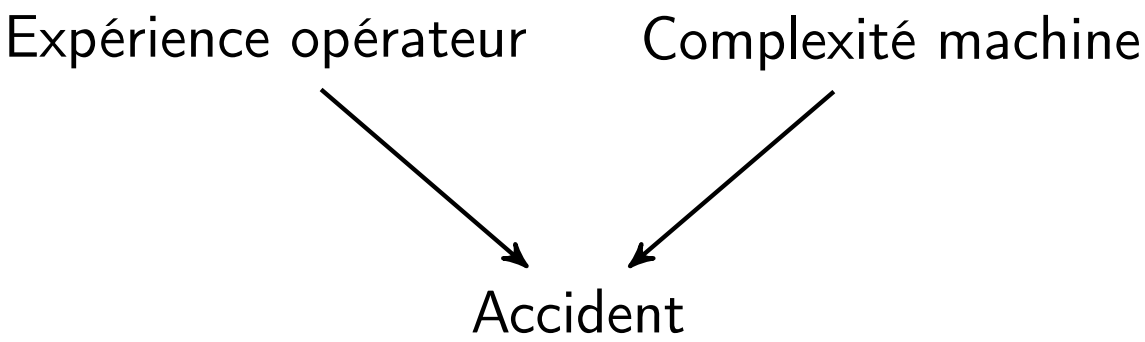
Exemple



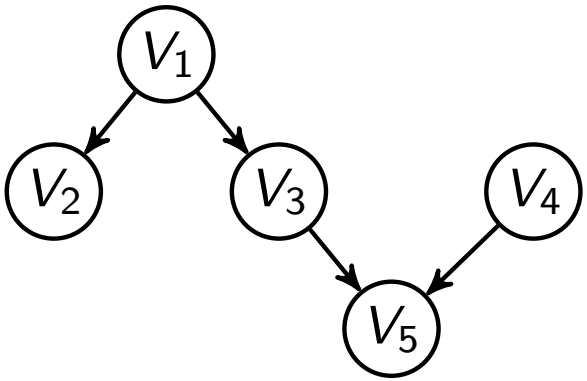
V-Structure: $A \not\perp\!\!\!\perp C | B$



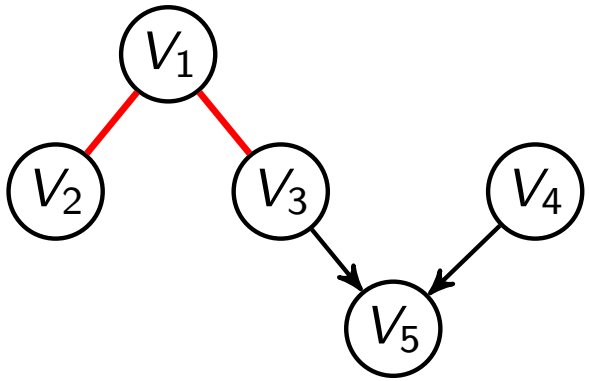
Exemple



Exemple CPDAG



(a) Le vrai DAG $\mathcal{G}$.

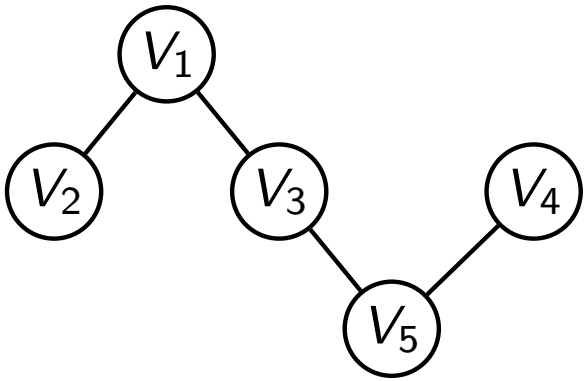


(b) Le CPDAG de $\mathcal{G}$.

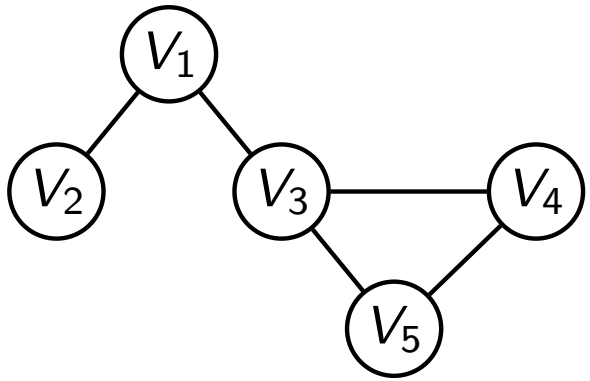
Lien entre squelette et graphe moral

- Le squelette d'un graphe dirigé est le graphe non dirigé obtenu en remplaçant tous les arcs dirigés par des arcs non dirigés.
- Le graphe moral $\mathcal{G}^m$ de $\mathcal{G}$ est construit en ajoutant un arc entre V_i et V_j pour chaque *v-structure* $V_i \rightarrow V_k \leftarrow V_j$ et en prenant le squelette du graphe résultant.
- Si G est une *perfect map* de X alors $\mathcal{G}^m$ est le **graphe des indépendances conditionnelles** de X (cf. cours 2).
- Le squelette d'un DAG est un sous-graphe de son graphe des indépendances conditionnelles.

Squelette et graphe moral



(a) Le squelette du DAG $\mathcal{G}$.



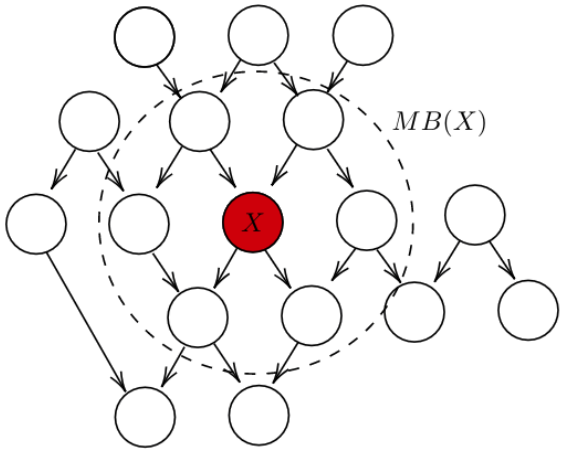
(b) Le graphe moral $\mathcal{G}^m$ de $\mathcal{G}$.

Couverture de Markov

- La couverture de Markov (*Markov Blanket*) du sommet V_i associé à la variable X_i est le sous ensemble minimal de variables $MB(V_i)$ de $V \setminus V_i$ tel que :

$$X_i \perp\!\!\!\perp X_{V \setminus (MB(X_i) \cup V_i)} | X_{MB(V_i)}$$

- Si $\mathcal{G}$ est une *perfect map* de X alors $MB(V_i)$ correspond à l'ensemble des voisins de V_i dans le graphe moral de $\mathcal{G}$: $MB(V_i) = nb_{\mathcal{G}^m}(V_i)$.
- L'identification de cette couverture de Markov est un problème général de sélection de variables (cf. fin du cours 2).



Relations d'adjacence dans un graphe non-dirigé $\mathcal{G}^m$ et dans un graphe dirigé $\mathcal{G}$

- Attention, la notation d'adjacence dans un graphe non-dirigé $\mathcal{G}^m$ n'est pas la même que dans un graphe dirigé $\mathcal{G}$.
- Si on fait les hypothèses de Markov et *faithfullness* :
  - Pour un graphe non dirigé $\mathcal{G}^m$ on a (cf. cours précédent) :

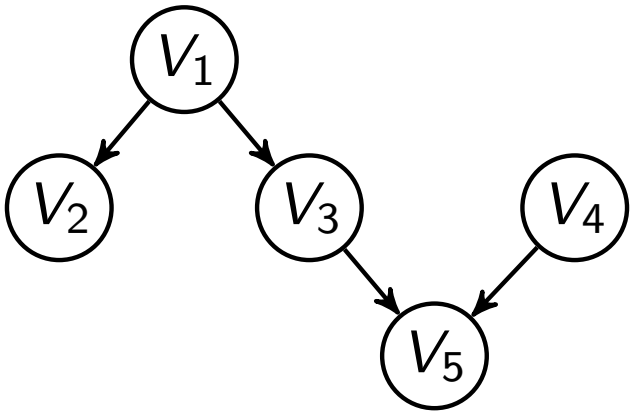
$$\{V_i, V_j\} \in E \iff X_i \not\perp\!\!\!\perp X_j | X_{\setminus i,j}$$

  - Pour un graphe dirigé $\mathcal{G}$:

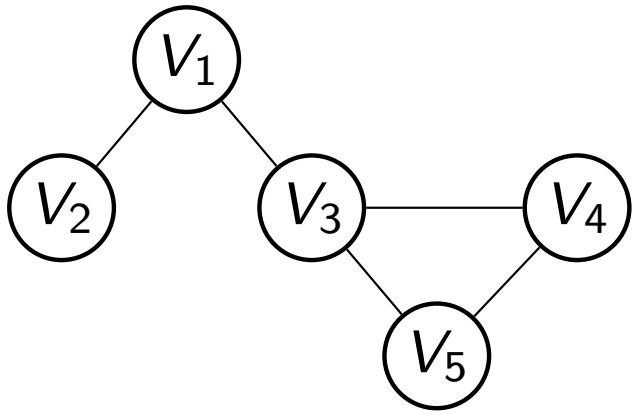
$$(V_i, V_j) \in E \text{ ou } (V_j, V_i) \in E \iff \nexists C, C \subseteq V \setminus \{V_i, V_j\}, X_i \perp\!\!\!\perp X_j | X_C$$

  - Pour un graphe dirigé $\mathcal{G}$, en particulier avec $C = \{\}$, si $X_i \perp\!\!\!\perp X_j$, alors $X_i \perp\!\!\!\perp X_j | C$, donc V_i et V_j ne sont pas adjacents dans le graphe.
  - Ou encore, si V_i et V_j sont adjacents dans le graphe alors on doit avoir $X_i \not\perp\!\!\!\perp X_j$.

Exemples : relations d'adjacence dans un graphe non-dirigé $\mathcal{G}^m$ et dans un graphe dirigé $\mathcal{G}$



(a) Le vrai graphe causal $\mathcal{G}$.



(b) Le graphe moral $\mathcal{G}^m$ de $\mathcal{G}$.

- Il y a une v-structure $V_3 \to V_5 \leftarrow V_4$. On observe $X_3 \perp\!\!\!\perp X_4$ et $X_3 \not\perp\!\!\!\perp X_4 | X_5$ (et aussi $X_3 \not\perp\!\!\!\perp X_4 | X_{\setminus 3,4}$).
- Comme $X_3 \perp\!\!\!\perp X_4$, les sommets V_3 et V_4 ne sont pas adjacents dans le graphe dirigé $\mathcal{G}$.
- Mais comme $X_3 \not\perp\!\!\!\perp X_4 | X_{\setminus 3,4}$, le lien $V_3 - V_4$ existe dans le graphe non dirigé $\mathcal{G}^m$.

Section 4

Réseau bayésien et modèle fonctionnel causal

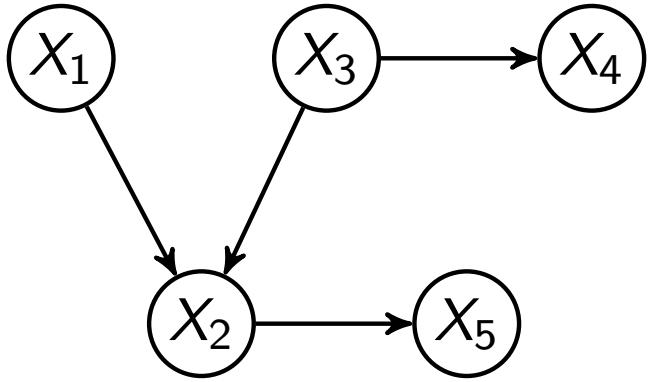
Factorisation de la distribution jointe suivant le graphe

- On dit que la distribution jointe $p(\mathrm{x})$ de X se factorise suivant le DAG $\mathcal{G} = (V, E)$ si on peut écrire :

$$p(\mathrm{x}) = \prod_{j=1}^{d} p(x_j | x_{pa_{\mathcal{G}}(v_j)}) \tag{3}$$

- On parle dans ce cas de **réseau bayésien**.
- Cette propriété de factorisation est équivalente à la propriété de Markov locale et globale dans le cas d'un DAG (Verma and Pearl, 1990).
- Intérêt de cette factorisation : la fonction de vraisemblance du modèle se factorise en d fonctions de vraisemblance locales qui peuvent se calculer séparément.

Exemple de factorisation



Sur le graphe ci-dessus, les V_i sont remplacés directement par les X_i pour plus de clarté.
La factorisation de $p(\mathrm{x})$ suivant $\mathcal{G}$ prend la forme :

$$p(\mathrm{x}) = p(x_1)p(x_2|x_1, x_3)p(x_3)p(x_4|x_3)p(x_5|x_2)$$

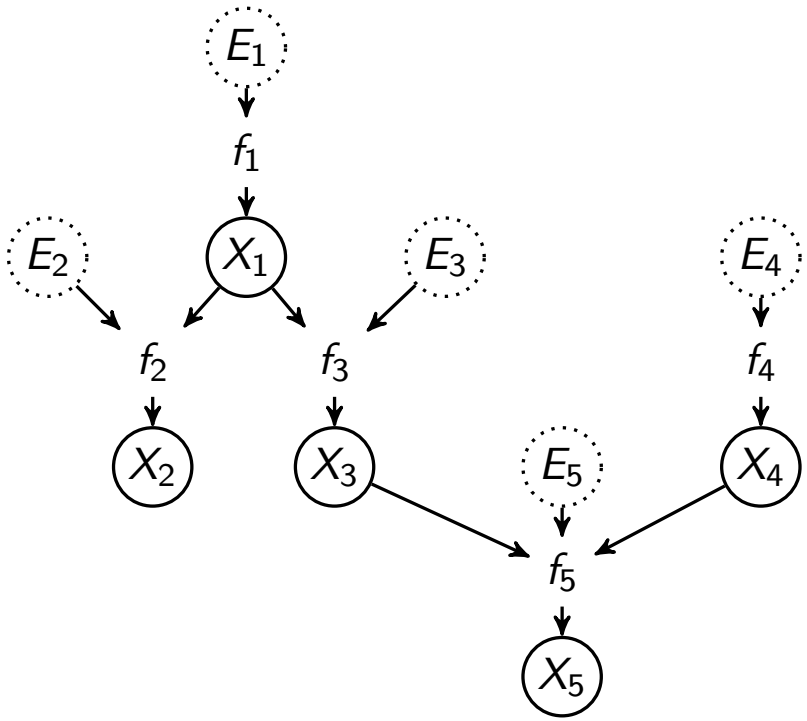
Modèle fonctionnel causal ou SEM (*Structural Equation Model*)

- Un DAG $\mathcal{G} = (V, E)$ peut aussi être vu comme un modèle fonctionnel causal (FCM : *Fonctional Causal Model*) ou modèle d'équations structurelles (SEM).
- Ici par simplicité, le sommet V_i est remplacé par la variable X_i.
- Si X satisfait la propriété de Markov par rapport à $\mathcal{G}$ alors il existe des variables aléatoires indépendantes E_i et des fonctions f_i telles que :

$$X_i \leftarrow f_i(X_{pa_{\mathcal{G}}(X_i)}, E_i), \text{ for } i = 1, \ldots, d \tag{4}$$

- De façon réciproque s'il existe un FCM satisfait par X alors X satisfait la propriété de Markov par rapport à $\mathcal{G}$.

Exemple FCM



$$\begin{cases} X_1 = f_1(E_1) \\ X_2 = f_2(X_1, E_2) \\ X_3 = f_3(X_1, E_3) \\ X_4 = f_4(E_4) \\ X_5 = f_5(X_3, X_4, E_5) \end{cases}$$

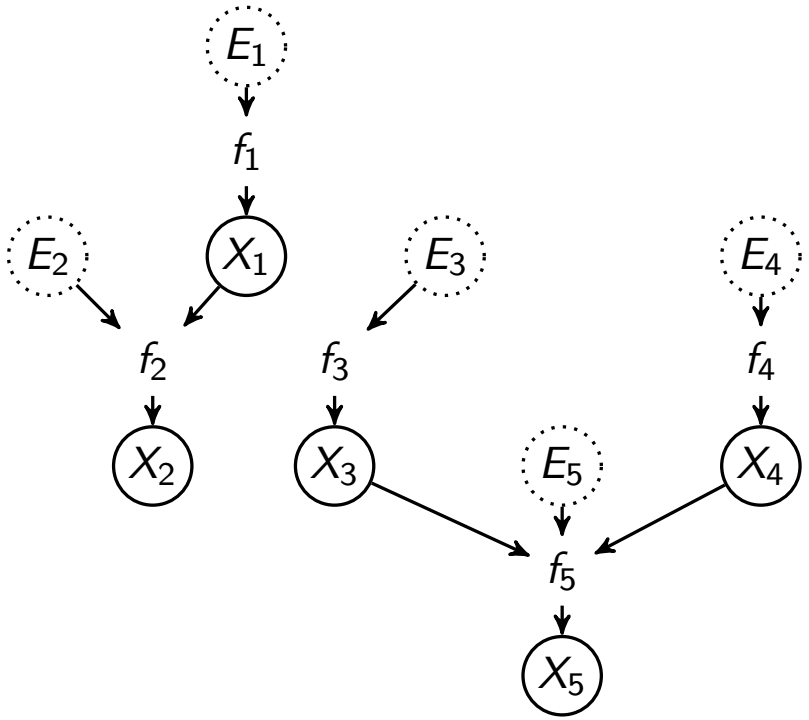
Figure 1: Exemple de modèle fonctionnel causal (FCM) avec $\mathrm{X} = (X_1, \ldots, X_5)$: A gauche : graphe causal $\mathcal{G}$; A droite : mécanismes causaux.

FCM et réseaux bayésiens

- Connaître le FCM, donne une factorisation de p suivant le graphe (réseau bayésien).
- Par contre connaître le réseau baysien et la factorisation de p ne donne pas l'expression du FCM.
- La représentation avec un FCM donne plus d'information que la représentation avec un réseau bayésien.
- La représentation avec un FCM permet de raisonner sur des interventions qui pourraient être effectuées sur le graphe causal.

Exemple d'intervention expérimentale

- A partir du graphe précédent, on peut choisir d'effectuer une intervention expérimentale sur X_3, de telle manière que X_3 soit générée à partir de E_3 uniquement et non plus à partir de X_1 et E_3 (on coupe le lien $X_1 \rightarrow X_3$).
- On peut en déduire la nouvelle distribution d'intervention résultante à partir du nouveau modèle causal :



$$\begin{cases} X_1 = f_1(E_1) \\ X_2 = f_2(X_1, E_2) \\ X_3 = f_3(E_3) \\ X_4 = f_4(E_4) \\ X_5 = f_5(X_3, X_4, E_5) \end{cases}$$

Lien entre FCM et modèle gaussien

- Si toutes les fonctions f_i sont linéaires et que les variables aléatoires E_i sont indépendantes entre elles et suivent chacune une distribution normale centrée réduite, alors on peut écrire :

$$\mathrm{X} = \mathrm{X}B + \mathrm{E},$$

avec la matrice $B \in \mathbb{R}^{d \times d}$ qui a des coefficient nuls $b_{i,j}$ si $X_i \notin pa_{\mathcal{G}}(X_j)$ et $\mathrm{E} = (E_1, ..., E_d)$, avec B triangulaire supérieure stricte (moyennant éventuellement un réindiçage des indices des variables).

- Ici $(I - B)$ est inversible car $|I - B| = 1$ ($I - B$ est triangulaire avec des 1 sur la diagonale).
- La solution du FCM est $\mathrm{X} = \mathrm{E}(I - B)^{-1}$.
- X suit donc une distribution gaussienne multivariée de matrice de covariance $\mathrm{Cov}(\mathrm{X}) = (I - B)^{-1^T} \mathrm{Cov}(\mathrm{E})(I - B)^{-1}$.

Section 5

Méthodes d'inférence d'un graphe dirigé

Subsection 1

Méthodes d'inférence locales à base de contraintes

Méthodes d'apprentissage locales à base de contraintes

- **Avantage :** méthode générique. Peut fonctionner quel que soit les distributions sous-jacentes des données (donc pas uniquement dans le cas gaussien multivarié).
- D'après les hypothèses de Markov, de faithfullness et de *causal sufficiency*, moyennant le fait d'établir toutes les relations d'indépendance conditionnelle entre les variables, on retrouve le CPDAG de $\mathcal{G}$.
- Pour qu'une garantie théorique d'identifiabilité soit apportée, il est nécessaire de supposer que les tests d'indépendance conditionnelle sont "parfaits" (oracle).
- En pratique on choisira un test indépendance conditionnel adapté aux données à traiter. Par exemple, si on sait que les données sont gaussiennes, on effectuera des tests de corrélation partielle (cf. cours 2).

Méthodes d'apprentissage locales à base de contraintes

- Ces algorithmes d'inférence se déroulent classiquement en trois étapes :
  1. retrouver le squelette du graphe,
  2. Identifier les v-structures,
  3. et appliquer des règles de propagation.
- **Avantage de ces méthodes :** tout type de test d'indépendance conditionnelle peut être utilisé (e.g. corrélation partielle ou tests non paramétriques comme KCI (Zhang et al., 2012)).
- **Désavantages :** en pratique difficile de faire ces tests en haute dimension + propagations d'erreurs possibles.

Retrouver le squelette

- Deux sommets V_i et V_j sont adjacents dans le graphe dirigé $\mathcal{G}$ si et seulement si il n'existe pas d'ensemble de sommets $C \subseteq V \backslash (V_i, V_j)$ (même vide) qui les d-sépare.
- Donc si $\mathcal{G}$ est une *perfect map* de X, on sait V_i et V_j sont adjacents dans le graphe dirigé $\mathcal{G}$ si et seulement si il n'existe pas de sous ensemble X_C de variables (même $X_C = \{\}$) de $X_{\backslash ij}$ tel que $X_i \perp\!\!\!\perp X_j | X_C$.
- Une approche naive pour savoir si X_i et X_j sont adjacents est de faire touts les tests d'indépendance conditionnelle par rapport à tous les sous-ensembles de variables $X_C \subseteq X_{\backslash ij}$ (Algorithmes SGS (Verma and Pearl, 1991)).
- Le nombre de tests à effecter augmente exponentiellement avec le nombre de variables.
- Une amélioration a été proposée par Spirtes et al. (2000) pour réduire le nombre de tests à effectuer (Algorithme PC).

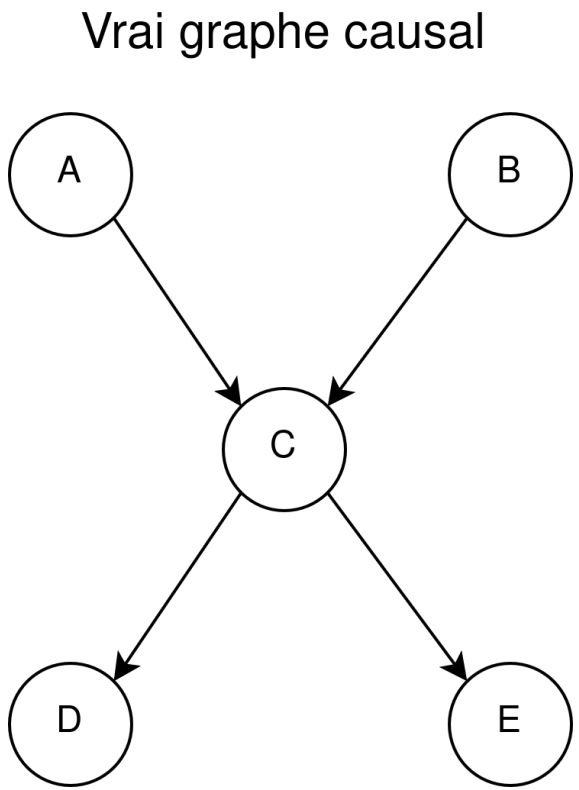
Algorithme PC Spirtes et al. (2000)

- L'idée est de limiter le nombre de tests d'indépendance en remarquant que si le graphe $\mathcal{G}$ est une *perfect map* alors on a l'équivalence suivante : V_i et V_j ne sont pas adjacents dans $\mathcal{G}$ si et seulement si $X_i \perp\!\!\!\perp X_j | X_{pa_{\mathcal{G}}(V_i)}$ ou $X_i \perp\!\!\!\perp X_j | X_{pa_{\mathcal{G}}(V_j)}$.
  Voir preuve pdf annexe du cours.
- Comme les parents ne sont pas connus, il faut s'assurer que tous les tests suivants sont faits : $X_i \perp\!\!\!\perp X_j | X_C$ pour tout C tel que $C \subseteq adj_{\mathcal{G}}(V_i)$ et $C \subseteq adj_{\mathcal{G}}(V_j)$.

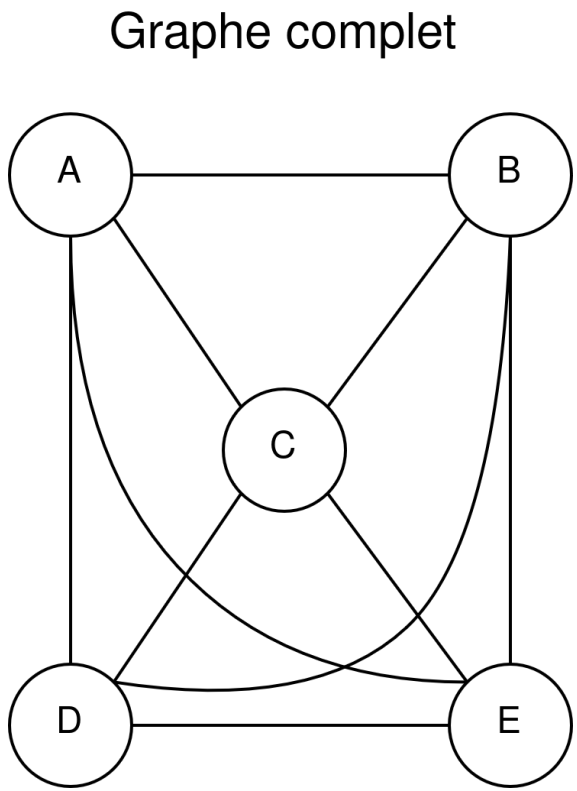
Algorithme PC (Spirtes et al., 2000)

1. Retrouver le squelette :
   - On part du graphe complet sur V.
   - On effectue les tests d'indépendance paire à paire et on retire l'arc correspondant si les variables ne sont pas dépendantes.
   - Pour chaque paire de variables encore adjacentes, on effectue des tests d'indépendance conditionnelle étant donné tous les autres sous-ensembles de variables de cardinalité $c = 1$. On retire un arc si une indépendance conditionnelle est trouvée.
   - On continue de cette manière en augmentant de 1 le paramètre c tant que $c < max_{V_i \in V} |adj_{\mathcal{G}'}(V_i)|$ où $\mathcal{G}'$ est l'état actuel du squelette.
2. Orienter les arcs :
   - Identifier toutes les v-structures $V_i \rightarrow V_k \leftarrow V_j$.
   - Appliquer des règles de propagation.

Exemple d'application de l'algorithme PC (Spirtes et al., 2000)

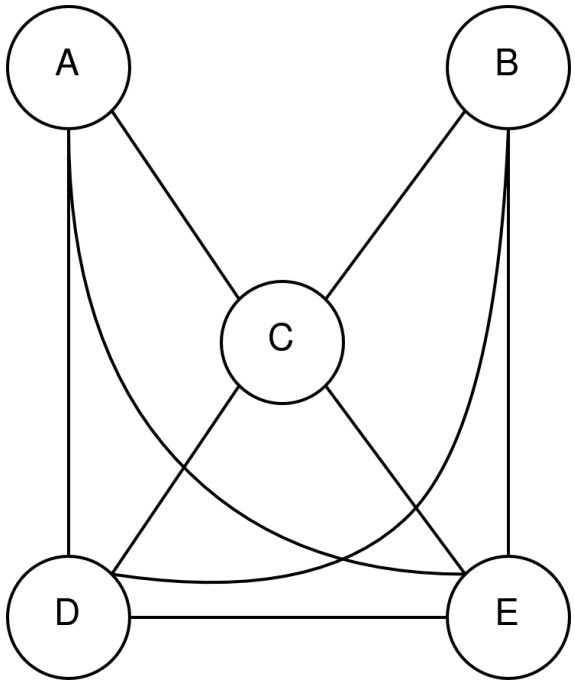


Etape 0 : le graphe complet



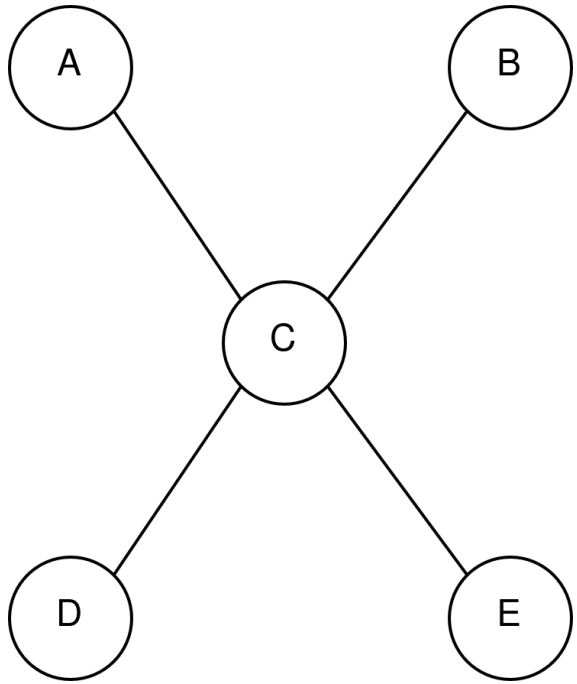
Etape 1 : tests d'indépendance paire à paire

- On commence avec $c = 0$.
- On retire un arc, si on observe $X \perp\!\!\!\perp Y | Z$, avec $Z = \{\}$.
- Dans ce cas, on observe que $A \perp\!\!\!\perp B | \{\}$, donc l'arc $A - B$ est retiré.
- On ne peut pas retirer d'autres arcs à ce stade.



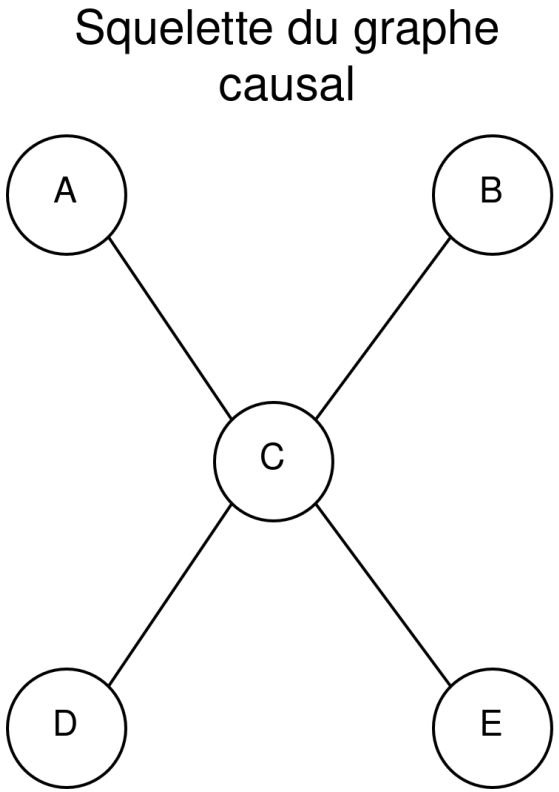
Etape 2 : tests d'indépendance conditionnelle pour $c = 1$

- Maintenant $c = 1$
- On retire un arc, si on observe $X \perp\!\!\!\perp Y|Z$, avec $|Z| = 1$.
- Dans ce cas, on observe que $A \perp\!\!\!\perp D|C$, $A \perp\!\!\!\perp E|C$, $B \perp\!\!\!\perp D|C$, $B \perp\!\!\!\perp E|C$ et $D \perp\!\!\!\perp E|C$.



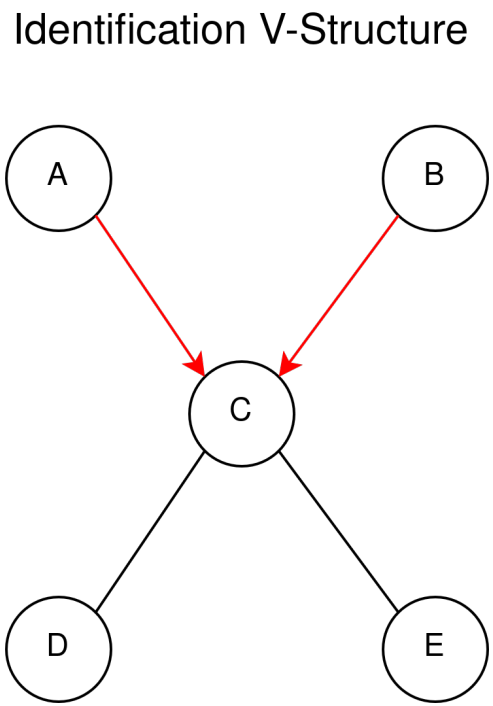
Etape 3 : autres tests d'indépendance conditionnelle

- Le nombre de voisins de C est de 4, donc on n'a pas encore atteint la condition d'arrêt.
- Mais pour $c = 2$, $c = 3$ et $c = 4$, on n'observe pas de nouvelles relations d'indépendance conditionnelle.
- Le squelette est identifié.



Etape 4 : identification des v-structures.

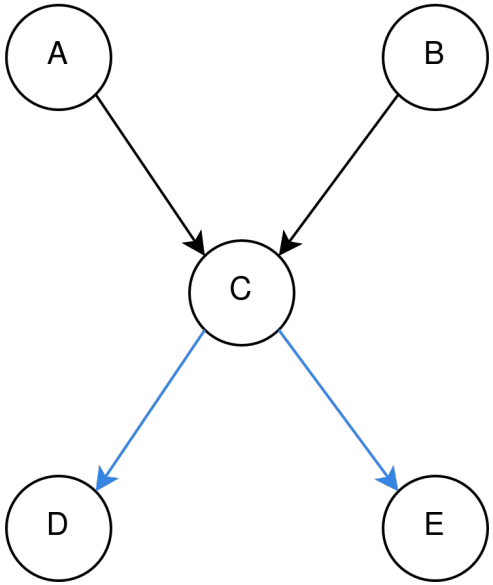
- On observe $A \not\perp\!\!\!\perp B|C$. Comme de plus, $A \perp\!\!\!\perp B$, alors la v-structure $A - C - B$ est identifiée.



Etape 5 : application des règles de propagation.

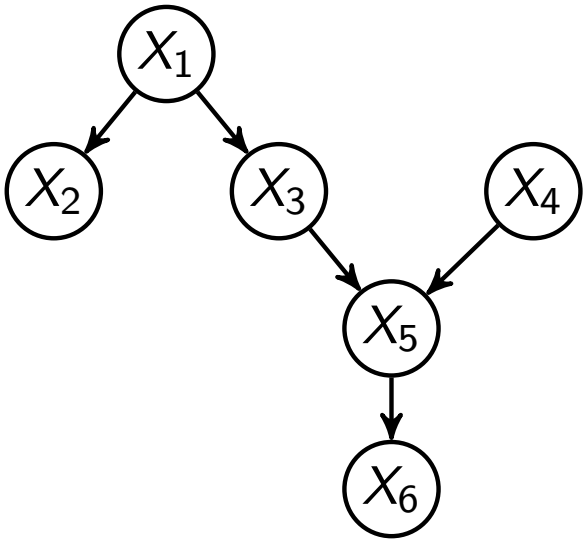
- On observe $A \perp\!\!\!\perp D | C$. Donc, il n'y a pas de v-structure $A - C - D$.
- De plus, forcément $C \rightarrow D$ ou bien $D \rightarrow C$, (car C et D sont adjacents dans le squelette), et donc comme $A \rightarrow C$, on en déduit que $C \rightarrow D$.
- De même on en déduit $C \rightarrow E$.

Règle de propagation

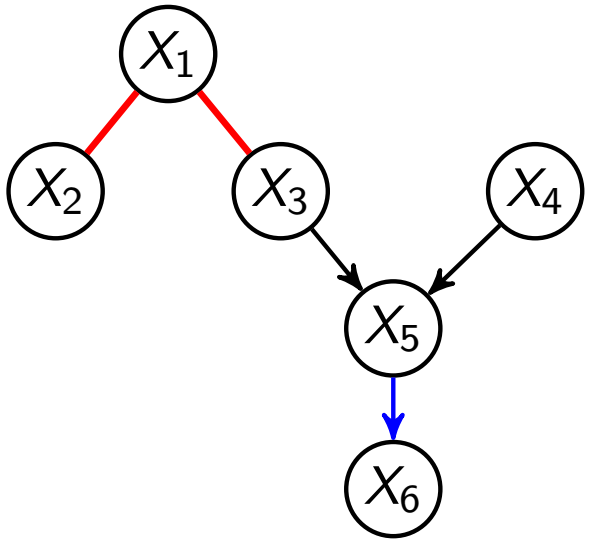


Identification du CPDAG de $\mathcal{G}$

- Dans certains cas, certains arcs ne peuvent pas être orientés.
- On ne retrouve donc pas complètement le vrai graphe causal.
- On obtient la classe d'équivalence de Markov du graphe, représentée par le CPDAG.



(a) Le vrai DAG $\mathcal{G}$.



(b) Le CPDAG de $\mathcal{G}$.

Subsection 2

Méthodes à base de scores de vraisemblance

Autre approche pour inférer un graphe dirigé : méthodes à base de calculs de vraisemblance

- Pour chaque variable observée X_j, on construit un modèle de régression de X_j conditionnellement à l'ensemble des autres variables $X_{\setminus j}$.
- On note $q_j(x_j|x_{\mathrm{Pa}(j;\hat{\mathcal{G}})}, \tau_j)$ la densité de ce modèle probabiliste. Pour simplifier, on note ici $X_{\mathrm{Pa}}(j;\hat{\mathcal{G}})$ à la place de $X_{pa_{\widehat{\mathcal{G}}}(V_j)}$.
- $X_{\mathrm{Pa}}(j;\hat{\mathcal{G}})$ correspond au sous ensemble des variables selectionnées dans le modèle, interprété dans ce cas comme les parents de V_j qui représente la variable X_j dans le graphe candidat $\widehat{\mathcal{G}}$.
- τ_i sont les paramètres utilisés pour prédire X_i à partir de $X_{\mathrm{Pa}}(j;\hat{\mathcal{G}})$.
- Il y a d modèles $q_j(x_j|x_{\mathrm{Pa}(j;\hat{\mathcal{G}})}, \tau_j)$ (Markov Kernels) à apprendre conjointement, en respectant la contrainte d'acyclicité globale sur le graphe $\widehat{\mathcal{G}}$.

Score de log-vraisemblance global

- On note $\tau = (\tau_1, \ldots, \tau_d)$ l'ensemble des paramètres des d Markov kernels à apprendre conjointement.
- Pour notre échantillon de données $\mathcal{D}$ supposées iid, pour chaque DAG candidat $\hat{\mathcal{G}}$, en utilisant la propriété de factorisation de Markov, la log vraisemblance négative pénalisée que l'on souhaite minimiser est

$$\mathcal{L}^n(\widehat{\mathcal{G}}, \tau, D) = -\frac{1}{n} \sum_{\ell=1}^{n} \sum_{j=1}^{d} \log q_j(x_j^{(\ell)} | x_{\mathrm{Pa}(j;\hat{\mathcal{G}})}^{(\ell)}, \tau_j) + \lambda_S |\widehat{\mathcal{G}}|, \qquad (5)$$

avec λ_S un hyperparamètre de la méthode et $|\widehat{\mathcal{G}}|$ le nombre total d'arcs dans le graphe candidat $\hat{\mathcal{G}}$.

Décomposition de la log-vraisemblance

On peut décomposer la log vraisemblance de chaque Markov kernel de la façon suivant (cf. fin du cours précédent) :

$$\log q_j(x_j^{(\ell)}|x_{\mathrm{Pa}(j;\hat{\mathcal{G}})}^{(\ell)}, \tau_j) = \log \frac{q_j(x_j^{(\ell)}|x_{\mathrm{Pa}(j;\hat{\mathcal{G}})}^{(\ell)}, \tau_j)}{p(x_j^{(\ell)}|x_{\mathrm{Pa}(j;\hat{\mathcal{G}})}^{(\ell)})} + \log \frac{p(x_j^{(\ell)}|x_{\mathrm{Pa}(j;\hat{\mathcal{G}})}^{(\ell)})}{p(x_j^{(\ell)}|\mathrm{x}_{-j}^{(\ell)})} + \log p(x_j^{(\ell)}|\mathrm{x}_{-j}^{(\ell)}) \quad (6)$$

Décomposition de la log-vraisemblance

- Chaque somme $\frac{1}{n}\sum_{\ell=1}^{n} \log\, p(x_j^{(\ell)}|x_{-j}^{(\ell)})$ converge vers la constante $H(X_j|X_{-j})$ quand n tend vers l'infini.
- D'après le slide 73, du cours précédent, si $X_{\overline{\mathrm{Pa}}(j;\hat{\mathcal{G}})}$ correspond à l'ensemble complémentaire de X_j et ses parent dans $\widehat{\mathcal{G}}$, alors $\frac{1}{n}\sum_{\ell=1}^{n} \log \frac{p(x_j^{(\ell)}|x_{-j}^{(\ell)})}{p(x_j^{(\ell)}|x_{\mathrm{Pa}(j;\hat{\mathcal{G}})}^{(\ell)})}$ est égal à l'information mutuelle conditionnelle empirique entre X_j et $X_{\overline{\mathrm{Pa}}(j;\hat{\mathcal{G}})}$, conditionné par $X_{\mathrm{Pa}(j;\hat{\mathcal{G}})}$:

$$\hat{I}^n(X_j, X_{\overline{\mathrm{Pa}}(j;\hat{\mathcal{G}})}|X_{\mathrm{Pa}(j;\hat{\mathcal{G}})}) = \frac{1}{n}\sum_{\ell=1}^{n} \log \frac{p(x_j^{(\ell)}, x_{\overline{\mathrm{Pa}}(j;\hat{\mathcal{G}})}^{(\ell)}|x_{\mathrm{Pa}(j;\hat{\mathcal{G}})}^{(\ell)})}{p(x_j^{(\ell)}|x_{\mathrm{Pa}(j;\hat{\mathcal{G}})}^{(\ell)})p(x_{\overline{\mathrm{Pa}}(j;\hat{\mathcal{G}})}^{(\ell)}|x_{\mathrm{Pa}(j;\hat{\mathcal{G}})}^{(\ell)})}. \tag{7}$$

Décomposition de la log-vraisemblance

- Le terme $\log \frac{q_j(x_j^{(\ell)}|x_{\mathsf{Pa}(j;\hat{\mathcal{G}})}^{(\ell)},\tau_j)}{p(x_j^{(\ell)}|x_{\mathsf{Pa}(j;\hat{\mathcal{G}})}^{(\ell)})}$, mesure la capacité du modèle de régression q_j à approcher $p(x_j^{(\ell)}|x_{\mathsf{Pa}(j;\hat{\mathcal{G}})}^{(\ell)})$ pour une selection de parents $X_{\mathsf{Pa}}(j;\hat{\mathcal{G}})$ donnée.
- Quand n tend vers l'infini, ce terme converge vers 0 si l'on considère un modèle génératif q_j suffisamment flexible pour approximer $p(x_j|x_{\mathsf{Pa}(j;\hat{\mathcal{G}})})$, même si $\widehat{\mathcal{G}} \neq \mathcal{G}$.
- En effet, d'après (Hyvärinen and Pajunen, 1999), il est toujours possible de trouver une fonction $\hat{f}_j$ telle que $X_j \sim \hat{f}_j(X_{\mathsf{Pa}(j;\hat{\mathcal{G}})}, E_j)$, avec $E_j \perp\!\!\!\perp X_{\mathsf{Pa}(j;\hat{\mathcal{G}})}$, correspondant à un modèle probabiliste conditionnel q_j tel que $q_j(x_j|x_{\mathsf{Pa}(j;\hat{\mathcal{G}})},\theta_j) = p(x_j|x_{\mathsf{Pa}(j;\hat{\mathcal{G}})})$.

Score de log-vraisemblance conditionnelle négative à minimiser

- Si on trouve des paramaètres $\tau = (\tau_1, \ldots, \tau_d)$ tels que chaque terme $\frac{1}{n}\sum_{\ell=1}^{n} \log \frac{q_j(x_j^{(\ell)} | x_{\mathsf{Pa}(j;\hat{\mathcal{G}})}^{(\ell)}, \tau_j)}{p(x_j^{(\ell)} | x_{\mathsf{Pa}(j;\hat{\mathcal{G}})}^{(\ell)})} \to 0$ pour $n \to \infty$ tendant vers l'infini, alors asymptotiquement on a :

$$\mathcal{L}^n(\widehat{\mathcal{G}}, \tau, D) \sim \sum_{j=1}^{d} \hat{I}^n(X_j, X_{\overline{\mathsf{Pa}(j;\hat{\mathcal{G}})}} | X_{\mathsf{Pa}(j;\hat{\mathcal{G}})}) + \lambda_S |\widehat{\mathcal{G}}| + cst. \qquad (8)$$

Identification de la classe d'équivalence de $\mathcal{G}$

- Dans ces conditions, si on suppose de plus que les hypothèses de Markov, *faithfulness* et *causal sufficiency* sont vérifiées, on a le résultat suivant d'après (Kalainathan et al., 2022) (voir preuve dans le papier):
  - Pour chaque DAG $\widehat{\mathcal{G}}$ dans la classe d'équivalence de $\mathcal{G}$,

$$\lim_{n\to\infty} \mathbb{P}(\mathcal{L}_S^n(\widehat{\mathcal{G}}, D) - \mathcal{L}_S^n(\mathcal{G}, D)) = 0.$$

  - Pour chaque DAG $\widehat{\mathcal{G}}$ qui n'est pas dans la classe d'équivalence de $\mathcal{G}$, il existe $\lambda_S > 0$ tel que:

$$\lim_{n\to\infty} \mathbb{P}(\mathcal{L}_S^n(\widehat{\mathcal{G}}, D) > \mathcal{L}_S^n(\mathcal{G}, D)) = 1.$$

- Donc asymptotiquement, les graphes dans la classe d'équivalence de $\mathcal{G}$ sont ceux avec le plus petit score $\mathcal{L}_S^n(\widehat{\mathcal{G}}, D)$ quand $n \to \infty$.

Approche par sélection de modèles pour la recherche du graphe causal

- Différentes méthodes de la littérature ont été proposées pour minimiser ce score global :

$$\mathcal{L}^n(\widehat{\mathcal{G}}, D) = -\frac{1}{n}\sum_{\ell=1}^{n}\sum_{j=1}^{d}\log q_j(x_j^{(\ell)}|x_{\mathrm{Pa}(j;\hat{\mathcal{G}})}^{(\ell)}, \tau_j) + \lambda_S|\widehat{\mathcal{G}}|. \qquad (9)$$

- Différents types de modèles q_j : modèles linéaire, modèles non-linéaires avec bruit additif, réseaux de neurones génératif, etc.
- Différentes façons pour explorer l'espace des DAGs $\widehat{\mathcal{G}}$ de façon à minimiser le score global :
  - avec un algorithme glouton (opérateur : ajouter un arc, enlever un arc, retourner un arc);
  - avec des techniques de descente de gradient globales dans l'espace des graphes.

Heuristiques de recherche locale dans l'espace des graphes

- Dans le cas gaussien multivarié, l'algorithme glouton GES (Chickering, 2002) a été proposé pour réalisé cette tâche. Dans cette approche, le FCM est modélisé comme :

$$X_i = \sum_{k \in \text{Pa}(i;\mathcal{G})} \beta_k X_k + E_i, \quad \text{pour } i = 1, \ldots, d \tag{10}$$

modèle

- Dans la méthode CAM (Bühlmann et al., 2014), le FCM est modélisé comme :

$$X_i = \sum_{k \in \text{Pa}(i;\mathcal{G})} f_k(X_k) + E_i, \quad \text{pour } i = 1, \ldots, d \tag{11}$$

modèle

- Ces deux méthodes effectuent une recherche locale dans l'espace des graphes pour retrouver le DAG qui minimise le score globale $\mathcal{L}^n(\widehat{\mathcal{G}}, D)$.

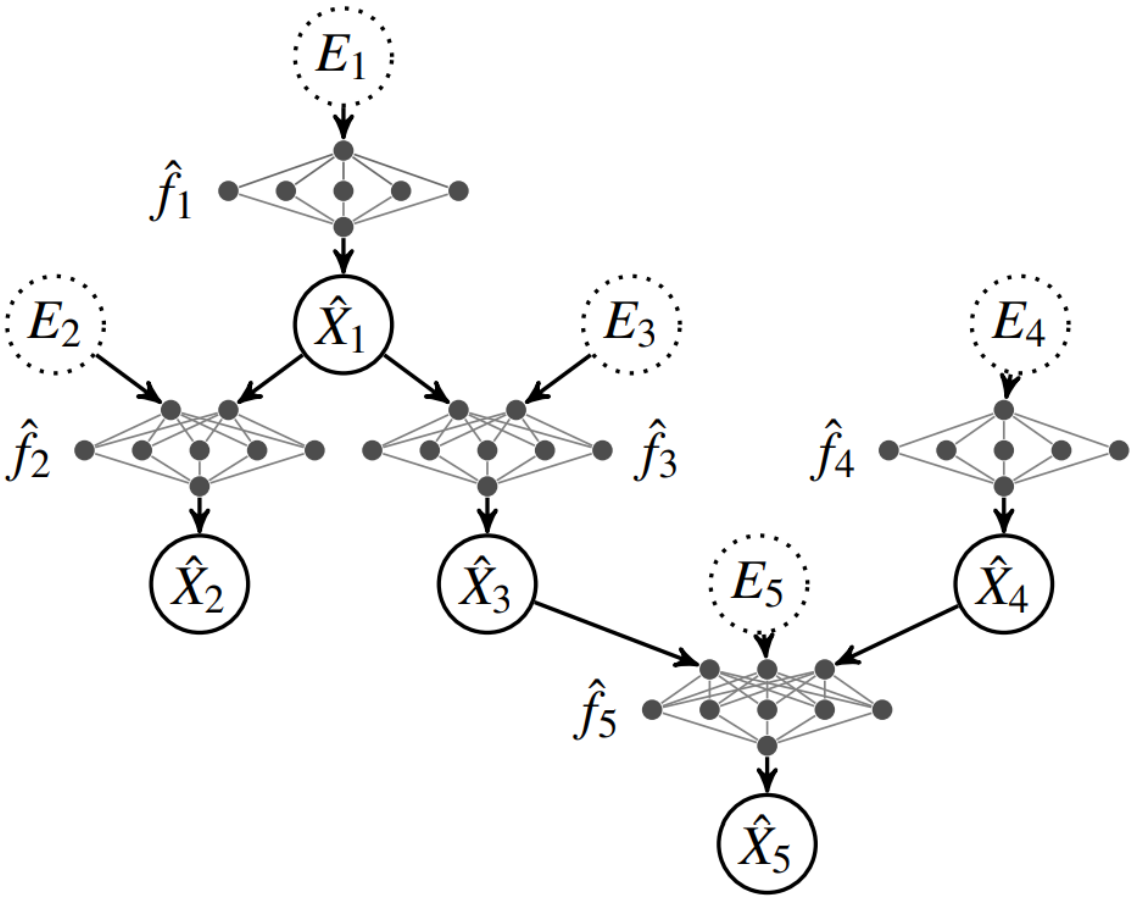
Causal Generative Neural Network (Goudet et al., 2017)

- Dans le cas général, sans faire d'hypothèse particulière sur la distribution des données.
- Apprentissage d'un modèle fonctionnel causal avec des réseaux de neurones génératifs.
- Modèle fonctionnel causal :

$$X_i \leftarrow f_i(X_{\mathrm{Pa}(i;\mathcal{G})}, E_i), \text{ for } i = 1, \ldots, d \tag{12}$$

- Idée : modéliser chaque mécanisme f_i par un réseau de neurones génératif.

Causal Generative Neural Network (CGNN)



$$\begin{cases} \hat{X}_1 = \hat{f}_1(E_1) \\ \hat{X}_2 = \hat{f}_2(\hat{X}_1, E_2) \\ \hat{X}_3 = \hat{f}_3(\hat{X}_1, E_3) \\ \hat{X}_4 = \hat{f}_4(E_4) \\ \hat{X}_5 = \hat{f}_5(\hat{X}_3, \hat{X}_4, E_5) \end{cases}$$

Heuristique de recherche globale dans l'espace des graphes

- Dans le cas gaussien multivarié, on peut utiliser le même type d'approche que la méthode du Glasso qui était utilisée pour apprendre un graphe non-dirigé.
- La matrice de précision de X est une fonction qui dépend des paramètres de B à apprendre à partir des données : $K(B) = (I - B)K_E(I - B)^T$.
- On note S l'estimée empirique de Cov(X), l'estimateur de vraisemblance du modèle gaussien graphique est (cf. cours 2) :

$$L(B) = \frac{1}{2} log(|K(B)|) - \frac{1}{2} \mathrm{Tr}(K(B)S) + cst \tag{13}$$

Apprentissage d'un graphe dirigé dans le cas gaussien multivarié

- Le problème a résoudre est :

$$\min_{B} \; -L(B) + \lambda \|B\|_1 + \mu \|B^d\|_1 \tag{14}$$

- On retrouve la même expression que pour la méthode Glasso (cf. cours précédent), mais avec en plus une contrainte d'acyclicité globale du graphe $\mu \|B^d\|_1$.
- Ce problème peut être résolue efficacement par descente de gradient directement sur les paramètres de la matrice B (voir par exemple Zheng et al. (2018)).

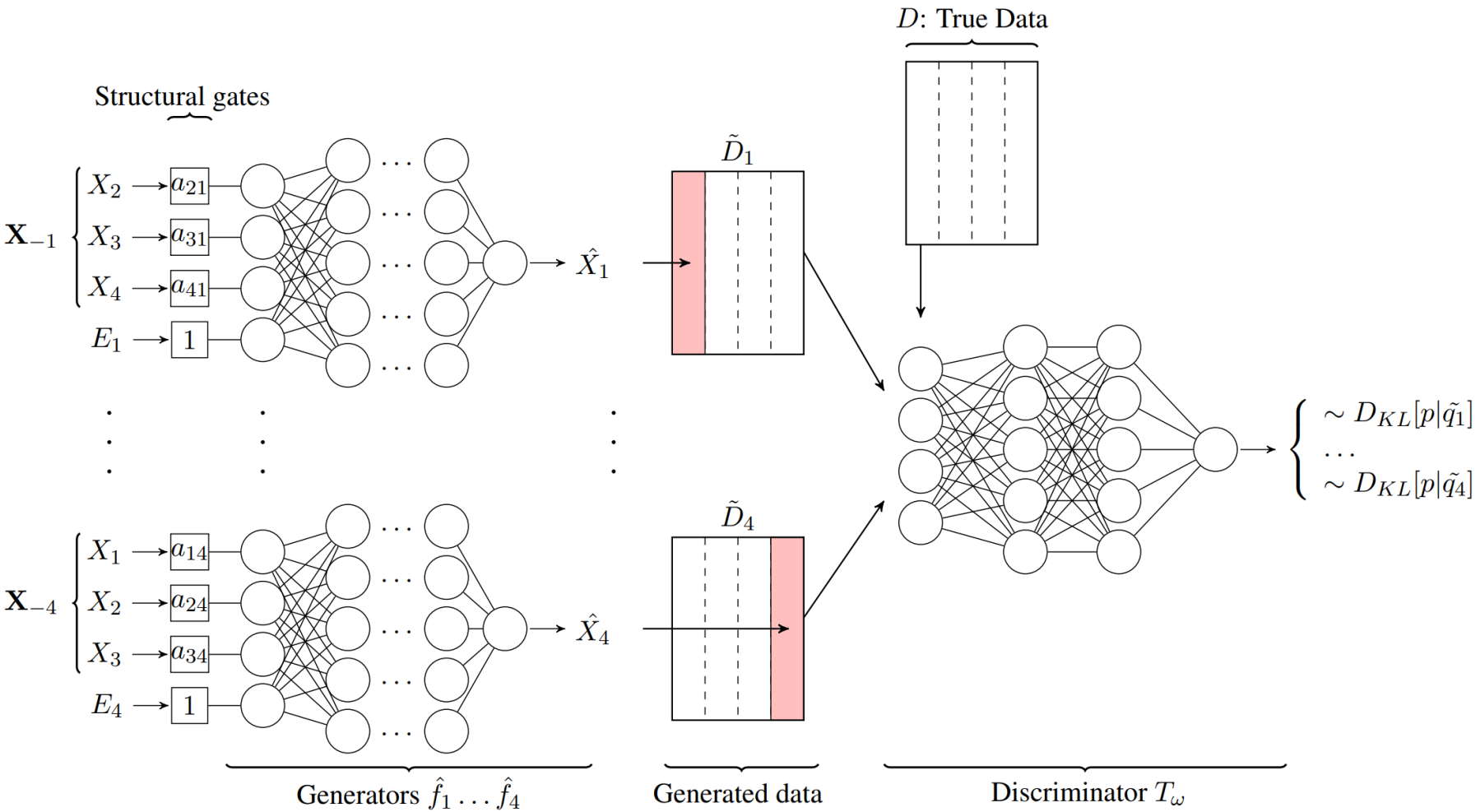
Structural agnostic model (SAM) (Kalainathan et al., 2022)

- Apprentissage d'un modèle fonctionnel causal avec des réseaux de neurones génératifs.
- Modèle fonctionnel causal :

$$X_i \leftarrow f_i(X_{\mathrm{Pa}(i;\mathcal{G})}, E_i), \text{ for } i = 1, \ldots, d \tag{15}$$

- Chaque mécanisme f_i par un réseau de neurones génératif.
- Un réseau de neurone discriminant (adversaire) est entraîné en simultané pour juger de la qualité de reproduction des données.
- Une contrainte d'acyclicité globale est imposée pour que le graphe soit un DAG.

Structural agnostic model (SAM)



Limites de ces algorithmes

- Toutes ces méthodes sont limitées à la recherche de la classe d'équivalence du DAG.
- En particulier si seulement deux variables sont observées, on ne peut pas identifier de v-structures.
- Est-il possible de retrouver quand même un DAG ?

Section 6

Références

Bühlmann, P., Peters, J., Ernest, J., et al. (2014). Cam: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 42(6):2526–2556.

Chickering, D. M. (2002). Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554.

Goudet, O., Kalainathan, D., Caillou, P., Lopez-Paz, D., Guyon, I., Sebag, M., Tritas, A., and Tubaro, P. (2017). Learning functional causal models with generative neural networks. *arXiv preprint arXiv:1709.05321*.

Hyvärinen, A. and Pajunen, P. (1999). Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439.

Kalainathan, D., Goudet, O., Guyon, I., Lopez-Paz, D., and Sebag, M. (2022). Structural agnostic modeling: Adversarial learning of causal graphs. *Journal of Machine Learning Research*, 23(219):1–62.

Lauritzen, S. L., Dawid, A. P., Larsen, B. N., and Leimer, H.-G. (1990). Independence properties of directed markov fields. *Networks*, 20(5):491–505.

Spirtes, P., Glymour, C. N., and Scheines, R. (2000). *Causation, prediction, and search*. MIT press.

Verma, T. and Pearl, J. (1990). Causal networks: Semantics and expressiveness. In *Machine intelligence and pattern recognition*, volume 9, pages 69–76. Elsevier.

Verma, T. and Pearl, J. (1991). *Equivalence and synthesis of causal models*. UCLA, Computer Science Department.

Zhang, K., Peters, J., Janzing, D., and Schölkopf, B. (2012). Kernel-based conditional independence test and application in causal discovery. *arXiv preprint arXiv:1202.3775*.

Zheng, X., Aragam, B., Ravikumar, P. K., and Xing, E. P. (2018). Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems*, 31.