Interpretable biological network reconstruction from observational data

Honghao Li

Université de Paris
Ecole Doctorale 130 - Informatique, Télécommunications et Electronique
Laboratoire Physico-Chimie Curie

Interpretable biological network reconstruction from observational data

par Honghao LI

Thèse de doctorat de Science des Données

Dirigée par Hervé ISAMBERT

Présentée et soutenue publiquement le 15 décembre 2021
devant un jury composé de :

Salem BENFERHAT	Professeur des Universités	Université d'Artois	Rapporteur
Loïc PAULEVE	Chargé de Recherche	LaBRI	Rapporteur
Antoine CHAMBAZ	Professeur des Universités	Université de Paris	Examinateur
Elisabeth REMY	Directrice de Recherche	Université d'Aix-Marseille	Examinatrice
Hervé ISAMBERT	Directeur de Recherche	Institut Curie	Directeur de thèse

Abstract

Title: Interpretable biological network reconstruction from observational data

Abstract: This thesis is focused on constraint-based methods, one of the basic types of causal structure learning algorithm. We use PC algorithm as a representative, for which we propose a simple and general modification that is applicable to any PC-derived methods. The modification ensures that all separating sets used during the skeleton reconstruction step to remove edges between conditionally independent variables remain consistent with respect to the final graph. It consists in iterating the structure learning algorithm while restricting the search of separating sets to those that are consistent with respect to the graph obtained at the end of the previous iteration. The restriction can be achieved with limited computational complexity with the help of block-cut tree decomposition of the graph skeleton. The enforcement of separating set consistency is found to increase the recall of constraint-based methods at the cost of precision, while keeping similar or better overall performance. It also improves the interpretability and explainability of the obtained graphical model.

We then introduce the recently developed constraint-based method MIIC, which adopts ideas from the maximum likelihood framework to improve the robustness and overall performance of the obtained graph. We discuss the characteristics and the limitations of MIIC, and propose several modifications that emphasize the interpretability of the obtained graph and the scalability of the algorithm. In particular, we implement the iterative approach to enforce separating set consistency, and opt for a conservative rule of orientation, and exploit the orientation probability feature of MIIC to extend the edge notation in the final graph to illustrate different causal implications. The MIIC algorithm is applied to a dataset of about 400 000 breast cancer records from the SEER database, as a large-scale real-life benchmark.

Keywords: Causal structure learning, Constraint-based method, separating set consistency, MIIC algorithm, interpretability, explainability, scalability, SEER database.

Résumé

Titre : Reconstruction de réseaux biologiques interprétables à partir de données d'observation

Résumé : Cette thèse porte sur les méthodes basées sur des contraintes. Nous présentons comme exemple l'algorithme PC, pour lequel nous proposons une modification qui garantit la cohérence des ensembles de séparation, utilisés pendant l'étape de reconstruction du squelette pour supprimer les arêtes entre les variables conditionnellement indépendantes, par rapport au graphe final. Elle consiste à itérer l'algorithme d'apprentissage de structure tout en limitant la recherche des ensembles de séparation à ceux qui sont cohérents par rapport au graphe obtenu à la fin de l'itération précédente. La contrainte peut être posée avec une complexité de calcul limitée à l'aide de la décomposition en block-cut tree du squelette du graphe. La modification permet d'augmenter le rappel au prix de la précision des méthodes basées sur des contraintes, tout en conservant une performance globale similaire ou supérieure. Elle améliore également l'interprétabilité et l'explicabilité du modèle graphique obtenu.

Nous présentons ensuite la méthode basée sur des contraintes MIIC, récemment développée, qui adopte les idées du cadre du maximum de vraisemblance pour améliorer la robustesse et la performance du graphe obtenu. Nous discutons les caractéristiques et les limites de MIIC, et proposons plusieurs modifications qui mettent l'accent sur l'interprétabilité du graphe obtenu et l'extensibilité de l'algorithme. En particulier, nous mettons en œuvre l'approche itérative pour renforcer la cohérence de l'ensemble de séparation, nous optons pour une règle d'orientation conservatrice et nous utilisons la probabilité d'orientation de MIIC pour étendre la notation des arêtes dans le graphe final afin d'illustrer différentes relations causales. L'algorithme MIIC est appliqué à un ensemble de données d'environ 400 000 dossiers de cancer du sein provenant de la base de données SEER, comme benchmark à grande échelle dans la vie réelle.

Mots clefs : Apprentissage de structures causales, méthode basée sur des contraintes, cohérence d'ensemble de séparation, algorithme MIIC, interprétabilité, explicabilité, extensibilité, base de données SEER.

Remerciements

Tout au long de ma carrière académique, j'ai été confronté à de nombreux choix et décisions, et j'ai toujours choisi la voie qui ne semble pas être la plus droite. Il y a 11 ans, alors que je venais de finir mes études au lycée, je me suis inscrite au programme de médecine clinique d'une école de médecine en Chine. 11 ans plus tard, je suis maintenant assis dans un bureau près du jardin de Curie, au centre de Paris, là où se trouvait Marie Curie, et je réfléchis à mon travail de doctorat en informatique. Depuis que j'ai quitté ma ville natale avec ma famille à l'âge de 10 ans, j'ai vécu dans différents endroits, et Paris est la ville où je suis resté le plus longtemps (si je peux compter Palaiseau comme Paris aussi). Je garde toujours un cercle social assez restreint où que j'aille, et j'apprécie chaque personne que j'ai rencontrée et avec laquelle j'ai construit une relation. Il est impossible de décrire pleinement dans ces courtes pages l'aide et le soutien, intentionnel ou non, que j'ai reçu de personnes qui me sont proches, de près ou de loin, et qui m'ont poussé à traverser ces trois longues années.

Je tiens à remercier tout d'abord les membres de mon jury de thèse. Loïc Paulevé et Salem Benferhat pour rapporter et évaluer mon travail, Elisabeth Remy pour présider le jury, et Antoine Chambaz pour la participation au jury. Merci à tous pour l'intérêt que vous avez porté à mon travail et pour les remarques pertinentes et les questions intéressantes durant la soutenance.

Je remercie chaleureusement mon directeur de thèse Hervé, qui m'a guidé ces trois dernières années à partir de mon stage de M2, sans qui aucun des travaux présentés ici ne serait possible. Je me souviens encore du jour où je vous ai demandé de faire le stage de M2 avec vous. Ayant une formation principalement en physique, j'étais inquiet de la transition transdisciplinaire vers l'informatique. Vous m'avez rassuré et encouragé en partageant votre expérience similaire, ce qui m'a donné la confiance nécessaire pour finalement étendre le stage à une thèse. Tout au long du projet, votre attitude optimiste et de soutien n'a pas changé. Je vous remercie profondément pour votre patience à avoir des discussions des heures avec moi, pour votre tolérance à mon entêtement sur les questions théoriques, et pour vos examens attentifs et vos commentaires détaillés de chacune de mes présentations.

Mes remerciements vont à l'ensemble du personnel du laboratoire Physico-Chimie

Curie, les directeurs successifs du laboratoire, l'équipe administrative et particulièrement à Giuliana. Je remercie également l'Université de Paris pour le soutien financier.

Je tiens à remercier tous les membres de l'équipe Isambert. J'ai toujours apprécié de travailler avec un groupe petit mais libre et énergique. Vincent, je te remercie pour ton aide précieuse au début de ma thèse, pour les beaux benchmarks que tu as réalisés pour nos travaux collectifs, et surtout, pour partager avec moi un même (et bon !) sens de l'humour. Nous n'avons pas commencé la thèse le même jour, mais nous la terminerons le même jour (Bonne chance !), et j'appelle cela le yuanfen. Ta passion pour le vélo me rappelle souvent mon vélo de route Schiwinn blanc que j'ai laissé aux États-Unis, et les balades passionnées que j'ai faites seul sur la Route 1. J'envie ton Specialized carbone. Marcel, je te remercie de m'avoir toujours soutenu, pour les discussions enrichissantes, que ce soit au travail ou en dehors, et pour les jeux de société que tu m'as prêtés, dont un que je garde encore chez moi. J'ai beaucoup apprécié les histoires de vie aventureuses de toi et de tes amis au Brésil. Tu es aussi la personne que je connais qui a le plus de retweets. Je te souhaite le meilleur pour les mois à venir, pour la vie de thésard, pour la vie parisienne avec Natália et pour le concours au Brésil. Nadir, je te remercie pour ton enthousiasme et ta sincérité, pour m'avoir invité chez toi avec des pizzas incroyables et des biscuits savoureux. Je remercie également les nouveaux membres de l'équipe : Franck, pour avoir supporté mes pinaillages sur les petits détails du code. Louise, pour la jeunesse que tu apportes au bureau (et maintenant je peux reconnaître le Scretlab chaque fois que je regarde une compétition de jeux vidéo). Liza, pour les perspectives uniques, rigoureuses et intéressantes que tu apportes au groupe. Je remercie tous les membres du groupe pour la surprise après la soutenance que vous avez préparée avec beaucoup de soin et qui m'a beaucoup touchée. Je profite également de l'occasion pour remercier Minh du même bureau, qui est toujours très gentil et poli.

Je remercie Michele Casula, mon tuteur de stage M1, Michel Ferrero, professeur du cours de physique numérique, Pierre et Noelle Wenger, ma famille d'accueil à Villeneuve-sur-Lot, mes amis EV2 à Palaiseau, Seb et Gabriel de Runity.

Je remercie profondément mes parents, qui ont fait tout ce qu'ils pouvaient pour m'offrir la meilleure éducation et les meilleures conditions de vie. Vous me soutenez toujours autant, même lorsque j'ai décidé de ne pas devenir docteur en médecine comme vous l'avez fait tous les deux. J'espère que vous n'êtes pas trop déçus de voir un docteur en informatique à la place.

Pour finir, je remercie ma femme, Wan'er, sans qui je ne peux pas imaginer comment serait ma vie et qui je deviendrais, you are the apple of my eye.

P.s. : Un merci spécial à mon meilleur assistant, M. MacBook Pro (Retina, 13-inch, Late 2013), un guerrier résistant.

致谢

一〇年大学入学前，我对招生老师说，汶川和海地的灾难让我决定报考临床医学专业。一年后，我回到了高中时的“老本行”物理系。如今，我坐在居里研究所一间老旧的办公室内，为三年信息学博士工作做总结。十岁时，我随父母从鄂西北迁往江浙沿海。过了通过随意哭泣来发泄情绪的年纪，却又还未来得及给自己装上青春期的叛逆，陌生的面孔与环境带给我深深的孤独感。如今，父母的良苦用心不言而喻，那份孤独感却伴随着“外地人”的标签时时萦绕着我。也许，它们正是我内心的驱动力，让我习惯一次次在道路分岔时选择看似更加昏暗的那条。于我而言，“故乡”是个模糊的概念，本应是对故乡的惦念被转移至每一个我在路上遇见的人。他们是乡间小路上头顶的繁星，是深夜公路上蒙着薄雾的路灯，有意无意，或近或远地为我照亮前路。

初中时，每一位老师对我的照顾与偏爱让我放下身份的焦虑，周围同学对我的包容让我的自信逐渐建立。即使后来我了解到，我在你们许多人心中都是一个“高冷”的人，我仍然感激你们曾试图接近。高中时那样纯粹的学习氛围，放在之后的任何一段经历中，都难以想象。同样难以想象的，是我未来妻子陈莞尔和挚友林祖谋的人生轨迹，都已在那个闷热的夏天与我第一次交汇。大一结束时，当我告诉父母打算从临床八年转至物理系时，我能从他们的惊讶中感受到些许失望，但他们对我的支持从未改变。大学在沪四年，来法亦已七年有余，我回家的次数从每年两次，每年一次，到如今已两年未归。视频电话另一端的父母总是面带笑容，而我也就尽力不让自己哽咽。最初被人问及为何选择法国，我也曾试图给予理想主义的回答，但想了想还是觉得“因为爱情”更为潇洒，而这也确实是主要原因，其中或许也夹杂着些许对法兰西自由浪漫的刻板印象。

感谢骑自行车送我去面试巴黎综合理工的陈帜，你把我直接送到了法国；感谢从邯郸路一路同行至帕莱索山头的顾陈琳、杜明星；感谢徐景科和他的免费健身课和二手健身视频；感谢吕昕作为我硕士第二年的学伴，感谢一同坚守在巴黎的丁文思、张泽坤夫妇和王一鸣；感谢刘畅、刘佳“兄妹”。

感谢我的妻子陈莞尔，无条件地支持我，接纳我的喜怒无常。

偶尔我也会想象，平行宇宙中在每一个岔路口选择不同的方向，会给我带来怎样的人生。每到这时，我都会想起初中时读到的那首弗罗斯特的小诗《未选择的路》。无论这些可能性多么精彩纷呈，它们共同的遗憾，大概就是失去和与我在现实生活中相遇之人相遇的机会吧。

The Road Not Taken

Two roads diverged in a yellow wood,
And sorry I could not travel both
And be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth;

Then took the other, as just as fair,
And having perhaps the better claim,
Because it was grassy and wanted wear;
Though as for that the passing there
Had worn them really about the same,

And both that morning equally lay
In leaves no step had trodden black.
Oh, I kept the first for another day!
Yet knowing how way leads on to way,
I doubted if I should ever come back.

I shall be telling this with a sigh
Somewhere ages and ages hence:
Two roads diverged in a wood,and I—
I took the one less traveled by,
And that has made all the difference.

Robert Frost

未选择的路

黄色的树林里分出两条路，
可惜我不能同时去涉足，
我在那路口久久伫立，
我向着一条路极目望去，
直到它消失在丛林深处。

但我却选了另外一条路，
它荒草萋萋，十分幽寂，
显得更诱人、更美丽；
虽然在这两条小路上，
都很少留下旅人的足迹；

虽然那天清晨落叶满地，
两条路都未经脚印污染。
呵，留下一条路等改日再见！
但我知道路径延绵无尽头，
恐怕我难以再回返。

也许多少年后在某个地方，
我将轻声叹息把往事回顾：
一片树林里分出两条路，
而我选了人迹更少的一条，
从此决定了我一生的道路。

[译] 顾子欣

Contents

1. Introduction

A fancy quote is required for an introductory chapter.

(A causal assumption)

Causality has been an eternal persuit of thoughtful minds throughout human history. It's the foundation of many ancient philosophies for the view of the world. When observing the environment around, we are not satisfied of merely descriptions, and causality lays the framework for connecting, ordering and classifying different events. It starts with a question of "why" to a phenomenon, usually an unexpected one like an eclipse, or sometimes one as common as the alternation of day and night. The answer of those questions points to some other phenomena which are less visible and regarded as more fundamental. Then by tracing the "source", a worldview is built. Aristotle attributes all of our knowledge about the world to *four causes*: material, formal, efficient and final, the four together should cover the explanations for any "why" questions about any phenomenon. In the eastern side, a more concentrated form of causality is preferred. In Buddhism, it can be roughly understood that all actions are driven by *karma*. In Daoism, *Taiji* is the undifferentiable singular initial state of the world, and is the cause of *Yin* and *Yang*, and thus of everything else.

Often alongside causality is the notion of time, one of the concepts that failed to be included in the causal framework. Indeed, there seems to be no answer for questions like "why is there time?", "what causes time?". Consequently, time is considered by many to be preceding the concept of causality, since the effect cannot precede in time the cause. After the birth of the theory of relativity, however, this picture is greatly altered. With the concepts of space and time fused together into the notion of spacetime, causality is intertwined with time more than ever. Gravity, one of the four foundamental forces of the universe, shapes the spacetime, thus giving time a proper "cause". Nevertheless, spacetime is imposing a constraint for all causal relationships, this time under the notion of light cone.

The entanglement reminds us of the metaphysical nature of causality, who has been a evergreen subject of philosophy. In science, though rarely brought up as an individual subject, the notion of causality is the implicit guideline for many disciplines, from

evolutionary biology to epidemiology, from high energy physics to cosmology. The early practice often involves drawing conclusion from experiences, observations, and by induction and inference. This practice has been followed until today, with increasingly powerful tools provided by scientific frameworks, often mathematical, that eventually form the subject of statistics. Correlation is one of the elemental tools in statistics that quantitatively connects the observations of two events. Cause and effect are undoubtedly correlated, therefore making it tempting to make use of the tool to quantify the otherwise qualitative claim of "A causes B". Quickly it is realised that correlation, as a symmetrical relation, only captures the most apparent dimension of the causality. Not only does it disregard the important asymmetry between cause and effect, but also it incorporates many more general associations than causation: Berkson's paradox, Simpson's paradox, indirect cause and effect, or merely coincidence. All these remarks lead to the canon of statistical study: correlation does not imply causation.

Statistics is largely about collecting and analysing numbers, with the latter involving primarily the aspects of modeling and interpreting the data. Causality is just about an inevitable word when it comes to interpretation. Unfortunately, owing to the aforesaid difficulty of correlation, and to the complication of the proper collection of data through randomized experiments that may alleviate those problems, the branch of interpretation has been seemingly less persuasive than its modeling counterpart, often with prudent associative descriptions, avoiding bold causal claims. Nonetheless, for real applications, the attempt to causally connect two correlated factors, intentionally or unintentionally, will not be easily stopped. As remarked by the authors in [34]:

> [...] We are struck by the fact that in the social and behavioral sciences, epidemiology, economics, market research, engineering, and even applied physics, statistical methods are routinely used to justify causal inferences from data not obtained from randomized experiments, and sample statistics are used to predict the effects of policies, manipulations, or experiments. [...]

Indeed, in front of a chart showing a strong correlation between a drug and the cure of a severe disease, an expert of causal inference or a college student who has taken some courses in statistics may remain skeptical, but a representative of a pharmaceutical company or a young couple desperately trying to save their sick baby will probably be very excited. This gap between theory and application motivates the authors of [34], along with many others, to establish a framework, mathematical but comprehensible to experts and laymen alike, that clarifies the conditions under which a causal statement can be made from statistical studies, particularly in the absence of randomized experiments. Ever since, the causality, as a proper discipline of its own in statistics, has been under

rapid development. It's undeniable that causality has been drawing the attention of many other subjects such as machine learning, mostly because, as Pearl puts it, "causality has been mathematized" [25]. Today, putting into a search engine the word "causality" or "causal" combined with a word specific to a random research field, we are likely find one or more publications that come out within the recent five years.

The curiosity for causality has been encoded into the DNA of human beings ever since we start to reflect on the world around us. We started by making observations, then we use causality to connects individual observations. Not satisfsed with qualitative descriptions, we turn to mathematics to quantify the causality. Limited by the notion of correlation, new frameworks are built to mathematize the causality. Along the way never changed is the twofold nature of causality. On the one hand, we want to draw causal conclusions from our past experiences, through observations, experiences and data. On the other hand, we want to validate our conclusions by predict the future with the help of the recovered causal relations.

The theories for formulating causality has been constantly evolving. Aristotle's four causes assign the trait of cause and effect to objects. Then, with the introduction of time, causality is viewed as a characteristic of a mechanism, a process, that involves but precedes specific objects [25]. The theory of conterfactuals is built to give a standard description of causal relations. Lacking the mathematical foundation, it relies mainly on assertions of logics and semantics, emphasizing the difference on the effect were the cause to be altered. Taking as example the self-contained quote at the beginning of the chapter, the introduction chapter is the cause of the quote, then the conterfactual description of this causal relation would be "if it were not an introduction chapter, then the quote would not have been there". Later, with the advancing of statistics, quantitive approaches emerge. Following the spirit of conterfactuals is the potential outcome framework, in which the causal effect is measured in terms of the difference in the outcome (effect) under different conditions of input (cause). Meanwhile, the adoption of probability and random variable allows for framing observations of events in a non-deterministic way. Under this notation, each observation is related to a random variable X following a distribution $P(X)$, and the observed value x is called a realisation of the random variable X. From this purely probabilistic point of view, if a random variable X is the cause of another random variable Y, then the occurrence of the cause $(X = x)$ would increase the probability of the occurrence of the effect $(\Pr(Y = y \mid X = x))$, as compared to that when the information on the cause is unknown $(\Pr(Y = y))$. This simple model, though failing to account for factors like confounding variables, is a giant leap in the evolution of the theory of causality, as it gives a mathematical description of a necessary condition for defining a causal relation. Another important framework is the graphical model,

where asymmetrical relation between cause and effect is naturally analogued to an arrow pointing from the cause to the effect. Combining probability with the well developed toolkit of graph provides a powerful framework that can formulate causality in a way that is both mathematical and intuitive. The study of causality with graphical model is usually twofold: to infer causal relationships from a graphical model, and to learn the underlying graphical model from a experimental study that is either observational, interventional or mixed. The two aspects are nonetheless closely related. In this thesis, we focus mainly on the aspect of causal structure learning.

The thesis is structured as follows. In chapter 2 we first present the shortcomings of constraint-based methods for causal structure learning, particularly in terms of interpretability, then we proposed an algorithm that may improve the interpretability of many state-of-the-art methods. We follow by a discussion comparing with score-based approaches. In chapter 3 we introduce a constraint-based information theoretic method called MIIC, then describe the integration of the proposed algorithm in the MIIC algorithm, while emphasizing the interpretability, and we detail the implementation of the proposed algorithms. Then in chapter 4 we show the application of the MIIC algorithm to the SEER breast cancer database, followed by discussions on the obtained networks and reflections of the causal relations found in the networks.

2. Interpretability of Constraint-Based Causal Structure Learning

2.1. Graphical model

Let $X = \{ X_i \}_{i=1}^{M}, M \in \mathbb{Z}^+$ be a set of random variables, $\mathcal{P}_X$ the power set of X, $\mathcal{Q}_X$ the set of pairwise disjoint triple of subsets of X, a statistical model usually involves the probability measure $P(X)$ (and thus $P(S)$ for all $S \in \mathcal{P}_X$) and a set of conditional independence relations $\{ (A \perp\!\!\!\perp_p B \mid C) \mid (A, B, C) \in \mathcal{Q}_X \}$, where $\perp\!\!\!\perp_p$ means independent in terms of probability.

Loosely speaking, a graphical model is a statistical model, but with some extra constraints imposed upon via a graph G. Specifically, the graph $G(V, E)$ is constructed with the set of variables $V = X$ as the set of vertices, and the statistical model must satisfy the *global Markov property* with respect to G:

$$A \perp\!\!\!\perp_g^G B \mid C \implies A \perp\!\!\!\perp_p B \mid C \tag{2.1}$$

where $\perp\!\!\!\perp_g^G$ means independent in terms of graph and relative to G, and is defined differently depending on the type of graph of G. For the sake of simplicity, in the following, we use the index i to represent the vertex of G that corresponds to the random variable X_i. If G is an undirected graph, then $\perp\!\!\!\perp_g$ is defined by *separation*. For $(A, B, C) \in \mathcal{Q}_V$, A and B are separated by C if for all $i \in A$ and all $j \in B$, every path $\gamma_{i;j}$ between i and j passes through at least one vertex $k \in C$. For all $i \in V$, denote by $\mathrm{adj}_G(i)$ the set of adjacent vertices of i in G, and by $\mathrm{nadj}_G(i) = V \setminus (\{ i \} \cup \mathrm{adj}_G(i))$ the set of non-adjacent vertices of i in G excluding i itself, an important corollary is the *local Markov property* of the statistical model with respect to G:

$$i \perp\!\!\!\perp_p \mathrm{nadj}_G(i) \mid \mathrm{adj}_G(i). \tag{2.2}$$

Another straightforward corollary of the above definition is

$$i \perp\!\!\!\perp_g^G j \mid C \implies i \perp\!\!\!\perp_p j \mid C. \tag{2.3}$$

That is, each non-adjacency relation in G ($(i,j) \notin E$, denoted by $i \nsim j$) implies a conditional independence relation in the statistical model.

If G is a directed graph, the definition of $\perp\!\!\!\perp_g$ is adapted to *d-separation* (directional separation). In particular, we focus on the case where G is a *directed acyclic graph* (DAG). For a path in G over a set of vertices $\{i, j, k\}$, j is called a *collider* if $i \to j \leftarrow k$, otherwise j is called a *non-collider*. Then, a path $\gamma_{i;k}$ is said to be *d-connected* by a set $C \subset V$ if and only if for all vertices j (except i and k) on $\gamma_{i;k}$,

1. if j is a non-collider, then $j \notin C$, and
2. if j is a collider, then at least one descendant of j is in C.

And $\gamma_{i;k}$ is said to be d-separated (or blocked) by C if it is not d-connected by C. Note that for convenience, we adopt the definition where j is a descendant of itself. Finally, for $(A, B, C) \in \mathcal{Q}_V$, A and B are d-separated by C if for all $i \in A$ and all $j \in B$, all paths $\gamma_{i;j}$ between i and j are blocked by C. For all $i \in V$, denote by $\mathrm{PA}_G(i)$ the set of parent vertices of i in G, and by $\mathrm{DE}_G(i)$ the set of descendant vertices of i in G, then the local Markov property of the statistical model with respect to a DAG G is

$$i \perp\!\!\!\perp_p V \setminus (\mathrm{DE}_G(i) \cup \mathrm{PA}_G(i)) \mid \mathrm{PA}_G(i). \tag{2.4}$$

Similarly, the implication of conditional independence in eq. (2.3) still holds.

Another important property of a DAG graphical model is the *factorization* of the probability density function of the associated statistical model. For the sake of simplicity, we suppose that all X_is are discrete. For a random variable X_i, denote by Ω_i the sample space of X_i, then the joint sample space for the probability distribution $P(X)$ is defined as $\Omega \coloneqq \prod_{i=1}^{M} \Omega_i$. Denote by $p : \Omega \to [0, 1]$ the probability density function of X, the factorization of p takes the form

$$p(x) = \prod_{i \in V} p_{i|\mathrm{PA}_G(i)}(x_i \mid \mathrm{pa}_G(i)) \quad \text{for all } x \in \Omega, \tag{2.5}$$

where $p_{i|\mathrm{PA}_G(i)}$ denotes the marginal probability density function. The factorization can be intuitively connected to another framework of causality modeling, the structural equation modeling, where each random variable X_i is regarded as being generated by a function $f_i : \Omega_{\mathrm{PA}_G(i)} \times \Omega_{U_i} \to \Omega_i$, with U_i the random variable modeling the error (noise).

A DAG graphical model is also called a Bayesian network for its integration with the statistical model. It is also a great model for the modeling of causality as it has an intuitive interpretation. For a DAG G and vertex $i \in G$, $\mathrm{PA}_G(i)$ is considered as the set of direct causes of i. In this sense, every d-separation relation $i \perp\!\!\!\perp_g^G j \mid C$ where $C \neq \emptyset$

encodes a non-causal correlation since $i \not\perp\!\!\!\perp_p j \mid \emptyset$.

For causal inference, the associated statistical model $P(X)$ provides a framework, with which we are able to answer various questions raised earlier in the chapter. Specifically, with the parameterization of $P(X)$, we are able to give the probability $\Pr(x)$ for any $x \in \Omega$. Therefore, for a discrete random variable X_i with $\Omega_{\mathrm{PA}_G(i)} = \{0, 1\}$, in an observational experiment where only $\mathrm{pa}_G(i) = 0$ is observable, we are able to give the conterfactual probability $\Pr(x_i \mid \mathrm{pa}_G(i) = 1) = p_{i|\mathrm{PA}_G(i)}(x_i \mid 1)$.

For intervention, the factorization in eq. (2.5) allows for a simple representation. In a classical senario where the intervention is targeted only at a random variable X_i, the statistical model after the intervention can be written as, in the language of *do-intervention*,

$$p(x \mid \mathrm{do}(x_j)) = f(x_j) \prod_{i \in V, i \neq j} p_{i|\mathrm{PA}_G(i)}(x_i \mid \mathrm{pa}_G(i)). \tag{2.6}$$

Graphically, the intervention on X_i corresponds to a DAG $G'(V, E')$ where all edges between X_i and its parent vertices are removed, that is, $E' = E \setminus \{ (j, i) \mid j \in \mathrm{PA}_G(i) \}$.

2.2. Causal structure learning

As mentioned earlier, another important aspect of causality modeling, apart from causal inference based on graphical models, is the learning of the causal structure itself. Generally speaking, given a dataset, causal structure learning concerns assigning to the dataset a graphical model that captures the causal relationships hidden in the data. Usually the first step is the pre-processing of data, where a data table of N_0 rows and M_0 columns is processed depending on different assumptions and requirements. Typically this includes the removal of NA values, the removal (or regrouping) of duplicated columns or rows and the regrouping of values, etc., such that at the end of the step, a table D of N rows and M columns is given, in which each column corresponds to a random variable X_i with sample space Ω_i, and the value at the jth row $x_i^j \in \Omega_i$ is a realisation of X_i. Depending on different data types (interventional, observational, mixed), different variable types (continuous, discrete, mixed) and different assumptions (acyclicity, background knowledge), different methods can learn different types of graphical models.

Interventional data is very informational. For example, the randomized controlled trials can be seen as a intervention on the treatment variable, often supposed to be a do-intervention, and is considered as the gold standard in many areas of application to measure the effect of treatment on the target, be it the cure of a disease or the expression of a certain gene, which can be viewed as pairwise cause and effect modeling. However, in practice, an ideal intervention is quite hard to come by, as often hindered by latent

confounding or selection variables, or more practically, by ethical or budgetary concerns.

Here we focus on the type of data that is purely observational. Given such a dataset D, there are various methods to construct a statistical model $P(X)$ over the set of random variables $X = \{ X_i \}_{i=1}^{M}$. Generally speaking, it involves learning the parametric structure of $P(X)$ by the maximization (or minimization) of a target function, e.g., the likelihood function, the a posteriori probability, or the loss function of a regression model, with a penalty term to avoid overfitting.

Next, to construct a graphical model from this statistical model, additional assumptions are required. The global Markov condition given by eq. (2.1) is not enough, as, naively speaking, it only allows for validating the d-separation relations in a DAG by statistical conditional independence ($\perp\!\!\!\perp_g \implies \perp\!\!\!\perp_p$), but not for constructing a d-separation relation from a conditional independence ($\perp\!\!\!\perp_p \implies \perp\!\!\!\perp_g$). Indeed, there may exist situation of cancelling causal effects where a variable A may directly or indirectly, and positively or negatively impact another variable B in more than two ways that cancel each other out, such that the combined effect is not observable, and A and B appear to be statistically (conditionally) independent. Under this situation, the construction of a graphical model is unfeasible without external intervention on the system. As a result, a necessary assumption is the *causal faithfulness assumption*, which rules out the "ill" situations as described above. Mathematically, the causal faithfulness assumption can simply be expressed as the converse of eq. (2.1). Together with the global Markov property, it forms a bijection between the set of d-separations in a DAG and the set of conditional independence relations in a statistical model. Then with this bijection, are we now ready to reconstruct a DAG from a statistical model? The answer to this question is unfortunately no, as the function from the space of DAGs to the space of set of d-separations (over a fixed set of variables) is neither injective nor surjective.

Firstly, multiple DAGs can have the same set of d-separations (non-injective). This means that given a set of d-separations, there can be multiple DAGs that comply to all elements of the set. In other words, these DAGs are indistinguishable (relative to the set of d-separations). As a compromise, we may group these indistinguishable DAGs into one graph that is considered as the representation of the group of DAGs, so as to establish an injection from the set of representative graphs to the set of d-separations. Obviously, the simplest implementation of a representation is a randomly chosen DAG from the group. From a deterministic point of view, however, this choice brings unnecessary stochasticity to the model. Also, the bit of "extra" information brought by the chosen DAG is purely artificial, and cannot be validated from only observational studies. As a result, a new class of graphs is introduced as the class of representative graphs of DAGs, the *completely partially directed acyclic graph* (CPDAG). Each CPDAG is said to represent a *Markov*

equivalent class of DAGs, denoted by $\mathcal{M}(G)$ where G is a DAG in the equivalent class, and all DAGs represented by the same CPDAG are said to be Markov equivalent to each other.

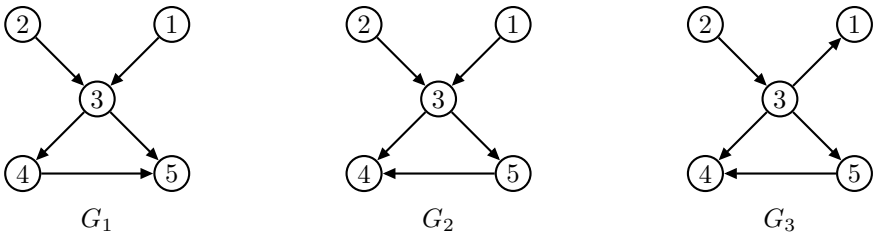

Figure 2.1. – Markov equivalence between DAGs. G_1 and G_2 belong to the same Markov equivalent class ($\mathcal{M}(G_2) = \mathcal{M}(G_1)$), whereas G_3 does not belong to the same class ($\mathcal{M}(G_3) \neq \mathcal{M}(G_1)$).

In practice, the indistinguishability within a set of DAGs comes down to the uncertainty between $i \to j$ and $i \leftarrow j$. Consequently, a CPDAG is constructed simply by combining all indistinguishable DAGs, keeping the common edges, and mark all uncertain edges as $i - j$. That is, for each endpoint of an uncertain edge in a CPDAG, there is at least one Markov equivalent DAG with an arrowhead on the endpoint, and at least one Markov equivalent DAG with an arrowtail on the endpoint. CPDAG resolves the non-injectivity issue of DAG with the set of d-separations, and hence, under the causal faithfulness assumption, with the statistical model. Each CPDAG uniquely maps to a set of d-separations, and vice versa. But in practice, how to tell if two DAGs belong to the same Markov equivalent class? Exausting and comparing the sets of d-separations is one evident answer, but is rather tedious, and most importantly, ungraphical. Fortunately, there is a simple graphical criterion for the determination of Markov equivalent class. To give the criterion, two additional concepts need to be introduced. The first concept is the *skeleton* of a DAG G, which is an undirected graph constructed by replacing all edges in G by undirected edges. From eq. (2.3), it's straightforward that two Markov equivalent DAGs must share the same skeleton. Then the second concept is a local structure of a DAG G, called the *v-structure*, defined as a set of three vertices in G, including a collider j and its two parents i, k, such that $i \not\sim k$ in G. Graphically, a v-structure is of the form $i \to j \leftarrow k$. As seen in the previous section, v-structure plays an important role in the definition of d-separation. Given these two concepts, Verma and Pearl [36] give the sufficient and necessary condition to identify a CPDAG:

Theorem 1 (Verma and Pearl (1991))**.** *Two DAGs are Markov equivalent if and only if they have the same skeleton and the same set of v-structures.*

Then there remains the problem of non-surjectivity, which means that there are situations where a set of d-separations cannot be represented by any DAG. Typically, this happens when there are unobserved (latent) variables, leaving the set of modeled variables a only subset of the full set of variables. The fact that the subset of a set of d-separations that is representable by a (CP)DAG is itself not always representable by a (CP)DAG is technically termed as not *closed under marginalization*. Practically, this means that (CP)DAGs are not suitable for modeling systems with latent variables. Meanwhile, latent variables can bring various consequences, including confounding and selection bias, that skew the reconstructed graphical model. Moreover, a complete observation of all relevant variables of interest only exists in the ideal setup, and in real applications we can never completely rule out the existence of latent variables.

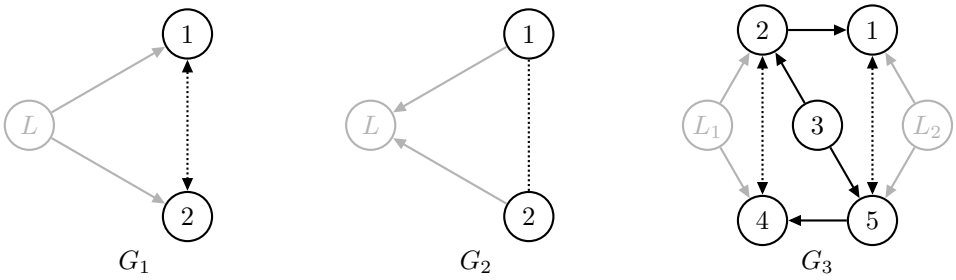

Figure 2.2. – Difficulties of (CP)DAGs in the presence of latent variables. In G_1, the latent confounding variable L creates a seemingly bidirected edge between 1 and 2, whereas in G_2 the latent selection bias L creates a seemingly undirected edge where neither endpoint can be an arrowhead. In G_3, the presence of two latent confounding variables L_1 and L_2 breaks the local Markov property (eq. (2.2)) for the pair $(1, 4)$, since we have $1 \not\perp\!\!\!\perp_d 4 \,|\, \{2, 5\}$ and $1 \perp\!\!\!\perp_d 4 \,|\, \{2, 3, 5\}$, and 3 is not adjacent to neither 1 nor 4.

Faced with difficulty, there are often two solutions. The first is a "passive" solution by imposing an additional assumption on the studied system, called the *causal sufficiency* assumption, which assumes no presence of such latent variables. Given a specific context and enough background knowledge, this assumption can sometimes be reasonable. Under the causal sufficiency assumption, CPDAG can be used as the class of graph for building a graphical model.

Then the second solution is about introducing yet another class of graphs that are closed under marginalization. This leads to the idea of *max ancestral graph* (MAG). Constrat to a DAG, a MAG allows for three types of edges: undirected $i - j$, directed, $i \to j$ and bidirected $i \leftrightarrow j$, where a bidirected edge implies a latent confounding variable and a undirected edge implies a latent selection variable that is implicitly conditioned on. Accordingly, the definition of d-separation in a DAG is adapted to *m-separation* in a

MAG. And similarly, the function from the set of MAGs to the set of m-separations is not injective, and the counterpart of CPDAG for a MAG is introduced as *partial ancestral graph* (PAG).

In the following, we focus on the methods that assume causal sufficiency and causal faithfulness. That is, we assume there is no latent variables in the studied system, and all potential d-separations can be captured by a statistical model. Under these assumptions, there are various causal structure learning methods that can be roughly divided into two categories.

2.2.1. Score-based causal structure learning

The first category of methods is the score-based methods. The core idea is that the causal sufficiency and the causal faithfulness assumptions, together with eq. (2.1), simplify the problem of graphical model learning to the problem of statistical model learning. In other words, it suffices to learn from the data the best parameterization $\theta(G, D)$, then the above assumptions ensure that the corresponding set of d-separations, hence the corresponding CPDAG, will be the best graphical model. The best parameterization is chosen based on a target function $l : \Omega_\theta \to \mathbb{R}$. In turn, each CPDAG G gets assigned a score $s_G = l(\theta_G)$, hence the name score-based. Finally, by choosing $\theta^* = \text{argmax}_{\theta \in \Omega_\theta} l(\theta)$, a corresponding CPDAG G^* can be constructed as the learned graphical model.

Though conceptually straightforward, the implementation is not trivial. The main difficulty lies in the cardinality of the search space Ω_θ (basically the number of CPDAGs) that grows exponentially with respect to the number of variables M. To circumvent the traversal of the whole search space, various greedy and heuristic algorithms [9, 23] are introduced, the details of which are beyond the scope of this thesis.

One characteristic, sometimes also the limit, of score-based causal structure learning is its reliance on the causal sufficiency assumption. On the one hand, this assures a CPDAG as final graph, which facilitates the causal inference part that follows after the structure learning. On the other hand, it largely limits the application of the methods in practice, since the presence of latent variable is often inevitable for real applications, and score-based methods have difficulty defining a score for the class of MAGs (PAGs), as there is no mapping to the statistical model from the set of m-separations like that from the set of d-separations in eq. (2.1). Though recently there are also attempts to include latent variables into the score-based framework under specific context [6].

2.2.2. Constraint-based causal structure learning

The second category of methods is the constraint-based methods. Contrary to the global view of score-based methods that search for the whole CPDAG, constraint-based methods, as the name suggests, rely on various local constraints to build the graphical model from scratch. One of the classics of constraint-based methods is the PC algorithm [33], as roughly described in algorithm 1, and sketched in fig. 2.3.

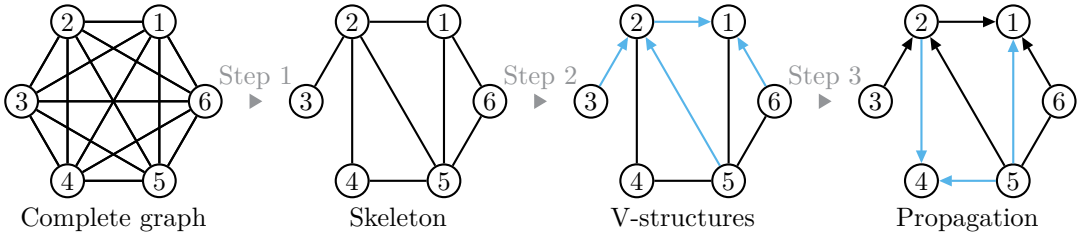

Figure 2.3. – General procedure of constraint-based methods.

The key idea of constraint-based methods is eq. (2.3) with causal faithfulness assumption. Recall that for a set of random variables V with an underlying graphical model $G(V, E)$, a statistical conditional independence relation between two distinct vertices $i, j \in V$, $i \perp\!\!\!\perp_p j \mid C$, ensures $i \not\sim j$ in G. As a result, the set of all statistical conditional independence relations among variables in V encodes all missing edges in G. Figuratively speaking, each of these conditional independence relations is like a LEGO® brick, and counterintuitively, building the graphical model "from scratch" here means building from a complete undirected graph, rather than from an empty graph. Then with each of these bricks, we knock one edge off the initially complete graph. Consequently, when all the bricks are used, we are left with a undirected graph which is the skeleton of the underlying graphical model.

Algorithm 1 Sketch of the PC algorithm

Require: Set of variables $V = \{ X_i \}_{i=1}^{M}$, dataset $D(V)$, significance level α.

Step 1: Find the graph skeleton and separating sets of removed edges;

Step 2: Identify and orient v-structures based on separating sets;

Step 3: Orient as many undirected edges as possible based on rules of propagation;

return Output graph $G(V, E)$.

Next step, if assuming causal sufficiency, is to further indentify the underlying CPDAG that corresponds to the reconstructed skeleton, for which theorem 1 demands that all v-structures of G be identified. To begin with, we need to pick from the skeleton of G all *unshielded triples*. An unshielded triple is a subset of three vertices $U = \{ i, j, k \} \subseteq V$,

such that the skeleton of the induced graph $G[U]$ has exactly two edges. In other words, it is identical to the skeleton of a v-structure. Graphically, it is of the form $i - j - k$. Apart from the v-structure $i \to j \leftarrow k$, there are two other possible types of structures that share the same unshielded triple: the *chain* $i \to j \to k$ and the *fork* $i \leftarrow j \to k$ (by symmetry the chain $i \leftarrow j \leftarrow k$ is equivalent to $i \to j \to k$). Therefore the task is to determine which of those unshielded triples belong to v-structures, and which belong to chains and forks. The key to the solution lies in the definition of d-separation. In the following, we always assume the causal faithfulness, and with eq. (2.3), $\perp\!\!\!\perp_g$ and $\perp\!\!\!\perp_p$ are equivalent, and will be uniformly denoted by $\perp\!\!\!\perp$.

Firstly, from the structure $i \not\sim k$ it follows immediately that there exists a set $C \subseteq V \setminus \{ i, k \}$ such that $i \perp\!\!\!\perp k \,|\, C$. According to the first step of the algorithm, we know that during the first step, we must have removed the edge between i and k using one of the bricks of statistical conditional independence relations, denoted by $i \perp\!\!\!\perp k \,|\, \mathrm{Sep}_{i;k}$, where $\mathrm{Sep}_{i;k} \subseteq V \setminus \{ i, k \}$, called the *separating set*, denotes the particular set that is used during the reconstruction of skeleton. In general, $\mathrm{Sep}_{i;k}$ is not unique [1], so we do not necessarily have $C = \mathrm{Sep}_{i;k}$. Still, the definition of d-separation requires that any C, hence any $\mathrm{Sep}_{i;k}$, blocks all paths between i and k in G. Subsequently, observation 2 and corollary 3 help distinguish v-structure from fork and chain structures:

Observation 2 (on unshielded triples)**.**

j d-connects i and k in v-structure $i \to j \leftarrow k$;

j d-separates i and k in chain $i \to j \to k$ and fork $i \leftarrow j \to k$.

Corollary 3 (identification of v-structure)**.** *Let $t : i - j - k$ be an unshielded triple in the skeleton of a DAG G, associated with the statistical conditional independence relation $i \perp\!\!\!\perp k \,|\, Sep_{i;k}$. t corresponds to a v-structure $i \to j \leftarrow k$ in $\mathcal{M}(G)$ if and only if $j \notin Sep_{i;k}$.*

It then follows as step 2 to check for all unshielded triples in G and orient all v-structures. Under the causal sufficiency assumption, i and k in the v-structure $i \to j \leftarrow k$ can be interpreted as the cause of j, that is, we have drawn causal conclusions from purely observational data. Indeed, graphically, the orientations in a v-structure seem to come out of nowhere. This widely conflicts with the traditional belief of "no causation without manipulation" [17]. In the absence of the causal sufficiency assumption, the CPDAG framework no longer applies. Instead, we need to refer to the aforementioned PAG (equivalent class of MAGs) framework, for which several defnitions and conclusions need to be adapted. In particular, [4, Theorem 2.1] gives the adapted version of theorem 1

1. For example, for the pair (i, l) in the graph $i \to j \to k \to l$, $\{ j \}, \{ k \}, \{ j, k \}$ are all valid candidates for $\mathrm{Sep}_{i;l}$.

for indentifying Markov equivalent class of MAGs under m-separation, in which the uniqueness of the set of v-structures is still a necessary condition. In that case, the causal conclusions cannot be drawn due to potential latent variables, but the arrowheads at j still hold. And we instead draw the conclusion that j is a non-ancestor of i or k (or not a direct or indirect cause of them). Graphically, this is marked by the two arrowheads at j, whereas the endpoints at i and k are undetermined.

Under the same condition, corollary 3 can be reformulated as "$i - j - k$ is a non-v-structure if and only if $j \in \mathrm{Sep}_{i;k}$". Since a non-v-structure can be either a fork or a chain, the natural question to follow is whether we can distinguish between a chain and a fork, the answer to which is negative. According to theorem 1, given an unshielded triple $i - j - k$, all possible non-v-structures encode the same information of d-separation, as they have the same skeleton (of the induced graph) and the same set of v-structures (the empty set). Therefore, this reformulation is not as magical as corollary 3 as to orient edges on its own. Nonetheless, in presence of already established v-structures, and under the causal sufficiency assumption, a chain structure can be identified.

Corollary 4 (identification of chain)**.** *Let $t : i \to j \leftarrow k$ be a v-structure in a graph G under reconstruction, $t' : i \to j - l$ a partially oriented unshielded triple in G associated with the statistical conditional independence relation $i \perp\!\!\!\perp l \mid Sep_{i;l}$. t' corresponds to a chain $i \to j \to l$ in $\mathcal{M}(G)$ if and only if $j \in Sep_{i;l}$.*

Again corollary 4 assumes causal sufficiency. If latent variables are considered, then for the edge $j - l$ the arrowhead from j to l no longer holds, whereas the arrowtail at j still applies. In that case, we say that j is a non-descendant of l.

Conventionally, the identification of all v-structures marks the end of the step 2 in fig. 2.3. And the above reasoning is part of the "rules of propagation" in step 3. However, it can be argued that the induced arrowtail in corollary 4 is also part of the identification of the unique set of v-structures of $\mathcal{M}(G)$, as it denies the possibility of any additional v-structure. The part of results obtained from corollaries 3 and 4 that does not depend on the causal sufficiency assumption, i.e., the arrowheads in v-structures and the arrowtails in chains, can be seen as the *evidence of causality* (though weak under the PAG framework) in the data, for that they are obtained based only on the results of statistical independence tests. Note, in particular, that the assumption of acyclicity, though always present, is not actually involved in the definition of d-separation.

Finally, the rest of the step 3 consists in orienting as many of the remaining unoriented edges as possible, while making sure that the graph G stays within the same quivalent class. The reasoning often involves contradiction. The PC algorithm, for example, assumes causal sufficiency and acyclicity, and will thus orient an edge $i - j$ as $i \to j$

if $i \leftarrow j$ would induce a new v-structure, or a cycle in G. In particular, it will identify a chain structure as in corollary 4, for a triple t', without the constraint on the related v-structure t, in order to ensure a CPDAG as the final output.

2.3. Limitation of constraint-based methods

Constraint-based methods have been proved long time ago to be sound and complete under ideal circumstances [25, 34, 39], where "ideal" often points to infinite sample size and absence of bias or noise. But its local, progressive, and somewhat non-parametric nature (as compared to score-based methods) can lead to its lack of robustness against bias and noise under finite sample size, that is, its weakness for real life applications.

Algorithm 2 Skeleton reconstruction (step 1 of original PC algorithm)

Require: Conditional independence assessment with significance level α
for all $\{ X_i, X_j \} \subseteq V$

1: $G \leftarrow G_c(V, E)$ ▷ *Complete undirected graph over V*
2: $\ell \leftarrow -1$
3: **repeat**
4: $\quad \ell \leftarrow \ell + 1$
5: $\quad$ **repeat**
6: $\quad\quad$ Select a new pair $(X_i, X_j) \in E$ such that $|\mathrm{adj}_G(X_i) \setminus \{ X_j \}| \geq \ell$
7: $\quad\quad$ **repeat**
8: $\quad\quad\quad$ Select a new set $C \subseteq \mathrm{adj}_G(X_i) \setminus \{ X_j \}$ such that $|C| = \ell$
9: $\quad\quad\quad$ **if** $(X_i \perp\!\!\!\perp_p^\alpha X_j \mid C)$ **then**
10: $\quad\quad\quad\quad E \leftarrow E \setminus (X_i, X_j)$ ▷ *$X_i \nsim X_j$ in G*
11: $\quad\quad\quad\quad \mathrm{Sep}_{X_i;X_j} \leftarrow \mathrm{Sep}_{X_j;X_i} \leftarrow C$
12: $\quad\quad$ **until** $X_i \nsim X_j$ or all such sets C have been considered
13: $\quad$ **until** all such pairs $(X_i, X_j) \in E$ have been considered
14: **until** $|\mathrm{adj}_G(X_i) \setminus \{ X_j \}| \leq \ell$ for all pairs $(X_i, X_j) \in E$
15: **return** $G(V, E)$, $\{ \mathrm{Sep}^G_{X_i;X_j} \}$ for all $X_i \nsim X_j$ in G

To have an idea of the context, let us look at the skeleton reconstruction step of the original PC algorithm, as shown in algorithm 2. Since the major source of constraints is statistical conditional independence tests, the first problem that all constraint-based methods have to deal with is the order of the search for conditional independence, or rather the order by which the tests are performed. The problem is twofold, it concerns firstly the order of pairs (X_i, X_j) to be considered (line 6), and secondly the order of

candidate separating sets to be tested for conditional independence (line 8) for each pair. Note that under ideal circumstances, the ordering of testing is not really a problem since the algorithm is sound, that is, it will give the right CPDAG $\mathcal{M}(G)$ given that the results of all conditional independence tests are correct with respect to $\mathcal{M}(G)$. However, this may not be the case for less-than-ideal situations. The solution of PC regarding the first concern is to give a predefined ordering, for example the lexicographic ordering, of the set of vertices, so that the order of chosen pairs is fixed according to this order. For the second concern, an additional order is imposed which is the increasing cardinality of the candidate separating sets, and the predefined order is used only within each group of sets of the same cardinality (line 8). On the one hand, this ordering of cardinality is in line with the idea of Occam's razor since, as mentioned earlier, $\mathrm{Sep}_{X_i;X_j}$ is not always unique, and candidates of simpler structure are preferred. On the other hand, the ordering is of practical meaning when the sample size is fixed and finite. Each variable added into the separating set increases the number of parameters in the statistical model, and the data is thus sliced into more pieces, meaning that there are less samples available for the estimation of each parameter, and that the test results are more prone to errors. Nonetheless, in [10] it is shown that when there are unfaithfull statistical conditional independence relations, not only may algorithm 2 give a wrong CPDAG $\mathcal{M}(G') \neq \mathcal{M}(G)$, but also the structure of $\mathcal{M}(G')$ may depend on the specific ordering. This dependence is largely due to the dependence of $\mathrm{adj}_G(X_i)$ (line 6, 8) on the results of previous tests. And [10] proposed a modified verson of PC, called PC-stable, to resolve the issue, as illustrated by algorithm 3. The main idea is to remove the problematic dependence simply by fixing $\mathrm{adj}_G(X_i)$ outside of the two inner loops, as highlighted by the dashed box.

But ordering is not the only issue. With erroneous conditional independence tests, another major limitation of constraint-based methods is that the set of d-separations (if assuming causal sufficiency) or m-separations (if assuming presence of latent variables) encoded in the recovered graphical model $\mathcal{M}(G')$ may conflict with the statistical model, and the conflict is directly reflected in the separating set. To shed some light on the problematic situations, we consider the reconstruction of the graph presented in fig. 2.3, with the essential results recaptured in fig. 2.4.

Suppose that we are provided with the following set of correct statistical independence relations (where $\perp\!\!\!\perp$ means $\perp\!\!\!\perp_p^\alpha$): $3 \perp\!\!\!\perp 5$, $1 \perp\!\!\!\perp \{3,4\} \mid \{2,5\}$ and $\{3,6\} \perp\!\!\!\perp 4 \mid \{2,5\}$, and with $2 \perp\!\!\!\perp 6 \mid \{3,5\}$ and $3 \perp\!\!\!\perp 6 \mid 1$ as the set of incorrect statistical independence relations. Following algorithm 3, first the skeleton in fig. 2.4 (marked by solid lines) will be constructed by removing edges from the complete directed graph $G_c(\{1,\cdots,6\})$ in the following order: $3 \not\sim 5$, $3 \not\sim 6$, $1 \not\sim 3$, $1 \not\sim 4$, $2 \not\sim 6$, $3 \not\sim 4$, $4 \not\sim 6$.

Algorithm 3 Skeleton reconstruction (step 1 of PC-stable algorithm)

Require: Conditional independence assessment with significance level α
for all $\{ X_i, X_j \} \subseteq V$

$G \leftarrow G_c(V, E)$ ▷ *Complete undirected graph over V*
$\ell \leftarrow -1$
repeat
  $\ell \leftarrow \ell + 1$
  for all vertices $X_i \in V$ **do**
    $a(X_i) \leftarrow \mathrm{adj}_G(X_i)$
  repeat
    Select a new pair $(X_i, X_j) \in E$ such that $|a(X_i) \setminus \{ X_j \}| \geq \ell$
    repeat
      Select a new set $C \subseteq a(X_i) \setminus \{ X_j \}$ such that $|C| = \ell$
      if $(X_i \perp\!\!\!\perp_p^{\alpha} X_j \mid C)$ **then**
        $E \leftarrow E \setminus (X_i, X_j)$ ▷ $X_i \nsim X_j$ *in G*
        $\mathrm{Sep}_{X_i;X_j} \leftarrow \mathrm{Sep}_{X_j;X_i} \leftarrow C$
    until $X_i \nsim X_j$ or all such sets C have been considered
  until all such pairs $(X_i, X_j) \in E$ have been considered
until $|a(X_i) \setminus \{ X_j \}| \leq \ell$ for all pairs $(X_i, X_j) \in E$
return $G(V, E)$, $\{ \mathrm{Sep}^G_{X_i;X_j} \}$ for all $X_i \nsim X_j$ in G

Figure 2.4. – Two scenarios of inconsistent conditional independence: $2 \perp\!\!\!\perp 6 \mid \{3, 5\}$ regarding the skeleton, where a path between 2 and 6 that does not go through 3 is expected but missing. And $3 \perp\!\!\!\perp 6 \mid 1$ regarding the partially directed graph, where 1 as the common descendant of 3 and 6 is not expected in the separating set.

Next, among the set of unshielded triples: $1 - 2 - 3$, $1 - 2 - 4$, $1 - 5 - 4$, $2 - 1 - 6$, $2 - 5 - 6$, $3 - 2 - 4$, $3 - 2 - 5$ and $4 - 5 - 6$, two v-structures $2 \to 1 \leftarrow 6$ and $3 \to 2 \leftarrow 5$ are identified.

Finally, using rules of propagation, the following edges are oriented: $2 \to 4$ (avoid v-structure), $5 \to 1$ and $5 \to 4$ (avoid cycles). In the reconstructed CPDAG $\mathcal{M}(G)$ in

fig. 2.4 right, the set of minimal (in terms of the cardinality of the separating set) d-separations is ($\perp\!\!\!\perp$ means $\perp\!\!\!\perp_d$): $1 \perp\!\!\!\perp \{3,4\} \mid \{2,5\}$, $2 \perp\!\!\!\perp 6 \mid 5$, $3 \perp\!\!\!\perp \{5,6\}$ and $4 \perp\!\!\!\perp \{3,6\} \mid \{2,5\}$. Compared with the given set of statistical conditional independence relations, there are two conflicts.

The first conflict is between $2 \perp\!\!\!\perp_d 6 \mid 5$ and $2 \perp\!\!\!\perp_p^\alpha 6 \mid \{3,5\}$, as highlighted by the colored vertices and dashed edge in fig. 2.4 left. According to the global Markov property, we would have $2 \perp\!\!\!\perp_p 6 \mid 5$, whereas $2 \perp\!\!\!\perp_p 6 \mid \{3,5\}$ and the searching order of increasing cardinality of separating set in algorithm 3 imply $2 \not\perp\!\!\!\perp_p 6 \mid 5$. Thus we say that the graphical model, or $\mathrm{Sep}_{2;6}$ in particular, is inconsistent with respect to $\mathcal{M}(G)$.

Graphically, this inconsistency can be attributed to the missing path $\gamma^3_{2;6}$ between 2 and 6 that passes through 3. Meanwhile algorithmically, the removal of the edge $2 - 6$ by $2 \perp\!\!\!\perp 6 \mid \{3,5\}$ is well expected, not only because $\{3,5\} \subset \mathrm{adj}(2) \setminus \{6\}$ is a legitimate candidate for $\mathrm{Sep}_{2;6}$, but also because of the fact that when $2 - 6$ is considered for conditional independence test, the edges $3 - 4$, $4 - 6$, hence $\gamma^3_{2;6} : 2 - 3 - 4 - 6$, is still there. Since the inconsistency appears during the reconstruction of skeleton, we call this type of inconsistency the *skeleton-inconsistency*.

The second conflict, highlighted in fig. 2.4 right, is between $3 \perp\!\!\!\perp_d 6$ and $3 \perp\!\!\!\perp_p^\alpha 6 \mid 1$, for that according to $\mathcal{M}(G)$ we must have $3 \not\perp\!\!\!\perp_d 6 \mid 1$. Thus $\mathrm{Sep}_{3;6}$ is inconsistent with respect to $\mathcal{M}(G)$. This conflict appears during the orientation phase, where, after the v-structures are identified, vertex 1 becomes a common descendant of 3 and 6 and d-connects the path $\gamma_{3;6} : 3 \to 2 \to 1 \leftarrow 6$. And we call this type of inconsistency the *orientation-inconsistency*.

Theoretically speaking, when a missing edge $i \nsim j$ has inconsistent $\mathrm{Sep}_{i;j}$ (of either type), the error can come from various possibilities, including

1. the erroneous statistical test $i \perp\!\!\!\perp_p^\alpha j \mid \mathrm{Sep}_{i;j}$ for the pair (i,j) itself,
2. the erroneous statistical tests of some other pairs that are considered *before* (i,j), which eventually excludes the right separating set for $i \nsim j$ from being considered,
3. the erroneous statistical tests of some other pairs that are considered *after* (i,j), which break some existing path $\gamma_{i;j}^{k \in \mathrm{Sep}_{i;j}}$ and thus make $\mathrm{Sep}_{i;j}$ inconsistent with respect to the skeleton, and
4. some incorrectly oriented edges that make $\mathrm{Sep}_{i;j}$ inconsistent with respect to the orientations.

They are all directly or indirectly related to those spurious statistical conditional independence relations, which, as mentioned at the begining of the section, are somehow mitigated by the increasing-cardinality ordering of the search of conditional independence relations. However, given the progressive and tightly interlocking skeleton pruning and

orientation steps, any error that occurs early during the process will likely accumulates and cascades, and eventually have quite an impact on the final reconstructed model.

For the purposes of causal analysis, these inconsistencies can influence the structure learning as well as the causal inference. From a model-selection point of view, score-based methods start with a predetermined class of models, and stay strictly within this model for the definition of score and for the search of best fitting candidate. Contrarily, constraint-based methods, though often also have an expected model class in mind before the reconstruction, are not restricted to a specific class of models, as seen in section 2.2.2. Some constraints imposed by a specific model are not applied until late in the orientation steps (e.g. the acyclicity). As a consequence, in the presence of spurious conditional independence relations, the reconstructed model may conflict with presumptions, as illustrated by the following example.

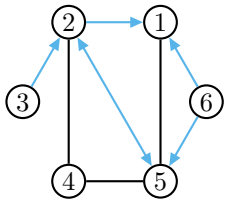

Figure 2.5. – A partially oriented graphical model reconstructed by PC-stable algorithm, based on the set of statistical conditional independence relations: $3 \perp\!\!\!\perp 5$, $2 \perp\!\!\!\perp 6 \mid 3$, $3 \perp\!\!\!\perp 6 \mid 1$, $1 \perp\!\!\!\perp \{3,4\} \mid \{2,5\}$, $\{3,6\} \perp\!\!\!\perp 4 \mid \{2,5\}$. Three v-structures: $2 \to 1 \leftarrow 6$, $2 \to 5 \leftarrow 6$ and $3 \to 2 \leftarrow 5$ are identified.

We take the same model as in fig. 2.4, and alter the assumed results of statistical tests, by changing from $2 \perp\!\!\!\perp 6 \mid \{3,5\}$ to $2 \perp\!\!\!\perp 6 \mid 3$ (thus still an erroneous relation). During the first step, the adjustment does not change the reconstructed skeleton, though the order by which edges are removed does change slightly. However, during the identification of v-structures, the fact that $5 \notin \text{Sep}_{2;6}$ gives rise to an additional v-structure $2 \to 5 \leftarrow 6$ (corollary 3) as compared to fig. 2.4 and shown in fig. 2.5. As a result, before proceeding to the rules of propagation, there is already a bidirected edge $2 \leftrightarrow 5$ in the graph. For methods that take into consideration latent variables, this result is nothing too exceptional, as within the PAG framework, such a bidirected implies one or more latent confounding variables that is the common ancestors of 2 and 5. However, for methods like original PC and PC-stable that assume causal sufficiency, such result goes immediately out of the presumed class of CPDAGs, and greatly reduces the credibility of the successive edge orientations. For example, the arrowheads induced in a chain structure are no longer justified.

To demonstrate the existence of such inconsistencies in applications, we generated a

set of datasets of random and scale-free DAGs of 50 vertices, with increasing sample size and varying parent-child interaction strength. Each dataset is given as input to an implementation of PC-stable algorithm from the R package *pcalg* [2] [20]. The results of reconstructions are summarized in fig. 2.6. For all datasets, both skeleton-inconsistency and orientation-inconsistency are present, which dominate at small sample size, and remain significant at large sample size. When the sample size inscreases, the statistical independence test results are more accurate, whereas when the interaction strength inscreases, the edges in the graph are more robust against sampling noise. As a result, both of them reduce the number of spurious statistical conditional independence relations, which results in a denser final graph and makes it more likely for the separating sets to be consistent.

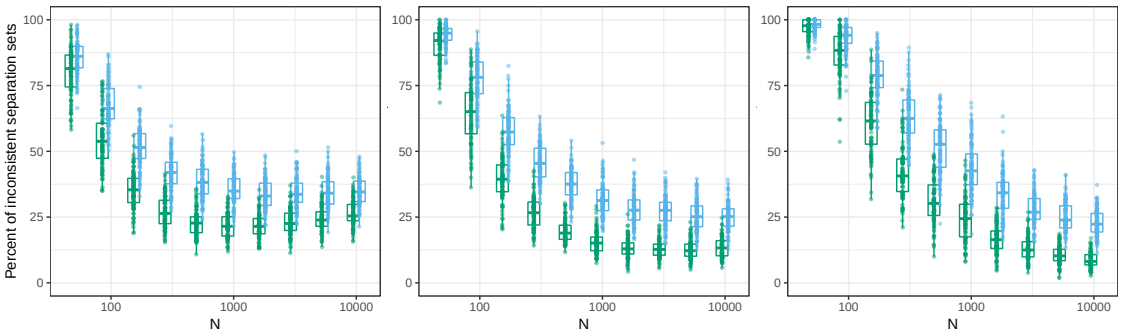

Figure 2.6. – Fractions of skeleton-inconsistent (green) and orientation-inconsistent (difference between blue and green) separating sets in networks reconstructed by PC-stable algorithm with fixed $\alpha = 0.05$. At each sample size, and in each subplot, 100 scale-free networks of 50 vertices are randomly generated with average degree $d(G) = 1.6$, and with different parent-child interaction strengths: strong (left), medium (middle) and weak (right).

Although for inference purposes, once a graphical model is selected, the inference will be conducted based on that model, rather than on the statistical "training set" used to reconstruct the model, making the inconsistencies between statistical model and graphical model less imperative, the overlooked divergence from presumed framework will likely make any following work based upon the model questionable. As shown above, these inconsistencies can often be reflected by the inconsistencies of separating sets with respect to the graphical model. Therefore, we may well expect that by somehow reducing or completely eliminating such inconsistencies of separating set with respect to the reconstructed graphical model, we can control the impact of those false negative edges due to erroneous statistical independence relations.

2. `https://cran.r-project.org/package=pcalg`.

Moreover, the presence of separating set inconsistencies is problematic for constraint-based methods in its own right, as one of the characteristics and motivations of learning and visualizing graphical models is arguably to be able to read off conditional independence relations directly from the graph [25, 34]. For real life applications, we naturally expect the implied conditional independence relations to be in accord with statistical evidences.

2.4. PC algorithm with consistent separating set

Well motivated by section 2.3, we aim to provide a solution for PC or PC-derived algorithms such that in the final reconstructed graph, all separating sets are consistent with respect to either the skeleton or the oriented graph.

2.4.1. Defining separating set consistency

In this section, we give and discuss the definitions concerning each of the two types of inconsistencies.

Definition 5a (Consistent set under causal sufficiency assumption)**.** Let $G(V, E)$ be a graph and $\{ i, j \} \subseteq V$ an ordered pair of vertices, the set of vertices consistent with respect to (i, j) and to the skeleton of G is

$$\text{Conskel}(i, j \mid G) = \{ k \in \text{adj}_G(i) \setminus \{ j \} \mid \text{there is at least one path } \gamma_{i;j}^{k} \text{ in } G \}.$$

The set of vertices consistent with respect to (i, j) and to G is

$$\text{Consist}(i, j \mid G) = \{ k \in \text{Conskel}(i, j \mid G) \mid k \text{ is not a child of } i \text{ in } G \}.$$

Since $\text{Sep}_{i;j}$ is not unique, under the causal sufficiency assumption, the local Markov property eq. (2.3) justifies the search of separating set within the set of adjacent vertices of each vertex. The above definitions then impose additional constraints on top. Note, in particular, that the constraint on the adjacency is only a sufficient condition, and so are the constraints on the consistency.

As an example, look at the following graph $G : i \to k \to l \to m \to j \leftarrow n$. When searching for set C such that $i \perp\!\!\!\perp_d j \mid C$, C can be any non-empty set $A \in \mathcal{P}_{\{k,l,m\}}$ (recall that $\mathcal{P}$ denotes the power set), or any $B = \{ n \} \cup A$. First, the increasing-cardinality searching order of PC algorithm excludes all B from being considered. Then the adjacency constraint excludes l. After that we have $\text{Conskel}(i, j|G) = \{ k \}$, $\text{Conskel}(j, i|G) = \{ m \}$, $\text{Consist}(i, j \mid G) = \emptyset$ and $\text{Consist}(j, i \mid G) = \{ m \}$. Although $\{ k \}$ is also consistent with respect to G, in a more complex graph $\gamma_{i;j}$ is most likely not unique. That k d-separating

one path $\gamma_{i;j}$ does not necessarily prevent it from d-connecting another path $\gamma'_{i;j}$. Plus the fact that finding and checking all paths between a pair of vertices impose an unfavorable algorithmic complexity. The non-child constraint in definition 5a is therefore a sufficient and reasonable condition.

In presence of latent variables, local Markov property, hence the adjacency constraint, no longer holds, as illustrated in G_3 of fig. 2.2. Accordingly, definition 5a needs to be adapted. Ideally, we would like to have the following definition:

Definition 5b (Consistent set without causal sufficiency assumption)**.** Let $G(V, E)$ be a graph and $\{ i, j \} \subseteq V$ a pair of vertices, the set of vertices consistent with respect to (i, j) and to the skeleton of G is

$$\text{Conskel}(i, j \mid G) = \{ k \in V \setminus \{ i, j \} \mid \text{there is at least one path } \gamma^k_{i;j} \text{ in } G \}.$$

The set of vertices consistent with respect to (i, j) and to G is

$$\text{Consist}(i, j \mid G) = \{ k \in \text{Conskel}(i, j \mid G) \mid k \text{ not a common descendant of } i \text{ and } j \text{ in } G \}.$$

Now the definitions are symmetric to i and j. Later we will see that practically, there is not much difference for $\text{Conskel}(i, j \mid G)$. However, for $\text{Consist}(i, j \mid G)$ the check for common descendant is not as trivial as for child. It necessarily involves the traversal of the space of $\{ \gamma_{i;j} \}$, and is algorithmically unfeasible. As a result, if sticking to the non-child constraint, some necessary non-adjacent vertices may be excluded from candidates for separating set, and the search for conditional independence relation for the concerned pair may fail, and eventually lead to a false positive in the reconstructed skeleton, which may in turn affect the orientations steps. Otherwise, we may discard the orientation-consistency constraint, and the final reconstructed graph may contain separating sets that are not consistent with respect to the oriented graph. In any case, the compromise is inevitable.

For now we bring back the causal sufficiency assumption and stick to definition 5a, and see what can those definitions bring to PC-stable algorithm. A natural step forward is to replace $a(X_i) \setminus \{ X_j \}$ by $\text{Consist}(X_i, X_j \mid G)$ at line 6 and 8 of algorithm 3, so as to make sure only consistent candidates will be tested. However, a closer look reveals several issues.

The first issue is the break of order independence. Recall that in the original PC algorithm, the order dependence comes mainly from the use of $\text{adj}_G(X_i)$ where G, and thus $\text{adj}_G(X_i)$, is evolving with the removal of each edge. Concerning consistent separating sets, the determination of the existence of path $\gamma^G_{X_i;X_j}$ is no exception. The

replacement will therefore bring back this dependence, undermining the modification of PC-stable algorithm. To circumvent the order dependence, the same trick can be used as in PC-stable. $a(X_i)$ in algorithm 3 can be understood as a snapshot of the structure of the graph taken at each level of cardinality of separating set. Following the same idea, we can take the snapshot of the whole graph, denoted by G_l, at the same time we create $a(X_i)$. Then $a(X_i) \setminus \{ X_j \}$ will be replaced by $\text{Consist}(X_i, X_j \mid G_l)$, which is now also order independent.

The second issue is more complicated. It is the fact that even with the replacement, be it dependent or independent on order, the separating sets are still not guaranteed to be consistent with respect to the final graph. For one thing, the modification applies only to the skeleton reconstruction phase, therefore only $\text{Conskel}(X_i, X_j \mid G)$ is taken into account. For another, even at the skeleton level, the separating sets are not guaranteed to be consistent. In fact, it is well possible that the replacement will have no effect at all on the choices of conditional independence tests, since at each level l of cardinality of separating set, the considered sets only need to be skeleton-consistent with respect to G_l, but the consistency can be broken by some other tests happening later during the process. Indeed, if looking back at the example of reconstruction used to demonstrate the existence of inconsistency in section 2.3, it turns out that all of the considered sets are subsets of the skeleton-consistent set at the moment of testing. At this point, our reasoning seems to fall into a circle: separating set consistency is required during the reconstruction of the final graph, but the definition of consistency is not possible until the graph is reconstructed.

2.4.2. An iterative approach

Since it seems difficult to build a consistent graph in one go, another way of thinking is to fix the inconsistencies in the reconstructed graph. A trivial graphical fix is to return a complete and undirected graph $G_c(V)$, which is most likely meaningless. Then a reasonable non-graphical way is, if the inconsistency concerns $i \perp\!\!\!\perp j \mid \text{Sep}_{i;j}$, to try to find another $\text{Sep}_{i;j}$ that is consistent with the final graph, though there is no guarantee for the existence of a consistent $\text{Sep}_{i;j}$. A plan B is therefore required.

To fix skeleton-inconsistencies, if the issue concerns a missing path between i and j, we may consider restoring the path by adding back certain removed edge(s). However for a realistic graph with many vertices, there are likely many ways to make appear a path $\gamma_{i;j}$, how can we decide which edge(s) to put back? That is, how do we determine which edges are incorrectly missing? As mentioned earlier, an inconsistency can be due to various sources of spurious conditional independence relations, not necessarily coming from the

concerned pair itself. Do we make the decision randomly (which may still require some effort), or just be "lazy" and assume that the only incorrectly missing edge is $i \nsim j$ itself, and just add $i - j$ back to the graph? The latter may seem like a good and simple choice, except the fact that it is just better than the trivial solution. In other words, we lose information, which are the conditional independence relations contained in data, in exchange for separating set consistency. Similarly for fixing orientation-inconsistencies, if a vertex k in $\text{Sep}_{i;j}$ is inconsistent because of the edge $i \to k$, we can either put back the edge $i - j$, or remove the arrowhead of the edge between i and k. Either way we trade information for consistency.

But keep in mind that the ultimate reason for demanding the consistency is to improve the interpretability of the information. It would be much less meaningful to have consistency if in the end there is little information left to be interpreted. For this reason, we need to adjust our objective from "enforcing separating set consistency", to "enforcing separating set consistency while keeping maximum information in the reconstructed graph". For example, for each edge added back or each removed arrowhead, it may concern not only $\text{Sep}_{i;j}$, but also some other originally inconsistent separating sets, so that these removed edges will not need to be added back to the graph. Moreover, the new edge $i - j$ may now give the possibilities for some new conditional independence relations like $k \perp\!\!\!\perp i \mid j$ where j was previously excluded from candidates for separating set because it was not in any of the adjacency sets of i and k.

As a result, to keep maximum information in the graph, after each fix by adding back edges or by removing arrowhead, we would like to do a check on the whole graph, or more specifically, on all edges in the graph, to see if any of them can be removed by a new conditional independence relation. But then, more problems appear. Practically it is quite some burden on the computation complexity by re-examine all edges frenquently. But more importantly, each of the newly removed edge may at the same time create new inconsistencies. Consequently, for each newly removed edge, a check on all missing edges is needed. And we again fall into a circle, this time of adding and removing edges. But this time, the repeated steps of adding and removing edges may inspire us with an iterative solution.

Instead of doing the checking after each step, we can run the check at the graph level, and reconstruct a whole new graph using the old one as the "instruction manual". More precisely, since we cannot directly build G that is consistent with G itself, we can instead build G_2 that is consistent to G_1. And since G_2 is not necessarily consistent with itself, we repeat the process and build G_3 that is consistent with G_2. By repeating this process, we hope to eventually get a graph G_k that is consistent with itself.

Now we can give the formal description of the above idea.

Definition 6. NewStep1($G_1 \mid G_2$) is a modified version of the Step 1 of PC-stable algorithm where

1. G_c is replaced by G_1, and
2. $a(X_i) \setminus \{ X_j \}$ is replaced by $a(X_i) \cap \text{Consist}(X_i, X_j \mid G_2)$.

The first condition changes the starting point of the algorithm, from a complete graph to a graph G_1 assigned beforehand. The second condition imposes the constraint over separating set consistency with respect to another input graph G_2. As discussed above, G_2 denotes a graph reconstructed at an earlier time that can be (partially) directed, so the non-child restriction will take effect. In particular, NewStep1($G_c|G_c$) recovers algorithm 3 unmodified, and NewStep1($G_c \mid G_\emptyset = (V, E = \emptyset)$) will return an undirected graph in which all removed edges correspond to statistical marginal independence $X_i \perp\!\!\!\perp_p^\alpha X_j \mid \emptyset$. In unmodified algorithm 3, this corresponds to the graph obtained at the end of the $\ell = 0$ outer loop. Since this loop is the only one that does not involve search of separating sets, we call it *Step1a*, to be distinguished from $\ell > 0$ loops that we call *Step1b*. The modified Step 1 allows for the definition of the modified PC-stable algorithm.

Definition 7. $S(G_1 \mid G_2)$ is a modified version of the PC-stable algorithm where Step 1 (algorithm 1) is replaced by NewStep1($G_1 \mid G_2$) (definition 6).

Let $G_3 = S(G_1 \mid G_2)$ be the output graph of the modified PC-stable algorithm, we finally have a mathematical representation of what is discussed earlier: build a graph G_3 that is consistent to G_2. And we are now ready to give a formal description of the iterative idea, as shown in algorithm 4.

For any $G^*(V)$, a call to NewStep1($G_c \mid G^*$) will have G_0 as an intermediate result after the $\ell = 0$ loop, so for computation efficiency, G_0 is placed out of the reconstruction loop and is used as the starting point for all repeated reconstructions. Accordingly, as an implementation detail, in all calls to the S algorithm, the cardinality of separating set ℓ should start from 1 instead of 0. At iteration k, $S_k(G_0 \mid G_{k-1})$ returns a graph with separating sets consistent to G_{k-1}. Ideally, we would like to find a graph that is consistent to itself, in terms of iterations, this corresponds to two successive graphs that are identical to each other, i.e., $G_{m-1} = G_m = S_m(G_0 \mid G_{m-1})$. In practice, however, such result is not guaranteed. Instead, we may find a series of graphs $G_{k-n}, G_{k-n+1}, \cdots, G_k$ such that $G_{k-n} = G_k$, then the set $\{ G_{k-n+1}, \cdots, G_k \}$ is called a *consistent cycle* of size n. Once such a cycle is found, the loop will stop, as all future iterations will be trapped inside the cycle. Finally, the union of all graphs in the consistent cycle will be taken as the final graph, and theorem 8 guarantees that the final graph has separating sets orientation-consistent to itself.

Algorithm 4 PC algorithm with orientation-consistent separating sets

Require: Set of variables $V = \{ X_i \}_{i=1}^{M}$, dataset $D(V)$, significance level α.

Ensure: $G(V, E)$ with orientation-consistent separating sets

1: $G_0 \leftarrow \text{NewStep1}(G_c \mid G_\emptyset)$ ▷ *Step 1a: find marginal independence relations.*
2: $k \leftarrow 0$
3: **repeat**
4: $\quad k \leftarrow k + 1$
5: $\quad G_k \leftarrow S_k(G_0 \mid G_{k-1})$
6: **until** $n \in \mathbb{Z}^+$ is found such that $G_{k-n} = G_k$ ▷ *Cycle* $\{ G_{k-n+1}, \cdots, G_k \}$ *detected.*
7: $G \leftarrow \bigcup_{i=k-n}^{k} G_i$ ▷ *Conflicting orientations discarded.*
8: **for all** $X_i \nsim X_j$ in G **do**
9: $\quad \text{Sep}^{G}_{X_i;X_j} \leftarrow \text{Sep}^{G_k}_{X_i;X_j}$
10: **return** $G(V, E)$, $\{ \text{Sep}^{G}_{X_i;X_j} \}$ for all $X_i \nsim X_j$ in G

Theorem 8. *The separating sets returned by algorithm 4 are orientation-consistent with respect to the final graph G.*

Proof. Firstly, the size of the consistent cycle n is guaranteed to be finite by the deterministic nature of the algorithm, hence by the finite set of graphs $\{ G_i \}$. Then according to definition 5a, if a separating set $\text{Sep}_{i;j}$ is orientation-consistent to G, then it is also orientation-consistent to any graph G' constructed by adding one or more undirected edges to G and/or by removing one or more arrowheads from G. Since the union of n graphs G can be seen as constructed this way from G_{k-1}, the returned separating sets of G_k, which are orientation-consistent to G_{k-1}, is also orientation-consistent to G. □

As mentioned earlier, inconsistencies come from spurious conditional independence relations. Each such relation either, in the better case, leads to an incorrect separating set, without negative impact on the rest of the graph, or, in the worse case, erroneously removes an edge from graph, which then lead to other errors during skeleton reconstruction or during the orientation phases. Thus, some orientation inconsistencies are actually due to errors in skeleton. Consequently, instead of repeating the reconstruction of oriented graph and taking the union, we may try to repeat only the skeleton phase, and orient the graph based on skeleton-consistent separating sets. This idea is summarized in algorithm 5 where for undirected graphs, the condition imposed by $\text{Consist}(X_i, X_j \mid G_{k-1})$ automatically reduces to $\text{Conskel}(X_i, X_j \mid G_{k-1})$, and the proof of consistency of separating sets is similar to that of algorithm 4. Figure 2.7 sketches the applications of both algorithms on the PC algorithm.

Note that in algorithm 5, orientation phases after the loop can bring some orientation-inconsistencies, and the part highlighted by dashed box fixes them simply by adding undirected edges to the final graph, thus making those skeleton-consistent separating sets also orientation-consistent. But as discussed before, this naive fix is against the idea of keeping maximum information in the final graph, and is thus not recommended in practice. After all, there is already algorithm 4 for orientation-consistency. And we will see in section 2.5 that sometimes it is better to have a balance between being informative and being interpretable, than to enforce absolute interpretability.

Algorithm 5 PC algorithm with skeleton-consistent separating sets

Require: Set of variables $V = \{ X_i \}_{i=1}^{M}$, dataset $D(V)$, significance level α.
Ensure: $G(V, E)$ with orientation-consistent separating sets

$G_0 \leftarrow \text{NewStep1}(G_c \mid G_\emptyset)$ ▷ *Step 1a: find marginal independence relations.*
$k \leftarrow 0$
repeat
    $k \leftarrow k + 1$
    $G_k \leftarrow \text{NewStep1}(G_0 \mid G_{k-1})$
until $n \in \mathbb{Z}^+$ is found such that $G_{k-n} = G_k$ ▷ *Cycle* $\{ G_{k-n+1}, \cdots, G_k \}$ *detected.*
$G \leftarrow \bigcup_{i=k-n}^{k} G_i$
for all $X_i \nsim X_j$ in G **do**
    $\text{Sep}^{G}_{X_i;X_j} \leftarrow \text{Sep}^{G_k}_{X_i;X_j}$
Step 2: orientation of v-structures in G
Step 3: propagation of orientations in G
for all $X_i \nsim X_j$ in G **do**
    if $\text{Sep}^{G}_{X_i;X_j} \not\subseteq \text{Consist}(X_i, X_j \mid G) \cup \text{Consist}(X_j, X_i \mid G)$ **then**
        Add $X_i - X_j$ to G
return $G(V, E)$, $\{ \text{Sep}^{G}_{X_i;X_j} \}$ for all $X_i \nsim X_j$ in G

A basic implementation of algorithms 4 and 5 is done based on the implementation of PC-stable algorithm from the R package pcalg [3]. We then take three relatively complex benchmark networks (average degree d between 3.5 and 3.85) from the *BNlearn* repository [31], and compare the performance of original PC-stable (algorithms 1 and 3), orientation-consistent PC-stable (algorithm 4) and skeleton-consistent PC-stable (algorithm 5) algorithms by reconstructing the three networks with $N = 1000$ samples under different significance levels α. The results are gathered and presented in fig. 2.8 in terms of the precision and recall of the adjacencies found in the reconstructed graphs with respect to

3. `https://github.com/honghaoli42/consistent_pcalg`

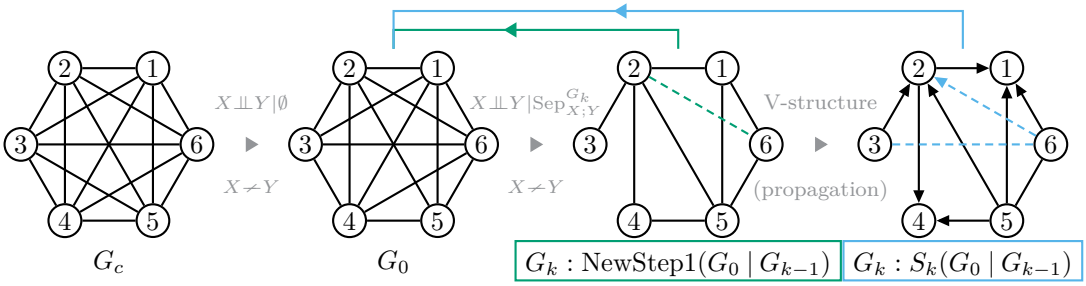

Figure 2.7. – Iterative approach to learn graphical models with orientation-consistent (algorithm 4) or skeleton-consistent (algorithm 5) separating sets. Dashed edges mark the difference between two successive iterations.

the true skeleton. It can be seen that for all three networks, original PC-stable algorithm gives high precision and low recall under all values of α, which is largely due to its tendency to recover spurious conditional independence relations, as discussed earlier. Contrarily, when enforcing the consistency of the separating sets, the reconstructed graphs have a significantly better recall, at the price of lower precision. Under the α that maximizes the F-score (dotted lines mark equal F-score) of each method, the two consistent PC-stable algorithms achieve an equivalent or slightly better F-score, while having a better balance between precision and recall. More importantly, the separating sets are more interpretable in the graphs reconstructed by the two consistent PC-stable algorithms. More benchmarks and details on the data generation can be found in [22].

2.5. Informativeness vs. interpretability

Following the previous section, although we have repeatedly accented the importance of the interpretability of constraint-based methods, the cost of it could be so high that we lose more than what we have gained. From time to time it may be benefitial to trade some of the interpretability for more information.

For example, in algorithm 5, the naive fix for orientation-consistency may add many "meaningless" undirected edges in the final graph. We can choose to discard the fix, and accept possible orientation-inconsistencies, but then there is a minor concern. In algorithm 4, the final oriented graph is the union of one or more oriented graphs, both the skeleton and the orientations have contributions from all graphs in the consistent cycle. On the contrary, if we simply discard the boxed part in algorithm 5, then the skeleton of the final graph is still based on all graphs in the skeleton-consistent cycle, but the orientations are based only on $\{ \mathrm{Sep}^{G_k}_{X_i,X_j;} \}$. Since we already drop the constraint on orientation-consistency, it may be worthwhile to find out what can other graphs in the

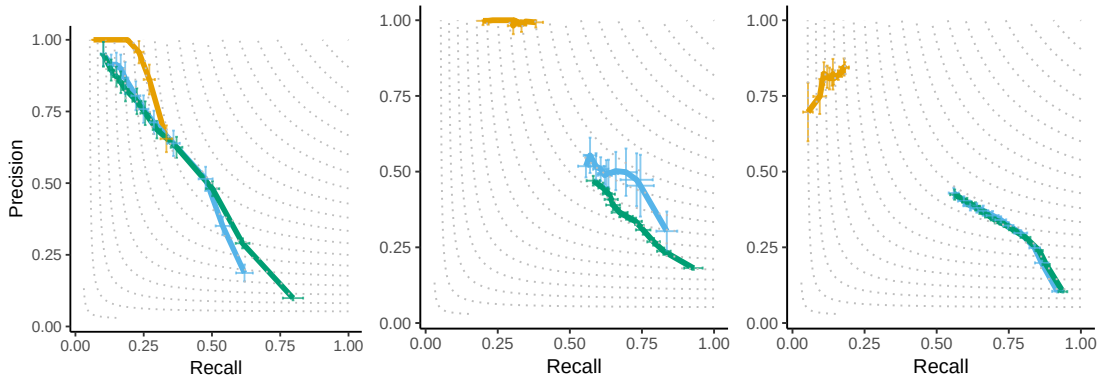

Figure 2.8. – Precision-recall curves for the original PC-stable (yellow), skeleton-consistent PC-stable (green) and orientation-consistent PC-stable (blue). The mean performances and standard deviations (error bars) obtained over 100 networks are shown for 12 values of the (conditional) independence significance threshold α between 10^{-25} and 0.5. Datasets with N=1000 samples were generated for the standard benchmarks Hepar2 (left), Insurance (middle) and Barley (right) networks from the BNlearn repository [31].

consistent cycle bring to the final graph in terms of orientation. This idea amounts to discarding the boxed part and then including Step 2 and Step 3 in the iterations, which is essentially to use algorithm 4, but replace all conditions involving $\text{Consist}(X_i, X_j \mid G_{k-1})$ by $\text{Conskel}(X_i, X_j \mid G_{k-1})$.

Another scenario is when in the consistent cycle in algorithm 4 there are too many graphs. Taking the union of such a cycle can potentially add many undirected edges like in the previous example, on top of that it may also remove many arrowheads, which further reduces the informativeness of the final graph. For example, in a consistent cycle of 10 graphs, for a pair of vertices (i, j), if in 9 out of 10 graphs, the edge is missing $(i \not\sim j)$, and only in one graph the edge is present $(i - j)$, then by taking the union, the edge will be present in the final graph G, even though it is likely that the wrong prediction is given by the single graph, but not by the 9 other graphs. As another example, if in 9 out of 10 graphs, the edge between i and j is $i \to j$, and in the only graph left, the arrowhead is missing because it leads to a orientation-inconsistency with respect to the previous graph. Then in the final union graph, the arrowhead will be discarded, which may actually indicate a correct causal relation.

To avoid such loss of information by the presence of some minority cases, we introduce the idea of *consensus graph*, associated with a *consensus threshold* α_c, as given by definition 9 and illustrated by fig. 2.9.

Definition 9 (Consensus graph)**.** Let $\{ G_1, \cdots, G_n \}$ over V be a consistent cycle of size n provided by algorithm 4, and $\alpha_c \in (0.5, 1]$ a real number. The consensus graph G^c is a partially oriented graph constructed as follows: for each pair (i, j) in V, count the number of presence of each possible edge type ($\nsucc$, —, $\to$ and $\leftarrow$) between i and j among the n graphs. The edge between i and j in G^c will take the type of the hightest count if the hightest count m satisfies $m/n > \alpha_c$, otherwise the edge will be marked as undirected.

At the cost of potential loss of separating set consistency, consensus graph can provide more information as the final graph. A consensus graph with $\alpha_c = 1$ is identical to the union graph returned by algorithm 4.

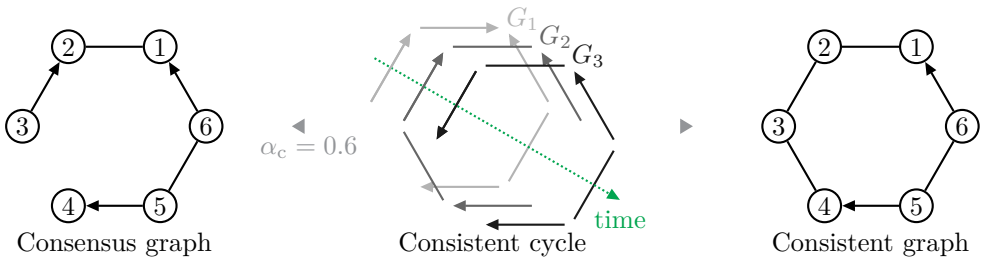

Figure 2.9. – A consistent cycle of size 3 (middle), a orientation-consistent union graph (right) and a consensus graph with $\alpha_c = 0.6$.

2.6. Test of separating set consistency

As seen in definitions 6 and 7 and algorithms 4 and 5, a frenquently requested operation is to check if a vertex k is skeleton/orientation-consistent with respect to a pair (i, j) and a graph G, which requires the computation of Consist$(i, j \mid G)$. Since the check on parent-child relation is trivial, the complexity burden comes down to the check of existence of the path $\gamma^k_{i;j}$.

As mentioned in section 2.4.1, the traversal of the space of $\{ \gamma_{i;j} \}$ is not acceptable in practice as the complexity grows exponentially with the number of vertices and the number of edges in the graph. Here, we provide a solution making use of some results in graph theory. For that, some terminologies and definitions need to be fixed.

Let G be a graph. A *connected* graph is a graph in which there is a path between each pair of its vertices. A *connected component* of G is a maximal connected subgraph of G. An *articulation point* (or *cut point*) of G is a vertex whose removal increases the number of connected components of G. A *biconnected graph* is a connected graph without cut point. A *biconnected component* (or *block*) of G is a maximal biconnected subgraph of G.

Definition 10 (Block-cut tree)**.** Let G be a connected graph. The *block-cut tree* decomposition of G is a graph $\mathcal{T}(B, C, \Sigma)$ where $B = \{ b_i \}_{i=1}^{m}$ is the set of biconnected components (or blocks) of G, $C = \{ c_j \}_{j=1}^{n}$ is the set of cut points of G, $B \cup C$ is the set of vertices of $\mathcal{T}$ and $\Sigma = \{ (b_i, c_j) \mid b_i \in B, c_j \in C \}$ is the set of edges in $\mathcal{T}$.

In a block-cut tree $\mathcal{T}(B, C, \Sigma)$, Σ describes the connections between biconnected components and cut points. By the definition of cut point, each $c \in C$ is adjacent to at least two vertices $b_1, b_2 \in B$. In other words, no $c \in C$ can be a leaf in $\mathcal{T}$. The connection between a vertex in G and a vertex in the block-cut tree $\mathcal{T}$ of G is given by the following definition.

Definition 11 (Block-cut tree representation)**.** Let $G(V, E)$ be a connected graph, $\mathcal{T}(B, C, \Sigma)$ the block-cut tree decomposition of G. $\mathrm{Trep} : V \to B \cup C; i \mapsto \mathrm{Trep}(i)$ is a function such that

1. if i is a cut point in G, then $\mathrm{Trep}(i) = c \in C$ such that $V_c = \{ i \}$ in G,
2. if i is not a cut point in G, then $\mathrm{Trep}(i) = b \in B$ such that $i \in V_b$ in G.

In the following we establish a relation between biconnected components and the path existence problem.

Lemma 12 (Menger's theorem for biconnected graph)**.** *Let $G(V, E)$ be a biconnected graph, $\{ i, j \} \subseteq V$ a pair of vertices. There is a cycle in G that contains i and j.*

Theorem 13. *Let $G(V, E)$ be an undirected graph, $H(V_H, E_H) \subseteq G$ a biconnected component of G, $\{ i, j \} \subseteq V_H$ a pair of vertices, and $k \in V_G$ a third vertex. There is a path $\gamma_{i;j}^{k}$ if and only if $k \in V_H$.*

Proof. If there is a path $\gamma_{i;j}^{k}$, suppose that $k \notin V_H$, then the subgraph H' of G over $V_H \cup \{ k \}$ is biconnected thanks to $\gamma_{i;j}^{k}$, and $H \subset H'$ is not a biconnected component of G as it is not maximal. Therefore we must have $k \in V_H$.

If $\{ i, j, k \} \subseteq V_H$, then lemma 12 guarantees a cycle that contains k and j. Since V_H contains at least three vertices, such a cycle contains $n \geq 1$ vertices other than k and j, and can be represented by two edge-distinct paths between k and j:

$$\gamma_{k;j}^{(1)} = k u_1 u_2 \cdots u_l j, \quad \gamma_{k;j}^{(2)} = k u_{l+1} u_{l+2} \cdots u_n j$$

where $l \in \mathbb{Z}^{\geq 0}$ (with $l = 0$ indicating a direct edge between k and j), $n \in \mathbb{Z}^{+}, l < n$ and $\{ u_i \}_{i=1}^{n}$ are distinct vertices. Since j is not an articulation point, there is a path $\gamma_{i;k}$ that does not contain j:

$$\gamma_{i;k} = i d_1 d_2 \cdots d_m k$$

where $m \in \mathbb{Z}^{\geq 0}$ and $\{ d_j \}_{j=1}^m$ are distinct vertices. If $\{ u_i \}_{i=1}^n \cap \{ d_j \}_{j=1}^m = \emptyset$, then there is a path

$$\gamma_{i;j}^k = \gamma_{i;k}\gamma_{k;j}^{(i)}, i \in \{ 1, 2 \}.$$

Otherwise, suppose $\{ u_i \}_{i=1}^n \cap \{ d_j \}_{j=1}^m = \{ d_{p_1}, d_{p_2}, \ldots, d_{p_t} \}$ where $t \in \mathbb{Z}^+$ and $p_1 < p_2 < \cdots < p_t$, and suppose $d_{p_1} = u_q$. If $q \leq l$, then there is a path

$$\gamma_{i;j}^k = id_1d_2 \cdots d_{p_1}(u_q)u_{q-1} \cdots u_1\gamma_{k;j}^{(2)},$$

if $q > l$, then there is a path

$$\gamma_{i;j}^k = id_1d_2 \cdots d_{p_1}(u_q)u_{q-1} \cdots u_{l+1}\gamma_{k;j}^{(1)}.$$

As a result, if $\{ i, j, k \} \subseteq V_H$, then there is always a path $\gamma_{i;j}^k$. □

Corollary 14. *Let $G(V, E)$ be a connected graph, $\mathcal{T}(B, C, \Sigma)$ the block-cut tree decomposition of G, $\{ i, j \} \subseteq V$ a pair of vertices, and $S(i, j) = \{ k \in V \setminus \{ i, j \} \mid$ there is at least one path $\gamma_{i;j}^k$ in $G \}$.*

1. *If Trep(i) = Trep(j) = $b \in B$, then $S(i, j) = V_b \setminus \{ i, j \}$.*
2. *If Trep$(i) \neq$ Trep(j), let $\nu = w_1 \cdots w_m$ be the path between Trep(i) and Trep(j) in $\mathcal{T}$ such that $w_1 =$ Trep(i) and $w_m =$ Trep(j), then $S(i, j) = (\bigcup_{i=1}^m V_{w_i}) \setminus \{ i, j \}$.*

Proof. Case 1. follows directly theorem 13. Without loss of generality[4], suppose $\nu = b_1c_1b_2c_2 \cdots b_xc_xb_{x+1}$ where $b_i \in B$, $c_i \in C$ and $x = (m-1)/2$. For clarity, in the following, we use c_i to also denote the only element in V_{c_i} of G. By definition of cut point, for all $i \in b_1, j \in b_{x+1}$, the path $\gamma_{i;j}$ in G must pass through all cut points $\{ c_i \}_{i=1}^x$. Each path can thus be expressed by edge-distinct segments

$$\gamma_{i;j} = \gamma_{i;c_1}\gamma_{c_1;c_2} \cdots \gamma_{c_x;j}. \tag{2.7}$$

Since each cut point belongs to all biconnected components it connects to, in G we have $c_i \in b_i$ and $c_i \in b_{i+1}$ for all $i \in \{ 1, \cdots, x \}$. Define $S'(i, j) = S(i, j) \cup \{ i, j \}$, then for each segment, according to case 1., $S'(i, c_1) = V_{b_1}$, $S'(c_x, j) = V_{b_{x+1}}$ and $S'(c_i, c_{i+1}) = V_{b_{i+1}}$ for all $i \in \{ 1, \cdots, x-1 \}$. For all $k \in Q = S'(i, c_1) \cup S'(c_x, j) \cup (\bigcup_{i=1}^{x-1} S'(c_i, c_{i+1})) \setminus \{ i, j \}$, there is a path $\gamma_{i;j}^k$ in G. Then by the uniqueness of ν we have

$$S(i, j) = Q = (\bigcup_{i=1}^{x+1} V_{b_i}) \setminus \{ i, j \} = (\bigcup_{i=1}^{m} V_{w_i}) \setminus \{ i, j \} \tag{2.8}$$

4. Indeed, if $\nu = c_0b_1 \cdots b_{x+1}c_{x+1}$, then c_0 and c_{x+1} can be removed without changing the conclusion, because $V_{c_0} \subset V_{b_1}$ and $V_{c_{x+1}} \subset V_{b_{x+1}}$ in G (a cut point belongs to all adjacent biconnected components).

which completes the proof. □

Each graph G can be decomposed into a set of connected components, and each connected component can be represented by a block-cut tree. Finally, based on corollary 14 and on an algorithm TreePath(i, j, G) that returns the unique path between two vertices in a tree G, algorithm 6 provides a solution for the search of candidates for consistent separating sets.

Algorithm 6 Search of consistent candidates

Require: (Partially directed) graph $G(V, E)$, set of block-cut tree decompositions $\{ \mathcal{T}_t(B_t, C_t, \Sigma_t) \}$ with respect to the skeleton of G, two vertices $\{ i, j \} \subseteq V$

Ensure: Consist$(i, j \mid G)$.

```
if i and j do not belong to the same T_t then
    return ∅
if Trep(i) ∩ Trep(j) = b ∈ B_t then
    return (adj_G(i) \ Child_G(i)) ∩ (V_b \ { i, j })
else
    ν ← TreePath(Trep(i), Trep(j), T_t) = w_1 w_2 ··· w_m
    return (adj_G(i) \ Child_G(i)) ∩ ((⋃_{i=1}^{m} V_{w_i}) \ { i, j })
```

For an undirected graph $G(V, E)$, the block-cut tree decomposition can be done simply with a single depth first search with complexity $\mathcal{O}(|V| + |E|)$ and is required only once per iteration. Then, suppose the block-cut tree decomposition of G is $\mathcal{T}(B, C, \Sigma)$, for each pair $\{ i, j \} \subseteq V$, the complexity of computing Consist$(i, j \mid G)$ depends on the relative position of i and j in G, and on the structure of $\mathcal{T}$. In the best scenario, $\text{Trep}(i) = \text{Trep}(j) = b \in B$, the complexity falls all to the operation of set intersection, which is of complexity $\mathcal{O}(|\text{adj}_G(i)| + |V_b|)$. In the worst scenario, G is itself a tree, and a call to TreePath of complexity $\mathcal{O}(|B| + |C| + |\Sigma|) \sim \mathcal{O}(|V| + |E|)$ is added on top of the complexity of set intersection.

2.7. Discussion

As mentioned in section 2.3, a separating set inconsistency of a pair of variables can come from many sources other than the error related to the pair itself. Without additional tests, there is no easy way to tell how should the inconsistency be fixed. Indeed, some methods try to make use of additional statistical independence tests to deal with some of the inconsistencies, notably the *conservative PC* [26] algorithm, and the *Boolean satisfiability* (SAT) based structure learning methods [19].

The conservative PC algorithm aims at the orientation steps of PC algorithm. For each unshielded triple $i - j - k$ in G, additional statistical conditional independence tests are conducted with candidate separating sets chosen from all subsets of $\text{adj}_G(()i)$ and of $\text{adj}_G(()k)$, regardless of the cardinality of the sets. A v-structure $i \to j \leftarrow k$ can be established only if j is not in any of such sets that successfully separate i and k. Similarly, the remaining unshielded triples can be considered for propagation of orientations only if j is in all such sets that successfully separates i and k. Unshielded triples that do not meet any of the two requirements are considered as ambiguous and left unoriented. Thus, conservative PC generally gives fewer orientations in the final graph as compared to the original PC.

These conservative rules are conceptually solid and indeed improve the accuracy of orientations of PC algorithm, but sometimes the rules are considered too conservative. Therefore [10] proposed a slight modification of the conservative rules, called the *majority rule PC* algorithm, by replacing the "any" and "all" conditions with a threshold of 50%, in order to have more orientations in the final graph. The relation between majority rule PC and conservative PC is much like the relation between consensus graph and consistent graph provided by the separating set consistent PC algorithms proposed in this chapter. However, there is a major difference between conservative PC and consistent PC, which is the fact that conservative PC only concerns the orientation steps, whereas consistent PC deals also with skeleton level inconsistencies. In fact, many orientation inconsistencies come from errors in the skeleton, thus a skeleton with consistent separating set may actually help avoid some of the orientation inconsistencies.

Compared with conservative PC, the SAT based methods try to resolve conflicts at a global level, by exhausting all available information, not only on the results of statistical conditional independence tests, but also on background knowledge and interventional data, with all constraints encoded into the Boolean satisfiability framework. As a result, SAT based methods also concern the consistency of separating sets at skeleton level. However, a major drawback of such methods is their high computational complexity, largely due to the exhausting of all possible constraints. By comparison, neither the iterative approach nor the search of consistent candidates with biconnected components decomposition imposes significant additional computational burden on the PC algorithm.

In brief, the idea of having consistent separating sets for constraint-based methods is nothing much novel. In terms of implementation, the idea of using biconnected components to narrow down candidates for separating sets was already briefly mentioned in [11] when extending the FCI algorithm to the RFCI algorithm, both dealing with latent variables. However, the fundamental characteristic that distinguishes the algorithms proposed in this chapter from other methods that aim to improve constraint-based causal

structure learning is its iterative nature. This idea of repeated reconstruction applies at a rather general level, without much concern about the details of the methods it is applied to. Simply put, all it asks for the method is to have a result that respects the constraints based on which the result is obtained, and it is done by having the method correct its own errors. In the language of artificial intelligence, this can be related to the idea of self-correcting learning [35], where the errors produced by the algorithm are fed back into the network for training future predictions.

We close this chapter by a discussion on the PC* algorithm, originally proposed in [34, section 5.4.2.3], which the author of this thesis came across after finishing the majority of the work presented in this chapter, and in which the idea of confining the search of candidates for separating sets to the set of vertices lying on a undirected path between the concerned pair of vertices was already brought up:

> [...] For a distribution faithful to a directed acyclic graph, [...] It is sufficient, then, to test for the conditional independence of A and B given subsets of variables adjacent to A and subsets of variables adjacent to B that are on undirected paths between A and B. Call the modified algorithm PC*.

But then two concerns were given on the performance of PC*. The first one concerns the correctness of PC*:

> [...] The PC* algorithm avoids one kind of error made by the PC algorithm. If, however, at an early stage the PC* algorithm mistakenly disconnects a path between X and Y it may then mistakenly leave the $X - Y$ edge in the undirected graph, [...]

Then follows the second concern about the complexity of the algorithm:

> [...] Moreover, whatever increased reliability the PC* algorithm may have is bought at great cost, since the algorithm must at each stage of step B) keep track of all of the undirected paths in the graph it considers at that stage. The number of undirected paths is typically very large, and the memory requirements of the PC* algorithm are not feasible save for relatively small numbers of variables, [...]

Indeed, the first issue is not easy to deal with for conventional approaches following a flat reconstruction procedure. Instead, the iterative approach in section 2.4.2 corrects in each iteration the errors of graph from the previous iteration, thus making the timing of the mistake within each iteration less relevant. Then the second issue is completely avoided with the use of biconnected components decomposition in section 2.6. Instead of keeping track of paths, we get directly, and in linear time, all candidates that meet the requirement.

3. The Interpretable MIIC algorithm

In the previous chapter, we discussed extensively the characteristics and limitations of constraint-based methods for causal structure learning. We proposed several algorithms and techniques to help improve the interpretability of the reconstructed graphs, and compared the proposed algorithms with some state-of-the-art methods. Across all sections, the focus of the discussion was on the choices and orders of application of given constraints so that they better comply with the presumed assumptions and restrictions.

In this chapter, we adopt a different view and put the focus on improving the constraints, notably the results of statistical conditional independence tests, themselves. After all, they are the principal source of errors, whose impact often gets amplified by the progressive nature of constraint-based methods.

We start by a discussion on the statistical conditional independence test based on the concept of *conditional mutual information*. Then we review some results of the *maximum likelihood* framework, and suggest a modified framework that enables local comparison between different graphical models, and that connects to the conditional mutual information. Based on those results we describe the *Multivariate Information based Inductive Causation* algorithm (MIIC algorithm, [1, 2, 7, 37]). Then we provide some recent modifications and improvements that emphasize the aspects of interpretability and scalability of MIIC, from a theoretical as well as technical point of view. Finally, we provide some details on the refactoring of the code base of MIIC, followed by some comments.

3.1. Conditional mutual information

For conditional independence test, there are various methods depending on the nature of the test targets. For example, if the model can be assumed as or approximated to multinomial, then chi-square test or G-test can be used. For linear Gaussian models, partial correlation can be used. These methods have constraints on the underlying statistical model of the tested variables. For more general cases, more sophisticated approaches are proposed, such as kernel-based conditional independence test [40] and several methods from the Invariant Causal Prediction (ICP) framework [16].

Here we are interested in another framework based on conditional mutual information, in which no prior assumption on the variables' distributions is needed in principle. For the sake of simplicity, assume that X and Y are two discrete random variables with sample space Ω_X and Ω_Y, and $P(X,Y)$ the joint probability distributions, the *mutual information* between X and Y is

$$I(X;Y) = \sum_{x\in\Omega_X}\sum_{y\in\Omega_Y} \Pr(x,y)\log\frac{\Pr(x,y)}{\Pr(x)\Pr(y)}. \tag{3.1}$$

$I(X;Y)$ is non-negative. In particular, we have as the condition for minimum

$$I(X;Y) = 0 \iff X \perp\!\!\!\perp_p Y, \tag{3.2}$$

which means that X and Y have zero mutual information if and only if they are statistically independent. This makes estimating mutual information a good test for pairwise statistical independence. However, as seen in the previous chapter, for inferring causal networks with constraint-based methods, statistical conditional independence test conditioning on a third variable Z is often required. It is then a natural attempt to consider the conditional mutual information, defined as

$$I(X;Y \mid Z) = \mathbb{E}_Z[I(X;Y) \mid Z] = \sum_{x\in\Omega_X}\sum_{y\in\Omega_Y}\sum_{z\in\Omega_Z} \Pr(x,y,z)\log\frac{\Pr(x,y \mid z)}{\Pr(x \mid z)\Pr(y \mid z)}, \tag{3.3}$$

which is also non-negative. In eq. (3.3) Z can also be a set of variables, in which case the conditioning takes place in the joint space of all variables in Z. Since the definition of $X \perp\!\!\!\perp_p Y \mid Z$ is $\Pr(x,y \mid z) = \Pr(x \mid z)\Pr(y \mid z)$ for all $x \in \Omega_X, y \in \Omega_Y, z \in \Omega_Z$, we immediately have

$$X \perp\!\!\!\perp_p Y \mid Z \implies I(X;Y \mid Z) = 0. \tag{3.4}$$

Equation (3.4) is not enough. For $I(X;Y \mid Z)$ to be used as an estimator, we need rather the opposite implication $\Longleftarrow$. Unfortunately, this is not the case. In particular, when X is statistically nearly identical to Z, then $I(X;Y \mid Z) = I(Y;X,Z) - I(Y;Z)$ will be close to zero regardless of the relation between X and Y. Nonetheless, in the following we will assume no presence of such "ill" case, and thus use conditional mutual information as an estimator for statistical conditional independence relation when reconstructing graphical model. On the one hand, such high similarity pairs of variables can be fairly easily detected using metrics like correlation, on the other hand, the presence of such similar variables in the dataset may prompt us to think about whether it is meaningful to include both variables in the model. Moreover, as we will see later (section 3.4.2),

even if such case indeed happens, the method we propose will effectively exclude Z from being considered in the candidate separating sets of any pair involving X.

Ideally, equipped with the estimator of conditional mutual information, it is straightforward to implement constraint-based methods like PC or PC-stable algorithms. However, in practice, the proper estimation of (conditional) mutual information is quite challenging. Methods for estimating MI and CMI is a whole subject in statistics research, and is beyond the scope of the thesis. Instead, specifically for the purpose of causal structure learning, we focus here on the order of candidate sets C by which we conduct the conditional mutual information based statistical independence tests. As we have seen in the previous chapter, in PC and PC-stable algorithms, one of such orders is the increasing-cardinality of the candidate sets.

Before that, let's first make a bit of detour to see some interesting results related to the maximum likelihood estimation, like for a search-and-score method (section 2.2.1).

3.2. Quasi maximum likelihood estimation

Within the framework of Bayesian networks, let $X = \{ X_i \}_{i=1}^{M}$ be a set of discrete random variables associated with a graphical model, then the model can be represented by a DAG G_0 that encodes a set of d-separations S_0. Equivalently, it can be represented by a probability distribution $P(X \,|\, \theta_0)$ whose probability density function follows a certain factorization compatible with S_0 as in eq. (2.5), and the set of parameters θ_0 associated with S_0 is defined by

$$\theta_0 = \{ \theta_{ijk} \coloneqq \Pr(X_i = j \mid \mathrm{PA}(X_i) = k) \}, \tag{3.5}$$

where $\mathrm{PA}(X_i)$ is the joint variable of all parents of X_i, and the sample space of $\mathrm{PA}(X_i)$ is the joint sample space of all parents of X_i.

Let $D = \{ x^l \in \Omega_X \}_{l=1}^{N}$ be a dataset of N observations independently and identically distributed (i.i.d.) as $P(X)$, G a random DAG proposed for D that encodes the factorization S and a probability distribution $Q(X; \theta)$ with parameter set θ. In the classical scenario of maximum likelihood estimation, we would assume that $S = S_0$, that is, we already know the structure of the underlying DAG. Then the *log-likelihood* function is given by

$$\log \mathcal{L}(D, G, \theta) = \log \Pr(D \mid G; \theta) \overset{\text{i.i.d.}}{=} \log \prod_{l=1}^{N} \Pr(x^l \mid G; \theta) = \sum_{l=1}^{N} \log \Pr(x^l \mid G; \theta). \tag{3.6}$$

Since $Q(X)$ factorizes the same way as $P(X)$, θ and θ_0 have the same structure. Then the maximum likelihood estimation amounts to find the set of parameters that maximizes the log-likelihood:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_{l=1}^{N} \log \Pr(x^l \mid G; \theta). \tag{3.7}$$

And it can be shown that the maximum likelihood estimator is given by

$$\hat{\theta}_{ijk} = \frac{\operatorname{count}_D(X_i = j, \operatorname{PA}(X_i) = k)}{\operatorname{count}_D(\operatorname{PA}(X_i) = k)}. \tag{3.8}$$

Now if the underlying graphical model is unknown, we can no longer assume the same factorization for G_0 and G. And the log-likelihood function becomes a *quasi log-likelihood* function [24, 38]. Despite the fact that the true model $P(X)$ may be out of reach for G, it can be shown that within the model family $Q(\theta)$, the quasi maximum likelihood estimator $\hat{\theta}$ minimizes the *Kullback-Leibler* distance $D_{\mathrm{KL}}(P \parallel Q)$ as in the following relation using the weak law of large numbers:

$$\begin{aligned} \log \mathcal{L}(D, G, \theta) &= \sum_{l=1}^{N} \log \Pr(x^l \mid G; \theta) = N \sum_{x \in \Omega_X} \Pr(x \mid G_0; \theta_0) \log \Pr(x \mid G; \theta) \\ &= -NH(P; Q) = -N[H(P) + D_{\mathrm{KL}}(P \parallel Q)]. \end{aligned} \tag{3.9}$$

In the above equation, $H(P; Q)$ denotes the cross entropy. Under the previous assumption where P and Q are in the same family of models, $D_{\mathrm{KL}}(P \parallel Q(\hat{\theta})) = 0$, which means that $\hat{\theta}$ uniquely maximizes the log-likelihood. Now in the absence of the assumption, it is generally unknown which model family $\mathcal{M}(G)$ will maximize the quasi log-likelihood. The problem thus goes from parameter estimation to model selection, which is exactly the subject of score-based methods (section 2.2.1). Within the maximum likelihood framework, for each considered model with maximized parameter set $\hat{\theta}$, its corresponding log-likelihood is given by

$$\begin{aligned} \log \mathcal{L}(D, G, \theta) &= \sum_{l=1}^{N} \log \Pr(x^l \mid G; \theta) = N \sum_{x \in \Omega_X} \Pr(x \mid G; \hat{\theta}) \log \Pr(x \mid G; \hat{\theta}) \\ &= -NH(\hat{Q}(\hat{\theta})), \end{aligned} \tag{3.10}$$

where $\hat{Q}$ denotes the empirical distribution such that $\hat{Q}(X_i = j \mid \operatorname{PA}(X_i) = k) = \hat{\theta}_{ijk}$. Then the idea is to choose the model that maximizes this parameter-maximized log-likelihood. It can be shown later (eq. (3.26) in section 3.4.1) that this maximization ultimately leads to a complete DAG, thus in practice, some model selection criteria are

proposed. They can be generally considered as appending a penalty (or *information criteria*, IC) term to the log-likelihood function, so as to control the complexity of the model family and thus to avoid over-fitting [3, 14, 15, 27].

3.3. Uniform likelihood comparison framework

Though having the problem of over-fitting, eq. (3.10) is a good tool to compare different model families. In general, this comparison is conducted at the level of the whole graph, as different families have different factorizations, hence different $\hat{\theta}$s. When two graphs are different only at a local scale, the factorization allows for comparison of only the parts of likelihood that is relevant to the local structure, which is essentially the idea of score-based methods.

The key idea of the section, and also of the method that is to be presented later in this chapter, is the following idea of *uniform likelihood comparison*: given two models G_1 and G_2 from different families (represented by different CPDAGs), instead of comparing the two models directly using eq. (3.10), we can choose a third model G_c as the reference model, and compare the log-likelihood of G_1 and G_2 with respect to G_c. The role of G_c is analogous to the role of a reference frame in physics when talking about relativity.

More specifically, suppose the maximized parameter set of G_c is $\hat{\theta}_c$, then we are interested in the log-likelihood of G_1 and G_2 when replacing the maximized parameter set of each model by that of the reference model $\hat{\theta}_c$, that is,

$$\log \mathcal{L}(D, G_1, \hat{\theta}_c) = -NH(\hat{Q}_1(\hat{\theta}_c)), \tag{3.11}$$

$$\log \mathcal{L}(D, G_2, \hat{\theta}_c) = -NH(\hat{Q}_2(\hat{\theta}_c)). \tag{3.12}$$

The motivation for the above substitution is as follows. When comparing two models from similar families that only differ in a local range, for example, in G_2 the variables X_i has one more parent X_j as compared to in G_1, the factorization (eq. (2.5)) of the two models will only differ in the term $p(X_i \,|\, \mathrm{PA}(X_i))$. Using the same parameter set will then allow us to measure this difference within a single estimation, instead of comparing two independent estimations based on each model. And later we will see that this eventually connects to the idea of conditional mutual information.

In principal, the reference model G_c can be any (CP)DAG, including G_1 and G_2. However, a randomly chosen reference has two major defects. The first one concerns the compatibility. The reference parameter set $\hat{\theta}_c$ may not be compatible with the models of interest. For example, there can be some parameters $\hat{\theta}_{ijk}$ that are simply missing in $\hat{\theta}_c$. Then the second one concerns the extensibility. Even if the parameter set is compatible

with the current models of interest, it may not be so when new models are taken into consideration.

Fortunately, there is one unique model that does not have any of the above issue, which is the model of complete DAG. Since there is no v-structure in a complete DAG, according to theorem 1, all complete DAGs over the same set of variables belong to the same family. Since the graph is complete, the model has the largest number of parameters as compared to any other model, and is compatible with any other model under marginalization. Intuitively, as mentioned above, $(G_c, \hat{\theta}_c)$ maximizes the maximized log-likelihood in eq. (3.10) among all possible models, thus it is the most complex one, and is supposed to be compatible with all simpler models and can fit any given dataset over the same set of variables. Another more practical advantage of G_c is that the maximized parameter set $\hat{\theta}_c$ simply takes the form of empirical joint probability,

$$\hat{\theta}_{cl} \coloneqq P(X = x_l) = \frac{\text{count}_D(X = x_l)}{N} \quad \text{for all } x_l \in \Omega_X, \tag{3.13}$$

all conditional probabilities can then be expressed in terms of ratio of marginal probabilities.

Now let us look at some interesting examples of the uniform likelihood comparison framework.

3.3.1. An isolated unshielded triple

Let $\{ X, Y, Z \}$ be a set of random variables, G_1 a v-structure $X \to Z \leftarrow Y$, G_2 a fork $X \leftarrow Z \to Y$ and D a dataset. Under the maximized parameter set $\hat{\theta}_c$ of G_c, the log-likelihoods of G_1 is

$$\begin{aligned} \log \mathcal{L}_{\text{v}}(\hat{\theta}_c) &= \log \mathcal{L}(D, G_1, \hat{\theta}_c) = -N H(\hat{Q}_1(\hat{\theta}_c)) \\ &= -N[H_{\hat{\theta}_c}(Z \mid X, Y) + H_{\hat{\theta}_c}(X) + H_{\hat{\theta}_c}(Y)] \\ &= -N[H_{\hat{\theta}_c}(X, Y, Z) + I_{\hat{\theta}_c}(X; Y)] \end{aligned} \tag{3.14}$$

where $I_{\hat{\theta}_c}(X;Y) = H_{\hat{\theta}_c}(X) + H_{\hat{\theta}_c}(Y) - H_{\hat{\theta}_c}(X, Y)$ is the mutual information estimated under the parameter set $\hat{\theta}_c$. The log-likelihoods of G_2 is

$$\begin{aligned} \log \mathcal{L}_{\text{nv}}(\hat{\theta}_c) &= \log \mathcal{L}(D, G_2, \hat{\theta}_c) = -N H(\hat{Q}_2(\hat{\theta}_c)) \\ &= -N[H_{\hat{\theta}_c}(X \mid Z) + H_{\hat{\theta}_c}(Y \mid Z) + H_{\hat{\theta}_c}(Z)] \\ &= -N[H_{\hat{\theta}_c}(X, Y, Z) + I_{\hat{\theta}_c}(X; Y \mid Z)] \end{aligned} \tag{3.15}$$

where $I_{\hat{\theta}_c}(X;Y \mid Z) = H_{\hat{\theta}_c}(X \mid Z) + H_{\hat{\theta}_c}(Y \mid Z) - H_{\hat{\theta}_c}(X,Y \mid Z)$ is the conditional mutual information estimated under the parameter set $\hat{\theta}_c$. It can be shown that the other two possible non-v-structures $X \to Z \to Y, X \leftarrow Z \leftarrow Y$ lead to the same log-likelihood as eq. (3.15), which is expected as they belong to the same model family.

Now consider the simple model selection problem over the unshielded triple $X - Y - Z$, and we would like to compare the two models G_1 and G_2 as given above. First, let us look at the following log ratio of likelihoods:

$$\log \frac{\mathcal{L}_{\mathrm{v}}}{\mathcal{L}_{\mathrm{nv}}} = -N[I_{\hat{\theta}_c}(X;Y) - I_{\hat{\theta}_c}(X;Y \mid Z)] = -N I_{\hat{\theta}_c}(X;Y;Z), \tag{3.16}$$

where $I_{\hat{\theta}_c}(X;Y;Z) \coloneqq I_{\hat{\theta}_c}(X;Y) - I_{\hat{\theta}_c}(X;Y \mid Z)$ is the *3-point information* (or *interaction information*). Note that eq. (3.16) is only possible under the uniform likelihood comparison framework, otherwise we generally have $H_{\hat{\theta}_1}(X,Y,Z) \neq H_{\hat{\theta}_2}(X,Y,Z)$.

3-point information has some interesting properties. Unlike mutual information, it is not non-negative [27]. Therefore, eq. (3.16) provides a simple criterion for choosing between G_1 and G_2 (assuming that they are the only candidates): if $I_{\hat{\theta}_c}(X;Y;Z) < 0$, then $\mathcal{L}_{\mathrm{v}} > \mathcal{L}_{\mathrm{nv}}$, interpreted as "a v-structure is more likely than a non-v-structure", and we choose G_1; if $I_{\hat{\theta}_c}(X;Y;Z) > 0$, then $\mathcal{L}_{\mathrm{nv}} > \mathcal{L}_{\mathrm{v}}$, and we choose G_2; otherwise the two models are equally likely, thus we cannot decide.

To have an idea of how this works, consider the ideal situation where G_1 is the true model. Since $\hat{\theta}_c$ estimates the joint probability, when the sample size is large enough, we have $\hat{\theta}_c \to \hat{\theta}_1 \to \theta_1$, thus $I_{\hat{\theta}_c}(X;Y) = I_{\theta_1}(X;Y) = 0$. Meanwhile, we have $I_{\hat{\theta}_c}(X;Y \mid Z) = I_{\theta_1}(X;Y \mid Z) > 0$, because in G_1, Z d-connects X and Y. As a result, $I_{\hat{\theta}_c}(X;Y;Z) < 0$. Similar reasoning applies when assuming G_2 as the true model.

In the following, except when specified explicitly, we will always assume the parameter set $\hat{\theta}_c$, and omit the notation.

Another property of 3-point information is its invariance under permutation, which is clear according to the following decomposition:

$$\begin{aligned} I(X;Y;Z) = {} & H(X) + H(Y) + H(Z) \\ & - H(X,Y) - H(Y,Z) - H(Z,X) \\ & + H(X,Y,Z). \end{aligned} \tag{3.17}$$

This implies that during eq. (3.16), the information over $X \nsim Y$ is lost. Intuitively, it can be understood as the following: both $\mathcal{L}_{\mathrm{v}}^{XY}$ and $\mathcal{L}_{\mathrm{nv}}^{XY}$ alone contain the same information $X \nsim Y$, then when the ratio is taken, the two pieces of information cancel out. Taking the ratio thus compresses the system and keeps only essential information. Consequently,

if candidate models are not restricted to G_1 and G_2, once we have a preference for, for example, v-structure, we will also want to consider $\mathcal{L}_{\text{v}}^{YZ}$ and $\mathcal{L}_{\text{v}}^{ZX}$. It can be shown that

$$\log \frac{\mathcal{L}_{\text{v}}^{XY}}{\mathcal{L}_{\text{v}}^{ST}} = \log \frac{\mathcal{L}_{\text{nv}}^{XY}}{\mathcal{L}_{\text{nv}}^{ST}} = -N(I(X;Y) - I(S;T)) \quad \text{for all } \{ S, T \}. \tag{3.18}$$

Thus choosing the model with the highest likelihood among the three candidates amounts to choosing the model with the lowest mutual information, which is well expected as in v-structures, the missing edge corresponds to zero mutual information, and in non-v-structures, this is in accordance with the *data processing inequality*.

3.3.2. Search of possible d-separation

Section 3.3.1 shows an useful way to choose between a v-structure and a non-v-structure. From a more general point of view, it is also a criterion, in the sense of model selection, to tell if Z d-separates X and Y in that local structure. This motivates us to generalize eq. (3.16) to the case when $\{ X, Y, Z \}$ are embedded in a DAG.

The generalization will be done in a inductive and progressive manner. Firstly, consider a a DAG G over a set of vertices $\{ X, Y, U_1, \cdots, U_m \}, m \in \mathbb{Z}^+$ in which $X \not\sim Y$, and the set $U = \{ U_i \}_{i=1}^m$ d-separates X and Y. With the help of the local Markov property eq. (2.2), it can be shown that within the uniform likelihood comparison framework, the graph G is asymptotically likelihood-equivalent to any of the three graphs: $X \leftarrow U \rightarrow Y$, $X \rightarrow U \rightarrow Y$ and $X \leftarrow U \leftarrow Y$, where U is the joint variable of $\{ U_i \}_{i=1}^m$. As an example, consider $G' : X \rightarrow U_1 \rightarrow U_2 \leftarrow U_3 \rightarrow Y$ and $G : X \leftarrow U \rightarrow Y$, with two useful relations $A \perp\!\!\!\perp B \mid C \Rightarrow H(A \mid C) = H(A \mid B, C)$ and $A \perp\!\!\!\perp B \Rightarrow I(A;B) = 0$, we have

$$\begin{aligned}
\log \mathcal{L}(G', \hat{\theta}_1) &= -N[H(X) + H(U_1 \mid X) + H(U_2 \mid U_1, U_3) + H(U_3) + H(Y \mid U_3)] \\
&= -N[H(X) + H(U_1 \mid X, U_3) + H(U_2 \mid U_1, U_3, X)] \\
&\quad -N[H(U_3) + H(Y \mid U_1, U_2, U_3)] \\
&= -N[I(X;U_3) + H(X, U_1, U_2, U_3) + H(Y \mid U_1, U_2, U_3)] \\
&= -N[H(X \mid U_1, U_2, U_3) + H(U_1, U_2, U_3) + H(Y \mid U_1, U_2, U_3)] \\
&= \log \mathcal{L}(G, \hat{\theta}_2).
\end{aligned} \tag{3.19}$$

Without loss of generality, we can thus use G as a representative in the following discussion.

Now an extra vertex Z is added to G, and we are interested in knowing whether, in the new graph G, Z is necessary to d-separate X and Y. Let us start with a specific setting of skeleton, where Z is adjacent to all three vertices X, Y and U, and we are given two

candidate models, as shown in fig. 3.1. In the first model G_1 all edges connected to Z have incoming arrowheads at Z. This model can be viewed as a generalized v-structure, since conditioning on Z will d-connects X and Y through the path $X \to Z \leftarrow Y$. The log-likelihood of G_1 can be written straightforwardly as

$$\begin{aligned} \log \mathcal{L}(D, G_1, \hat{\theta}_c) &= -N[H(Z \mid X, Y, U) + H(X \mid U) + H(Y \mid U) + H(U)] \\ &= -N[H(Z, X, Y, U) + I(X; Y \mid U)]. \end{aligned} \tag{3.20}$$

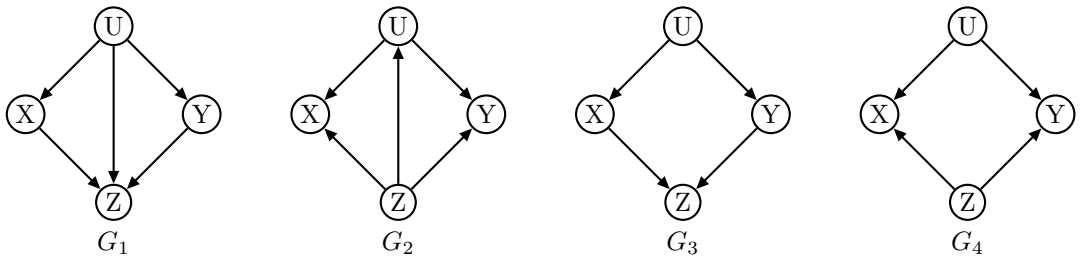

Figure 3.1. – Generalized v-structures (G_1, G_3) and non-v-structures (G_2, G_4).

Then in the second model G_2, all edges connected to Z have outgoing arrowheads from Z, just the opposite of G_1, and the model can be viewed as a generalized non-v-structure. And this time, Z must be included in the separating set in order to block any path $\gamma^Z_{X;Y}$. And for log-likelihood, we have

$$\begin{aligned} \log \mathcal{L}(D, G_2, \hat{\theta}_c) &= -N[H(Z) + H(X \mid U, Z) + H(Y \mid U, Z) + H(U \mid Z)] \\ &= -N[H(Z, X, Y, U) + I(X; Y \mid U, Z)]. \end{aligned} \tag{3.21}$$

It can be shown that within the same Markov equivalent family, any edge configuration other than G_1 leads to the same log-likelihood as eq. (3.21). Following the idea of the previous seciton, we can look at the log ratio of likelihoods:

$$\log \frac{\mathcal{L}_{G_1}}{\mathcal{L}_{G_2}} = -N[I_{\hat{\theta}_c}(X; Y \mid U) - I_{\hat{\theta}_c}(X; Y \mid U, Z)] = -N I_{\hat{\theta}_c}(X; Y; Z \mid U), \tag{3.22}$$

where $I(X; Y; Z \,|\, U) = I(X; Y \,|\, U) - I(X; Y \,|\, U, Z)$ is the conditional 3-point information.

Similar to the isolated case, we obtain an information term that can help us choosing between a generalized v-structure and a generalized non-v-structure. The conditional 3-point information is also permutation invariant, thus, additional comparisons are necessary to decide the edge between which pair is more likely missing. Not surprisingly,

we have

$$\log \frac{\mathcal{L}_{\mathrm{v}}^{XY}}{\mathcal{L}_{\mathrm{v}}^{ST}} = \log \frac{\mathcal{L}_{\mathrm{nv}}^{XY}}{\mathcal{L}_{\mathrm{nv}}^{ST}} = -N(I(X;Y \mid U) - I(S;T \mid U)) \quad \text{for all } \{ S,T \}, \tag{3.23}$$

which is just as intuitive and reasonable as in the isolated case.

So far so good, but keep in mind that all of the computations above for the "specially generalized" case are based on the same skeleton where Z is supposed to be adjacent to X, Y and U. This requirement is not as trivial as it seems. If one or more edges are missing between Z and the other vertices, although the graph looks simpler (sparser), the computations are counterintuitively not. Indeed, in eqs. (3.20) and (3.21), if G_1 and G_2 are replaced by G_3 and G_4 as in fig. 3.1, even if we can still rearrange to make appear the conditional information term, there will be extra terms that is generally non-zero, and will not cancel out each other when taken the ratio. Further reflection will make it clear that sparser graphs like G_3 and G_4 actually encode more information on d-separations than G_1 and G_2, and are, in this sense, more complex models.

Nonetheless, that extra bit of complication will not make eq. (3.22) any less useful. Unlike in the isolated case, we are not, at least not yet, counting on the sign of the conditional 3-point information to actually orient any edge. The important conclusion here is that, given a pair of variables X and Y, a possible (most likely necessary but not sufficient) set of candidate separating set $\{ U_i \}$, and another variable Z, a positive conditional 3-point information $I(X;Y;Z \mid \{ U_i \})$ implies that Z is more likely to be included in the candidate separating set $\{ U_i \}$, while a negative $I(X;Y;Z \mid \{ U_i \})$ implies that Z is more likely to be excluded from the separating set.

In fact, all scenarios presented here are to mimic some local structures during the reconstruction of the graph, when the skeleton has likely not been fully constructed. At the beginning of the section, we assumed that U d-separates X and Y, but that d-separation is restricted to the local induced graph and not necessarily true when the structure is embedded in a large graph. Therefore, as mentioned earlier, the motivation to study these local cases with the addition of Z is to find a reasonable way to search for possible vertices in the separating sets. Recall that in PC and PC-stable algorithms, this search is done in increasing-cardinality order, and randomly or lexicographically within each cardinality set. By contrast, here we have found a quantity that can not only give an indication of the possibility for each vertex to participate in a certain separating set, but also provide us a rather straightforward way to rank all candidates, and to possibly choose the best which may also be the right one, as we will see in the next section.

3.3.3. The 3off2 scheme

Definition of 3-point information provides a decomposation of the mutual information between a pair of variables X and Y [1, 2], as

$$I(X;Y) = I(X;Y;U_1)+I(X;Y;U_2|U_1)+\cdots+I(X;Y;U_n|\{U_i\}_{i=1}^{n-1})+I(X;Y|\{U_i\}_{i=1}^{n}). \tag{3.24}$$

Two particularities (or rather issues) can be observed in eq. (3.24). Firstly, the decomposation can continue non-stop provided that there are still variables left. Secondly, there is no fixed order on variables in the set $U = \{U_i\}$. In other words, there are a lot more than one way to decompose $I(X;Y)$ when the size of U is large. Given these two observations, eq. (3.24) alone is not more interesting than being a nice-looking reformulation. However, if put in the context of graphical model learning, it starts to shine.

Recall from eq. (3.2) that statistical conditional independence is equivalent to a zero mutual information. If in a graphical model G, there is a pair of variables $\{X, Y\}$ and a set of n variables U such that $X \perp\!\!\!\perp_p Y \mid U$, then we have $I(X;Y \mid U) = 0$. Then a slight rearrange of eq. (3.24) leads to

$$\begin{aligned} I(X;Y \mid U = \{U_i\}_{i=1}^{n}) =& I(X;Y) - I(X;Y;U_1) - I(X;Y;U_2 \mid U_1) \\ & - \cdots - I(X;Y;U_n \mid \{U_i\}_{i=1}^{n-1}) \\ =& 0. \end{aligned} \tag{3.25}$$

This equation gives a intuitive picture of how each of the vertices in the separating set U takes away a bit of information from $I(X;Y)$, eventually resulting in a zero conditional mutual information. However, 3-point information, as mentioned earlier, is not non-negative, how do we know that each of the 3-point information term in eq. (3.25) is taking away information out of, rather than giving information back to $I(X;Y)$? The answer to this question lies in the conclusion of section 3.3.2. If the separating set U is obtained not by random (or ordered like in PC-stable) statistical conditional independence tests, but by collecting each possible candidate one by one as described in section 3.3.2, then we are sure that each term in eq. (3.25) must be positive, as the recursive decomposation matches perfectly the candidate collecting procedure. Therefore section 3.3.2 gives a meaning to each 3-point information term of eq. (3.25), and the first of the two issues mentioned above is solved when a separating set exists: to be somewhat meaningful, the decomposation should be limited within the separating set.

Moreover section 3.3.2 also gives a attractive answer to the second issue on the ordering of the decomposation. While the sign of the 3-point information is a qualitative measure of a vertex's eligibility to participate in the d-separation, the value of the positive 3-

point can naturally serve as a quantitive measure of the contribution of each vertex to the d-separation. Therefore, for a pair of vertices $\{ X, Y \}$ and a half-built candidate separating set U, if there are two or vertices $\{ Z_i \}_{i \leq 2}$ that satisfy $I(X;Y;Z_i \mid U) > 0$, then the vertex with the largest positive 3-point information should be the next to be tested, since according to eq. (3.25) it can bring the largest reduce in the conditional mutual information between X and Y.

In summary, given a dataset over a set of variables V, and a pair of vertices $\{ X, Y \} \subset V$ with $I(X;Y) > 0$, to search for potential statistical conditional independence between X and Y, we start with set $U = \emptyset$ of potential separating vertices, and compute $I(X;Y;Z \mid U)$ for all $Z \in V \setminus \{ X, Y \}$, then choose $Z^* = \operatorname{argmax}_Z I(X;Y;Z \mid U)$ to be put into the set U, and restart the search in the rest of the vertices for the next candidate, until among all candidates, $\max I(X;Y;Z \mid U) \leq 0$ or $I(X;Y \mid U) = 0$. Since at each round, we maximize the 3-point information, the obtained conditional mutual information, even if non-zero, will be minimized. This scheme for the search of conditional independence is given a figurative name, called the *3off2* scheme.

The idea of the 3off2 scheme is analogous to that of the gradient descent optimization. Here the target function is the mutual information, and the direction of the next step is decided by the value of the positive 3-point information.

Finally, after quite a detour to the world of maximum likelihood, we return to the subject brought up at the beginning of this chapter: to test for statistical conditional independence using conditional mutual information, and to find a reasonable order for conducting those tests. Now equipped with the 3off2 scheme, we can introduce the following constraint-based structure learning method.

3.4. The MIIC algorithm

Before giving the description of the algorithm, several important concepts regarding the application need to be clarified, among which the most important one is the finite size correction when estimating (conditional) mutual information.

3.4.1. Correction for finite size estimation

As discussed in section 3.3, when comparing two locally different networks, using the same parameter set $\hat{\theta}_c$ connects the difference in log-likelihoods of the two networks with the local conditional mutual information and 3-point information. Since, according to section 3.3.3, we want to select graphical models based on the estimation of (conditional) mutual inforinformation, and to choose the order of conditional independence tests based

on the 3-point information, the estimation of information terms should not have a priori preference for any specific model. This suggests estimating information using the $\hat{\theta}_c$ parameter set, which does not assume any conditional independence relation.

For a given set of variables, $\hat{\theta}_c$ is complete, thus under ideal circumstances (no bias or error in data, infinite amount of data), the estimation of information is correct (convergence to the truth), so is the recovered d-separation structure. For real applications, however, since $\hat{\theta}_c$ is the maximized set for the family of complete DAGs, any estimation based on $\hat{\theta}_c$ will be consistently biased towards the complete DAG. This can be easily shown by looking at the log ratio of likelihoods between two DAGs that differ in only one edge $X \to Y$. With the help of eq. (2.5), this is

$$\begin{aligned} \log \frac{\mathcal{L}(G(V,E),\hat{\theta}_c)}{\mathcal{L}(G(V,E\setminus\{X\to Y\}),\hat{\theta}_c)} =& -N[H(Y\mid \mathrm{PA}(Y)) - H(Y\mid \mathrm{PA}(Y)\setminus\{X\})] \\ =& NI(X;Y\mid \mathrm{PA}(Y)\setminus\{X\}) \geq 0. \end{aligned} \tag{3.26}$$

In real life settings, $I(X;Y\mid \mathrm{PA}(Y)\setminus\{X\})$ is almost never zero, so using θ_c we have almost always $\mathcal{L}(G(V,E)) > \mathcal{L}(G(V,E\setminus\{X\to Y\}))$. In other words, graph with more edges is always preferred. This leads thus to a complete graph eventually.

As already mentioned in section 3.2, to offset this preference towards complex models, a penalty on the model complexity is required. Note, however, that this penalty is associated with the specific parameter set that encodes the information on d-separations contained in the graphical model, that is, the parameter set that factorizes according to a certain Bayesian network. For instance, suppose the complexity term $k(\theta_1)$ is associated with the log-likelihood of a model G_1 with parameter set θ_1. Under the uniform likelihood comparison framework, although the parameter set θ_c does not belong to any of the models under comparison, since we keep the factorization of different graphical models in eqs. (3.14), (3.15), (3.20) and (3.21), it is still a convenient and reasonable choice to use the same complexity term associated with each of these models. For example, if $k(\theta_1)$ is associated with $L_1(\hat{\theta}_1)$, under the uniform likelihood comparison framework, we can still use $k(\theta_1)$ for $L_1(\hat{\theta}_c)$.

In practice, commonly used criteria include the *Akaike Information Criterion* (AIC, [3]), the *Bayesian Information Criterion* (BIC, [14, 27]) and the Normalized Maximum Likelihood (NML, [28, 32]). They can all be regarded as different formulations under the principle of *Minimum Description Length* (MDL, [5, 27]), with the idea of choosing the simplest model possible that is compatible with the observed data. The details on the choice and implementation of model complexity is beyong the scope of this thesis. Here we just give a brief description of the chosen model, which is based on the idea of

factorized NML (fNML, [29]). Under the fNML framework, the complexity of a (CP)DAG model is decomposable into local terms, in the same manner as the factorization of the log-likelihood into empirical entropy. As a result, each entropy term $H(X \mid \mathrm{PA}(X))$ is associated with a complexity term $k_{X|\mathrm{PA}(X)}$. Accordingly, the entropy complexity can be adapted into (conditional) mutual information complexity, and each conditional mutual information estimation $I(X;Y \mid U)$ is associated with a complexity term $k_{X;Y|U}$.

To sum up, the introduction of model complexity comes down to appending to each estimation of information a penalty term, which leads to the shifted estimations of information given a sample of size N:

$$I'(X;Y \mid U) = I(X;Y \mid U) - \frac{k_{I(X;Y|U)}}{N}, \tag{3.27}$$

$$I'(X;Y;Z \mid U) = I(X;Y;Z \mid U) + \frac{k_{I(X;Y;Z|U)}}{N}. \tag{3.28}$$

The effect of these terms is to penalize and thus to avoid the choice of complex models. Most importantly, with eq. (3.27), the criterion for conditional independence becomes $I'(X;Y \mid U) \leq 0$, which can be interpreted as "when the estimated information is not more informative than the bare model itself", and which makes the search of conditional independence much more practical. Therefore, in the formulation of the 3off2 scheme, all information terms I should be replaced by the corresponding shifted information terms I'.

Details on the choice of complexity can be found in [1, 2]. The efficient computation is adapted from [21]. Related details on the estimation of information for mixed-type variables can be found in [7]. In the following, we will always assume the presence of the complexity term.

3.4.2. Pairwise score

Section 3.3.3 provides a 3-point information based ordered search for the separating set. However, a critical point is not addressed, which is the permutation invariance of the 3-point information. Given an edge $X - Y$ to be tested for conditional independence, if Z is chosen according to the 3off2 scheme with the highest 3-point information $I'(X;Y;Z)$, this does not necessarily mean that Z is a good candidate for separating X and Y, it may equally imply that X can be a good candidate for separating Y and Z, and Y a good candidate for separating Z and X. Thus, we need a measure that removes this ambiguity, i.e. a score that is unique to the pair X, Y.

Equations (3.18) and (3.23) suggest a solution by maximizing the (conditional) mutual information, but it is intended to compare different *bases* of conditional independence

$X - Y$, $Y - Z$, $Z - X$. and not for comparison different Zs with respect to the same base $X - Y$. Plus the fact that (conditional) mutual information is not comparable with 3-point information, so we cannot just combine the two information terms and take the maximum as the score. In order to make them comparable, we can create a likelihood-ratio based probability for each information term. As shown in eqs. (3.16) and (3.22), 3-point information is related to the log ratio of likelihoods between v-structure and non-v-structure, then the normalized ratio

$$P_{\mathrm{nv}}(X;Y;Z \mid U) = \frac{\mathcal{L}_{\mathrm{nv}}}{\mathcal{L}_{\mathrm{nv}} + \mathcal{L}_{\mathrm{v}}} = \frac{1}{1 + e^{-NI'(X;Y;Z|U)}} \tag{3.29}$$

takes its value in $[0, 1]$, and can be interpreted as "the probability of choosing a non-v-structure when given a non-v-structure and a v-structure as candidates for the same triple". Also, as it is monotone regarding the 3-point information, the order thereof will be kept. Similarly for conditional mutual information,

$$P_{\mathrm{b}}(Z;X,Y \mid U) = \frac{\mathcal{L}_{\mathrm{nv}}^{XY}}{\mathcal{L}_{\mathrm{nv}}^{XY} + \mathcal{L}_{\mathrm{nv}}^{YZ} + \mathcal{L}_{\mathrm{nv}}^{ZX}} = \frac{1}{1 + \frac{e^{-NI'(Y;Z|U)}}{e^{-NI'(X;Y|U)}} + \frac{e^{-NI'(Z;X|U)}}{e^{-NI'(X;Y|U)}}} \tag{3.30}$$

can be interpreted as "the probability of having a missing edge $X \nsim Y$ in the graph if one must be chosen among $X \nsim Y$, $Y \nsim Z$ and $Z \nsim X$". And the order on $I'(X;Y \mid U)$ is also kept. As it also takes value in $[0, 1]$, P_{nv} and P_{b} are now comparable, and we can use the score

$$S_{\mathrm{lb}}(Z;X,Y \mid U) = \min\left(P_{\mathrm{nv}}(X;Y;Z \mid U), P_{\mathrm{b}}(Z;X,Y \mid U)\right) \tag{3.31}$$

as the lower bound of our confidence that Z will contribute to the separation between X and Y. In particular, consider the "ill" case mentioned in section 3.1 where Z is almost identical to X, this is when conditional mutual information is not as effective as a measure of conditional independence. In that case, we have $I'(Y;Z \mid U) \approx I'(X;Y \mid U)$ and $I'(Z;X \mid U) \gg I'(X;Y \mid U)$, then eq. (3.30) ensures that $P_{\mathrm{b}}(Z;X,Y \mid U) \approx 1/2$, effectively preventing Z from being considered as a candidate for separating X and Y.

As a result, for each given edge $X - Y$, S_{lb} can be used to replace $I(X;Y;Z \mid U)$ in the 3off2 scheme when considering the order of all 3-point information terms. More importantly, S_{lb} also enables inter-pair comparison by comparing the maximized S_{lb} for each pair, given by

$$R(X,Y \mid U) = \max_{Z} S_{\mathrm{lb}}(Z;X,Y \mid U). \tag{3.32}$$

Compared to the classical lexicographic order on pairs used in PC and PC-stable al-

gorithms, $R(X, Y \mid U)$ gives a information based quantitive order that is much more reasonable.

3.4.3. Skeleton reconstruction

Algorithm 7 Skeleton reconstruction of the MIIC algorithm

Require: Set of variables $V = \{ X_i \}_{i=1}^{M}$, dataset $D(V)$ of size N.

$G \leftarrow G_c(V, E)$ ▷ *Complete undirected graph over V*
for all $X_i - X_j$ in G **do**
  if $I'(X_i; X_j) \leq 0$ **then**
    $E \leftarrow E \setminus (X_i, X_j)$ ▷ *$X_i \nsim X_j$ in G*
    $\mathrm{Sep}_{X_i;X_j} \leftarrow \emptyset$
  else ▷ *Compute simultaneously*
    $\mathrm{rank}(X_i, X_j) \leftarrow R(X_i, X_j \mid \emptyset)$
    $\mathrm{Sep}_{X_i;X_j} \leftarrow \{ \mathrm{argmax}_{Z \in \mathrm{adj}(X_i) \cup \mathrm{adj}(X_j)} S_{\mathrm{lb}}(Z; X, Y \mid \emptyset) \}$
Sort E by decreasing $\mathrm{rank}(X_i, X_j)$
$(S, T) \leftarrow$ first pair in E with highest rank
while $\mathrm{rank}(S, T) > 1/2$ **do** ▷ *Rank has a probabilistic meaning*
  if $I'(S; T \mid \mathrm{Sep}_{S;T}) \leq 0$ **then**
    $E \leftarrow E \setminus (S, T)$ ▷ *$S \nsim T$ in G*
  else ▷ *Compute simultaneously*
    $\mathrm{rank}(S, T) \leftarrow R(S, T \mid \mathrm{Sep}_{S;T})$
    $\mathrm{Sep}_{S;T} \leftarrow \mathrm{Sep}_{S;T} \cup \{ \mathrm{argmax}_{Z \in \mathrm{adj}(S) \cup \mathrm{adj}(T)} S_{\mathrm{lb}}(Z; S, T \mid \mathrm{Sep}_{S;T}) \}$
    Sort E by decreasing $\mathrm{rank}(X_i, X_j)$
  $(S, T) \leftarrow$ first pair in E with highest rank
return $G(V, E)$, $\{ \mathrm{Sep}_{X_i;X_j} \}$ for all $X_i \nsim X_j$ in G

With all the requisites, the skeleton reconstruction of MIIC can be described in a rather compact manner, as shown in algorithm 7. Starting from a complete graph, the first step is similar to other PC-derived algorithms where edges corresponding to marginal independence relations are removed, except that 1. the independence tests are done based on the estimation of mutual information; 2. for pairs not marginally independent, their most probable contributor to separation is found using eq. (3.32), alone with the corresponding score R. After the first round of survey, all remaining edges are ordered by their score. Then the algorithm enters the iterative mode and repeatedly looks at the pair with the current top score on the sorted edge list. If conditional independence is found, then the edge is removed from the list and the algorithm proceeds to the next top edge

on the list; otherwise the next most probable contributor for the pair is found with the updated score, in that case the whole list of edges will also be sorted again. The iteration stops when the pair with the highest score (X, Y) under consideration has a score less than one half, in terms of S_{lb}, this means that either the candidate contributor has $I'(X;Y;Z\,|U)$ negative, implying rather a non-contributor, or the value of the conditional mutual information $I(X;Y\,|\,U)$ is not low enough, implying that $X - Y$ is not the right based to be considered for conditional independence.

3.4.4. Edge orientation

The orientation step of MIIC differs more from the PC and PC-stable algorithms. On the one hand, the identification of v-structures and orientation propagations are integrated in a ordered manner based on the similar idea of information score as in the skeleton reconstruction step; on the other hand, the propagations of MIIC is more conservative than PC and PC-stable, and applies only the propagation rule described by corollary 4. As discussed in section 2.2.2, corollaries 3 and 4 are based on a relatively basic set of assumptions and on the data themselves, without extra model-based constraints.

At the beginning of the orientation step, we need to collect all unshielded triples. For PC and PC-stable, there is no specific order on the applications of rules on each individual triple, except for the general rules of v-structure first, followed by repeating the rules R_1, R_2, R_3. When given two unshielded triples that may possibly be v-structures, one may wonder which one to start first. Indeed, not only is it possible that two unshielded triples may give conflicting instructions, like in $X \to Z \leftarrow Y$ and $W \to X \leftarrow Z$, but also successive propagation steps may depend on the particular order of treatment of each of the earlier treated triple.

In MIIC however, the situation is much clearer, thanks again to the quantitive measure of 3-point information. So far we have been quantitive using only the positive 3-point information for the search of conditional independence. But we know that negative 3-point information can be as useful, especially when dealing with an unshielded triple, as seen already in section 3.3.1. Although the negative sign alone is already enough for a qualitative indication of v-structure. The absolute value of the 3-point information allows for comparing the "strength" of two possible v-structure. The one with higher absolute value will be considered as more probable as v-structure than the other one, as by definition it shows a higher log ratio of likelihoods of a v-structure against a non-v-structure.

Therefore, for each collected unshielded triple $X - Z - Y$, we can assign the absolute value of the 3-point information $|I'(X;Y;Z\,|\,\text{Sep}'_{X;Y})|$, where $\text{Sep}'_{X;Y} = \text{Sep}_{X;Y} \setminus \{\, Z \,\}$

(note that Z may be part of $\mathrm{Sep}_{X;Y}$), as its orientation score, analogous to the score $S_{\mathrm{lb}}(Z; X, Y \mid U)$ used for the search of conditional independence, except that this time, we are already sure about the missing of edge between X and Y, so S_{lb} is not needed. The orientation steps under this concept is sketched in algorithm 8.

Algorithm 8 Orientation step of the MIIC algorithm

Require: Graph skeleton G, set of unshielded triples $T = \{ X_i - X_k - X_j \}$, set of 3-point information $\{ I'(X_i; X_k; X_j \mid \mathrm{Sep}'_{X_i;X_j}) \}$ for each triple.

Sort T by decreasing $|I'(X_i, X_k, X_j \mid \mathrm{Sep}'_{X_i;X_j})|$
repeat
    $t_0 : X - Z - Y \leftarrow$ first triple in T with the highest rank
    $T \leftarrow T \setminus \{ t_0 \}$
    if $I'(X; Z; Y \mid \mathrm{Sep}'_{X;Y}) < 0$ **then**
        if $Z \notin \mathrm{Sep}_{X;Y}$ and no arrowheads to Z **then**
            Oriente t_0 as $X \rightarrow Z \leftarrow Y$
    else if $I'(X; Z; Y \mid \mathrm{Sep}'_{X;Y}) > 0$ and $X \rightarrow Z$ already established **then**
        Oriente t_0 as $X \rightarrow Z \rightarrow Y$
    Update newly oriented edges to triples sharing the same edges
until No more edges can be oriented

3.4.5. Orientation probability

While the plain score of 3-point information can help us decide the starting order of each triple for orientation, it fails to consider the evolvement of edge orientations that may simultaneously have an impact on a group of triples on the list. For example, when an unshielded triple is oriented as a v-structure $X \rightarrow Z \leftarrow Y$ following corollary 3, the orientation of another unshielded triple $X - Z - W$ should be updated to $X \rightarrow Z - W$, because the orientation $X \rightarrow Z$ comes necessarily from a v-structure with higher score. Then how should the score of the triple (X, Z, W) be updated? If we keep the old score, then it fails to reflect the fact that one of its edges has already been settled by a more reliable source; if we replace the score with that of the triple (X, Z, Y) (which is the current highest), then this will immediately bring (X, Z, W) to the place of the most trustworthy triple, whereas its own original score should be lower.

The dilemma suggest us have something similar to S_{lb} that allows us to combine scores of two different triples. Not only that, while keeping a score for each triple, we would also like to have a orientation-level measure that applies individually to each arrowhead (arrowtail), which shows our confidence for each of the orientations. In the example

above, after the update of the orientations of the triple (X, Z, W) to $X \to Z - W$, we would like to write down somewhere that we are pretty confident about the edge $X \to Z$, while less confident for whatever going to happen for the other edge between Z and W, which could be set using corollary 3 or corollary 4, depending on the sign of the associated 3-point information.

This leads us eventually to the idea of *orientation probability*, a measure given to each endpoint of an edge. For an edge $X - Z$, denote by p_{x2z} (p_{z2x}) the probability of having an arrowhead at Z (X), as $X \to Z$ ($X \leftarrow Z$) in the final graph, and by $1 - p_{x2z}$ the probability of having an arrowtail at the same endpoint. Consequently, if $p > 0.5$ then the endpoint is likely an arrowhead, if $p < 0.5$, then the endpoint is likely an arrowtail, and if $p = 0.5$, the endpoint is said to have undetermined orientation.

Given an unshielded triple $X - Z - Y$, if the associated 3-point information $I'(X; Y; Z \mid \mathrm{Sep}'_{X;Y})$ is positive, then the triple is likely a non-v-structure, with the probability given by eq. (3.29). However, due to the Markov equivalence of different non-v-structures, we cannot assign this probability to any of the four associated orientation probabilities $p_{z2x}, p_{x2z}, p_{y2z}, p_{z2y}$. If $I'(X; Y; Z \mid \mathrm{Sep}'_{X;Y})$ is negative, the triple is more likely a v-structure, and this time, we can orient the triple as $X \to Z \leftarrow Y$, with the probability set similar to eq. (3.29):

$$P_{\mathrm{v}}(X; Y; Z \mid \mathrm{Sep}'_{X;Y}) = \frac{\mathcal{L}_{\mathrm{v}}}{\mathcal{L}_{\mathrm{nv}} + \mathcal{L}_{\mathrm{v}}} = \frac{1}{1 + e^{NI'(X;Y;Z|\mathrm{Sep}'_{X;Y})}}. \tag{3.33}$$

Therefore, we can set $p_{x2z} = p_{y2z} = \{\, 1 + \exp[NI'(X; Y; Z \mid \mathrm{Sep}'_{X;Y})] \,\}^{-1}$, and, at the same time, set the score of the triple to the same value. In principal, in the above example, we should also have $p_{z2x} = 1 - p_{x2z}$ and $p_{z2y} = 1 - p_{y2z}$. While this is required under the causal sufficiency assumption, it is not mandatory for the orientation step, and can be relaxed. And we will see later (section 3.5.1) that the relaxation of this constraint can provide a more powerful and more meaningful representation of the final graph.

Now return to the original problem of combining the score of two triples $t_1 : X \to Z \leftarrow Y$ and $t_2 : X \to Z - W$ during the propagation of orientations. Firstly, since orientation probability is endpoint distinct, we can first update the proba $p^{t_2}_{x2z} = p^{t_1}_{x2z} = \{\, 1 + \exp[NI'(X; Y; Z \mid \mathrm{Sep}'_{X;Y})] \,\}^{-1}$. Then after the propagation, we are confident that the endpoint from W to Z is an arrowtail, and the probability of this tail is given by

$$\begin{aligned} p^{\mathrm{tail}}_{w2z} =& p_{x2z} \cdot P_{\mathrm{nv}}(X; Y; Z \mid \mathrm{Sep}'_{X;Y}) \\ =& \frac{p_{x2z}}{1 + e^{-NI'(X;Y;Z|\mathrm{Sep}'_{X;Y})}}. \end{aligned} \tag{3.34}$$

p^{tail}_{w2z} can be interpreted as the joint probability of having an arrowhead at Z for edge $X - Z$ due to a v-structure and the unshielded triple $X - Z - W$ being a non-v-structure, or more generally, as the probability of applying the propagation rule using corollary 4. Since we choose to keep the probability of having an arrowhead, after the propagation, we will set $p_{w2z} = 1 - p^{\text{tail}}_{w2z}$. More over, p^{tail}_{w2z} combines the original scores of the two triples, and can thus be used as the new score of the triple (X, Z, W) after the propagation.

Under this endpoint distinct probabilistic representation, the score for each triple is no longer a measure of the likelihood of it being a v-structure or non-v-structure, but rather a measure of its strongest newly built probability. To understand the idea of "newly built", consider the triple $t_2 : X \to Z \to W$ obtained after the propagation. The orientation $X \to Z$ is an "old" one coming from $t_1 : X \to Z \leftarrow Y$, whereas the orientation $Z \to W$ is newly built, and thus all other triples that contain the undirected edge $Z - W$ should be updated by this newly built orientation. A more detailed description of the orientation step of MIIC using the notion of orientation probability is given by algorithm 9.

3.5. Interpretability of MIIC

As a constraint-based method, MIIC incorporates the features of likelihood and conditional mutual information, to allow for quantitive orderings of conditional independence tests during the skeleton reconstruction, and of unshielded triples during the edge orientation. In the following, we discuss several modifications that emphasize on the interpretability and scalability of the reconstructed network.

3.5.1. Genuine vs. putative causal edges

The introduction of orientation probability in section 3.4.5 distinguish MIIC from other PC-derived methods. Here we propose to further exploit this idea to enhance the representation of the edge orientations in the reconstructed graph.

As seen in algorithm 9, the orientation probabilities are not only used to order the triples during the the orientation step, but also used to decide the type of each endpoint of an edge. Although each end point is assigned a different probability related to the absolute value of their corresponding 3-point information, this differentiation is not reflected on the final orientation of the edges. For example, compare the edge $X \to Z$ with $p_{\text{x2z}} = 0.95$ and another edge $S \to T$ with $p_{\text{s2t}} = 0.6$, they both appear to be directed edges in the final graph, except that by comparing p_{x2z} and p_{s2t}, we know that we have more confidence about the arrowhead in $X \to Z$ than that in $S \to T$. Therefore,

Algorithm 9 Orientation step of the MIIC algorithm using orientation probability

Require: Graph skeleton G, set of unshielded triples $T = \{ X_i - X_k - X_j \}$, set of 3-point information $\{ I'(X_i; X_k; X_j \mid \mathrm{Sep}'_{X_i;X_j}) \}$ for each triple.

for all triples $t : X_i - X_k - X_j \in T$ **do**
  $p_{\mathrm{k2i}} = p_{\mathrm{i2k}} = p_{\mathrm{j2k}} = p_{\mathrm{k2j}} \leftarrow 1/2$
  if $I'(X_i; X_j; X_k \mid \mathrm{Sep}'_{X_i;X_j}) < 0$ **then**
    $\mathrm{rank}(t) \leftarrow P_{\mathrm{v}}(X_i; X_j; X_k \mid \mathrm{Sep}'_{X_i;X_j})$
    $p_{\mathrm{i2k}} = p_{\mathrm{j2k}} \leftarrow \mathrm{rank}(t)$, mark both as new
  else
    $\mathrm{rank}(t) \leftarrow 0$
while T is not empty **do**
  Sort T by decreasing $\mathrm{rank}(t)$
  $t_0 : X - Z - Y \leftarrow$ first triple in T with the highest rank
  $T \leftarrow T \setminus \{ t_0 \}$
  if $\mathrm{rank}(t_0) \leq 1/2$ **then**
    break ▷ *Rank has a probabilistic meaning*
  for all triples $t : X_i - X_k - X_j \in T$ **do**
    if p_{x2z} is new and t and t_0 share the edge $X - Z : X \equiv X_i, Z \equiv X_k$ **then**
      $p_{\mathrm{i2k}} \leftarrow p_{\mathrm{x2z}}$ mark as old
      if both p_{i2k} and p_{j2k} are old **then**
        $T \leftarrow T \setminus \{ t \}$
      else if $p_{\mathrm{i2k}} > 1/2$ and $I'(X_i; X_j; X_k \mid \mathrm{Sep}'_{X_i;X_j}) > 0$ **then**
        $p_{\mathrm{j2k}} \leftarrow 1 - p_{\mathrm{i2k}} \cdot P_{\mathrm{nv}}(X_i; X_j; X_k \mid \mathrm{Sep}'_{X_i;X_j})$ mark as new
        $\mathrm{rank}(t) \leftarrow 1 - p_{\mathrm{j2k}}$

to not waste the precious information carried by those orientation probabilities, we need a way to differentiate edges with different probabilities in the final graph.

In terms of visualization, we can do so by using varying colors for the edges depending on their probability *extremity*, defined as $|p - 0.5|$, but this can make the final graph quite garish and difficult to grasp the key information. Here we propose a more succinct representation, by setting a *orientation probability threshold* $p^* \in (0.5, 1]$, that filters the probability of all endpoints. Instead of determining the type of endpoint by comparing with $1/2$, we require that an endpoint must have $p > p^*$ to be considered as an arrowhead in the final graph, and $p < 1-p^*$ to be an arrowtail, whereas endpoints with $1-p^* < p < p^*$ will be considered as having undetermined type.

With the filtering, we can give a quantitive description of our confidence to all ar-

rowheads present in the final graph. Furthermore, under certain condition, we can also distinguish arrowtail from undetermined endpoint, both of which are for now represented as arrowtail in the final graph. The condition is to treat two endpoints of an edge separatedly during the orientation step. More specifically, this requires that in algorithm 9, when the orientation probability p_{x2z} is determined, the other endpoint probability p_{z2x} will not be automatically updated to $1 - p_{\mathrm{x2z}}$. This is as if we drop the causal sufficiency assumption specifically for orientation step. As mentioned earlier, this sometimes allow the final graph to capture conflicting orientations that lead to bidirected edge $X \leftrightarrow Z$, which implies the existence of a possible latent confounding variable. More importantly, with the separated treatment of endpoints, we can now have two directed edges, one $X \to Z$ with endpoint probabilities $(0.05, 0.95)$, and the other $S \to T$ with endpoint probabilities $(0.5, 0.95)$. While they look identical in the final graph, based on the endpoints probabilities we are confident that $X \to Z$ is indeed a directed edge that implies a causal relation between X and Z, whereas for $S \to T$ we can only conclude that T is not a parent of S, but S is not necessarily a parent of T, because the endpoint probability on S implies an undetermined type, which could be either an arrowtail (in which case the relation will be causal) or an arrowhead (in which case a latent confounding variable is implied).

Since the two types of edges have distinct causal implications, we want to make the difference visible in the final graph. This leads to the concepts of *genuine causal edge* and *putative causal edge*, where the naming "putative" comes from [25]. Given an edge $X \to Z$ with endpoint probabilities $(p_{\mathrm{z2x}}, p_{\mathrm{x2z}})$, and a threshold p^*, the edge is called an genuine causal edge if the endpoint probabilities satisfy $p_{\mathrm{x2z}} > p^*$ and $p_{\mathrm{z2x}} < 1 - p^*$; the edge is called an putative causal edge if $p_{\mathrm{x2z}} > p^*$ and $1 - p^* \leq p_{\mathrm{z2x}} \leq p^*$. Visually, genuine causal edge will have highlighted arrowhead, indicating a probable causal relation. The difference between different types of edges and their relation to endpoint probabilities is sketched in fig. 3.2.

3.5.2. Conservative orientations

During the orientation step of MIIC, although the identification of v-structures is done in principle according to corollary 3, the actual criterion used, as shown in algorithm 9, is the negativity of the 3-point information. While under ideal circumstances, this condition is equivalent to corollary 3, when the sample size is finite and errors and biases exist in the dataset, the condition may fail to represent a true v-structure, and will lead to erroneous orientations, sometimes even with the probability filtering introduced in the previous section.

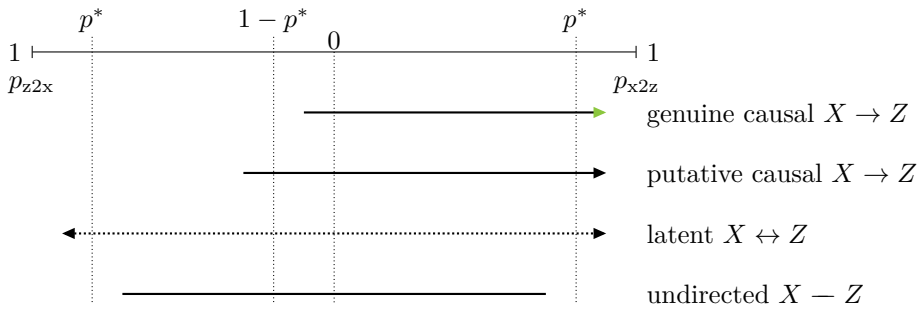

Figure 3.2. – Different types of edges supported by MIIC, arrowhead of a genuine causal edge is highlighted as it implies a probable causal relation.

In corollary 3, the key condition for $X - Z - Y$ to form a v-structure is that Z must not be in the separating set of X and Y. In algorithm 9, this condition is tranformed into $I'(X;Y;Z \mid \mathrm{Sep}_{X;Y}) = I'(X;Y \mid \mathrm{Sep}_{X;Y}) - I'(X;Y \mid \mathrm{Sep}_{X;Y}, Z) < 0$. Under ideal circumstances, the conditional independence between X and Y means $I(X;Y \mid \mathrm{Sep}_{X;Y}) = 0$, and we must also have $I(X;Y \mid \mathrm{Sep}_{X;Y}, Z) > 0$ as the conditioning on Z d-connects X and Y. In practice, with the the complexity terms introduced by the finite size correction (section 3.4.1), a valid v-structure predicted by $I'(X;Y;Z \mid \mathrm{Sep}_{X;Y}) < 0$ will expect $I'(X;Y \mid \mathrm{Sep}_{X;Y}) < 0$ and $I'(X;Y \mid \mathrm{Sep}_{X;Y}, Z) > 0$. However, the latter condition is not actually tested, and this gives space to some ill instances where we have instead $I'(X;Y \mid \mathrm{Sep}_{X;Y}) < I'(X;Y \mid \mathrm{Sep}_{X;Y}, Z) < 0$. This means that though not actually acquired, the vertex Z can be part of the separating set of X and Y, even if the inclusion will slightly increase the conditional mutual information. As a result, a seemingly false v-structure will be identified by MIIC, and since this happens typically at the very beginning of the orientation step, the orientations will be propagated and further accumulate the possible errors in the final graph. While under strong data noises and inaccurate estimation of information, this the v-structure could indeed be an error, in some other situations, sepecially when the size of the dataset is small but the structure of the underlying network is relatively complex, the identified v-structure could well be a true one, just without enough data to be properly recovered.

To avoid the identification of such v-structures, we propose to rectify all negative values of (conditional) mutual information to zero, so that a negative $I'(X;Y;Z \mid \mathrm{Sep}_{X;Y})$ can only result from a positive $I'(X;Y \mid \mathrm{Sep}_{X;Y}, Z)$. While this modification is intended to correct the orientation step only, the skeleton reconstruction step, in principle, may also get affected due the slight change of the specific value of score R for each triple and thus the change of ordering of conditional independence tests. But this change will not invalidate any theoretical part on which the algorithm is built.

At first glance, the rectification of negative (conditional) mutual information seems like an over-kill. Indeed, if only aiming at false v-structures, conducting an additional check on the term $I'(X;Y\,|\,\mathrm{Sep}_{X;Y},Z)$, and rejecting any non-positive value will be just enough. However, as we eventually realize, there is a deeper meaning behind the rectification, which is related to the general definition of mutual information.

Concerning the estimation of conditional mutual information, MIIC can handle discrete, continuous, as well as mixed variables. While for discrete variables, the estimation can are done by simple frequency counting in the joint space of variables (recall that we assume the parameter set θ_c), the continuous and mixed variables are more complicated to work with. In such cases, MIIC's estimation of conditional mutual information is based on the discretization of these variables, through an approximate optimum partitioning scheme, which is supported by the general definition of mutual information as the supremum of all possible finite partitioning of random variables [12]. The details on the efficient implementation of the scheme by an dynamic programming algorithm can be found in [7].

As we are dealing with finite sized sample, model complexity must be taken into consideration. Thus the actual quantity that we maximize when estimating mutual information is the shifted mutual information where the mutual information term is appended by the complexity term, as discussed in section 3.4.1. Following the idea of optimum partitioning, it can be shown that for both mutual information and conditional mutual information with finite size correction concerning a pair of continuous (or mixed) variables (X, Y), if the estimation gives negative value, we can always achieve a zero-value estimation by enforcing a partitioning of a single bin for X or Y. Algorithmically speaking, the single bin partitioning will cut abruptly both the value of mutual information and the value complexity term to zero, which is thus a singularity for the dynamic programming based algorithm. Thus the rectification is necessary if always assuming the sumpremum of mutual information, and is not only for the correction of false v-structures. Given the similarity of the motivation of the modification with the conservative PC algorithm (see section 2.7), we call this version of MIIC the *conservative MIIC*.

To have an idea on the impact of the negative value correction on MIIC for the orientation, we first compare the performance of original MIIC that allows for negative estimation of conditional mutual information and PC-stable algorithm with majority orientation rules (see section 2.7), for both discrete and continuous dataset. Note that due to the discretization algorithm, the estimation of mutual information is already forced to be non-negative. As shown in fig. 3.3, compared to PC-stable with majority rules, MIIC greatly reduces the imbalance between precision and recall for all sample sizes. It also significantly reduces the precision gap between skeleton and oriented graphs

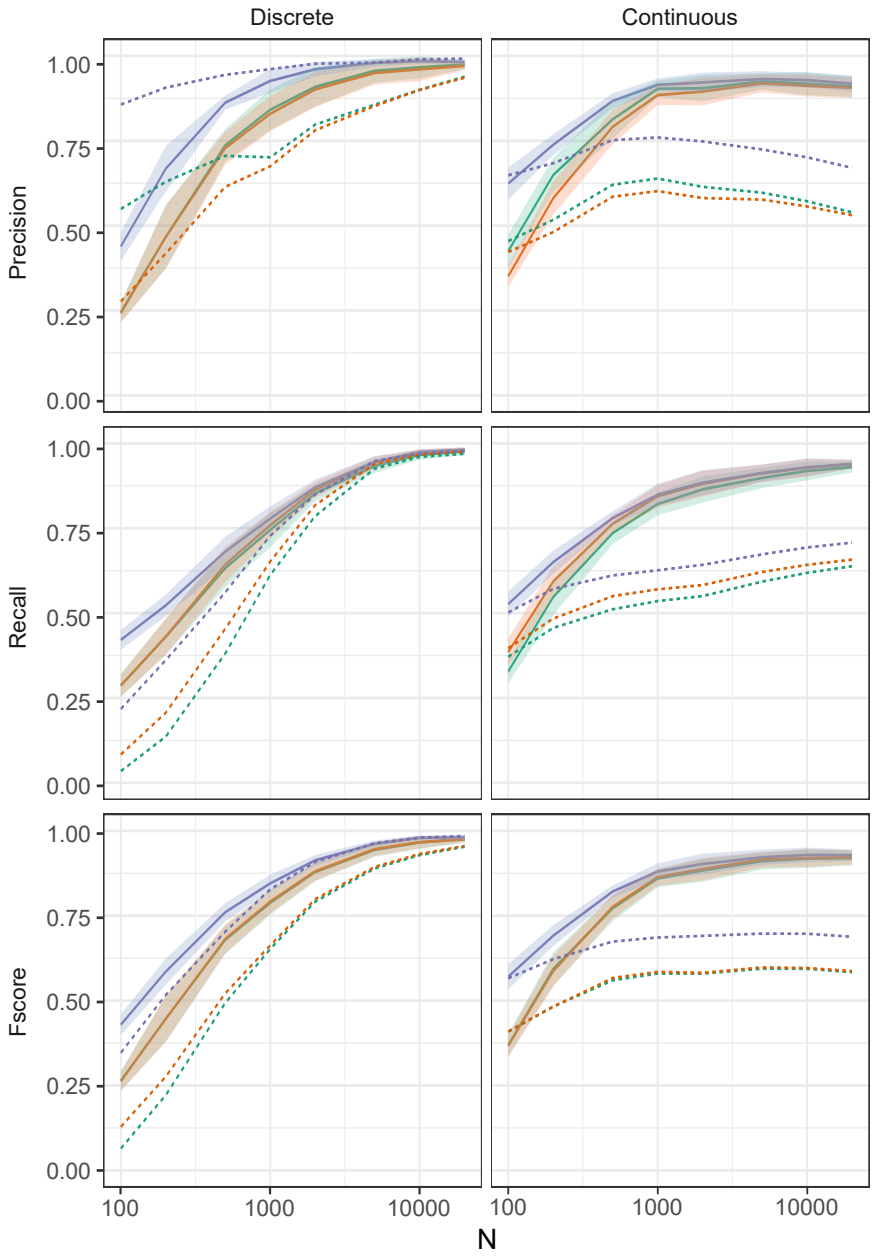

Figure 3.3. – MIIC allowing for negative conditional mutual information estimation (solid lines) and PC-stable with majority rules (dashed lines), on discrete (left column) and continuous (right column) data. Datasets are generated from random 100-node DAGs with average degree 2.7 and maximum degree 4. Performance is measured at the level of skeleton (blue), CPDAG (red) and oriented-edge-only subgraph (green).

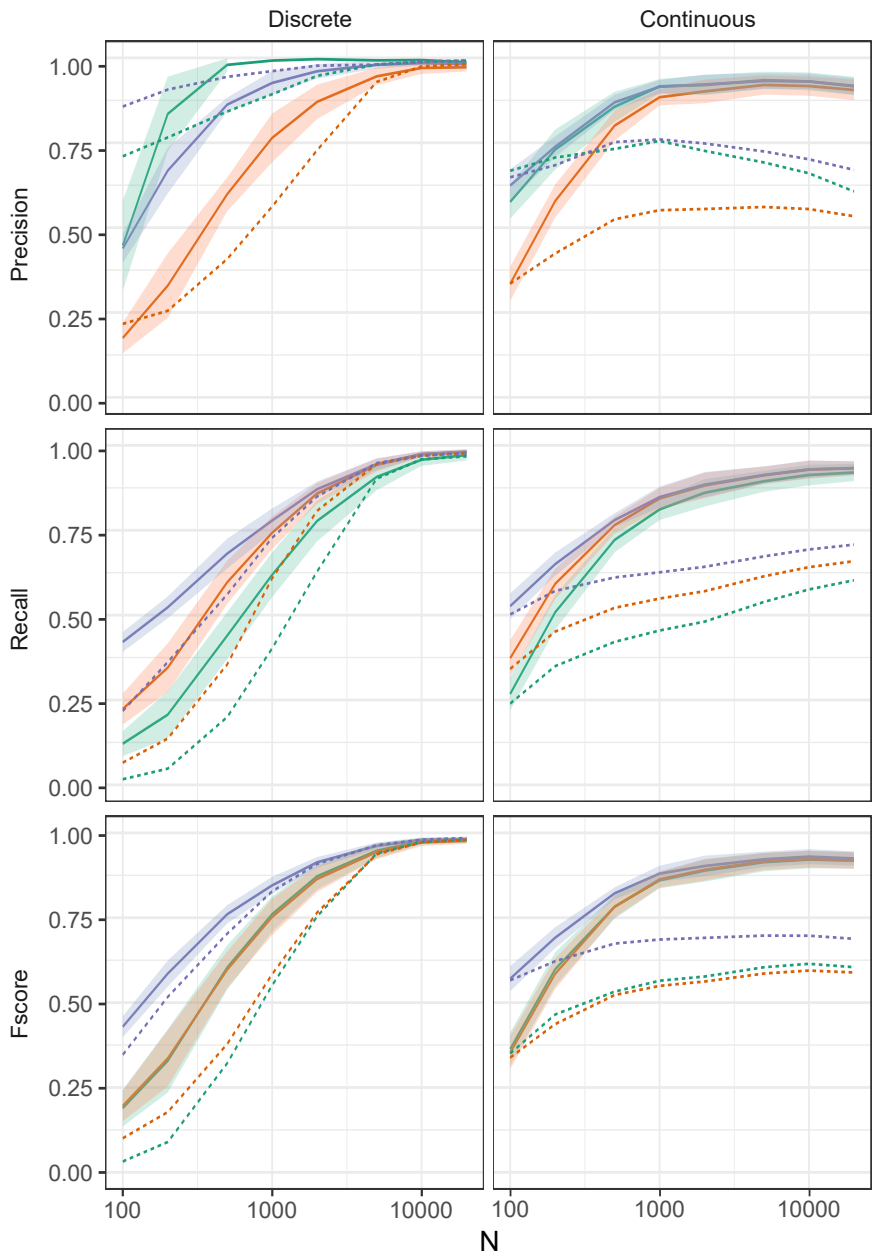

Figure 3.4. – MIIC enforcing non-negative conditional mutual information estimation (solid lines) and PC-stable with conservative rules (dashed lines), on discrete (left column) and continuous (right column) data. Datasets are the same as in fig. 3.3. Performance is measured at the level of skeleton (blue), CPDAG (red) and oriented-edge-only subgraph (green).

for larger datasets. However, a substantial loss in precision remains between skeleton and oriented graphs for smaller datasets.

Then the same comparison is done for MIIC that rectifies all negative conditional mutual information estimations and PC-stable algorithm with conservative orientation rules, as shown in fig. 3.4. Compared to the original version, conservative MIIC, as expected, hardly affects the performance of the skeleton reconstruction, while it has a clear influence on both oriented graphs. For CPDAGs, the overall performance of conservative MIIC seems to be worse for discrete data, this implies that the false v-structures corrected by conservative MIIC are likely those that are true but cannot be recovered due to the limited sample size, as discussed previously. By contrast, for oriented-edge-only subgraphs, conservative MIIC greatly improves the precision of predicted orientations for discrete data, with a small loss in recall. On continuous data, conservative MIIC also achieves a significant increase in orientation precision, even making it at par with the precision of skeleton, for datasets of all sizes, which also clearly overperforms conservative PC-stable algorithm.

3.5.3. Separating set consistency

The implementation of the iterative approach to ensure separating set consistency is rather straightforward. In particular, to keep the balance between interpretability and informativeness, the implemented skeleton-consistent algorithm is the modified version as discussed in section 2.5. The idea of consensus graph discussed in the same section is also implemented in MIIC with an adjustable consensus threshold α_c, to prepare for situations where the consistent cycle may have a relatively large size. Moreover, the feature of having different edge types as discussed in section 3.5.1 is also kept in the consensus graph, by using the average probability of all graphs in the consistent cycle as the edge probabilities for the consensus graph.

3.5.4. Contextual variables

Apart from the introduction of different edge types regarding their causal implication, another advantage brought by the the separate probabilistic treatment of the two endpoint of each edge also allows to include prior knowledge about certain head or tail orientations. In particular, it is often interesting to include in graphical models, a few *contextual variables* characterizing for instance the patient profile, a control parameter or specific experimental conditions, depending on the nature of the dataset. Unlike most other variables of the dataset, such contextual variables are not stochastically varying, but instead externally set or selected. Such variables, if the possibility of potential selection

bias are carefully examined and excluded, should have all their edges without incoming arrowhead. For example, if a variable X is considered as a contextual variable, then we will enforce $p_{\mathrm{y2x}} = 0$ for all edges $Y - X$ connected to X. This expresses our prior knowledge that the particular contextual variable cannot be the consequence of other observedvariables as they actually correspond to manually set external parameters or conditions.

3.5.5. Scalability issue

The scores in terms of probabilities enable the comparison between different sources of confidence. However, their exponential nature makes them extremely non-linear with respect to the information terms based on which they are defined. Moreover, since the sample size N is also involved in the exponent, in practice the comparison between different scores are often unfeasible when the exponent is so large that the term $e^{-NI'(X;Y;Z|\mathrm{Sep}_{X;Y})}$ goes below the typical precision of floating points, and cannot be precisely stored in most modern machines.

To circumvent the issue, we need an alternative representation of the scores that are comparable even at large N. In eq. (3.29), the probability $P_{\mathrm{nv}}(X;Y;Z \mid U)$ is monotone with respect to the 3-point information $I'(X;Y;Z \mid U)$, this reminds us that if all scores can be expressed in the same form, then instead of the probabilities we can compare the exponents, which hardly goes beyond the range of floating points.

For example, during the skeleton reconstruction, the probability $P_{\mathrm{b}}(Z;X,Y \mid U)$ in eq. (3.30) can be rewritten as

$$\begin{aligned} P_{\mathrm{b}}(Z;X,Y \mid U) =& \frac{1}{1+\frac{e^{-NI'(Y;Z|U)}}{e^{-NI'(X;Y|U)}}+\frac{e^{-NI'(Z;X|U)}}{e^{-NI'(X;Y|U)}}} \\ =& \frac{1}{1+e^{-N(I'_{YZ}-I'_{XY})}+e^{-N(I'_{ZX}-I'_{XY})}} \\ =& \frac{1}{1+e^{-[I'_{\min}-\log\left(1+e^{-(I'_{\max}-I'_{\min})}\right)]}}, \end{aligned} \tag{3.35}$$

where $(I'_{\min}, I'_{\max}) = \mathrm{minmax}(NI'_{YZ} - NI'_{XY}, NI'_{ZX} - NI'_{XY})$. Then we can define

$$\mathrm{score}_{\mathrm{nv}} = NI'(X;Y;Z \mid U), \tag{3.36}$$

$$\mathrm{score}_{\mathrm{b}} = I'_{\min} - \log\left(1+e^{-(I'_{\max}-I'_{\min})}\right), \tag{3.37}$$

and then redefine eq. (3.31) as

$$S_{\mathrm{lb}}(Z;X,Y \mid U) = \min\left(\mathrm{score}_{\mathrm{nv}}, \mathrm{score}_{\mathrm{b}}\right). \tag{3.38}$$

Accordingly, the stop condition in algorithm 7 should be adapted to $\text{rank}(S, T) > 0$.

Similar idea applies to the orientation step, where we can define

$$\text{rank}_{\text{nv}} = NI'(X; Y; Z \mid U), \tag{3.39}$$

$$\text{rank}_{\text{v}} = -NI'(X; Y; Z \mid U). \tag{3.40}$$

And for a triple t_i with positive 3-point information, its rank after being applied the propagation rule with another v-structure will be

$$\text{rank}_i = \text{rank}_{\min} - \log\left(1 + e^{-(\text{rank}_{\max} - \text{rank}_{\min})} + e^{-\text{rank}_{\max}}\right), \tag{3.41}$$

where $(\text{rank}_{\min}, \text{rank}_{\max}) = \text{minmax}\left(\text{rank}_{\text{nv}}, \text{rank}_{\text{v}}\right)$. As a result, all probabilities in algorithm 9 should be replaced by the new rank terms, and the break condition will be adapted to $\text{rank}(t_0) \leq 0$.

With those new expression for scores and ranks, the precision of the computation of the algorithm when dealing with large dataset is greatly improved.

3.6. Notes on the codebase refactoring

Code refactoring is by no means an easy task, and is believed by some to be more difficult than restarting from scratch. The original code of MIIC has, due to historical reasons, a mixed flavor of R, C, and C++. Some parts of C++ codes are named and coded in the R style, while others are in the C style. Traces of the evolution of the codebase can be found everywhere, including obsolete functions and declarations, as well as temporary testing and debugging codes. As a result, the codes are quite difficult for addition or deletion of any functionality, and a simple algorithmic change can result in modifications of several files or even the whole codebase. As the chinese saying goes, it is like that "the whole body is affected by a strand of hair". Additionally, the often obscure variable naming and the lack of comments make it quite hard for anyone to understand, improve, or debug the code except for the authors themselves. All these issues encourage a thorough revision and refactoring of the codebase.

Before digging into details, it may be worth first having an overall look at the result of the refactoring.

Readability

This is among the aspects of premier priorities in a collaborative project. It's not something that can be achieved by a single commit or a paragraph of description. Rather, it requires consistent effort of many aspects combined. By renaming

variables, restructuring functions, regrouping methods and adding comments, the code is now much easier to follow than before the refactoring.

Standardization

By referring to text books, the Internet and the codebase of some famous C++ projects, we make sure the best/common practice is used whenever possible throughout the code base.

Testability

The unit test (as a common and good practice) is not yet implemented, but the simplified code/function structure now allows almost all the functions to be reasonably tested.

Bug fixes

A by-product of the improved testability and readability.

Computation complexity

With simplified structures and optimized algorithms, MIIC now works around 40%-60% faster on continuous dataset.

Codebase size

Without any loss of functionality, the codebase at the end of the refactoring has around 5.3k lines of code, as compared to around 13.3k lines before the refactoring, this amounts to a reduction of 60% in the size of the codebase.

Along with the above mentioned, the codebase is now publicly available on GitHub [1], where we find the most open-source projects nowadays.

3.6.1. Data structure organization

During the refactoring, many functions are rewritten, declarations are regrouped or divided, and different modules are created. All the changes are for the purpose of having a cleaner and more logical data and code structure, allowing for easy modifications and additions of functionality, as well as improving the overall maintainability of the codebase. At the same time, it will make it easier for readers to follow the logic flow, as if they were reading the pseudocode of the algorithms.

In summary, the changes in data structure and code structure are guided by the following rules: objects at the lowest level, functions at the intermediate level, separate modules at the highest level. More precisely, at the lowest level, simiar or closely related concepts are grouped together. For example, the concept of an edge includes its associated edge type, mutual information, conditional mutual information, complexity

1. `https://github.com/miicTeam/miic_R_package`

terms, separating set as well as the next best contributor. Then at the intermediate level, the code should consist of a group of functions, each giving an instruction about a specific task, for example, to initialize all edges, to search for the next best contributor for a pair of vertices, or to check for the existence of a consistent cycle. Finally at the highest level, functions that belong to the same algorithm/step will be put into the same file to create a module, for example, the module for skeleton reconstruction, the module for edge orientation, the module for information estimation. The modules of MIIC after the refactoring is illustrated in fig. 3.5.

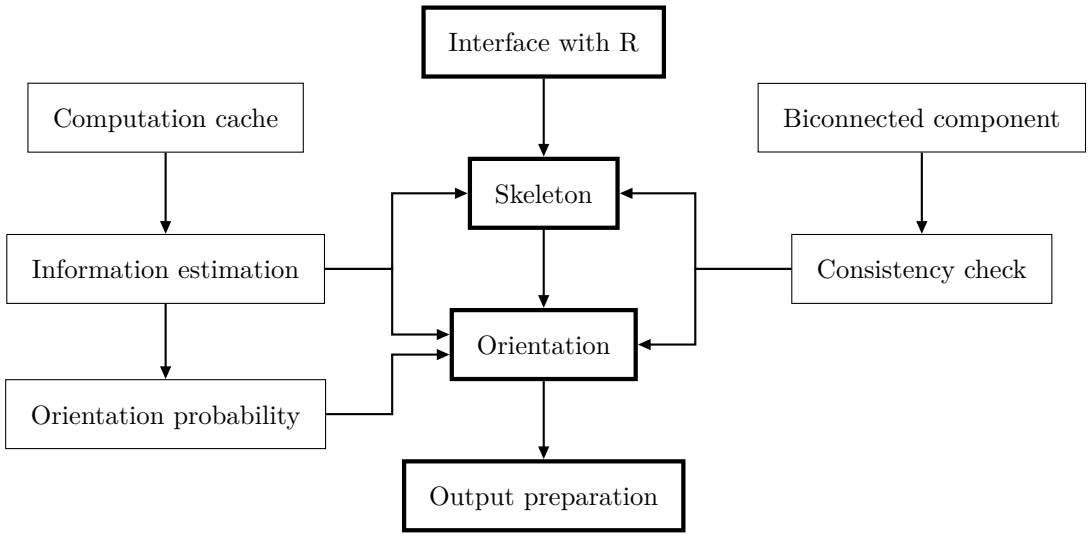

Figure 3.5. – Different modules of MIIC.

3.6.2. Algorithmic efficiency

At the function level, many factors, including improper use of vectors, function calls at the wrong scale (inside or outside loop), redundant computations and excessive demand for resources, can lead to a bloated call stack during the execution of the program, which will unnecessarily consume more resources, slow down the program, and also make the code more prune to bugs when making modifications. After the rewriting of functions and reorganization of data structures, the typical call stack size of MIIC, according to the profiling, is significantly reduced.

Additionally, after the implementation of separating set consistency algorithms, a new module is created to cache the results of estimations of information, where each entry is registered by the key (X, Y, Z, U), and is to be reused during the separating set consistency iterations. The reuse of computation cache significantly reduces the time spent in successive iterations when the separating set consistency option is active.

3.6.3. Memory management

Before the refactoring, the memory management in MIIC was done mostly mannually, with `new`s (`malloc`s) and `delete`s (`free`s) everywhere, often unpaired that lead to leaks of memory. Now C++ flavored memory management with vectors and smart pointers takes control. In particular, for the computation-intensive parts where frequent allocation and deallocation of memory can cause excessive overhead, a simple linear allocator is in place, which, at the beginning of the program, reserves a large contiguous block of memory of precalculated size that will meet the expection of the worst case. Then during the computation, the allocation and deallocation of memory over the preassigned block will have almost no cost, which significantly accelerates the computation of MIIC, especially for the estimation of information for continuous variables, where multiple levels of iterations can happen at the same time. However, a major defect of the linear allocator is its lack of flexibility, in the sense that the reserved space must not be exceeded, and thus must be maintained with extreme caution when adding or removing computation related functionalities, though this is more of a burden to the programmer than to the program.

4. Application to the SEER Databse

SEER is a largely studied dataset of cancer incidence and population data supported by the Surveillance Research Program (SRP) in the National Cancer Institute's (NCI) Division of Cancer Control and Population Sciences (DCCPS). It has been collecting data from patients diagnosed with cancer in certain regions of the United States of America since 1973. Currently, SEER collects cancer incidence data from population-based cancer registries covering approximately 34.6% of the US population [18]. In total, SEER contains data records of over 11,000,000 individuals in the entire database.

In our study, a subset of records of 396,179 patients are selected. Firstly, only patients diagnosed with breast cancer are considered. Besides, due to changes in the database throughout the years, including the recent addition of several treatment variables, only patients diagnosed between the years 2010 and 2016 are included in the subset. Additionally, we only consider data of the primary tumor of the first occurrence of breast cancer in those patients. Though quite rare, a patient may develop a new primary later in life or during later stages of the treatment.

4.1. Breast cancer in SEER database

Even though the SEER database contains clinical data about various types of cancer diagnosed in patients in the US, not all types of cancer have specific variables for their pathophysiology in the database. Patients diagnosed with breast cancer, on the contrary, have fields specifically created in the database to register information about their path of treatment, which is not surprising considering that breast cancer is the most common cancer in women and the leading cause of death in women among all types of cancer. Besides, its incidence ratio has not been stable, and has increased much faster than population growth in the US between the years 1970 and 2014 (242% versus 56.8%) [30].

Given these contexts, and the amount of data, both in number of independent samples and number of variables, there are already numerous studies on breast cancer with the SEER database. Throughout the years, many claims have been made based on correlations, some with causal interpretation, but without the required causal methodology.

4.2. Network of dataset of 400,000 Breast cancer patients

As a large clinical dataset, SEER is of particular interest for us to test the performance of MIIC concerning the scalability and the ability to deal with heterogeneous dataset. Since a large part of the relations between variables in the SEER database has been inspected in previous studies by researchers and clinicians, the validity of the causal relations recovered by MIIC can be assessed by those expert knowledge, as well as by intuition, which makes SEER a good real-life benchmark.

Therefore, as part of an ongoing project, MIIC is applied on the pre-processed dataset of 396,179 patients with 51 variables. We present here the network reconstructed by MIIC with the skeleton-consistent option (section 2.4) and the conservative orientation rule (section 3.5.2), as shown in fig. 4.1. The whole reconstruction took around 24 hours.

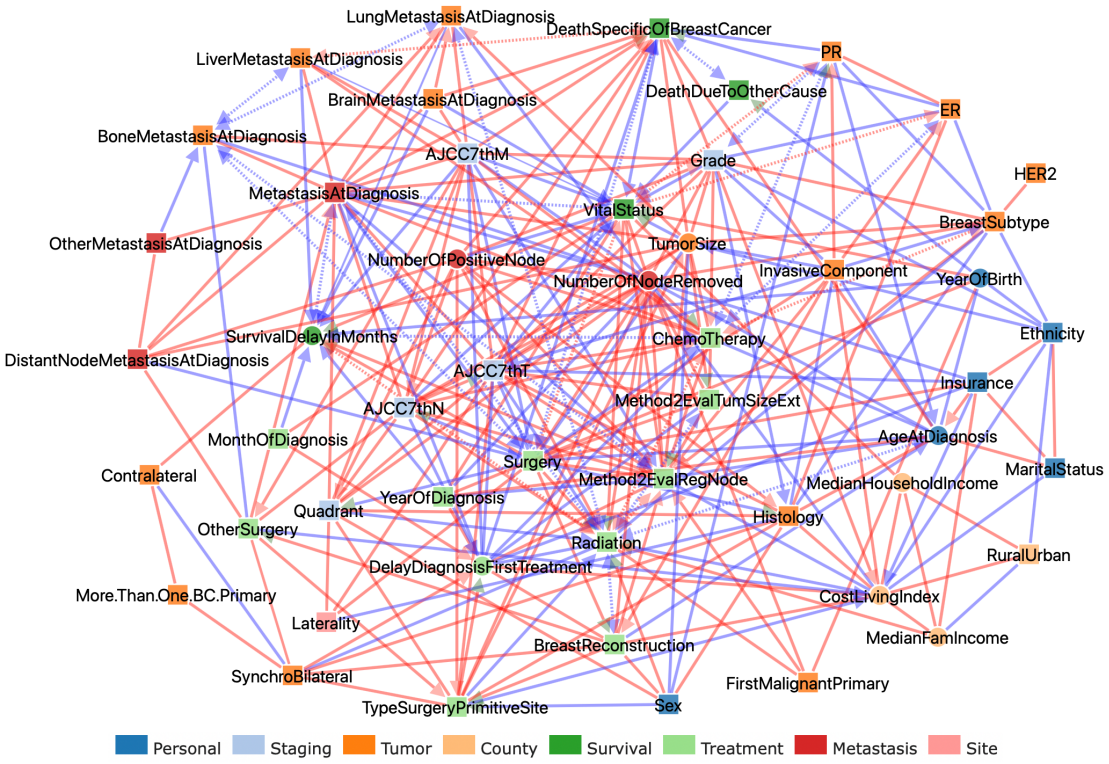

Figure 4.1. – Clinical network reconstructed by iMIIC from 396,179 breast cancer patients in the SEER database.

With the reconstructed network, we gathered all causal predictions, including both genuine causal and putative causal edges (section 3.5.1), and analysed their validity based on intuition and expert knowledge retrieved from literature. It is found that 90% of them

appear to be "correct" (evidence from literature) while 98% can be seen as "plausible" (no direct evidence from literature but are intuitive).

In the following, we first study the robustness of the reconstructed network by a sub-sampling analysis, then we select a group of variables of interest, and discuss some of the causal relations related to those variables that either are recovered by MIIC, or are expected but missed in the network.

4.3. Sub-sampling analysis

To validate the reconstructed network, 9 subsamples (3 of 100k patients, 3 of 10k patients and 3 of 1k patients) are drawn independently and randomly from the full dataset. For each dataset, three networks are reconstructed by MIIC using the option consistent={no, skeleton, orientation}, respectively, with the rest of parameters identical to that used for the full dataset.

For the 27 obtained networks, it is observed that for each dataset, there is very little difference between the skeleton-consistent network and the network that does not require any sep-set consistency. Of all 9 subsamples, only one 1K subsample reports a difference, and the observed difference is of only one edge. This means that for all other subsamples, the network reconstructed without consistency requirements is already sep-set consistent with respect to the skeleton. Therefore, in the following, it is reasonable to focus only on the skeleton-consistent and the orientation-consistent networks, which sum up to a total of 18 networks.

4.3.1. Skeleton

Neglecting all orientations, the 18 obtained skeletons are divided into 6 groups by different sample size and consistency option, such that each group contains 3 networks reconstructed from different sub-samplings of the same size and with the same consistency option. For each group an Euler diagram is constructed based on common edges among the three networks, as shown in fig. 4.2.

In each diagram of fig. 4.2, triply layered area represents edges that are shared by all three networks, doubly layered areas represent edges that are exclusively shared by two of the networks, and singly layered areas represent edges that are exclusive to each network. Inside each area, the sum represents the number of edges, the left summand represents the number of edges that are shared with the full dataset, and the right summand represents the number of edges absent in the full dataset. Highlighted by red contour is the network with the median number of total edges among the three networks, all percentages are

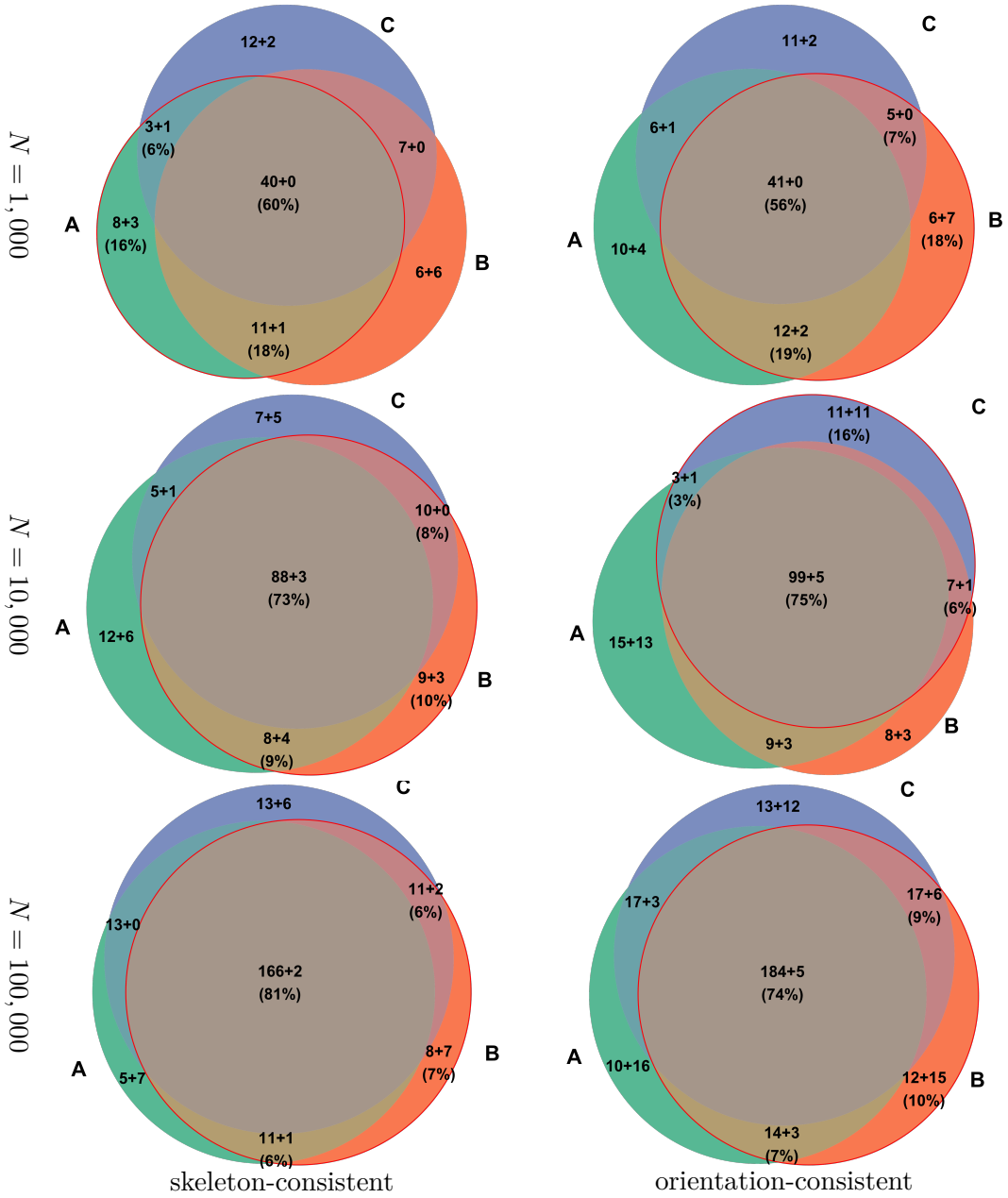

Figure 4.2. – Euler diagrams for networks of subsamples with different sample size (N) and separating set consistency option.

with respect to the median network.

At the top row left of fig. 4.2, among the three skeleton-consistent networks of 1000 samples, the median network contains 67 edges, 60% of which are shared by the median network and the other two networks. There are only 5 edges conflicting with the full dataset (conflicts are edges that are absent in the full dataset, and are represented by the right summand in each area). Similar statistics are found for orientation-consistent networks of the same sample size (top row right). Then at the mid row, when N is increased to 10,000, both skeleton-consistent and orientation-consistent diagrams improve significantly in terms of common-edge percentage, while the number of conflicts remains relatively low in the triply and doubly layered area when compared with the full dataset. However, the number of conflicts in some singly layered areas does increase significantly. At the bottom row, when N is increased to 100,000, the common-edge percentage slightly increases for skeleton-consistent diagram, while it slightly decreases for orientation-consistent diagram. As for the number of conflicts with the full dataset, when compared to the $N = 10,000$ case, it stays at the same level in the triply and doubly layered areas for skeleton-consistent diagram, and slightly increases for orientation-consistent diagram. By contrast, in the singly layered areas, it continues to grow significantly.

The relatively low common-edge percentage at smaller sample size is expected. On the one hand, information contained in a small sample is limited, thus fewer edges will remain in the final network. On the other hand, smaller random samplings from a large dataset tend to have more heterogeneity, that is, each sub-sampling is more likely to contain information from different parts of the full dataset. As the sampling size increases, the heterogeneity among sub-samplings decreases, while more edges will remain in the final network, which results in the increasing common-edge percentage.

When it comes to conflicts with the full dataset, as N increases, singly layered areas reveal a development vastly different from doubly and triply layered areas. The conflict number staying relatively low in multi-layered areas implies that edges that are shared by at least two networks are highly likely to be also present in the full dataset. This is to be compared with the increasing number of conflicts in singly layered areas, which implies that edges that are unique to only one network are probably also not shared by the full dataset. In other words, those edges are likely due to noise or sub-sampling bias. Together, the different behaviors in singly and multiply layered areas validate those shared edges.

In summary, two features are of major interest from the sub-sampling diagrams in fig. 4.2: the common-edge percentage and the number of conflicts with full data set. The increasing common-edge percentage with increasing sample size illustrates how consistently MIIC manages to recover these edges. The different behavior of the number

of conflicts in singly and multiply layered areas illustrates the robustness of the method and the reconstructed network.

4.3.2. Oriented edges

In principle, the same method based on Euler diagrams can be applied to directed graphs. However, since MIIC assigns a probability to each end point of each edge and orients the edge according to these probabilities, Euler diagrams will unfortunately hide all information of probability. For example, an edge with probabilities $\{0, 0.5\}$ (tail – unsure) in one network will appear to be conflicting with the same edge in another network with edge probabilities $\{0, 1\}$ (tail – head), even though probability-wise they are not really conflicting: both networks confirm the tail, but only one of them is able to confirm the head. This is to be compared with the case when a real conflict at the probability level occurs, but at the level of Euler diagram we are unable to distinguish it from the first case. By consequence, for oriented edges, we will mainly focus on the probability conflicts, both among sub-samplings and between sub-sampling and full dataset.

Out of the 18 networks, we look at only those from 100K sub-samplings, as it is observed that networks from smaller sub-samplings didn't add many new edges as compared to the 100K ones. We start by the three skeleton-consistent networks, represented by the Euler diagram at mid row left in fig. 4.2. Then, we select all edges that are either shared by at least two of the sub-samplings, or by one of the sub-samplings and the full dataset. In other words, in the Euler diagram, we pick all edges except those that belong to the right summand of each singly layered area, since, as discussed in the previous section, they are more susceptible to noise or sampling bias. For each edge, we first check if it has the same orientations among all networks in which it appears, if it is not the case, we then check if the difference in orientation is due to an actual conflict of orientation probabilities or simply due to a probability that is undetermined in certain networks.

The result of the analysis is as follows: for the 100K skeleton-consistent sub-sampling networks, there are 232 shared edges, including 60 edges with different orientations among sub-samplings (that is, not considering the full-dataset network), but only 27 (12%) of which have conflicting orientation probabilities, as the orientations of the remaining 33 edges could only be assessed in certain sub-sampling(s) and remained undetermined in the other(s). Then, among the remaining 172 edges that have the same orientations in all three sub-sampling networks, 44 edges exhibit different orientations in the full-dataset network, but only 17 (10%) of which have conflicting orientation probabilities, as the orientations of the remaining 27 edges remained undetermined in either the sub-sampling

networks or the full-dataset network. Similarly, for the 100K orientation-consistent sub-sampling networks, there are 284 shared edges, 82 edges have different orientations among sub-samplings, but only 36 (13%) of which have conflicting orientation probabilities, as the orientations of the remaining 46 edges could only be assessed in certain sub-sampling(s) and remained undetermined in the other(s). Then among the remaining 202 edges that have the same orientations in all three sub-sampling networks, 45 edges have different orientations in the full-dataset network, but only 19 (9%) have conflicting orientation probabilities, as the orientations of the remaining 26 edges remained undetermined in either the sub-sampling networks or the full-dataset network.

4.4. Analysis on selected variables

Survival

In fig. 4.1, the variable group Survival (deep green) contains the variable Vital Status which represents the vital status (dead or alive) of each patient at the end of 2016, and two additional variables specifying the cause of death, either from breast cancer or from any other cause. Known factors affecting the death due to breast cancer are correctly recovered by MIIC, such as metastasis at diagnosis causing death due to breast cancer and ER status reducing the risk of death due to breastcancer (presumably mediated by the variable hormonotherapy which is not recorded in the available SEER dataset). Similarly, MIIC correctly infers the causal relations between year of birth and death due to other cause, year of birth and vital status, tumor size and vital status. Also note that the death of patients (irrespective of its cause) is predicted to lead to a reduction in their survival delays, as expected. Similarly, though less intuitively, early death within the first few months after diagnosis is predicted to prevent radiotherapy (Radiation) for some patients who might have otherwise received the treatment had they lived longer. This short term causal effect between vital status and radiotherapy is also robustly inferred in all three 100k patient sub-sampling networks (section 4.3), suggesting that this short term causal effect outweights the causally reversed beneficial effect of radiotherapy.

Histology

MIIC infers a number of direct associations between the variable Histology and other variables, such as age at diagnosis (in agreement with early report in [13]) and synchro bilateral primaries (detected within 6 months of the first diagnosis), with the latter almost twice more likely to occur when lobular carcinoma is present.

By contrast, no significant association is found with contralateral primary tumors (detected more than 6 months after diagnosis). MIIC also uncovers a somewhat unexpected genuine causal link from Surgery to Histology. This reflects, however, the fact that histological types are frequently refined, after surgery, by the pathologist based on the surgical specimen. In particular, only 36% to 50% of patients have undergone surgery when the generic histological types *adenocarcinoma, NOS* or *carcinoma, NOS* are reported, whereas more than 90% of the patients have undergone surgery when the reported histological type includes specific tissues, such as ducts or lobules.

Socio-economic county variables

Figure 4.1 includes four socio-economic variables pertaining to the county of residence of each patient: Median Family Income, Median Household Income, Cost Living Index and the Rural-Urban population size of each county. These four socio-economic variables actually form a fully connected subgraph (a clique) indicating their strong interdependencies. Interestingly, Vital Status is only connected to this county variable clique through Median Household Income, which is consistent with earlier reports on the association between life expectancy and incomes [8]. By contrast, all other patient specific variables connected to the county clique (such as tumor grade, radiotherapy, breast reconstruction, insurance, etc.) have in fact at least one link with the cost of living index, which highlights the integration of the healthcare system into the global economy. In particular, there is a direct association between higher cost of living and more favorable breast cancer prognosis (e.g. lower grade and fewer invasive components at diagnosis). This is presumably due to better preventive health care including easier access to breast cancer screening centers and more comprehensive insurance coverage. Yet, there are also strong disparities between counties, as manifested by the opposite associations of Insurance and Radiation with Median Family Income versus Cost of Living Index. These intriguing findings can be traced back to Los Angeles (LA) county, amounting to about 10% of the whole dataset, which presents a lower than average median family income (29-38% percentile range) despite a higher than average cost of living index (58-67% percentile range) . This must have led to an exacerbated financial burden for many of the 39, 089 breast cancer patients diagnosed in LA county between the years 2010 and 2016. Although 18% of these patients benefited from medicaid insurance (as compared to 10% in the whole dataset), many had to opt for affordable but limited private insurance including significant co-payment policies or even to become uninsured, especially before the application of the Affordable Care Act in

January 2014 (3.4% uninsured in 2013 against 1.5% in 2014). As a result, many LA patients appear to have renounced to undergo expensive treatments. In particular, only 30% of the patients underwent radiotherapy in LA, as compared to 50% of patients in the overall dataset. Moreover, an estimated 7% of LA patients even appear to have dropped off therapy or moved to a different county not included in SEER database (against 1.5% nationwide, excluding LA county). This corresponds to the fraction of patients having had their last medical contact less than one year after diagnosis and more than one year before the end of this study in December 2016.

5. Conclusion and perspectives

The principle of causal analysis has two major aspects, causal inference and causal structure learning. Causal inference concerns the analysis of causal networks. Causal networks reconstructed from biological or clinical data allow for analysis of possible causal relationships between different treatments, environmental and geopolitical factors and the stage or development of a disease, or between different gene mutations and the expression of a certain protein. They can help clinicians in the treatment of diseases, or biologists in the design of experiments. Causal structure learning concerns methods for reconstructing causal networks. Information theoretic methods have become ubiquitous for quantitatively analyzing these information-rich data. In this thesis, we focused on methods for reconstructing causal networks from purely observational data, in the absence of a randomized controlled trial or interventional study.

Part I

In the first part, we presented and discussed a simple modification of a classical method of constraint-based causal structure learning, the so-called PC algorithm, by enforcing separating set consistency on the final reconstructed graph. This is achieved by repeating the constraint-based causal structure learning scheme iteratively, while searching for separation sets that are consistent with the graph obtained in the previous iteration. Ensuring the consistency of the separation sets can be done at a limited complexity cost, by using the biconnected component decomposition of the graph skeletons. It also significantly improves the sensitivity of constraint-based methods while maintaining good overall structure learning performance. Last but not least, the assurance of consistency of the sepsets improves the interpretability of constraint-based models for real applications.

From a methodological point of view, the self-correcting nature of the proposed iterative approach determines that it cannot intend huge improvement on the overall performance of any particular method. Instead, it shines by its simplicity and maximal compatibility as a plug-in with any constraint-based method that follows the skeleton-orientation procedure. When the method itself is already flawless, the plug-in does not have any impact on the final graph which is already separating set consistent. When errors are

present, the separating consistency is enforced at the cost of loss of structural details of the final graph, which could be either a conditional independence relation (skeleton) or a causal implication (orientation). In this perspective, the plug-in may be used to evaluate the performance of a certain method by comparing the final graphs with and without the constraint on the separating set consistency. A significant difference between the two graphs may indicate the lack of reliability of the statistical conditional independence tests.

There are several limitations of the proposed iterative approach.

Extra tests and background knowledge

When the considered method is heavily defective, either because of lack of samples or because of theoretical limitations, the gain by enforcing separating set will be outweighed by the loss of structural details, as discussed in section 2.5. The proposed consensus graph can partly alleviate the problem, but it is only useful when the number of graphs in the consistent cycle is large enough to make the consensus statistically meaningful. When such is not the case, it may worth considering combining the consistent cycle with either extra statistical tests or background knowledge, so as to make more meaningful decisions while keeping a moderate computational cost.

Convergence study

Consider algorithms 4 and 5. From the theoretical point of view, although the number of iterations necessary before reaching the stop condition is guaranteed to be finite by the deterministic nature of the PC or PC-stable algorithm, it will not be the case were any stochasticity to be included in a PC-derived algorithm. It would be worth digging around in this direction to see if, under certain assumptions, any estimation on the convergence rate of the iterations could be given.

Latent variables

As brought up in section 2.4.1, definition 5a must assume causal sufficiency, otherwise the more general definition 5b should be used. We are curious to know if any computational-complexity-friendly algorithm exists for the check of the non-descendant condition in definition 5b.

Part II

The second part concerned the modification of the MIIC method developed by our team. MIIC is a hybrid method that is constraint-based but handles different constraints based on the information-based score. MIIC significantly improves the reconstruction

of causal networks from purely observational data. Yet, a substantial loss of accuracy remained between skeleton and directed graph predictions for small data sets. We proposed and implemented a few modifications to MIIC, which emphasize the improving of interpretability of the final reconstructed network. One of them is the conservative estimation of orientation based on a general regularized mutual information supremum principle for finite data sets, called conservative MIIC. This modification is shown to significantly improve the reliability of predicted orientations, for samples of all sizes, with only a small loss of sensitivity compared to the original MIIC orientation rules. Conservative MIIC is particularly attractive for improving the reliability of causal discovery for real-life observational data applications. In the same sections, we addressed a number of important improvements to significantly increase its causal analysis performance on large-scale synthetic and real-life datasets. In summary, these modifications

- quantitatively improve confidence in edge orientations;
- distinguish "genuine" from "putative" causal relationships;
- introduce contextual and stochastic variables
- strengthen the global structural consistencies of the reconstructed networks;
- allow for scalability to very large datasets.

Some implementation details are also given at the end of this part, in particular regarding the refactoring of the MIIC codebase, which is an open-source project available on CRAN and on GitHub. Finally, in the third section, we discussed briefly the application of MIIC to the reconstruction of an interpretable clinical network from the analysis of medical records of nearly 400,000 breast cancer patients from the Surveillance, Epidemiology, and End Results (SEER) database. The analysis on the networks reconstructed from sub-samplings shows the robustness of the reconstructed network from the full dataset. The analysis on certain variables demonstrates the interest of using MIIC to help with the causal analysis in real-life clinical studies.

Regarding the two aspects of causal analysis, MIIC is distinguished from many other methods by its nonparametric structure learning. Although influenced by the ideas of maximum likelihood, MIIC does not aim to provide a specific parametrization for the learned structure, which can be seen as both a disadvantage and an advantage over score-based methods that typically learn the best parametrization alongside the structure. On the one hand, the parametrization is often essential for practical causal inference based on a given structure, as represented by the conditional probability $\Pr(Y = y \mid X = x)$ of a certain outcome $Y = y$ given an observation of its causes $X = x$. On the other hand, the learning of parameters often necessitates prior knowledge or additional assumptions on the models and distributions to be learned, which is sometimes unfeasible

and is often approximated for real-life applications. Eventually, such prior knowledge and assumptions may have an unknown impact on the learning of structure itself. Based on the notion of conditional mutual information, MIIC does not require any prior knowledge or assumption on a given dataset in order to recover the underlying causal structure. On the downside, MIIC stops at the structural level, giving a result that is not immediately usable for quantitative inference. Nontheless, chapter 4 shows the usefulness of MIIC in the reliable recovery of known causal relations and in the finding of potential novel causal relations. Moreover, the nonparametric structure learned by MIIC can be further used by other methods for separate quantitative parameter learning, effectively prevent any unintended impact of prior knowledge or assumption of the parametrization on the structure itself.

Bibliography

[1]S. Affeldt and H. Isambert, « Robust reconstruction of causal graphical models based on conditional 2-point and 3-point information », in Proceedings of the thirty-first conference on uncertainty in artificial intelligence, UAI 2015 (2015), pp. 42–51.

[2]S. Affeldt, L. Verny, and H. Isambert, « 3off2: a network reconstruction algorithm based on 2-point and 3-point information statistics », BMC Bioinformatics **17**, 10.1186/s12859-015-0856-x (2016).

[3]H. Akaike, « Information theory and an extension of the maximum likelihood principle. Proceedings of the 2nd international symposium on information theory », in *Second international symposium on information theory* (1973).

[4]A. R. Ali and T. S. Richardson, « Markov equivalence classes for maximal ancestral graphs », in UAI '02, proceedings of the 18th conference in uncertainty in artificial intelligence, university of alberta, edmonton, alberta, canada, august 1-4, 2002, edited by A. Darwiche and N. Friedman (2002), pp. 1–9.

[5]A. Barron, J. Rissanen, and B. Yu, « The Minimum Description Length Principle in Coding and Modeling », IEEE Transactions on Information Theory **44**, 2743–2760 (1998).

[6]D. Bernstein, B. Saeed, C. Squires, and C. Uhler, « Ordering-based causal structure learning in the presence of latent variables », in Proceedings of the twenty third international conference on artificial intelligence and statistics, Vol. 108, edited by S. Chiappa and R. Calandra, Proceedings of Machine Learning Research (Aug. 2020), pp. 4098–4108.

[7]V. Cabeli, L. Verny, N. Sella, G. Uguzzoni, M. Verny, and H. Isambert, « Learning clinical networks from medical records based on information estimates in mixed-type data », PLOS Computational Biology **16**, e1007866 (2020).

[8]R. Chetty, M. Stepner, S. Abraham, S. Lin, B. Scuderi, N. Turner, A. Bergeron, and D. Cutler, « The association between income and life expectancy in the United States, 2001-2014 », JAMA - Journal of the American Medical Association **315**, 1750–1766 (2016).

[9]D. M. Chickering, « Optimal structure identification with greedy search », in Journal of machine learning research, Vol. 3, 3 (2002), pp. 507–554.

[10]D. Colombo and M. H. Maathuis, « Order-independent constraint-based causal structure learning », Journal of Machine Learning Research **15**, 3741–3782 (2014).

[11]D. Colombo, M. H. Maathuis, M. Kalisch, and T. S. Richardson, « Learning high-dimensional directed acyclic graphs with latent and selection variables », Ann. Statist. **40**, 294–321 (2012).

[12]T. M. Cover and J. A. Thomas, « Differential entropy », in *Elements of information theory* (John Wiley & Sons, Ltd, 2005) Chap. 8, pp. 243–259.

[13]C. J. Fisher, M. K. Egan, P. Smith, K. Wicks, R. R. Millis, and I. S. Fentiman, « Histopathology of breast cancer in relation to age », British Journal of Cancer **75**, 593–596 (1997).

[14]S. G, « "Estimating the Dimension of a Model." », The Annals of Statistics **6**, 461–464 (1978).

[15]E. J. Hannan and B. G. Quinn, « The Determination of the Order of an Autoregression », Journal of the Royal Statistical Society: Series B (Methodological) **41**, 190–195 (1979).

[16]C. Heinze-Deml, J. Peters, and N. Meinshausen, « Invariant causal prediction for nonlinear models », Journal of Causal Inference **6**, `10.1515/jci-2017-0016` (2018).

[17]P. W. Holland, « Statistics and causal inference », Journal of the American Statistical Association **81**, `10.1080/01621459.1986.10478354` (1986).

[18]K. T. Hwang, J. Kim, J. Jung, J. H. Chang, Y. J. Chai, S. W. Oh, S. Oh, Y. A. Kim, S. B. Park, and K. R. Hwang, « Impact of breast cancer subtypes on prognosis of women with operable invasive breast cancer: A Population-based Study Using SEER Database », Clinical Cancer Research **25**, 1970–1979 (2019).

[19]A. Hyttinen, P. O. Hoyer, F. Eberhardt, and M. Järvisalo, « Discovering cyclic causal models with latent variables: a general sat-based procedure », in Proceedings of the twenty-ninth conference on uncertainty in artificial intelligence, UAI'13 (2013), pp. 301–310.

[20]M. Kalisch, M. Mächler, D. Colombo, M. H. Maathuis, and P. Bühlmann, « Causal inference using graphical models with the R package pcalg », J. Stat. Softw. **47**, 1–26 (2012).

[21]P. Kontkanen and P. Myllymäki, « A linear-time algorithm for computing the multinomial stochastic complexity », Information Processing Letters **103**, 227–233 (2007).

[22]H. Li, V. Cabeli, N. Sella, and H. Isambert, « Constraint-based causal structure learning with consistent separating sets », in Advances in neural information processing systems, Vol. 32 (2019).

[23]P. Nandy, A. Hauser, and M. H. Maathuis, « High-dimensional consistency in score-based and hybrid structure learning », Annals of Statistics **46**, 10.1214/17-AOS1654 (2018).

[24]R. Nishii, « Maximum likelihood principle and model selection when the true model is unspecified », Journal of Multivariate Analysis **27**, 392–403 (1988).

[25]J. Pearl, *Causality: Models, reasoning, and inference, second edition*, 2nd (Cambridge University Press, Cambridge, 2009).

[26]J. Ramsey, P. Spirtes, and J. Zhang, « Adjacency-faithfulness and conservative causal inference », in Proceedings of the 22nd conference on uncertainty in artificial intelligence, UAI (2006), pp. 401–408.

[27]J. Rissanen, « Modeling by shortest data description », Automatica **14**, 465–471 (1978).

[28]J. Rissanen and I. Tabus, « Kolmogorov's Structure Function in MDL Theory and Lossy Data Compression », in *Advances in minimum description length*, 1st (The MIT Press, 2005) Chap. 10, pp. 245–262.

[29]T. Roos, T. Silander, P. Kontkanen, and P. Myllymäki, « Bayesian network structure learning using factorized NML universal models », in 2008 information theory and applications workshop - conference proceedings, ita (2008), pp. 272–276.

[30]A. P. Schneider, C. M. Zainer, C. K. Kubat, N. K. Mullen, and A. K. Windisch, *The breast cancer epidemic: 10 facts*, 2014.

[31]M. Scutari, « Learning Bayesian Networks with the bnlearn R Package », Journal of Statistical Software **35**, 1–22 (2010).

[32]Y. M. Shtar'kov, « UNIVERSAL SEQUENTIAL CODING OF SINGLE MESSAGES. », Problems of information transmission **23**, 175–186 (1987).

[33]P. Spirtes and C. Glymour, « An algorithm for fast recovery of sparse causal graphs. », Social Science Computer Review **9**, 62–72 (1991).

[34]P. Spirtes, C. Glymour, and R. Scheines, *Causation, prediction, and search*, 2nd (The MIT Press, 2000).

[35]E. Talvitie, « Self-correcting models for model-based reinforcement learning », in 31st aaai conference on artificial intelligence, aaai 2017 (2017).

[36]T. Verma and J. Pearl, « Equivalence and synthesis of causal models », in Proceedings of the sixth conference on uncertainty in artificial intelligence (1991).

[37]L. Verny, N. Sella, S. Affeldt, P. P. Singh, and H. Isambert, « Learning causal networks with latent variables from multivariate information in genomic data », PLoS Computational Biology **13**, e1005662 (2017).

[38]R. W. Wedderburn, « Quasi-likelihood functions, generalized linear models, and the gauss-newton method », Biometrika **61**, 439–447 (1974).

[39]J. Zhang, « On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. », Artif. Intell. **172**, 1873–1896 (2008).

[40]K. Zhang, J. Peters, D. Janzing, and B. Schölkopf, « Kernel-based conditional independence test and application in causal discovery », in Proceedings of the 27th conference on uncertainty in artificial intelligence, uai 2011 (2011), pp. 804–813.

A. Résumé Substantiel

Le principe de l'analyse causale comprend deux aspects majeurs, l'inférence causale et l'apprentissage de la structure causale. L'inférence causale concerne l'analyse des réseaux causaux. Les réseaux causaux reconstruits à partir de données biologiques ou cliniques permettent d'analyser les relations causales possibles entre différents traitements, facteurs environnementaux et géopolitiques et le stade ou le développement d'une maladie, ou entre différentes mutations de gènes et l'expression d'une certaine protéine. Elles peuvent aider les cliniciens dans le traitement des maladies, ou les biologistes dans la conception d'expériences. L'apprentissage des structures causales concerne les méthodes de reconstruction des réseaux causaux. Les méthodes de la théorie de l'information sont devenues omniprésentes pour analyser quantitativement ces données riches en informations.

Dans cette thèse, nous nous concentrons sur les méthodes de reconstruction de réseaux causaux à partir de données purement observationnelles, en l'absence d'essai randomisé contrôlé ou d'étude interventionnelle. La thèse peut être divisée en trois parties, qui s'appuient sur des manuscrits publiés, en cours de révision pour publication ou en préparation.

La première partie est méthodologique, elle concerne la modification d'une méthode classique d'apprentissage de structures causales, dite méthode basée sur des contraintes, en imposant l'interprétabilité du graphe final reconstruit. On y parvient en répétant le schéma d'apprentissage de la structure causale basé sur les contraintes, de manière itérative, tout en recherchant les ensembles de séparation qui sont cohérents avec le graphe obtenu à l'itération précédente. Assurer la cohérence des ensembles de séparation peut être fait à un coût de complexité limité, par l'utilisation de la décomposition en composantes biconnexes des squelettes de graphes, et il s'avère que cela augmente leur validité en termes de d-séparation réelle. Elle améliore également de manière significative la sensibilité des méthodes basées sur les contraintes tout en conservant une bonne performance globale d'apprentissage de la structure. Enfin et surtout, l'assurance de la cohérence des sepsets améliore l'interprétabilité des modèles basés sur les contraintes pour les applications réelles.

La deuxième partie concerne la modification de la méthode MIIC développée par notre

équipe. MIIC est une méthode hybride qui est basée sur des contraintes mais qui traite différentes contraintes en fonction du score basé sur l'information. MIIC améliore de manière significative la reconstruction de réseaux causaux à partir de données purement observationnelles. Pourtant, une perte substantielle de précision est restée entre les prédictions de squelette et de graphe orienté pour les petits ensembles de données. Nous nous concentrons dans cette partie sur l'amélioration de l'interprétabilité et de l'extensibilité de la méthode MIIC, et nous proposons la nouvelle méthode MIIC interprétable (iMIIC) qui apporte un certain nombre d'améliorations importantes pour augmenter considérablement ses performances d'analyse causale sur des jeux de données synthétiques et réels à grande échelle. En résumé, la méthode iMIIC (i) améliore quantitativement la confiance dans les orientations des arêtes, (ii) distingue les relations *causales authentiques* des relations *causales putatives*, (iii) introduit des variables contextuelles et stochastiques, (iv) renforce les cohérences structurelles globales des réseaux reconstruits et, enfin, (v) permet l'extensibilité à de très grands ensembles de données.

À la fin de cette partie, il y a une section sur les détails de l'implémentation, en particulier le refactoring, du codebase de MIIC, qui est un projet open-source disponible sur CRAN et sur GitHub.

Enfin, dans la troisième partie, nous discutons de l'application de iMIIC à la reconstruction d'un réseau clinique interprétable à partir de l'analyse des dossiers médicaux de près de 400,000 patientes atteintes d'un cancer du sein provenant de la base de données Surveillance, Epidemiology, and End Results (SEER).

Apprentissage de structure causale basé sur des contraintes avec des ensembles de séparation cohérents

Les méthodes d'apprentissage de la structure causale peuvent être divisées en deux catégories : les méthodes basées sur des scores et les méthodes basées sur des contraintes. Chaque catégorie a ses avantages et ses limites. Nous nous concentrons ici sur les méthodes basées sur les contraintes. Par rapport aux méthodes basées sur les scores, elles ont l'avantage de ne pas être limitées à la classe des (CP)DAGs, mais sont limitées par leur manque de robustesse par rapport au bruit d'échantillonnage pour les jeux de données finis.

Cette partie concerne, plus spécifiquement, l'incohérence des ensembles de séparation utilisés pour éliminer les arêtes dispensables, de manière itérative, sur la base de tests d'indépendance conditionnelle. Cette incohérence est due au fait que certains ensembles de séparation peuvent ne plus être compatibles avec le graphe final, s'ils n'étaient pas

déjà incompatibles avec le squelette actuel, lors du test d'indépendance conditionnelle pendant le processus d'élagage. De telles incohérences peuvent être considérées comme un défaut majeur des méthodes basées sur les contraintes, car la motivation première pour apprendre et visualiser des modèles graphiques est sans doute de pouvoir lire les indépendances conditionnelles directement à partir de la structure du graphe.

La procédure générale des méthodes basées sur les contraintes est présentée dans la fig. A.1. Etant donné un ensemble de données sur un ensemble de variables (sommets), on part d'un graphe complet G. Par une série de tests statistiques sur chaque paire de variables, toutes les arêtes $X - Y$ sont retirées si une indépendance conditionnelle et un ensemble de séparation C peuvent être trouvés, c'est-à-dire si $X \perp\!\!\!\perp Y \mid C$ (étape 1). Le graphe non orienté résultant est appelé le squelette de G. Les v-structures sont ensuite identifiées, $X \rightarrow Z \leftarrow Y$, si $X \perp\!\!\!\perp Y \mid C$ et $Z \notin C$ (étape 2). Des hypothèses supplémentaires (par exemple, l'acyclicité) permettent la propagation des orientations des v-structures à certaines des arêtes non dirigées restantes (étape 3).

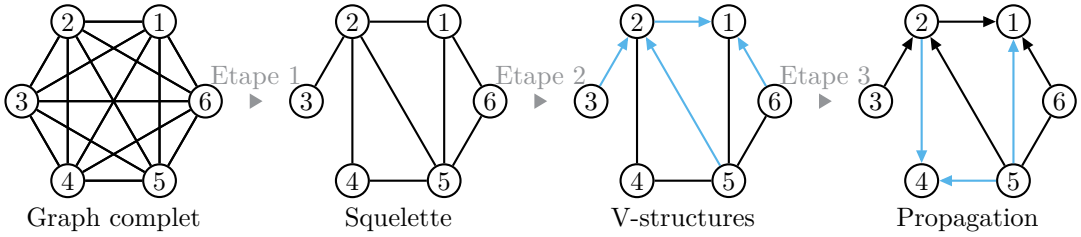

FIGURE A.1. – Procédure générale des méthodes basées sur les contraintes

L'incohérence des ensembles de séparation peut prendre différentes formes, concernant soit le squelette (type I), soit le graphe final (partiellement) orienté (type II), comme l'illustre la fig. A.2.

FIGURE A.2. – Incohérence des ensembles de séparation.

Une incohérence de type I correspond à une relation d'indépendance conditionnelle telle que $2 \perp\!\!\!\perp 6 \mid 3, 5$ dans la fig. A.2, pour laquelle il n'existe aucun chemin entre le sommet 2 et 6 qui passe par 3. Ce type d'incohérence concerne souvent des arêtes évaluées au début du processus d'élagage, lorsque peu d'arêtes ont été supprimées, et donc que l'espace

combinatoire des ensembles de séparation possibles est encore grand. En particulier, l'arête 3 — 6, qui est finalement supprimée dans le graphe final, peut encore exister lorsque l'arête 2 — 6 est considérée.

Une incohérence de type II est un type différent d'incompatibilité provenant de l'orientation du squelette. Elle se produit, en particulier, lorsque dans l'ensemble de séparation d'une relation d'indépendance conditionnelle, il y a au moins un descendant commun de la paire d'intérêt dans le graphe final, par exemple $3 \perp\!\!\!\perp 6 \mid 1$ dans la fig. A.2 à droite. Comme elle découle de l'orientation des arêtes (étapes 2&3), l'origine des incohérences de type II est généralement plus complexe et résulte d'une cascade d'erreurs à la fois dans les tests d'indépendance conditionnelle et dans l'orientation.

Pour résoudre ces incohérences, nous introduisons les définitions suivantes sur l'ensemble de séparation cohérent :

Definition 15 (Ensembles cohérent)**.** Soit $G(V, E)$ un graph orienté et $\{ X; Y; Z \} \subseteq V$ un ensemble de sommets. L'ensemble de sommets cohérents par rapport à (X, Y) et au squelette de G est

$$\text{Conskel}(X, Y \mid G) = \{ Z \in \text{adj}(X) \setminus \{ Y \} \mid \text{il existe au moins un chemin } \gamma^{Z}_{X;Y} \text{ dans } G. \}$$

L'ensemble de sommets cohérents par rapport à (X, Y) et à G est

$$\text{Consist}(X, Y \mid G) = \{ Z \in \text{Conskel}(X, Y \mid G) \mid Z \text{ n'est pas un enfant de } X. \}$$

A l'aide de ces définitions, nous proposons un algorithme itératif avec deux variantes, comme l'illustre la fig. A.3.

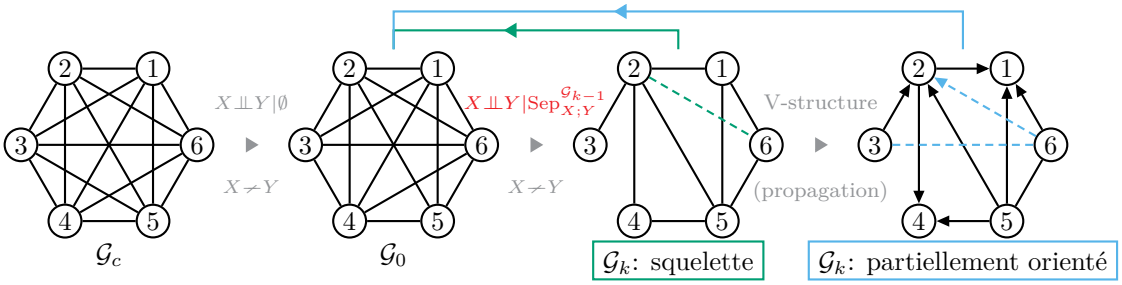

FIGURE A.3. – Illustration de la procédure itérative pour apprendre des modèles graphiques avec des ensembles de séparation cohérents. Les arêtes en pointillés marquent la différence entre deux itérations successives.

L'idée est de toujours partir d'un graphe complet, puis de supprimer tous les arêtes qui correspondent à une indépendance marginale. A la fin de cette étape, on obtient G_0. En

partant de G_0, les deux variantes divergent. Pour la variante cohérente par rapport au squelette, on répète la reconstruction du squelette, de manière itérative, tout en cherchant des ensembles séparateurs cohérents avec le squelette obtenu à l'itération précédente, jusqu'à atteindre un cycle limite de graphes successifs. Alors l'union des squelettes du cycle est prise comme le squelette cohérent et est ensuite orientée comme dans les étapes 2 et 3 de la fig. A.1. Pour la variante cohérente par rapport au graphe orienté, on répète plutôt la reconstruction du graphe orienté, et l'union des graphes orientés dans le cycle est le graphe final orienté et cohérent.

Pour limiter la complexité temporelle de l'algorithme, nous proposons un algorithme efficace pour trouver Conskel$(X, Y \mid G)$, à l'aide de l'analyse des composantes biconnexes basée sur la décomposition de block-cut tree, qui peut être effectuée en temps linéaire avec un parcours en profondeur.

Theorem 16. *Soit $G(V, E)$ un graphe non orienté, $H(V_H, E_H) \subseteq \mathcal{G}$ une composante biconnexe de G, $\{ X, Y \} \subseteq V_H$ une paire de sommets, et $Z \in V_G$ un troisième sommet. Il existe un chemin $\gamma^Z_{X;Y}$ si et seulement si $Z \in V_H$.*

Le theorem 16 transforme la recherche d'un ensemble séparateur cohérent en une décomposition de composantes biconnexes. Plus précisément, si X et Y appartiennent à la même composante biconnexe, alors l'ensemble de séparation cohérent est l'ensemble des sommets de la composante biconnexe moins X et Y. Si X et Y sont dans des composantes biconnexes différentes, nous pouvons diviser le problème en segments où les sommets entrants et sortants de chaque segment appartiennent à la même composante biconnexe.

Découverte causale fiable basée sur le principe du supremum d'information mutuelle pour les ensembles de données finis

Il a été reconnu [10, 26] que les orientations prédites par les méthodes basées sur les contraintes sont souvent peu fiables, ce qui a largement limité, en pratique, l'application des méthodes basées sur les contraintes pour découvrir des relations causales dans des données d'observation de la vie réelle.

Il a été démontré que la méthode MIIC, qui combine les cadres basés sur les contraintes et le maximum de vraisemblance, améliore considérablement la situation en réduisant fortement le déséquilibre entre la précision et le rappel, pour toutes les tailles d'échantillon [7, 37]. Par rapport aux méthodes traditionnelles basées sur les contraintes, MIIC réduit également de manière significative l'écart de précision entre les graphes squelettiques et les graphes orientés pour des ensembles de données suffisamment grands, comme nous

le verrons plus loin. Cependant, une perte substantielle de précision subsiste entre les graphes squelettiques et orientés pour les ensembles de données plus petits.

En tant que méthode basée sur des contraintes, MIIC trouve et applique des contraintes basées sur des scores liés au cadre de vraisemblance maximale, à la fois pendant la reconstruction du squelette et pendant l'orientation des arêtes. Le concept clé pour la recherche de l'indépendance conditionnelle pour chaque paire de variables est le schéma "3off2" [1, 2]. Cela revient à découvrir progressivement les indépendances conditionnelles les mieux supportées par les données, c'est-à-dire, $I(X;Y\,|\,\{\,A_i\,\}_n) \simeq 0$, en récupérant itérativement les contributions indirectes les plus significatives de l'information conditionnelle positive à 3 points, $I(X;Y;A_k\,|\,\{\,A_i\,\}_{k-1}) > 0$, de chaque information (mutuelle) à 2 points, $I(X;Y)$, de la manière suivantes :

$$I(X;Y\,|\,\{\,A_i\,\}_n) = I(X;Y) - I(X;Y;A_1) - I(X;Y;A_2\,|\,A_1) - \cdots - I(X;Y;A_n\,|\,\{\,A_i\,\}_{n-1}). \tag{A.1}$$

Dans la pratique, cette information est régularisée pour une taille d'échantillon finie par un terme de complexité (pénalité) $k_N^{\mathrm{NML}}(X;Y\,|\,\{\,A_i\,\})/N$ dans le cadre du maximum de vraisemblance normalisé (NML), et l'information mutuelle pénalisée

$$I'_N(X;Y\,|\,\{\,A_i\,\}) = I_N(X;Y\,|\,\{\,A_i\,\}) - \frac{1}{N} k_N^{\mathrm{NML}}(X;Y\,|\,\{\,A_i\,\}) \tag{A.2}$$

est utilisée pour établir l'indépendance conditionnelle pour $I'(X;Y\,|\,\{\,A_i\,\}) \leq 0$, lorsque des contributions positives indirectes suffisantes et significatives ont pu être collectées de manière itérative dans l'eq. (A.1) pour justifier l'élimination de l'arête $X - Y$.

Ensuite, pendant les étapes d'orientation, les contraintes sont ordonnées en fonction du signe et de la valeur absolue des termes d'information conditionnelle à 3 points régularisés par NML [1, 37], ce qui correspond à la différence entre deux termes d'information mutuelle conditionnelle régularisée par NML :

$$I'_N(X;Y;Z\,|\,\{\,A_i\,\}) = I'_N(X;Y\,|\,\{\,A_i\,\}) - I'_N(X;Y\,|\,\{\,A_i\,\},Z). \tag{A.3}$$

En particulier, chaque terme d'information conditionnelle négatif à 3 points régularisé par NML, $I'_N(X;Y;Z\,|\,\{\,A_i\,\}) < 0$, conduit à la prédiction d'une v-structure, $X \to Z \leftarrow Y$, tandis que chaque terme d'information conditionnelle positif à 3 points régularisé par NML, $I'_N(X;Y;Z\,|\,\{\,A_i\,\}) > 0$, implique une non-v-structure.

En pratique, cependant, pour les petits ensembles de données ou les ensembles de données comprenant des variables avec de nombreux niveaux discrets, les complexités NML peuvent facilement dépasser les termes d'information mutuelle conditionnelle pour

les variables faiblement dépendantes. Par conséquent, MIIC a tendance à inférer certaines orientations comme v-structure, pour lesquelles les deux termes d'information mutuelle conditionnelle régularisés par NML dans l'eq. (A.3) sont négatifs, c'est-à-dire, $I'_N(X;Y \mid \{A_i\}) < I'_N(X;Y \mid \{A_i\}, Z) < 0$, ce qui suggère que Z pourrait en fait être inclus dans un ensemble de séparation de la paire (X, Y), ce qui est en contradiction avec la v-structure inférée, $X \to Z \leftarrow Y$.

Plus récemment, le MIIC a été étendu aux variables continues et mixtes, en estimant l'information mutuelle par un schéma de discrétisation optimal approximatif, basé sur un principe général de supremum d'information mutuelle [12] régularisé pour les ensembles de données finis, et en utilisant un algorithme de programmation dynamique efficace [7]. Ce schéma présente également l'avantage unique de fournir un test d'indépendance efficace, lorsque le partitionnement de variables X et Y dans d'un seul bin maximise l'information mutuelle régularisée par NML.

Pourtant, le principe de partitionnement optimal ne s'applique qu'à l'information mutuelle, et non à l'information mutuelle conditionnelle, qui doit être estimée par la différence entre les termes optimaux d'information mutuelle NML-régularisée, comme $I'_N(X;Y \mid U) = I'_N(Y;X,U) - I'_N(Y;U) = I'_N(X;Y,U) - I'_N(X;U)$ [7]. Par conséquent, les estimations approximatives régularisées par NML peuvent parfois être négatives et conduire à la contradiction entre une v-structure et une indépendance conditionnelle, comme discuté pour le cas discret ci-dessus.

Dans le même esprit que [26] pour l'étape d'orientation de l'algorithme PC, nous proposons pour MIIC la rectification de toutes les valeurs négatives de l'information mutuelle conditionnelle régularisées par NML en valeurs nulles, ce qui, dans la situation susmentionnée, conduira à une information à 3 points régularisée par NML qui s'évanouit, c'est-à-dire $I'_N(X;Y;Z \mid \{A_i\}) = 0$, et empêchera l'orientation du triple $X - Z - Y$ en tant que v-structure.

Cette simple modification de l'algorithme MIIC, appelée MIIC conservatrice, permet d'améliorer considérablement la précision des orientations prédites, même pour des ensembles de données relativement petits. Cette modification est obtenue au prix d'une légère perte de rappel des orientations, mais elle améliore considérablement la fiabilité des orientations prédites pour toutes les tailles d'échantillon. Elle est particulièrement intéressante, en pratique, pour améliorer la fiabilité de la découverte causale pour les applications de données d'observation de la vie réelle.

Application à 400 000 dossiers médicaux de patientes atteintes de cancer du sein.

MIIC améliore la faible sensibilité des méthodes traditionnelles basées sur les contraintes qui sont responsables du grand nombre d'arêtes inférées qui sont faussement négatives [10, 22]. MIIC peut également gérer les données manquantes ainsi que les variables latentes non observées, qui sont omniprésentes dans de nombreuses applications de la vie réelle. En outre, MIIC peut maintenant assurer la cohérence de l'ensemble de séparation avec le schéma itératif proposé dans [22], et améliorer la fiabilité de la découverte causale par les règles d'orientation conservatrices. Cependant, MIIC présente encore un certain nombre de limitations que nous cherchons à surmonter avec la nouvelle méthode iMIIC (interpretable MIIC).

Les méthodes traditionnelles basées sur les contraintes ainsi que la méthode MIIC originale ne font que découvrir des relations causales putatives, en découvrant des orientations de v-structure, qui sont en fait compatibles à la fois avec une relation cause-effet réelle et avec une arête biorientée provenant de causes communes non observées. En revanche, iMIIC distingue les arêtes causales authentiques des arêtes causales putatives en excluant l'effet d'une cause commune non observée pour chaque arête causale authentique prédite. Pour ce faire, il évalue les probabilités distinctes de la tête et de la queue de la flèche pour toutes les arêtes orientées. Les arêtes causales authentiques sont alors prédites si les probabilités de la tête et de la queue de la flèche sont statistiquement significatives, tandis que les arêtes causales restent putatives si leur probabilité de queue n'est pas statistiquement significative ou ne peut être déterminée à partir de données purement observationnelles. De même, les arêtes biorientées correspondent à deux probabilités de tête significatives, tandis que tous les autres cas sont représentés graphiquement comme des arêtes non orientées.

Le cadre probabiliste distinct des orientations de la tête et de la queue de la flèche mis en œuvre dans iMIIC permet également d'inclure des connaissances préalables sur certaines orientations de la tête ou de la queue. En particulier, il est souvent intéressant d'inclure dans les modèles graphiques, quelques variables contextuelles caractérisant par exemple le profil du patient, un paramètre de contrôle ou des conditions expérimentales spécifiques, selon la nature de l'ensemble de données. Contrairement à la plupart des autres variables de l'ensemble de données, de telles variables contextuelles ne varient pas de manière stochastique, mais sont plutôt définies ou sélectionnées de manière externe, et devraient, par hypothèse, avoir toutes leurs arêtes sans tête de flèche entrante. Cela exprime notre connaissance préalable que les variables contextuelles ne peuvent pas être la conséquence d'autres variables observées ou non observées, car elles correspondent en

fait à des paramètres ou des conditions externes définis manuellement.

Ces capacités accrues, qui reposent à la fois sur de nouvelles avancées conceptuelles et sur un refactoring technique majeur du codebase de MIIC, ont été appliquées pour reconstruire un réseau clinique interprétable à partir de l'analyse des dossiers médicaux de près de 400,000 patientes atteintes d'un cancer du sein provenant de la base de données SEER.