Rapport : Robustification de l'algorithme PC

UNIVERSITÉ DE
SHERBROOKE

Math pour l'intelligence artificielle

STT760

Étudiants (CIP) :
Tahar Amairi (amat0601)
Omar Chida (chim2708)
Corentin Pommelec (pomc0601)
Céline Zhang (zhac3201)

Enseignants :
Felix Camirand Lemyre
Samuel Valiquette

2 décembre 2022

Le papier de recherche *Robustification of the PC-Algorithm for Directed Acyclic Graphs* [3] introduit brièvement quelques notions utilitaires dans le domaine des modèles graphiques probabiliste (**MGP**), ainsi que l'algorithme **PC**. Ensuite, les auteurs discutent de leur approche pour *robustifier* l'algorithme contre les données aberrantes. Enfin, ils fournissent un algorithme hybride qui utilise les deux approches après avoir fourni une analyse approfondie de la cohérence de l'algorithme PC standard par rapport à la version robuste suggérée.

L'algorithme PC, nommé sur ses auteurs Peter et Clark, a pour but d'estimer le **CPDA** [1] (*completed partially directed acyclic graphs*), qui encode tous les **DAG** (*directed acyclic graphs*) de la classe d'équivalence de la distribution probabiliste donnée en paramètre. Dans les grandes lignes, l'algorithme se déroule en deux étapes. La première consiste à trouver le squelette C du vrai graphe en utilisant des tests d'indépendance conditionnelle. Ces tests mènent à l'obtention d'un ensemble de séparation S [2]. La deuxième étape utilise la sortie de la première partie pour diriger le squelette C, construisant ainsi le CPDAG. Les auteurs du papier ont introduit trois versions différentes pour la phase 1 de l'algorithme décrites ci-après :

— **Population version** : La parfaite connaissance de toutes les relations d'indépendance conditionnelle est nécessaire pour cette version. Cet algorithme sera la base de la deuxième version et de la version robustifiée. L'idée globale de cette version est de commencer par un graphe complet et de supprimer des arêtes entre chaque paire de noeud X et Y si $X \perp\!\!\!\perp Y \mid Z$ pour tous les ensembles de nœuds Z qui sont adjacents à X dans C (hormis Y). Puis d'enregistrer dans l'ensemble S, les nœuds Z. La condition de suppression de l'arête entre deux nœuds sera la seule différence entre les trois versions différentes de la première partie de l'algorithme PC.

— **Sample version** : Dans cette version, on se limite au cas **gaussien**, où tous les nœuds correspondent à une variable aléatoire de distribution normale. De plus, on suppose que le modèle est fidèle [3]. Puisqu'on est dans le cas gaussien, les indépendances conditionnelles peuvent être déduites de la corrélation partielle, définie récursivement comme suit : $\forall i, j \in V$, $\mathbf{k} \subset V \setminus \{i, j\}$ et un $h \in \mathbf{k}$, on a :

$$\rho_{i,j|\mathbf{k}} = \frac{\rho_{i,j|\mathbf{k}\setminus h} - \rho_{i,h|\mathbf{k}\setminus h}\rho_{j,h|\mathbf{k}\setminus h}}{\sqrt{(1-\rho^2_{i,h|\mathbf{k}\setminus h})(1-\rho^2_{j,h|\mathbf{k}\setminus h})}} \tag{1}$$

L'estimation de $\rho_{i,j|\mathbf{k}}$ est déduite donc récursivement à l'aide de la matrice de corrélation $[\rho_{i,j}]_{i,j=1\ldots p}$. Pour l'estimer, dans le cas gaussien, on utilise l'estimateur du maximum de vraisemblance gaussien noté $\hat{\rho}_{i,j}$. Ensuite, la transformée en Z de **Fisher** est calculée, à l'aide de la formule suivante :

$$Z(i, j|\mathbf{k}) = \frac{1}{2}\log\left(\frac{1+\hat{\rho}_{i,j|\mathbf{k}}}{1-\hat{\rho}_{i,j|\mathbf{k}}}\right) \tag{2}$$

Ce calcul nous permet de tester si la corrélation partielle est égale à 0 ou non. L'hypothèse nulle $H_0(i, j|\mathbf{k}) : \rho_{i,j|\mathbf{k}} = 0$ est rejetée si : $\sqrt{n - |\mathbf{k}| - 3}|Z(i, j|\mathbf{k})| > \Phi^{-1}(1 - \alpha/2)$ où Φ est la fonction de distribution cumulative de $N(0, 1)$. Ainsi, la version **Sample** diffère avec celle de la **Population** uniquement sur la condition requise pour rejeter l'hypothèse nulle permettant de supprimer les arêtes entre les variables indépendantes.

1. Un CPDAG encode toutes les informations d'indépendances contenues dans une classe d'équivalence. Deux CPDAG sont identiques si et seulement s'ils représentent la même classe d'équivalence.

2. Un ensemble de séparation entre deux noeuds X et Y contient les noeuds Z par auxquels X et Y sont conditionnellement indépendants

3. Une distribution P est dite fidèle par rapport à un graphe G si des indépendances conditionnelles de cette dite distribution peuvent être déduites de la d-séparation dans le graphe G et inversement.

— **Robustified version** : C'est la version robustifiée proposée par les auteurs. Étant donné que la seule quantité qui doit être estimée à partir des données afin d'exécuter l'algorithme PC est la corrélation entre toutes les paires de variables, les auteurs ont pris le même modèle que la **Sample** version (détaillée ci-dessus) tout en modifiant simplement l'estimateur puisque, l'estimateur gaussien du maximum de vraisemblance est connu pour sa faible robustesse. En effet, une petite quantité de valeurs aberrantes suffit pour déformer complètement le graphe résultant. Pour robustifier l'algorithme, les auteurs remarquent que la corrélation $\rho(X_i, X_j) = \rho_{i,j}$, peut être exprimée à l'aide de l'écart-type :

$$\rho_{i,j} = \rho_{X^{(i)},X^{(j)}} = \frac{\sigma^2_{aX^{(i)}+bX^{(j)}} - \sigma^2_{aX^{(i)}-bX^{(j)}}}{\sigma^2_{aX^{(i)}+bX^{(j)}} + \sigma^2_{aX^{(i)}-bX^{(j)}}} \tag{3}$$

où $a = \frac{1}{\sigma_{X^{(i)}}}$ et $b = \frac{1}{\sigma_{X^{(j)}}}$, (voir l'article de Huber 1981 [2]). Pour renforcer l'estimation d'échelle, les auteurs ont remplacé l'écart-type empirique par une estimation d'échelle robuste appelé l'estimateur Q_n (Rousseeuw et Croux 1993) définie comme suit :

$$Q_{n;X} = d(|X_i - X_j|; i < j)_{(k)} \tag{4}$$

L'estimateur Q_n a été choisi en raison de ses caractéristiques attrayantes telles que sa formulation simple et son adéquation aux distributions asymétriques. De plus, Q_n peut être calculé en $\mathcal{O}(n \log(n))$ complexité temporel et $\mathcal{O}(n)$ complexité spatiale. En substituant la variance par Q_n dans la formule (3), on obtient notre estimateur robuste pour les corrélations individuelles. Il suffit maintenant de substituer l'ancien estimateur par l'estimateur Q_n pour obtenir une version robuste de la corrélation partielle (analogue à la formule (1)) et sa Z-valeur correspondante (de même en substituant dans la formule (2)). Enfin, pour obtenir la version robuste de l'algorithme PC, il suffit juste de substituer l'ancienne Z-valeur par la nouvelle utilisant l'estimateur Q_n. Par ailleurs, α est donc le seul paramètre qui peut changer l'issue de la première partie de l'algorithme pour les deux versions **Population** et **Sample**.

La deuxième phase de l'algorithme PC est commune, peu importe la version de la première phase utilisée. Cette phase consiste à orienter le squelette. Pour cela, on oriente tout d'abord les *v-structures* à l'aide de l'ensemble de séparation S puis on utilise les trois règles *R1-R2-R3* telles que données dans l'algorithme 2 de [3] pour orienter le maximum d'arêtes possibles. Les auteurs ont abordé la cohérence de l'algorithme PC à la fois **analytiquement** et **empiriquement**. L'approche analytique a été basée sur leur ancien papier de 2007 en traitant la version robuste comme un cas spéciale tout en prenant trois hypothèses :

1. La distribution P est une gaussienne multivariable finie avec une dimensionnalité $p < \infty$.
2. La distribution P est fidèle au DAG G.
3. La distribution P n'est pas dégénérée.

A la fin de l'analyse, les auteurs concluent que la version robuste est bien consistante. En effet, ils se basent notamment sur la mesure de test de la robustesse du *breakdown point*[4]. Or, plus ce dernier est élevé, plus il est robuste, dans notre cas la valeur maximum est atteinte avec un breakdown point de 50%. L'approche empirique, quant à elle, a été effectuée en appliquant l'algorithme sur différents ensembles d'exemples en variant le paramètre α qui, seul, peut influencer les résultats de l'algorithme PC. L'erreur est mesurée à partir de la moyenne de la **distance structurelle de Hamming**[5].

4. C'est la fraction de valeurs arbitraires que peut recevoir l'estimateur sans qu'il ne se détériore.
5. Nombre de changement d'arêtes pour transformer le CPDAG estimé en un CPDAG correct. Un grand DSH indique un mauvais ajustement, tandis qu'un petit DSH indique un bon ajustement.

Les tests qui ont été effectués avec une distribution gaussienne ayant un bruit de 10% à partir d'une distribution **T3** [6] ont montré que l'algorithme PC standard fonctionnait mieux pour des valeurs $\alpha \approx 0.01$. Tandis que les tests effectués avec 10% de bruit provenant d'une distribution de **Cauchy** ont montré la supériorité de l'algorithme PC robuste pour des valeurs $\alpha < 0.03$. Par ailleurs, les résultats sont similaires pour les deux algorithmes pour des valeurs $\alpha \in [0.1; 0.3]$. Bien plus, des tests qui mesurent aussi le **taux de vrais positifs** (TVP) et **faux positifs** (TFP) ont été effectués et leurs résultats sont cohérents avec ceux mentionnés ci-dessus. Ces derniers montrent principalement que, contrairement à la méthode standard, les TVP et TFP de la méthode robuste ne changent pas beaucoup, lorsque des valeurs aberrantes sont introduites. De plus, la méthode robuste atteint un TVP significativement plus élevé tout en gardant un TFP à un niveau comparable à la méthode standard.

Finalement, d'un point de vue de la performance, la mesure du temps d'exécution montre, en moyenne, que la version robuste est 25 fois plus lente que la version standard. Ceci est expliqué par le fait que la version robuste est à un facteur de $\log(n)$ près de la complexité de la version standard. La complexité de la version robuste est donc de l'ordre de $\mathcal{O}(n \log(n) \max(p^q, p^2))$.

En considérant l'analyse faite dans le paragraphe précédent, les auteurs suggèrent une approche dite **hybride** où l'un des deux algorithmes est utilisé selon la situation pour maximiser la consistance. Pour cela, deux heuristiques sont proposées afin de décider quel algorithme exécuter en fonction de l'entrée. La première consiste à utiliser la distribution normale comme distribution de référence et appliquer l'algorithme PC robuste à toutes les données qui semblent contenir plus de valeurs aberrantes que la distribution normale appropriée. Cette approche résulte en un TVP de 0.97 et TFP de 0.05. La deuxième consiste à appliquer la méthode robuste que dans le cas où le bruit est pire qu'une distribution normale avec 10% de valeurs aberrantes d'une distribution T3. Cette approche résulte généralement en un TVP de 1 et TFP de 0.

Pour mieux comprendre l'intérêt concret de la robustification du PC-Algorithm, il est intéressant d'effectuer une revue d'une de ses applications. Dès lors, nous avons été contraints par un nombre restreint de résultats correspondant à cette recherche. Cependant, l'article de recherche *PC Algorithm for Nonparanormal Graphical Models* [1], faisant partie des sujets proposés à l'étude pour ce projet, semble parfaitement convenir à nos attentes. En effet, cet article est paru, en 2013, dans le *Journal of Machine Learning Research* (JMLR) qui est un forum de publication d'articles scientifiques dans le domaine de l'apprentissage automatique. Ses auteurs sont Mathias Drton ancien professeur agrégé de statistiques de l'Université de Chicago et Naftali Harris diplômé d'un master en statistique à l'Université de Stanford. En outre, une autre preuve de la fiabilité de l'article est que le site du JMLR indique qu'il a été cité sur **137** autres papiers scientifiques. Au-delà de sa fiabilité, la pertinence de ce choix repose sur son utilisation de l'algorithme PC robustifié, fidèle à ce à quoi l'on pourrait attendre d'un bon exemple d'utilisation.

Effectivement, l'application se place dans le contexte particulier de l'utilisation de modèles graphiques **nonparanormaux**. Dans ce contexte, les auteurs effectuent plusieurs simulations dans lesquelles ils confrontent l'efficacité du PC algorithme robustifié face à celles du *Rank PC-Algorithm* (RPC) et du *Pearson-PC algorithm*, correspondant à l'algorithme standard. Leurs différentes simulations ont été réalisées à l'aide de la librairie R **pcalg**. Ils prennent en considération trois différents types de données que sont les données **normales multivariées**, les données avec **copule gaussienne**, ainsi que **des données bruitées** qui ne sont pas nonparanormales.

6. distribution T avec 3 degrés de liberté

Ils créent leurs échantillons à partir de deux distributions de graphes, un de dix sommets et un de cent sommets. Bien plus, ils créent différents échantillons de graphes aléatoires, pour chaque type de données listées précédemment, puis échantillonnent plusieurs données du graphe. Ensuite, ils exécutent chacune des trois versions du PC algorithme, sur chaque combinaison résultante en faisant varier α qui est le seul paramètre d'influence. Pour chaque squelette estimé, ils calculent les proportions de vrais et de faux positifs, ainsi que **l'AUC**, qui est une mesure globale des performances pour tous les seuils de classification possibles. Ces métriques sont répertoriées dans trois tableaux, correspondants aux différents types de données utilisées, desquels ils peuvent tirer des interprétations. Concentrons-nous sur les résultats de l'algorithme robuste.

Premièrement les auteurs nous indiquent que pour des données normales multivariées, ce dernier fonctionne bien sur un échantillon de plus large taille, mais ce n'est pas aussi bon sur des tailles d'échantillons plus petites. Deuxièmement, sur des données nonparanormales, les résultats se détériorent par rapport à son application sur des données normales. Enfin, sur des données bruitées, curieusement, l'algorithme robuste fonctionne moins bien que l'algorithme standard sur ces données. Cette dernière conclusion est celle qui se rapproche le plus du sujet de l'article principal étudié et renvoie à l'importance d'essayer plusieurs versions de l'algorithme dans un cas réel d'utilisation, notamment en adoptant l'approche hybride explicitée.

Ainsi, via cette simulation les auteurs sont capables de tirer des premières conclusions sur le type de version à prioriser selon les cas d'utilisation. Parmi elles, nous pouvons discuter de la version RPC qui a fait partie de leur simulation et qui selon les auteurs, a nettement surpassé les autres versions sur des données nonparanormales et semble avoir des différences négligeables sur des données normales. Une autre manière disponible d'aborder cette comparaison d'efficacité, selon le type de données, serait de comparer les différentes versions de plusieurs algorithmes de découverte causale. Une comparaison entre les versions de l'algorithme PC et celles du *Fast Causal Inference Algorithm*, est par exemple envisageable.

Bibliographie

[1] Naftali Harris et Mathias Drton. « PC Algorithm for Nonparanormal Graphical Models ». In : *Journal of Machine Learning Research* 14.69 (2013), p. 3365-3383. URL : http://jmlr.org/papers/v14/harris13a.html.

[2] P.J. Huber, J. Wiley et W. InterScience. *Robust statistics*. Wiley New York, 1981.

[3] Markus Kalisch et Peter Bühlmann. « Robustification of the PC-Algorithm for Directed Acyclic Graphs ». In : *Journal of Computational and Graphical Statistics* 17.4 (2008), p. 773-789. DOI : 10.1198/106186008X381927. eprint : https://doi.org/10.1198/106186008X381927. URL : https://doi.org/10.1198/106186008X381927.

[4] MBA Skool Team. *Breakdown Point - Meaning & Definition*. URL : https://www.mbaskool.com/business-concepts/statistics/8606-breakdown-point.html.

[5] Michail Tsagris. « The FEDHC Bayesian Network Learning Algorithm ». In : *Mathematics* 10.15 (2022). ISSN : 2227-7390. DOI : 10.3390/math10152604. URL : https://www.mdpi.com/2227-7390/10/15/2604.

[6] Yan Yan, Boyao Wu, Tianhai Tian et Hu Zhang. « Development of Stock Networks Using Part Mutual Information and Australian Stock Market Data ». In : *Entropy* 22.7 (2020). ISSN : 1099-4300. DOI : 10.3390/e22070773. URL : https://www.mdpi.com/1099-4300/22/7/773.