Inférence de Graphes Acycliques Dirigés par maximum de vraisemblance pénalisé

Magali Champion [1] & Victor Picheny [2] & Matthieu Vignes [3]

[1] *Laboratoire MAP5, Université Paris Descartes, 45 rue des Saints-Pères, 75270 Cédex 06, France*
{magali.champion@parisdescartes.fr}
[2] *Unité MIAT, Université de Toulouse, INRA, 24 Chemin de Borde Rouge, 31326 Castanet-Tolosan cedex, France*
{victor.picheny@toulouse.inra.fr}
[3] *Institute of Fundamental Sciences - Massey University, Palmerston North, New Zealand*
{m.vignes@massey.ac.nz}

Résumé. Nous nous intéressons à l'apprentissage de graphes acycliques dirigés engendrés par des données bruitées, avec un intérêt particulier pour la grande dimension. Nous proposons une procédure originale basée sur une formulation spécifique du maximum de vraisemblance pénalisé en norme ℓ_1 qui décompose le problème d'estimation de graphes en deux sous-problèmes d'optimisation : l'apprentissage de la structure topologique et de l'ordre des nœuds. Nous présentons des inégalités de convergence pour le graphe estimé ainsi que GADAG, un algorithme destiné à la résolution du problème induit, sous la forme d'un programme convexe intégré à un algorithme génétique. Nous appliquons enfin GADAG à des données simulant des réseaux de régulation géniques, et montrons qu'il se compare favorablement à l'état de l'art.

Mots-clés. Graphes, Optimisation, Grande dimension, Programme convexe.

Abstract. We are interested in learning the structure of directed acyclic graphs spanned by noisy observations with a particular interest on the high-dimensional case. We propose an original procedure based on a specific formulation of the ℓ_1-norm penalized maximum likelihood, which breackdowns the graph estimation into two optimization subproblems : topological structure and node order learning. We provide convergence inequalities for the graph estimator. We also present GADAG, an algorithm devoted to solve the induced problem, in the form of a convex program embedded in a genetic algorithm. We apply GADAG to simulated data that mimic gene regulatory networks and show that it compares favorably to state-of-the-art methods.

Keywords. Graphs, Optimization, High dimension, Convex program.

Introduction

Notre travail se situe dans le cadre d'équations structurelles gaussiennes, décrites par Meinshausen et Bühlmann (2006). On souhaite reconstuire un graphe inconnu $\mathcal{G}_0$ dont les p nœuds sont associés à des variables aléatoires $X^1, ..., X^p$, lesquelles dépendent linéairement les unes des autres :

$$\forall j \in [\![1, p]\!], \quad X^j = \sum_{i=1, i\neq j}^{p} (G_0)_i^j X^i + \varepsilon^j, \tag{1}$$

où $\varepsilon^j \sim \mathcal{N}(0, \sigma_j^2)$ (σ_j connus) est un bruit résiduel gaussien. L'ensemble des arêtes de $\mathcal{G}_0$ correspond aux coefficients non nuls de la matrice G_0, le coefficient $(G_0)_i^j$ représentant la relation qui existe entre le nœud i et le nœud j. Etant donné un échantillon i.i.d. d'observations $(X^1, ..., X^p)$ de taille n ($n << p$) suivant le modèle (1), on souhaite estimer la matrice G_0 pour retrouver la structure du graphe $\mathcal{G}_0$ associé.

Afin d'introduire de la causalité, nous supposons que $\mathcal{G}_0$ est un Graphe Acyclique Dirigé (DAG). Nous supposons en outre que les variances σ_j^2 des bruits associés à chacune des variables sont identiques ($\forall j \in [\![1, p]\!], \sigma_j^2 := \sigma^2$), ce qui permet d'assurer l'identifiabilité du modèle (1) (Peters et al., 2011). Pour estimer la matrice G_0 et retrouver la structure du DAG associé, notre approche consiste à maximiser la vraisemblance, que l'on pénalise pour en limiter le nombre d'arêtes. Ce problème d'optimisation, que nous présentons Section 1, est particulièrement complexe à résoudre dû à la grande dimension ($n << p$) et à l'espace des contraintes (ensemble des DAGs). Une reparamétrisation du problème nous permet de proposer un nouvel estimateur dont nous garantissons la convergence (Section 2) ainsi qu'un algorithme permettant de le calculer (Section 3).

1 Estimation par maximum de vraisemblance

Le modèle (1) peut être réécrit sous la forme matricielle suivante :

$$X = XG_0 + \varepsilon, \tag{2}$$

où X est la matrice de taille $n \times p$ correspondant à n observations des variables $X^1, ..., X^p$, $G_0 := \left((G_0)_i^j\right)_{1\leq i,j\leq p}$ est la matrice associée au DAG $\mathcal{G}_0$ qui a servi à générer les données et ε est la matrice de taille $n \times p$ des bruits.

Une procédure naturelle pour estimer la matrice G_0 et retrouver la structure du DAG $\mathcal{G}_0$ associé consiste à maximiser la log-vraisemblance. Pour assurer la reconstruction d'un graphe parcimonieux, c'est-à-dire d'un graphe ayant seulement un petit nombre d'arêtes, on régularise ce problème d'optimisation en ajoutant une pénalité en norme ℓ_1 :

$$\hat{G} = \underset{G\in\mathcal{G}_{DAG}}{\operatorname{argmin}} \left\{ \frac{1}{n}\|X - XG\|_F^2 + \lambda\|G\|_1 \right\}, \tag{3}$$

où, pour toute matrice $M := (M_j^i)_{1\leq i,j\leq p}$, on note $\|M\|_F^2 := \sum_{i,j}(M_j^i)^2$ la norme de Frobenius et $\|M\|_1 := \sum_{i,j}|M_j^i|$ la norme ℓ_1. L'espace des contraintes $\mathcal{G}_{DAG}$ désigne l'ensemble des DAGs et le paramètre de pénalisation λ définit la parcimonie du modèle.

Afin de simplifier la résolution du problème (3), nous nous appuyons sur une remarque de Bühlmann (2011), qui écrit la matrice d'adjacence G d'un DAG $\mathcal{G}$ comme une combinaison d'une matrice de permutation et d'une matrice triangulaire inférieure stricte.

Proposition 1.1 (Bühlmann, 2011)**.** *Une matrice d'adjacence G est compatible avec un DAG $\mathcal{G}$ si et seulement si il existe une matrice de permutation P et une matrice triangulaire inférieure stricte T telle que :*

$$G = PTP^T.$$

Graphiquement, la matrice de permutation définit un ordre topologique entre les nœuds du graphe suivant le nombre d'arêtes entrantes. Elle n'est généralement pas unique, sauf dans le cas où il existe un chemin reliant l'ensemble des nœuds. La matrice triangulaire inférieure stricte fixe quant à elle la structure du graphe (nombre d'arêtes). Cette décomposition permet de réécrire le problème d'optimisation (3) sous la forme suivante :

$$(\hat{P},\hat{T}) = \underset{P\in\mathbb{P}_p(\mathbb{R}),T\in\mathbb{T}_p(\mathbb{R})}{\operatorname{argmin}} \left\{\frac{1}{n}\|X - XPTP^T\|_F^2 + \lambda\|T\|_1\right\}, \tag{4}$$

où $\mathbb{P}_p(\mathbb{R})$, respectivement $\mathbb{T}_p(\mathbb{R})$, est l'ensemble des matrices de permutation, respectivement triangulaires inférieures strictes, de taille p. Une formulation similaire a été proposée par van de Geer et Bühlmann (2013) pour l'étude théorique du problème de maximum de vraisemblance pénalisé en norme ℓ_0 mais n'a en revanche jamais été exploitée d'un point de vue algorithmique.

2 Inégalités de convergence en prédiction et estimation

D'un point de vue théorique, deux questions se posent :

— l'ordre des variables (information contenue dans P) n'étant pas unique, peut-on garantir que $\hat{P}$, solution de l'équation (4), est compatible avec le vrai graphe $\mathcal{G}_0$ au sens où il existe une matrice triangulaire inférieure stricte T telle que $\hat{P}T\hat{P}^T = G_0$?

— le graphe associé à $\hat{G} := \hat{P}\hat{T}\hat{P}$, où $(\hat{P},\hat{T})$ sont solutions de l'équation (4) est-il proche du vrai graphe $\mathcal{G}_0$?

Sous des hypothèses concernant principalement la dimension du problème ($p\log p = \mathcal{O}(n)$), nous avons obtenu le résultat suivant :

Théorème 2.1. *Si $\lambda = 2C\sqrt{s\log(p)/n}$, où s^2 désigne le nombre d'arêtes de $\mathcal{G}_0$, alors, avec grande probabilité, $\hat{P}$ est compatible avec $\mathcal{G}_0$. De plus, les deux inégalités suivantes*

sont satisfaites avec grande probabilité :

$$\frac{1}{n}\|X\hat{G} - XG_0\|_F^2 \leq Cs^3\frac{\log p}{n},$$
$$\|\hat{G} - G_0|_1 \leq Cs^{5/2}\sqrt{\frac{\log p}{n}}.$$

Le Thérorème 2.1 se base sur les travaux de van de Geer et Bühlmann (2013) autour du maximum de vraisemblance pénalisé en norme ℓ_0 et ceux de Bickel et al. (2009) autour du Lasso. Il assure un bon comportement de l'estimateur (4) considéré, sous réserve d'un compromis entre nombre de variables p, taille de l'échantillon n et parcimonie s^2.

3 L'algorithme GADAG

Dans cette section, nous présentons GADAG (Genetic Algorithm for learning DAG), un algorithme dont le but est de résoudre le problème de double optimisation (4). Il faut remarquer que si l'ordre des variables au sein du graphe est connu (P fixé), le problème (4) se ramène à un problème classique de minimisation de fonctions convexes sous contraintes convexes, que l'on peut résoudre à l'aide d'un algorithme de descente de gradient. La difficulté réside donc dans l'exploration de l'ensemble $\mathbb{P}_p(\mathbb{R})$ des matrices de permutation, qui nous a amené à nous intéresser aux algorithmes génétiques (Michalewicz, 1994).

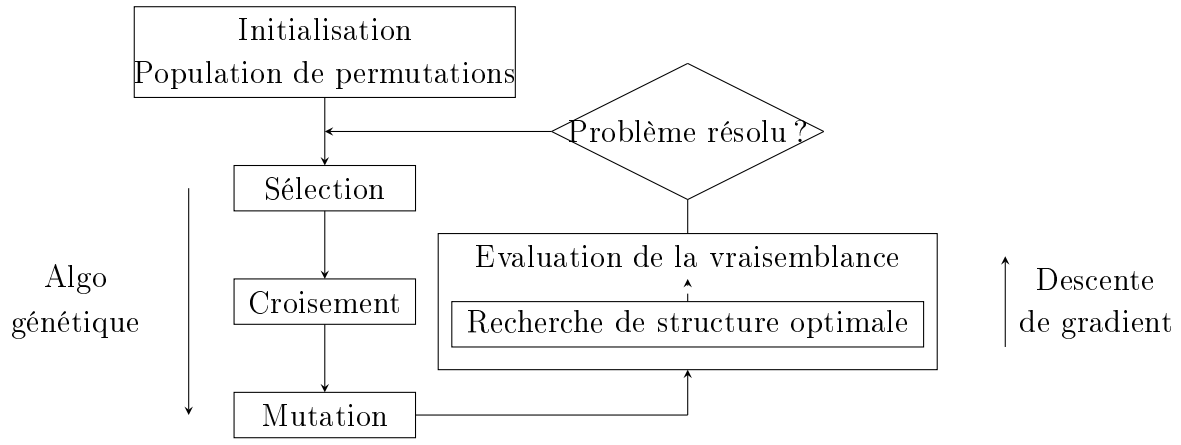

FIGURE 1 – Algorithme GADAG pour la résolution du problème d'inférence de DAGs.

L'algorithme GADAG est ainsi composé d'une boucle externe consistant à explorer intelligemment l'ensemble des permutations par un algorithme génétique dont nous redéfinissons les opérateurs classiques de croisement, de mutation et de sélection pour qu'ils s'adaptent à notre problème. Une boucle interne permet de mesurer le comportement de

chacune des permutations explorées vis-à-vis du problème considéré en associant, à chaque permutation, le T^* optimal qui lui est associé, obtenu en résolvant le sous-problème de minimisation (4) à P fixé (voir Figure 1).

4 Simulations

Pour valider la méthode proposée, nous utilisons des jeux de données simulés issus du challenge DREAM4. Bien que simulées, ces données reproduisent des régulations qui existent entre un ensemble de gènes ($p = 100$) dans des réseaux de régulation géniques (5 au total). Nous comparons GADAG à quatre algorithmes classiquement utilisés pour résoudre des problèmes d'inférence de graphes : Genie3 (Huynh-Thu et al., 2010), basé sur des forêts aléatoires, BootLasso, une version bootstrappée du Lasso (Bach, 2008), et les algorithmes bayésiens GSE (Chickering, 2002) et PC (Spirtes et al., 2000).

Pour comparer ces différentes méthodes, nous mesurons, à paramètre de pénalisation λ fixé, le nombre de vrais positifs VP (arêtes correctement prédites), de faux positifs FP (arêtes prédites à tort), de faux négatifs FN (arêtes manquantes) et de vrais négatifs VN (arêtes correctement non prédites). Cela nous permet alors de calculer le recall ($VP/(TP+FN)$), correspondant à la puissance de reconstruction de la méthode, ainsi que sa précision ($VP/(VP+FP)$). En faisant varier λ entre 0 et $+\infty$, nous traçons alors les courbes précision-recall, qui montrent l'évolution de ces deux quantités en fonction du nombre d'arêtes du graphe. Les résultats obtenus sont présentés Figure 2.

De manière générale, l'algorithme GADAG se place idéalement en comparaison des autres algorithmes, avec une plus grande aire sous la courbe. Ceci est particulièrement vrai sur les réseaux B, C et D.

Références

[1] Bickel, P.J., Ritov, Y., and Tsybakov, A.B. (2009), Simultaneous analysis of lasso and Dantzig selector, *The Annals of Statistics*, 37(4) :1705–1732.

[2] Bühlmann, P. (2011), Causal statistical inference in high dimensions, *Mathematical Methods of Operations Research*, 77 :357-370.

[3] Chickering, D. M. (2002), Optimal structure identification with greedy search, *J Mach Learn Res*, 3 :507-554.

[4] Huynh-Thu, V., Irrthum, A., Wehenkel, L. and Geurts, P. (2010) Inferring regulatory networks from expression data using tree-based methods, *PLoS ONE*, 5(9) :e12776.

[5] Meinshausen, N. and Bühlmann, P. (2006), High-dimensional graphs and variable selection with the lasso, *The Annals of Statistics*, 34(3) :1436-1462.

[6] Michalewicz (1996), *Genetic Algorithms + Data Structures = Evolution Programs,* Springer-verlag edition.

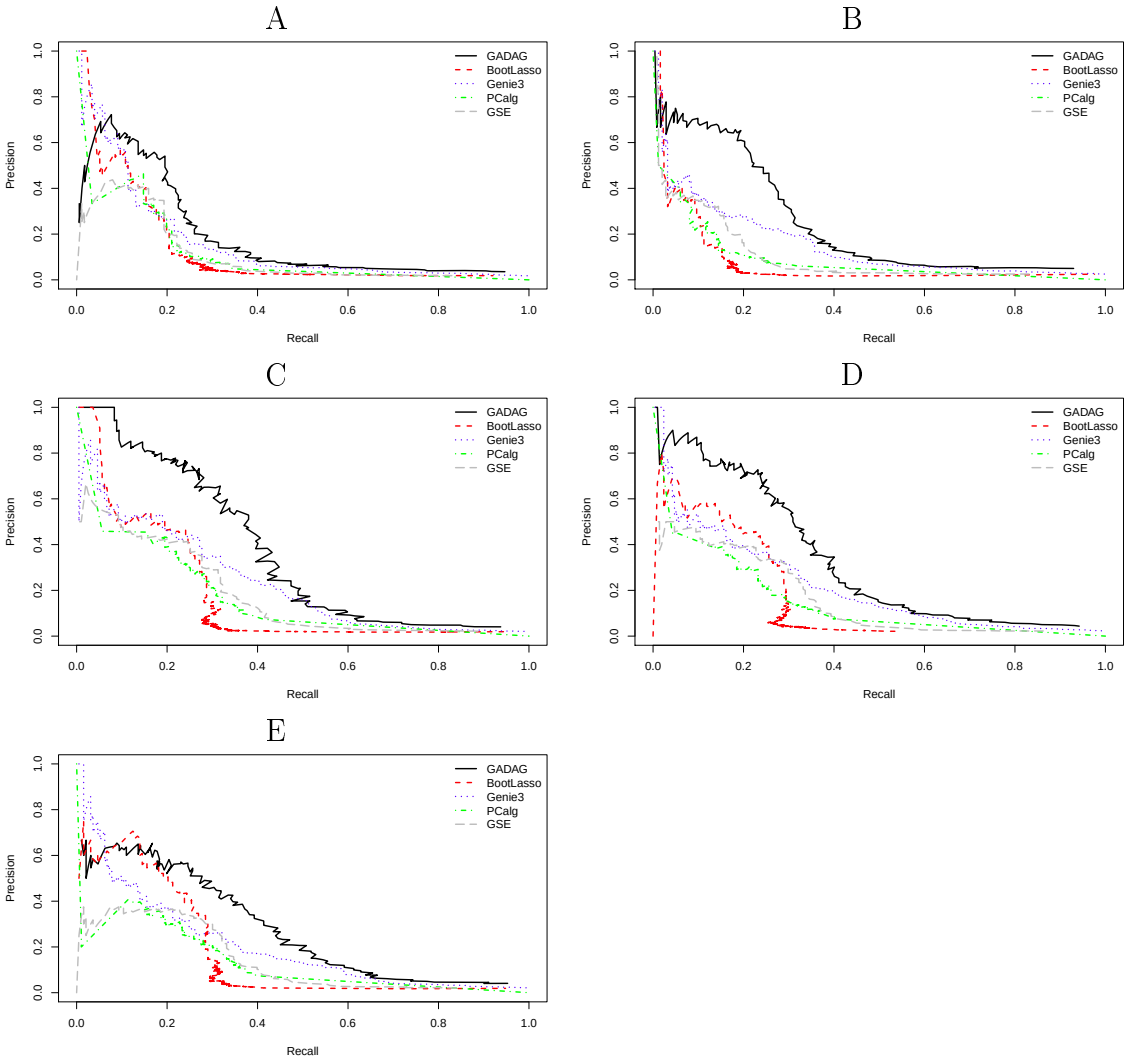

FIGURE 2 – Courbes précision-recall pour les cinq jeux de données DREAM4 et les cinq méthodes comparées.

[7] Peters, J., Mooij, J., Janzing, D. and Schölkopf, B. (2011), Identifiability of causal graphs using functional models, *27th Conference on Uncertainty in Artificial Intelligence (UAI 2011)*.

[8] Spirtes, P., Glymour, C. and Scheines, R. (2000), *Causation, prediction, and search*, Adaptive Computation and Machine Learning, 2nd edition.

[9] Tibshirani, R. (1996), Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1) :267-288.

[10] van de Geer, S. and Bühlmann, P. (2013), ℓ_0-penalized maximum likelihood for sparse directed acyclic graphs, *The Annals of Statistics*, 41(2) :536-567.