GOUVERNEMENT

Liberté
Égalité
Fraternité

Régulation des algorithmes : une alternative à la transparence

Audit en boîte noire

Victor Amblard, Nicolas Rolin, Denis Rousselle, Lucas Verney et Nicolas Deffieux

June 14, 2022

PEReN

Le PEReN

- Pôle d'Expertise de la Régulation Numérique, créé le 31 août 2020
- Sous l'autorité de 3 ministres : Économie, Culture et Numérique

Deux missions :

- Fournir une assistance technique à leur demande aux services et administrations ayant des compétences de régulation sur les plateformes numériques
- Apporte son expertise dans le cadre de travaux de recherche : études à caractère exploratoire ou scientifique

Les protocoles d'audit

Principales questions des régulateurs ? $\Rightarrow$ mathématiser

- Mesures d'exposition croisée contenus / utilisateurs

Exemple

"Pourcentage de cold users masculins qui reçoivent des recommandations de contenus violents vs pourcentage de cold users féminins qui reçoivent les mêmes recommandations ?"

- Détection de biais potentiels dans des algorithmes

Exemple

"Quelles catégories de contenus (TBD) bénéficient le plus de la poussée algorithmique ?"

Principales questions des régulateurs ? $\Rightarrow$ mathématiser

- Comprendre les mécaniques d'un algorithme de recommandation = modèle simplifié de l'algorithme
  - Valider les paramètres annoncés comme influençant le plus les résultats d'un algorithme (P2B / model cards)

Exemple

"Quelles sont les principales caractéristiques des utilisateurs / des contenus qui expliquent les résultats d'un algorithme de recommandation ?"

Deux visions de l'audit des algorithmes

- Approche par questionnaires, avec des KPIs et des descriptions de segments utilisateurs, de fenêtres temporelles, de catégories de contenus... qui sont envoyés aux plateformes
  - Processus lourd, à la granularité limitée.
  - Questionnement peu interactif.
  - Réponses de la plateforme possiblement biaisées ?
- Transparence totale du code source et des données utilisées
  - Propriété intellectuelle ? RGPD ?
  - Online learning ?
  - Charge de travail ($100k+$ lignes de code, plusieurs TB de données) ?

Une troisième vision Audit en boite noire (deep sampling)

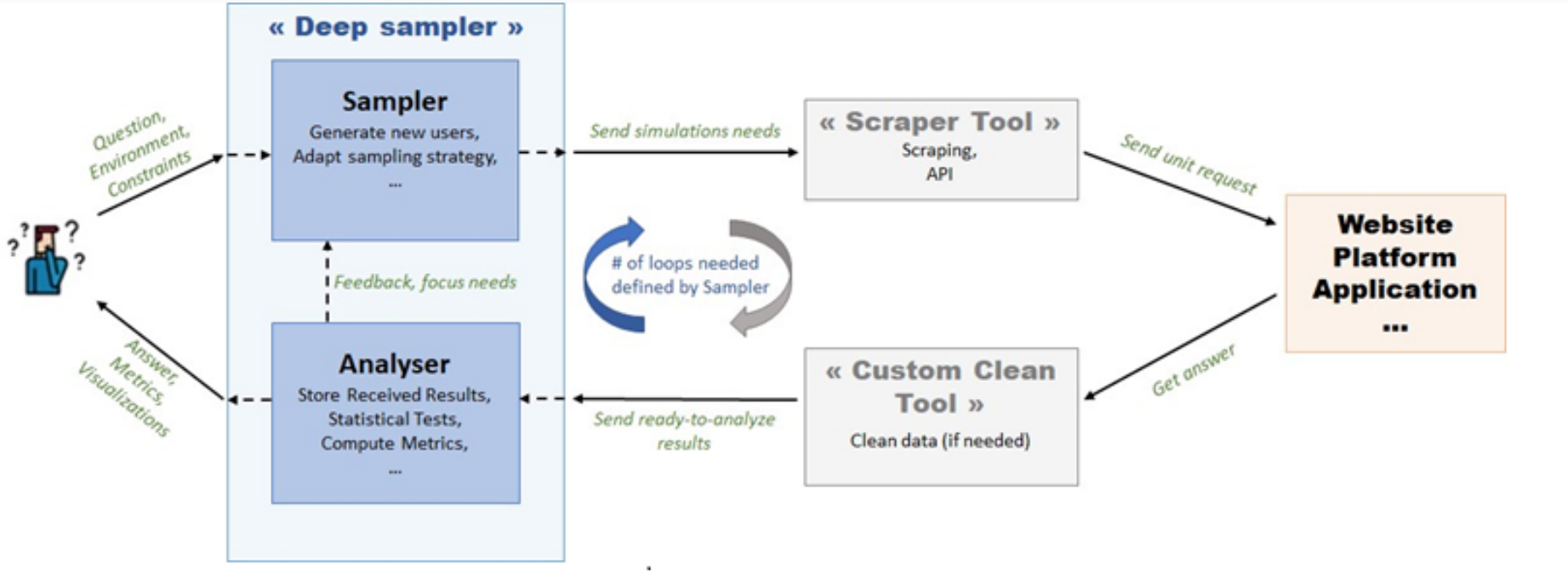
Approche co-développée avec Inria/REGALIA : auditer des algorithmes in situ !

Principe :

- traduire la "question" posée par le régulateur en un biais formel ou une propriété à étudier : ex: "Le pourcentage de cold users masculins recevant des recommandations de contenu violent (dans le top 10) est-il plus élevé que le pourcentage de cold users féminins recevant ce contenu ?".
- réaliser un échantillonnage adaptatif de l'algorithme pour tester cette propriété de façon statistique.

$\Rightarrow$ Le processus d'échantillonnage doit être le plus proche possible des cas d'usage réels pour éviter des biais de test.

Audit en boite noire (deep sampling)



Premières réalisations : algorithmes de détermination de causalité

Obtenir des données

Besoin

Données pour étalonner les techniques et déterminer quelles quantités de données sont nécessaire pour avoir des résultats fiables

- Problématiques juridiques pour collecter des données réelles
- Besoin de contrôle sur l'algorithme testé pour vérifier les hypothèses

$\Rightarrow$ recours à des données simulées

Hubert, votre ami VTC

Objectif

Simuler un algorithme plausible de plateforme de VTC

Les variables indépendantes générées :

- *start_lat*, *start_lon*, *end_lat*, *end_lon* qui correspondent aux données géographiques données par l'utilisateur et qui sont utilisées dans le calcul du prix
- *rain*, *time_of _the_day* utilisé par la plateforme pour calculer le prix (mais pas directement disponible par l'utilisateur)
- *age_of _the_captain* et *butterflies_in_bresil* qui ne seront pas utilisés pour calculer le prix

Simulation avec Hubert

Le prix est calculé par une fonction déterministe assez simple :

$$\max(2, 50 * \text{demand} * (\text{start\_attractivity}/\text{end\_attractivity}) * \text{distance})$$

où

$$\text{demand} = 0.5 * \text{rain} * \sin((\text{time\_of\_the\_day} + 6) * \pi/6) + 2)$$

Test d'indépendance conditionnelle

But

Répondre à la question : la variable X (ex: "est-ce qu'il pleut ?") est elle utilisée par l'algorithme de la plateforme pour prédire la variable Y (ex: prix de la course), toutes choses égales par ailleurs

Les tests existants :

- Paramétriques : Fisher-Z, χ^2,...
- Non-paramétriques : méthodes à noyau : KCI [3] (*Kernel-based conditional independence*)

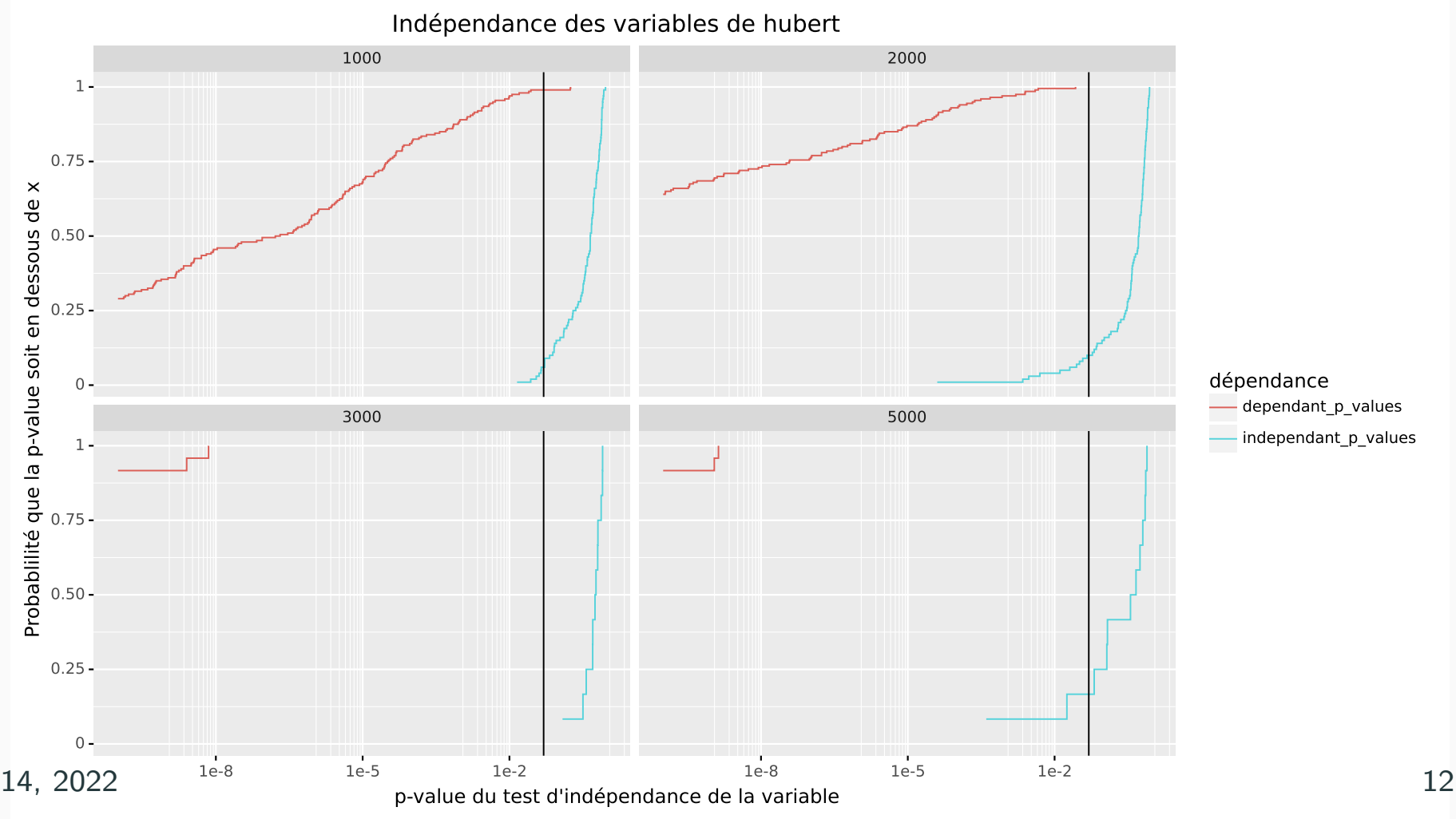
Test non paramétrique KCI

Test retenu : KCI

- Avantage : Ne fait pas de suppositions sur la répartition des données
- Problème : Très lent (complexité : $\mathcal{O}(n^3)$ due au calcul des valeurs propres de la matrice du noyau), donc inapplicable sur tout le jeu de données

Alternative : KCIT [2] (approximation avec des *random Fourier features*)

Test d'indépendance conditionnelle simulé



Graphe causal

But

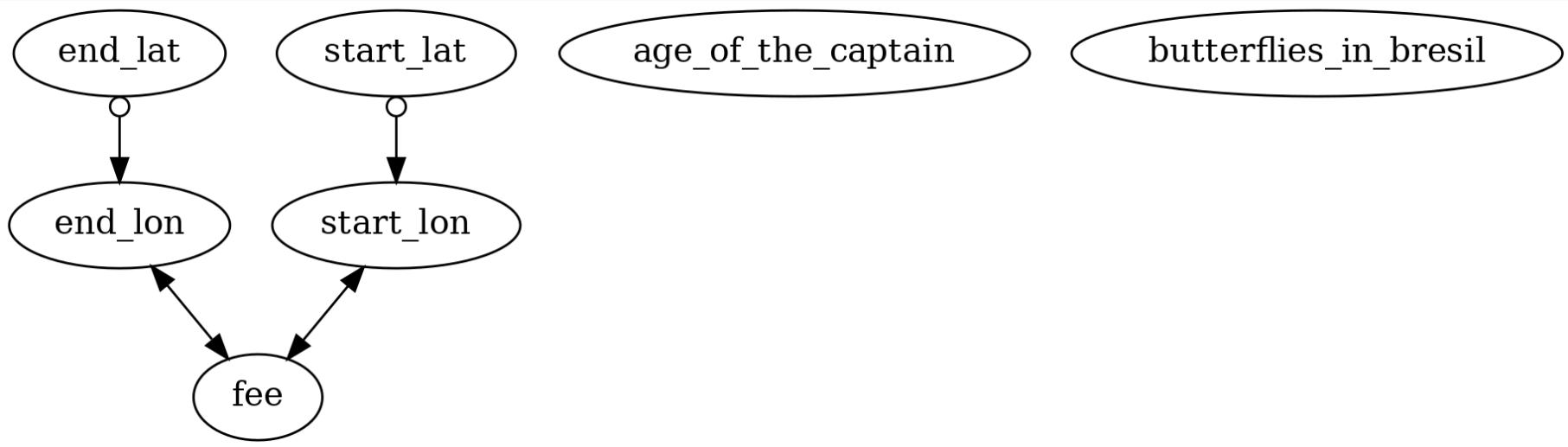
Permettre de tracer un graphe des relations de dépendance des différentes variables

Différentes approches [1]

- Basées sur des modèles fonctionnels (LINGAM pour les modèles gaussiens linéaires, ...)
- Basées sur les contraintes (PC, FCI, ...)

On ne retrouve pas *le* graphe causal G, mais la classe d'équivalence $\overline{G}$ (avec $G \in \overline{G}$) qui permet d'identifier les relations entre les variables, y compris en présence de variables confondantes non mesurées.

Graphe causal



Découverte de variables cachées

But

Déterminer si les données récoltées sont suffisantes pour prédire les résultats de l'algorithmes, ou si l'algorithme utilise une autre variable cachée non accessible à l'auditeur

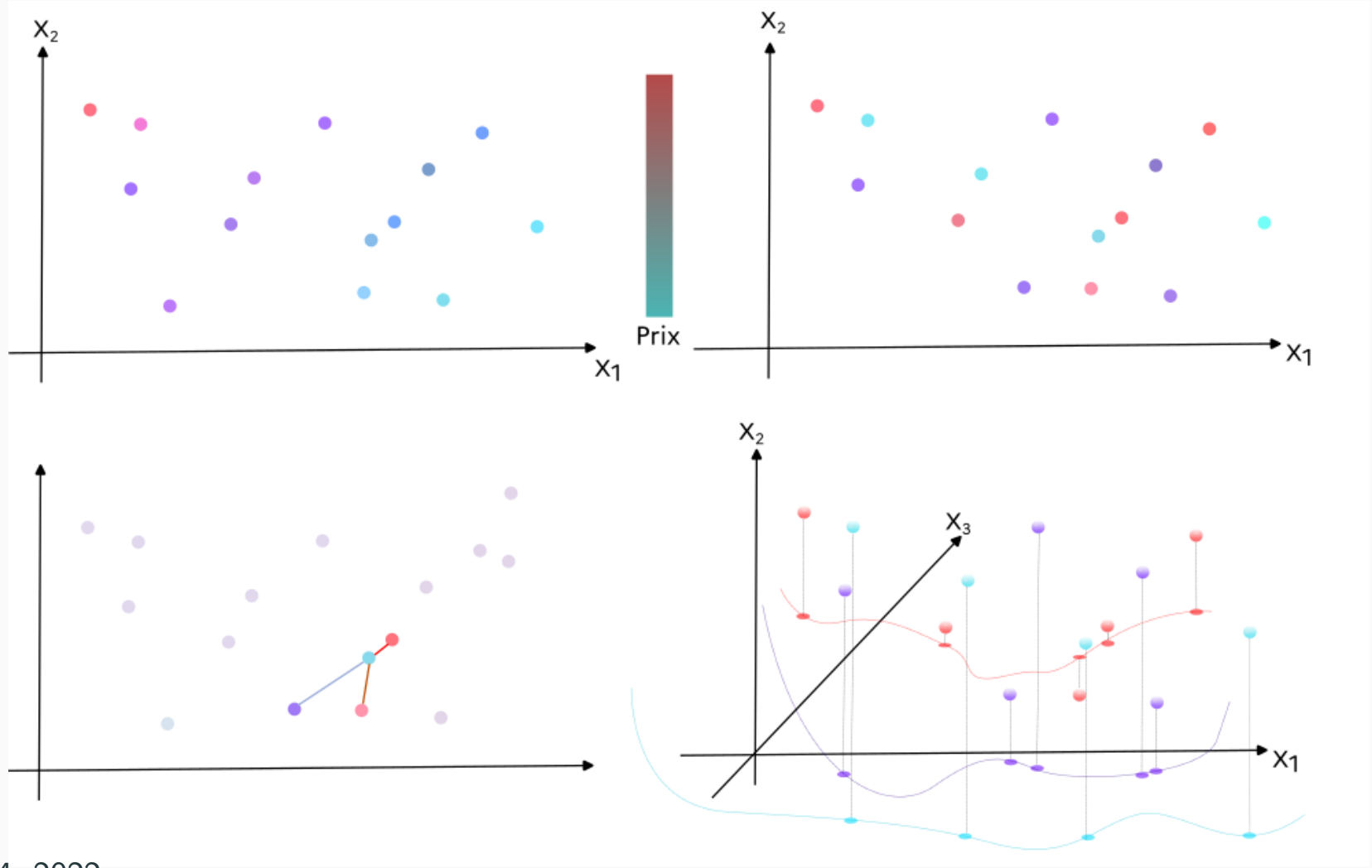
On cherche à trouver dans les données une preuve de la non-injectivité de la prédiction de la variable cible

Découverte de variables cachées (2)

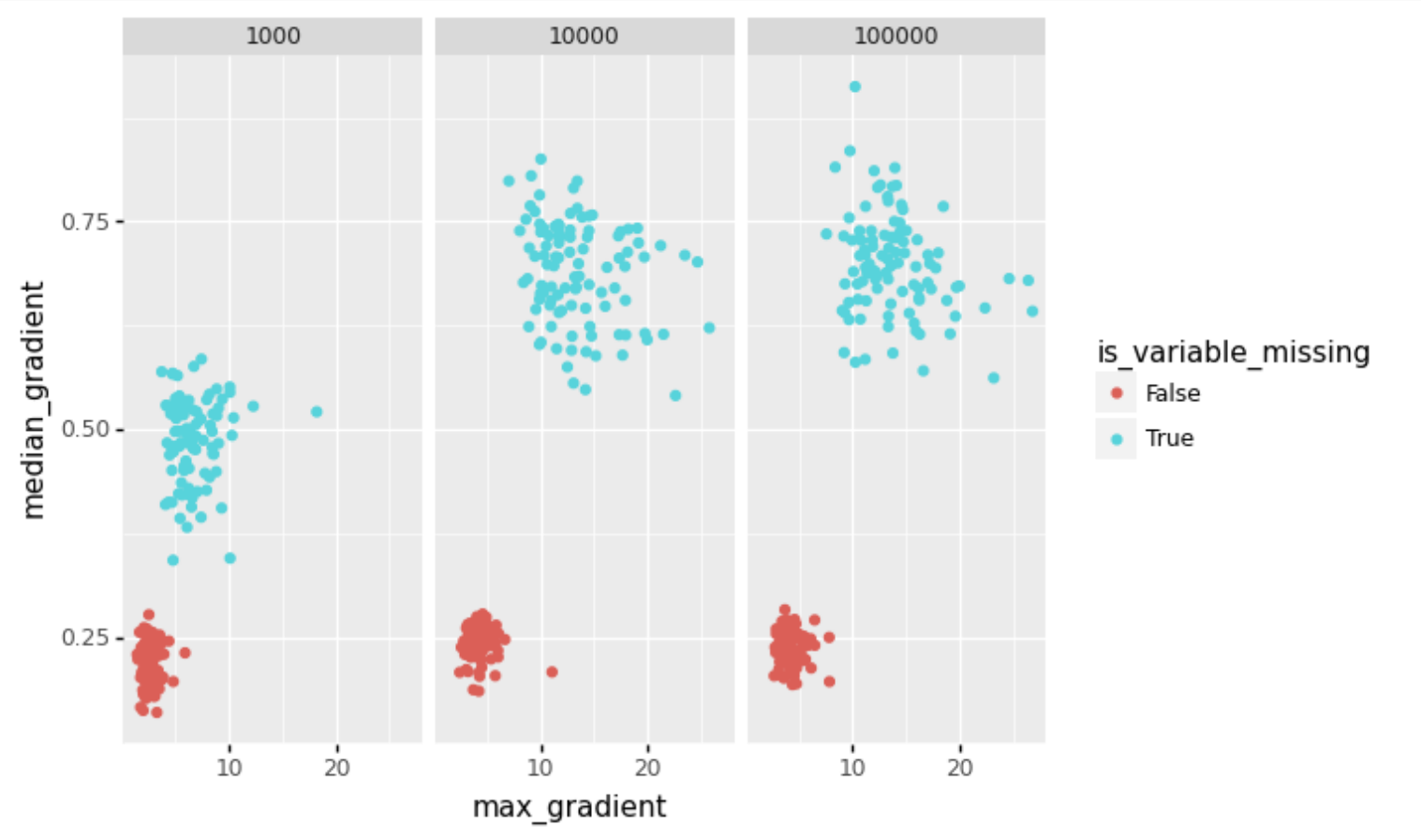
On suit l'heuristique suivante :

- On sépare l'espace à l'aide d'un KD Tree, pour grouper les points proches
- On calcule des gradients locaux en comparant tous les points deux à deux
- On calcule la médiane et le max des gradients obtenus que l'on compare à des hyperparamètres savamment choisis

Découverte de variables cachées



Découverte de variables cachées simulées



Bibliographie

[1] Clark Glymour, Kun Zhang, and Peter Spirtes. "Review of Causal Discovery Methods Based on Graphical Models". In: *Frontiers in Genetics* 10 (2019). ISSN: 1664-8021. DOI: 10.3389/fgene.2019.00524. URL: https://www.frontiersin.org/article/10.3389/fgene.2019.00524.

[2] Eric V Strobl, Kun Zhang, and Shyam Visweswaran. "Approximate kernel-based conditional independence tests for fast non-parametric causal discovery". In: *Journal of Causal Inference* 7.1 (2019).

[3] Kun Zhang et al. "Kernel-based conditional independence test and application in causal discovery". In: *arXiv preprint arXiv:1202.3775* (2012).