Projets de Recherche et Développement: Optimisation Stochastique

Makhlouf HADJI
makhlouf.hadji@irt-systemx.fr

December 14, 2021

Abstract

Ce projet s'intéresse à la détection de fraudes dans des graphes ou réseaux de très grande dimensions. En fonction du contexte, une fraude peut être assimilée à un comportement anormal ou suspect comparé à l'ensemble d'individus, d'objets ou de tout composant constituant la donnée. On s'intéresse dans cette étude à la détection de ces comportements et aux algorithmes dédiés pour aider dans la prise de décision en cas de détection.

1 Détection de fraudes dans les réseaux de grandes tailles

Dans des réseaux de grandes tailles, comme Twitter, Amazon, Netflix ou autres, une fraude peut être décrite et identifiée selon différentes modalités:

- **Exemple d'Amazon**:Les clients qui achètent des produits sur Amazon peuvent évaluer la qualité du produit acheté, ou la performance du vendeur sur Amazon, en mettant un score. Une fraude dans ce cas de figure, consiste à emmettre des "fake reviews" qui sont disponibles à la vente pour augmenter le score d'un produit ou d'un vendeur.
- **Exemple de Twitter**:Les followers et les followees dans Twitter sont sujets à des fraudes sous forme d'achat de milliers de followers "fake" pour tromper l'utilisateur.
- **Exemple de TripAdvisor**: Dans TripAdvisor, une fraude revient à sur-évaluer le score d'un restaurant ou d'un hotel. Des sites dédiès sont utilisés pour acheter des scores ou d'évaluations positives mais qui restent "Fake" (voir par exemple.

Dans ce projet, nous cherchons à mettre en place des algorithmes de détection de fraudes sur des réseaux de grandes tailles similaires à ceux qui sont pré-cités. La complexité du problème adjacent est de pouvoir arriver à une détection efficace et que d'autres fraudes ne passent pas sous le radar (i.e. l'algorithme proposé). Pour rappel, certains sites peuvent bien être utilisés pour tromper l'usager final de ces réseaux. On en cite par exemple: buy1000followers.co, boostlikes.com and buyamazonreviews.com

1.1 Objectifs du projet

On propose d'étudier des algorithmes de détection de fraudes basés sur des datasets réels (Amazon, Twitter, et TripAdvisor). Les data sets mentionés peuvent être téléchargés à partir de ces liens:

- **Dataset d'Amazon**: https://snap.stanford.edu/data/#amazon ou bien http://times.cs.uiuc.edu/~wang296/Data/
- **Dataset de Twitter**: https://github.com/ANLAB-KAIST/traces/tree/main/data
- **Dataset de TripAdvisor**: https://www.kaggle.com/stefanoleone992/tripadvisor-european-restaurants/version/1

1.2 Modélisation mathématique

Dans l'état de l'art, il existe des solutions de détection de fraudes mais qui ne filtrent pas souvent les différentes failles. En effet certaines fraudes passent bien sous le radar, et ainsi peuvent continuer à fausser l'interprétation d'un score, ou l'évaluation d'un produit ou d'une vidéo/film.

Certaines méthodes sont basées sur les approches spectrales comme celle proposée dans [FBox][1] dont le code se télécharge ici : http://www.cs.cmu.edu/~neilshah/code/

D'autres approches (voir référence [Fraudar][2]) peuvent améliorer le taux de détection de fraudes dans différents réseaux comme ceux de Twitter, Amazon, etc. Le c ode est téléchargeable ici: https://github.com/rgmining/fraudar

1.3 Nouvelles approches

Dans ce qui suit, on considère des réseaux dynamiques et stochastiques. En effet, les réseaux ou graphes mentionnés comme cas d'usage dans ce projet, ne sont jamais déterministes. En d'autres mots, on ne connaît pas en avance, l'existence d'une arête sur un graphe ou réseau comme celui de Twitter, facebook, etc.

C'est pour ces raisons, que nous proposons une modélisation qui prend en compte les graphes avec incertitude, sur lesquels on effectue la détection de fraude.

Avant de détailler notre modélisation mathématique, on suppose que le graphe aléatoire considérér dans notre étude est noté par $G = (V = U \cup O, E, F_1, F_2)$ ou V est l'ensemble des sommets du réseau considéré. Il est composé d'un ensemble d'utilisateurs noté par U, et d'un ensemble d'objets noté par O. E est l'ensemble d'arêtes (non orientées) qui relient les deux parties U et O. On dit que G est alors bi-partit.
On introduit aussi une application F_1 qui associe à chaque noeud du graphe G un **degré de suspicion**. De la même manière, F_2 représente le **degré de suspicion des arêtes** E.

Pour un ensemble $S = (A, B)$ avec $A.U$ et $B.O$, on introduit la fonction de suspicion totale donnée comme suit:

$$f(S) = \sum_{i \in S} a_i + \sum_{e \in E(S)} c_e \tag{1}$$

avec a qui représente le degré de suspicion d'un noeud dans S, et c représente le degré de suspicion d'une arête.

Si $a = 0$ pour tous les noeuds du graphe G et que $c = 1$ pour toutes les arêtes de G, alors, on introduit une métrique de suspicion donnée par:

$$g(S) = \frac{|E(S)|}{|S|}$$

Ces résultats sont approfondis dans le papier [FRAUDAR].

Dans le cas ou a et c ne sont pas connus, on considère alors des graphes aléatoires. En effet, les distributions de probabilité de ces coefficients ne sont pas connues. On souhaite donc proposer une approche qui prend en compte cette incertitude tout en détectant un maximum de fraude dans les réseaux mentionnés.

1.3.1 Approche par programmation linéaire

Dans cette partie, on suppose que le graphe étudié est déterministe. Maximiser l'équation (**??**) peut se faire via le programme linéaire suivant:

$$\max \sum_{(ij) \in E} x_{ij} \tag{3}$$

[1] Spotting Suspicisous Link Behavior with fBox: An Adversarial Perspective, N.Shah et al. Pittsburg, PA
[2] FRAUDAR: Bounding Graph Fraud in the Face of Camouflage, B. Hooi et al, CMU

$$x_{i,j} \leq y_i, \forall (i,j) \in E \tag{4}$$

$$x_{i,j} \leq y_j, \forall (i,j) \in E \tag{5}$$

$$\sum_{i \in \mathcal{V}} y_i \leq 1, \tag{6}$$

$$x, y \geq 0, \tag{7}$$

1.3.2 Approche stochastique

Pour développer l'approche stochastique, merci de vous référer à [ZOU][3] et plus exacetment à la page 3.

1.4 Tâches à réaliser

Les tâches à réaliser pour mener à bien ce projet sont décrites comme suit:

- Installer et configurer un environnement de travail en y incluant le solveur d'optimisation CPLEX (idéalement Python)
- Déployer le code récupéré depuis Git des algorithmes cités ci-dessus (approche spectrale fBOX et l'approche FRAUDAR)
- Développer le modèle mathématique d'optimisation stochastique fourni ci-dessus.
- Ajouter une interface graphique permettant de simuler sur différents data sets pour permettre de "benchmarker" les solutions
- Proposer des simulations numériques pour valider et comparer l'ensemble des algorithmes proposés ci-dessus. On peut utiliser plusieurs métriques pour pouvoir les comparer (taux de détection, etc.)
- Rédiger un rapport pour résumer le projet de bout en bout

[3] Z.Zou, Polynomial time algorithm for finding densest subgraphs in uncertain graphs, HIT, China