Algorithmes d'approximation et sketches pour les problèmes de clustering

David Saulpic[1]

Cet article est le résumé de la thèse de doctorat[2] *de son auteur, qui a reçu le prix de thèse Gilles Kahn 2023, décerné par la Société informatique de France.*

Le titre de ma thèse cache plusieurs termes techniques, que je vais m'attacher à expliquer un à un, pour motiver les questions abordées. Mais tout d'abord, un peu de contexte. Ces travaux s'inscrivent dans le domaine de l'informatique théorique ; les problèmes que nous avons étudiés ont leurs racines dans le traitement des « big data », mais notre apport est avant tout théorique : avec quelle précision peut-on espérer répondre à ces problèmes, même dans des cas très contraints ? Comment simplifier leur structure et les données d'entrée avant de répondre au problème ? L'objectif premier n'est pas d'implémenter, mais d'abord de comprendre — ce qui, parfois, mène à des implémentations efficaces, comme nous avons pu l'expérimenter dans ce travail.

Les problèmes de clustering

Ce nom peut avoir de nombreuses interprétations, et cache différents problèmes. Pendant ma thèse, nous nous sommes concentrés principalement sur les problèmes des k-moyennes et k-médianes. Leur but est de répondre à la question

1. Institut de recherche en informatique fondamentale (IRIF), université Paris Cité & CNRS.
2. https://theses.fr/2022SORUS194.

DOI:10.48556/SIF.1024.24.145

suivante : comment compresser un nuage de points de façon fidèle ? Si l'objectif est de représenter le nuage par un point unique, la solution habituelle est de choisir la moyenne — voir Figure 1. Si les points sont simplement des nombres réels (i.e., le nuage de points est de dimension 1), alors on peut aussi choisir la médiane.

Mais si on s'autorise à représenter le nuage par plus d'un point, disons k points, comment les choisir ? Pour formaliser le problème, étant donné un ensemble de points $P \subset \mathbb{R}^d$, on cherche à calculer un ensemble S de k représentants qui soit *fidèle*. Au début des années 60, une fonction de coût a été définie (notamment par Lloyd [4]) de la façon suivante : si on considère que chaque point de P est remplacé par son représentant le plus proche, l'erreur faite est la distance entre le point et son représentant. Le *coût* d'un ensemble de représentants S est alors la somme des erreurs au carré. En équation,

$$\mathrm{cost}(P, S) = \sum_{p \in P} \min_{s \in S} \mathrm{dist}(p, s)^2. \tag{1}$$

Le but du problème des k-moyennes est de calculer l'ensemble S de taille k qui minimise la fonction cost, avec l'espoir qu'un faible coût implique une bonne représentativité. Le nom vient du fait que le problème généralise la moyenne : pour $k = 1$, le point minimisant le coût est la moyenne de P. Le problème des k-médianes est similaire, mais le coût est simplement la somme des erreurs. Cette fonction objectif généralise la médiane : si $k = 1$ et $P \subset \mathbb{R}$, alors la meilleure solution est la médiane.

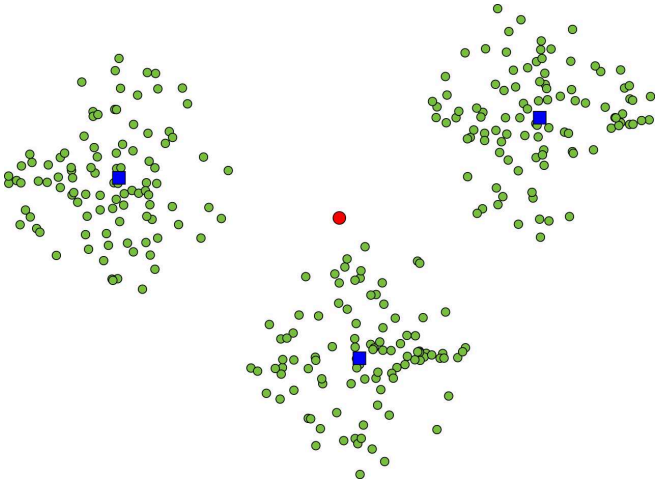

Fig. 1. Un nuage de points avec trois groupes distincts. La moyenne, le cercle rouge, est un point artificiel ne correspondant à rien dans le nuage de point. En revanche, les trois carrés bleus représentent plus correctement le nuage.

L'intérêt de compresser un nuage de points par k points plutôt qu'un seul est évident dès que les points sont regroupés en *clusters*, comme dans l'exemple présenté en figure 1 : dans ce cas, il y a trois groupes de points distincts, et choisir $k = 3$ permet une bien meilleure représentation que de garder simplement la moyenne. En un sens, les représentants permettent de détecter les clusters du nuage, d'où le nom *clustering*.

Algorithmes d'approximation

Malheureusement, le problème de minimiser la fonction cost est NP-complet [5]. À la place, on peut chercher à calculer une solution aussi bonne que possible : plutôt que de chercher la solution optimale, on peut essayer de calculer une solution qui a un coût $(1+\varepsilon)$ fois l'optimal. On appelle une telle solution une $(1+\varepsilon)$-approximation. Plus ε est petit, meilleure est la solution : la grande question des algorithmes d'approximation est de savoir à quel point ε peut être proche de zéro. Cette étude du problème permet de mieux comprendre sa structure, et de développer des outils techniques qui sont souvent utilisables dans d'autres scénarios.

Pour les k-moyennes, un résultat récent [1] montre qu'un algorithme simple dit de *recherche locale* permet de calculer une $(1+\varepsilon)$-approximation pour n'importe quel ε fixé, au prix d'un temps de calcul $nk(\log n)^{(d/\varepsilon)^{O(d)}}$. Dans la première partie de ma thèse, nous avons eu deux objectifs : d'abord, réduire ce temps de calcul le plus possible ; ensuite, étendre le résultat à des espaces métriques les plus généraux possibles.

En effet, P n'est pas forcément un nuage de points dans un espace euclidien (par exemple, la distance entre points n'est pas forcément la distance euclidienne mais peut être une autre norme ℓ_p). Nous avons cherché à réduire le plus possible les hypothèses sur P, pour identifier quelle propriété de P rend le problème « facile ». La propriété que nous avons identifiée est la suivante : il faut que, dans l'espace métrique, toute boule de rayon R soit couverte par 2^d boules de rayon $R/2$ — pour tout R. On dit alors que la *doubling dimension* de l'espace est d. On a construit un algorithme qui permet, dans de tels espaces, de calculer une $(1+\varepsilon)$ approximation avec un temps de calcul essentiellement $n \cdot 2^{1/\varepsilon^{d^2}}$. Quand n et k sont grands, c'est beaucoup mieux que le temps de calcul précédent — mais le résultat reste théorique et l'algorithme éloigné de toute application !

En revanche, les techniques et résultats théoriques que l'on a développés lors de ce travail trouvent des applications plus pratiques : on a montré comment les utiliser pour développer un algorithme efficace en pratique, et qui a en plus des garanties de *confidentialité différentielle* — d'une certaine manière, cet algorithme respecte la confidentialité des utilisateurs et utilisatrices qui ont fourni les données du nuage de points.

Sketches

Le dernier mot clef du titre concerne la seconde partie de ma thèse. En informatique théorique, un *sketch* (« esquisse » en français) est un résumé de l'entrée, qui utilise peu de mémoire mais permet de calculer une solution (approchée) au problème considéré. Le but ici est de construire des sketches qui utilisent le moins de mémoire possible, par exemple pour résoudre le problème dans des cas de « big data » où l'entrée du problème ne tient pas dans la mémoire d'une unique machine. En d'autres mots, pour le problème des k-moyennes, on cherche à compresser l'entrée pour pouvoir trouver une solution qui minimise le coût : donc, on compresse pour pouvoir compresser ! Mais ce procédé ne se mord pas la queue, parce que les propriétés désirables de la première compression sont bien différentes de celles de la seconde.

Une forme de sketches, celle que j'ai le plus étudié, s'appelle un *coreset*. On dit que Ω est un ε-coreset pour P si, pour tout ensemble S de k représentants,

$$\mathrm{cost}(P,S) = (1 \pm \varepsilon)\mathrm{cost}(\Omega, S). \tag{2}$$

Si Ω contient peu de points différents, on peut donc oublier P et résoudre le problème des k-moyennes directement sur Ω. Un des principaux résultats de ma thèse est de caractériser des propriétés de P qui permettent de calculer un petit coreset. Si $P \subset \mathbb{R}^d$, on a montré l'existance de coreset de taille $k\varepsilon^{-4}$ — donc indépendant de la taille de P ! Pour un k fixé et une précision ε donnée, n'importe quel ensemble de points P, qu'il contienne 10^6 ou 10^{100} points, peut se résumer en un ensemble de même taille. Cette propriété était déjà connue (voir par exemple [3]) : notre contribution est de réduire la taille du coreset, et ce dans un cadre très général : par exemple, notre résultat s'étend directement si l'entrée à une faible doubling dimension, ou bien si la distance entre les points est donnée par un graphe structuré.

Perspectives

Les contributions de ma thèse sont avant tout de nature théorique, mais les problèmes de k-moyennes et k-médianes font partie des principaux outils à la fois du traitement de « big data » (qu'ils permettent de compresser) et d'apprentissage non-supervisé, car ils permettent une classification de l'entrée. Pour de nombreuses raisons, je ne me reconnais pas forcément dans ces applications. Notamment, je me méfie d'un problème identifié par exemple par Cathy O'Neil [6] : les algorithmes traitant des données à grande échelle peuvent rendre dramatique le moindre biais, ou la moindre fuite de confidentialité.

J'essaie de réorienter ma recherche vers de nouvelles perspectives, en combinant l'informatique théorique avec d'autres disciplines. Par exemple, des contraintes légales ou sociétales imposent désormais aux algorithmes une

certaine confidentialité et une certaine équité. Je cherche actuellement à développer des algorithmes qui respectent ces contraintes, et aimeraisquestionner les techniques et définitions actuelles pour en comprendre la portée et les limites. J'aimerais aussi utiliser les outils et la manière de penser de l'informatique théorique dans d'autres disciplines. Par exemple, avec un groupe de chercheurs et chercheuses de Sorbonne Université, l'université de Paris-Cité et l'université Panthéon-Assas, nous menons des recherches sur le découpage de la France en circonscriptions pour les élections législatives, et essayons de mesurer la représentativité du découpage actuel [2].

Références

[1] Vincent Cohen-Addad. 2018. A fast approximation scheme for low-dimensional k-means. In (SODA '18). Retrieved from `http://dl.acm.org/citation.cfm?id=3174304.3175298`.

[2] Thomas Ehrhard, Solal Attias, Evripidis Bampis, Vincent Cohen-Addad, Bruno Escoffier, Claire Mathieu, Fanny Pascual, Adèle Pass-Lanneau, and David Saulpic. 2022. Découpage électoral des circonscriptions législatives en france. *Revue francaise de science politique* 72, 3:333–364.

[3] Dan Feldman and Michael Langberg. 2011. A unified framework for approximating and clustering data. In *Proceedings of the 43rd ACM symposium on theory of computing, STOC 2011, san jose, CA, USA, 6-8 june 2011*, 569–578.

[4] Stuart Lloyd. 1982. Least squares quantization in PCM. *IEEE transactions on information theory* 28, 2:129–137.

[5] Nimrod Megiddo and Kenneth J. Supowit. On the complexity of some common geometric location problems. *SIAM J. Comput.* 13.

[6] Cathy O'Neil. 2018. *Algorithmes, la bombe à retardement*. Les arènes.