Locality Sensitive Hashing

Le *Locality Sensitive Hashing* (LSH) est l'une des techniques phares mises au point ces deux dernières décennies pour contourner le fléau de la dimension dans les requêtes de voisinage en grande voire très grande dimension. Elle a notamment permis une percée majeure sur le problème du calcul approché de la distance au plus proche voisin, qui est la question qui nous intéresse ici. Cette technique s'inspire des tables de hachage utilisées depuis de nombreuses années en informatique comme structure de données pour l'encodage d'ensembles non-ordonnés (`set` en Python, `unordered_set` en C++).

Pour rappel, le principe du hachage est le suivant : étant donné un espace X dont seront issus les éléments à stocker, on choisit une *fonction de hachage*, c'est-à-dire une simple fonction

$$h\colon X \to \mathbb{Z}$$

sans hypothèses particulière pour le moment. Puis, étant donné un ensemble fini $P = \{p_1, \cdots, p_n\}$ d'éléments distincts de X, on crée un tableau `tab` pouvant stocker jusqu'à $m \geq n$ éléments de X, indexé par les indices 0 à $m-1$, ensuite on *hache* les éléments de P dans ce tableau, c'est-à-dire que chaque élément p_i est inséré dans `tab` à l'indice $(h(p_i) \mod m)$. Ce faisant, il se peut que des *collisions* surviennent, c'est-à-dire que deux éléments distincts de P soient hachés au même endroit dans le tableau. Dans ce cas, plusieurs stratégies sont envisageables pour résoudre la collision, comme par exemple le *chaînage*, qui consiste à stocker dans chaque case de `tab` une liste chaînée contenant tous les éléments de X insérés à cet endroit. Le choix de la stratégie importe peu en théorie, le principe étant qu'une bonne fonction de hachage doit répartir les éléments de P aussi uniformément que possible dans le tableau, de manière à prévenir les collisions avec forte probabilité. De manière générale, on dit qu'une fonction de hachage $h\colon X \to \mathbb{Z}$ est *bonne* si elle est injective, c'est-à-dire qu'elle vérifie la propriété suivante :

$$\forall x, y \in X,\ \begin{cases} h(x) = h(y) & \text{si } x = y \\ h(x) \neq h(y) & \text{si } x \neq y \end{cases} \tag{1}$$

Le but du LSH est de généraliser ce principe au cas où l'espace X est muni d'une métrique d et où l'on souhaite pouvoir déterminer, via une fonction de hachage $h\colon X \to \mathbb{Z}$, si x et y sont distants d'au plus un seuil fixé $r \geq 0$:

$$\forall x, y \in X,\ \begin{cases} h(x) = h(y) & \text{si } \mathrm{d}(x, y) \leq r \\ h(x) \neq h(y) & \text{si } \mathrm{d}(x, y) > r \end{cases} \tag{2}$$

Le test d'égalité $x \stackrel{?}{=} y$ décrit dans l'équation (1) correspond alors au cas particulier où $r = 0$.

Malheureusement, tandis que l'on peut démontrer l'existance de fonctions de hachage ayant la propriété (1), il n'existe en général pas de fonction de hachage ayant la propriété (2) lorsque $r > 0$. Il faut alors relaxer la contrainte, ce que l'on fait des deux manières suivantes à la fois :

- d'une part, on remplace le seuil de distance r par deux seuils distincts $r_1 < r_2$;
- d'autre part, on demande seulement à h de vérifier chaque cas de la propriété avec une certaine probabilité, cette dernière étant sur le choix de la fonction h parmi une certaine famille $\mathcal{H}$ de fonctions.

Formellement, cela donne le concept de LSH défini ci-dessous, que l'on utilisera tout au long du problème :

Définition 1. Soit (X, d) un espace métrique. Étant donnés des seuils de distance $0 \leq r_1 < r_2$ et de probabilités $1 \geq \pi_1 > \pi_2 \geq 0$, une famille $\mathcal{H}$ de fonctions de hachage $h\colon X \to \mathbb{Z}$ munie d'une mesure de probabilité μ est dite (r_1, r_2, π_1, π_2)-*sensible* si

$$\forall x, y \in X,\ \begin{cases} \Pr\left[h(x) = h(y)\right] \geq \pi_1 & \text{si } \mathrm{d}(x,y) \leq r_1 \\ \Pr\left[h(x) = h(y)\right] \leq \pi_2 & \text{si } \mathrm{d}(x,y) \geq r_2 \end{cases} \tag{3}$$

où les probabilités sont sur le choix de la fonction $h \in \mathcal{H}$ selon la mesure de probabilité μ.

Remarquez que cette définition n'impose aucune condition sur la probabilité de collision lorsque $r_1 < \mathrm{d}(x,y) < r_2$. Par ailleurs, même si elle suppose que les fonctions de hachage sont à valeurs dans $\mathbb{Z}$, cette hypothèse n'est pas nécessaire pour énoncer les axiomes de l'équation (3) ; dans la suite on prendra typiquement $\mathbb{Z}^k$ pour un certain $k \geq 1$ comme espace d'arrivée des fonctions de hachage.

0.1 LSH dans le cube de Hamming

À partir de maintenant, et pour le reste du problème, on se place dans le cas particulier du *cube de Hamming*, c'est-à-dire l'hypercube $X = \{0,1\}^d$ muni de la norme ℓ^1 notée $\|\cdot\|_1$. Remarquez que seuls les sommets de l'hypercube appartiennent à X. En d'autres termes, les éléments de X sont les chaînes de bits de longueur d. Par exemple, pour $d = 2$, les éléments de X sont 00, 10, 01 et 11. Dans ce contexte, $\|x-y\|_1 := \sum_{i=1}^{d} |x_i - y_i|$, où la notation z_i désigne le i-ème bit de la chaîne z, est appelée *distance de Hamming* entre les chaînes x et y.

Question 0.1. *Prouvez que la norme ℓ^1 compte simplement le nombre de bits distincts entre chaînes de bits, c'est-à-dire :*

$$\forall x, y \in \{0,1\}^d,\ \|x - y\|_1 = \#\{1 \leq i \leq d \mid x_i \neq y_i\} \tag{4}$$

où # désigne l'opérateur de cardinalité, qui compte le nombre d'éléments distincts dans l'ensemble qui le suit.

Dans le cube de Hamming, on propose d'utiliser la famille des projections sur les différents axes de l'espace :

$$\mathcal{H} = \{x \mapsto x_i \mid 1 \leq i \leq d\}$$

On munit cette famille de d projections de la mesure de probabilité uniforme, c'est-à-dire que chaque projection est choisie avec probabilité $1/d$ lorsqu'on fait un tirage aléatoire. Soient r et ε tels que $0 \leq r \leq r(1+\varepsilon) \leq d$, et tirons $h \in \mathcal{H}$ aléatoirement.

Question 0.2. *Montrez que, pour tous $x, y \in X$ tels que $\|x - y\|_1 \leq r$, on a $\Pr[h(x) = h(y)] \geq 1 - r/d$.*

Question 0.3. *De manière similaire, montrez que, pour tous $x, y \in X$ tels que $\|x - y\|_1 \geq r(1+\varepsilon)$, on a $\Pr[h(x) = h(y)] \leq 1 - r(1+\varepsilon)/d$.*

Ainsi, la famille $\mathcal{H}$ définie ci-dessus est $(r,\ r(1+\varepsilon),\ \pi_1(r),\ \pi_2(r,\varepsilon))$-sensible pour tous r et ε tels que $0 < r < r(1+\varepsilon) \leq d$, où l'on pose $\pi_1(r) = 1 - r/d$ et $\pi_2(r,\varepsilon) = 1 - r(1+\varepsilon)/d$.

Question 0.4. *Que se passe-t-il pour $r = 0$ ou $\varepsilon = 0$?*

Il s'avère que cette famille $\mathcal{H}$ n'est pas suffisamment sensible pour nos besoins. Nous allons donc la remplacer par la famille $\mathcal{G}_k$ de fonctions à valeurs dans $\mathbb{Z}^k$ (et non plus $\mathbb{Z}$) dans laquelle chaque élément $g = (h_1, \cdots, h_k) \colon X \to \mathbb{Z}^k$ est obtenu par tirages aléatoires indépendants successifs de k projections $h_1, \cdots, h_k \in \mathcal{H}$ que l'on concatène ensuite pour former le vecteur de projections g. Ici, $k \in \mathbb{N}^*$ est un paramètre que nous fixerons plus tard.

Question 0.5. *Montrez que la famille $\mathcal{G}_k$ ainsi construite est $(r,\ r(1+\varepsilon),\ \pi_1(r)^k,\ \pi_2(r,\varepsilon)^k)$-sensible pour tous r et ε tels que $0 < r < r(1+\varepsilon) \leq d$.*

0.2 Un détour par le problème du (r, ε)-voisinage

Rappelons que notre espace X est le cube de Hamming $\{0,1\}^d$ muni de la norme ℓ^1. Avant d'attaquer le problème du calcul de la distance au plus proche voisin, intéressons-nous à sa version décisionnelle : étant donné un nuage de points $P = \{p_1, \cdots, p_n\} \subseteq X$ et un rayon $r > 0$, notre tâche est de construire une structure de données permettant de décider, pour tout point de requête $q \in X$, si le plus proche voisin de q parmi les points de P se trouve à une distance plus petite que r (dans ce cas on répond `OUI`) ou plus grande que r (dans ce cas on répond `NON`). En réalité on va regarder la version approchée de ce problème, paramétrée par r et ε tels que $0 < r < r(1+\varepsilon) \leq d$:

- si $\min\limits_{1 \leq i \leq n} \|q - p_i\|_1 \leq r$ alors répondre `OUI` ;
- si $\min\limits_{1 \leq i \leq n} \|q - p_i\|_1 > r(1+\varepsilon)$ alors répondre `NON` ;
- sinon ($r < \min\limits_{1 \leq i \leq n} \|q - p_i\|_1 \leq r(1+\varepsilon)$), les deux réponses `OUI` ou `NON` sont acceptées.

Cette version approchée est appelée *problème du (r, ε)-voisinage*. Notez que, quand $r(1+\varepsilon) = d$, il suffit de répondre systématiquement `OUI` pour résoudre le problème. Par ailleurs, avoir $r > 0$ revient à avoir $r \geq 1$ dans le cube de Hamming. Nous supposons donc à partir de maintenant que $1 \leq r < r(1+\varepsilon) < d$, et nous détaillons ci-dessous la structure de données, fondée sur le LSH, que nous allons utiliser pour résoudre le problème :

> Étant donnés deux paramètres $\tau, k \in \mathbb{N}^*$ à fixer plus tard, nous tirons indépendamment τ fonctions de hachage aléatoires $g^1, \cdots, g^\tau \in \mathcal{G}_k$, où $\mathcal{G}_k$ est la famille de fonctions de la question 0.5. Puis, nous construisons τ tables de hachage distinctes $H^1, \cdots, H^\tau$. Enfin, **nous insérons chaque point $p_i \in P$ dans chacune des tables**, en utilisant la fonction de hachage g^j pour l'insertion dans la table H^j. Notez que, g^j étant à valeurs dans $\mathbb{Z}^k$, la table H^j prend la forme d'un tableau à k dimensions ; on supposera pour simplifier que ce tableau est infini, indexé par $\mathbb{Z}^k$ tout entier, afin de ne pas avoir à gérer de modulos dans les calculs et l'analyse. Pour gérer les collisions éventuelles, nous adoptons la technique du chaînage décrite au début du problème. Ainsi, à la fin du processus, chaque table de hachage H^j contient une copie des n points de P, indexés par la fonction g^j et chaînés lorsqu'il y a collision.

À l'aide de cette structure de données nous pouvons répondre aux requêtes de (r, ε)-voisinage sur le nuage de points P, pour $1 \leq r < r(1+\varepsilon) < d$, en appliquant la procédure suivante :

> Étant donné un point de requête $q \in X$, nous itérons sur les tables de hachage $H^1, \cdots, H^\tau$: pour chaque table H^j, nous itérons sur la liste des points de P stockés à l'indexe $g^j(q)$: si l'un de ces points (mettons p_i) vérifie $\|q - p_i\|_1 \leq r(1+\varepsilon)$, alors nous arrêtons la procédure et répondons `OUI` à la requête ; sinon, nous poursuivons l'itération. Comme le nombre total de points de P en collision avec q sur l'ensemble des τ tables de hachage peut s'avérer trop grand (plus grand que n), **nous interrompons la procédure après que 2τ points de P au total (comptés avec multiplicité, c'est-à-dire qu'un point de P considéré dans m tables différentes compte pour m points) ont été vérifiés** et nous répondons alors `NON` à la requête. Nous répondons également `NON` si l'itération sur les tables de hachage se termine sans qu'aucun point p_i vérifié n'ait satisfait la condition $\|q - p_i\|_1 \leq r(1+\varepsilon)$.

Question 0.6. *En supposant que l'évaluation de chaque fonction $h \in \mathcal{H}$ sur le point q prend un temps constant, de même que l'accès à une case quelconque des tables H^j, montrez que*

la procédure décrite ci-dessus pour répondre à une requête de (r,ε)-voisinage s'exécute en temps $O(\tau(k+d))$. Pour cela vous négligerez le temps de lire l'entrée (q,r,ε) de la requête.

Question 0.7. *Montrez que la réponse fournie par la procédure est correcte de manière déterministe lorsque* $\min_{1\leq i\leq n} \|q-p_i\|_1 > r(1+\varepsilon)$.

On se place maintenant dans le cas où il existe un point $p_{i_0} \in P$ tel que $\|q-p_{i_0}\|_1 \leq r$. La réponse à la requête doit alors être `OUI`, et montrons que c'est bien la réponse fournie par notre procédure avec une probabilité non-nulle.

Question 0.8. *Montrez que*

$$\Pr\left[g^j(q) = g^j(p_{i_0}) \text{ pour au moins un indice } 1\leq j\leq \tau\right] \geq 1-\Big(1-\pi_1(r)^k\Big)^\tau$$

où l'on pose $\pi_1(r) = 1-r/d$ comme à la section 0.1.

Question 0.9. *Montrez que*

$$\Pr\left[\#\left\{(i,j) \mid \|q-p_i\|_1 > r(1+\varepsilon) \text{ et } g^j(q)=g^j(p_i)\right\} \geq 2\tau\right] \;\leq\; \frac{n\,\pi_2(r,\varepsilon)^k}{2}$$

où l'on pose $\pi_2(r,\varepsilon) = 1-r(1+\varepsilon)/d$ comme à la section 0.1. Pour cela vous pourrez utiliser l'inégalité de Markov, qui stipule que $\Pr[Y\geq\alpha] \leq \frac{\mathbb{E}[Y]}{\alpha}$ pour toute variable aléatoire réelle non-négative Y et tout réel $\alpha>0$.

Question 0.10. *Déduisez des deux questions précédentes que la probabilité que la procédure réponde `OUI` dans le cas où $\min_{1\leq i\leq n}\|q-p_i\|_1 \leq r$ est au moins :*

$$1-\Big(1-\pi_1(r)^k\Big)^\tau - \frac{n\,\pi_2(r,\varepsilon)^k}{2}$$

Rappelons que $1\leq r < r(1+\varepsilon) < d$, ce qui fait que $1>\pi_1(r)>\pi_2(r,\varepsilon)>0$. On pose à présent $k=\log_{1/\pi_2(r,\varepsilon)} n$ et $\tau = n^\varrho$, où $\varrho = \frac{\ln \pi_1(r)}{\ln \pi_2(r,\varepsilon)} \in]0,1[$, et on suppose pour simplifier que k et τ sont bien des entiers strictement positifs comme convenu initialement.

Question 0.11. *Montrez que $\varrho \leq \frac{1}{1+\varepsilon}$ et $k \leq \frac{d}{1+\varepsilon}\ln n$ (pour cela vous pourrez utiliser le développement en série entière $\ln(1-t) = -\sum_{i=1}^\infty t^i/i$ pour $t\in[0,1[$). Qu'en déduisez-vous concernant la complexité en temps de la procédure, notamment sur sa dépendance en n, d et ε ?*

Question 0.12. *Montrez que la probabilité de succès de la procédure est alors au moins $1/2-1/e>0$, quelle que soit la valeur de $\min_{1\leq i\leq n}\|q-p_i\|_1$.*

0.3 Retour au calcul approché de la distance au plus proche voisin

Rappelons que notre espace X est le cube de Hamming $\{0,1\}^d$ muni de la norme ℓ^1. Nous revenons à présent au problème du calcul de la distance au plus proche voisin, ou plutôt à sa version approchée, paramétrée par un $\varepsilon>0$ fixé : étant donné un nuage de points $P=\{p_1,\cdots,p_n\}\subseteq X$, construire une structure de données permettant de calculer, pour tout point de requête $q\in X$, une valeur r telle que

$$\frac{1}{1+\varepsilon}\min_{1\leq i\leq n}\|q-p_i\|_1 \;\leq\; r \;\leq\; (1+\varepsilon)\min_{1\leq i\leq n}\|q-p_i\|_1 \tag{5}$$

Nous allons résoudre ce problème en le réduisant à sa version décisionnelle dite du (r, ε)-voisinage, étudiée à la section 0.2. Pour cela, considérons une famille $(r_l)_{-1 \leq l \leq L}$ de valeurs r possibles, donnée par la suite géométrique de raison $(1+\varepsilon)$ commençant à 1 à laquelle on ajoute la valeur 0 :

$$\begin{cases} r_{-1} = 0 \\ r_l = (1+\varepsilon)^l \quad \text{pour } 0 \leq l \leq L = \lfloor \log_{1+\varepsilon} d \rfloor - 1 \end{cases}$$

Lors d'une requête, nous allons choisir parmi ces valeurs celle à retenir en fonction des réponses aux différents problèmes du (r, ε)-voisinage pour $r = r_{-1}, \cdots, r_L$. Le cas $r = r_{-1} = 0$ doit être traité à part car la structure de données proposée à la section 0.2 pour résoudre le problème du (r, ε)-voisinage ne s'applique que pour $r \geq 1$.

Question 0.13. *Proposez une structure de données pour résoudre des requêtes de $(0, \varepsilon)$-voisinage sur le nuage de points P de manière déterministe en temps $o(n)$. Précisez l'ordre de grandeur de la complexité en temps, en fonction de n, d et ε.*

Notre structure de données pour répondre aux requêtes de calcul approché de distance au plus proche voisin parmi P est donc la suivante :

> D'abord, nous construisons la structure de données de la question 0.13 pour le cas r_{-1}. Ensuite, pour chacune des autres valeurs r_l, $0 \leq l \leq L$, nous construisons une instance séparée de la structure de données décrite à la section 0.2, avec comme paramètres $r = r_l$, $k = \log_{1/\pi_2(r,\varepsilon)} n$ et $\tau = n^{\varrho}$, où $\varrho = \frac{\ln \pi_1(r)}{\ln \pi_2(r,\varepsilon)}$, $\pi_1(r) = 1 - r/d$ et $\pi_2(r, \varepsilon) = 1 - r(1+\varepsilon)/d$. Ici le paramètre ε est fixé donc identique pour toutes les instances.

Notez que, dans le cas particulier où $r_L(1+\varepsilon) = d$, c'est-à-dire que d est une puissance entière de $1+\varepsilon$, on a $\pi_2(r_L, \varepsilon) = 0$ et donc la structure de données décrite à la section 0.2 ne peut pas être construite pour $r = r_L$. Dans ce cas on la remplace par une procédure qui répond systématiquement `OUI` pour résoudre le problème de (r_L, ε)-voisinage, comme indiqué juste après la définition du problème de (r, ε)-voisinage au début de la section 0.2. Afin d'éviter cette technicalité supplémentaire, à partir de maintenant et dans tout le reste du problème nous supposons que $r_L(1+\varepsilon) < d$, c'est-à-dire que d n'est pas une puissance entière de $1+\varepsilon$.

Avec la structure de données décrite dans l'encadré ci-dessus nous répondons aux requêtes de calcul ε-approché de la distance au plus proche voisin parmi P par la procédure suivante :

> Étant donné un point de requête $q \in X$, nous utilisons d'abord la structure de données de la question 0.13 pour répondre à une requête de $(0, \varepsilon)$-voisinage sur q, puis pour chaque rayon $r = r_0, \cdots, r_L$ indépendamment nous appliquons la procédure de la section 0.2 sur la structure de données correspondante à r pour répondre à une requête de (r, ε)-voisnage sur q. La réponse finale à la requête de calcul ε-approché de distance au plus proche voisin de q parmi P est alors le plus petit r parmi $r_{-1}, r_0, \cdots, r_L$ tel que la réponse à la requête de (r, ε)-voisinage sur q a été `OUI`. Si aucune des réponses aux requêtes de (r, ε)-voisinage sur q n'a été `OUI`, alors la réponse finale est $r = d$.

Question 0.14. *Montrez que la probabilité de succès de cette procédure est au moins $1/2 - 1/e > 0$. On rappelle qu'il y a succès lorsque la réponse r donnée par la procédure satisfait l'équation* (5).

Question 0.15. *Donnez une borne supérieure sur la complexité en temps de la procédure, en fonction de n, d et ε (pour simplifier vous pourrez supposer que $\varepsilon < 1$). Commentez votre borne.*

Question 0.16. *Peut-on ajuster la procédure de manière à calculer la distance de q à son plus proche voisin de manière exacte et non plus seulement approchée, avec une probabilité au moins constante (strictement positive), en temps sous-linéaire en n et polynômial en d ? Justifiez votre réponse.*