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.*

Locality Sensitive Hashing

Question 0.1. Par définition, $\|x-y\|_1 = \sum_{i=1}^{d} |x_i - y_i|$. Comme $x_i, y_i \in \{0,1\}$, la quantité $|x_i - y_i|$ vaut 1 si $x_i \neq y_i$ et 0 sinon. Ainsi,

$$\|x-y\|_1 = \sum_{i=1}^{d} \mathbb{1}_{x_i \neq y_i} = \#\{1 \leq i \leq d \mid x_i \neq y_i\}$$

Question 0.2. Pour tout $1 \leq i \leq d$, notons $h^{(i)}$ la projection $z \mapsto z_i$. On peut alors effectuer le calcul de probabilités suivant, dans lequel l'égalité $\Pr\left[h = h^{(i)}\right] = \frac{1}{d}$ provient du fait que la famille $\mathcal{H}$ est munie de la mesure de probabilité uniforme :

$$\begin{aligned}\Pr[h(x) = h(y)] &= \sum_{i=1}^{d} \Pr\left[h = h^{(i)}\right] \mathbb{1}_{h^{(i)}(x) = h^{(i)}(y)} = \sum_{i=1}^{d} \frac{1}{d} \mathbb{1}_{x_i = y_i} = \frac{1}{d} \sum_{i=1}^{d} \mathbb{1}_{x_i = y_i} \\ &= \frac{1}{d} \#\{1 \leq i \leq d \mid x_i = y_i\} = \frac{1}{d}\left(d - \|x_i - y_i\|_1\right) = 1 - \|x-y\|_1/d\end{aligned}$$

Le résultat s'ensuit du fait que l'on a supposé que $\|x-y\|_1 \leq r$.

Question 0.3. Le calcul de probabilités de la question précédente ne fait aucune hypothèse sur $\|x-y\|_1$, il reste donc valide ici et le résultat s'ensuit du fait que l'on suppose maintenant que $\|x-y\|_1 \geq r(1+\varepsilon)$.

Question 0.4. Quand $r = 0$ ou $\varepsilon = 0$, on a $r = r(1+\varepsilon)$ et donc $\pi_1(r) = \pi_2(r, \varepsilon)$. On ne peut donc pas parler de $(r, r(1+\varepsilon), \pi_1(r), \pi_2(r,\varepsilon))$-sensibilité car la définition 1 impose que les deux rayons (de même que les deux bornes de probabilité) soient distincts.

Question 0.5. Soient $x, y \in X$ et soit $g = (h_1, \cdots, h_k) \in \mathcal{G}$ tiré aléatoirement. Comme les projections h_i sont tirées indépendamment selon la même loi, on a :

$$\begin{aligned}\Pr[g(x) = g(y)] &= \Pr[h_i(x) = h_i(y)\ \forall 1 \leq i \leq k] \\ &= \prod_{i=1}^{k} \Pr[h_i(x) = h_i(y)] \\ &= \Pr[h_1(x) = h_1(y)]^k\end{aligned}$$

Les bornes de probabilités des questions 0.2 et 0.3 sont alors mises à la puissance k, ce qui donne le résultat.

Question 0.6. Pour chaque table de hachage H^j on calcule le vecteur $g^j(q)$, ce qui prend un temps $O(k)$ car par hypothèse chaque coordonnée du vecteur prend un temps constant à calculer. Une fois le vecteur calculé, l'accès à la case correspondante dans la table H^j prend un temps constant par hypothèse. L'itération sur les éléments de la liste stockée dans cette case prend un temps linéraire en le nombre n_j d'éléments considérés. Pour chacun d'eux, le calcul de la distance à q prend un temps $O(d)$. Au total, le temps passé sur la table H^j est donc en $O(k + d\,n_j)$. En sommant sur les τ tables de hachage, on obtient une complexité en $O(\tau k + d \sum_{j=1}^{\tau} n_j)$. Enfin, comme on impose une borne 2τ sur le nombre total de points considérés, c'est-à-dire sur la somme des n_j, on obtient une complexité en $O(\tau(k+d))$.

Question 0.7. Le test $\|q - p_i\|_1 \overset{?}{\leq} r(1+\varepsilon)$ effectué lors de l'inspection de chacun des points p_i considérés par la procédure échoue systématiquement lorsque $\min_{1 \leq i \leq n} \|q - p_i\|_1 > r(1+\varepsilon)$. Par conséquent, la réponse de la procédure est systématiquement `NON` (et donc correcte) dans ce contexte.

Question 0.8. On considère l'événement contraire et on utilise le fait que les fonctions de hachage g^j sont tirées indépendamment selon la même loi :

$$\begin{aligned}\Pr\left[g^j(q) \neq g^j(p_{i_0})\ \forall 1 \leq j \leq \tau\right] &= \prod_{j=1}^{\tau} \Pr\left[g^j(q) \neq g^j(p_{i_0})\right] \\ &= \Pr\left[g^1(q) \neq g^1(p_{i_0})\right]^\tau \\ &= \left(1 - \Pr\left[g^1(q) = g^1(p_{i_0})\right]\right)^\tau\end{aligned}$$

Comme $\|q - p_{i_0}\|_1 \leq r$ et que la famille $\mathcal{G}_k$ est $(r, r(1+\varepsilon), \pi_1(r)^k, \pi_2(r,\varepsilon)^k)$-sensible, le membre de droite est majoré par $\left(1 - \pi_1(r)^k\right)^\tau$. D'où le résultat.

Question 0.9. La quantité N qui nous intéresse ici est le nombre total de points de P éloignés de plus de $r(1+\varepsilon)$ de q qui collisionnent avec q dans au moins une table de hachage, comptés avec multiplicité (un même point pouvant collisionner avec q dans plusieurs tables de hachage). N est bien sûr une variable aléatoire et peut se réécrire de la manière suivante :

$$N := \#\left\{(i,j) \mid \|q - p_i\|_1 > r(1+\varepsilon) \text{ et } g^j(q) = g^j(p_i)\right\} = \sum_{i=1}^{n}\sum_{j=1}^{\tau} \mathbb{1}_{\|q-p_i\|_1 > r(1+\varepsilon)}\, \mathbb{1}_{g^j(q)=g^j(p_i)}$$

Par linéarité de l'espérance, l'espérance de N est égale à

$$\mathbb{E}[N] = \sum_{i=1}^{n}\sum_{j=1}^{\tau} \mathbb{1}_{\|q-p_i\|_1 > r(1+\varepsilon)}\, \mathbb{E}\left[\mathbb{1}_{g^j(q)=g^j(p_i)}\right]$$

Or, comme la famille $\mathcal{G}_k$ est $(r, r(1+\varepsilon), \pi_1(r)^k, \pi_2(r,\varepsilon)^k)$-sensible, pour tout p_i tel que $\|q-p_i\|_1 > r(1+\varepsilon)$ on a $\Pr\left[g^j(q) = g^j(p_i)\right] \leq \pi_2(r,\varepsilon)^k$ et donc $\mathbb{E}\left[\mathbb{1}_{g^j(q)=g^j(p_i)}\right] \leq \pi_2(r,\varepsilon)^k$. Il s'ensuit que

$$\mathbb{E}[N] \leq \sum_{i=1}^{n}\sum_{j=1}^{\tau} \mathbb{1}_{\|q-p_i\|_1 > r(1+\varepsilon)}\, \pi_2(r,\varepsilon)^k \leq n\,\tau\,\pi_2(r,\varepsilon)^k$$

Et par l'inégalité de Markov :

$$\Pr[N \geq 2\tau] \leq \frac{\mathbb{E}[N]}{2\tau} \leq \frac{n\,\pi_2(r,\varepsilon)^k}{2}$$

ce qui est le résultat attendu.

Question 0.10. Une condition suffisante pour que la procédure réponde `OUI` dans le cas où il existe un point $p_{i_0} \in P$ tel que $\|q - p_{i_0}\|_1 \leq r$ est que p_{i_0} collisionne avec q dans au moins une table de hachage et qu'il n'y ait pas plus de $2\tau - 1$ points de P éloign'es de plus de $r(1+\varepsilon)$ de q qui collisionnent avec q (comptés avec multiplicité). Deux remarques importantes :

- Cette condition garantit non pas que p_{i_0} lui-même va être considéré par la procédure, mais qu'au moins un point de P à distance au plus r de q va être considéré.
- Les deux évenements (p_{i_0} collisionne avec q dans au moins une table de hachage d'une part, il n'y a pas plus de $2\tau-1$ points de P éloignés de plus de $r(1+\varepsilon)$ de q qui collisionnent avec q d'autre part) ne sont a priori pas indépendants.

Ainsi donc, en posant N comme à la question précédente, on a :

$$\begin{aligned}\Pr[\text{la procédure répond } \texttt{OUI}] &\geq \Pr\left[N \leq 2\tau - 1 \text{ et } g^j(q) = g^j(p_{i_0}) \text{ pour au moins un indice } 1 \leq j \leq \tau\right] \\ &= 1 - \Pr\left[N \geq 2\tau \text{ ou } g^j(q) \neq g^j(p_{i_0}) \text{ pour tout indice } 1 \leq j \leq \tau\right] \\ &\geq 1 - \Pr[N \geq 2\tau] - \Pr\left[g^j(q) \neq g^j(p_{i_0}) \text{ pour tout indice } 1 \leq j \leq \tau\right] \\ &\geq 1 - \frac{n\,\pi_2(r,\varepsilon)^k}{2} - \left(1 - \pi_1(r)^k\right)^\tau\end{aligned}$$

où la dernière inégalité provient des deux questions précédentes tandis que l'avant-dernière provient de l'inégalité de Bool.

Question 0.11. On utilise le développement en série entière de $\ln(1-t)$ pour $t \in [0,1[$:

$$\varrho = \frac{\ln(1-r/d)}{\ln(1-r(1+\varepsilon)/d)} = \frac{\sum_{i=1}^{\infty} \frac{1}{i} (r/d)^i}{\sum_{i=1}^{\infty} \frac{1}{i} (r/d)^i (1+\varepsilon)^i}$$

Comme $1+\varepsilon \geq 1$, on a $(1+\varepsilon)^i \geq (1+\varepsilon)$ et donc

$$\varrho \leq \frac{\sum_{i=1}^{\infty} \frac{1}{i} (r/d)^i}{\sum_{i=1}^{\infty} \frac{1}{i} (r/d)^i (1+\varepsilon)} = \frac{1}{1+\varepsilon}$$

De manière similaire, on calcule :

$$\begin{aligned} k = \log_{1/\pi_2(r,\varepsilon)} n = \frac{\ln n}{-\ln(1-r(1+\varepsilon)/d)} = \frac{\ln n}{\sum_{i=1}^{\infty} \frac{1}{i} (r/d)^i (1+\varepsilon)^i} \\ \leq \frac{\ln n}{(1+\varepsilon)\left(\sum_{i=1}^{\infty} \frac{1}{i} (r/d)^i\right)} \leq \frac{\ln n}{(1+\varepsilon) r/d} \leq \frac{d}{r(1+\varepsilon)} \ln n \end{aligned}$$

ce qui implique que $k \leq \frac{d}{1+\varepsilon} \ln n$ puisque par hypothèse on a $r \geq 1$.

Ainsi, la complexité en temps de la procédure est en $O\left(\frac{d}{1+\varepsilon} n^{1/(1+\varepsilon)} \ln n\right)$. À paramètre d'approximation ε fixé, la complexité est donc sous-linéaire en n tout en étant seulement polynômiale (et même linéaire) en d, **l'approche ne subit donc pas le fléau de la dimension !** Toutefois, lorsque le paramètre ε tend vers 0, la borne sur la complexité devient linéaire en n à la limite, et un calcul de développement limité montre que la complexité en temps de l'algorithme devient elle-même linéaire en n.

Question 0.12. Si $\min_{1 \leq i \leq n} \|q - p_i\|_1 > r(1+\varepsilon)$ alors on a vu à la question 0.7 que la probabilité de succès de la procédure est 1 (en fait le succès est systématique).

Si $r(1+\varepsilon) \geq \min_{1 \leq i \leq n} \|q - p_i\|_1 > r$ alors d'après la description du problème du (r,ε)-voisinage donnée dans l'énoncé la réponse de la procédure peut être arbitraire donc elle est également correcte avec probabilité 1 (en fait de manière déterministe).

Enfin, si $r \geq \min_{1 \leq i \leq n} \|q - p_i\|_1$ alors la question 0.10 garantit que la probabilité de succès est au moins

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

Comme $k = \log_{1/\pi_2(r,\varepsilon)} n$, le dernier terme ci-dessus est égal à

$$-\frac{n\, \pi_2(r,\varepsilon)^{\log_{1/\pi_2(r,\varepsilon)} n}}{2} = -\frac{n/n}{2} = -1/2$$

Et comme $\tau = n^{\varrho}$, le terme médian ci-dessus est égal à

$$\begin{aligned} -\left(1 - \pi_1(r)^{\log_{1/\pi_2(r,\varepsilon)} n}\right)^{n^{\varrho}} &= -\left(1 - n^{\frac{\ln \pi_1(r)}{\ln 1/\pi_2(r,\varepsilon)}}\right)^{n^{\varrho}} = -\left(1 - n^{-\varrho}\right)^{n^{\varrho}} \\ &= -e^{n^{\varrho} \ln\left(1-n^{-\varrho}\right)} = -e^{-n^{\varrho} \sum_{i=1}^{\infty} \frac{1}{i} n^{-\varrho\, i}} \geq -e^{-n^{\varrho} n^{-\varrho}} = -1/e \end{aligned}$$

Ainsi, dans tous les cas la probabilité de succès de la procédure est au moins $1/2 - 1/e > 0$.

Question 0.13. Comme $0(1+\varepsilon) = 0$, on veut simplement savoir si le point de requête q coïncide avec l'un des points $p_i \in P$ ou non. Pour répondre à cette question nous avons l'embarras du choix, et au moins trois approches possibles, énumérées ci-dessous par ordre décroissant de pertinence :

1. Utiliser l'ordre lexicographique sur les coordonnées dans $\mathbb{R}^d$ pour trier les points de P, puis faire une recherche dichotomique sur ces points pour en trouver un qui coïncide avec q, si un tel point existe. La requête prend alors un temps $O(d \log n)$, le facteur d provenant de la comparaison de q avec les différents points de P considérés selon l'ordre lexicographique sur les coordonnées. Il n'y a pas de dépendance de la borne en ε car le facteur d'approximation ne joue aucun rôle ici.
2. Utiliser le hachage traditionnel et stocker les points de P dans une simple table de hachage. Pour la fonction de hachage, on peut par exemple choisir parmi celles existant pour les chaînes de caractères, en traitant tout point $x \in X$ comme une chaîne de bits. En supposant la fonction de hachage parfaite et le taux de remplissage de la table adéquat (facile à gérer car le nuage de points P est fixé), la requête prend alors un temps moyen $O(d)$, le facteur d provenant du coût de l'évaluation de la fonction de hachage en q. Cette solution est toutefois moins satisfaisante que la précédente car, bien que la réponse soit correcte de manière déterministe, le temps de calcul n'est exprimé qu'en moyenne, et en principe il pourrait devenir linéaire en n sur certaines instances.
3. S'appuyer sur notre réponse à la question 0.4 et construire la structure de données proposée dans le sujet avec des valeurs de paramètres τ, ϱ adaptées aux nouvelles valeurs de $r_1 = 0, r_2, \pi_1, \pi_2$. On peut alors espérer un temps de requête sous-linéaire en n. Toutefois, les détails restent à écrire du fait du changement des constantes. Par ailleurs, cette approche, bien qu'habile, est clairement moins efficace que les deux autres car elle permet seulement d'être légèrement sous-linéaire en n. Enfin, plus fondalement, elle ne répond correctement qu'avec une certaine probabilité au lieu de le faire de manière déterministe comme demandé dans l'énoncé.

Question 0.14. Appelons $\delta = \min_{1 \le i \le n} \|q - p_i\|_1$ la vraie distance de q à son plus proche voisin parmi P, et r la réponse de la procédure. Comme vu aux questions 0.7 et 0.13, pour tout $r_l < \delta/(1+\varepsilon)$ la réponse à la requête de (r_l, ε)-voisinage est `NON` de manière déterministe. En conséquence, on a soit $r = r_l$ pour un certain $r_l \ge \delta/(1+\varepsilon)$, soit $r = d \ge \delta$, et donc dans tous les cas $r \ge \delta/(1+\varepsilon)$.

Montrons à présent que $r \le \delta(1+\varepsilon)$ avec probabilité au moins $1/2 - 1/e$. Regardons d'abord le cas où $\delta > r_L(1+\varepsilon)$: dans ce cas, on a $r \le d \le r_L(1+\varepsilon)^2 < \delta(1+\varepsilon)$. Supposons maintenant que $\delta \le r_L(1+\varepsilon)$, et soit l_0 le plus petit l tel que $r_l \ge \delta$. Si $l_0 = -1$ alors $\delta = r_{-1} = 0$ et donc la structure de données de la question 0.13 répond `OUI` demanière déterministe à la requête de $(0, \varepsilon)$-voisinage, ce qui implique que $r = 0 = \delta$. Si au contraire $l_0 \ge 0$ alors $\delta > 0$, ce qui, dans le cube de Hamming, implique que $\delta \ge 1$. La définition de la sous-famille $(r_l)_{0 \le l \le L}$ implique alors que $\delta \le r_{l_0} < \delta(1+\varepsilon)$, et donc une condition suffisante pour que $r \le \delta(1+\varepsilon)$ est que la requête de (r_{l_0}, ε)-voisinage réponde correctement `OUI`, ce qui arrive avec probabilité au moins $1/2 - 1/e$ d'après la question 0.12.

Question 0.15. La borne sur la complexité dépend du choix de structure de données à la question 0.13 pour gérer le cas $r = r_{-1} = 0$. Appelons $C_0(n, d, \varepsilon)$ la complexité en temps de la procédure correspondante pour répondre à la requête de $(0, \varepsilon)$-voisinage.

Pour chaque cas $r = r_l$, $0 \le l \le L$, la question 0.11 nous donne une complexité en $O(\frac{d}{1+\varepsilon}\, n^{1/(1+\varepsilon)} \ln n)$. Comme il y a $L+1$ cas à traiter, avec (pour $\varepsilon < 1$)

$$L \le \log_{1+\varepsilon} d = \frac{\ln d}{\ln(1+\varepsilon)} \le \frac{\ln d}{\varepsilon - \varepsilon^2/2} \le \frac{2 \ln d}{\varepsilon}$$

on obtient une complexité totale en $O\left(C_0(n,d,\varepsilon) + \frac{d \ln d}{\varepsilon(1+\varepsilon)}\, n^{1/(1+\varepsilon)} \ln n\right)$ pour l'ensemble des requêtes de (r,ε)-voisinage à traiter. La détermination du plus petit r_l tel que le résultat de la requête est `OUI` peut être faite à la volée, par exemple en itérant sur les r_l par ordre croissant. Ainsi, la borne ci-dessus est valable également pour l'ensemble de la procédure de calcul approché de la distance au plus proche voisin de q parmi P.

On suppose maintenant que la première structure de données proposée à la question 0.13 est utilisée pour traiter le cas $r = r_{-1} = 0$. Le terme $C_0(n,d,\varepsilon)$ est alors dominé par l'autre terme dans la borne de complexité. Dans ce cas, comme à la question 0.11, à $\varepsilon > 0$ fixé la borne est sous-linéaire en n tout en étant polynômiale (en fait quasi-linéaire) en d, ce qui fait que l'approche ne subit pas le fléau de la dimension. Toutefois, contrairement à la question 0.11, la borne ne devient pas seulement linéaire en n lorsque ε tend vers 0, en fait elle diverge à l'infini.

Question 0.16. L'observation-clé est que la distance prend des valeurs entières dans le cube de Hamming. En conséquence, en remplaçant la suite géométrique $(r_l)_{-1\leq l\leq L}$ précédente par la suite arithmétique $(r_l)_{0\leq l\leq d}$ définie par $r_l = l$ pour tout l, et en prenant $\varepsilon < 1/d$ (qui assure que $r(1+\varepsilon) - r = r\varepsilon \leq d\varepsilon < 1$ pour tout $0 \leq r \leq d$), chaque requête de (r_l,ε)-voisinage détermine si $\min_{1\leq i\leq n} \|q-p_i\|_1 \leq l$ ou bien $\min_{1\leq i\leq n} \|q-p_i\|_1 > l$. Ainsi, notre procédure calcule $\min_{1\leq i\leq n} \|q-p_i\|_1$ exactement. La probabilité de succès reste inchangée (par le même argument qu'à la question 0.14), égale à $1/2-1/e > 0$. La borne de complexité devient $O\left(d^2\, n^{d/(d+1)} \ln n\right)$ en prenant ε de l'ordre de $1/d$ (par exemple $\varepsilon = 1/2d$). Cette borne reste sous-linéaire en n et polynômiale en d.