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.