

Contrôle écrit avec **corrigé** INF442

— Traitement des données massives —

École Polytechnique, Département informatique

Léo Liberti

Frank Nielsen

Promotion X2013
Mercredi 10 juin 2015
9h-12h

Informations générales

1. Les quatre exercices sont tous *indépendants* les uns des autres, et peuvent donc être traités dans n'importe quel ordre sur la copie. Ils ont une difficulté (estimée) différente.
2. On appréciera la *clarté* (en premier lieu) et la *concision* des réponses (en second lieu). Tout document du cours INF442 autorisé (polycopié, memento C++ et planches des cours en amphithéâtre).
3. Tous les problèmes sont structurés en plusieurs sous-problèmes. Indiquez précisément la numérotation des sous-problèmes que vous traitez sur vos feuilles d'examen.
4. Cet examen peut être fini en 2h45m. Prenez donc 10 minutes de votre temps pour bien lire les énoncés avant de commencer, et gardez au moins 5 minutes à la fin pour la relecture finale.
5. Vos correcteurs ne sont pas des génies graphologues : toute solution illisible ou incompréhensible sera invalide *sans possibilité d'appel*.
6. Si votre erreur est dans la catégorie **erreur communes** (listées tout au long des solutions en bas), ce n'est pas la peine de porter plainte ou de chercher de se faire revoir la note, qui ne sera pas changée. Pour toute demande de revision de note, il faut savoir que la note peut être revue en hausse mais aussi en baisse, si d'autres soucis apparaissent pendant la revision.

Barème

Les problèmes ci-dessous comptent pour la note finale de la façon suivante :

problème	%	difficulté	temps estimé
1	10%	faible	30 minutes
2	20%	moyenne	30 minutes
3	30%	moyenne	45 minutes
4	40%	grande	60 minutes

Les pourcentages des questions de chaque sous-problème sont donnés dans le texte des problèmes correspondants.

Problème 1 : Accélération et loi d'Amdahl

Difficulté : faible ; temps conseillé : 30 minutes ; barème : 10%.

Soit \mathcal{P} un programme séquentiel qui est à 75% parallélisable.

- (10%) Quelle est l'accélération théorique pour $P = 3$ processeurs, et $P = 4$ processeurs ?
- (10%) Quelles sont l'accélération et l'efficacité théoriques asymptotiques quand $P \rightarrow +\infty$?

Soit maintenant donné un programme séquentiel $\mathcal{P} = \mathcal{P}_1; \mathcal{P}_2$ qui est constitué de deux routines consécutives \mathcal{P}_1 et \mathcal{P}_2 . La première routine est parallélisable à 50% et demande un temps t_1 . La seconde routine est parallélisable à 75% et demande un temps t_2 . De plus, pour l'algorithme séquentiel, la seconde routine prend toujours deux fois plus de temps que la première routine : $t_2 = 2t_1$.

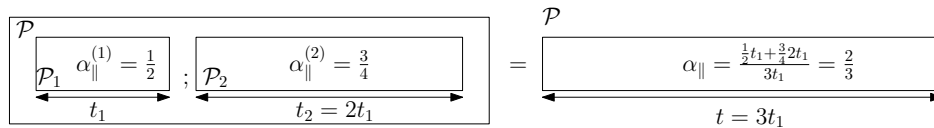
- (40%) Quelle est l'accélération que l'on obtient pour P processeurs ?
- (20%) En déduire une valeur asymptotique de l'accélération quand $P \rightarrow \infty$.

On considère un algorithme parallèle sur un jeu de données de taille n qui a une complexité en temps parallèle pour P processeurs de $O_{\parallel} \left(\frac{n^2}{\sqrt{P}} \right)$. On supposera que les n données tiennent toutes dans les mémoires locales des processeurs. Vous disposez en pratique d'un cluster de machines avec seulement $P_0 \ll P$ processeurs pour faire tourner le programme avec P processus (`mpirun -np P programme`) (avec P supposé un multiple de P_0 : $P \bmod P_0 = 0$).

- (20%) En simulant un "cluster virtuel" à P machines sur un "cluster physique" à P_0 machines (en mappant plusieurs processus sur un même processeur), quelle est la complexité du temps de calcul de l'algorithme parallèle avec P processus sur ce cluster physique à P_0 machines ?

Solution

- On applique la loi d'Amdahl $S(P) = \frac{1}{\alpha_{\text{seq}} + \frac{1-\alpha_{\text{seq}}}{P}}$ avec $\alpha_{\text{seq}} = 1 - \alpha_{\parallel} = \frac{1}{4}$. On obtient $S(3) = 2$ et $S(4) = \frac{16}{7} \simeq 2,29$.
- L'accélération asymptotique est $S = \frac{1}{\alpha_{\text{seq}}} = 4$ et l'efficacité asymptotique est 0.
- On calcule la proportion globale parallélisable comme $\alpha_{\parallel} = \frac{\alpha_{\parallel}^{(1)} t_1 + 2\alpha_{\parallel}^{(2)} t_1}{3t_1} = \frac{2}{3}$. On a donc $\alpha_{\text{seq}} = 1 - \alpha_{\parallel} = \frac{1}{3}$, et on applique la loi d'Amdahl : $S(P) = \frac{1}{\frac{1}{3} + \frac{2}{3P}} = \frac{3}{1+2/P}$.



- L'accélération asymptotique est 3.
- On mappe sur chaque processeur physique $\frac{P}{P_0}$ processus virtuels en *time sharing*. La complexité parallèle est donc en $O_{\parallel} \left(\frac{n^2}{\sqrt{P}} \frac{P}{P_0} \right) = O_{\parallel} \left(n^2 \frac{\sqrt{P}}{P_0} \right)$.

Problème 2 : Inférence statistique en parallèle avec MPI

Difficulté : moyenne ; temps conseillé : 30 minutes ; barème : 20%.

Soit $X = \{x_1, \dots, x_n\} \subset \mathbb{R}$ un grand jeu de données supposés être des échantillons aléatoires indépendamment tirés d'une loi normale univariée $N(\mu, \sigma^2) : x_1, \dots, x_n \sim N(\mu, \sigma^2)$. On rappelle que les estimateurs non-biaisés du maximum de vraisemblance, pour, respectivement, la moyenne et la variance, sont :

$$\hat{\mu}_n = \frac{1}{n} \sum_{i=1}^n x_i \quad (1)$$

$$\hat{\sigma}_n^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \hat{\mu}_n)^2. \quad (2)$$

1. **(25%)** Montrez que $\hat{\mu}_n$ et $\hat{\sigma}_n^2$ peuvent se calculer uniquement avec n , $S_1 = \sum_{x \in X} x$ et $S_2 = \sum_{x \in X} x^2$. Donner les formules de $\hat{\mu}_n$ et $\hat{\sigma}_n^2$ explicitement en fonction de n , S_1 et S_2 (et d'aucune autre variable).
2. **(75%)** On suppose les données X stockées sur un des ordinateurs dans un cluster distribué avec p ordinateurs. En utilisant les primitives MPI `scatter()` et `reduce()`, écrire le pseudocode d'un algorithme distribué qui :
 - partage les données de manière équilibrée sur chacun des ordinateurs du cluster,
 - calcule $\hat{\mu}_n$ et $\hat{\sigma}_n^2$ de manière distribuée pour l'ensemble total X .

Solution

1. Soient $S_1 = \sum_{i \leq n} x_i$ et $S_2 = \sum_{i \leq n} x_i^2$. On a :

$$\begin{aligned} \hat{\mu}_n &= \frac{S_1}{n} \\ \hat{\sigma}_n^2 &= \frac{1}{n-1} \sum_{i \leq n} (x_i - \hat{\mu}_n)^2 = \frac{1}{n-1} \left(S_2 + n\hat{\mu}_n^2 - 2\hat{\mu}_n \sum_{i \leq n} x_i \right) = \\ &= \frac{S_2 + \frac{S_1^2}{n} - 2\frac{S_1}{n}S_1}{n-1} = \frac{S_2 + \frac{1}{n}(S_1^2 - 2S_1^2)}{n-1} = \frac{1}{n-1} \left(S_2 - \frac{S_1^2}{n} \right) \end{aligned}$$

2. Le pseudocode suivant est un exemple de solution.

```


$p$  = number of processes;  

 $r$  = rank of the current process;  

master = rank of the root process;  

 $\text{chunksize} = n/p$ ;  

 $\text{lastchunksize} = n \bmod p$ ;  

if  $\text{lastchunksize} = 0$  then  

     $\text{chunksize} \leftarrow \text{chunksize} + 1$ ;  

end  

 $m = p \times \text{chunksize}$ ;  

resize  $X$  to have  $m$  vectors;  

pad with zero vectors the last  $m - n$  vectors of  $X$ ;  

let  $Y_r$  be an array of  $\text{chunksize}$  vectors;  

scatter( $X, Y, \text{master}$ );  

if  $r = p - 1$  then  

     $\text{chunksize} = \text{lastchunksize}$ ;  

end  

 $\text{slaveS1} = 0$ ;  

 $\text{slaveS2} = 0$ ;  

for  $i \leq \text{chunksize}$  do  

     $\text{slaveS1} \leftarrow \text{slaveS1} + Y_r[i]$ ;  

     $\text{slaveS2} \leftarrow \text{slaveS2} + Y_r^2[i]$ ;  

end  

reduce( $\text{slaveS1}, S_1, \text{master}, \text{MPI\_SUM}$ );  

reduce( $\text{slaveS2}, S_2, \text{master}, \text{MPI\_SUM}$ );  

if  $r = \text{master}$  then  

     $\hat{\mu}_n = S_1/n$ ;  

     $\hat{\sigma}_n = (S_2 - S_1^2/n)/(n - 1)$ ;  

end


```

Problème 3 : Communication sur l'hypercube

Difficulté : moyenne. Temps conseillé : 45 minutes ; barème : 30%.

Dans ce problème, vous pouvez faire référence aux fonctions suivantes :

- les opérateurs binaires Booléens AND, OR, XOR, NOT ;
- `bin`, qui transforme un entier dans son encodage en base 2 ;
- `Gray`, qui transforme un entier dans son code de Gray ;
- 2^i , la i -ème puissance de 2 ;
- `dec`, qui transforme un vecteur binaire dans l'entier correspondant ;
- `leftmost`, qui donne la composante la plus à gauche d'un vecteur quelconque : par exemple, $\text{leftmost}((x_4, x_5, x_6, x_7)) = x_4$;
- `significant`(g), définie pour tout argument $g \neq 0$, qui donne l'index i de la composante non-zéro la plus à gauche dans le vecteur binaire $g = (g_{d-1}, \dots, g_0)$: par exemple, en prenant $g = (0, 1, 0, 1)$ et $d = 4$, on a $\text{significant}((0, 1, 0, 1)) = d - 2 = 2$;
- `rightfill`, qui donne le vecteur binaire obtenu en changeant en 1 tous les zéros les plus à droite d'un vecteur binaire (g_{d-1}, \dots, g_0) donné : par exemple, $\text{rightfill}((0, 1, 0, 0)) = (0, 1, 1, 1)$.

1. **(3%)** Donnez un exemple de topologie irrégulière, et une de topologie régulière de degré 3, avec 4 sommets. Est-ce que l'hypercube a une topologie régulière ? Combien de voisins à chaque nœud de l'hypercube de dimension d ?
2. **(14%)** Donnez deux exemples de topologie régulière de degré 3 : une avec 12 sommets et l'autre avec 16 sommets. Généralisez à une famille de topologies régulières de degré 3 avec $4k$ sommets pour tout $k \geq 3$.
3. **(3%)** Calculez la distance de Hamming entre $(0, 1, 0, 1)_2$ et $(1, 1, 1, 0)_2$.
4. **(13%)** Quels sont les voisins du nœud $(0, 1, 1, 0)_2$ dans la topologie de l'hypercube avec le code de Gray ? Plus généralement, étant donné un nœud étiqueté $g = (g_{d-1}, \dots, g_0) \in \mathbb{F}_2^d$ dans le code de Gray, donnez la formule d'une fonction $\text{neighbor}_i : \mathbb{F}_2^d \rightarrow \mathbb{F}_2^d$ pour obtenir l'étiquetage de son i -ème voisin (pour tout $i \in \{0, \dots, d-1\}$).
5. **(27%)** Dessinez un carré en 2D ($d = 2$), et étiquetez les nœuds avec leur code de Gray. Considérez la fonction f , qui associe un vecteur binaire à un intervalle à bornes entières, comme suit :

$$\begin{array}{ll} (0, 0) & \rightarrow [0, 3] \\ (0, 1) & \rightarrow [1, 1] \\ (1, 0) & \rightarrow [2, 3] \\ (1, 1) & \rightarrow [3, 3] \end{array}$$

Maintenant dessinez un cube en 3D ($d = 3$), et étiquetez les nœuds avec leur code de Gray. Considérez l'extension de la fonction f au cas $d = 3$:

$$\begin{array}{ll} (0, 0, 0) & \rightarrow [0, 7] \\ (0, 0, 1) & \rightarrow [1, 1] \\ (0, 1, 0) & \rightarrow [2, 3] \\ (0, 1, 1) & \rightarrow [3, 3] \\ (1, 0, 0) & \rightarrow [4, 7] \\ (1, 0, 1) & \rightarrow [5, 5] \\ (1, 1, 0) & \rightarrow [6, 7] \\ (1, 1, 1) & \rightarrow [7, 7] \end{array}$$

Généralisez la fonction f au cas d'un hypercube quelconque en d dimensions : c'est-à-dire, donnez une formule, valide pour tout entier d , pour associer les vecteurs binaires en \mathbb{F}_2^d à des intervalles appropriés, cohérents avec les intervalles donnés dans les tableaux en ci-dessus.

[C'est évident qu'il y a plusieurs solutions possibles à cet exercice ; on veut évaluer ici votre capacité de *généralisation* et d'*abstraction* dans l'identification d'une extension de f qui soit cohérente et utile pour le reste de l'exercice.]

6. (40%) Le processus maître sur un cluster de 2^d processeurs en topologie d'hypercube en dimension d contient le tableau $x = (x_0, \dots, x_{p-1})$, où $p = 2^d$. Donnez le pseudo-code d'un algorithme **scatter** (diffusion personnalisée) sur l'hypercube (avec nœud racine 0) grâce auquel l' i -ème processeur reçoit la donnée x_i . Vous pouvez utiliser la fonction f définie au point 5.

Solution

1. Un graphe complet sur 4 sommets est régulier de degré 3. Un arbre binaire complet a une topologie irrégulière : sa racine a degré 2, tandis que les feuilles ont degré 1. L'hypercube en d dimensions a une topologie régulière de degré d , avec d voisins par sommet.

Erreurs communes : (a) la *régularité* d'un graphe dépend du degré, pas de l'orientation des arêtes ; (b) la "grille" est une famille de graphes, dont la plupart sont effectivement irréguliers, mais pas tous : par exemple la grille sur 4 sommets est un carré, qui est régulier de degré 2 ; (c) de même pour l'"étoile" : car l'étoile sur un seul sommet est régulière de degré zéro. En général, porter une entière famille de graphes en tant qu'exemple marche seulement si TOUS les graphes dans la famille feraient un bon exemple ; et souvent, les cas extrêmes (ou triviaux) à 0 ou 1 sommet sont réguliers. Et (mais ceci n'est pas une erreur), pourquoi plus de la moitié des élèves écrivent que "les sommets jouent le même rôle" ? Dans les graphes, les sommets n'ont aucun rôle, ce n'est que quand on définit des algorithmes sur ces graphes que les sommets pourraient "jouer un rôle", et encore, il faut le définir formellement, ce rôle, avant de pouvoir en parler ! Donc, étant cette phrase une phrase vide du point de vue formel, je l'ai toujours ignorée.

2. Le graphe *flower snark* J_k se construit sur k étoiles

$$(\{O_i, A_i, B_i, C_i\}, \{\{O_i, A_i\}, \{O_i, B_i\}, \{O_i, C_i\}\})$$

pour tout $i \leq k$, en connectant O_1, \dots, O_k avec un cycle simple, A_1, \dots, A_k avec un autre cycle simple, et $B_1, \dots, B_k, C_1, \dots, C_k$ avec un troisième cycle simple (qui sera de longueur double par rapport aux autres). Les J_k sont tous réguliers de degré 3 et forment une famille infinie. J_3 a 12 sommets (3 étoiles à 4 sommets chacune), et J_4 en a 16 (4 étoiles à 4 sommets chacune).

Le flower snark n'est pas la seule réponse possible à cette question : il y a aussi l'anneau de taille $4k$ avec toutes les diamètres, ainsi que d'autres constructions.

Erreurs communes : (a) dire "trivialement on généralise le cas à 12 sommets au cas à 16 et à $4k$ " ne suffit pas à convaincre les correcteurs : il faut donner des dessins des cas à 12 et 16 sommets, et une description analytique de la construction généralisée à $4k$ sommets ; (b)

décrire une famille de graphes par un dessin incluant trois points de suspension ici et là, censés donner l'idée de séquence d'éléments similaires, n'est pas une réponse admissible ; (c) comme nous étudions les réseaux de processeurs, et qu'un tel réseau est complètement inutile s'il n'est pas connexe, on fait l'hypothèse très raisonnable que les exemples doivent être *connexes* (tout exemple de réseau non connexe sera considéré invalide) ; (d) l'*analogie* n'est pas une preuve mathématique acceptée : dire qu'on construit une certaine famille de graphes "d'une manière similaire" à la construction d'une autre famille donnée peut servir à un enseignant pour faire comprendre un principe, mais pas à un élève pour démontrer à son enseignant avoir compris ce principe de manière précise et formelle ; (d) la géométrie et la topologie sont des disciplines bien distinctes : donner des instructions *strictement* géométriques (impliquant cercles, carrés, oppositions diamétrales et rotations, entre autres) pour décrire un graphe ne suffit pas, car on doit faire l'hypothèse qu'il existe un plongement de la topologie qu'on cherche de définir dans un espace géométrique, ce qu'il faut démontrer (et pour cela il est nécessaire de fournir une description entièrement topologique du graphe en question !, donc "back to square one") — j'ai toléré le mélange de descriptions géométriques et topologiques, si ces dernières suffisaient à construire la famille : heureusement, certains mots géométriques sont aussi définies par les graphes, c'est ainsi que j'ai pu accorder à certains élèves le bénéfice du doute.

3. La distance de Hamming est le nombre de composantes différentes dans entre deux vecteurs binaires, en particulier $\text{dist}_H((0, 1, 0, 1), (1, 1, 1, 0)) = 3$.
4. Quand les sommets d'un hypercube sont numérotés en utilisant le code de Gray, tout pair de sommets voisins ont une distance de Hamming égale à 1. Les voisins de $(0, 1, 1, 0)$ sont donc, en ordre : $(0, 1, 1, 1)$, $(0, 1, 0, 0)$, $(0, 0, 1, 0)$, $(1, 1, 1, 0)$. Si $g = (g_{d-1}, \dots, g_0)$ est un élément d'un code de Gray, la formule cherchée est :

$$\forall i \in \{0, \dots, d-1\} \quad \text{neighbor}_i(g) = g \text{ XOR } \text{bin}(2^i).$$

Erreurs communes : une *formule* n'est pas la même chose qu'un pseudocode !

5. Pour tout vecteur binaire τ , la formule est :

$$[\text{dec}(\tau), \text{dec}(\text{rightfill}(\tau))].$$

Une autre formule valide peut s'écrire avec le bit 1 moins significatif (c'est-à-dire le plus à droite), renvoyé par la fonction `insignificant()`

$$[\text{dec}(\tau), \text{dec}(\tau) + 2^{\text{insignificant}(\tau)} - 1].$$

Erreurs communes : (a) une *formule* n'est pas la même chose qu'un pseudocode ; (b) un ensemble d'équations récursives ne sont pas la même chose qu'un formule ; (c) la borne supérieure de l'intervalle cherché n'est pas une fonction de `significant(g)`, car ça renvoie l'index de la composante non-zéro de g la plus à gauche, c'est-à-dire la plus significative, autrement dit si on écrit $\text{dec}(g) = \sum_{j=0}^{d-1} g_j 2^j$, `significant(g)` est le j tel que $g_j 2^j$ est *maximum* entre tous les termes de la somme. Par exemple, si $d = 3$ et $g = (1, 0, 1)$ (i.e. $\text{dec}(g) = 5$), l'intervalle cherché est $[5, 5]$, mais `significant(g) = 2` (car $(g_2, g_1, g_0) = (1, 0, 1)$ et l'index de la composante non-zéro la plus à gauche est 2), donc $\text{dec}(g) + 2^{\text{significant}(g)} - 1 = 5 + 2^2 - 1 = 8$, qui n'est même pas dans l'intervalle $[0, 7]$.

6. Voici une implementation de l'algorithme de **scatter** sur l'hypercube. Il faut le deployer sur le processus 0 (maître) ; la donnée x_p sera stockée dans la variable locale χ sur le processus p .

```

Data: le nœud maître doit être 0
Result: scatter( $d, x, \chi$ )
 $p \leftarrow \text{world.rank}()$ ;
 $g \leftarrow \text{Gray}(p)$ ;
if  $p \neq 0$  then
    //  $g$  receives data  $y$  from neighbor
     $i \leftarrow \text{significant}(g)$ ;
    receive(neighbor $_i(g), y$ );
    //  $\chi$  contains a local copy of  $x_p$ 
     $\chi \leftarrow \text{leftmost}(y)$ ;
else
    |  $i \leftarrow d$ ;
end
for  $\tau \in (\text{neighbor}_{i-j}(g) \mid 1 \leq j < i)$  do
    | // send correct chunk of data to neighbor
    | send(  $\tau, (x_j \mid \text{dec}(\tau) \leq j \leq \text{dec}(\text{rightfill}(\tau)))$  );
end

```

Et une autre implementation (faite par un élève).

```

Data: le nœud maître doit être 0
Result: scatter( $d, x, \chi$ )
for  $i \in (d, d-1, \dots, 1)$  do
    if  $p \bmod 2^i = 0$  then
        |  $\tau \leftarrow p + 2^{i-1}$ ;
        | send(  $\tau, (x_j \mid \text{dec}(\tau) \leq j \leq \text{dec}(\text{rightfill}(\tau)))$  );
    else if  $p \bmod 2^i = 2^{i-1}$  then
        |  $\phi \leftarrow p - 2^{i-1}$ ;
        | receive(  $\phi, (x_j \mid \text{dec}(\phi) \leq j \leq \text{dec}(\text{rightfill}(\phi)))$  );
        |  $\chi \leftarrow x_{\text{dec}(\phi)}$ ;
    end
end

```

Erreurs communes : (a) une description en Français (ou autre langage naturel) n'est pas un pseudocode; (b) un dessin n'est pas un pseudocode; (c) on ne peut pas implementer la fonction **scatter** en fonction d'elle même sans fournir un "base case" (et en tout cas la bonne implementation n'est pas recursive); (c) confondre le calcul distribué avec la récursion ne mène pas à la bonne solution; (d) s'il y a des **send()** alors il doit y avoir les **receive()** correspondants; (e) tout **send()** requiert au moins 2 arguments : *quoi* envoyer, et à *qui* (symétriquement, tout **receive()** requiert quoi recevoir, et par *qui*); (f) on ne peut pas définir une variable du pseudocode en Français dans un commentaire.

Problème 4 : Centroïde et variance

Difficulté : importante ; temps conseillé : 1h ; barème : 40%.

Pour tout ensemble fini $Z \subseteq \mathbb{R}^d$, on note $\mathbf{c}(Z) \stackrel{\text{def}}{=} \frac{1}{|Z|} \sum_{z \in Z} z$ le centre de masse (ou centroïde) de Z , avec $|Z|$ la cardinalité de l'ensemble Z . Pour deux vecteurs $x = (x^{(1)}, \dots, x^{(d)})$ et $y = (y^{(1)}, \dots, y^{(d)})$ de \mathbb{R}^d , on définit le produit scalaire $x \cdot y = \sum_{i=1}^d x^{(i)} y^{(i)}$, la norme Euclidienne au carré $\|x\|^2 = x \cdot x$, et la *distance Euclidienne au carré* entre eux par

$$\|x - y\|^2 \stackrel{\text{def}}{=} (x - y) \cdot (x - y) = \|x\|^2 - 2x \cdot y + \|y\|^2. \quad (3)$$

Soient $X, Y \subseteq \mathbb{R}^d$ deux ensembles non-vides tels que $X \cap Y = \emptyset$.

1. (7.5%) Démontrez que :

$$\mathbf{c}(X \cup Y) = \frac{|X|}{|X| + |Y|} \mathbf{c}(X) + \frac{|Y|}{|X| + |Y|} \mathbf{c}(Y).$$

2. (7.5%) Soit $\mathbf{v}(Z) \stackrel{\text{def}}{=} \sum_{z \in Z} \|z - \mathbf{c}(Z)\|^2$. Démontrez que :

$$\mathbf{v}(Z) = \sum_{z \in Z} \|z\|^2 - |Z| \|\mathbf{c}(Z)\|^2. \quad (4)$$

3. (20%) Soit $\Delta(X, Y) \stackrel{\text{def}}{=} \mathbf{v}(X \cup Y) - \mathbf{v}(X) - \mathbf{v}(Y)$. Démontrez que :

$$\Delta(X, Y) = \frac{|X||Y|}{|X| + |Y|} \|\mathbf{c}(X) - \mathbf{c}(Y)\|^2.$$

4. On étend la définition de la variance $\mathbf{v}(Z)$ (non-normalisée) pour qu'elle soit définie par rapport à un point $x \in \mathbb{R}^d$ quelconque :

$$\mathbf{v}(Z, x) \stackrel{\text{def}}{=} \sum_{z \in Z} \|z - x\|^2.$$

Remarquez que la définition précédente est le cas particulier $x = \mathbf{c}(Z)$, c'est-à-dire $\mathbf{v}(Z) = \mathbf{v}(Z, \mathbf{c}(Z))$.

- (a) (20%) Démontrez d'abord que :

$$\mathbf{v}(Z, x) = \mathbf{v}(Z) + |Z| \|\mathbf{c}(Z) - x\|^2. \quad (5)$$

- (b) (20%) En utilisant l'Eq. (5), démontrez ensuite que :

$$\mathbf{v}(Z) \leq \min_{x \in Z} \mathbf{v}(Z, x) \leq 2\mathbf{v}(Z).$$

5. Le *médioïde* d'un cluster $C \subseteq \mathbb{R}^d$ est défini comme $\arg \min_{x \in C} \sum_{z \in C} \|z - x\|^2$. On supposera dans la suite tous les médioïdes calculés uniques (ce n'est pas vrai en absolu si on considère, par exemple, pour un cluster les trois sommets d'un triangle équilatéral ; En cas de non-unicité,

on ordonne lexicographiquement les points pour le $\arg \min$ ou on perturbe initialement tous les points par un petit bruit aléatoire). Considérez l'algorithme de Lloyd en prenant en tant que prototype représentatif de chaque cluster C le médioïde à la place du centroïde, mais avec la même fonction objectif à minimiser, c'est-à-dire la somme, sur tout cluster, des distances intra-cluster entre les points du cluster et le prototype.

- (a) (7.5%) Démontrez que l'algorithme qui minimise la fonction objectif converge monotonement à partir de n'importe quelle configuration des k prototypes.
- (b) (7.5%) Bornez supérieurement (et asymptotiquement) le nombre maximal d'itérations avant convergence en terme du nombre des points n dans le *data set*, et du nombre de clusters k .
- (c) (10%) Démontrez aussi que la valeur de la solution optimale du problème k -moyennes en utilisant les médioïdes ne dépasse jamais le double de la valeur de la solution optimale du problème des k -moyennes classique (avec les centroïdes).

Solution

Une **erreur commune** a presque toutes les questions dans cet exercice, c'est de ne pas donner suffisamment de passages logiques pour que je puisse évaluer votre compréhension. Comme il s'agit pour la plupart de preuves, dont l'élève connaît hypothèse et thèse, si je vois des passages nontriviaux au milieu d'une dérivation logique, je vais devoir penser que l'élève n'a su que faire quelque passage à partir du début, quelque passage à l'arrière à partir de la fin, et un bricolage au milieu.

1. Comme $X \cap Y = \emptyset$, on a

$$\sum_{z \in X \cup Y} \|z\|^2 = \sum_{z \in X} \|z\|^2 + \sum_{z \in Y} \|z\|^2, \quad (6)$$

et donc :

$$\begin{aligned} c(X \cup Y) &= \frac{1}{|X \cup Y|} \sum_{z \in X \cup Y} z = \\ (\text{par } X \cap Y = \emptyset) &= \frac{1}{|X| + |Y|} \left(\sum_{z \in X} z + \sum_{z \in Y} y \right) = \\ (\text{par définition de } c(\cdot)) &= \frac{|X|}{|X| + |Y|} c(X) + \frac{|Y|}{|X| + |Y|} c(Y). \end{aligned}$$

2. On a :

$$\begin{aligned}
v(Z) &= \sum_{z \in Z} \|z - c(Z)\|^2 = \\
(\text{par l'Eq. (3), 2ème partie}) &= \sum_{z \in Z} \|z\|^2 - 2 \sum_{z \in Z} (z \cdot c(Z)) + \sum_{z \in Z} \|c(Z)\|^2 = \\
(\text{termes indép. de } z \text{ hors des sommes}) &= \sum_{z \in Z} \|z\|^2 - 2 \left(\sum_{z \in Z} z \right) \cdot c(Z) + |Z| \|c(Z)\|^2 = \\
(\text{par définition de } c(\cdot)) &= \sum_{z \in Z} \|z\|^2 - 2|Z|(c(Z) \cdot c(Z)) + |Z| \|c(Z)\|^2 = \\
(\text{par l'Eq. (3), 1ère partie}) &= \sum_{z \in Z} \|z\|^2 - 2|Z| \|c(Z)\|^2 + |Z| \|c(Z)\|^2 = \\
&= \sum_{z \in Z} \|z\|^2 - |Z| \|c(Z)\|^2.
\end{aligned}$$

3. Par le point 2, on a :

$$v(X) = \sum_{z \in X} \|z\|^2 - |X| \|c(X)\|^2 \quad (7)$$

$$v(Y) = \sum_{z \in Y} \|z\|^2 - |Y| \|c(Y)\|^2 \quad (8)$$

$$v(X \cup Y) = \sum_{z \in X \cup Y} \|z\|^2 - |X \cup Y| \|c(X \cup Y)\|^2, \quad (9)$$

Par définition de $\Delta(X, Y)$, on a :

$$\begin{aligned}
\Delta(X, Y) &= v(X \cup Y) - v(X) - v(Y) = \\
(\text{par les Eq. (7)-(9)}) &= \sum_{z \in X \cup Y} \|z\|^2 - |X \cup Y| \|c(X \cup Y)\|^2 + \\
&\quad - \sum_{z \in X} \|z\|^2 + |X| \|c(X)\|^2 - \sum_{z \in Y} \|z\|^2 + |Y| \|c(Y)\|^2 = \\
(\text{par l'Eq. (6)}) &= |X| \|c(X)\|^2 + |Y| \|c(Y)\|^2 - |X \cup Y| \|c(X \cup Y)\|^2 \quad (\star)
\end{aligned}$$

Par l'Eq. (3), 1ère partie, on a $\|c(X \cup Y)\|^2 = c(X \cup Y) \cdot c(X \cup Y)$, et on peut re-écrire le résultat du point 1 comme :

$$(|X| + |Y|)c(X \cup Y) = |X|c(X) + |Y|c(Y), \quad (10)$$

d'où on a :

$$\begin{aligned}
|X \cup Y| \|c(X \cup Y)\|^2 &= (|X| + |Y|)c(X \cup Y) \cdot c(X \cup Y) = \\
(\text{Eq. (10)}) &= c(X \cup Y) \cdot (|X|c(X) + |Y|c(Y)) = \\
(\text{encore Eq. (10)}) &= \frac{1}{|X| + |Y|} (|X|c(X) + |Y|c(Y)) \cdot (|X|c(X) + |Y|c(Y)) =
\end{aligned}$$

$$= \frac{|X|^2}{|X|+|Y|} \|c(X)\|^2 + \frac{|Y|^2}{|X|+|Y|} \|c(Y)\|^2 + \frac{2|X||Y|}{|X|+|Y|} c(X) \cdot c(Y). \quad (11)$$

Maintenant, on applique (11) au troisième terme de la partie droite de l'Eq. (*), pour arriver à :

$$\begin{aligned} \Delta(X, Y) = (*) &= \left(|X| - \frac{|X|^2}{|X|+|Y|} \right) \|c(X)\|^2 + \left(|Y| - \frac{|Y|^2}{|X|+|Y|} \right) \|c(Y)\|^2 + \\ &- \frac{2|X||Y|}{|X|+|Y|} c(X) \cdot c(Y) = \\ &= \frac{|X||Y|}{|X|+|Y|} (\|c(X)\|^2 + \|c(Y)\|^2 - 2c(X) \cdot c(Y)) = \\ (\text{par l'Eq. (3)}) &= \frac{|X||Y|}{|X|+|Y|} \|c(X) - c(Y)\|^2. \end{aligned}$$

4. (a) Remarquez que la forme de l'Eq. (5) est très similaire à celle de l'Eq. (4). Il suffira d'adapter la preuve employée au point 2 :

$$\begin{aligned} v(Z) &= \sum_{z \in Z} \|z - c(Z)\|^2 = \\ (\text{pour tout } x) &= \sum_{z \in Z} \|(z - x) + (c(Z) - x)\|^2 = \\ (\text{par (3)}) &= \sum_{z \in Z} \|z - x\|^2 - 2 \sum_{z \in Z} (z - x) \cdot (c(Z) - x) + \sum_{z \in Z} \|c(Z) - x\|^2 = \\ (\text{défn. de } v(\cdot)) &= v(Z, x) - 2 \left(\sum_{z \in Z} z - |Z|x \right) \cdot (c(Z) - x) + |Z| \|c(Z) - x\|^2 = \\ (\text{défn. de } c(\cdot)) &= v(Z, x) - 2|Z|(c(Z) - x) \cdot (c(Z) - x) + |Z| \|c(Z) - x\|^2 = \\ (\text{par (3)}) &= v(Z, x) - 2|Z| \|c(Z) - x\|^2 + |Z| \|c(Z) - x\|^2 = \\ &= v(Z, x) - |Z| \|c(Z) - x\|^2. \end{aligned}$$

On a donc $v(Z, x) = v(Z) + |Z| \|c(Z) - x\|^2$ pour tout $x \in \mathbb{R}^d$.

- (b) Pour ce qui concerne la borne inférieure $\min_{x \in Z} v(Z, x) \geq v(Z)$, on donne une preuve plus longue que nécessaire ; l'argument étant simple, seront également acceptées preuves beaucoup plus succinctes, si correctes. **Erreurs communes** : des pénalités seront imposées lorsqu'on ne trouvera pas trace des certains passages logiques estimés cruciaux.

On considère le problème relâché $\min_{x \in \mathbb{R}^d} v(Z, x)$, où on a enlargi la région admissible, qui était $Z \subset \mathbb{R}^d$, à tout \mathbb{R}^d . Par conséquent, le minimum x sur Z doit être nécessairement dans \mathbb{R}^d . On a donc $\min_{x \in Z} v(Z, x) \geq \min_{x \in \mathbb{R}^d} v(Z, x)$. Maintenant on calcule $\min_{x \in \mathbb{R}^d} v(Z, x)$, comme dans le Théorème 3 du photocopié. Comme la fonction $v(Z, x)$ est strictement convexe dans chaque composante de $x \in \mathbb{R}^d$, il suffit de poser sa dérivée égale à zéro :

$$\begin{aligned} \forall j \leq d \quad \frac{\partial}{\partial x_j} \sum_{z \in Z} \|z - x\|^2 &= \frac{\partial}{\partial x_j} \sum_{z \in Z} (\|z\|^2 + \|x\|^2 - 2z \cdot x) = \\ &= \sum_{z \in Z} \frac{\partial}{\partial x_j} (\|x\|^2 - 2z \cdot x) = \sum_{z \in Z} (2x_j - 2z_j) = |Z|x_j - \sum_{z \in Z} z_j = 0, \end{aligned}$$

d'où $\forall j \leq d \ x_j = \frac{1}{|Z|} \sum_{z \in Z} z_j$, qui implique que $x = c(Z)$ au minimum, et donc que $\min_{x \in \mathbb{R}^d} v(Z, x) = v(Z, c(Z)) = v(Z)$.

Pour ce qui concerne la borne supérieure $\min_{x \in Z} v(Z, x) \leq 2v(Z)$, remarquez que chaque terme dans la somme $\sum_{x \in Z} v(Z, x)$ excède le terme de valeur minimale. On a donc :

$$\begin{aligned}
\sum_{x \in Z} \min_{x \in Z} v(Z, x) &\leq \sum_{x \in Z} v(Z, x) \Rightarrow \\
\min_{x \in Z} v(Z, x) &\leq \frac{1}{|Z|} \sum_{x \in Z} v(Z, x) = \\
(\text{par l'Eq. (5)}) &= \frac{1}{|Z|} \sum_{x \in Z} (v(Z) + |Z| \|c(Z) - x\|^2) = \\
&= v(Z) + \frac{|Z|}{|Z|} \sum_{x \in Z} \|c(Z) - x\|^2 = \\
(\text{par définition de } v(Z)) &= v(Z) + v(Z) = 2v(Z).
\end{aligned}$$

La preuve la plus courte (faite par un élève). Borne inférieure : par la question 4a, $\forall x \in \mathbb{R}^d \ v(Z) = v(Z, x) - |Z| \|c(Z) - x\|^2$; comme le dernier terme est toujours non-négatif, $v(Z) \leq v(Z, x)$, d'où $v(z) \leq \min_{x \in Z} v(Z, x)$. Borne supérieure : pour une valeur particulière de x , c'est-à-dire $x = c(Z)$, on a $v(Z, x) = v(Z) \leq 2v(Z)$; cette borne est donc valide pour $\min_{x \in Z} v(Z, x)$ aussi.

5. La fonction objectif des k -moyennes minimise la somme des distances carrées intra-cluster sur tout cluster $C \in \mathcal{C}$ (où $|C| = k$) :

$$F(\mathcal{C}) = \sum_{C \in \mathcal{C}} \sum_{x \in C} \|x - \rho(C)\|^2, \quad (12)$$

où, pour tout $C \in \mathcal{C}$, $\rho(C)$ est le prototype représentatif du cluster C . Si l'on prend $\rho(C) = c(C)$, on a la fonction objectif des k -moyennes classiques :

$$F'(\mathcal{C}) = \sum_{C \in \mathcal{C}} v(C).$$

Si l'on considère les médioides, on a :

$$F''(\mathcal{C}) = \sum_{C \in \mathcal{C}} \min_{x \in C} v(C, x).$$

- (a) En partant d'une configuration donnée des pour les k -prototypes, les deux phases de l'algorithme de Lloyd, c'est-à-dire allocation au plus proche médiotide, et mise à jour des k prototypes, font toutes les deux baisser la fonction objectif des k -moyennes. La convergence est donc monotone.

erreur commune : par la question 4(b), on peut argumenter que (coût avec centroïdes \leq coût avec médioides $\leq 2 \times$ LHS). Ces bornes ne suffisent pas à prouver la convergence monotone, car on pourrait imaginer que le coût avec médioides aurait des oscillations entre les bornes données.

- (b) Le nombre d'itérations est borné puisque la fonction objectif des k -moyennes est positive ou nulle. Puisqu'on ne répète jamais deux fois la même configuration des k -médioïdes et qu'on a au plus $\binom{n}{k}$ choix, on borne le nombre d'itérations en $O\left(\binom{n}{k}\right)$, ou $O(n^k)$ si k est une constante donnée.

La borne donnée par le nombres de Stirling du deuxième type, qui donnent le nombre de partitions de cardinalité k d'un ensemble donné, sera aussi acceptée. Par ailleurs les nombres de Stirling du deuxième type sont $O(k^n)$ pas $O(n^k)$ si k est fixe.

erreur commune : dire que le nombre des partitions d'un ensemble fini est $\binom{n}{k}$ (il est en fait le nombre de Stirling du deuxième type).

- (c) Soit \mathcal{C}' la solution optimale des k -moyennes classiques (fonction F'), et \mathcal{C}'' la solution optimal des k -moyennes à medioïdes (fonction F''). Alors $F''(\mathcal{C}'') \leq 2F'(\mathcal{C}'')$ par la question 4(b), et $F'(\mathcal{C}'') \leq F'(\mathcal{C}')$ par optimalité de \mathcal{C}' par rapport à F' . D'où $F''(\mathcal{C}'') \leq 2F'(\mathcal{C}')$.

erreur commune : se limiter à dire que la preuve est impliquée par la question 4(b).