

Projets : Algorithme et Programmation Procédurale C. ING 1 GM Apprentissage 2020- 2021

Sujet 1 : Comparateur de vols

L'objectif de ce projet est de réaliser une application qui calcule le plus court temps de voyage entre un aéroport de départ et un aéroport d'arrivée. Pour cela vous utiliserez l'algorithme de Dijkstra. (Voir lien ci-dessous)

<https://www.maths-cours.fr/methode/algorithmme-de-dijkstra-etape-par-etape/>

Vous réfléchirez aux structures de données permettant l'application de cet algorithme.

Pour aller plus loin, vous pourrez ajouter d'autres critères de comparaison des vols et proposer une interface graphique.

Pour vos tests un fichier json contenant des vols est fourni.

Sujet 2 : Cryptographie (cryptanalyse avec fréquences et dictionnaire)

Les services des communications cherchent souvent à transmettre des messages, si possibles cryptés. Ces messages doivent pouvoir être décodés par le destinataire et non par l'adversaire. A l'inverse si vous êtes l'adversaire, votre objectif va être de briser le code sans connaître la clé de décryptage.

Dans une nouvelle de E. A. Poe, le héros décrypte un message secret en étudiant les pourcentages d'apparition des lettres, c'est ce que vous allez réaliser dans ce projet.

Une méthode pour coder un message est de transposer chaque lettre par une autre, en vérifiant seulement que la transformation est une bijection de l'alphabet dans l'alphabet (Voir lien ci-dessous)

[https://www.dcode.fr/chiffrecesar#:~:text=Le%20code%20C%C3%A9sar%20est%20un,identique%20pour%20un%20m%C3%Aame%20message\).](https://www.dcode.fr/chiffrecesar#:~:text=Le%20code%20C%C3%A9sar%20est%20un,identique%20pour%20un%20m%C3%Aame%20message).)

Une manière élégante de décoder un texte de ce type est d'étudier les pourcentages d'apparition des lettres dans le texte codé et de les mettre en relation avec les pourcentages

d'apparition des lettres dans les textes d'une langue donnée. Ainsi on peut trouver la correspondance pour chacune des lettres.

Pour aider au processus on peut s'appuyer sur un dictionnaire. L'objectif de ce projet est de réaliser une application permettant de

- Mettre à jour une base statistique des fréquences à partir de textes issues de bases de données libres.
- Réaliser une proposition de décodage pour un texte crypté donné (A vous de fabriquer un codeur de texte).

Le processus de décryptage sera le suivant :

Choisir un mot : appliquer les statistiques et sortir toutes les solutions possibles de décodage de ce mot (taux de précision à définir). Tester les résultats dans le dictionnaire pour ne garder que ceux qui existent. Compléter les matrices de décodage possibles.

Tant que les matrices ne sont pas complètes, choisir un autre mot avec des lettres inconnues et recommencer pour ces lettres et compléter les matrices existantes, ajouter les matrices nouvelles et éventuellement supprimer les matrices inutiles.

Pour les solutions acceptables : traduire l'ensemble du texte et valider la pertinence de la traduction à l'aide du dictionnaire.

Sujet 3 : Classification de données.

La classification de données consiste à partir d'un ensemble de données caractérisées par plusieurs valeurs à les regrouper automatiquement en différents groupes. L'objectif de la classification est de regrouper ensemble les données similaires. L'objectif de ce projet est de mettre en place le classificateur « Affinity Propagation Clustering » (algorithme en annexe).

Vous proposerez une interface graphique. Vous pouvez proposer des données à classifier à partir de bases de données libres.

Voici des exemples de sites qui proposent des datas.

<https://www.data.gouv.fr/f>

<http://data.grandlyon.com/>

<https://www.census.gov/compendia/statab/>

<https://homengo.com/immobilier/sources/>

<http://www.prix-carburants.gouv.fr/>

<http://www.nosdonnees.fr/>

<https://www.data.gouv.fr/fr/organizations/institut-national-de-la-statistique-et-des-etudes-economiques-insee/>

Pour aller plus loin vous pouvez proposer un domaine application de cette algorithmme et une interface graphique.

Algorithme

On définit $S(i,k)$ matrice carrée de taille nbPoint*nbPoints des similarités telle que si deux points sont proches x_i, x_k sont plus similaires que x_l, x_m alors $S(i,k) > S(l,m)$.

En pratique on prend $S(i,k) = -||x_i - x_k||^2$

On pose $S(i,i) = pre$ avec

$$pre = \min_{i,k \ i \neq k} S(i,k) \text{ ou bien } pre = \text{moyenne}_{i,k \ i \neq k} S(i,k) \text{ ou encore } pre = \text{median}_{i,k \ i \neq k} S(i,k)$$

On pose les matrices nulles $R(i,k)$ et $A(i,k)$

On pose $lamp=0.5$

Pour chaque itération

$$Rold = R$$

$$r(i,k) \leftarrow s(i,k) - \max_{k' \text{ s.t. } k' \neq k} \{a(i,k') + s(i,k')\}$$

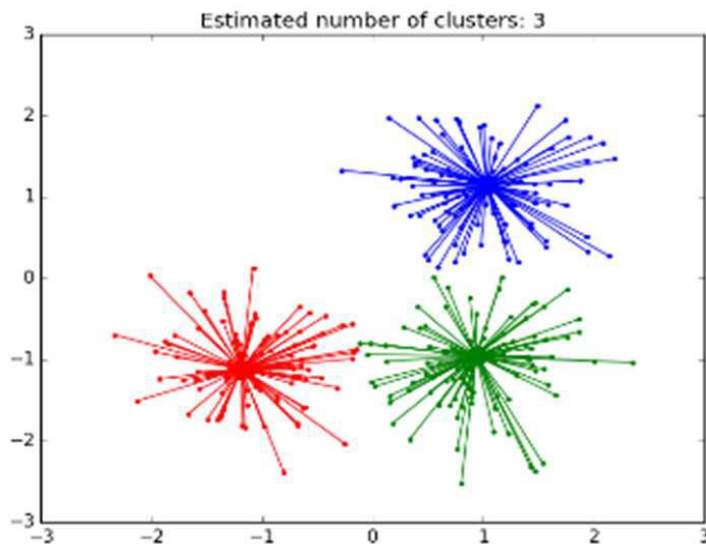
$$R = (1-lamp) * R + lamp * Rold$$

$$Aold = A$$

$$a(i,k) \leftarrow \min \left\{ 0, r(k,k) + \sum_{i' \text{ s.t. } i' \notin \{i,k\}} \max \{0, r(i',k)\} \right\}$$

$$A = (1-lamp)*A + lamp * Aold$$

La récupération des clusters se déroule :



Ci-dessous un lien pour voir quelques domaines applicatifs des algorithmes de clustering

<https://www.lemagit.fr/conseil/Trois-exemples-concrets-dalgorithmes-de-Machine-Learning>

Sujet 4 : Jeu Awalé



Le but de ce projet est d'implémenter le jeu Awalé en langage C.

Le jeu s'effectue avec deux joueurs, un plateau de 12 cases (deux fois six cases) et 48 graines

Une partie d'Awalé se joue à tour de rôle. Chaque joueur se place devant l'une des rangées de six cases : son camp. Chacun débutant avec quatre graines dans chacune des cases (soit

24 graines). Le premier joueur prend les quatre graines (un lot appelé semis) dans la case de son choix et les distribue une à une dans les cases suivantes dans le sens inverse des aiguilles d'une montre. Et ainsi de suite. Toutefois, si le semis repasse par la case dans laquelle il a été prélevé, le joueur doit sauter cette case sans y poser aucune graine. Pour mener à bien une partie d'Awalé, il faut respecter les règles suivantes.

La prise simple

Dans le jeu d'Awalé, la prise simple survient lorsqu'un joueur est obligé de poser la dernière graine de son semis dans le camp adverse. Si cette case contient une ou deux graines au départ, le joueur prend ces graines et celle qu'il devait y déposer.

La prise multiple

La prise multiple découle de la prise simple. Si la case précédant la case du camp adverse dans laquelle le joueur vient de prélever les graines se trouve aussi chez l'adversaire, il prend également les graines de la case précédente (toujours en suivant le sens inverse des aiguilles d'une montre).

Donner à manger

Comme un joueur ne peut jouer deux fois de suite, il doit se débrouiller pour que l'adversaire ait toujours des graines dans son camp. Il doit donc s'arranger pour placer certaines de ses graines dans le camp adverse pour que la partie continue. De la même façon, le joueur ne peut pas jouer un coup qui prenne toutes les graines de son adversaire. S'il y est obligé, il continue le cours de la partie, mais ne prend aucune graine.

Comment gagner une partie d'Awalé

Le gagnant d'une partie d'Awalé est celui qui a remporté le plus de graines. Le jeu prend fin lorsque l'un des joueurs n'a plus de graines dans son camp et que son adversaire n'a pas eu l'opportunité de lui en redonner.

<http://www.lecomptoirdesjeux.com/l-awale.htm#:~:text=L'awal%C3%A9%20se%20joue%20sur,qui%20deviennent%20alors%20son%20camp.>



Grande École de sciences,
d'ingénierie, d'économie et de
gestion de **CY Cergy Paris Université**