### **Clustering**

Présenté à

Pierre-Paul Monty

Par

Maxime Denis

Mohamed Ilias

Munteanu Dan

### 

Cégep du Vieux-Montréal

19/12/2019

### **Table des matières**

1 Introduction ……………………………………………………………………...1

2 Abstrait …………………………………………………………………………. 1

3 Méthodologie ……………………………………………………………………1

3.1 Représentation de vecteurs

dans un espace à N dimensions et une distance……………………..2

4. Mesure de similarité entre objets ………………………….………………....3

5. Méthode du K-means…………………………………………………………..3

6. Méthode du KNN (Innovation)………………………………………………...4

7.Problèmes rencontrés …………………………………………...……….4

8. À quel point les clusters trouvés sont-ils bons ?…………………………….5

9. Le Multiprocessing……………………………………………………………...7

10. Conclusion.…………………………………………………………………….9

### 

### 

### 

### 

### **1. Introduction**

Au cours de la dernière décennie, les quantités de données générées par des systèmes informatisés ont augmenté de manière exponentielle. La multiplication des interactions génère continuellement un flot de données brut. Les progrès technologiques et économiques permettent d’analyser en profondeur ces données afin d’en tirer des corrélations pertinentes.

En traitant d’énormes données, des limites computationnelles apparaissent et enfreignent l’optimisation des données. De ce fait, il existe plusieurs techniques d’optimisation tels que la maximisation des similarités intra-Cluster, la minimisation des similarités inter-Cluster et le calcul parallèle . Cette dernière est une technique forte intéressante, car elle permet de maximiser l’utilisation simultanée de plusieurs ressources computationnelle afin de maintenir le flux de données. La parallélisation a pour concept de distribuer la tâche de travail à différents nœuds.

### **2. Abstrait**

Dans le cadre de notre étude, nous examinerons l’approche par le partitionnement de données. Ce processus permet de subdiviser en conteneurs des données ayant des similarités. Ces regroupements d’objets forment des classes communément appelées clusters.

### **3. Méthodologie**

Le mariage entre notre approche par le clustering et la parallélisation offre d’intéressantes perspectives. Cette architecture en nœud (Cluster) permet de gérer l’efficience de nos calculs étant donnée de la réduction des sollicitations de puissances matérielle par un même et unique cœur. Le but de notre expérimentation consiste à entraîner un système grâce à un apport de romans littéraires afin de reconnaître des synonymes grâce à leurs rôles grammaticaux, syntaxiques et sémantiques.

En effet, un mot se retrouvant souvent à la même position dans un contexte a tendance à avoir la même fonction. L’hypothèse posée au départ stipule que les mots qui ont tendance à avoir le même contexte ont tendance à avoir une sémantique similaire. De cette manière, le système expérimental permettra d’identifier des mots qui ont une sémantique similaire.Nous utilisons une fenêtre glissante comme étant notre critère de proximité ce qui tient en compte les cooccurrences entre les mots avoisinant le mot central de la fenêtre. De cette manière, en regroupant les mots dans une matrice symétrique, nous prédisons la similarité des mots en fonction de la distance de leurs vecteurs respectifs.

L’expérimentation se divise en 3 grandes parties. La première étape est l’entraînement qui consiste à entraîner le système en analysant le flux de textes et d’en tirer les cooccurrences entre les différents mots selon une taille de fenêtre fixe. La seconde étape consiste à déterminer une liste de “meilleurs synonymes fortement possibles” grâce à notre hypothèse de proximité vectorielle en calculant la distance. Puis, la dernière étape est celle du système de partitionnement par cluster.

3.1 Représentation de vecteurs dans un espace à N dimensions et une distance:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N/A | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0 | **d** | **d** | **d** | **d** | **d** | **d** | **d** | **d** | **d** |
| 2 | **d** | 0 | **d** | **d** | **d** | **d** | **d** | **d** | **d** | **d** |
| 3 | **d** | **d** | 0 | **d** | **d** | **d** | **d** | **d** | **d** | **d** |
| 4 | **d** | **d** | **d** | 0 | **d** | **d** | **d** | **d** | **d** | **d** |
| 5 | **d** | **d** | **d** | **d** | 0 | **d** | **d** | **d** | **d** | **d** |
| 6 | **d** | **d** | **d** | **d** | **d** | 0 | **d** | **d** | **d** | **d** |
| 7 | **d** | **d** | **d** | **d** | **d** | **d** | 0 | **d** | **d** | **d** |
| 8 | **d** | **d** | **d** | **d** | **d** | **d** | **d** | 0 | **d** | **d** |
| 9 | **d** | **d** | **d** | **d** | **d** | **d** | **d** | **d** | 0 | **d** |
| 10 | **d** | **d** | **d** | **d** | **d** | **d** | **d** | **d** | **d** | 0 |

\***[[1]](#footnote-1)**

### 

### **4. Mesure de similarité entre objets**

La similarité entre objets calculée selon leurs caractéristiques est exprimée en termes de fonctions de distance : d(o1,o2) . Ainsi, deux données qui se ressemblent auront une distance de dissimilarité réduite, alors que deux objets différents auront une distance de séparation plus grande. Il existe une multitude de méthodes afin de calculer des distances. Dans le cadre de notre expérimentation, la mesure de qualité de chaque cluster a été calculée selon la distance entre les objets appartenant au cluster et la distance entre les clusters.

Distance euclidienne:



### **5. Méthode du K-means**

Au début, par l’intermédiaire de la ligne de commandes, l’utilisateur a pour choix de choisir de manière arbitraire des mots représentant les centroïdes ou aléatoirement. Par la suite, l’algorithme distribue les points dans les différents centroïdes en fonction de leur distance par rapport aux points initiaux, soit la mesure de similarité entre objets précédemment expliquée.

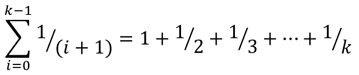
Dès que la répartition des points se termine, l’algorithme rééquilibre le cluster afin de déterminer le prochain centroïde selon la moyenne des points du cluster. Simultanément , cette étape recalcule les clusters jusqu’au point de stabilité trouvé.

Nous avons déterminé le nombre de clusters de manière empirique ,en essayant différentes valeurs de K.Puis nous déterminons la qualité du K selon la cohérence des résultats . Un grand nombre de clusters réduit considérablement la découverte des corrélations . De l’autre part, peu de clusters généralisent les points dans un cluster surdimensionné.

### **6. Méthode du KNN (Innovation)**

Le KNN repose sur un principe simple qu’on assume que les voisins proches d’un point sont également de la même famille. Si l’on regarde le mot dont la distance est la plus courte, il est donc fort probable que celui-ci soit du même genre. Le procédé est appelé 1NN. Par contre, l'échantillon prélevé est trop pauvre pour en venir à un résultat plus définitif.

Nous allons alors choisir un ensemble de K points (où K représente un certain nombre de points) pour tenter de minimiser la marge d’erreur. D’où le terme KNN. De plus, il nous faut considérer la distance entre le point d’origine et les autres points, car celle-ci varie. Pour ce faire, nous donnons un poids en fonction de la distance pour chaque point avec la formule qui suit:



### **7. Problèmes rencontrés**

Nous avons déterminé le nombre de clusters de manière empirique, en essayant différentes valeurs de K. Puis nous déterminons la qualité du K selon la cohérence des résultats.

Une des faiblesses de K-means est due au fait qu’il faut enclencher le lancement de plusieurs exécutions avec différents états initiaux et retenir la configuration jugée la meilleure. Les variantes des résultats du K-means sont en fonction des techniques de calcul vectoriel et du choix initial de cluster.

Étant donné que le choix de K est assez vaste, il est difficile de faire un choix raisonnable. Différentes valeurs de K ressortent des corrélations différentes, la difficulté réside dans le choix d’un nombre K intéressant permettant de mettre en lumière des modèles intéressants. Il existe une approche afin de déterminer le nombre de clusters optimal.

### 

### **8. À quel point les clusters trouvés sont-ils bons ?**

Résultat interprétable et utilisable

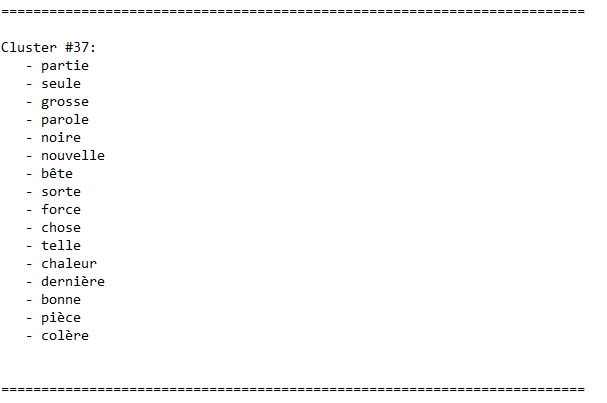
Extraction de clusters en intégrant des contraintes spécifiées par l’utilisateur

Capacité à traiter les données bruitées et les exceptions

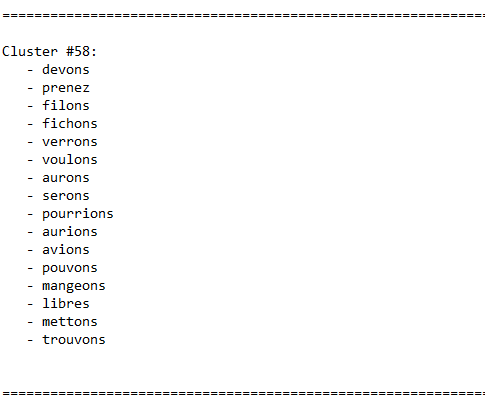
Nos résultats sont plutôt concluants puisqu’ils ont du sens puisqu’on ne retrouve pas de données en double et les mots de chaque cluster sont similaires. Par exemple, sur 100 centroïdes aléatoires nous avons obtenu les résultats suivants:



\*Les verbes conjugués à l’imparfait sont regroupés ensemble.



\*Dans ce cluster, tous les mots finissent en « e ».



\*Presque tous les verbes finissent en « ons »

En bref, nous sommes satisfaits de nos résultats puisque la majorité des clusters ont tous un point en commun. De plus, nos résultats se raffinent lorsqu’on met de plus en plus de centroïdes. On voit déjà la différence entre nos résultats avec 2 centroïdes puis ceux avec 100 centroïdes. Ceci est tout à fait normal puisque les mots vont se rapprocher le plus possible dépendamment du choix des différents centroïdes disponibles.

### 9.Le Multiprocessing

Nous avons tenté d’implémenter un système de multiprocessing dans le but d'accroître potentiellement les performances du programme en soi. Cependant, adapter le code à cette solution s’est avérée plus ardu que prévu.

En effet, faire du multiprocessing avec des données aussi massives que la matrice vectorielle n’est pas chose facile. L’un des premiers problèmes rencontrés fut que le processus ne peut pas se dérouler à l’intérieur d’une classe. Étant donné que toute l’information essentielle est contenue dans notre classe appelée Clustering, il faut donc extraire l’information pour la passer à une fonction globale. Or mis, le multiprocessing démarre plusieurs applications distinctes qui ne partagent pas nécessairement la même mémoire, ce qui résulte par un échec lors de la procédure.

Une alternative fut découverte. Il est effectivement possible de partager la mémoire entre chaque processus d’exécution. Néanmoins, cette tentative s’est aussi soldée par un échec, car les versions de Python installées sur les ordinateurs du cégep ne peuvent partager des données de plus de 4G de mémoire. La matrice vectorielle dépassant cette limite de mémoire et le peu de temps restant nous à réviser nos plans malheureusement.

### **10. Conclusion**

En conclusion, l'hypothèse de départ est valide vu le résultat présenté. Sémantiquement, les mots semblables ont de fortes probabilités d'être des synonymes entre eux. Notons que le résultat semble véridique parmi les dix premiers mots en moyenne. Le résultat se fausse après ces 10 premiers mots dus aux différences sémantiques.

### **11. Reconnaissance**

Les auteurs voudraient remercier notre professeur Pierre-Paul Monty pour nous avoir supportés avec ses vastes connaissances en la matière tout au long du projet.

### **12. Bibliographie**

* elbow point: <https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/>
* Clustering: <https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/>
* KNN: <https://mrmint.fr/introduction-k-nearest-neighbors>
* K-mean:<https://medium.com/analytics-vidhya/how-to-determine-the-optimal-k-for-k-means-708505d204eb>

1. Exemple à 10 dimensions.

   **d** = Donnée de la fréquence [↑](#footnote-ref-1)