Université Paris Cité

LLMs : Étude du clustering combiné aux méthodes de réduction de la dimension

Important :

- Ce TP évalué est à déposer au plus tard à **17h30.**
- Il doit être soumis sous la forme de **notebook Python**.
- Le rendu doit être complet : **toutes les étapes doivent être rigoureusement commentées.**
- Toute ressemblance entre projets rendus entraînera l'attribution de la note 0.
- Ce travail peut se faire par **binôme** ou **seul**.
- Cette étude servira à préparer votre examen.

Pour ce TP, vous allez analyser **2 ensembles de données** en utilisant des représentations vectorielles. L'ensemble de données de **PubMed est obligatoire**, et vous avez le **choix entre BBC News et Content Web** pour le second ensemble de données.

- **PubMed 20k RCT (obligatoire) :** Une collection de résumés d'articles de recherche clinique provenant de la base de données PubMed. Ce jeu de données est structuré en **5 classes**, chacune correspondant à une étape distincte de l'étude clinique, telle que l'objectif, les méthodes, les résultats et les conclusions.
- **BBC News (au choix) :** Un ensemble d'articles de presse couvrant divers sujets. Ce jeu de données est typiquement segmenté en **5 classes** correspondant à différentes catégories de nouvelles (par exemple, politique, sport, etc.).
- **Web Content (au choix) :** Ensemble diversifié de textes issus de pages web, classés en **16 catégories** couvrant une large gamme de secteurs et d'intérêts (éducation, actualités, commerce électronique, sport, etc.).

Chaque ensemble de données sélectionné pour ce projet est représenté par des embeddings générés à l'aide de **GPT**.

Vous trouverez les données dans les fichiers **`pubmed_dataset.pickle`**, **`bbc_dataset.pickle`** et **`webcontent_dataset.pickle`** accessibles via les liens suivants :

```
https://cifre.s3.eu-north-1.amazonaws.com/pubmed_dataset.pickle
https://cifre.s3.eu-north-1.amazonaws.com/bbc_dataset.pickle
https://cifre.s3.eu-north-1.amazonaws.com/webcontent_dataset.pickle
```

Ce projet a pour but de réaliser une analyse de clustering sur les jeux de données fournis. `K-means` sera employé en association avec des méthodes de réduction de dimension telles que l'ACP, `TSNE et UMAP`.

1. Commencez par déterminer la dimension de chaque jeu de données (nombre de documents et nombre de caractéristiques).

   important: Pour chaque étape, il est essentiel de comparer les performances des méthodes de réduction de dimension en utilisant l'algorithme `K-means`, en mettant ces résultats en parallèle avec ceux obtenus à partir des données initiales sans réduction de la dimension. L'objectif principal est d'étudier l'impact de ces méthodes de réduction de dimension sur la qualité du clustering.

2. Effectuer une ACP suivie d'un K-means, en testant différents **nombres de composantes principales** (par exemple, 2, 3, 5, 10, 15, 20). Utiliser des métriques internes (**comme le score de silhouette**) et externes (**comme la NMI)** pour identifier le nombre optimal de composantes.
3. Appliquer TSNE pour la réduction de dimension, en ajustant le **paramètre de perplexité**, un paramètre crucial qui peut influencer la formation des clusters. Réaliser ensuite un clustering avec K-means sur les données transformées par TSNE.
4. Employer UMAP sur vos jeux de données, en variant le **nombre de composantes** et en modifiant les hyper-paramètres tels que le **nombre de voisins** (*n_neighbors*), la **distance minimale** (*min_dist*) et la **métrique de distance** (*metric*).
5. Pour affiner le choix des paramètres pour l'ACP, TSNE et UMAP, utiliser le package **pyDRMetrics** (disponible à l'adresse : https://github.com/zhangys11/pyDRMetrics).

   Important : **pyDRMetrics** est un package Python qui permet une évaluation quantitative des méthodes de réduction de dimension. Il fournit des scores pour évaluer la capacité des représentations dans un espace de dimension réduite à conserver les relations et la structure de l'espace de données original (pour plus d'informations, voir la première référence).

6. Pour approfondir votre analyse, vous avez la possibilité d'utiliser le package **QVisVis** (disponible sous R à l'adresse : https://github.com/MDSOPT/QVisVis) afin d'évaluer de manière plus poussée la qualité de la réduction de dimension effectuée en utilisant le **coefficient d'agrément** (*agreement metric*). Il est aussi possible d'utiliser le script suivant qui a été développé dans le but de calculer ce score sous Python.

.

   https://cifre.s3.eu-north-1.amazonaws.com/agreement_metric.py

   important : L'interprétation des valeurs de ce coefficient consiste à évaluer le score pour différentes tailles de voisinage k. Des valeurs plus élevées indiquent une meilleure préservation des relations de voisinage dans la représentation de dimension inférieure, ce qui implique une représentation plus fidèle de la structure originale des données.

7. Que peut-on dire de cette étude ?

Références

Zhang, Y., Shang, Q., & Zhang, G. (2021). pyDRMetrics-A Python toolkit for dimensionality reduction quality assessment. Heliyon, 7(2).

France, S. L., & Akkucuk, U. (2021). A review, framework, and R toolkit for exploring, evaluating, and comparing visualization methods. The Visual Computer, 37(3), 457-475.