Approche de Clustering de Flux basée sur les Graphes de Voisinage

Ibrahim Louhi[*,**] Lydia Boudjeloud-Assala[*]
Thomas Tamisier[**]

[*]Université de Lorraine, Laboratoire d'Informatique Théorique et Appliquée.
{ibrahim.louhi, lydia.boudjeloud-assala}@univ-lorraine.fr
[**]Luxembourg Institute of Science and Technology.
{ibrahim.louhi, thomas.tamisier}@list.lu

Dans plusieurs domaines les données sont générées d'une façon continue et souvent à une fréquence très rapide. Ce type de données est connu sous le nom de flux de données. Les flux de données sont caractérisés principalement par l'aspect temporel et par leur grande taille, ce qui rend le processus de clustering des éléments du flux une tâche laborieuse.

Traiter les éléments d'une manière séparée, au fur et à mesure de leur apparition, conduit souvent à des erreurs dans leur affectation aux nouveaux clusters. La principale idée de notre approche consiste à traiter un groupe de nouveaux éléments arrivant presque simultanément au lieu de traiter chaque élément séparément. Cela permet de prendre en compte les caractéristiques d'un groupe de données arrivant dans la même période temporelle. Nous supposons que deux éléments générés successivement sont probablement causés par les mêmes facteurs, ce qui implique qu'il y a de fortes chances qu'ils se ressemblent. Le but de notre approche est de construire incrémentalement un graphe de voisinage permettant de traiter et de visualiser le flux de données.

En premier lieu, nous attendons l'arrivée du premier groupe d'éléments (les groupes ont une taille fixe définie par l'utilisateur). Nous appliquons un clustering basé sur le voisinage sur les éléments du premier groupe : nous calculons la distance entre chaque couple d'éléments et nous considérons que deux éléments sont voisins si leur distance est inférieure à un seuil (qui est fixé également par l'utilisateur). Nous considérons que chaque ensemble de voisins constitue un cluster. Nous déterminons ensuite le centroid de chaque cluster (l'élément le plus proche du reste des éléments du cluster). Les clusters obtenus sont représentés dans un graphe de voisinage : pour chaque cluster, chaque élément est représenté par un noeud, les arêtes représentent la distance entre chaque élément et le centroid de son cluster.

Les éléments du groupe suivant sont traités, indépendamment dans un premier temps, avec le même processus que les éléments du premier groupe. De la même manière nous obtenons de nouveaux clusters et nous identifions également leurs centroids. Les nouveaux clusters sont utilisés pour mettre à jour le graphe de voisinage : nous calculons la distance entre chaque centroid des nouveaux clusters et les centroids des anciens clusters, si la distance entre deux centroids de clusters est inférieure au seuil, les deux clusters sont reliés. Cela se traduit par la création d'une arête entre les noeuds représentant les deux centroids. Dans le cas ou un nouveau cluster n'est similaire à aucun des anciens clusters, il est rajouté au graphe sans qu'il ne soit relié avec un autre cluster (ce qui représente l'apparition d'un nouveau cluster dans le

flux). Ainsi de suite, chaque groupe d'éléments qui arrive participe, d'une façon continue et incrémentale, à la construction du graphe de voisinage.

Affecter un groupe de nouveaux éléments aux clusters, en se basant seulement sur des éléments représentatifs des clusters (les centroids), évite de comparer les éléments un à un et d'augmenter la complexité. Dans le but d'éviter que le nombre d'éléments dans le graphe n'augmente au point de saturer la visualisation, nous fixons un nombre maximal d'éléments à visualiser. Au delà de ce nombre, nous supprimons les clusters qui ont disparu du flux. Un cluster disparu du flux est un cluster qui n'a pas été mis-à-jour depuis une longue période. Le graphe et ses changements sont visualisés par l'utilisateur en temps réel.

Notre approche a été testée sur des jeux de données labellisés, et en même temps été comparée à trois algorithmes de clustering de flux de données : CluStream (Aggarwal et al. (2003)), ClusTree (Kranen et al. (2009)) et DStream (Chen et Tu (2007)). Nous avons effectué des évaluations avec des critères externes et des critères internes du clustering. Ces évaluations ont montré que notre approche obtient généralement de meilleurs résultats.

Le choix des valeurs des paramètres des algorithmes peut avoir un impact sur les résultats obtenus. Par exemple dans le cas de notre approche, la taille des groupes joue un rôle important dans le temps d'exécution de l'approche, et le seuil de la distance peut changer le nombre de clusters dans chaque groupe. Lors de nos évaluations, un paramétrage optimal est utilisé pour chaque algorithme dans le but de comparer les meilleurs résultats possible. Sachant que l'utilisateur peut choisir les valeurs des paramètres selon ses préférences.

En perspective nous envisageons d'étudier l'impact des valeurs des paramètres sur les résultats obtenus par notre approche. Ensuite, valider l'approche visuelle et de traitement de flux sur tout type de données. Notre objectif à long terme est d'améliorer et d'intégrer cette approche dans un environnement interactif de fouille visuelle de flux de données, où l'utilisateur pourra interagir avec le processus du traitement à partir de la visualisation.

Références

Aggarwal, C. C., J. Han, J. Wang, et P. S. Yu (2003). A framework for clustering evolving data streams. In *Proc. of the 29th inter. conf. on Very large data bases-V29*, pp. 81–92. VLDB.

Chen, Y. et L. Tu (2007). Density-based clustering for real-time stream data. In *Proc. of the 13th SIGKDD inter. conf. on Knowledge discovery and data mining*, pp. 133–142. ACM.

Kranen, P., I. Assent, C. Baldauf, et T. Seidl (2009). Self-adaptive anytime stream clustering. In *Proc. of the 9th Inter. Conf. on Data Mining*, pp. 249–258. IEEE.

Summary

We propose a neighborhood-based approach for data streams clustering. Instead of processing each new element one by one, we propose to process each group of new elements simultaneously. A neighborhood-based clustering is applied on each new group. We also define an incremental construction method of the neighborhood graph based on the stream evolution. To validate the approach, we apply it to multiple data sets and we compare it with various stream clustering approaches.