N° d'ordre NNT : 2017LYSE2123

THESE de DOCTORAT DE L'UNIVERSITÉ DE LYON

Opérée au sein de

L'UNIVERSITÉ LUMIÈRE LYON 2

École Doctorale : ED 512 Informatique et Mathématiques

Discipline : Informatique

Soutenue publiquement le 29 novembre 2017, par :

Xinyu WANG

Toward Scalable Hierarchical Clustering and Co-clustering Methods:

Application to the Cluster Hypothesis in Information Retrieval

Devant le jury composé de :

Christel VRAIN, Professeure des universités, Université d'Orléans, Présidente

Lynda TAMINE-LECHANI, Professeure des universités, Université Toulouse 3, Rapporteure

Gilbert SAPORTA, Professeur, Conservatoire National des Arts et Métiers, Rapporteur

Marie-Jeanne LESOT, Maître de Conférences HDR, Université Paris 6, Examinatrice

Jérôme DARMONT, Professeur des universités, Université Lumière Lyon 2, Directeur de thèse

Julien AH-PINE, Maître de Conférences, Université Lumière Lyon 2, Co-Directeur de thèse

Contrat de diffusion

Résume de la Thése

Titre: Méthodes de regroupement hiérarchique agglomératif et co-clustering, leurs applications aux tests d'hypothèse de cluster et implémentations distribuées

Xinyu Wang, Xinyu.Wang@univ-lyon2.fr

Supervisée par : Julien Ah-Pine and Jérôme Darmont

Julien.Ah-Pine@univ-lyon2.fr, Jerome.Darmont@univ-lyon2.fr

Laboratoire ERIC (EA 3083), Université de Lyon, Lyon 2

5, avenue Pierre Mendès France, 69676 Bron Cedex, France

Comme une méthode d'apprentissage automatique non supervisé, la classification automatique est largement appliquée dans des tâches diverses. Différentes méthodes de la classification ont leurs caractéristiques uniques. La classification hiérarchique, par exemple, est capable de produire une structure binaire en forme d'arbre, appelée dendrogramme, qui illustre explicitement les interconnexions entre les instances de données. Le co-clustering, d'autre part, génère des co-clusters, contenant chacun un sous-ensemble d'instances de données et un sous-ensemble d'attributs de données.

Dans cette thèse, nous travaillons sur des données textuelles. Compte tenu d'un corpus de documents, nous adoptons l'hypothèse de «bag-of-words» et applique le modèle vectoriel. Nos données saisies sont transformées à une matrice de document-terme, qui est remplie de poids TF-IDF. L'avantage de regrouper des documents à l'aide du regroupement hiérarchique est qu'il organise des documents et ne nécessite pas de nombre du groupes prédéfinis. Cependant, la procédure du calcul est coûteuse en

raison d'une haute complexité. Dans le cadre de cette thèse, nous travaillons sur les techniques classiques de classification ascendante hiérarchique et nous proposons le Sim_AHC [1]. C'est une expression de la formule de Lance et Williams [2] en fonction de produits scalaires plutôt qu'en termes de distances. Nous établissons les conditions dans lesquelles cette nouvelle expression est équivalence à la méthode initiale. L'intérêt de cette approche est double. Tout d'abord, nous pouvons étendre naturellement les techniques classiques de classification ascendante hiérarchique aux fonctions noyaux. Ensuite, le raisonnement sur des matrices de produits scalaires est davantage propice à la définition de méthodes de seuillage de mesures de proximités. Nous proposons alors de pré-traiter la matrice de proximités de façon à la rendre éparse afin de permettre un meilleur passage à l'échelle de ces techniques de classification.

Avec la formule de Lance et Williams et les méthodes classiques de classification ascendante hiérarchique comme notre base de référence, nos expériences utilisant des proximités générées par le noyau gaussien et le noyau linéaire démontrent que, les résultats obtenus par Sim_AHC sont identiques à ceux produits par les méthodes initiales. Plus important encore, lorsque la matrice de proximités est rendue plus en plus éparse, l'utilisation de la mémoire et du temps de fonctionnement diminuent. Par contre, la qualité de la classification est garantie, grâce au fait que les bruits sont supprimés par le seuillage.

Contrairement à la classification hiérarchique, le co-clustering effectue la classification dans l'espace de données et l'espace des attributs. Cependant, le co-clustering ne peut pas préserver l'interconnexion des éléments qu'il regroupe. Pour surmonter cet inconvénient, nous proposons SHCo-Clust [3], la méthode de co-clustering hiérarchique basée sur la similar-

ité. Il est considéré comme une méthode hybride de Sim_AHC et de co-clustering spectral. Concrètement, dans SHCoClust, nous modelons un corpus de documents comme une graphe bipartite, dont les sommets sont des documents et des termes. Ensuite, nous appliquons la méthode de spectral-SVD [4] pour couper la graphe en plusieurs sous-graphes, chaque étant un co-cluster. La méthode de spectral-SVD, en fait, construit une espace avec les vecteurs propres de la matrice laplacienne de la graphe bipartite. Puis, elle projette la matrice originale dans cette espace. Nous appliquons la classification hiérarchique Sim_AHC sur la matrice transformée.

SHCoClust hérite des caractéristiques de la classification hiérarchique Sim_AHC et du co-clustering spectral. Il produit un dendrogramme composé à la fois de documents et de termes. En coupant le dendrogramme, nous pouvons obtenir un certain nombre de co-clusters, dont chacun est un co-cluster hiérarchique, c'est-à-dire que les interconnexions de documents et de termes dans un co-cluster sont préservées. Plus important encore, comme SHCoClust utilise également des proximités du produit scalaire, nous pouvons également l'étendre aux fonctions du noyau et nous pouvons appliquer une stratégie du seuillage pour rendre la matrice éparse. Nos expériences démontrent que la qualité de la classification de SHCoClust est en grande partie améliorée par rapport aux méthodes de la classification hiérarchique conventionnelle. Par rapport à la méthode de co-clustering spectral, SHCoClust réalise une amélioration lorsque sa matrice de proximités est rendu éparse. En outre, nous constatons que, en épargnant la matrice de proximités, bien que moins de mémoire et moins de temps soient nécessaires pour effectuer le calcul, la qualité de la classification peut être garantie. Dans nos ensembles de données testés, en moyenne,

un gain du mémoire et un gain du temps jusqu'à 75 % sont obtenus sans nuire à la qualité de la classification.

L'hypothèse de cluster [5] est une hypothèse fondamentale pour les applications de la recherche d'informations basées sur la classification automatique. Il indique que les documents dans le même groupe ont tendance à être pertinents pour la même requête. En testant cette hypothèse, il nous permet de examiner comment une requête est répondu. De nombreux travaux [6–10] passés effectuent des tests sur cette hypothèse, en utilisant des méthodes de la classification hiérarchique conventionnelle. Certains de ces travaux vérifient si l'hypothèse de cluster répond à un ensemble de données testé. Certains comparent des stratégies de recherche d'information dans un dendrogramme. Autres examinent lequel méthode de la classification hiérarchique donne la meilleur efficacité de recherche.

Cependant, les conclusions des travaux par rapport à la méthode de la classification la plus efficace ne sont pas cohérentes, en raison des différences dans les mesures d'évaluations, les paramètres expérimentaux et les ensembles de données testés. En outre, l'efficacité du calcul n'est pas discutée. Puis, seulement quartes méthodes de la classification hiérarchique conventionnelle sont testés. Intéressés à fournir une référence mise à jour et plus complète pour les tests de la l'hypothèse de cluster, nous proposons deux nouveaux tests en appliquant les méthodes proposées, le Sim_AHC [11] et le SHCoClust. Pour chaque méthode, nous obtenons d'abord des dendrogrammes générés par les méthodes de la classification. Ensuite, nous utilisons l'E mesure pour évaluer l'efficacité de la recherche sur un dendrogramme. L'E mesure est une mesure impartiale qui calcule la moyenne harmonique de la précision et du rappel. Une valeur haute signifie une bonne efficacité de recherche. L'utilisation de l'E mesure est dans

le contexte de la recherche de cluster optimale, qui permet de trouver le cluster le plus pertinent à une requête dans un dendrogramme. Dans nos expériences, nous utilisons l'E mesure pour testr et comparer l'efficacité des méthodes de la classification dans le Sim_AHC et dans le SHCoClust. Puis, pour examiner l'efficacité du calcul, nous aussi testons l'influence de rendre la matrice de proximités éparse sur l'efficacité de la recherche, et nous appliquons une test statistique pour comparer les résultats obtenus par le Sim_AHC et par le SHCoClust.

Par rapport à la méthode la plus efficace, nos expériences utilisant des proximités générées par le noyau linéaire et le noyau gaussien montrent que la méthode du lien moyen et la méthode de Ward sont les méthodes les plus efficaces lors de l'utilisation de Sim_AHC. Cependant, lors de l'utilisation de SHCoClust, les méthodes les plus efficaces deviennent le lien simple, le lien moyen, le centroïde, Ward et McQuitty. Nos résultats sont partiellement d'accord avec des découvertes dans les travaux passés sur le même sujet. En termes d'influence de rendre la matrice du proximités éparse sur l'efficacité de recherche, nous constatons que l'efficacité a tendance à être invariante. En fait, les valeurs de l'E mesure se gardent presque au même niveau, même si la matrice de proximités est rendu plus en plus éparse. C'est un résultat intéressant. Il signifie que c'est possible d'avoir la même efficacité de recherche en utilisant beaucoup moins de mémoire et de temps pour effectuer le calcul.

En comparant les résultats de la test du Sim_AHC et de la test du SHCoClust à l'aide d'un test de Student, nous découvrons que Sim_AHC est plus efficace que SHCoClust lorsqu'il utilise des méthodes de lien simple, lien complet, lien moyen, McQuitty et Ward dans de petits ensembles de données. Cependant, SHCoClust est plus efficace que Sim_AHC en

utilisant des méthodes de lien simple, lien moyen et centroïde dans des ensembles de données relativement plus grandes.

Intéressés par effectuer le calcul pour des ensembles de données vastes, nous choisissons l'Apache Spark[1] pour implémenter Sim_AHC et SHCoClust. Le Spark est une plate-forme du calcul distribué. Il utilise la capacité du calcul collective d'un groupe de noeuds pour traiter des ensembles de données vastes. En général, le Spark fonctionne sur un système de fichier distribué, qui est établi sur un groupe de machines. Après une initialisation, le Spark coupe des tâches du calcule et les assigne aux noeuds, qui ensuite effectuent leurs tâches simultanément. Le concept de base du Spark est les données distribuées résilientes (RDDs) [12]. Comme une abstraction grossier, une RDD est en fait un morceau de données qui est mit en mémoire-cache distribué. Le RDD est immuable et il y a deux groupes de fonctions qui peuvent traiter RDDs. Ces sont les fonctions de transformation et d'actions. Les fonctions de transformation permet aux RDDs de croître en une forme de lignée, et les fonctions d'actions coupe la lignée de RDDs, effectue le calcul et renvoie les résultats. La caractéristique de RDD demande une façon différente que les programme conventionnels en terme d'implémentation.

Bien que le Spark gère automatiquement la planification des travaux, la réplication des données et la communication réseau parmi les noeuds, il existe encore de nombreux paramètres à régler afin d'optimiser l'efficacité et l'évolutivité du calcul, par exemple, le nombre de morceaux des RDDs, le nombre de tâches à assigner et la taille de mémoire-cache à utiliser dans chaque noeud. Il est aussi important de contrôler la longueur de la lingée des RDDs. Nous fournissons deux implémentations distribuées,

[1] `https://spark.apache.org/`

le Sim_AHC distribué et la méthode de spectral-SVD distribuée. Pour chaque implémentation, nous illustrons la procédure et la performance du calcul. Nous trouvons que l'implémentation du Sim_AHC distribué ne fonctionne pas aussi bien que prévu. Après avoir essayé des solutions différentes, nous concluons que le Spark est peut-être pas une plate-forme du calcul appropriée pour un algorithme comme Sim_AHC, dans lequel une itération dépend l'itération précédente. Le Spark est plutôt un bon choix pour les algorithmes qui fonctionnent aux itérations indépendantes par lot. Dans cette thèse, nous partageons nos connaissances savantes de nos expériences, en croyant qu'elles seraient utiles pour d'autres chercheurs, qui s'intéressent à la mise en oeuvre de la méthode hiérarchique à l'aide de Spark. Pour l'implémentation distribuée de SHCoClust, nous proposons une façon distribuée de réaliser la méthode de spectral-SVD. Dans nos expériences, nous utilisons des données de tailles différentes pour examiner la caractéristiques de l'échelle.

Mots-clés : classification ascendante hiérarchique, co-clustering, recherche d'informations, l'hypothèse de cluster, calcul distribué.

Bibliography

[1] Julien Ah-Pine and Xinyu Wang. Similarity based hierarchical clustering with an application to text collections. In *International Symposium on Intelligent Data Analysis*, pages 320–331. Springer, 2016.

[2] Godfrey N Lance and William Thomas Williams. A general theory of classificatory sorting strategies: Ii. clustering systems. *The computer journal*, 10(3):271–277, 1967.

[3] Xinyu Wang, Julien Ah-Pine, and Jerome Darmont. Shcoclust, a scalable similarity-based hierarchical co-clustering method and its application to textual collections. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 17), Naples, Italy*, July 2017.

[4] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[5] Nick Jardine and Cornelis Joost van Rijsbergen. The use of hierarchic clustering in information retrieval. *Information storage and retrieval*, 7(5):217–240, 1971.

[6] W Bruce Croft. A model of cluster searching based on classification. *Information systems*, 5(3):189–195, 1980.

[7] Alan Griffiths, Lesley A Robinson, and Peter Willett. Hierarchic agglomerative clustering methods for automatic document classification. *Journal of Documentation*, 40(3):175–205, 1984.

[8] Ellen M Voorhees. The cluster hypothesis revisited. In *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 188–196. ACM, 1985.

[9] Abdelmoula El-Hamdouchi and Peter Willett. Techniques for the measurement of clustering tendency in document retrieval systems. *Journal of Information Science*, 13(6):361–365, 1987.

[10] Alan Griffiths, H Claire Luckhurst, and Peter Willett. Using interdocument similarity information in document retrieval systems. *Readings in Information Retrieval, Morgan Kaufmann Publishers, San Francisco, CA*, pages 365–373, 1997.

[11] Xinyu Wang, Julien Ah-Pine, and Jerome Darmont. A new test of cluster hypothesis using a scalable similarity-based agglomerative hierarchical clustering framework. In *Rencontres Jeunes Chercheurs en Recherche d'Information (CORIA 17), Marseille*, March 2017.

[12] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pages 2–2. USENIX Association, 2012.