UNIVERSITÉ DE SHERBROOKE

DÉPARTEMENT D'INFORMATIQUE

IFT 501 : Recherche d'information et forage de données

TP #2 et 3 — Automne 2017

Analyse des textes par le clustering

À remettre le vendredi 24 novembre 2017

Ce travail consiste à développer une application de clustering pour analyser des textes. Les données à analyser sont des résumés des projets de recherche qui ont été subventionnés par le NSF (National Science Foundation des États Unis). Elles sont de grande quantité et de très grande dimensionalité. Les buts de cette analyse sont multiples. Le clustering de ces données consitue un defi majeur au niveau de la représentation, la comparaison, et le regrougpement (clustering) des documents. D'autre part, l'analyse par le clustering devrait permettre d'identifier la description des thèmes qui sont "cachés" dans les textes pour comprendre finalement quels sont des sujets de recherche qui ont été subventionnés. On pourrait également découvrir si de différents comités de programmes de la foundation subventionne des projets similaires pour mieux comprendre les chevauchements qui pourraient exister entre les comités.

Voici quelques explications de base concernant le système de concours pour les subventions de recherche. Les chercheurs, majoritairement professeurs d'Université, participent à ces concours pour obtenir de l'argent dont ils ont besoin pour effectuer leur recherche scientifique. L'octoi d'une subvention de recherche repose essentiellement sur la qualité du chercheur (ou du groupe de chercheurs) en terme de ses (leurs) réalisations scientifiques passées, l'originalité et la faisabilité du projet (ou du programme) proposé, ainsi que l'historique de la formation des personnels hautement qualifiés. Une fondation nationale comme NSF a le devoir de décider les octois de façon juste et équitable dans le meilleur intérêt du pays. Les évaluations se font par des groupes de paires dans un processus transparent. Chaque domaine important de recherche scientifique a un groupe (ou comité) de paires en charge. Comme il n'y a pas toujours une frontière très claire entre des domaines de recherche, il est donc possible qu'un chercheur ait plusieurs choix des groupes d'évaluation où il peut soumettre son projet.

Ce TP représente 50% des notes des TPs ou 20 points dans les notes finales. Il est à remettre le vendredi 24 novembre 2017. Votre remise doit comprendre un rapport et les programmes que vous aurez développés pour ce TP, de même que des données modifiées ou

nouvelles. La remise doit se faire par **turnin** sur http ://opus.dinf.usherbrooke.ca .

Sujet : Clustering et analyse d'une base de données de textes

Un répertoire de données est fourni dans le répertoire public du cours pour ce TP. Tel que ci-haut mentionné, ces données concernt des informations sur les projets subventionnés par le NSF entre 1990-2003. Par soucis de complétitude, tous les fichiers de cet ensemble de données sont inclus dans le répertoire du TP2_3. Voici les fichiers ou répertoires que vous devez regarder et utiliser.

- Les deux fichiers html qui expliquent l'ensemble des données ;
- Le fichier words.txt qui numérote tous les mots utilisés par l'ensemble des qq 129000 "abstracts". Il y en a 30799 mots (qui sont bien évidement trop nombreux). L'ordre de cette numérotation n'a aucun d'importance cependant. Ce fichier permet aux autres fichiers d'accéder plus directement les mots utilisés par des chercheurs.
- Les fichiers nsfabs_partX_out.zip (ici $X = 1, 2, et 3$). Chaque fichier zip se décompresse en un répertoire contenant quatre fichiers. C'est le fichier docwords.txt qui est le plus utile pour ce TP. docwords.txt contient des informations concernant la fréquence des mots dans chaque document. Les autres fichiers contiennent des informations qui pouront être utiles si vous faite des analyses poussées des résultats. Vous devez décompresser les trois fichiers zip pour récupérer les trois fichiers docwords.txt dont vous avez besoin pour le TP.

Les principales tâches du TP sont listées ci-dessous, et des pécisions leur concernant suivent après.

- Tâche 1 : Réduire, de façon très sigificative, le nombre des mots utilisés comme attributs ; Adopter une représentation des textes ;
- Tâche 2 : Implanter une méthode de clustering en se basant sur le *k-means* ou *FCM* avec ou sans la détermination automatique du nombre de clusters ;
- Tâche 3 : Développer une version étendue de votre méthode implantée pour traiter le problème de la sélection automatique des variables (Voir l'algorithme suggéré ci-dessous) ;
- Tâche 4 : Analyser les clusters obtenus en terme de la "pureté" et des "thèmes".
- Tâche 5 : Développer une version hierarchique pour faire les même analyses.

Tâche 1 - Réduction des mots et représentation des textes : Voici quelques suggestions pour des gens curieux. Toutes les approches ne sont pas exigées, lisez bien les recommandations.

- Par la méthode manuelle : si vous avez la patience, vous pouvez passer à travers les 30799 mots un par un pour éliminer ou fusionner des mots selon votre propre connaissance de l'utilité de chaque mot. Une démarche plus réaliste est peut-être de passer à travers ces mots pour en éliminer certains qui sont clairement inutiles. Vous pouvez choisir à ignorer cette méthode.
- Par des (simples) comparaisons morphologiques pour fusionner des mots similaires. Évidemment, celles-ci comportent des risques de fusionner des mots de forme similaire, mais sémantiquement très différents. C'est un risque à prendre. Il n'y rien de parfait,

de toute façon. Si vous êtes perfectionniste, il existe des outils d'analyse avancée dans le domaine publique tel que *The Stanford NLP* pour les *Stemming and lemmatization* et des dictionnaire des synonymes. La racinisation de Porter [1] serait un bon outil pour faire ce travail.
Une attention particulière devrait être faite à propos de la fréquence des mots après fusion. Les fichiers docwords.txt doivent donc être modifiés en conséquence.

— Par le calcul des poids selon la formule *TF-IDF* (approche recommandée). Je en ai parlé en classe. Vous pouvez trouver sa definition dans le livre de *Tan et al* ou sur la page web Wikipedia. Le TF-IDF donne une indice de l'importance des mots pour la représentation du contenu de texte dans un corpus. En jouant avec le seuillage, vous pouvez réduire autant des mots que vous voulez tout en gardant ceux qui sont les plus utiles (statistiquement parlant !).
Je suggère que vous réduisez 80% (ou plus) des mots en utilisant une combinaison des méthodes suggérées.

Pour la représentation finale, vous pouvez utiliser la représentation *TF-IDF*. Mais la représentation TF, i.e. la simple fréquence des termes, serait préférable, car vous avez déjà utilisé *TF-IDF* pour le prétraitement. D'autre part, des considérations pour l'analyse des résultats de clustering pourraient militent en faveur d'une méthode simple de représentation. Vous êtes cependant libres de choisir.

Tâche 2 - Clustering par *k-means* ou *FCM* ou *k-mode* :

— (mon conseil) Implanter la version de base de l'algorithme de votre choix avec l'initialization au hasard. Vous pouvez choisir d'implanter une méthode améliorée pour l'initialisation. Parlez-en avec moi si vous n'êtes pas certains si ça en vaut la peine.
Évidemment, si vous optez pour le clustering des données catégoriques tel que le k-mode, vous deviez d'abord opter pour une représentation catégoriques des textes, par exemple, binaire.

— Il faut déterminer les valeurs des paramètres. Le plus important est le nombre de clusters. Vous devez faire des essais et erreurs. Une indice de départ est le nombre de programmes de NSF (NSF Program). Malheuresement, vous devez probablement aller chercher ces informations dans les vrais résumés Abstracts_PartX.zip ($X = 1, 2, et 3$). Lors que vous roulez votre programme de clustering, je suggère que vous commenciez avec les valeurs de K égale environ deux fois du nombre de programmes de NSF. Ce n'est qu'une suggestion.

— Optionnellement, vous pouvez implanter un algorithme pour déterminer automatiquement le nombre de clusters. Je vous donnerai quelques suggestions plus tard. Je ne suis pas au courant de cas très bien réussi pour des données de très grandes dimensions comme celui-ci cependant.

Tâche 3 - Sélection de variable durant le processus de clustering :

Ici, on vise à étendre l'algorithme de clustering (que vous avez implanté) afin qu'il puisse identifier les dimensions importantes pour chaque cluster. Cette extension se réalise par l'introduction des poids propres à chaque cluster et par l'integration d'un processus d'opti-

1. http ://tartarus.org/ martin/PorterStemmer/java.txt

misation dans l'algorithme de clustering.

— Pour chaque cluster k et chaque dimension (ou variable) j, créer une variable de poids $w_{k,j}$. Le nombre total de poids $w_{k,j}$ doit être le nombre-de-clusters * nombre-de-mots-utilisés. $w_{k,j} \geq 0$ et la norme des poids pour chaque clusters $||(w_{k,j}|| \equiv C$, i.e. la norme reste constante. C pourrait être 1 ou le nombre-de-mots-utilisés, ou une constante positive quelconque.

— Modification de la mesure de distance. Je présume que vous utilisez la distance Euclidienne dans votre algorithme de clustering et que vous avez besoin de calculer, à chaque itération, la distance entre un objet (texte) x et le centre d'un cluster k, i.e. v_k (si ce n'est pas tout à fait votre cas, signalez moi and on en discutera). Cette distance doit être de forme suivante :

$$d(x, v_k, w_{k,\cdot}) = \sqrt{\sum_{i=1}^{D} w_{k,i}(x_i - v_{k,i})^2} \tag{1}$$

Ici D est le nombre des dimensions (le nombre des mots utilisés).

— Mise-à-jour de l'ensemble des poids $\{w_{k,j}\}$. Cette mise-à-jour se fera après une époque d'itérations de clustering couvrant tous les objets (et non après avoir vu un ou quelques uns des objets seulement). Elle se réalise en deux étapes : 1) Décrémenter $w_{k,j}$ pour pénaliser les dimensions de grande variance. Pour chaque cluster k et chaque dimension j, faire

$$w_{k,j} = \frac{w_{k,j}}{1 + variance\_de\_variable\_j\_dans\_clusterk} \tag{2}$$

Attention : la variance de la variable j dans chaque cluster k se calcule différemment dépendant de votre choix de l'algorithme de clustering. Dans le cas du choix *FCM*, cela inclut l'utilisation de TOUS les objets. Dans ce cas, ce sont les memberships par rapport au cluster k qui sont impliqués dans le calcul. Dans le cas de l'algorithme *k-means*, le calcul est un peu plus simple.

2) La deuxième étape de la mise-à-jour consiste à la normalisation comme suit :

$$w_{k,j}^{new} = \frac{C * w_{k,j}}{\sqrt{\sum_{i=1}^{D} (w_{k,i})^2}} \tag{3}$$

Tâche 4 - Analyse des résultats :

Tel que mentionné plus tôt, il y a deux aspects de cette analyse.

— La "pureté" des clusters en terme la variété des *NSF programs* impliqués dans les demandes de subventions regroupées dans un cluster. Moins qu'il y a des *programs* dans un cluster, plus pur l'est. Attention, on ne fait pas un concours de pureté. En autre termes, il ne faut pas chercher tous les moyens, par exemple par augmentation du nombre de clusters, pour augmenter la pureté de chacun.

— La recherche l'existance d'un "Thème" dans chaque cluster. Ici, un thème correspond à une certaine mesure de distribution des mots les plus importants pour chaque cluster. Dans le cas de l'extension de l'algorithme implanté, cette tâche est facile. Vous allez just ordonner toutes les valeurs de $w_{k,j}$ pour le cluster k. S'il y a une minorité des valeurs $w_{k,j}$ qui sont significatives, alors vous avez un thème bien ressortie de ce cluster. Ça ne veut pas dire que ce thème est sémantiquement utile ou intéressant. Mais, au moins c'est quelque chose qu'on peut s'en servir.

Dans le cas de l'algorithme de base, l'analyse directe est plus compliquée. Je vous laisse

de concevoir une méthode. Un façon de faire est de calculer un indice de variation en s'inspirant de la formule de mise à jour des $w_{k,j}$. En tout cas, dans un cluster, plus qu'il y a des variations dans une variable, moins qu'elle soit importante dans la formulation de ce cluster (et le thème associé à ce cluster).

Tâche 5 - Développer une version hierarchique (divisive) à partir d'un algorithme de clustering que vous avez implanté :

Le but de cette tâche est de se familiariser avec (et d'explorer) le clustering hierarchique. Pour cette tâche, vous pouvez choisir soit l'algorithme de base de la tâche 2, soit l'algorithme intégrant la sélection de variables de la tâche 3. Vous devez réaliser le clustering selon la bisection, c'est-à-dire, selon le processus suivant :

- Appliquer l'algorithme de clustering par partition pour faire une séparation de données en deux clusters ; Les deux clusters obtenus sont des "feuilles" de l'arbre en construction.
- identifier un cluster feuille à diviser (vous devez développer votre critère pour choisir le cluster), puis appliquer l'algorithme de clustering par partition pour faire une séparation de données en deux clusters afin d'ajouter deux branches d'arbre à l'arbre en construction. Puis itérer sur ce processus.
- Vous pouvez choisir une condition d'arrêt simple telle que le nombre de feuilles dépassant un seuil. Vous pouvez aussi développer une mesure pour arrêter le processus automatiquement.
- Finalement, vous devez faire une analyse de performance en sélectionnant soit le critère de pureté, soit l'existance des thèmes.

Comme pour le TP1, vous avez toujours beaucoup de liberté pour développer vos propres solutions. Vous devez faire preuve d'imagination. Il n'y a pas de meilleure solution et ce n'est pas le but du TP non plus. Le développement d'un esprit critique, la recherche de solutions et le savoir de "se défendre" (justification) est bien plus important. C'est pourquoi, vous devez mettre du temps et de l'energie pour bien rédiger votre rapport. Votre rapport doit décrire clairement les différentes étapes de traintement incluant les prétraitements effectués, les résultats obtenus, vos commentaires et vos conclusions.

N'oubliez pas de remettre aussi vos données si elles sont modifiées des données originales. Normalement, le correcteur ne regarde pas les prétraitements s'ils ne sont pas intégrés dans le programme principal.