Classification non supervisée et visualisation 3D de documents

Nicolas Bonnel, Annie Morin, Alexandre Cotarmanac'H

Classification non supervisée et visualisation 3D de documents

Nicolas Bonnel*,**, Annie Morin*, Alexandre Cotarmanac'h**

*IRISA, Campus Universitaire de Baulieu
Avenue du Général Leclerc, 35042 Rennes Cedex - France
{nicolas.bonnel,annie.morin}@irisa.fr
** France Telecom R&D, 35512 Cesson-Sévigné Cedex

Résumé. Le nombre de documents issus d'une requête sur le Web devient de plus en plus important. Cela nous amène à chercher des solutions pour aider l'utilisateur qui est confronté à cette masse de données. Une alternative possible à un affichage linéaire d'une liste triée selon un critère, consiste à effectuer une classification des résultats. C'est dans ce but que l'on s'intéresse aux cartes auto-organisatrices de Kohonen qui sont issues d'un algorithme de classification non supervisée. Cependant il faut ajouter des contraintes à cet algorithme afin qu'il soit adapté à la classification des résultats d'une requête. Par exemple, il doit être déterministe. De plus, la classification obtenue dépend fortement de la distance utilisée pour comparer deux documents. On évalue alors l'impact de différentes distances ou dissimilarités, afin de trouver la plus adaptée à notre problème. Un compromis doit également être trouvé entre le temps d'exécution de l'algorithme et la qualité de la classification obtenue. Pour cela, l'utilisation d'un échantillonnage est envisagée. Enfin, ces travaux sont intégrés dans un prototype qui permet de visualiser les résultats en trois dimensions et d'interagir avec eux.

1 Introduction

Avec l'augmentation constante des données disponibles sur le World Wide Web, il devient de plus en plus difficile d'extraire l'information pertinente pour une recherche donnée. Les moteurs de recherche, qui sont un moyen de représentation du Web pour les utilisateurs, retournent un nombre si important de résultats qu'il faut chercher de nouvelles méthodes de gestion de ces résultats. En effet, il devient nécessaire de trouver une alternative au simple affichage de listes ordonnées selon un seul critère (généralement un rang représentant la "pertinence").

Les résultats ou documents que l'on cherche à classer sont des pages Web. Seule la partie textuelle de ces documents est utilisée. Elle permet d'obtenir une représentation vectorielle (vecteurs de mots) qui est largement utilisée dans le domaine de la recherche d'informations. Ce sont ces vecteurs qui servent de données d'entrées pour la classification. On s'oriente vers des méthodes de classification automatique et plus particulièrement vers les cartes auto-organisatrices. La classification obtenue est ensuite

Revue des Nouvelles Technologies de l'Information (RNTI-E-3), Numéro spécial - Actes des Cinquièmes Journées Francophones "Extraction et Gestion des Connaissances" (EGC'05), volume 2, pages 557-562, Paris, France, 19-21 janvier 2005.

proposée à l'utilisateur via une métaphore de visualisation 2D ou 3D. Le choix de la méthode de visualisation est important car il doit mettre en avant l'efficacité de l'organisation des résultats. C'est dans cet objectif que l'on s'intéresse de plus en plus à des visualisations 2D (graphes ou cartes) ou 3D (graphes, paysages ou villes). Cependant, les visualisations 2D ne semblent pas toujours adaptées face à l'augmentation des résultats (absence d'une vue d'ensemble, illisibilité des graphes). Contrairement à la 2D, la 3D va permettre d'afficher un grand nombre de résultats, qui n'est pas limité par la taille de l'écran mais par la perception de l'utilisateur. Dans cet article, le choix d'une visualisation 3D a été fait. La 3D, proche de l'esprit humain d'un point de vue cognitif, offre de nouvelles possibilités d'interactions mais elle rend aussi la navigation dans l'espace indispensable et plus complexe. Enfin, on cite la méthode AVE (Wiza et al. 2004) et son système Periscope qui sont des travaux proches des notres sur l'aspect visualisation et utilisant aussi des interfaces mixtes (scène 3D et interface 2D).

Ce travail s'intéresse à la classification non supervisée des documents ainsi qu'à leur visualisation 3D. Dans la section suivante, on traite la partie classification des documents dont l'efficacité influe directement sur la pertinence de la représentation donnée à l'utilisateur. Puis, on présente une visualisation 3D de notre prototype. Enfin la dernière section permet de faire le bilan.

2 Classification non-supervisée des documents

Il existe de nombreuses techniques de classification de données. Dans cette section, l'objectif est d'obtenir, par un processus automatique, une classification efficace des documents. Pour cela, on a choisi de s'intéresser aux méthodes de classification non-supervisée et plus particulièrement d'utiliser les cartes auto-organisatrices de Kohonen (Kohonen 1995). Ce choix est motivé par certaines propriétés de cette méthode. Elle est non supervisée ; elle possède une notion de voisinage (deux documents voisins sur la carte correspondent à deux documents ayant des vecteurs de mots proches) ; et elle organise les documents sur une grille de dimension pré-définie, ce qui garantit une bonne utilisation de l'espace lors de la visualisation. Les cartes auto-organisatrices permettent aussi d'avoir différents niveaux de hiérarchie ou encore une taille de carte dynamique (Dittenbach et al. 2000), ce qui peut se révéler intéressant dans notre cadre applicatif.

2.1 Choix d'une distance

La version classique des cartes de Kohonen utilise la distance euclidienne pour comparer les vecteurs documents aux vecteurs neurones. Cette distance n'est pas forcément la plus adaptée pour le cas des vecteurs de mots. On s'intéresse alors à d'autres approches telles que la distance du χ^2, la pondération *tf.idf*[1] (Sparck Jones 1972) ou encore une combinaison $\chi^2/tf.idf$ (obtenue par moyenne des deux pondérations). En fait, ces approches ne sont que des pondérations différentes de la distance euclidienne.

[1] *tf* pour *term frequency* et *idf* pour *inverse document frequency*

Soit un corpus composé de D documents et de M mots, où $\mathbf{x_j}$ représente le *jème* document du corpus. $\mathbf{x_j}$ est un vecteur de dimension M où chaque dimension représente un mot du corpus noté w_p. Ainsi, $\mathbf{x_j}_p$ représente simplement le nombre d'occurrences du mot w_p dans le document $\mathbf{x_j}$. On définit aussi les expressions suivantes :

$$X = \sum_j \sum_p \mathbf{x_j}_p \quad , \quad x_{.p} = \sum_j \mathbf{x_j}_p \quad , \quad x_{j.} = \sum_p \mathbf{x_j}_p \quad et \quad \forall p \;\; \bar{\mathbf{x_j}}_p = \frac{\mathbf{x_j}_p}{x_{j.}} \tag{1}$$

On note alors $\bar{\mathbf{x_j}}$ le profil-ligne d'un document et chaque profil-ligne représente une distribution conditionnelle. Soit une carte composée de N neurones où $\mathbf{m_i}$ représente le *ième* neurone. $\mathbf{m_i}$ est également de dimension M et peut être assimilé à un document fictif. On note C_i l'ensemble des documents associés au neurone i. La distance entre un document et un neurone peut alors être définie par les relations suivantes (la première pour l'approche du χ^2 et la seconde pour l'approche *tf.idf*) :

$$d^2_{\chi^2}(\bar{\mathbf{x_j}}, \mathbf{m_i}) = \sum_p \frac{X}{x_{.p}} (\bar{\mathbf{x_j}}_p - \mathbf{m_i}_p)^2 \quad , \quad d^2_{tf.idf}(\bar{\mathbf{x_j}}, \mathbf{m_i}) = \sum_p ((\log \frac{D}{n_p})^2 \times (\bar{\mathbf{x_j}}_p - \mathbf{m_i}_p))^2 \tag{2}$$

où n_p est le nombre de documents du corpus contenant le mot w_p. Cependant, ayant des matrices documents/mots très creuses, ces deux approches deviennent très sensibles aux mots absents. Une solution consiste alors à calculer la distance entre deux documents en utilisant uniquement les mots appartenant à l'intersection des deux documents. Certaines mesures utilisent déjà ce principe comme la dissimilarité de Kullback-Leibler. Nous utilisons ici une version symétrisée de cette dissimilarité (Lebart et Rajman 1998) :

$$d(\bar{\mathbf{x_j}}, \mathbf{m_i}) = \sum_{p, \mathbf{m_i}_p . \mathbf{x_j}_p \neq 0} (\bar{\mathbf{x_j}}_p - \mathbf{m_i}_p) \log \frac{\bar{\mathbf{x_j}}_p}{\mathbf{m_i}_p} \tag{3}$$

On peut aussi introduire une pondération (de type χ^2 ou *tf.idf*) dans le calcul cette dissimilarité. Des mesures[2] ont été effectuées sur ces distances (Tableau 1). L'erreur calculée correspond à l'erreur moyenne de quantification (MQE) de la carte auto-organisatrice :

$$MQE = \frac{1}{N} \sum_i \frac{1}{|C_i|} \sum_{\mathbf{x_j} \in C_i} ||\bar{\mathbf{x_j}} - \mathbf{m_i}|| \tag{4}$$

Pour que ces erreurs soient comparables, les paramètres des cartes ainsi calculées sont fixés et identiques (parmi ces paramètres : carte de taille 8×8, corpus de 8570 documents représentés par des vecteurs en 164 dimensions). On cherche alors la distance la plus adaptée pour la recherche d'informations textuelles. Mais une évaluation basée sur l'erreur de quantification semble insuffisante pour départager ces distances, bien que la dissimilarité de Kullback-Leibler ait une erreur plus faible. De plus, la définition proposée pour l'erreur de quantification favorise la distance euclidienne par rapport

[2]Les temps d'exécution n'ont pas été optimisés.

Distance	Euclidienne	*tf.idf*	χ^2	χ^2/*tf.idf*	Kullback Leibler
MQE	0.408	0.428	0.430	0.428	0.301
Temps (s)	40479	48869	49350	48475	4452

TAB. 1 – Calcul de l'erreur moyenne selon les distances utilisées.

Facteur	MQE_{app}	MQE	Temps (s)
0.5	0.419	0.423	16736
0.25	0.420	0.439	8343

TAB. 2 – Calcul de l'erreur moyenne sur différents échantillons.

aux pondérations (χ^2 et *tf.idf*). Une solution serait alors de modifier ce critère afin qu'il prenne en compte les différentes pondérations. On doit aussi considérer d'autres critères d'évaluation (Lesot et al. 2003), d'autres corpus ou encore une évaluation par l'utilisateur. Des tests réalisés sur des corpus de plus petite taille ont cependant révélé de meilleurs résultats (d'un point de vue organisation) pour la distance du χ^2.

2.2 Application aux cartes auto-organisatrices

Dans l'algorithme traditionnel des cartes auto-organisatrices, on remplace la distance euclidienne par celle du χ^2. Dans l'attente de résultats plus complets sur les différentes distances et notamment sur la dissimilarité de Kullback-Leibler, on choisit la distance du χ^2 qui offre bien souvent une meilleure organisation des données (surtout sur de petits corpus, ce qui est le cas dans notre prototype). Les neurones sont étiquetés par la méthode labelSOM (Rauber et al. 2001). Dans le cas de la recherche d'informations, on souhaite obtenir une classification déterministe, ce qui nous impose de choisir la version *batch* de l'algorithme ainsi qu'une initialisation fixe des neurones.

Afin de réduire le temps de calcul, on décide d'effectuer l'apprentissage de la carte sur un échantillon choisi aléatoirement. Les données n'appartenant pas à l'échantillon sont aussi projetées sur la carte à la fin de l'algorithme. On s'intéresse alors à l'impact de l'échantillonnage sur l'erreur de quantification moyenne et sur le temps de calcul. Le tableau 2 montre alors les résultats obtenus (avec la distance euclidienne) pour différentes tailles d'échantillon qui sont définies par un facteur multiplicatif appliqué à la taille du corpus. Les paramètres utilisés sont ceux de la sous-section précédente et MQE_{app} est l'erreur de quantification moyenne liée uniquement aux données de l'échantillon. L'échantillonnage utilisé ici nécessite peu de calculs préalables et permet une diminution linéaire du temps d'exécution, sans trop augmenter l'erreur de quantification. Cette dernière permet d'évaluer la qualité d'un groupe mais ne prend pas en compte l'organisation des groupes les uns par rapport aux autres. Nous envisageons donc de tester l'impact de l'échantillonnage sur d'autres critères d'évaluation.

FIG. 1 – Prototype SmartWeb avec une métaphore de visualisation de type ville.

3 Visualisation des documents

Les travaux présentés dans la section précédente entrent dans le cadre d'un prototype développé par France Telecom R&D. Ce prototype, appelé SmartWeb, s'apparente à un moteur de recherche classique du point du vue requête et base de données. Par contre l'objectif est de fournir à l'utilisateur les meilleures organisation et visualisation possibles des résultats de sa requête, sans solliciter une quelconque intervention de sa part dans le processus. Les points essentiels de ce prototype sont l'organisation des données (détaillée à la section précédente) et leur visualisation interactive.

L'architecture du prototype possède un côté serveur et un côté client. Le premier est constitué d'une base de données et d'un ensemble de modèles d'interfaces. Le second est composé d'une page HTML regroupant une applet Java (interface 2D) et un navigateur VRML (interface 3D). L'interface graphique et les interactions sont générées dynamiquement par l'applet. On propose ici une visualisation 3D des résultats (Figure 1). L'objectif est de voir dans quelle mesure l'ajout d'une dimension nous permet d'améliorer la visualisation. Cependant, la visualisation des résultats est fortement dépendante de nombreux critères tels que l'objectif de la recherche, la catégorie de l'utilisateur ou encore le type et le nombre de résultats. C'est pourquoi il n'existe pas de solution unique en terme de visualisation. Le prototype possède donc une caractéristique intéressante : l'adaptabilité de la visualisation en fonction de certains critères.

4 Conclusion

Cet article présente une méthode de classification et de visualisation en 3D de résultats issus d'une requête. La visualisation 3D proposée a l'avantage d'être interac-

tive, adaptative et générée dynamiquement. Concernant la classification, elle repose sur une carte auto-organisatrice particulière de par la distance utilisée (distance du χ^2). Mais l'évaluation de l'influence des différentes distances proposées va être appronfondie. L'organisation des résultats peut aussi être améliorée en intégrant une classification hiérarchique ascendante dans les cartes auto-organisatrices ; bien que cet ajout nécessite une réflexion sur les modifications à apporter à la visualisation et notamment aux liens entre les différents niveaux hiérarchiques. L'idée est alors de pouvoir dégager différents sens sémantiques de la requête lorsque l'on monte dans la hiérarchie. Un autre point à développer concerne le temps d'exécution de l'algorithme d'organisation.

Références

Dittenbach M., Merkl D. et Rauber A. (2000), Using growing hierarchical self-organizing map, Proceedings of European Symposium on Artificial Neural Networks, pp 7-12.

Kohonen T. (1995), Self-Organizing Maps, Springer.

Lebart L. et Rajman M. (1998), Similarités pour données textuelles, Proceedings of 4th International Conference on Statistical Analysis of Textual Data, pp 545-555.

Lesot M.-J., d'Alché-Buc F. et Siolas G. (2003), Evaluation des cartes auto-organisatrices et de leur variante à noyaux, Actes de la conférence CAp.

Rauber A. et Merkl D. (2001), Automatic Labeling of Self-Organizing Maps for Information Retrieval, JSRIS, 10(10) 23-45.

Sparck Jones K. (1972), A statistical interpretation of term specificity and its application in retrieval, Journal of Documentation, Vol. 28(1), pp 11-20.

Wiza W., Walczak K. et Cellary W. (2004), Periscope - A System for Adaptive 3D Visualization of Search Results, Proc. of Int. Conf. on 3D Web technology, pp 29-40.

Summary

The results number of a web query becomes greater and greater. That's why we need to find solutions to help users facing this amount of data. A solution to replace the linear display of a sorted list consists in making a results classification. For this purpose we are interested in Kohonen self-organizing maps that are computed by an unsupervised classification algorithm. However we need to add constraints to this algorithm in order to adapt it to a query results classification. For example, it has to be determinist. Moreover the obtained classification depends on the distance used to compare two documents. Then we evaluate the impact of different distances and dissimilarities in order to find the most adapted to our case. A compromise need also to be found between the computing time and the quality of the obtained classification. To do that we use a sampling method. At least these works are integrated in a prototype which allows to visualize results in three dimensions and to interact with them.

Revue des Nouvelles Technologies de l'Information (RNTI-E-3), Numéro spécial - Actes des Cinquièmes Journées Francophones "Extraction et Gestion des Connaissances" (EGC'05), volume 2, pages 557-562, Paris, France, 19-21 janvier 2005.