Sur l'utilisation de LDA en RI Pair à Pair

Sylvie Cazalens, Esther Pacitti, Sylvie Calabretto, Yulian Yang

Sur l'utilisation de LDA en RI pair-à-pair

S. Cazalens[*] — Y. Yang[**] — S. Calabretto[**] — E. Pacitti [***]

[*] *LINA - UMR 6241*
Université de Nantes - 2, rue de la Houssinière. F44322 Nantes Cedex

[**] *LIRIS - CNRS - UMR 5205*
Université de Lyon - Campus de la Doua, F69622 Villeurbanne

[***] *Université de Montpellier 2, INRIA et LIRMM, équipe Zenith*
161 rue Ada, F34095 Montpellier Cedex 5

Sylvie.Cazalens@univ-nantes.fr, {Sylvie.Calabretto, Yulian.Yang}@liris.cnrs.fr, Esther.Pacitti@lirmm.fr

RÉSUMÉ. *Nous revisitons la problématique de définition d'un système de Recherche d'Information pair-à-pair lorsque le profil thématique associé à chaque pair est obtenu par l'Allocation Latente de Dirichlet. Cette méthode, pensée pour une collection centralisée, offre une représentation riche des thèmes et des documents. Nous décrivons deux façons de la mettre en oeuvre dans un système distribué et analysons leurs avantages et inconvénients. Puis nous illustrons l'utilisation de ces profils thématiques dans deux systèmes différents. L'un, non structuré, se base sur l'utilisation d'un algorithme épidémique pour regrouper dynamiquement les pairs proches d'un point de vue thématique. Cela nécessite de définir une mesure de similarité entre profils. L'autre utilise des super-pairs et maintient un index thématique des pairs du système, mémorisé dans une table de hachage distribuée. Les clés sont calculées à partir des profils thématiques.*

ABSTRACT. *We revisit the problem of defining a peer-to-peer system for Information Retrieval when each peer's topic-based profile is obtained using Latent Dirichlet Allocation. This method, defined for a centralized collection, provides a rich representation of the topics and of the documents. We describe two ways of using it in a distributed system and analyze their advantages and drawbacks. Then, we illustrate the use of the obtained topic-based profiles within two systems. The first one is unstructured and uses a gossip-based algorithm to obtain dynamic overlays of topically related peers. This requires defining a similarity between profiles. The second one uses super-peers and maintains a topic-based index of the peers, which is recorded in a distributed Hash table. The keys are derived from the topic-based profiles.*

MOTS-CLÉS : *Recherche d'information, systèmes pair-à-pair, Allocation Latente de Dirichlet (LDA).*

KEYWORDS: *Information retrieval, P2P systems, Latent Dirichlet Allocation (LDA).*

1. Introduction

Les systèmes pair-à-pair (P2P), où chaque pair joue à la fois le rôle de client et de serveur, sont reconnus pour leurs propriétés de passage à l'échelle, tolérance aux panne, dynamicité ainsi qu'autonomie car chaque pair peut choisir les ressources qu'il souhaite partager.

Un système de Recherche d'Information (RI) pair-à-pair doit proposer une architecture et des protocoles qui assurent la gestion de l'organisation choisie ainsi que l'évaluation des requêtes initiées par les pairs pour obtenir les documents les plus pertinents présents au sein du système. Un exemple d'initiative en ce sens est Sciencenet (cf. Lütjohann *et al.* (2011)) qui propose un moteur de recherche distribué basé sur une technologie P2P. Nous nous intéressons ici aux systèmes où *(i)* les pairs publient des documents indépendamment les uns des autres ; *(ii)* il n'y a pas d'index terminologique global ; *(iii)* chaque pair gère un moteur de recherche local basé sur une indexation et une politique de diffusion qui lui sont propres.

L'évaluation d'une requête dépend de l'architecture considérée, mais il est possible d'utiliser pour l'améliorer des connaissances supplémentaires, tels des centres d'intérêts comme dans Bertier *et al.* (2010) ou des thèmes comme dans Crespo et Garcia-Molina (2004). Dans le cas où le contenu d'un pair est décrit par un ensemble de thèmes, la première étape de l'évaluation d'une requête consiste à : *(i)* extraire les thèmes de la requête ; *(ii)* trouver des pairs pertinents d'un point de vue thématique ; *(iii)* leur envoyer la requête. En particulier, lorsque la requête concerne les thèmes du pair initiateur et que ses voisins sont thématiquement proches de lui, la requête peut être évaluée de manière très satisfaisante en n'interrogeant que son voisinage proche.

Dans les travaux de Crespo et Garcia-Molina (2004), la représentation d'un thème se réduit à un simple mot. L'objectif de cet article est de revisiter la problématique de définition d'un système P2P lorsque les thèmes sont définis en utilisant l'Allocation Latente de Dirichlet (LDA) qui offre une représentation plus riche d'un thème et des documents. Nous présentons d'abord cette méthode et comment calculer les thèmes dans un système P2P (Section 2). Puis nous décrivons les grandes lignes de deux organisations : l'une basée sur le regroupement thématique (Section 3), l'autre utilisant un index thématique (Section 4).

2. Extraction des thèmes par LDA

L'allocation latente de Dirichlet (LDA) (cf. Blei *et al.* (2003)) permet de calculer les thèmes d'une collection de documents. C'est un modèle génératif probabiliste où chaque document est vu comme un mélange de thèmes. Chaque thème correspond à une distribution de probabilités sur l'ensemble des mots de la collection. Par exemple, avec une collection de 230887 résumés d'articles MEDLINE du corpus TREC9 et 100 thèmes, les 10 mots de plus forte pondération obtenus pour un thème sont :
$(cells, 0.0271)$ $(human, 0.0203)$ $(hours, 0.0127)$ $(tumor, 0.0102)$ $(line, 0.0077)$
$(skin, 0.0068)$ $(culture, 0.0060)$ $(mice, 0.0060)$ $(malignant, 0.0051)$

$(study, 0.0043)$.
En pratique, on considère les k mots les plus pondérés.

Le nombre n_t de thèmes est un paramètre qui doit être fixé, le processus fournissant n_t thèmes et pour chaque document, dans quelle proportion il se rapporte à chacun des thèmes. Par exemple, s'il y a trois thèmes T_1, T_2, T_3, un document peut relever à 30% du thème T_1, à 30% du thème T_2, et à 40% du thème T_3. Comme souligné dans Deveaud *et al.* (2013), une bonne estimation du paramètre n_t peut être obtenue en testant le processus avec plusieurs valeurs. Pour chaque modèle correspondant, on calcule la distance entre chaque paire de thèmes. En sommant ces distances, on obtient une valeur globale de dis-similarité. On retient la valeur de n_t correspondant au modèle pour lequel la valeur globale est maximale.

Dans notre contexte, chaque pair gère une collection de documents et définit l'ensemble, éventuellement ordonné, de thèmes qui décrit le mieux son contenu. Nous appelons *profil thématique* un tel ensemble. Deux approches sont possibles, selon que le système calcule des thèmes de référence ou non.

2.1. *Approche 1 : obtention de thèmes de référence*

Sur un serveur central S connu de tous les pairs, LDA est mis en oeuvre pour calculer les thèmes de référence du système et un identifiant est associé à chacun d'eux. Tout pair peut interroger S et recevoir la liste des thèmes de référence qui spécifie les mots associés à chaque identifiant. Le pair peut alors utiliser LDA localement pour calculer la proportion de chaque thème dans ses documents. Pour définir son profil, il choisit au plus Max_{profil} thèmes qui le représentent le mieux, où Max_{profil} est le nombre de thèmes maximum autorisé. Il peut par exemple considérer le nombre de documents qui relèvent d'un thème au dessus d'un certain pourcentage comme dans Draidi *et al.* (2011).

La question principale consiste à déterminer sur quelle collection de documents se base le serveur S. Il est peu vraisemblable qu'il puisse disposer de la totalité des collections de tous les pairs du système. Il faut donc qu'il puisse travailler sur une collection représentative de l'ensemble des pairs. Dans Draidi *et al.* (2011), chaque pair envoie à S un ensemble de documents limité mais représentatif de sa propre collection.

Le profil d'un pair étant constitué d'un simple ensemble d'identifiants, si t_{id} est la taille maximum d'un identifiant, la taille d'un profil peut atteindre $Max_{profil} \times t_{id}$ octets. Cette représentation compacte est un avantage car les pairs peuvent échanger des informations sur leur profil sans risquer de surcharger le réseau. De plus, elle permet de calculer la similarité entre deux profils en utilisant des mesures usuelles.

Un point négatif est la présence d'un serveur centralisé dans un système largement distribué. Un autre est l'obtention d'une collection réellement représentative de l'en-

semble des pairs au niveau de S. En effet, si elle ne l'est pas, un pair peut avoir des difficultés à définir un profil qui corresponde effectivement à son contenu.

2.2. *Approche 2 : absence de thèmes de référence*

A l'inverse de l'approche précédente, il n'y a pas de serveur central. Chaque pair utilise LDA pour calculer directement ses propres thèmes sur la base de sa seule collection de documents, et donc indépendamment des autres. La façon dont le pair définit son profil thématique n'est pas affectée et il peut procéder comme expliqué précédemment.

L'avantage de cette approche est l'absence de serveur centralisé qui calcule des thèmes de référence. Elle est donc plus cohérente dans un système largement distribué. De plus, chaque pair peut choisir les thèmes constituant son profil. Il s'en suit une meilleure représentativité du contenu du pair.

Néanmoins, chaque thème doit être décrit par les mots qu'il contient. En supposant que les pairs utilisent au maximum k mots par thème, et que la taille maximum d'un mot est t_{Max_mot} la taille d'un profil peut atteindre $Max_{profil} \times k \times t_{Max\_mot}$. Ceci est à considérer lorsque les pairs échangent des informations sur leurs profils, afin de ne pas surcharger le réseau. De même, le calcul de la similarité entre deux profils est plus complexe car il faut comparer des thèmes définis sur des vocabulaires qui ne partagent pas tous les mots.

3. Organisation basée sur le regroupement thématique

Nous nous situons dans le cadre d'un système non-structuré. Chaque pair stocke ses propres données et les index correspondant. Il maintient une vue partielle du système où chaque entrée correspond à la description d'un autre pair. Des travaux récents, cf. Jelasity *et al.* (2009) ou Bertier *et al.* (2010), ont montré l'apport des algorithmes épidémiques pour créer des groupes dynamiques de pairs similaires. Nous décrivons comment il est possible de rapprocher les pairs selon leur profil thématique.

Algorithme de création des groupes thématiques. Chaque pair calcule son profil thématique, constitué de thèmes et de son numéro IP. Il le met à jour en cas de modification conséquente de sa collection. Les pairs échangent leurs profils de sorte que la topologie du système évolue. Tout pair est constitué de deux threads : un actif et un passif. Via son thread actif, chaque pair p initie régulièrement une communication avec un autre pair, choisi aléatoirement. Quand un pair p' est contacté, sur son thread passif, il doit répondre en renvoyant une liste de profils. Les deux pairs reçoivent des informations qu'ils utilisent pour construire leur vue partielle du système. Chaque pair p doit :

1) *Sélectionner un pair avec qui échanger.* Il sélectionne aléatoirement un pair p' dans sa vue locale. Il connait donc le profil du pair p'. Si sa vue locale ne contient pas

assez de profils pour découvrir de nouveaux voisins, le service d'échantillonnage de pairs défini par Jelasity *et al.* (2004) est invoqué pour rafraîchir la vue.

2) *Sélectionner les profils à envoyer.* Il ordonne les profils de sa propre vue selon leur similarité décroissante avec celui de p' et lui envoie les profils les plus proches. Ceci permet de faire converger l'algorithme plus rapidement. Le nombre de profils envoyés est étudié pour ne pas surcharger le réseau.

3) *Traiter les profils reçus.* Il réalise l'union des profils de sa vue locale et des profils envoyés par p'. Si l'espace de stockage est limité, seules les v_{max} entrées sont conservées. Les n profils les plus proches du profil de p définissent ses voisins.

Quand un nouveau pair arrive dans le système, il obtient d'abord des voisins quelconques. Au cours des échanges, il découvre des pairs dont le profil est proche du sien et il les rajoute à sa vue.

Similarités entre profils thématiques. L'algorithme de création des groupes repose sur la définition d'une similarité/dis-similarité entre profils thématiques. Celle-ci dépend de leur contenu et de la représentation choisie pour les thèmes (cf. Section 2).

Présence d'un ensemble de thèmes de référence. Un profil est représenté par un simple ensemble d'identifiants. Parmi les similarités usuelles entre ensembles, le coefficient de Dice utilisé par Draidi *et al.* (2011) permet de considérer le nombre de thèmes non partagés des deux profils, tout en étant moins pénalisant que le coefficient de Jaccard. Il est défini entre deux profils p et p' par la formule :

$$sim_{Dice}(p, p') = \frac{2 \cdot \mid p \cap p' \mid}{\mid p \mid + \mid p' \mid}$$

Absence d'un ensemble de thèmes de référence. Chaque profil contient la description de chacun des thèmes le constituant. En notant $d(T_i, T'_j)$ la distance entre deux thèmes, nous proposons de définir la dis-similarité entre deux profils p et p' par :

$$d_{iss}(p, p') = \frac{1}{2} \cdot \left(\frac{\sum_{i=1}^{|p|} \min_{T'_j \in p'} d(T_i, T'_j)}{\mid p \mid} + \frac{\sum_{j=1}^{|p'|} \min_{T_i \in p} d(T'_j, T_i)}{\mid p' \mid}\right)$$

En considérant uniquement le poids des mots dans les thèmes, on peut par exemple définir $d(T_i, T'_j)$ comme le cosinus de l'angle formé par les deux vecteurs. Si l'on veut considérer de plus l'ordre des mots tel que produit par LDA, les travaux de Kumar et Vassilvitskii (2010) peuvent être adaptés.

Interrogation. Cette organisation rend très efficace le traitement de toute requête en rapport avec le profil thématique du pair qui l'a émise, car les pairs les plus susceptibles de répondre sont dans son voisinage proche. Pour les autres requêtes, le pair doit faire appel à des pairs quelconques jusqu'à trouver des pairs dont le profil thématique correspond à la requête.

4. Organisation basée sur un index thématique

L'indexation thématique vise à indexer les pairs d'un réseau P2P en fonction des thèmes des documents contenus dans les pairs. Ce type d'indexation peut fournir un service de recherche d'information efficace. Nous considérons ici un réseau P2P dynamique où des pairs rejoignent ou quittent le réseau fréquemment, et où le contenu des pairs évolue régulièrement. Ces comportements dynamiques nécessitent une structure de réseau flexible. C'est pourquoi nous avons implémenté cette notion d'indexation thématique dans une architecture de réseau P2P super-pairs comportant deux couches logiques : dans la couche haute, les super-pairs sont utilisés pour construire et gérer les index thématiques (services d'indexation et de recherche). Dans la couche basse, tous les pairs du réseau communiquent entre eux par gossiping (cf. Jelasity *et al.* (2009)). Ce mécanisme d'indexation permet non seulement de retrouver des pairs pertinents pour une requête donnée mais également de retrouver les pairs similaires pour un nouveau pair qui rejoint le réseau.

Indexation thématique : Nous utilisons Chord pour déployer les index thématiques (cf. Stoica *et al.* (2003)). Chord est un protocole pour les tables de hachage distribuées (DHT) en pair-à-pair. Une DHT stocke des paires clé-valeur. Elle est gérée de manière distribuée dans le réseau en assignant des clés aux différents pairs. Un pair va stocker les valeurs de toutes les clés dont il est responsable. Le protocole Chord spécifie comment les clés sont assignées aux pairs, et comment un pair peut découvrir la valeur d'une clé donnée en localisant tout d'abord le pair responsable de la clé. Une clé est définie comme la valeur de hachage d'un thème ou d'un ensemble de thèmes, et une valeur est définie comme un tuple d'informations représentant un pair dans le réseau.

Les index sont construits de manière incrémentale lorsque les pairs sollicitent le service de recherche. Chaque pair du réseau va alors interroger la DHT pour retrouver des pairs similaires et se connecter à ces pairs. Si aucune réponse n'est retournée (il n'existe aucun pair similaire), le pair effectue alors du gossiping avec ses voisins de manière aléatoire pour trouver des pairs similaires. Ce pair est ensuite indexé dans la DHT par ses thèmes et pourra ainsi être retourné comme réponse à une prochaine requête. Si la DHT retourne une liste de pairs similaires, ce pair se connecte à ces pairs et il est également indexé dans la DHT. Ainsi, le même processus s'applique à tous les pairs qui interrogent la DHT, qui, au final, représente une table de stockage des index thématiques des pairs du réseau.

Le profil thématique d'un pair p_i, est représenté par un ensemble de thèmes ordonné de manière décroissante selon leur poids $T_i = \{t_{i,1}^{w_{i,1}}, t_{i,2}^{w_{i,2}}, t_{i,3}^{w_{i,3}}, ...t_{i,k_i}^{w_{i,k_i}}\}$. L'indexation thématique du pair s'effectue par insertion et division.

Insertion : (i) lorsqu'un pair ne reçoit pas de réponse pertinente de la DHT, nous sélectionnons le thème de poids le plus élevé de sa représentation thématique et obtenons ainsi une paire (thème, pair). S'il existe un autre thème avec un poids proche du thème principal selon un seuil ε, nous le sélectionnons également pour former une autre paire (thème, pair). Les paires (thème, pair) sont codées par des paires (clé,

valeur) dans la DHT. La clé constitue la valeur de hachage du thème, et la valeur correspond au pair. Une valeur est formellement définie par $< T_{index}, T_{left}, ip >$ où T_{index} est l'ensemble des thèmes indexés (avec les poids les plus élevés), T_{left} est l'ensemble des autres thèmes et ip est l'adresse IP du pair. (ii) lorsqu'un pair p_i reçoit une liste de paires similaires thématiquement en réponse à une requête par le clé key_q sur la DHT, nous insérons directement la valeur du pair p_i dans l'entrée de l'index de key_q.

Division : A une clé peut correspondre plusieurs valeurs car un thème ou un ensemble de thèmes peuvent être partagés par plusieurs pairs. Lorsque les valeurs d'une clé sont trop nombreuses (selon un seuil prédéfini), nous les divisons en plusieurs listes, chaque liste correspondant à une nouvelle clé inférée à partir de la clé originelle. Pour effectuer cette division, les thèmes de plus forts poids dans l'ensemble T_{left} de chaque valeur sont sélectionnés et regroupés. Les groupes ayant une taille inférieure à un seuil seront considérés comme une autre entrée d'index avec une nouvelle clé. Cette nouvelle clé est générée en utilisant une fonction de hachage sur l'union des ensembles de T_{index} et le thème qui est utilisé pour former le groupe. Puis, nous insérons la nouvelle entrée d'index dans la DHT. Ensuite, nous ajoutons l'information sur la nouvelle clé et le super-pair responsable à la fin de l'index originel. Ceci permettra de faciliter le processus de recherche d'information que nous allons décrire en détail dans la prochaine sous-section.

Interrogation : la requête est également décrite par une représentation thématique. La fonction de hachage est la même que celle utilisée pour la DHT. Les valeurs de hachage sont calculées pour le thème ayant le poids le plus élevé puis pour les deux premiers thèmes avec les poids les plus élevés puis pour les trois premiers thèmes avec les poids les plus élevés, etc. Ces valeurs de hachage sont utilisées comme clés pour l'interrogation. La requête est alors représentée par $< key_1, key_2, ...key_k, ip_q, type >$ où $key_1, key_2, ...key_k$ sont les valeurs de hachage des thèmes avec les poids les plus élevés, ip_q est l'adresse IP du pair qui soumet la requête, et $type$ est une valeur booléenne qui indique le type de requête(0 s'il s'agit de trouver des pairs similaires à un nouveau pair et 1 s'il s'agit de trouver des pairs pertinents pour la requête).

La requête est alors envoyée dans la couche haute des super-pairs pour chercher le super-pair responsable de la clé key_1. S'il existe une entrée d'index avec la clé key_1, nous vérifions si cet index a été divisé et si la clé key_2 correspond à une des clé de la division. Nous continuons ce processus jusqu'à ce que nous trouvions une clé qui n'a pas été divisée, et nous retournons la liste des valeurs de cette clé.

5. Conclusion

En recherche d'information pair-à-pair, les profils thématiques des pairs peuvent être utilisés pour améliorer le routage des requêtes des utilisateurs. Notre étude montre qu'il est possible d'utiliser LDA pour la définition du profil thématique des pairs, mais qu'il faut être attentif à la manière de le faire. Pour cela, nous avons comparé les avan-

tages et les inconvénients de la définition d'un ensemble de thèmes de référence par rapport à un calcul des thèmes par chaque pair indépendamment. Nous avons ensuite illustré l'utilisation des profils thématiques en proposant deux types d'organisation permettant un rapprochement thématique des pairs. La première utilise un algorithme épidémique, la deuxième gère en plus un index thématique. Nous travaillons actuellement à l'évaluation des propositions d'un point de vue recherche d'information, phase où les expérimentations nécessitent beaucoup de mise au point.

Remerciements. Les auteurs remercient G. Verger et M. Servajean pour les discussions communes sur ce sujet.

Références

Bertier M., Frey D., Guerraoui R., Kermarrec A.-M., Leroy V., « The Gossple Anonymous Social Network », *Middleware*, 2010, p. 191-211.

Blei D. M., Ng A. Y., Jordan M. I., « Latent Dirichlet Allocation », *Journal of Machine Learning Research*, vol. 3, 2003, p. 993-1022.

Crespo A., Garcia-Molina H., « Semantic Overlay Networks for P2P Systems », *3rd International Workshop on Agents and Peer-to-Peer Computing (AP2PC)*, 2004, p. 1-13.

Deveaud R., Bonnefoy L., Bellot P., « Quantification et identification des concepts implicites d'une requête », *Conférence sur la Recherche d'Information et ses Applications (CORIA)*, 2013.

Draidi F., Pacitti E., Kemme B., « P2PRec : A P2P Recommendation System for Large-Scale Data Sharing », *T. Large-Scale Data- and Knowledge-Centered Systems*, vol. 3, 2011, p. 87-116.

Jelasity M., Guerraoui R., Kermarrec A.-M., van Steen M., « The Peer Sampling Service : Experimental Evaluation of Unstructured Gossip-Based Implementations », *5th International Middleware Conference*, 2004, p. 79–98.

Jelasity M., Montresor A., Babaoglu Ö., « T-Man : Gossip-based fast overlay topology construction », *Computer Networks*, vol. 53, n° 13, 2009, p. 2321-2339.

Kumar R., Vassilvitskii S., « Generalized Distances between Rankings », *Proceedings of the 19th International Conference on World Wide Web (WWW)*, 2010.

Lütjohann D. S., Shah A. H., Christen M. P., Richter F., Knese K., Liebel U., « Sciencenet - towards a global search and share engine for all scientific knowledge », *Bioinformatics*, vol. 27, n° 12, 2011, p. 1734-1735.

Stoica I., Morris R., Liben-Nowell D., Karger D. R., Kaashoek M. F., Dabek F., Balakrishnan H., « Chord : a scalable peer-to-peer lookup protocol for internet applications », *IEEE/ACM Trans. Netw.*, vol. 11, n° 1, 2003, p. 17–32.