Comparaison empirique de dix-sept méthodes de classification non-supervisée sur un corpus textuel

Martine Cadot, Alain Lelu, Michel Zitt

Version française : comparaison empirique de dix-sept méthodes de classification non-supervisée sur un corpus textuel.

BENCHMARKING SEVENTEEN CLUSTERING METHODS

Additional material for the article

Bibliometric delineation of scientific fields

to be published in

Springer Handbook of quantitative Science and Technology Research

2018 edition, Wolfgang Glänzel editor

Martine Cadot[1], Alain Lelu, Michel Zitt

FRENCH VERSION UPDATED ON 25/02/2018

---

1 - Introduction

Nous avons choisi d'examiner les principales méthodes de clustering de textes d'un corpus - qu'elles soient directes ou basées sur l'extraction de « communautés » de mots ou de documents issues de cet ensemble de textes - dans l'optique d'*utilisateur* qui est celle des chercheurs confrontés à la délimitation des domaines scientifiques à partir de bases de résumés bibliographiques.

Nous nous sommes posés quelques questions simples :

- Leurs résultats sont-ils reproductibles - par la même méthode appliquée avec des conditions initiales différentes au même corpus ? Par d'autres méthodes ?
- Traduisent-ils une structure réelle présente dans les données, pouvant tenir le rôle de "gold standard" ?
- En cas de structure déséquilibrée, c'est à dire de coexistence de *classes* (selon la terminologie ci-dessous) de grande taille et de classes de petite taille, la méthode utilisée permet-elle de détecter cette structure ?

Nous avons fait le choix de confronter la quinzaine de méthodes présentées autour du traitement d'un corpus réel, de taille raisonnable, mais surtout d'accès public, et de structure nette, incontestable, susceptible de faire l'objet de mesures numériques de distance aux structures extraites par les diverses méthodes.

[1] LORIA, Equipe MULTISPEECH,

Un point de terminologie : nous appellerons indifféremment "cluster" ou "topic" les regroupements d'éléments (documents, termes, etc.) effectués par les méthodes d'analyse des données testées, par opposition au mot "classe", qui désignera les catégories d'éléments étiquetés manuellement en amont du processus. L'apprentissage automatique ("machine learning") est la branche du traitement des données qui cherche à généraliser l'attribution de ces catégories à des documents qui n'ont pas fait l'objet de cet étiquetage humain, coûteux. Nous la désignerons de façon condensée par le mot "apprentissage", et en parlerons peu, ce processus étant peu utilisé pour le problème de la délimitation. On parlera donc de "clustering", ou" topic extraction", plutôt que de "classification" dans la suite de ce texte. A noter aussi que nous réserverons le terme de "cartographie intégrée" (embedded mapping) aux seules méthodes qui intègrent intrinsèquement ce processus, à savoir les méthodes factorielles, et la cartographie de Kohonen (Kohonen maps). Les cartes obtenues à partir des autres méthodes fournissant les topics sont issues d'un traitement secondaire plaçant les topics les uns par rapport aux autres dans deux dimensions. Leurs tableaux de sortie documents X topics peuvent faire l'objet, par exemple, d'une Analyse en Composantes principales, d'un algorithme de placement bi-dimensionnel des points d'un graphe, ou d'un échelonnement multidimensionnel (MDS) sur leurs colonnes topics.

2 - Les données de test

Notre choix s'est porté sur le corpus de dépêches Reuter's 21 578 [1][Lewis et al. 2004] dans sa version épurée "Mode Apté Split" [2] [Apté et al. 1994] . Dans un objectif de reproductibilité des résultats , et éviter des pré-traitements linguistiques et/ou statistiques toujours difficiles à spécifier à 100%, nous avons opté pour la matrice documents X mots mise directement à la disposition du public sur le site associé à [3] [Cai et al. 2005], même si le traitement lexical en a été très basique (pas de termes composés, tous les nombres sont considérés comme des mots ...). Nous avons choisi de ne garder que les mots d'occurrences totales supérieures à 15, pour limiter les exigences en espace mémoire et temps de calcul. Il en résulte une matrice documents X mots de taille 6829 X 3244.

L'intérêt et le défi posé par ces données tient au déséquilibre entre les tailles de classes attribuées par les indexeurs de Reuter's. Pour limiter la difficulté, et bien qu'il soit souvent l'usage de sélectionner les 10 premières classes, nous nous contenterons des 6 premières, où deux classes constituent 84% du corpus, et les 4 suivantes se partagent équitablement le reste. Voici les tailles, intitulés concis et aperçus du contenu de ces classes de documents (ici des dépêches) :

(3713) [earn] : opportunités d'investissement.

(2055) [acq] : fusions-acquisitions d'entreprises.

(321) [money fix] : cours des changes.

(298) [crude] : cours du pétrole brut.

(245) [trade] : commerce national et international.

(197) [interest] : taux d'intérêt bancaires.

Le deuxième intérêt de ce jeu de données est que cette classification correspond à une structure réelle, comme le suggère visuellement le tableau des cosinus entre vecteurs-documents, seuillés à 0.5 (cf. figure 1) : une grosse classe [earn] très homogène, une autre [acq] moins dense et liée à la première, 3 petites classes [money fix] [crude] et [trade] homogènes et liées à [earn], et une dernière petite classe [interest] liée à [trade]. Ce qui correspond aux principales difficultés rencontrées en pratique, à savoir les disparités de tailles et de densités entre classes.

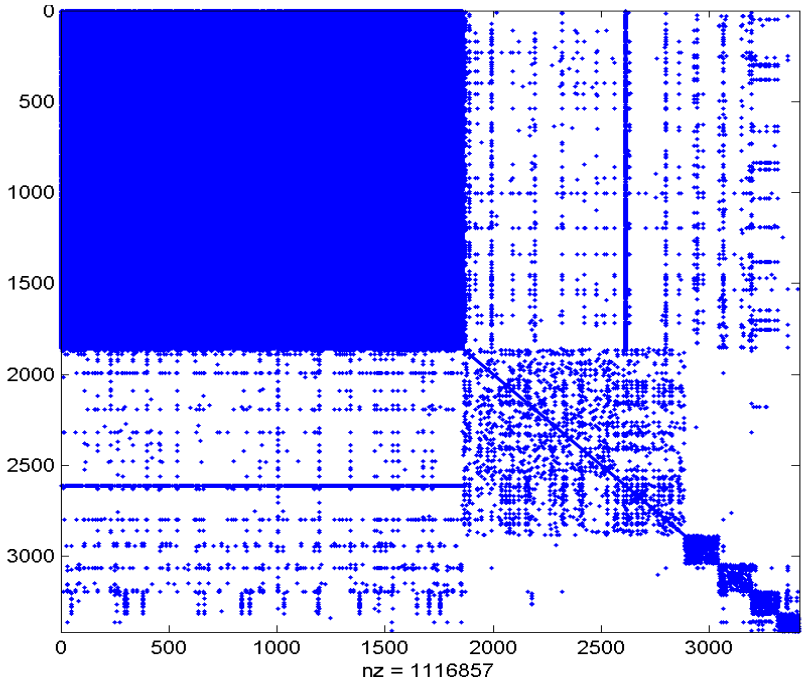

Figure 1 : *les cosinus entre vecteurs-documents Reuter's. Les points foncés représentent des cosinus supérieurs à 0.5. L'ordre des documents est celui des classes Reuter's. Pour des questions de lisibilité on n'a représenté qu'un document sur deux.*

Cette impression est confirmée numériquement par le tableau des densités[2] dans et hors de ces classes (cf. table 1 ci-dessous). On attend d'une méthode non supervisée qu'elle confirme plus ou moins cette structure.

	Cl.1	Cl.2	Cl.3	Cl.4	Cl.5	Cl.6
Cl.1 : earn	**.35**					
Cl.2 : acq	.07	**.13**				
Cl.3 : money fix	.06	.06	**.13**			
Cl.4 : crude	.04	.04	.05	**.18**		
Cl.5 : trade	.06	.05	.05	.06	**.15**	
Cl.6 : interest	.04	.06	.05	.05	**.11**	**.25**

Table 1 : densités intra- et inter-classes Reuter's Mode Apté split (6 classes)

[2] La densité au croisement de deux classes C1 et C2, d'effectifs n_1 et n_2, contenant respectivement les documents d_{1i} (i de 1 à n_1) et d_{2j} (j de 1 à n_2), est définie comme $\sum_{i,j} \cos(d_{1i},d_{2j})/(n_1.n_2)$. Le non-seuillage des cosinus pour le calcul des densités peut expliquer des différences mineures entre l'impression visuelle donnée par la figure 1 et le tableau des densités.

3 - Critères de comparaisons

Pour comparer aux classes les clusters trouvés, nous avons choisi deux indicateurs de comparaison de partitions, le Normalized Mutual Information (NMI) [4][Cover, Thomas 1991] et l'Adjusted Rand Index (ARI) [5][Rand 1971] qui ne nécessitent pas , comme d'autres, d'établir une correspondance biunivoque entre clusters et classes, ni n'en exigent un nombre strictement égal. Leur valeur est nulle quand les deux partitions sont indépendantes ou que l'une d'entre elles est la partition triviale (une seule classe). Leur valeur est 1 quand elles sont identiques. La suite nous montrera la différence de comportement de ces deux indicateurs.

4 - Aperçu des méthodes testées.

Il existe de nombreuses méthodes pour créer des clusters, nous les avons réparties en 5 catégories : méthodes hiérarchiques, factorielles, hybrides, probabilistes, et de voisinages.

4.1 - Méthodes hiérarchiques de construction des clusters

Les clusters sont des groupes d'objets similaires entre eux. Il y a de nombreuses façons traditionnelles de les construire, qui nécessitent toutes le choix d'au moins 2 outils : une mesure de similarité entre 2 objets définie à partir des caractéristiques des objets, et une méthode d'agrégation des objets en groupes (les clusters) à partir de la matrice de leurs similarités deux à deux. Parmi la grande palette disponible pour chaque type, le choix va être guidé non seulement par la nature des objets et leurs caractéristiques mais également par l'information qu'on souhaite voir traduire par les clusters.

a) Outil 1 : mesures de similarité entre 2 objets décrits par des caractéristiques binaires

Par exemple si les objets sont les textes d'un corpus et leurs caractéristiques sont la présence/absence de mots d'une liste de N mots, les indices de similarité entre 2 textes i et j sont définis par une formule dans laquelle figurent les 4 effectifs suivants :

- a, le nombre de mots simultanément présents dans les 2 textes
- b, le nombre de mots présents dans le premier texte et absents du second
- c, le nombre de mots présents dans le deuxième texte et absents du premier
- d, le nombre de mots simultanément absents des 2 textes

On peut poser a+b+c+d=N, a+b=occ(i), a+c=occ(j) où occ(k) désigne le nombre d'occurrences dans k, c'est-à-dire le nombre de mots présents dans le texte k.

Dans la table 2, on a donné les valeurs de quelques indices de similarité parmi les plus courants pour les associations de 2 (qu'on appellera 2-motifs) parmi 4 documents E, F, G et H, pour lesquels on note

la présence/absence de N=50 mots[3]. Entre parenthèses figure en rouge le numéro d'ordre obtenu en ordonnant les motifs selon la valeur décroissante de l'indice.

i	j	Occ (i)	Occ (j)	a	b	c	d	Support	MaxInc	Jaccard	Ochiai	*p*	*Affinity*	*Ochiai2*	*Simple matching*
E	F	20	10	5	15	5	25	5 (3)	0.50 (3)	0.20 (3)	0.35 (3)	1.25 (3)	0.22 (3)	0.26 (2)	30 (3)
E	G	20	30	9	11	21	9	9 (1)	0.45 (4)	0.22 (1)	0.37 (1)	0.75 (5)	-0.30 (5)	0.14 (5)	18 (5)
E	H	20	7	4	16	3	27	4 (4)	0.57 (2)	0.17 (5)	0.34 (5)	1.43 (2)	0.34 (2)	0.25 (3)	31 (2)
F	G	10	30	6	4	24	16	6 (2)	0.60 (1)	0.18 (4)	0.35 (4)	1.00 (4)	0 (4)	0.20 (4)	22 (4)
F	H	10	7	3	7	4	36	3 (5)	0.43 (5)	0.21 (2)	0.36 (2)	2.14 (1)	0.64 (1)	0.31 (1)	39 (1)
G	H	30	7	2	28	5	15	2 (6)	0.29 (6)	0.06 (6)	0.14 (6)	0.48 (6)	-0.60 (6)	0.07 (6)	17 (6)
				Rank 1 2-itemsets				EG	FG	EG	EG	FH	FH	FH	FH
				Rank 2 2-itemsets				FG	EH	FH	FH	EH	EH	EF	EH

Table 2 : valeurs de quelques indices d'association entre 2 vecteurs-documents parmi 4 (E, F, G, H).

- Support : le nombre a de cooccurrences de mots entre les 2 textes, que certains auteurs normalisent en a/N;
- MaxInc : Max(a/(a+b) ; a/(a+c)), le maximum des cooccurrences des 2 textes relativement aux occurrences de chaque texte;
- Jaccard : a/(a+b+c)
- Occhiaï : a/sqrt((a+b)*(a+c))
- Spécialisation, utilisé dans (Zitt et al. 2000) : $(p^2-1)/(p^2+1)$, où $p=(a*N)/((a+b)*(a+c))$ (i.e. rapport à l'hypothèse d'indépendance des lignes et des colonnes)
- Occhiaï2 : (a*d)/ sqrt ((a+b)*(a+c)*(b+d)(c+d))
- SimpleMatching : a+d, ou (a+d)/N

On peut constater que les ordres diffèrent selon ces 9 indices. Les 4 derniers indices (p, Affinity, Occhiaï2 et SM) s'opposent aux précédents de par l'importance qu'ils donnent à l'effectif d, qui est le nombre des mots absents simultanément dans les 2 textes. Les indices p et Specialisation prennent en compte une loi de probabilité (ici le Chi2 d'indépendance). Le support et SimpleMatching ne considèrent que des effectifs bruts, les autres indices les relativisent par un effectif variable. Et ce n'est qu'un aperçu de la liste qui ne cesse de s'accroître (près d'une centaine actuellement) de ces indices dont le but est de quantifier le lien entre deux textes d'après la présence/absence de mots. Le lecteur souhaitant un exposé plus complet et plus détaillé peut se reporter à [6].

Nous nous sommes limités ici à des caractéristiques binaires (1: présence d'un mot versus 0:absence), alors qu'on pourrait considérer des caractéristiques numériques ou "poids" (nombre de répétitions de chaque mot dans le texte, TF-IDF, BM25/Okapi, etc.). Toutefois, le foisonnement d'indices de similarités n'est pas aussi important que dans le cas binaire, il se limite la plupart du temps à des variantes du coefficient de corrélation de Bravais-Pearson (corrélation de Spearman, corrélation bisériale) ou à une transformation des dissimilarités que sont les distances classiques (city-bloc, euclidienne, etc.) en similarités.

[3] On a indiqué dans l'annexe les données correspondant aux documents E, F, G et H.

On peut même avoir une combinaison de caractéristiques de divers types. Le choix de pondérations des caractéristiques dans les formules augmente alors la variété des résultats (par exemple calculer la similarité entre 2 personnes, connaissant leur poids en kg, leur taille en cm et leur âge en années s'avère une tâche délicate !).

Dans ces cas complexes, il arrive souvent qu'on recode les caractéristiques en binaire pour trouver plus facilement un indice de similarité bien adapté au problème que l'on souhaite traiter.

b) Outil 2 : méthode d'agrégation des objets

Exemple d'un algorithme d'agrégation utilisé pour la construction d'une hiérarchie.

Pour construire les hiérarchies, on peut procéder de façon ascendante ou descendante. De façon ascendante, on part de la matrice de similarités entre les p objets pris 2 à 2, on la transforme en matrice de dissimilarités. A l'étape 1, on considère chaque objet comme un groupe, et à l'étape suivante, on fusionne les 2 groupes ayant la plus petite dissimilarité (i.e. la plus grande similarité) en un seul groupe, ce qui donne p-1 groupes. Puis on calcule les dissimilarités entre le groupe obtenu par fusion et les p-2 groupes restants grâce à une formule de « lien » combinant les dissimilarités de chaque élément du nouveau groupe avec les p-2 groupes restants. Cette étape de fusion est reproduite tant qu'on n'a pas atteint le nombre souhaité de groupes (ou un seul groupe si ce nombre n'a pas été spécifié).

La mise à jour des dissimilarités à chaque fusion peut se faire en utilisant divers « liens », la plus courante étant de prendre le Max des dissimilarités des éléments du groupe.

Par exemple, en prenant comme indice de dissimilarité le complément à 1 de l'indice d'Occhiaï2 et le lien Max, et en se limitant à 2 groupes :

	E	F	G	H
E	0	0,74	0,87	0,75
F			0,80	0,69
G				0,93
H				0

	E	G	**FH**	
E	0	0,87	**0,75**	=Max(dis(EF),dis(EH))=Max(0,74;0,75)
G		0	0,93	=Max(dis(GF),dis(GH))=Max(0,80;0,93)
FH			0	

Table 3 : création des 2 clusters avec l'algorithme d'agrégation

On obtient les 2 clusters (EFH) et (G).

c) Note sur les limites des méthodes basées sur les distances

Les méthodes de clustering, ainsi que les méthodes factorielles et hybrides, utilisent explicitement des distances, définies entre mots ou documents deux à deux. On verra plus loin que les méthodes probabilistes peuvent s'exprimer comme des décompositions matricielles, c'est à dire comme des

méthodes hybrides utilisant implicitement des distances, elles aussi. L'association globale entre deux entités efface l'information d'interaction apportée par le caractère séquentiel des suites de mots dans un document, et plus généralement par les associations n à n, dites k-motifs. Pour illustrer ce point peu connu, on poursuit ici l'exemple donné précédemment au sujet des méthodes hiérarchiques.

Les méthodes hiérarchiques s'appuient sur les ressemblances entre objets pris 2 à 2 pour les agréger, en suivant le principe simple « l'ami de mon ami est mon ami », c'est-à-dire des données « aplaties » en un tableau des dissimilarités entre les objets pris 2 à 2. Le problème est le même avec les analyses factorielles classiques, qui utilisent des matrices de variances/covariance ou de corrélation. Mais la réalité est plus complexe. Si on considère par exemple les 3 objets E, F et H qui ont été regroupés dans le tableau 3 ci-dessus, en supposant que ce soient des documents, les fortes similarités EF, EH et FH montrent que ces documents ont beaucoup de mots communs entre eux pris 2 à 2, mais on ne sait pas si les mots communs aux 3 documents sont nombreux ou non. Toutefois, le groupe EFH est créé à partir des 2 groupes E et FH dans la section précédente en postulant que la dissimilarité entre E, F et H ne peut être supérieure à celle entre E et F et celle entre E et H, cette distance « ultramétrique » correspondant en termes mathématiques à une géométrie dans laquelle tous les triangles sont isocèles. Cette hypothèse réductrice permet d'inférer une relation ternaire, celle-ci ne pouvant en aucune manière être reconstituée à partir de la matrice des relations de similarité binaires. Pour illustrer ce pointt, on a donné dans l'annexe deux tableaux documents X mots bruts, c'est à dire deux possibilités de distributions des 50 mots dans les 4 documents E, F, G, H produisant les mêmes indices de la table 2 mais dont les supports des 3-motifs EFG et EFH diffèrent.

Et bien sûr des relations d'ordre supérieur (quaternaires et au-delà) sont aussi construites ainsi. Les autres liens permettant le recalcul des dissimilarités donneront des clusters qui peuvent différer, mais aucun ne remet véritablement en cause le fait que « l'ami de mon ami » soit « mon ami ».

Avec l'explosion des capacités de calcul et de stockage des ordinateurs, sont arrivées des méthodes puissantes, permettant de brasser à chaque étape des listes de documents avec leurs mots sans se limiter à leurs cooccurrences deux à deux. Parmi celles-ci, citons la recherche de motifs fréquents en Data Mining. Son principe est le suivant : pour calculer l'indice du 2-motif EF, on crée la liste des mots communs à E et F, et pour calculer l'indice du 3-motif EFH, on utilise non pas l'indice du 2-motif EF, mais sa liste de mots qu'on confronte à celle de H. Ainsi si EF a un indice fort, ainsi que EH, l'indice de EFH sera fort ou faible selon que E, F et H ont une liste de mots communs plus ou moins importante. Avec ces algorithmes la taille de stockage ou le temps de recalcul des listes de mots des k-motifs est énorme, et est fonction exponentielle du nombre de mots et/ou documents. Pour optimiser les algorithmes, on peut appliquer des seuils sur les valeurs des indices de l'étape k-1 afin de limiter le nombre de k-motifs extraits, la façon la plus courante de faire étant de supprimer les listes de mots communs trop petites. Des méthodes plus sophistiquées statistiquement sont également possibles [Cadot, Lelu 2012]..

Pour donner du « relief » et introduire partiellement l'interaction dans les méthodes à base de distances, il est possible de coder comme de nouvelles variables tout ou partie des n-grammes de mots, pour n petit, ou plus généralement les n-motifs. Une solution « naturelle » et moins lourde est de prendre en compte tout ou partie des expressions composées, qui ne sont que des n-grammes de mots sélectionnés par l'usage.

d) Les méthodes hiérarchiques testées

Nous avons testé trois méthodes correspondant à trois types d'agrégation différents, généralement considérés comme les plus satisfaisants :

- Group average link : le lien moyen entre deux clusters consiste à calculer la moyenne des distances entre les individus de chaque cluster
- Ward D2 : la distance de Ward vise à maximiser l'inertie inter-classe, via

dist(C1,C2) = (n1*n2/(n1+n2)) dist(G1,G2) où n1 et n2 sont les effectifs des clusters C1 et C2, G1 et G2 leurs centres de gravité respectifs.

- Mc Quitty : à chaque examen de la matrice des distances inter-clusters, on fusionne les paires de clusters en situation de voisinage réciproque (C1 est le plus proche voisin de C2, *et* C2 est le plus proche voisin de C1).

4.2 - Les méthodes factorielles

Les méthodes factorielles aujourd'hui classiques (PCA, AFC, LSA) sont basées sur la décomposition d'une matrice de r lignes (documents) et c colonnes (termes) en trois matrices plus petites, de taille respectivement (r,k) (k,k)et (c,k), où k est le nombre de facteurs extraits, dite décomposition aux valeurs singulières (SVD), opération basique de l'algèbre linéaire :

$$\mathbf{X \sim U \Delta V'}$$

Δ est une matrice diagonale, dite matrice des valeurs propres, et les matrices **U** et **V** ("vecteurs propres") ont leurs colonnes orthogonales et de norme euclidienne 1[4]. **U** et **V** représentent la projection des documents (resp. termes) sur les k axes factoriels, ce qui donne lieu à directement à une visualisation de ces éléments dans des cartes à 2 dimensions, généralement les 2 premiers facteurs. Alors que le monde anglo-saxon utilise souvent une variante "dépliage non-linéaire" de ce type de cartes, à savoir le Multidimensional Scaling (MDS[5]) , en Europe latine et aux Pays-Bas l'Analyse des Correspondances (AFC) [Benzécri 1973] continue à susciter intérêt théorique[6] et pratique depuis un demi-siècle. Mais ces méthodes conviennent pour des données de taille petite ou moyenne, et deux dimensions sont généralement insuffisantes pour exprimer la richesse des "Big Data". C'est pour cela que le l'Analyse Sémantique Latente (LSA, ou LSI)[7], ou application directe de la SVD à de grandes matrices textes X mots, rompt avec toute volonté de visualisation ou

[4] D'autres méthodes factorielles anciennes, principalement utilisées par les psychologues, aboutissant pour certaines à des facteurs obliques, ont été quelque peu oubliées à notre époque de "big data". Pour sa part, l'Analyse en Composantes Principales (PCA) crée des facteurs orthogonaux et exige une matrice de données aux variables centrées-réduites, ce qui la rend impropre à traiter de grandes masses de données. Les algorithmes plus récents tirent parti du caractère "creux" (*sparse*) de ces données, pour réduire à la fois les exigences en espace-mémoire et en calculs.

[5] Basée sur un principe différent : minimiser le "stress", qui mesure la différence entre les distances "vraies" dans l'espace multidimensionnel et les distances dans une représentation 2D.

[6] Le tableau transformé dont l'AFC effectues la SVD a des liens avec le "Laplacien" des graphes dont nous parlerons plus bas à propos de du clustering spectral [7bis][Von Luxburg 2007].

interprétation individuelle des facteurs, et se contente d'offrir un espace de dimensions "réduites" (généralement quelques centaines de facteurs pour des tableaux ayant pour plus petite dimension quelques milliers d'éléments) dans lequel peuvent se calculer des distances plus pertinentes que dans l'espace d'origine. En particulier des documents sans mots communs mais d'un même champ sémantique y sont identifiés comme proches.

Un avantage incontestable des méthodes factorielles est d'être déterministes, c'est à dire d'obéir au principe "un jeu de données, une méthode, un seul résultat".

L'Analyse en Composantes Indépendantes (ICA) [8][Hérault, Ans 1984] est très utilisée en traitement du signal, où elle résout le problème dit de "la cocktail party" (démêler n conversations à partir de n micros répartis dans la salle). Elle impose aux composantes dégagées beaucoup plus que la non-corrélation (ou orthogonalité) requise par les méthodes factorielles habituelles. Elle commence par transformer l'espace des données en l'espace des premiers vecteurs propres, défini par la matrice documents X topics dont toutes les colonnes sont de variance un. Il faut spécifier au départ le nombre de dimensions de cet espace "sphéré", ainsi que le nombre de composantes indépendantes que l'on désire obtenir. Sa variante FastICA [9][Hyvärinen 1999] est susceptible de traiter des matrices documents-mots, mais reste peu utilisée dans ce domaine.

4.3 - Les méthodes hybrides analyses factorielles et clustering

La **Non-negative Matrix Factorization (NMF)**[10][Lee, Seung 1999] part du principe d'éviter les projections négatives inhérentes aux analyses factorielles afin de caractériser les descripteurs, comme les objets décrits, par des indicateurs de "saillance" positifs ou nuls. En analyse d'image par exemple, des coefficients négatifs n'ont aucun sens. L'usage en NMF est de réaliser une décomposition en deux matrices seulement **X=HV'** où seules les colonnes de la matrice **V** sont normalisées, la matrice **H** étant l'équivalent du produit **U* Δ** vu plus haut. Le contrainte de non-négativité de **H** et **V** est respectée grâce à un algorithme multiplicatif de mise à jour de **V** et **H**, à la différence des méthodes factorielles où les corrections de mises à jour sont additives. Les résultats montrent un "clustering effect" : les valeurs de **V** et **H** tendent à prendre des valeurs soit proches de zéro, soit nettement plus élevées. De fait, les axes définis par les colonnes de ces matrices pointent vers les zones de forte densité des données, et on peut considérer NMF comme une méthode de clustering produisant des clusters flous et recouvrants. Les axes sont le plus souvent obliques, formant des angles inférieurs ou égaux à 90°.

Une méthode plus ancienne et plus proche d'un clustering proprement dit a été dénommée **K-Moyennes Axiales (AKM)**[11][Lelu 1994] Elle aboutit à un résultat voisin, à savoir des coefficients de "typicité" positifs et inférieurs à 1 pour les documents d'un cluster (défini ici explicitement), nuls ou faiblement positifs pour les documents qui n'en font pas partie, ce qui rend aussi possible une interprétation en termes de clusters flous et recouvrants. Chaque axe de cluster est l'axe principal, au sens de l'Analyse Factorielle Sphérique [12], du sous-nuage de points de ce cluster à la surface de la sphère unité. Comme tous les algorithmes de K-means, cette méthode est rapide, peu gourmande en mémoire et adaptée aux données de grande taille.

Mais ces deux méthodes comportent, comme beaucoup d'autres, un inconvénient de taille : leur sensibilité aux valeurs d'initialisation. Leurs algorithmes ne leur permettent de converger que vers des optima locaux. Nous reviendrons plus bas d'un point de vue expérimental sur ce problème.

*Remarques sur les **Self Organizing Maps (SOM)** [Kohonen 1998]*

Le modèle d'inspiration neuronale « Carte auto-organisatrice » (SOM, Self-Organizing Map) demande à spécifier, en plus du nombre de topics et d'une graine d'initialisation au hasard, une disposition géométrique des topics uns par rapport aux autres, le plus souvent sous la forme d'une grille carrée ou rectangulaire. L'avantage est d'obtenir directement une visualisation 2D pertinente des topics les uns par rapport aux autres. L'inconvénient est que, en fonction de la graine d'initialisation, des effets de bord génants peuvent se produire, si un topic central se retrouve situé au périmètre de la grille. Des graines d'initialisation différentes peuvent créer des cartes très différentes, difficilement reconnaissables entre elles.

4.4 - Les modèles probabilistes

a) Probabilistic Latent Semantic Analysis (pLSA)

Pour répondre au caractère aveugle du LSA et à l'impossibilité d'interpréter individuellement les axes extraits, tout en offrant un fondement statistique à ce type de méthodes, [14][Hofman 1999] a créé le pLSA, basé sur la décomposition :

$P(d,w) = \sum_z P(z)\, P(d/z)\, P(w/z)$

où P(d,w) est la probabilité conjointe du document d et du mot w qui modélise le nombre d'occurrence de w dans d divisé par le total des occurrences de mot dans le corpus - une normalisation particulière de la matrice des données -, P(d/z) est la probabilité conditionnelle du document d sachant la valeur de la variable catégorielle z (la somme pour tous les documents de ces probabilités est égale à 1) ; même principe pour le mot w. P(z) est la probabilité de chaque catégorie (ou topic) z. Le nombre Z de catégories est fixé, D et W sont respectivement les nombres de documents et de mots.

Cette décomposition s'exprime matriciellement selon le même schéma **P**=**U D V'** vu plus haut, mais avec des colonnes non-orthogonales pour **U** comme pour **V**. Chaque colonne de ces matrices s'interprète comme un "topic", à la manière de NMF.

Hoffman modélise P(d/z) et P(w/z) comme des lois multinomiales, de nombre de paramètres[7] D.Z pour la première, W.Z pour la deuxième - des nombres considérables, nous reviendrons sur ce fait dans la partie Influence de l'initialisation, plus bas. Partant d'une initialisation des matrices de P(z),

[7] Les paramètres de ces lois multinomiales sont les probabilités d'apparition de chacune des D catégories pour chacune des Z sous-populations, par exemple de mourir d'une certaine cause quand on est un homme ou qu'on est une femme. Dans ce cas, il y a D.Z paramètres, dont certains contrastés (cancers du sein, de la prostate...), et d'autres non. Leur somme fait 1 (hélas !) pour chacune des sous-populations.

P(d/z), P(w/z) au hasard, il utilise l'algorithme Expectation Maximization (EM) [15] [Dempster et al. 1977] pour converger, via la mise à jour de la "pile" de matrices intermédiaires P(z/d,w), vers un optimum local de la fonction objectif optimisée par cet algorithme, à savoir la log-vraisemblance [16] [Fisher 1912] des données par rapport aux lois multinomiales obtenues à chaque pas d'itération. *A contrario* les méthodes à base d'algèbre linéaire comme la SVD optimisent un autre critère, la somme des carrés des écarts entre données reconstituées et données d'origine, basé, lui, sur le concept de variance.

b) Latent Dirichlet Allocation (LDA)

La LDA [17][Blei et al. 2003] a été conçue à partir d'une critique faite au modèle statistique du pLSA : celui-ci est basé sur un mélange de lois multinomiales, dont le nombre de paramètres croît linéairement avec le nombre de documents, qui peut ainsi croître indéfiniment, contrairement au nombre de mots ; on parle alors de "surapprentissage"[8] (overfitting). On préfèrerait modéliser les documents par une loi "générative", avec peu de paramètres. C'est pourquoi LDA obtient une réduction drastique du nombre de paramètres en modélisant les Z répartitions des documents conditionnellement aux topics par une loi de Dirichlet - approximation, à Z paramètres, d'un ensemble de Z distributions multinomiales. Restent les W.Z paramètres des mots - il est encore difficile de modéliser des ensembles de lois Zipfiennes...

Un aspect intéressant de la LDA est qu'elle permet de concevoir de nombreuses variantes et raffinements : prise en compte des N-grammes de mots, des subdivisions des textes, des aspects dynamiques, …

c) Une méthode de clustering flou : Fuzzy C-Means (FCM)

C'est une méthode ancienne [18][Dunn 1973] qui produit pour chaque document une probabilité d'appartenance à chaque topic. La somme de ces probabilités est égale à 1, alors que pour pLSA et LDA c'est la somme pour chaque topic, et pour NMF c'est la somme des carrés des coefficients de typicité pour chaque topic également qui doivent être égales à un. Les KMA pour leur part produisent pour chaque document des indicateurs de centralité dans le topic, à valeur entre 0 et 1 - ceci aussi pour les documents hors du topic d'appartenance, ce qui autorise également une interprétation en termes de clusters flous et recouvrants.

Pour les FCM il faut préciser au départ le nombre de topics désirés. Elles optimisent un critère basé sur des sommes de ces valeurs de probabilité à la puissance, avec α supérieur à 1.

Nous examinerons empiriquement plus bas si l'oubli dans lequel sont tombés les FCM était mérité ou pas…

[8] On verra plus bas par quoi se traduit concrètement la notion de surapprentissage, plus courante dans le contexte de l'apprentissage supervisé.

4.5 - Méthodes de voisinages

a) *Une méthode de clustering densitaire : DBSCAN [19][Ester et al. 1996]*

Un rayon R définit le voisinage d'un point (ici un document), et le paramètre *minpts* fixe le nombre minimum de points dans son voisinage pour que ce point soit considéré comme l'amorce d'un cluster (ou son extension). Sinon, il est considéré comme du bruit. L'algorithme fonctionne par extension progressive des clusters trouvés. Son implémentation "naïve" est très lente, mais peut être fortement accélérée par diverses structures informatiques des données.

Une des caractéristiques de cette méthode est de pouvoir détecter des clusters de forme quelconque, non nécessairement linéairement séparables, ainsi que des points isolés ("outliers") et des points-frontière entre 2 clusters. Ceci constitue un avantage dans le cadre de certaines applications, mais pas dans celui de la bibliométrie où il est plus opérationnel et lisible par des experts de délimiter des agrégats homogènes que des continuums entre deux ou plusieurs pôles. A noter un avantage, qui est de ne pas obliger à spécifier de nombre de clusters au départ. Et son avantage principal est d'être déterministe : un jeu de données, un jeu de paramètres, un seul résultat. Mais ses deux paramètres sont délicats à ajuster, et ne peuvent pas permettre d'extraire des clusters de densités différentes.

b) Des méthodes de clustering de graphes

Louvain

C'est une méthode [20][Blondel et al. 2008] d'optimisation d'un indicateur global de modularité du graphe, par comparaison au graphe de même répartition globale des liens, mais dont les valeurs des arêtes sont calculées sous l'hypothèse d'indépendance des noeuds (« null model »). Il a le mérite de ne pas comporter de paramètre à ajuster, ni de nombre de topics à ajuster,mais il est établi depuis [22][Lancichinetti, Fortunato 2011] que son critère de modularité porte en lui le problème de la « resolution limit » : plus le graphe est étendu, plus le nombre de clusters tend à décroître – ce qui est peu approprié dans le cadre des Big Data en général, de la bibliométrie en particulier.

Plusieurs algorithmes inspirés de Louvain tentent de passer outre à cette limite en introduisant un paramètre de résolution, par exemple **Smart Local Moving Algorithm** [21][van Eck et al. 2010]. Mais [22][Lancichinetti, Fortunato 2011] ont montré que ces approches comportaient elles aussi des difficultés de prise en compte de clusters de tailles et densités différentes.

Affinity Propagation

Cette méthode [Dueck, Frey 2007] fonctionne par passation de « messages » et mises à jour successives de deux matrices : celle dite de « responsabilité » quantifie l'aptitude de chaque item à

servir de type exemplaire, de « modèle », à chacun des autres ; celle dite de « disponibilité » [availability] quantifie la capacité de chaque item à prendre pour type exemplaire chaque autre, compte tenu de l'ensemble des préférences pour celui-ci. Elle ne demande pas de spécifier un nombre de clusters, mais comporte un paramètre de résolution, dit « préférence ».

Spectral Clustering

Cette méthode [24] [Meila, Shi 2000] se fonde sur l'utilisation des K-means dans un espace transformé, celui des K premiers vecteurs propres non triviaux de la matrice dite Laplacienne du graphe, celui que produit par ailleurs l'AFC [25] [Lelu, Cadot 2010]. Dans cet espace sémantique latent de dimensions réduites peuvent se dégager des clusters de forme quelconque – ce qui n'est pas nécessairement un avantage pour les applications bibliométriques. En plus du nombre K de clusters à spécifier, elle est soumise à l'aléa d'initialisation propre aux K-means. Pour une réponse à la question : quel est le nombre K* de clusters intrinsèquement présents dans le graphe (significatifs au sens statistique), et non constitués par du bruit, [26][Lelu, Cadot 2013] fournissent une réponse basée sur des simulations de Monte Carlo.

InfoMap

Cette méthode [27][Rosvall, Bergstrom, 2007] a des fondements solides en théorie de l'information. Elle quantifie le « paysage de densité » d'un graphe directionnel (ou pas) en affectant des codes binaires aux noeuds, d'autant plus courts qu'ils sont traversés souvent par des « marcheurs aléatoires » parcourant le graphe. La description du graphe se trouve ainsi comprimée, et le graphe se trouve partitionné en modules flous, ou pas : chaque module se voit attribuer de la même manière un code d'autant plus court qu'il est fréquenté par les marcheurs aléatoires. La description globale la plus courte du graphe fournit le nombre et la composition des clusters dégagés, avec le degré de centralité de chaque nœud dans son cluster. La méthode ne nécessite aucun paramétrage ni spécification d'un nombre de clusters désiré.

Density Peaks

L'originalité et l'intérêt de cette méthode [28][Rodriguez, Laio 2014] résident principalement, en plus de son caractère déterministe, dans la possibilité offerte à l'utilisateur de choisir les amorces de cluster dans un graphique dit "decision graph" donnant pour chaque entité sa densité et sa distance minimum à une entité plus dense, résolvant ainsi les problèmes que posent les paysages de densité "rugueux" pour la détection des pics de densité. Plusieurs méthodes de calcul de densité sont possibles, et un paramètre de résolution dit "neighbor rate" est nécessaire. Elle opère à partir de la matrice des distances inter-entités.

5 - Comparaisons expérimentales et conclusions : influence décisive de l'initialisation et du paramétrage

5.1 - Méthodes déterministes

a) AFC

Si on cherche peu de *topics*, disons 6 au plus, une façon simple de procéder est de réaliser une analyse des correspondances de la matrice documents X mots (c'est un matrice de comptages) qu'on limitera aux K premiers vecteurs propres non triviaux, puis d'en déduire des clusters en cherchant , pour chaque document, l'axe sur lequel sa projection est de plus grand module et de même signe que la majorité[9]. Avantage : un seul passage, reproductible, interprétation aisée via les projections des mots, possibilité de traiter des corpus importants tant que la taille du vocabulaire ne dépasse pas quelques dizaines de milliers de mots. Ainsi 6 facteurs ont été extraits de notre ensemble de test en moins de 1 seconde avec une unité centrale Pentium 6core i7, 3.33 GHz, avec les résultats honorables ARI=.39 et NMI=.41. Nous avons vérifié que cette valeur était maximale pour K=6.

b) *Méthodes hiérarchiques*

A notre surprise, ces méthodes se sont révélées parmi les meilleures de notre benchmark, et même la meilleure en ce qui concerne Group average link : celle-ci surpasse largement ses concurrentes en termes d'ARI (.63, contre .46 pour la meilleure des méthodes non-hiérarchiques, LDA), moins nettement en termes de NMI (.52 contre .51 pour KMA). McQuitty et Ward D2 obtiennent respectivement la 2e place en termes d'ARI et la 3e en termes de NMI.

c) DBSCAN

Le choix des deux paramètres "rayon du voisinage" et "nombre minimal de voisins" nous a conduit à de nombreux essais. Beaucoup parvenaient à une structure "Un cluster dominant plus une poussière de clusters quasi-individuels", ou à rejeter comme bruit la majorité des points, à part quelques clusters de petite taille. Le moins mauvais compromis a été trouvé pour R=1.1 et minpts=5, où 63% du corpus est réparti sur 3 clusters de tailles déséquilibrées. La valeur ARI=.46 en résultant n'a pas de signification dès lors que 37% du corpus ont été versés dans le pot commun des "rejetés". L'incapacité de DBSCAN à prendre en compte des clusters de taille et densités différentes nous paraît donc rédhibitoire.

[9] Les projections de plus grand module se trouvent généralement d'un même côté de l'axe, soit positif, soit négatif. La méthode Alceste [Reinert 1985] est une méthode plus rigoureuse d'extraction d'un nombre limité de clusters dans le même espace des K premiers facteurs d'AFC : elle réalise une partition descendante hiérarchique de ces axes factoriels.

d) Louvain

Cette méthode de partitionnement de graphes opère sur une matrice d'adjacence entre documents, qu'il est possible de définir de nombreuses façons, ici par les cosinus inter-documents. Le code dont nous avons disposé était très lent (plusieurs heures pour chaque matrice testée). La matrice des cosinus entre vecteurs-documents a engendré 3 clusters et un ARI médiocre de .22. La seuiller à .5 a produit 4 clusters de tailles comparables, mais un ARI de .15. Seul un codage binaire des cosinus seuillés à .65 a permis de retrouver le niveau ARI=.22. Ces résultats confirment l'inconvénient souvent pointé [27] du critère "modularity" optimisé ici, à savoir sa "limite de résolution" : plus le corpus est important, moindre est sa sensibilité à l'existence de clusters, d'où le faible nombre de clusters extraits. Si l'on ne vise qu'un nombre limité de topics, il semble qu'on aurait des résultats de meilleure qualité avec l'AFC, d'autant que l'efficacité informatique de cette dernière est bien supérieure (dans le cadre des logiciels dont nous disposons), et que son facteur limitant est le nombre de mots, et non de documents comme Louvain.

e) InfoMap

Bien que cette méthode introduise l'aléa au plus profond et au plus bas niveau de son algorithme (génération d'un grand nombre de marches aléatoires dans un graphe), on peut la qualifier de stable et reproductible, au sens où ses meilleurs résultats – mesurés par la longueur de description du graphe découpé en modules – sur une dizaine de passages semblent très proches. Pour ce qui concerne notre corpus de test, elle aboutit à 5 clusters de tailles déséquilibrées. La valeur produite de l'indicateur ARI (.2420) n'est que dans la fourchette moyenne de l'ensemble des méthodes. Mais la valeur de l'indicateur NMI (.4359) est dans la fourchette haute, résultat d'autant plus remarquable qu'elle fait partie, seule avec Louvain, des méthodes sans paramètre à ajuster ni nombre de clusters à préciser - et on a constaté que le NMI est l'indicateur le plus proche de nos critères de jugement, car il prend en compte la justesse de reconstitution des petites classes autant que des grandes. Ici cette méthode individualise bien la petite classe 5, et coupe même en deux la petite classe 6.

f) Density Peaks

Cette méthode s'est révélée peu sensible à l'existence de clusters de tailles et densités différentes, et il a fallu chercher des amorces de clusters peu évidentes sur le "decision graph", ainsi qu'un "taux de voisinage" très faible (0.08%) pour obtenir des valeurs décevantes d'ARI (.26) et NMI (.40) .

5.2 - Méthodes à optima locaux :

Un point commun des méthodes NMF, KMA, pLSA, LDA, ICA, Spectral Clustering est de demander en entrée un nombre K de topics désirés, et de fournir en sortie K vecteurs (valeurs des documents pour chaque topic). L'AFC le fait aussi, dans un cadre d'optimisation globale, et ses vecteurs sont orthogonaux, ce qui n'est pas généralement le cas pour les autres méthodes, qu'on pourrait

rassembler sous le vocable de "méthodes à facteurs obliques". Bien que des interprétations en termes de clusters flous et recouvrants soient toujours possibles à partir de cette structure générale, il est en pratique plus facile - et plus simple à commenter - d'en tirer des clusters stricts par simple repérage du maximum des valeurs pour chaque document.

Chaque méthode possède en propre sa fonction objectif, que les algorithmes ne savent actuellement faire converger que vers un optimum *local*. D'où l'importance des procédures d'initialisation pour la qualité du résultat final. Celles-ci incorporent toujours un tirage au hasard de valeurs initiales engendrant directement ou pas la matrice de sortie documents X topics. Sauf à spécifier les lignes de code utilisées et la graine d'initialisation, les résultats ne sont pas reproductibles. Il faut répéter les passages avec des graines d'initialisation différentes pour s'approcher des valeurs optimales de la fonction objectif, correspondant à des partitions qui peuvent être très différentes entre elles si le nombre de clusters désiré est élevé (quelques dizaines au moins). Le plus souvent, on parvient en une ou deux dizaines de répétitions à des valeurs optimales qui plafonnent à quelques centièmes ou millièmes près de la meilleure valeur obtenue de la fonction objectif sur quelques milliers de répétitions.

Un problème supplémentaire majeur est qu'un optimum de fonction objectif ne correspond pas nécessairement à une partition satisfaisante selon nos critères d'utilisateurs (ou selon des critères qui s'en révèleraient proches) On pourrait citer à nouveau le cas de NMI, qui nous paraît le plus proche de nos critères "naturels" de jugement. Nous avons exploré ce problème avec les méthodes LDA et KMA. C'est la LDA qui a fourni la plus forte valeur des critères ARI et NMI[10] qui mesurent la similarité entre les partitions obtenues et la partition de référence, sur une ou deux dizaines de passages. Le tableau 3 ci-après montre que l'ARI (ou NMI) maximum est loin de coïncider avec celui obtenu quand la fonction objectif est à sa meilleure valeur[11]. Les chances de tomber sur une initialisation menant à un optimum de partition humainement satisfaisant sont donc infimes.

Affinity Propagation et Smart Local Moving Algorithm pour leur part peuvent également être classées parmi les méthodes à fixation du nombre K de clusters, car elles comportent un ou deux paramètres de résolution dont l'ajustement aboutit au même résultat. Elles ne sont pas non plus déterministes.

Par rapport aux autres méthodes, c'est LDA qui s'est montrée la moins critiquable de ce point de vue. Elle aboutit, à son minimum de *perplexity*, à une partition en 3 clusters, en pratique, dont deux correspondent plus ou moins aux classes 1 et 2, et le 3ème aux 4 petites classes restantes, mélangées à des éléments des classes 1 et 2 - ce qui n'est pas vraiment le résultat recherché. Les autres méthodes ne font pas mieux : KMA a tendance à créer des clusters encore plus équilibrés, et est la deuxième moins mauvaise du lot. Après ces deux méthodes donnant des ARI au dessus de .4, deux autres méthodes, NMF et ICA, produisent des ARI autour de .3. Enfin pLSA ferme la marche avec un ARI de .13.

[10] Après avoir confronté toutes les méthodes présentées ici à la partition de référence en classes Reuter's au moyen du critère ARI, nous avons constaté que le critère NMI, qui varie à peu près dans le même sens, rendait mieux compte des similarités entre clusters et petites classes.

[11] Selon les méthodes, on recherche un maximum de la fonction objectif (pLSA, ICA, KMA) ou un minimum (NMF, LDA).

5.3 - Initialisations et paramétrages volontairement biaisés :

Pour disposer d'éléments de comparaison de ces performances, il est intéressant de savoir quel niveau maximum d' ARI peut être obtenu[12] sur une ou deux dizaines de passages qui, selon les méthodes, peuvent être initiés au hasard, ou faire l'objet d'ajustement de leurs paramètres: ici aussi Group average link vient nettement en tête avec des valeurs d'ARI et NMI (resp. .71 et .64) remarquables, optimales pour une coupure à 10 clusters, qui reconstitue bien les petites classes ; Smart Local Moving Algorithm se détache aussi avec des valeurs d'ARI et NMI (resp. .6019 et .5484) parmi les meilleures de l'ensemble du test – et des valeurs médiocres si on n'ajuste pas ses paramètres... Viennent ensuite LDA (.55), puis ICA(.54) dans l'espace des 10 premiers vecteurs propres, KMA (.48), et Spectral Clustering (.41)) dans l'espace des 7 premiers vecteurs propres. Enfin NMF, Fuzzy C-means et Affinity Propagation donnent des valeurs inférieures à .4.

Tous ces éléments montrent 1) que hors Group average link, la plupart des fonctions objectif des méthodes testées ne sont pas adaptées à la structure de classes proposée, pourtant peu originale (2 grosses classes, l'une dense et l'autre pas, 4 petites classes, dont deux bien individualisées), 2) que seules InfoMap et Louvain s'approchent, sans paramètre à ajuster ni nombre de clusters à spécifier, de la structure « vraie » des données, 3) qu'il peut exister des fonctions objectif plus adéquates, comme le montre Smart Local Moving Algorithm sous un paramétrage particulier, et donc une large place pour de nouvelles recherches. Elles n'ont d'ailleurs pas cessé depuis plus d'un demi-siècle, et semblent se multiplier et se complexifier (cf. le développement du Deep Learning) plutôt que s'étioler.

5.4 - Essai de recommandations

L'expérience présentée ici des principales méthodes du continuum "mapping-clustering-community detection", et limitée au test d'un jeu de données - mais ce jeu présente la plupart des difficultés rencontrées dans les applications pratiques - montre que l'état de l'art actuel n'est pas satisfaisant, qu'il a plutôt régressé après l'apparition des méthodes hiérarchiques dans les années 1960, et qu'il serait imprudent de tirer des conclusions péremptoires au sujet d'une délimitation bibliométrique de domaines scientifiques issue d'une de ces méthodes, surtout quand elle doit être assortie d'une initialisation et d'un paramétrage particuliers. Seules InfoMap et Louvain ne demandent aucun de ces éléments, et Infomap est susceptible de fournir, sans exigence de connaissances expertes préalables, une partition en clusters plus proche de notre appréhension de cette notion difficile à définir, en rapport avec des fluctuations significatives de la densité d'un nuage de points. En attendant de nouvelles percées méthodologiques, une partition InfoMap peut fournir une base que d'autres méthodes pourvues de paramètres, comme Group average link, Fast Local Moving Algorithm, ou LDA, jointes à la prise en compte de connaissances expertes sur le domaine, peuvent corriger et affiner.

[12] Cas qui ne se produit jamais, par définition, dans un cadre de non-supervision.

5.5 - Perspectives

Nous sommes conscients que le travail présenté ici a été réalisé dans d'inévitables contraintes de temps et de moyens. Nous serons heureux qu'il soit poursuivi, par d'autres et par nous-mêmes :

- Nous avons tenu à présenter des résultats obtenus à partir de données publiquement accessibles, résultats que chacun peut vérifier ou contester. Celles-ci sont issues d'un seul mode d'indexation du corpus Reuters 21578. Il sera intéressant d'examiner l'influence d'autres modes d'indexation, plus basiques, ou au contraire plus élaborés du point de vue linguistique – prenant en compte par exemple les termes composés, ou la catégorie grammaticale des mots.
- Nous avons utilisé des versions publiques des algorithmes choisis, principalement des codes Matlab, Octave, Scilab, ou des exécutables Windows publiés. Il sera intéressant de vérifier si d'autres implantations aboutissent aux mêmes résultats.
- D'innombrables variantes des méthodes présentées ici n'ont pas été mentionnées : en tester les plus prometteuses permettra d'enrichir encore le débat.
- Nous avons délibérément pris le parti de mesurer la qualité d'algorithmes non-supervisés à l'aune d'un corpus destiné à tester des méthodes d'apprentissage supervisé : nous estimons que cet éclairage est nécessaire, sans être suffisant pour la tâche de délimitation des champs scientifiques, prise dans l'injonction paradoxale de détecter des émergences partiellement ou totalement cachées aux yeux des acteurs, tout en confirmant ou corrigeant à la marge l'appréhension que ces acteurs et institutions scientifiques ont de leur propre place.

5.6 - Tableau des principaux résultats

Pour l'indicateur ARI (resp. NMI) la colonne de gauche réunit les performances intrinsèques, sans connaissance de la partition de référence visée, celle de droite, l'inverse.

Table 4 des principaux résultats

Passage avec :		Fonction objectif	ARI	NMI
AFC	Pour K=6	Déterministe	.3934	.4082
	meilleur ARI (pour K=5)		.4052	.4072
CAH / Group average link	coupure pour 6 clusters	Déterministe	**.6292**	**.5207**
	coupure pour 10 clusters		.7097	.6430
CAH / Mc Quitty	coupure pour 6 clusters	Déterministe	**.4979**	.4639
	coupure pour 13 clusters		.5719	.5511
CAH / Ward D2	coupure pour 6 clusters	Déterministe	.3176	**.4759**
	coupure pour 12 clusters		.2371	.5158
DBSCAN	ARImax,pour 6 clusters, *R*=1.1, *minpts*=5	Déterministe	.4601	
NMF	meilleure fonction objectif (10 pass.) pour K=6	.7743	.3095	
	meilleur ARI	.7744	.3863	.4480
AKM (=KMA)	meilleure fonction objectif (20 pass.) pour K=6	.2614	.4349	**.5079**
	meilleur ARI	.2558	.4795	
	initialisation "ex-post"	.2573	[.8265]	[.6410]
pLSA	initialisation au hasard pour K=6	-22 173	.1344	.0673
	initialisation "ex-post"	-14 690	[1.000]	[1.000]
	initialisation autour de "ex-post"	-14 384	[.6768]	[.6357]
LDA	meilleure fonct. obj. (=min, 20 pass.) pour K=6	503.80	**.4625**	.4052
	meilleur ARI	523.50	*.5460*	.5345
	initialisation "ex-post"	480.63	[.6184]	[.7333]
ICA (=ACI)	meilleure fonct. obj. (pour 7 vecteurs propres)		.2818	
	meilleur ARI (pour 10 vecteurs propres)		.5390	
Fuzzy C-Means (FCM)	meilleur ARI (*power*=1.082) pour K=6	4300.90	.3337	
Louvain	COS (remplissage : 99.66%)		.2794	.4212
	COS seuillé à 0.1 (remplissage : 43.30%)		.2750	.4230
	COS seuillé à 0.5 (remplissage : 9.58%)		.1506	

Spectral clustering	meilleur ARI (pour 7 vect. propres)	N.C.	.4178	
Affinity Propagation	Pour 6 clusters, *Preference*=-18.2		.1955	
Smart Local Moving Algorithm	Meilleur ARI (pour *Resolution*=1000, *minpts*=30)		**.6019**	**.5484**
InfoMap	COS seuillé à 0.1	.2420		**.4359**
Density Peaks	meilleur ARI (pour noyau Gaussien et *neighbor rate* = 0.08%)	.2624		.4018

6 - Bibliographie :

[1][Lewis et al. 2004] Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, *5*(Apr), 361-397. http://www.daviddlewis.com/resources/ testcollections/rcv1/

[2] [Apté et al. 1994] Apté, Chidanand, Fred Damerau, and Sholom M. Weiss. "Automated learning of decision rules for text categorization." *ACM Transactions on Information Systems (TOIS)* 12.3 (1994): 233-251.

[3] [Cai et al. 2005] Cai, Deng, Xiaofei He, and Jiawei Han. "Document clustering using locality preserving indexing." *IEEE Transactions on Knowledge and Data Engineering* 17.12 (2005): 1624-1637. http://www.cad.zju.edu.cn/home/dengcai/Data/ TextData.html

[4][Cover, Thomas 1991] Cover, Thomas M., and Thomas, Joy A. "Entropy, relative entropy and mutual information." *Elements of information theory*2 (1991): 1-55.

[5][Rand 1971] Rand, William M. "Objective criteria for the evaluation of clustering methods." *Journal of the American Statistical association* 66.336 (1971): 846-850.

[Zitt et al. 2000] Zitt, Michel, Elise Bassecoulard, and Yoshiko Okubo. "Shadows of the past in international cooperation: Collaboration profiles of the top five producers of science." *Scientometrics* 47.3 (2000): 627-657.

[6][Choi et al. 2010] Choi, Seung-Seok, Sung-Hyuk Cha, and Charles C. Tappert. "A survey of binary similarity and distance measures." *Journal of Systemics, Cybernetics and Informatics* 8.1 (2010): 43-48.

[Cadot, Lelu 2012] Martine Cadot, Alain Lelu. Combining Explicitness and Classifying Performance via MIDOVA Lossless Representation for Qualitative Datasets. International Journal On Advances in Software, IARIA, 2012, 5 (1&2), pp.1-16. 〈hal-00596718〉

[7] [Deerwester et al. 1988] Scott Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, Beck L.: Improving information retrieval with latent semantic indexing, Proceedings of the 51st Annual Meeting of the American Society for Information Science 25, 36–40 (1988)

[7bis][Von Luxburg 2007] Von Luxburg, Ulrike. "A tutorial on spectral clustering." *Statistics and computing* 17.4 (2007): 395-416.

[8][Hérault, Ans 1984] J. Hérault and B. Ans, "Réseau de neurones à synapses modifiables: décodage de messages sensoriels composites par apprentissage non supervisé et permanent", *Comptes Rendus de l'Académie des Sciences Paris, série 3*, 299: 525-528, 1984.

[9][Hyvärinen 1999] Hyvärinen, Aapo. "Fast and robust fixed-point algorithms for independent component analysis." *IEEE Transactions on Neural Networks* 10.3 (1999): 626-634.

[10][Lee, Seung 1999] Lee, Daniel D., and H. Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization." *Nature* 401.6755 (1999): 788-791.

[11][Lelu 1994] Alain Lelu: Clusters *and* factors: Neural algorithms for a novel representation of huge and highly multidimensional data sets. In: New Approaches in Classification and Data Analysis, ed. by Edwin Diday, Yves Lechevallier, Martin Schader, Patrice Bertrand (Springer, Berlin 1994) pp.241–248

[12][Domengès, Volle 1979] Domengès, Dominique, and Michel Volle. "Analyse factorielle sphérique: une exploration." *Annales de l'INSEE*. Institut national de la statistique et des études économiques, 1979.

[13][Kohonen 1998] Kohonen, Teuvo. "The self-organizing map." *Neurocomputing*21.1 (1998): 1-6.

[14][Hofman 1999] Thomas Hofmann: Probabilistic latent semantic indexing, SIGIR'99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, Berkeley, CA 1999, ed. by Fredric Gey, Marti Hearst, Richard Tong (ACM, New York, NY 1999) 50–57

[15] [Dempster et al. 1977] A.P. Dempster, N.M. Laird et Donald Rubin, « Maximum Likelihood from Incomplete Data via the EM Algorithm », *Journal of the Royal Statistical Society.* Series B (Methodological), vol. 39, no 1, 1977, p. 1–38

[16][Fisher 1912] Ronald Fisher, « On an absolute criterion for fitting frequency curves », Messenger of Mathematics, no 41, 1912, p. 155-160

[17][Blei et al. 2003] David M. Blei, Andrew Y. Ng, Michael I. Jordan: Latent Dirichlet Allocation, Journal of Machine Learning Research 3, 993–1022 (2003)

[18][Dunn 1973] Dunn, J. C. (1973-01-01). "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters". Journal of Cybernetics. 3 (3): 32–57

[19][Ester et al. 1996] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu: A densitybased algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise, KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, OR 1996, ed. by Evangelos Simoudis, Jiawei Han, Usama Fayyad (AAAI, Palo Alto 1996) 226–231

[20][Blondel et al. 2008] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre: Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment 2008(10), P10008 (2008)

[21][van Eck et al. 2010] Nees Jan van Eck, Ludo Waltman: Software survey: VOSviewer, a computer program for bibliometric mapping, Scientometrics 84(2), 523–538 (2010)

[22][Lancichinetti, Fortunato 2011] Lancichinetti, Andrea, and Santo Fortunato. "Limits of modularity maximization in community detection." *Physical review E* 84.6 (2011): 066122.

[23][Dueck, Frey 2007] Frey, Brendan J., and Delbert Dueck. "Clustering by passing messages between data points." Science 315.5814 (2007): 972-976.

[24] [Meila, Shi 2000] Marina Meila, Jianbo Shi: Learning Segmentation by Random Walks, NIPS'00: Proceedings of the Neural Information Processing Systems Conference, Denver, CO 2000, ed. by Todd K. Leen, Thomas G. Dietterich, Volker Tresp (MIT Press, Cambridge, MA 2000) 873–879

[25] [Lelu, Cadot 2010] Alain Lelu, Martine Cadot. Espace intrins`eque d'un graphe et recherche de communaut´es. Fr´ed´eric Amblard. Premi`ere conf´erence sur les Mod`eles et l'Analyse des R´eseaux : Approches Math´ematiques et Informatique - MARAMI 2010, Oct 2010, Toulouse, France. pp.1, 2010. HAL Id: hal-00516865

[26][Lelu, Cadot 2013] Alain Lelu, Martine Cadot. A Proposition for Fixing the Dimensionality of a Laplacian Low-rank Approximation of any Binary Data-matrix. The Fifth International Conference on Information, Process, and Knowledge Management - eKNOW 2013, Feb 2013, Nice, France. IARIA, pp.70-73, 2013. 〈hal-00773436〉

[27][Rosvall, Bergstom 2007] Martin Rosvall, Carl T. Bergstrom: An information-theoretic framework for resolving community structure in complex networks, Proceedings of the National Academy of Sciences 104(18), 7327–7331 (2007)

[28][Rodriguez, Laio 2014] A. Rodriguez, A. Laio: Clustering by fast search and find of density peaks, Science 344(6191), 1492–1496 (2014)

[Reinert 1986] Max Reinert: Un logiciel d'analyse lexicale, Les cahiers de l'analyse des données 11(4), 471–481 (1986)

[Benzécri 1973] Jean-Paul Benzécri: L'analyse des correspondances, Analyse des données, Vol.2 (Dunod, Paris 1973)

[Robertson et al. 1994] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline HancockBeaulieu, Mike Gatford: Okapi at TREC-3, TREC'94: Proceedings of the 3rd Text REtrieval Conference, Gaithersburg, MA 1994, ed. by Donna K. Harman (NIST, Gaithersburg, MA 1994) 109–126

7 - Annexe : les données pour constituer la table 2

Les 2 tableaux suivants donnent les présences de 50 mots dans les 4 documents E, F, G et H de la table 2, avec dans la première ligne les supports pour les 1-motifs E, F, G, H, pour les 2-motifs EF, EG, …, GH, pour les 3-motifs et pour le 4-motif EFGH. Seuls les supports de EFG et EFH diffèrent dans les 2 tableaux

Tot.	20	10	30	7	5	9	4	6	3	2	**4**	**0**	0	2	0
N°mot	E	F	G	H	EF	EG	EH	FG	FH	GH	EFG	EFH	EGH	FGH	EFGH
1	1	1	1	0	1	1	0	1	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
9	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0
10	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0
11	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0
12	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0
13	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0
14	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0

15	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0
16	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0
17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	1	1	1	0	0	0	1	1	1	0	0	0	1	0
22	0	1	1	1	0	0	0	1	1	1	0	0	0	1	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
41	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
42	1	1	1	0	1	1	0	1	0	0	1	0	0	0	0
43	1	1	1	0	1	1	0	1	0	0	1	0	0	0	0
44	1	1	1	0	1	1	0	1	0	0	1	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0
49	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0
50	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0

occ	20	10	30	7	5	9	4	6	3	2	**1**	**1**	0	2	0
N°mot	E	F	G	H	EF	EG	EH	FG	FH	GH	**EFG**	**EFH**	EGH	FGH	EFGH
1	1	1	1	0	1	1	0	1	0	0	1	0	0	0	0
2	1	1	0	1	1	0	1	0	1	0	0	1	0	0	0
3	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0
4	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0
5	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0
6	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0
7	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0
8	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0
9	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0
10	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0
11	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0
12	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0
13	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0
14	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0
15	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0
16	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0

17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	1	1	1	0	0	0	1	1	1	0	0	0	1	0
22	0	1	1	1	0	0	0	1	1	1	0	0	0	1	0
23	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0
24	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0
25	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0
26	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
41	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
45	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table des matières