Clustering de données textuelles en grande dimension

Stéphanie Combes (INSEE)

Plan du cours

- Analyse textuelle et clustering de textes
- Architecture distribuée et algorithmes distribués pour le clustering de textes

Clustering de données textuelles

Basics

- L'analyse textuelle vise d’abord à **transformer le texte en données numériques manipulables**.
- On peut ensuite utiliser diverses techniques (clustering en particulier) dont certaines sont spécifiques au texte

Un format complexe...

Les données textuelles ont pour particularité d’être **non structurées**, l'information cherchée est perdue au milieu d'une (grande) quantité d'information et doit être interprétée.

Il y a d’assez faibles chances pour que deux textes même lorsqu’ils expriment le même message, soient rédigés exactement avec **les mêmes mots**.

S’ajoutent à cela, un certain nombre de problèmes...

… à normaliser

En effet, les données texte sont souvent :

- **bruitées** (orthographe, fautes de frappe)
- **changeantes** (nouveaux termes, nouveaux sens)
- **complexes** (structure, syntaxe, morphologie)
- **ambiguës** (synonymie, polysémie, intentionnalité)
- **langue-dépendantes** (avec un manque de traducteur robuste pour toutes les langues)
- d’une **grande dimension** (grand nombre de mots)

Représenter le texte sous format numérique

Transformer le texte

Jargon : lorsque les “observations statistiques” sont des données textuelles on parle de “**documents**”, dont l’ensemble forme le “**corpus**”.

Etape 1 : chaque texte/document est décomposé en **composants** (*tokens*) : termes, ponctuation, nombres, et harmonisé (casse, orthographe : dictionnaire, matching, phonétique).

Informatiquement parlant, ça veut dire que la chaîne de caractère “le ciel est bleu !” devient le vecteur [‘le’,’ciel’,’est’,’bleu’,’!’]

Transformer le texte

On peut “extraire” le **vocabulaire** de notre corpus, c’est-à-dire la liste des mots utilisés au moins une fois dans un document :

Doc 1 : “le ciel est bleu aujourd’hui”

Doc 2 : “le ciel est toujours gris à Paris”

Vocabulaire : {“le”, “ciel”, “est”, “toujours”, “bleu”, “gris”, “aujourd’hui”, “a”, “Paris”}

Les documents peuvent alors être représentés dans l’espace du vocabulaire :

Doc 1 : [1, 1, 1, 0, 1, 0, 1, 0, 0]

Doc 2 : [1, 1, 1, 1, 0, 1, 0, 1, 1]

Transformer le texte

Ici on a choisi un **codage binaire** : absence/présence d'un mot dans le document, mais on peut aussi calculer la **fréquence** (nombre d'occurrences), ou d'autres types de pondérations (cf infra).

On empile ces vecteurs pour former la matrice **documents x termes**, qui est numérique, de taille : taille du corpus x taille du vocabulaire

Cette matrice peut ensuite être manipulée pour faire des analyses statistiques :

- Mesurer la similarité entre deux mots, documents
- Classer les documents
- Catégoriser les documents
- Identifier les thématiques...

	call	time	date	conference	release	meeting	corporation	earnir.
document 1	2	1	3	2	1	1	1	
document 2	1		2	1	2	1	1	1
document 5		1	2		2	1	1	1
document 6	1	2	1	1	3	1	1	1
document 7	1						1	
document 8			1		1		1	1
document 9	2		1	3	1	1	1	1
document 10	2	1		1	1		1	1
document 13					1			2
document 14							3	
document 15	1			2			1	2

Remarque importante

On fait référence à cette approche par le terme “**sac de mots**”, en effet dans cette approche l’ordre des mots est perdu.

Imaginons que le vocabulaire soit : {“le”, “chat”, ”chien”, “oiseau”, “a”, “mangé”, “l”}

Les documents ci-dessous ont la même représentation pourtant ils n’ont pas le même sens

Doc 1 : “le chat a mangé l oiseau”, Doc 1 : “l oiseau a mangé le chat”

Représentation vectorielle : [1, 1, 0, 1, 1, 1, 1]

Remarque importante

Ces premières étapes peuvent s'avérer assez fastidieuses car pour extraire le vocabulaire, il faut d'abord **harmoniser** le texte, faire en sorte que "ciel" et "Ciel" soient considérés comme le même mot.

En général, on commence systématiquement par :

- Harmoniser la **casse**, on retire les espaces en trop
- On essaie de corriger l'**orthographe** (on peut notamment utiliser des correcteurs orthographiques type Aspell, des expressions régulières ou la phonétique)
- On enlève les chiffres, la ponctuation, s'ils ne sont pas utiles à l'analyse

Réduire la dimension

Réduire la dimension

Le vocabulaire utilisé dans un corpus peut être très grand, on peut avoir des dizaines voire centaines de milliers de colonnes.

Même si la matrice est très creuse, pour manipuler ces matrices, on va naturellement chercher à **réduire la dimension**.

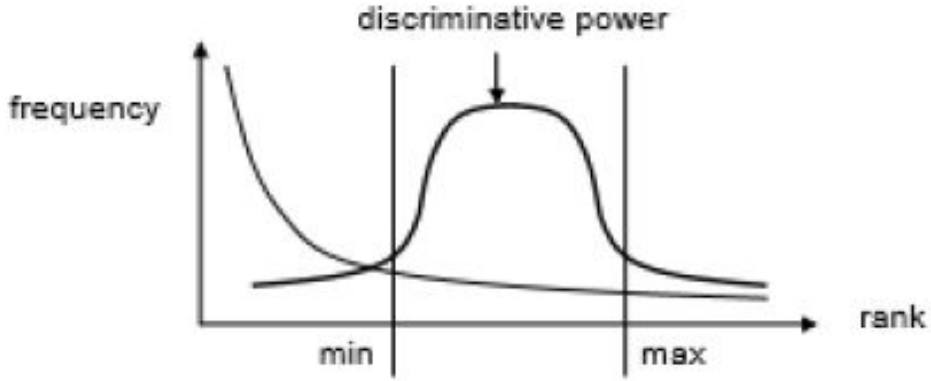
On va retirer les mots présents partout et tout le temps, ils sont inutiles à l'analyse statistique, ils ne serviront ni à discriminer des catégories, ni à identifier les sujets traités.

Réduire la dimension

Les **mots outils** (*stopwords*) font partie de ces mots, il s'agit de mots tels que "au", "avec", "dans", "mais", "elle", "lui", "nos"… qui ont de bonnes chances de se trouver dans à peu près n'importe quel document.

"Le ciel est bleu à Paris" devient "ciel bleu paris".

On retire également les mots trop rares (on parle d'"hapax" pour les mots présents une seule et unique fois)



Réduire la dimension

On peut également tirer partie de “classes d’équivalence” : on peut considérer que différentes formes d'un même mot (pluriel, singulier, conjugaison) sont équivalentes et les remplacer par une même **forme** dite **canonique**, deux approches classique :

- La **lemmatisation** qui requiert la connaissance des statuts grammaticaux ou POS tagging, exemple : chevaux -> cheval
- La **racinisation** plus fruste mais plus rapide, exemple : chevaux -> chev (peut être confondu avec cheveu, chevet par exemple)

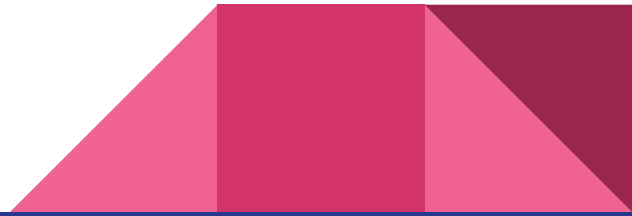
Les deux approches sont langage dépendantes.

Réduire la dimension

Enfin, on peut s’appuyer sur les **pondérations**.

En créant la matrice documents-termes, on a accès aux fréquences des mots dans les documents, donc on peut retenir les colonnes qui sont non nulles plus de 5% des lignes par exemple (le mot est présent dans au maximum 95% des documents).

On procède de même pour retirer les mots qui seraient des artefacts statistiques (présents trop rarement).



Réduire la dimension

Une autre pondération classique : **TF-IDF** pondère au sein de la matrice, chaque mots en fonction de son apparition au sein d'un document et du corpus :

TF-IDF pondère plus les mots qui apparaissent souvent dans peu de documents

retirer 5% des mots qui ont une pondération trop faible permet donc de filtrer les mots non discriminants car trop répandus ou les mots trop rares d'un coup.

$$TF - IDF(t, d, D) = \frac{TF(t, d)}{DF(t, D)} = TF(t, d) \times IDF(t, D)$$

Ces pondérations peuvent être utilisées par les moteurs de recherche pour **classer les documents** par pertinence en fonction d'une requête. Pour établir le rang d'un document, il suffit de sommer par document les scores tf-idf des termes composant une requête.

Réduire la dimension (avancé)

Une façon plus avancée de réduire la dimension, et d'essayer d'**identifier des concepts**, pour remplacer plusieurs colonnes par le concept plus général correspondant, soit explicitement :

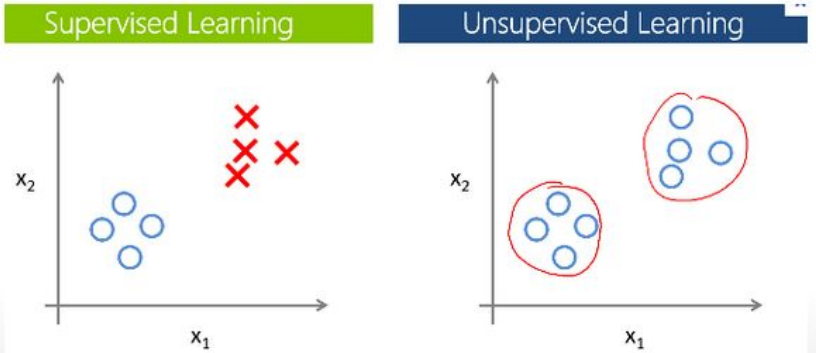
- En utilisant un **dictionnaire de synonymes** par exemple pour remplacer toutes les colonnes de mots synonymes par une seule colonne (cf infra)

- En utilisant des **techniques de clustering** (LSA, LDA, NMF, word2vec … cf infra)

Que faire avec cette matrice ?

La représentation sous forme matricielle facilite la comparaison des mots / documents et l'utilisation d'outils de machine learning (entre autres !), les objectifs peuvent être multiples :

- Mesurer la **similarité** entre deux termes (utilisés dans les mêmes documents), la similarité entre deux documents (composés des mêms termes)
- **Catégoriser, labelliser** automatiquement des documents pour filtrer ou produire des statistiques (par exemple identifier les articles de presse traitant d'économie pour ne retenir que ceux-là dans une étude), identifier du spam, coder des intitulés d'offres d'emploi
- **Identifier les thèmes** abordés par exemple dans les commentaires d'une enquête (notre sujet)

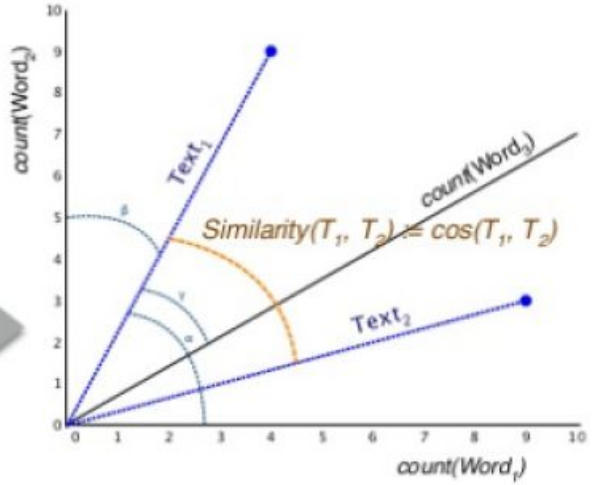


La similarité cosinus

Text 1: He that not wills to the end neither wills to the means.

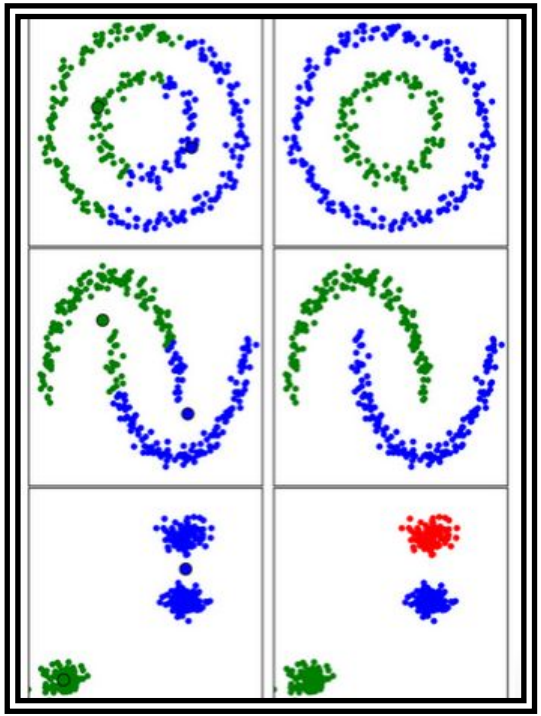
Text 2: If the mountain will not go to Moses, then Moses must go to the mountain.

tokens	Text 1	Text 2
end	1	0
go	0	2
he	1	0
if	0	1
means	1	0
Moses	0	2
mountain	0	2
must	0	1
not	1	1
that	1	0



$$sim(\vec{x}, \vec{y}) = cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}||\vec{y}|} = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2}\sqrt{\sum y_i^2}}$$

Explorer, partitionner



Le fait de représenter les documents sous forme de vecteur numérique facilite le calcul de distance entre ceux-ci, on rappelle que le but du clustering est de **regrouper des observations similaires** : ici il s'agit soit des documents, soit des termes.

On peut tout à fait appliquer une CAH à la matrices termes x documents par exemple

Un autre algorithme classique en clustering est les **k-means** (exemple suivant).

Il existe également des approches plus spécifiques au texte : **LSA, LDA** : on parle de **topic modelling**

K-means, les profils recherchés dans les offres d’emploi

Les offres d’emploi sur internet

Les instituts nationaux de statistiques s’intéressent aux **offres d’emploi** publiées sur internet (agrégateurs, sites d’entreprises), à des fins de production statistique.

Ces offres peuvent être intéressantes pour **compléter des indicateurs** d’emplois vacants, de tension sur le marché de l’emploi...

… mais peuvent également s’avérer intéressantes pour l’**étude** plus micro des métiers, compétences recherchées, afin mieux cibler les besoins du marché et de mieux orienter les chercheurs d’emploi, en termes de formations professionnelles …

Données

59046 offres d’emplois ont été extraites du site le bon coin au milieu du mois de décembre.

date	23 décembre à 17:44
siren	412481301
desc	Nous cherchons le responsable financier & ...
ref	1612cu
commune	Nice 06000
formation	Agent de maîtrise/Bac+3
sect	Services
exp	5 ans et plus
salaire	2 000 €
tps	Temps plein
contrat	Comptable unique (H/F)
offre	CDD
fonction	Comptabilité/Gestion/Finance

Identifier les compétences

Les descriptions des offres, lorsqu'elles sont un peu longues, sont en général structurées en plusieurs parties :

- **Présentation de l'entreprise**, de la mission
- Description du **profil recherché**, des compétences, appétences, formations requises
- Orientations pour la **suite du processus** (lettre de motivation, prise de contact)

leboncoin ACCUEIL DÉPOSER UNE ANNONCE OFFRES DEMANDES MES ANNON

Description :

Assistants Managers (Futurs Managers) en Restauration Rapide H/F

Entreprise :

Rejoignez Eat Sushi, enseigne leader de la restauration rapide japonaise en livraison, vente à emporter et sur place. Eat Sushi dispose d'un réseau de plus de 20 restaurants en France.
Depuis plus de 10 ans, Eat Sushi affirme son positionnement haut gamme en proposant à la fois un service premium mais également des produits de haute qualité fabriqués sur place.

Dans le cadre de notre expansion, nous sommes à la recherche d'un Assistant Manager (Futurs Managers) H/F (75015) en CDI.

leboncoin ACCUEIL DÉPOSER UNE ANNONCE OFFRES DEMANDES MES ANNONCES

Missions :

• Veiller au respect des normes d'hygiène et de sécurité, à la propreté, à l'entretien et à la tenue du restaurant

• Satisfaire les clients dans une ambiance conviviale que ce soit sur place, à emporter ou en livraison dans les délais impartis.

• Veiller au bon déroulement des livraisons

• Vous êtes responsable de la vérification et du suivi des caisses et supervisez le suivi administratif. Les différentes étapes de clôture du magasin (Administration, Finances et Nettoyage)

• Participer au bon déroulement du service de l'ouverture à la fermeture du restaurant.

• Superviser une équipe et maintenir une bonne ambiance de travail et d'un esprit d'équipe

leboncoin ACCUEIL DÉPOSER UNE ANNONCE OFFRES DEMANDES MES ANNON

Profil recherché :

Vous avez le sens du contact client (pourvu d'un sens commercial inné) et des responsabilités. Vous êtes rigoureux(se), organisé(e), autonome et disponible. Vous aimez travailler en équipe. Vous avez l'esprit d'initiative, du leadership et une bonne résistance au stress. Vous êtes flexible et ouvert au travail en coupure.

- Dynamique, Ponctuel, Consciencieux avec une bonne présentation
- Disponible midi, soir et WE
- Expérience exigée au moins 2 ans dans le secteur de la restauration rapide avec des foctions similaires

Formation(s) exigée(s) :
- Études secondaires (niveau Bac)

Détecter les compétences dans les offres

- On découpe le contenu des offres en phrases (soit **315670 phrases)**, en s'appuyant sur la **ponctuation**.

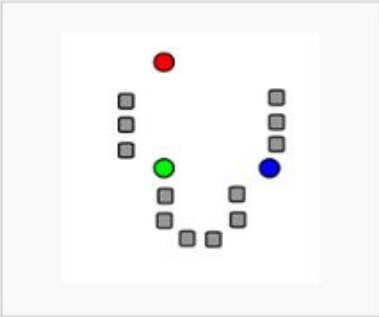
- On considère chaque phrase, si elle n'est pas utile à l'analyse (ne traite pas de compétences), elle sera classée 0, dans le cas contraire, elle sera classée 1.

- On labellise manuellement **746 phrases** d'offres variées (522 négatifs et 224 positifs), on **calibre un algorithme de classification supervisée binaire** sur ces documents une fois **harmonisés** et nettoyés et transformés en matrice **documents x termes**, **on prédit la classe de l'ensemble des 315670 phrases**

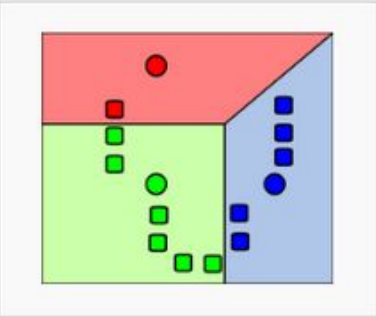
- On peut désormais **filtrer sur la classe prédite,** et éliminer tous les documents classés 0 pour nous concentrer sur les compétences.

Interprétation des résultats

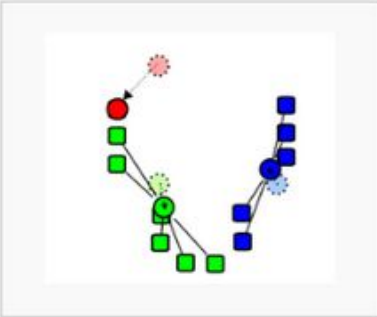
On applique sur le texte des phrases positives un algorithme de clustering classique, ici les **K-means** à la matrice termes documents.



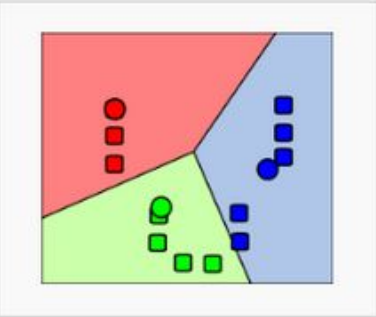
1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).



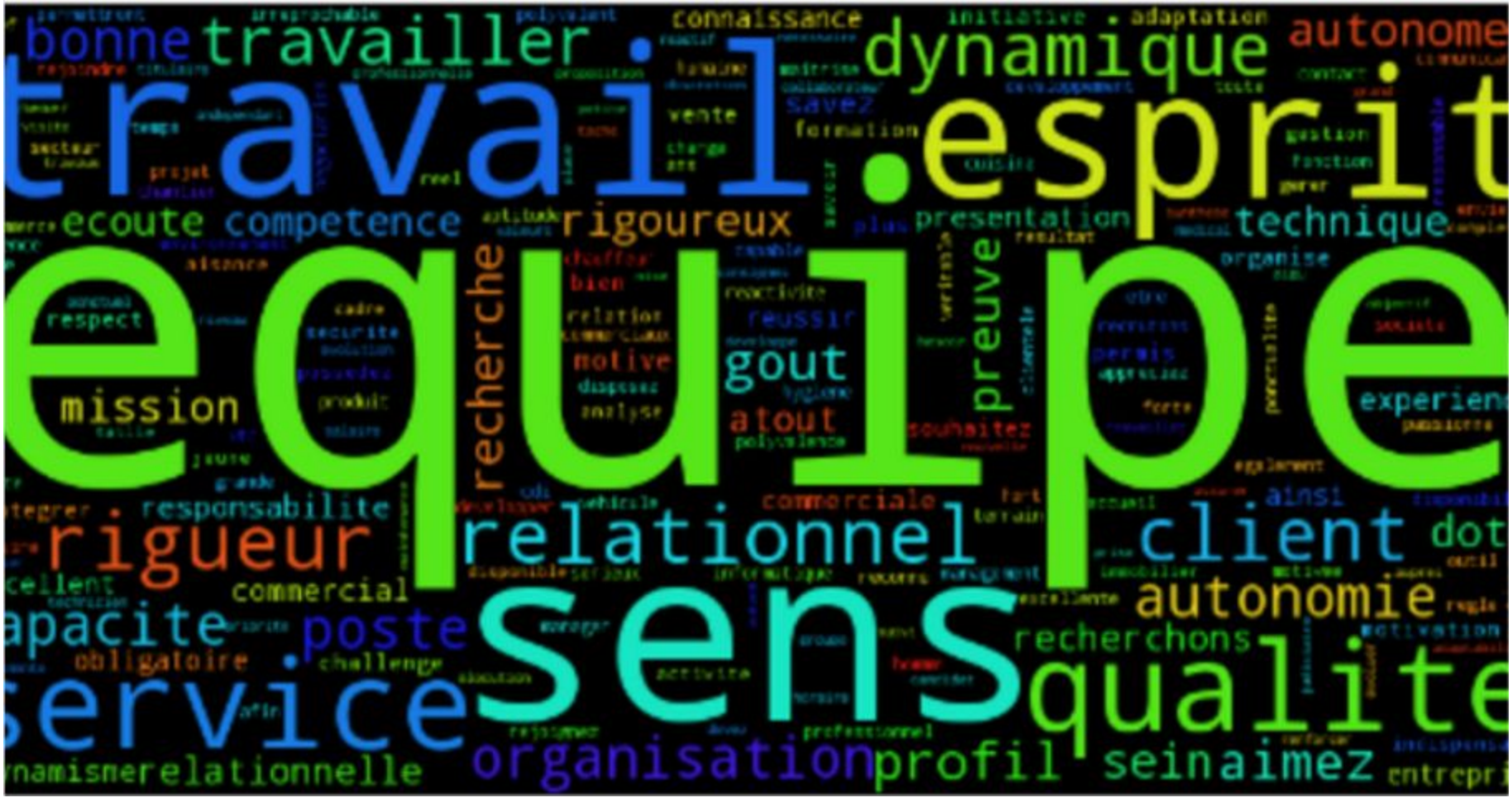
2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

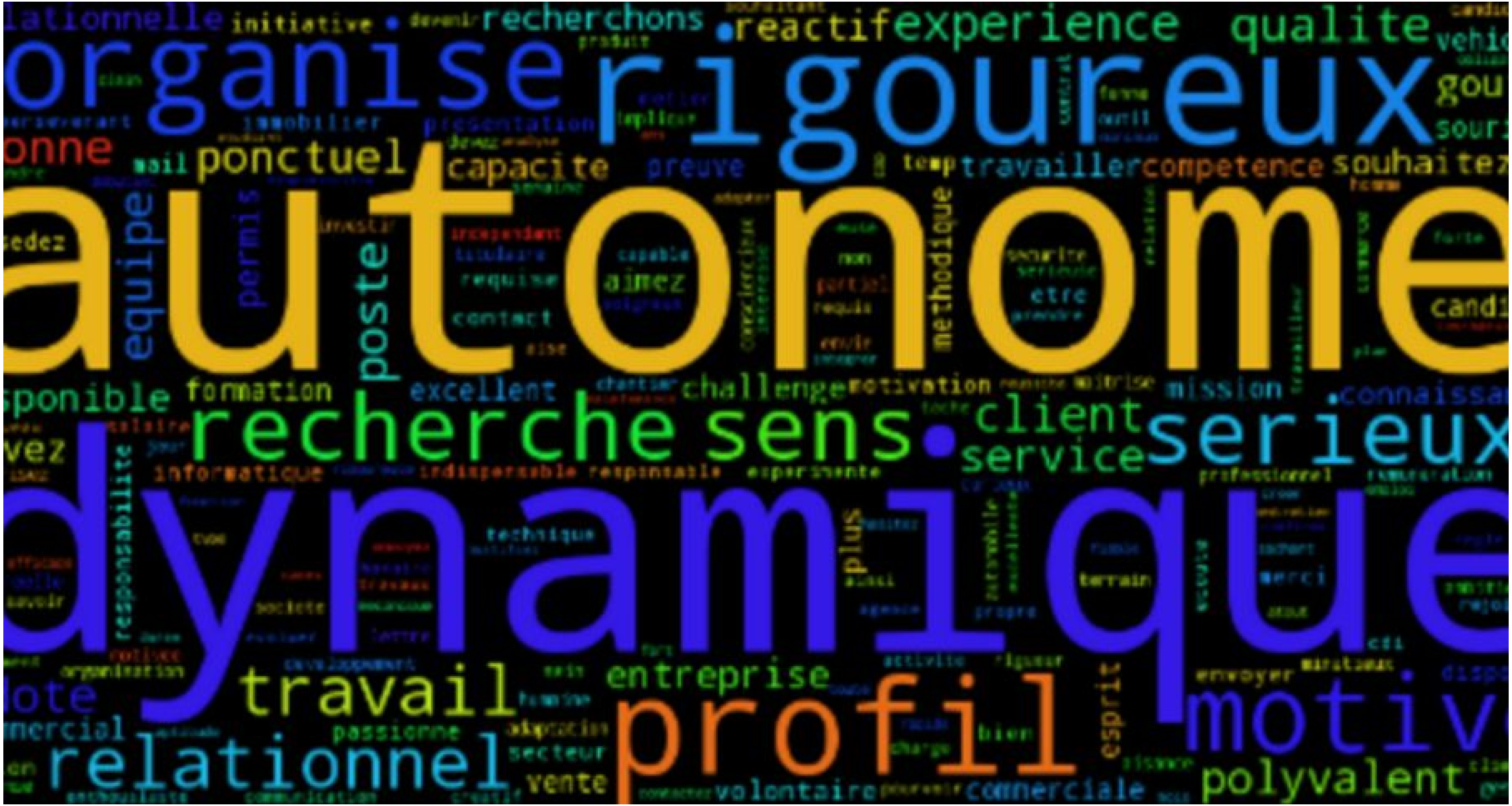


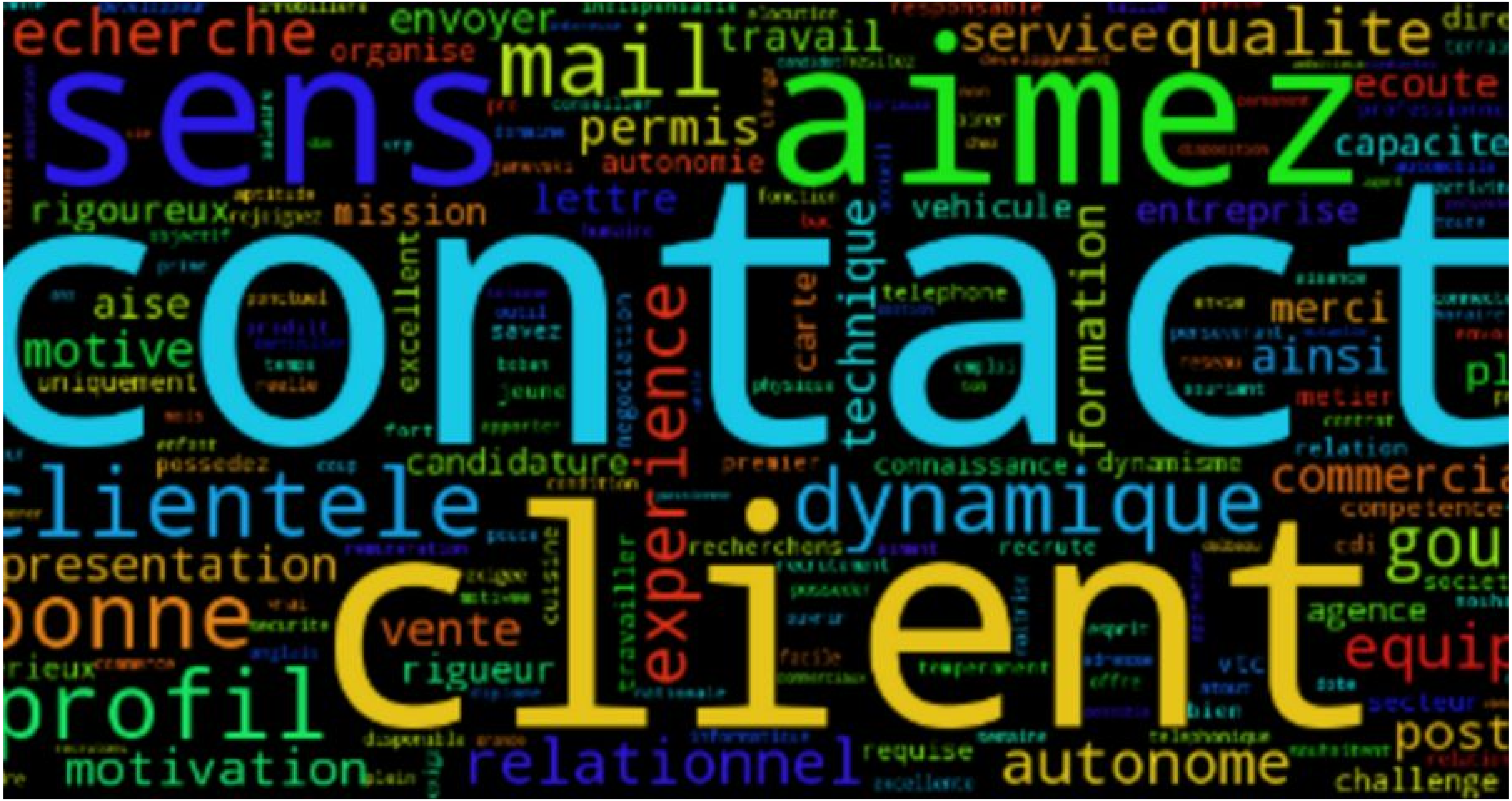
3. The centroid of each of the *k* clusters becomes the new mean.

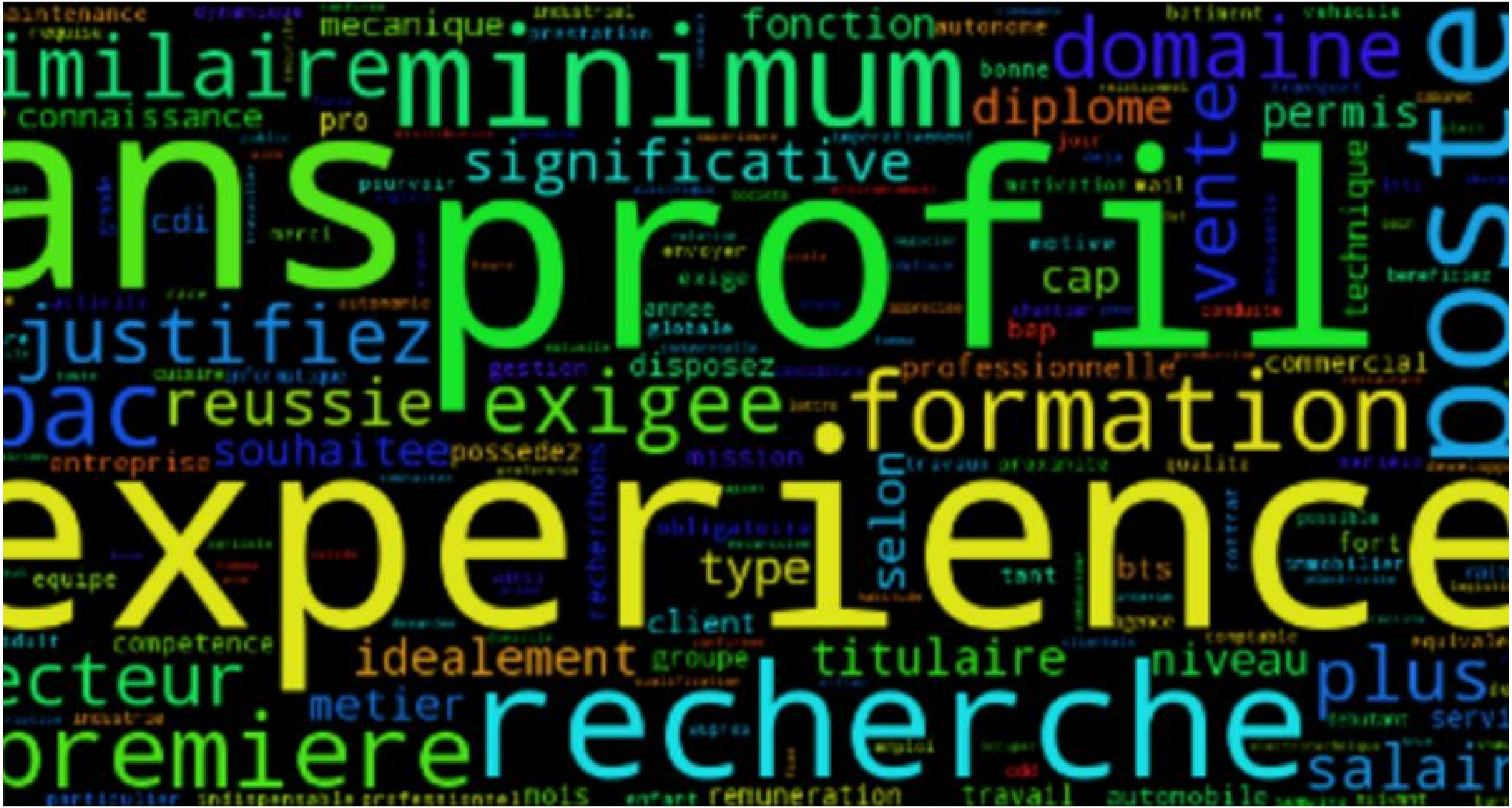


4. Steps 2 and 3 are repeated until convergence has been reached.

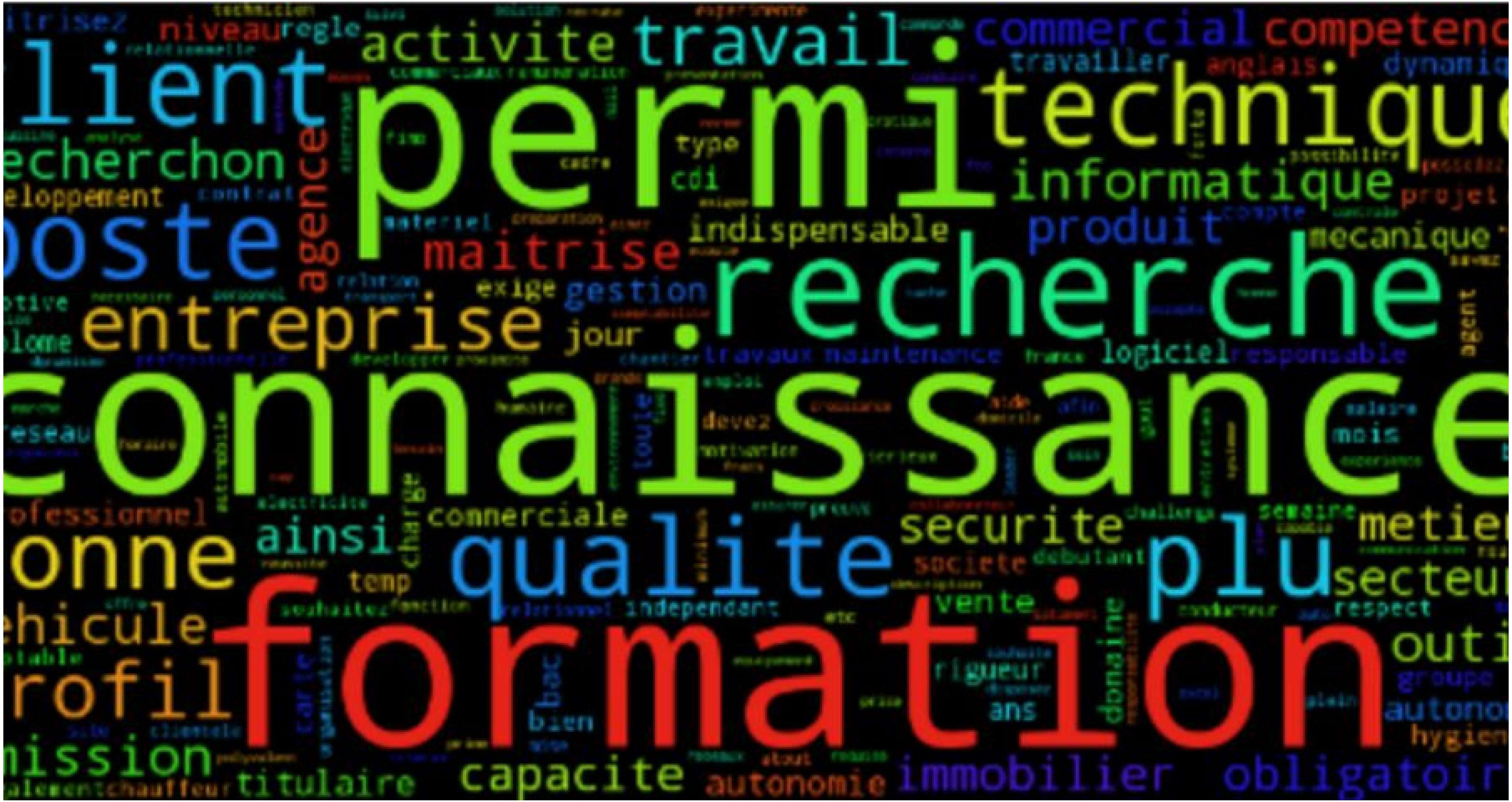

travail
esprit
equipe
sens
qualite
service
bonne
travailler
dynamique
autonome
connaissance
initiative
adaptation
savez
vente
formation
gestion
fonction
ecoute
competence
rigoureux
presentation
technique
recherche
preuve
organise
respect
reussir
gout
motive
atout
mission
experien
souhaitez
commerciale
ainsi
responsabilite
rigueur
relationnel
client
dot
commercial
autonomie
recherchons
activation
apacite
poste
obligatoire
challenge
organisation
profil
seinaimez
entrepri
ynamismerelationnelle


initiative
recherchons
reactif
experience
qualite
organise
rigoureux
ponctuel
capacite
preuve
travailler
competence
souhaitez
autonome
equipe
permis
poste
aimez
etre
contact
methodique
formation
excellent
challenge
motivation
mission
recherche
sens
client
serieux
service
informatique
indispensable
responsable
dynamique
technique
plus
responsabilite
terrain
merci
travail
entreprise
profil
motive
relationnel
passionne
secteur
vente
volontaire
commerciale
polyvalent
esprit
bien


contact
client
aimez
sens
mail
clientele
dynamique
profil
experience
relationnel
autonome
presentation
motivation
permis
autonomie
travail
service
qualite
envoyer
organise
ecoute
capacite
rigoureux
mission
lettre
vehicule
entreprise
technique
formation
carte
aise
motive
uniquement
excellent
merci
ainsi
metier
candidature
connaissance
dynamisme
commercia
relation
telephone
recherchons
recrute
vente
rigueur
agence
equip
challenge
requise
travailler
cuisine



experience
profil
recherche
minimum
formation
domaine
exigee
reussie
justifiez
significative
diplome
permis
fonction
mecanique
connaissance
titulaire
idealement
niveau
plus
metier
vente
cap
selon
type
souhaitee
possedez
entreprise
disposez
professionnelle
commercial
competence
client
groupe
remuneration
travail
automobile
technique
cdi
bts
bep
pro
mission
equipe
bonne
autonome
mois
fort
tant

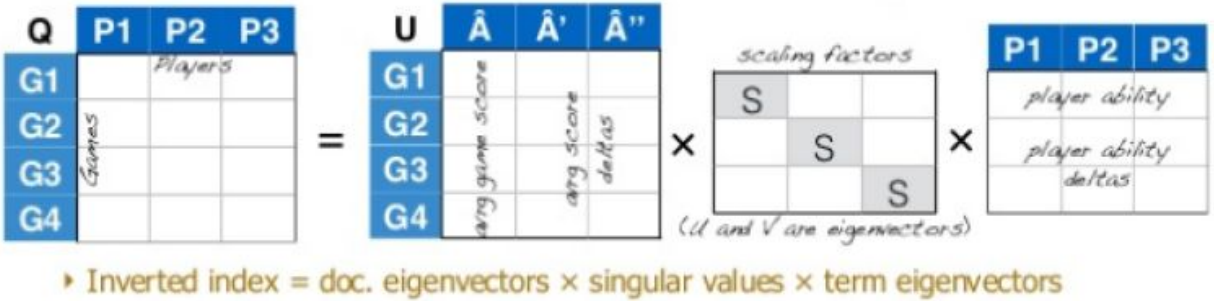


formation
connaissance
permi
recherche
technique
qualite
plu
entreprise
travail
activite
commercial
informatique
produit
indispensable
maitrise
mecanique
securite
capacite
immobilier
ainsi
commerciale
gestion
logiciel
responsable
travaux maintenance
niveau
regle
travailler
anglais
type
cdi
exige
jour
agence
vente
rigueur
ans
bien
titulaire
autonomie
independant
societe
debutant
respect
domaine
bac
temp
charge
devez
toute

LSA / LDA

L’ analyse sémantique latente (LSA)

Mathématiquement parlant, il s’agit d’abord de trouver la **décomposition en valeurs singulières** de matrice documents termes : DTM = USV’.



L’indexation/analyse sémantique latente est une façon de capturer les “**dimensions" sémantiques principales** dans le corpus de textes, ce qui permet de détecter les principaux **“sujets”** et de résoudre, parallèlement, la question de la **synonymie** et de la **polysémie**.

L’ analyse sémantique latente (LSA)

Prenons par exemple une collection de 6 documents utilisant les termes cosmonaute, astronaute, lune, voiture et camion.

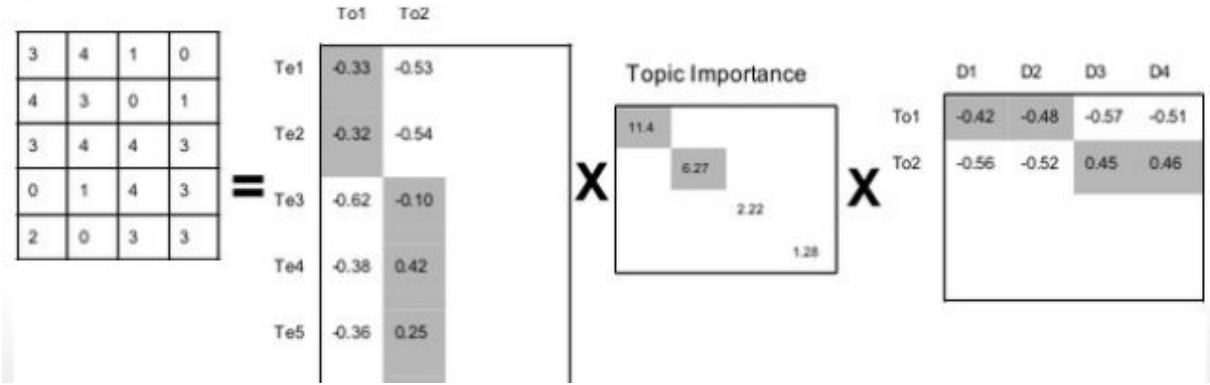
On peut voir que comme cosmonaute et astronaute sont des synonymes, ils n’apparaissent **jamais simultanément** dans les mêmes documents, la similarité cosinus (ou produit scalaire) entre d2 et d3 est donc nulle. Mais ils apparaissent toujours avec le mot lune.

De même, la similarité cosinus vaut 0 entre d5 et d6 alors que les 2 documents contiennent des mots sémantiquement proches.

	d_1	d_2	d_3	d_4	d_5	d_6
cosmonaut	1	0	1	0	0	0
astronaut	0	1	0	0	0	0
moon	1	1	0	0	0	0
car	1	0	0	1	1	0
truck	0	0	0	1	0	1

L’ analyse sémantique latente (LSA)

Pourtant il y a deux dimensions sémantiques : l’espace d’une part, le transport d’autre part qu’il faut parvenir à extraire. L’idée de la LSA va être de réduire l’espace de 5 dimensions (mots) à 2 dimensions (topics/concepts), en utilisant les associations de mots (ici astronaute est utilisé avec le mot lune, tout comme le mot astronaute, les trois forment un “champ lexical” de l’espace).



Les “scores”/**importance des thèmes** dans les documents peuvent se lire par ligne de la matrice U, les scores des termes dans les thèmes, par colonne de la matrice V (faible interprétabilité).

L’ analyse sémantique latente (LSA)

On peut s’en servir à **titre exploratoire** (regarder les “contributions” des documents ou des termes aux axes principaux) ou comme technique de **réduction de la dimension** (en retirant les colonnes de U associées aux plus petites valeurs propres de S, DTM -> U’S’ de taille mxk)

Comme pour un algorithme tel que les k-means ou une ACP, le **nombre de topics doit être choisi par l’utilisateur,** par exemple avec une règle du coude.

La différence avec l’ACP provient de ce que l’ACP s’applique à une matrice de covariance, la CAH s’applique à une matrice de distance (des dissimilarités document x document ou termes x termes), tandis que la LSA s’applique à la matrice documents x termes et **classe simultanément les documents ou les termes**.

L’ allocation latente de Dirichlet (LDA)

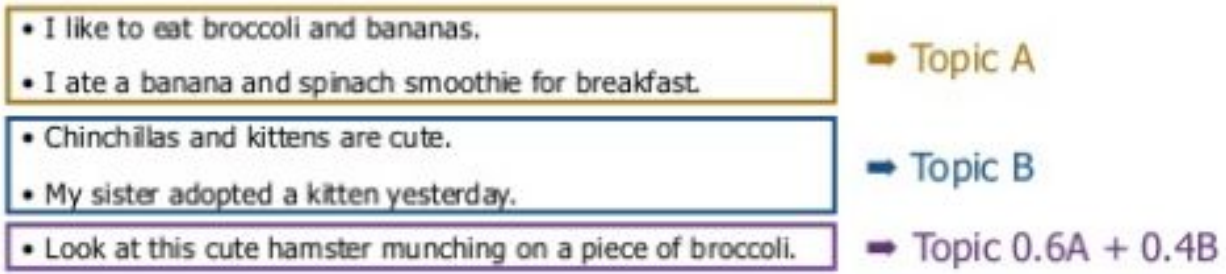
La LDA est une alternative permettant également d’à la fois de d’identifier des groupes de documents et de mots.

Au contraire de la LSA qui est une procédure d’analyse de données, la LDA relève de la **modélisation**.

L’idée sous-jacente est de dire que les documents textuels sont générés à partir de thèmes (et seront donc caractérisés par la distribution des thèmes qu’ils abordent), ces mêmes thèmes étant caractérisés par une certaine distribution de mots.

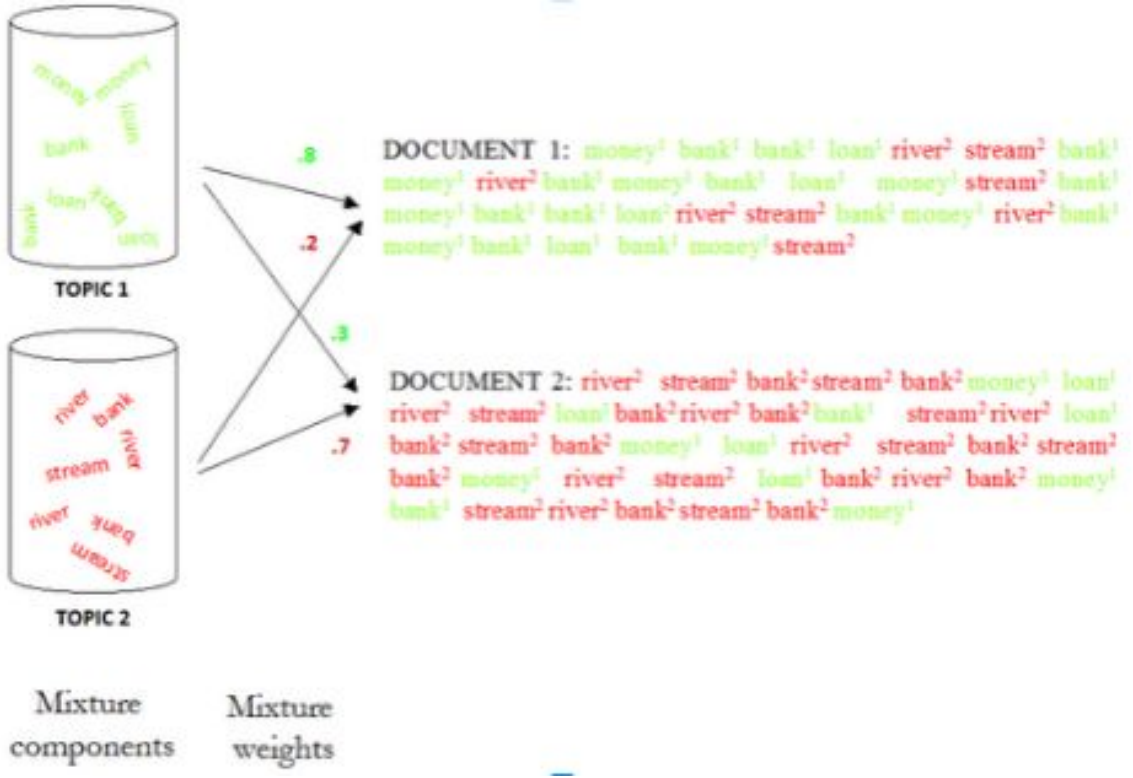
L’ allocation latente de Dirichlet (LDA)

Les documents sont donc composés de “thèmes” intervenant dans des proportions variables d’un document à l’autre



L’ allocation latente de Dirichlet (LDA)

Cependant les thèmes ne sont **pas directement observés**, ils sont identifiables par le biais de mots les caractérisant (sachant qu’un mot peut intervenir dans des thèmes différents).



L’ allocation latente de Dirichlet (LDA)

Formellement, soit T thèmes, la probabilité que le mot i (sur V mots), se trouve dans le document d fait intervenir **l’importance du mot dans les thèmes**, pondérée par **l’importance des thèmes dans le document** :

$$p(w_i) = \sum_{j=1}^{T} p(w_i, z_j) = \sum_{j=1}^{T} p(w_i|z_j)p(z_j)$$

On note $\phi_{i,j} = p(w_i|z_j)$ le **poids** du mot i dans le thème zj, on note la matrice VxT Phi

On note $\theta_{d,j} = p(z_j)$ le **poids** du thème zj pour un document donné d. La matrice TxD est notée Théta

L’ allocation latente de Dirichlet (LDA)

On **choisit deux distributions de Dirichlet dans lesquelles on tire Phi et Theta.**

On parle de distributions **a priori** car elles rendent compte des ‘croyances’ de l’utilisateur avant observation des données (si on pense que les thèmes/mots sont distinctifs ou pas, cf infra)

Le processus générateur des données peut donc s’écrire :

$$p(\mathbf{w}, \mathbf{z}, \theta, \phi | \alpha, \beta) = p(\phi|\beta)p(\theta|\alpha)p(\mathbf{z}|\theta)p(\mathbf{w}|\phi_z)$$

$$\phi^{(k)} \sim Dirichlet(\beta)$$

$$\theta_d \sim Dirichlet(\alpha)$$

L’ allocation latente de Dirichlet (LDA)

Concrètement, on a au départ les mots et les documents, et à l’arrivée **on veut les pondération des thèmes par documents et des mots par thèmes (z, Phi, Theta)**.

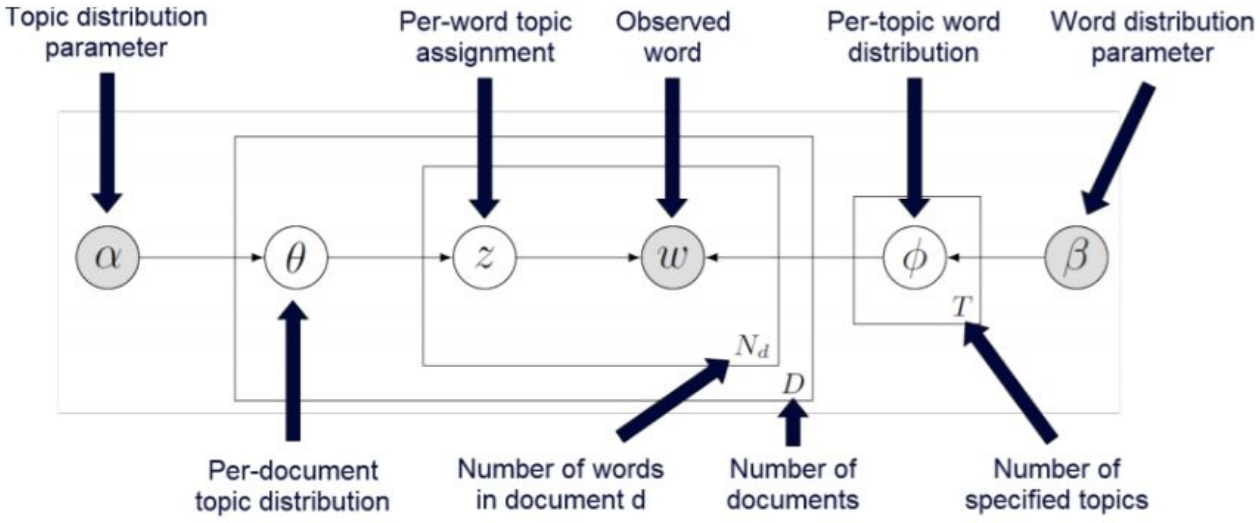
Avec la formule de Bayes, on a la distribution a posteriori :

$$p(\theta, \phi, \mathbf{z}|\mathbf{w}, \alpha, \beta) = \frac{p(\theta, \phi, \mathbf{z}, \mathbf{w}|\alpha, \beta)}{p(\mathbf{w}|\alpha, \beta)}$$

On procède par Gibbs Sampling (les distributions ne sont pas directement calculables mais les conditionnelles oui) ou par inférence variationnelle

Représentation sous forme de modèle graphique

1. Pour chaque thème $j \in \{1, \ldots, T\}$, tirer une distribution de mots ϕ_j à partir de Dirichlet$_W(\beta)$.
2. Pour chaque document $d \in \{1, \ldots, D\}$:
   (a) Tirer une distribution de thèmes θ_d à partir de Dirichlet$_T(\alpha)$;
   (b) Pour chaque mot d'indice $n \in \{1, \ldots, N_d\}$ dans le document d :
      i. Tirer un thème $z_{d,n}$ à partir de Multinomial$_T(\theta_d)$;
      ii. Tirer un mot $w_{d,n}$ à partir de Multinomial$_W(\phi_{z_{d,n}})$.

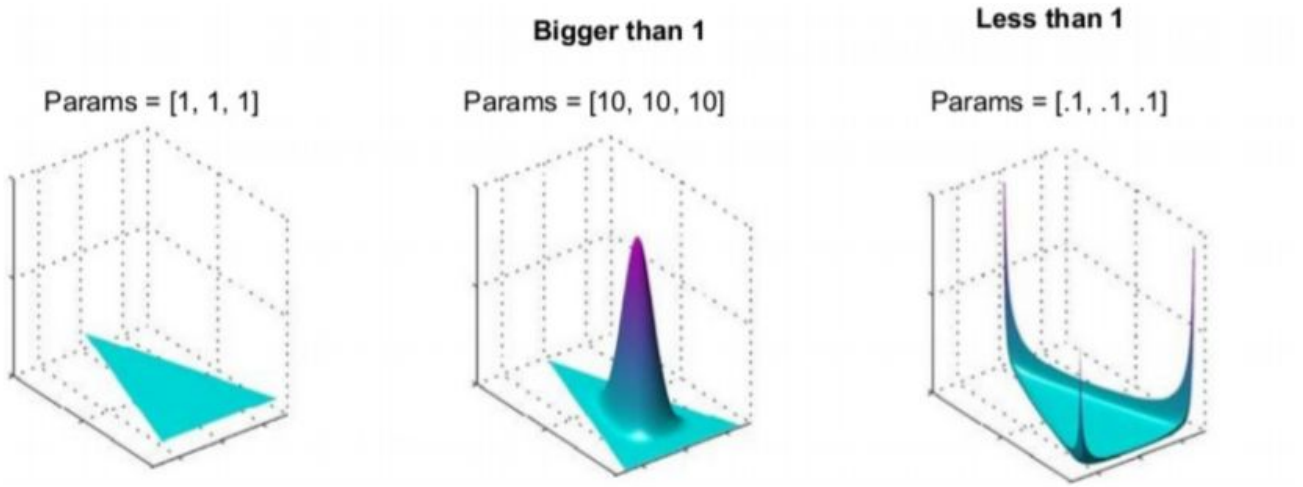


Les noeuds sont les variables aléatoires, s'ils sont grisés ils sont fixés ou observés, les arêtes indiquent une dépendance conditionnelle.

L’ allocation latente de Dirichlet (LDA)

Le recours à la **distribution de Dirichlet** permet de contrôler à quel point on accepte que les documents/mots soient multithématiques.

La distribution de Dirichlet est une distribution de probabilité multivariée continue définie par un vecteur de paramètres réels positifs α.

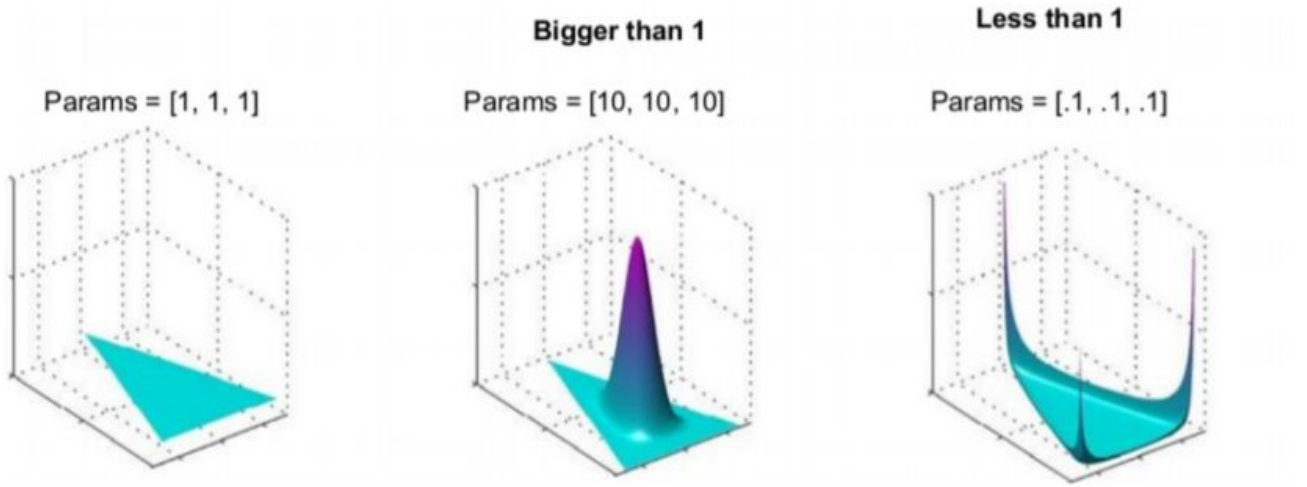


L’ allocation latente de Dirichlet (LDA)

α contrôle **la forme de la distribution**. Si toutes les composantes de α sont égales, la distribution est dite **symétrique** (hypothèse courante pour les Topic Models).

Une distribution de Dirichlet de dimension T a pour support le simplexe (= généralisation du triangle) de dimension T − 1 :

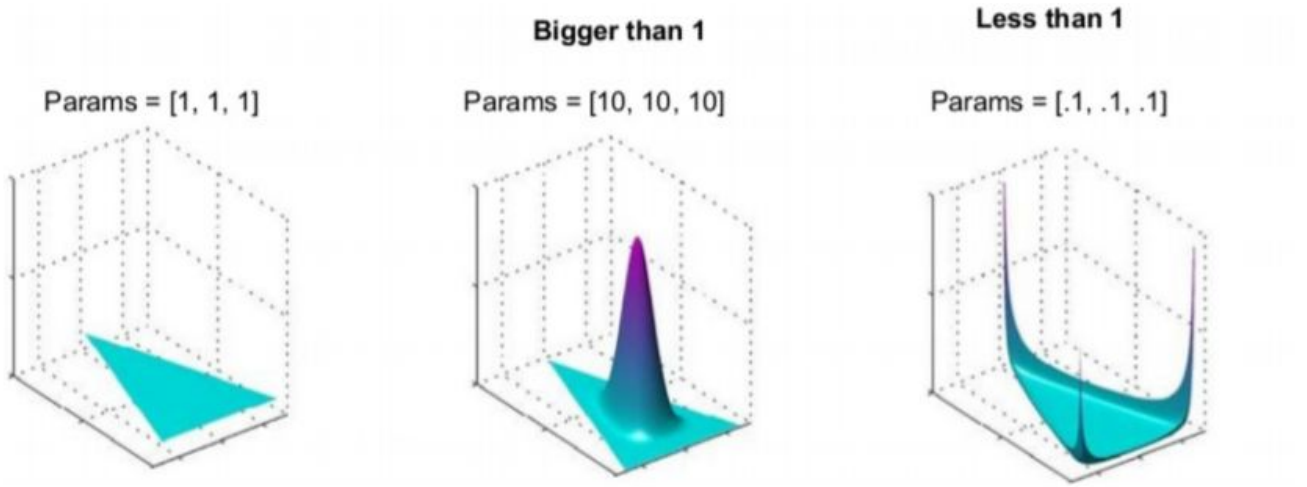
si θ ~ Dirichlet$_T$(α), alors $0 < \theta_t < 1$ et $\sum_{t=1}^{T} \theta_t = 1$



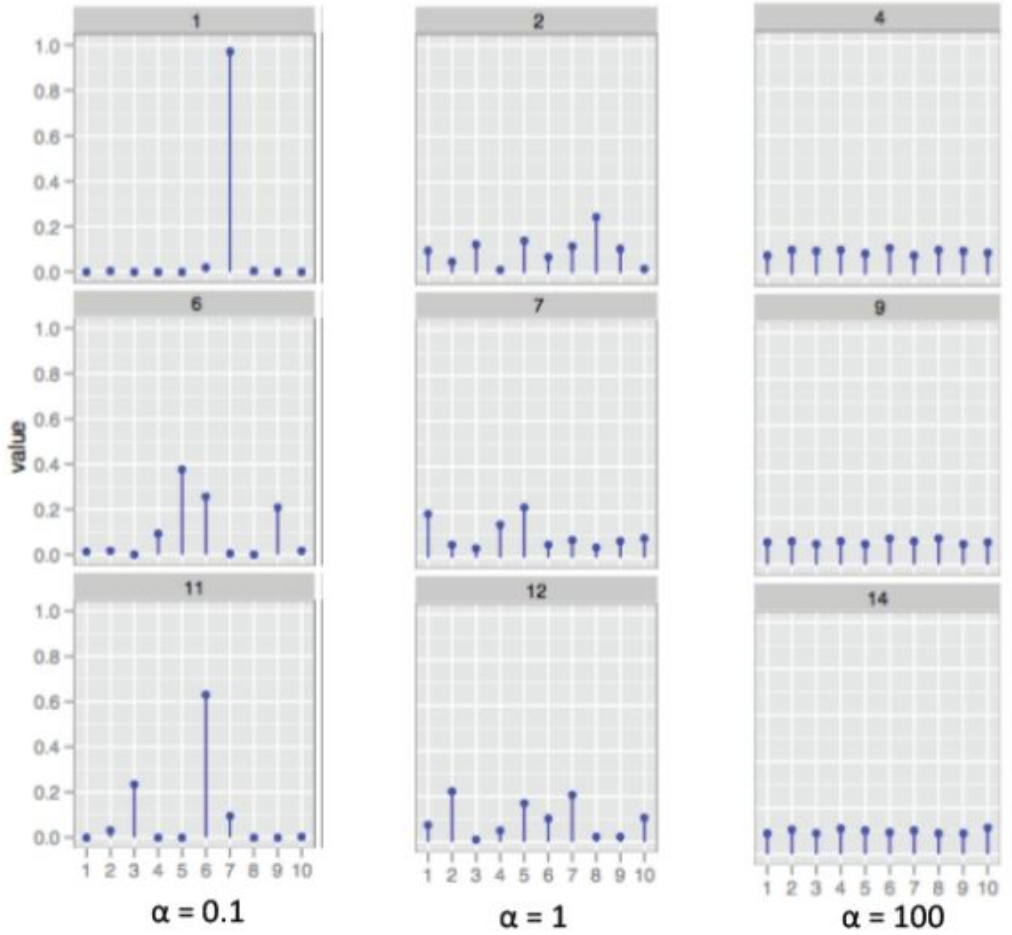
L’ allocation latente de Dirichlet (LDA)

Considérons ici que l’on a 3 mots, le triangle représente le support de la **densité des distributions multinomiales possibles sur les 3 mots**, on considère les thèmes issus d’un tirage dans cette distribution de distributions.

Chaque coin du triangle correspond à la distribution telle que la probabilité 1 est donnée pour l’un des 3 mots. Le milieu d’une arête donne une probabilité de 0.5 à chacun des 2 mots des angles et le centre du triangle correspond à la distribution uniforme sur les 3 mots.



L’ allocation latente de Dirichlet (LDA)



Exemple,
les champs libres des dépôts de plainte

Base STIC DSPAP petite couronne (92, 93, 94)

La table contient 3153503 lignes et 31 colonnes.

Chaque ligne correspond à un individu impliqué. Si la ligne est du type C (“Constaté”), alors cet individu est une victime ; si c’est un type E (“Élucidé”), alors c’est un mis en cause.

Chaque ligne comporte des informations sur :

- la **procédure** : numéro de procédure, service en charge, date de transmission...
- le **fait** : date du fait, tranche horaire du fait, jour de la semaine, nature du lieu, commune... Les index relatifs à la nature du fait sont renseignés dans 3 champs : index1, index2, index3 (par ordre de gravité décroissante), avec en regard le nombre de faits pour chacun (dans nb1, nb2, nb3).
- la **personne** (victime ou mise en cause selon si la ligne est ‘E’ ou ‘C’) : date de naissance, sexe, profession, commune de résidence, nature drogue..
- **champ libre** ‘Véhicule préjudice’

Objectif poursuivi

La nature du préjudice renseigné dans les index1, index2 et index3 peut être regroupée en **24 modalités**, dont certaines se rapportant à des violences ou des infractions à l'encontre de la famille, mais il n'existe pas une catégorie propre aux violences familiales.

Le champ libre peut contenir des informations permettant de préciser la nature du préjudice.

En particulier, on cherche, dans un volet de cette étude à identifier les cas de violences familiales, le champ libre peut donner des indices (par exemple l'expression exacte "violences conjugales", mais également d'**autres formulations inconnues *a priori)*.**

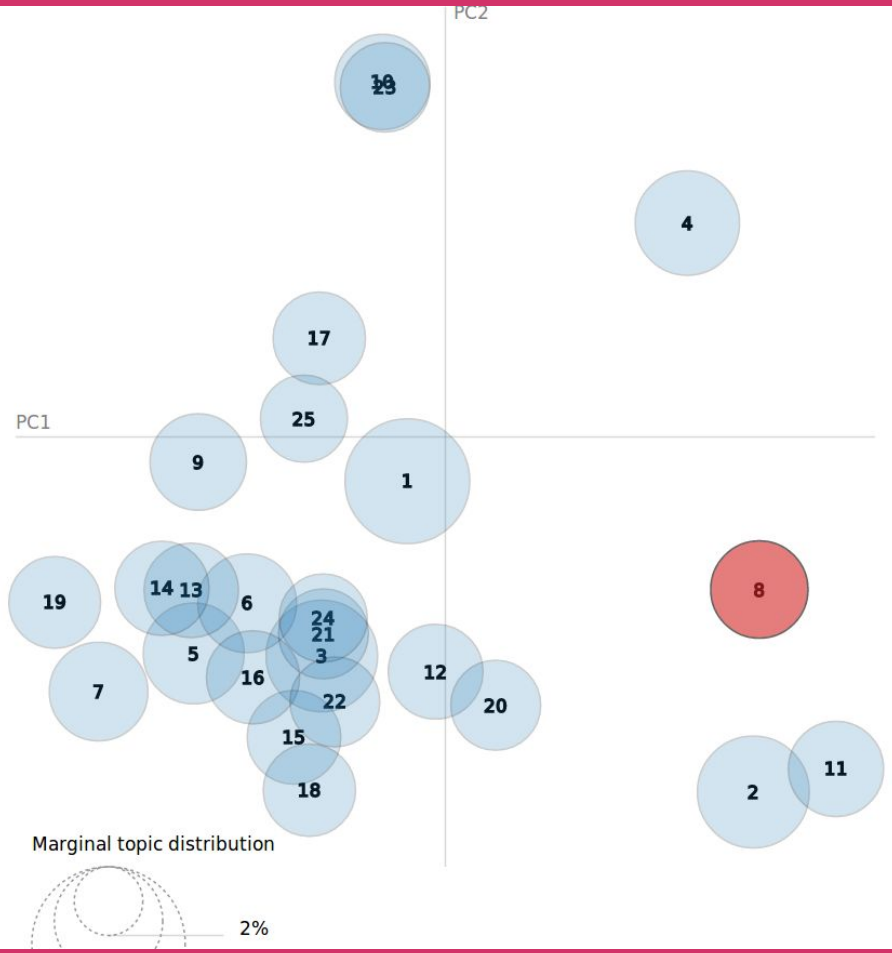
On dispose de 2532416 lignes au champ libre **non vide**

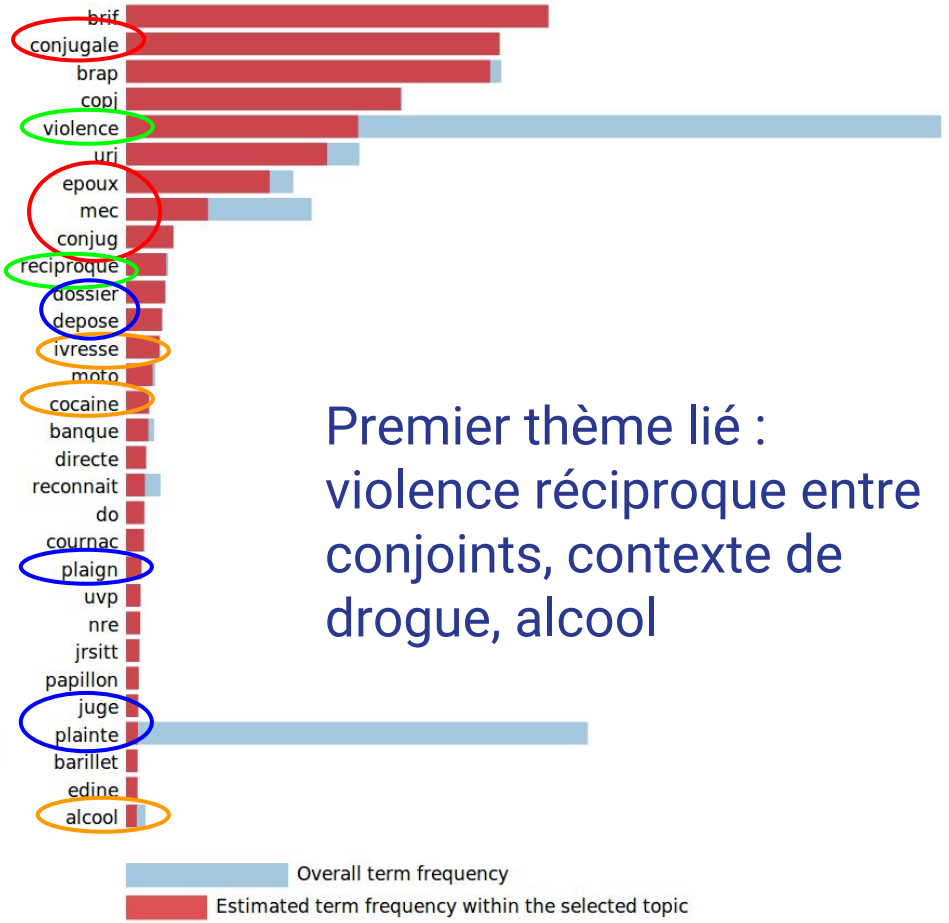
Exemples de champs libres

```
                  appels tel malveillants petit ami
frappe par mari maitresse autres ind itt cmj b...
ile aprf san retention entrave circulation tra...
          suite vol effrac cellule vitre descellee
           ile aucune decision pref gav defere sur
           ile aucune decision pref gav sur defere
                 outrage sur pcmsp agent ratp tvm
                     ile mise demeure quitter css
                   citroen picasso rouge rayures
tentative vol effraction pesee portefenetre pa...
tentative vol effractio fenetre fils allum lum...
       violences volontaires agravees dup req itt
         violences volontaires coup poing dup med
 violences conjugales conj alcool affc epoux itt
                              delaissement mineur
```

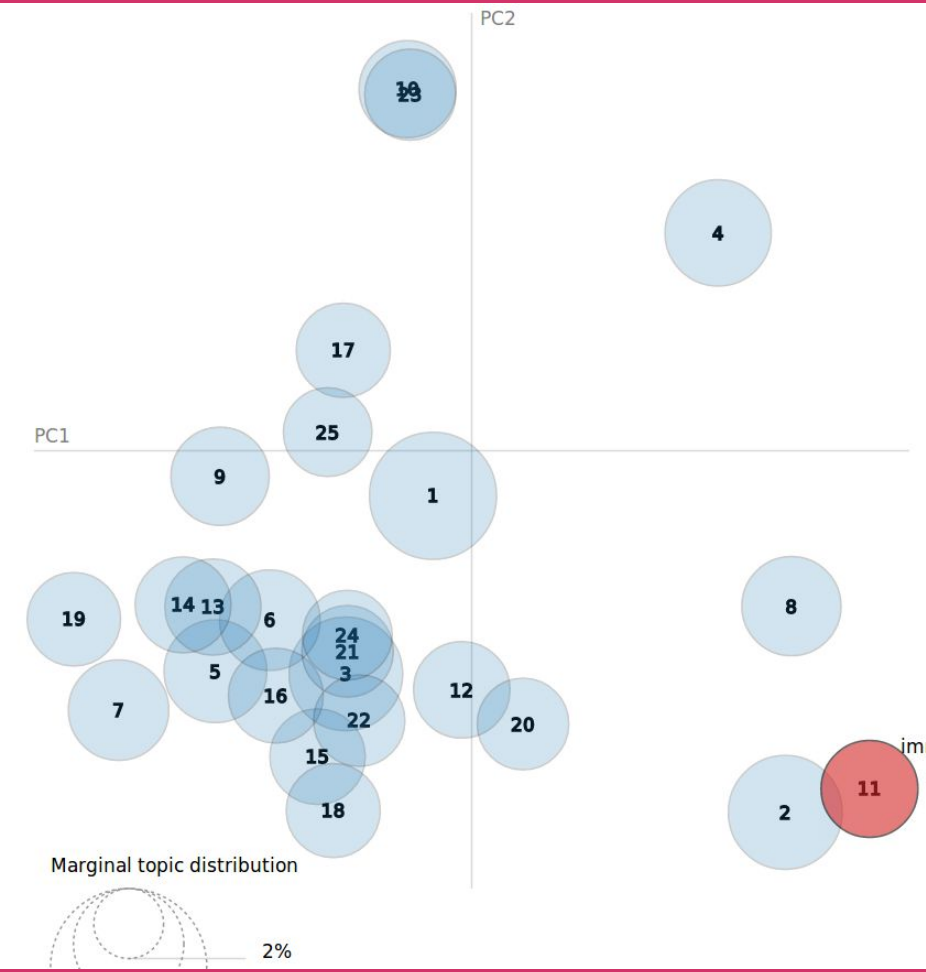
Avant de tenter déterminer les cas de violences conjugales à partir de cette information textuelle en mobilisant des techniques **de classification supervisée**, on commence par explorer les champs pour avoir une meilleure idée de leur contenu.

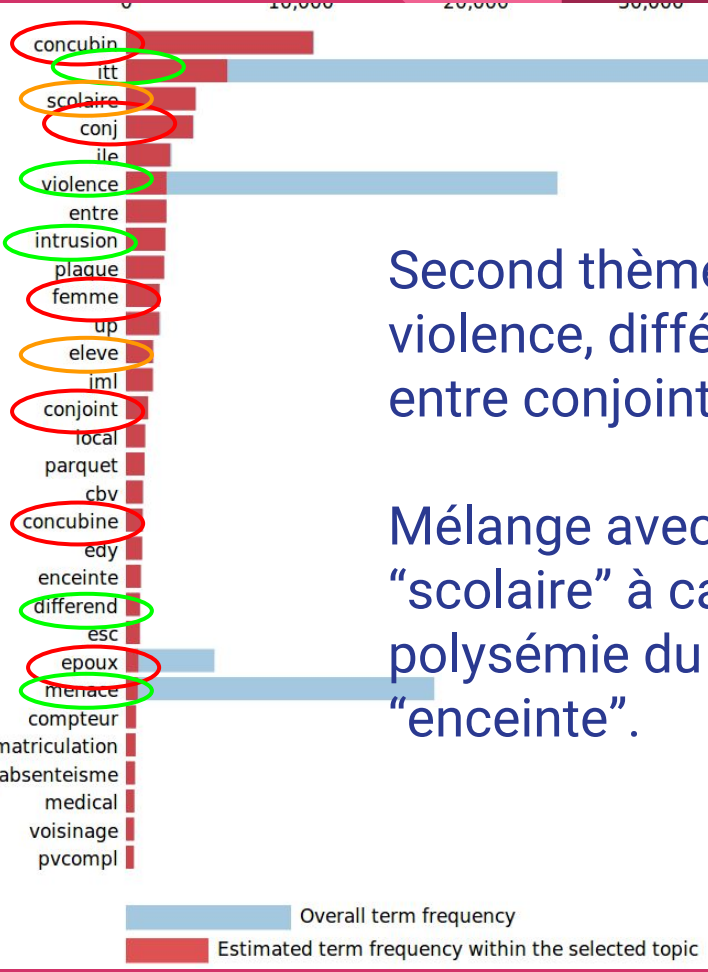
	vol	conj	coup	violence	menace	famille	malveillant	voiture
document 1		1			1	1	1	
document 2	1			1	2			1
document 5						1	1	1
document 6	1		1	1				1
document 7	1							
document 8			1		1			1
document 9			1		1	1		1
document 10		1		1	1		1	
document 13					1			
document 14								
document 15							1	





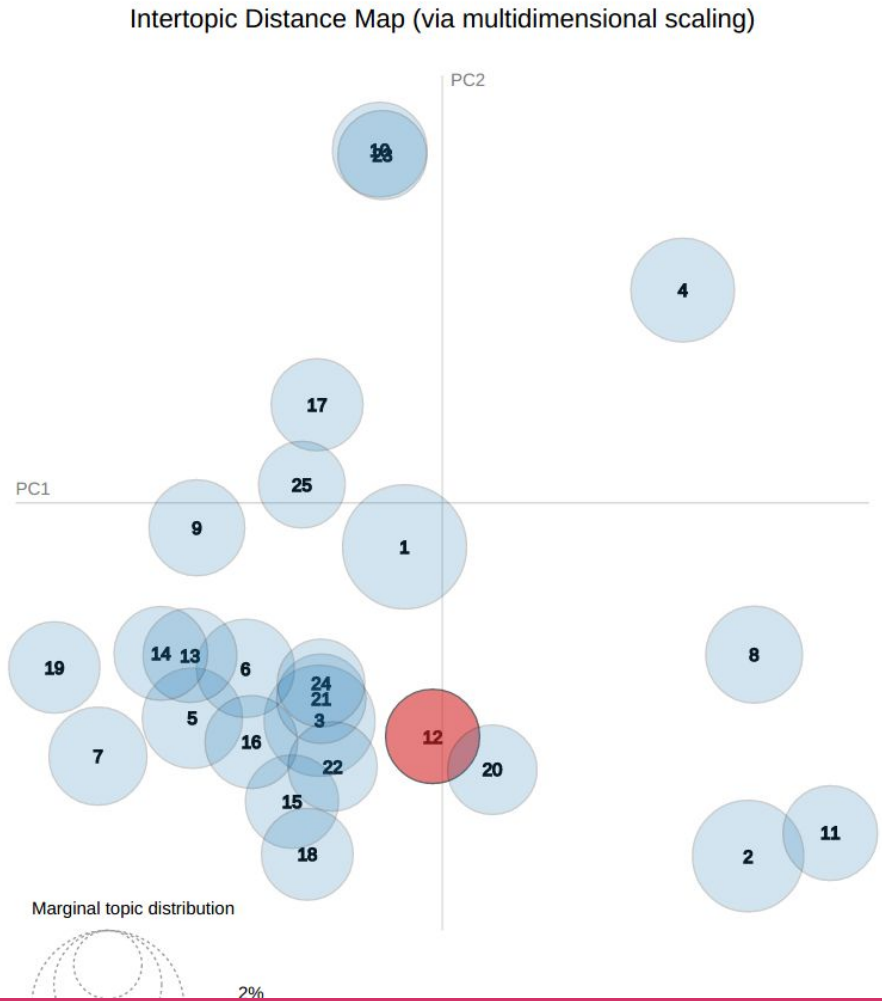
Premier thème lié : violence réciproque entre conjoints, contexte de drogue, alcool

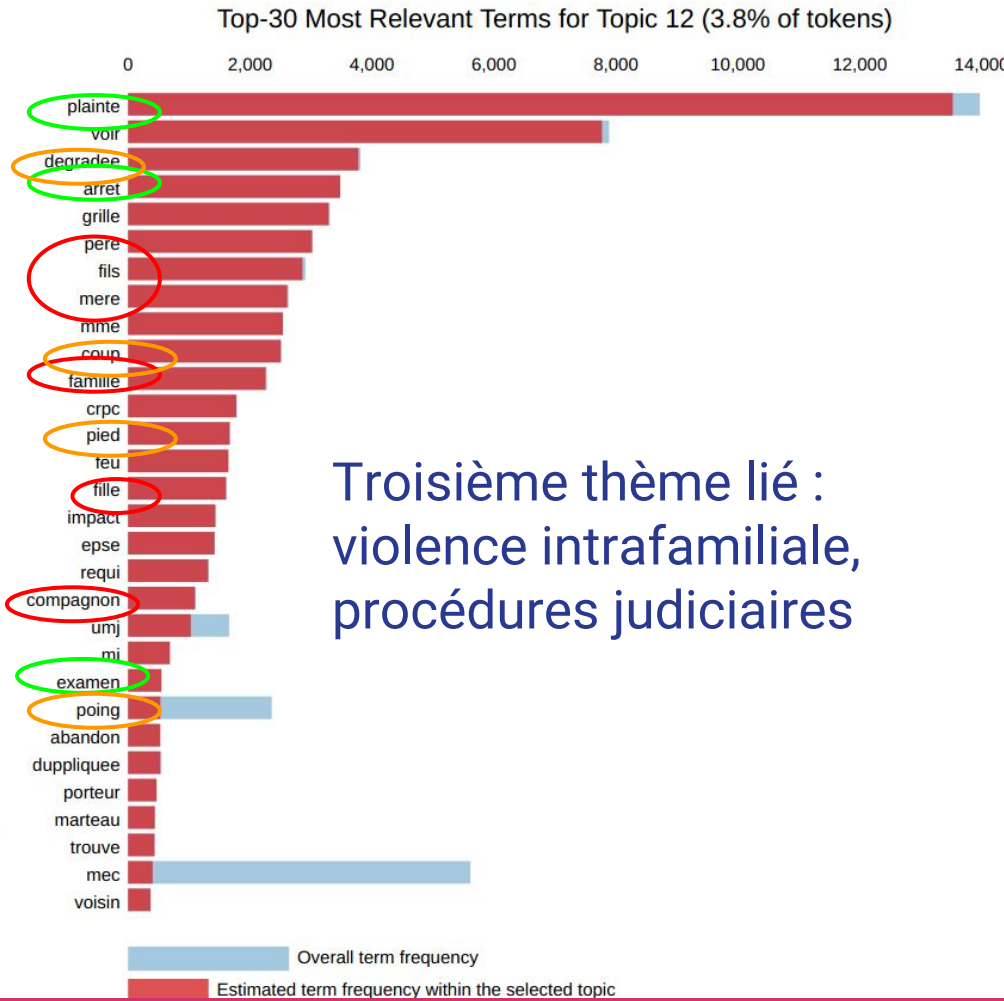




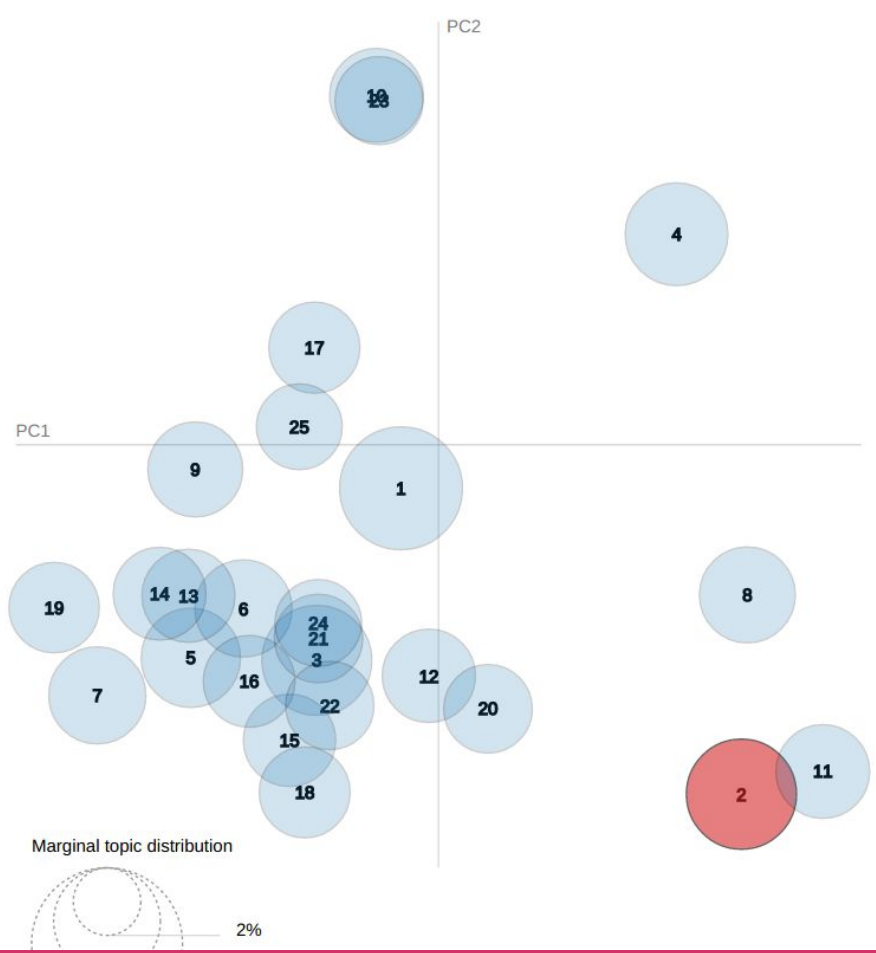
Second thème lié : violence, différend, entre conjoint, avec itt.

Mélange avec le thème “scolaire” à cause de la polysémie du terme “enceinte”.

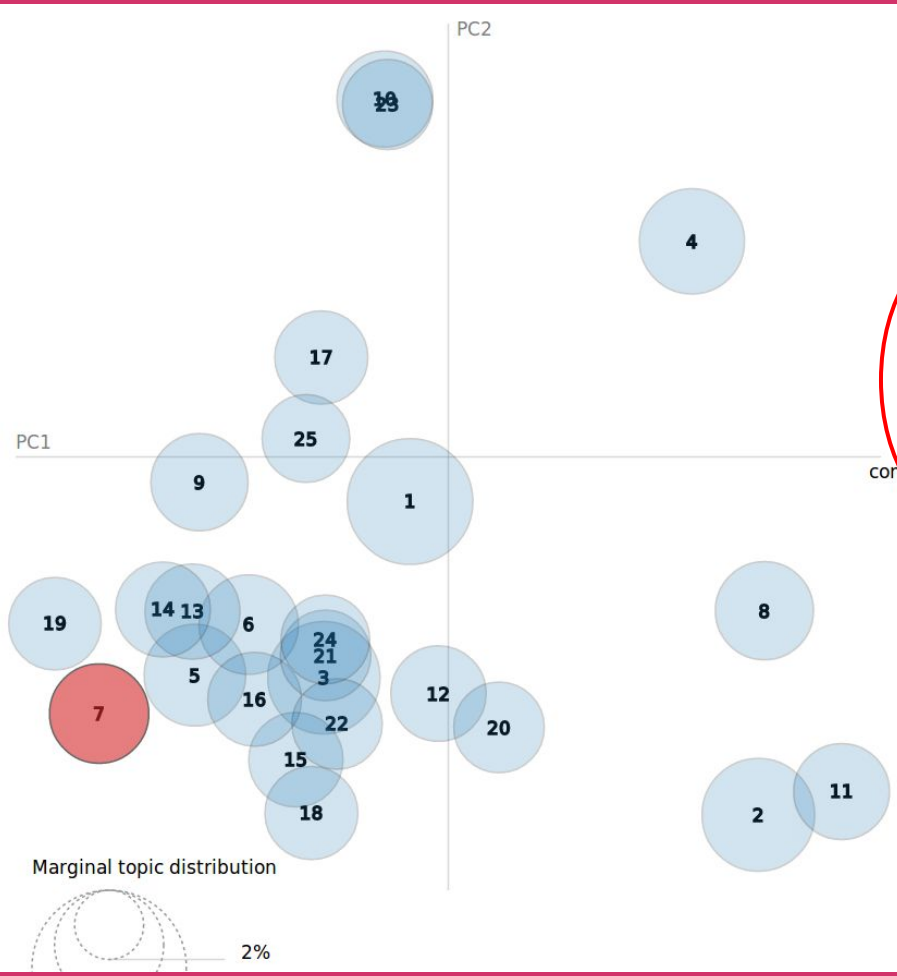


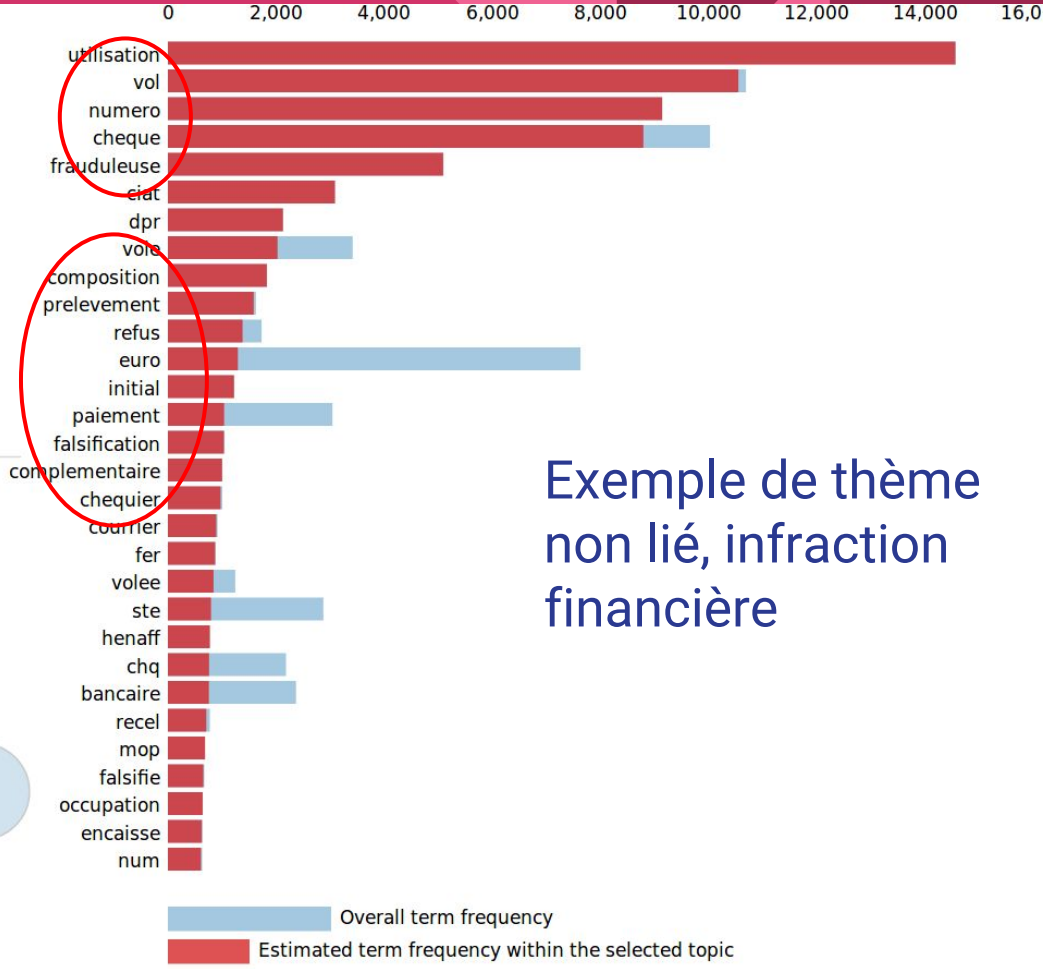


Troisième thème lié : violence intrafamiliale, procédures judiciaires

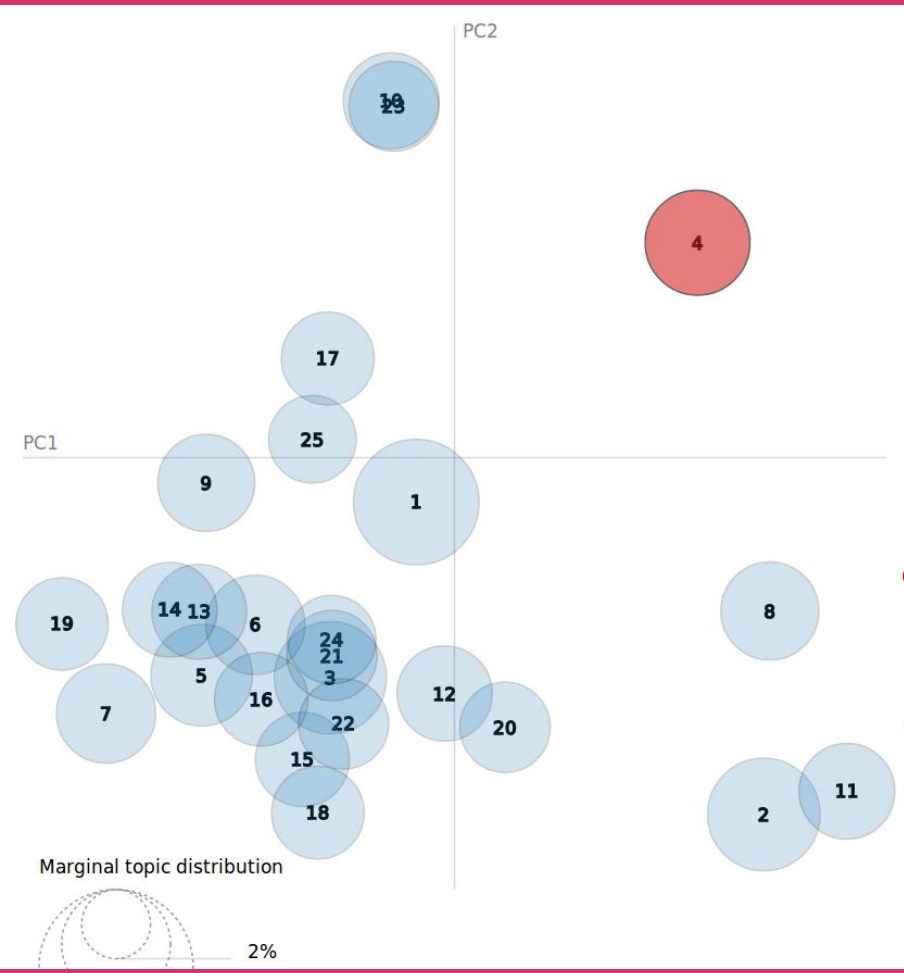


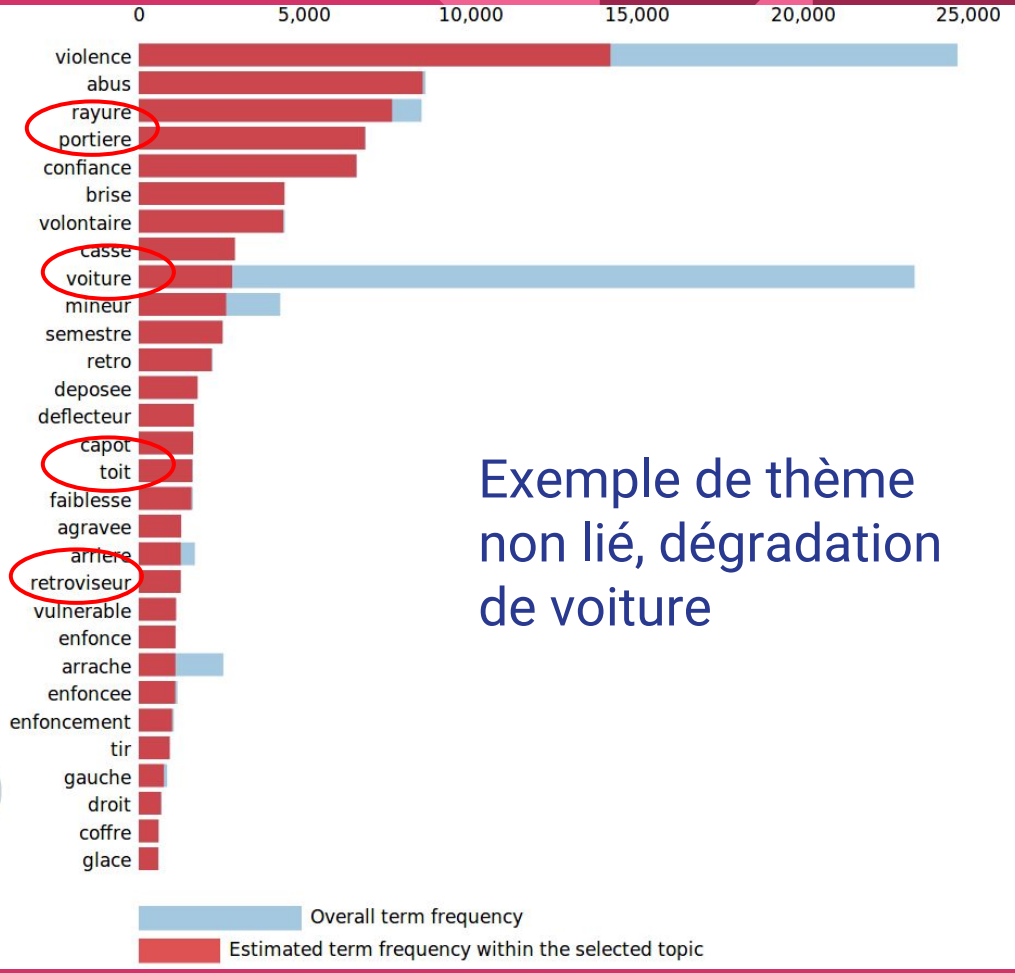
Exemple de thème non lié, mais sémantiquement proche : violence entraînant des itt



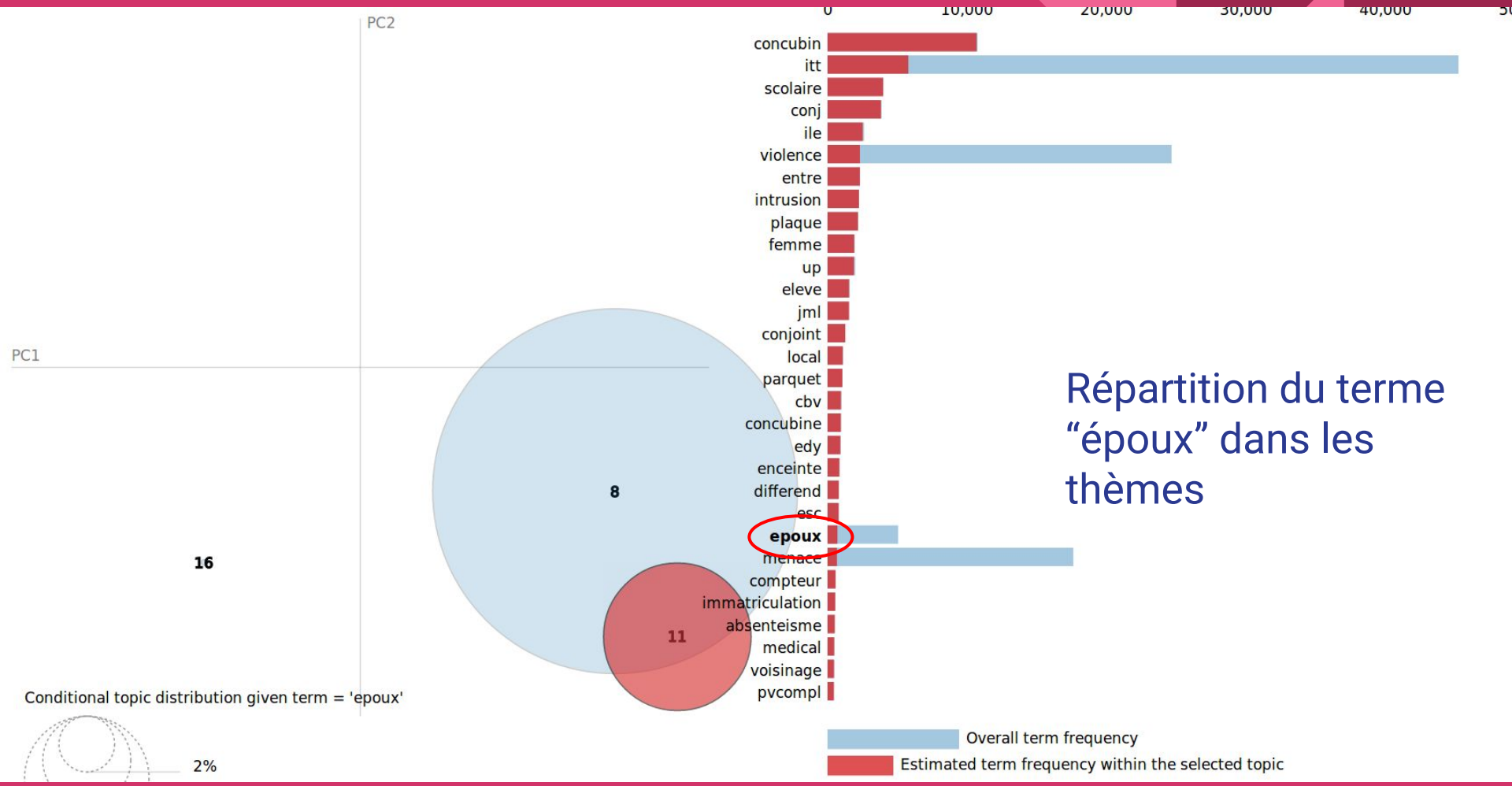


Exemple de thème non lié, infraction financière

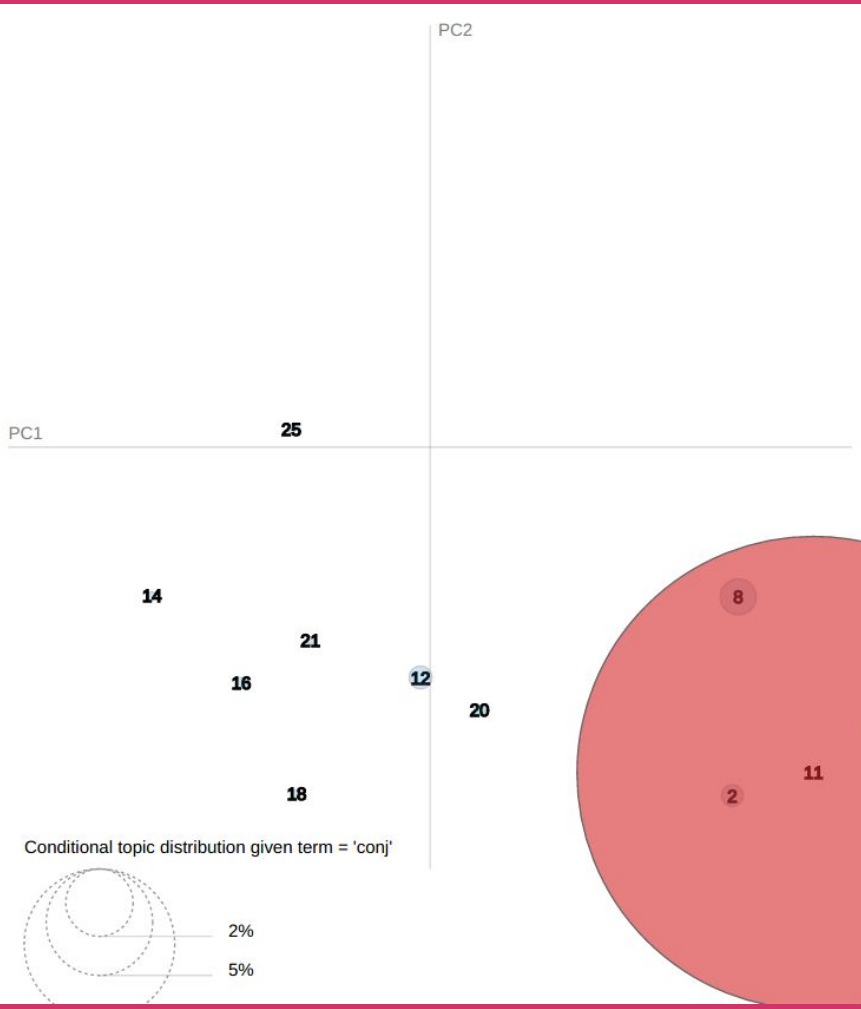


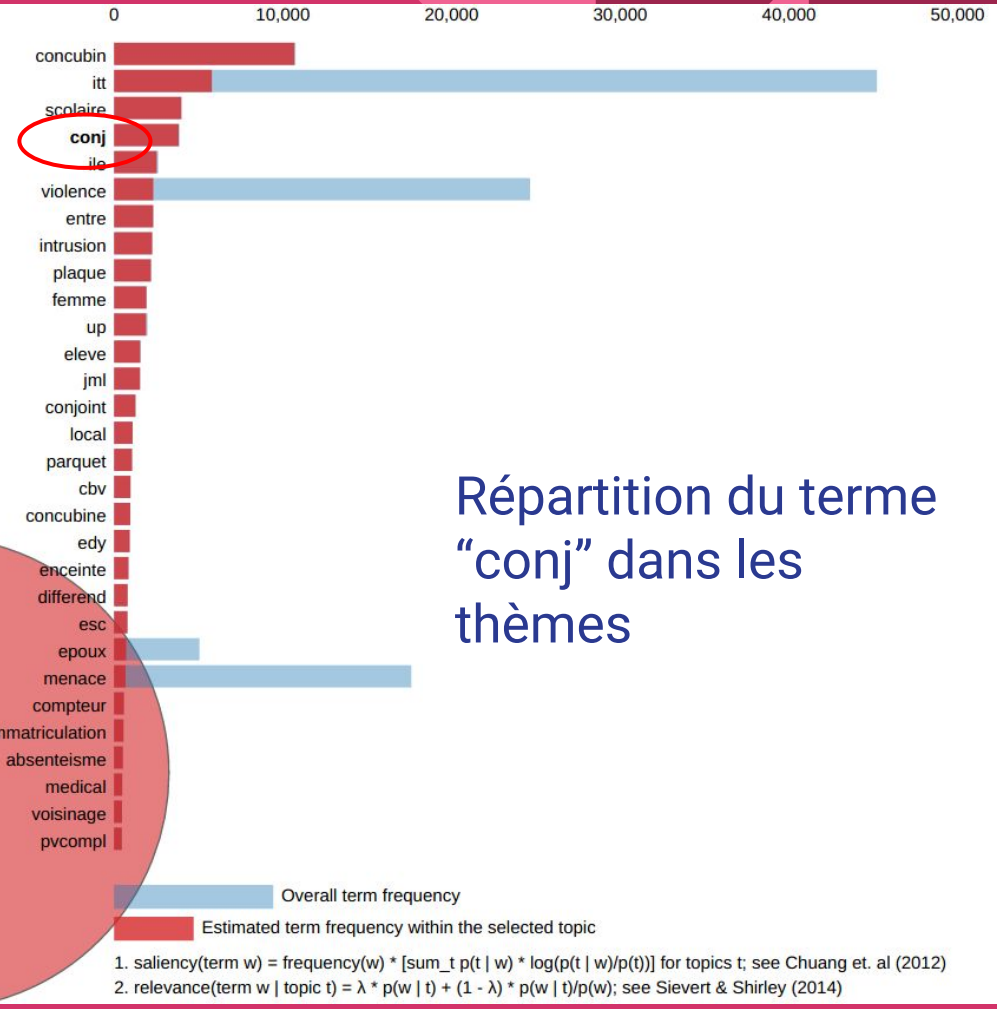


Exemple de thème non lié, dégradation de voiture

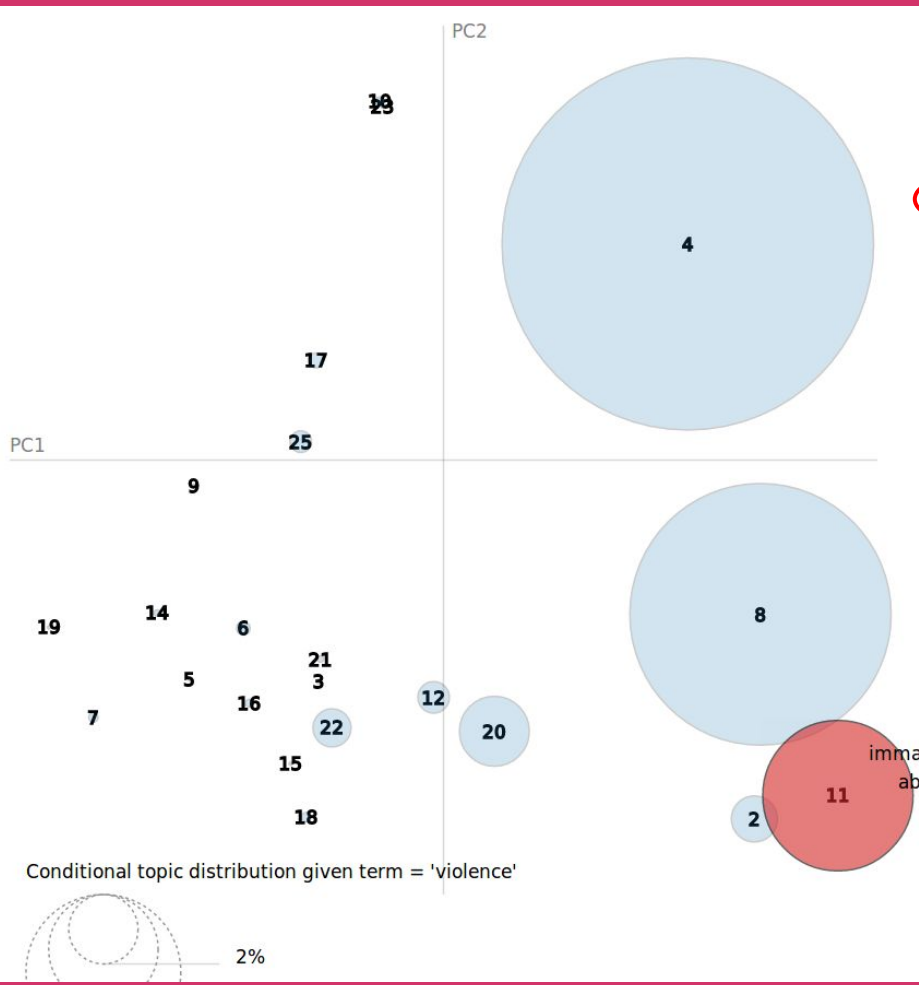


Répartition du terme “époux” dans les thèmes

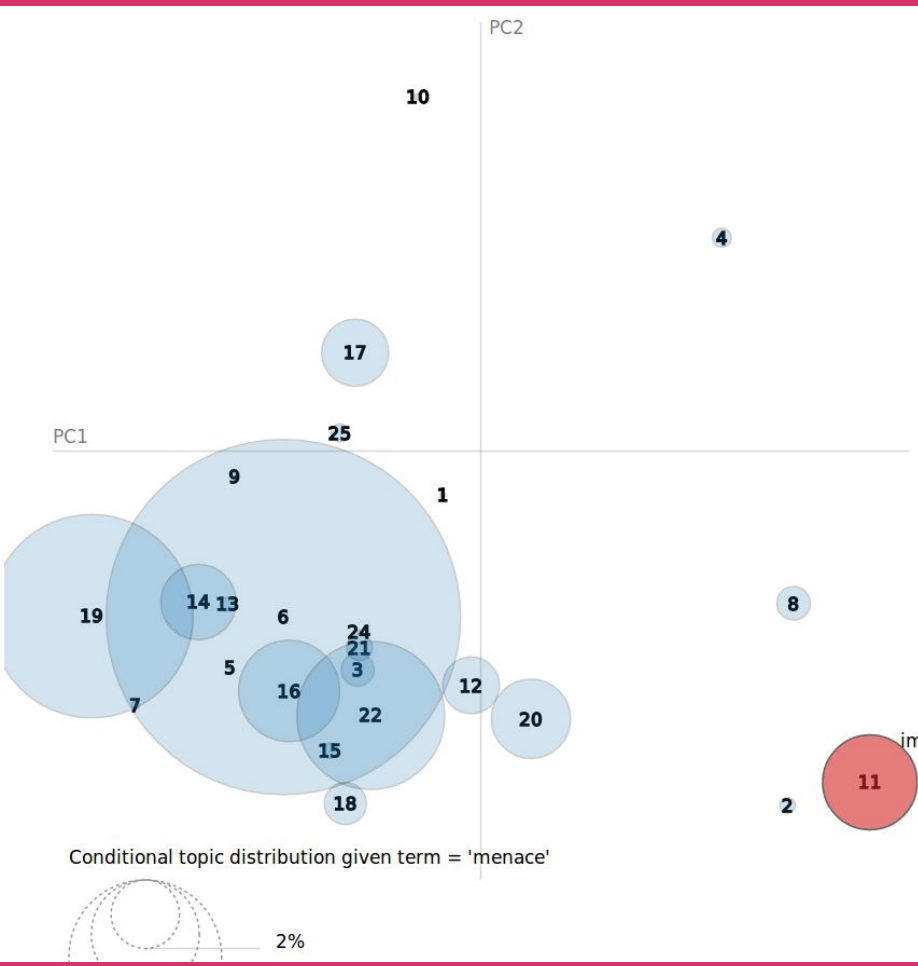




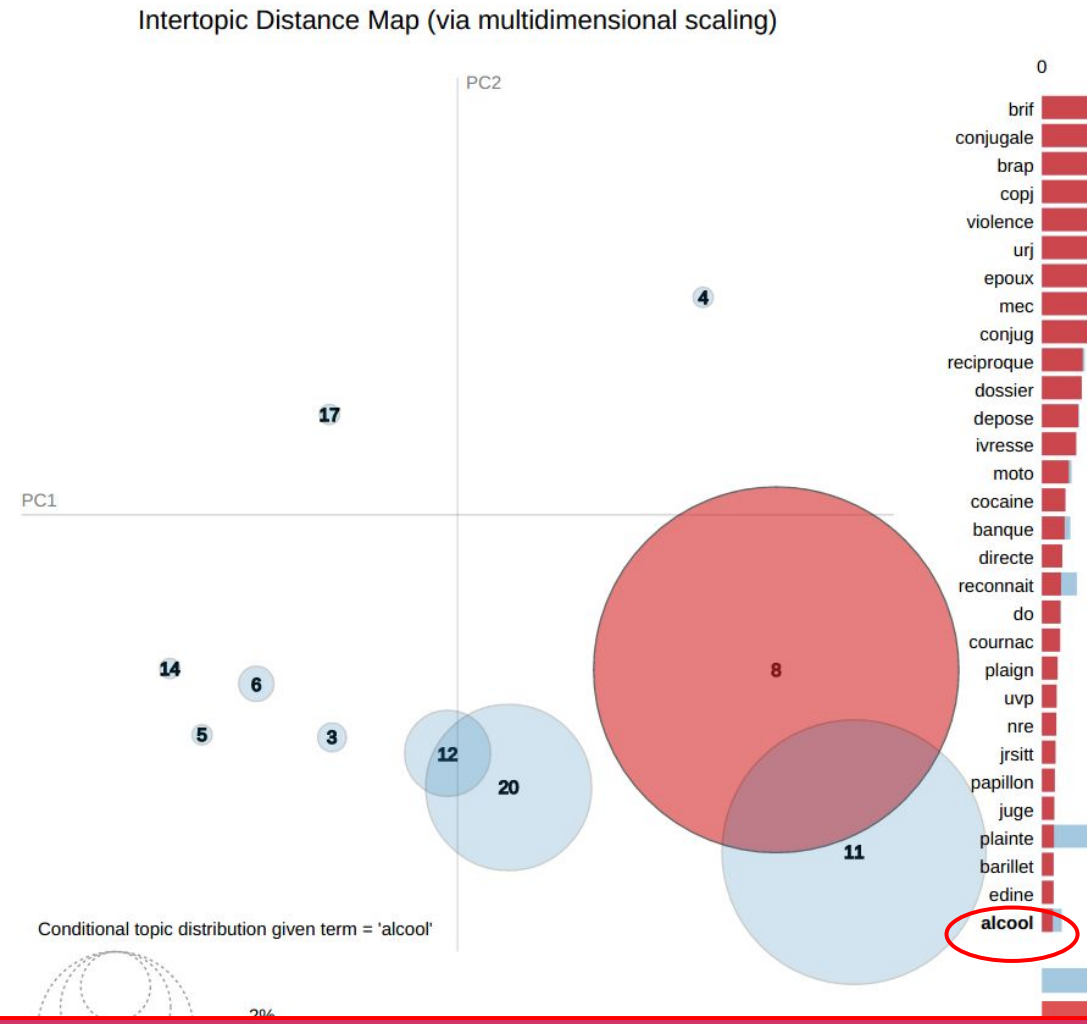
Répartition du terme “conj” dans les thèmes

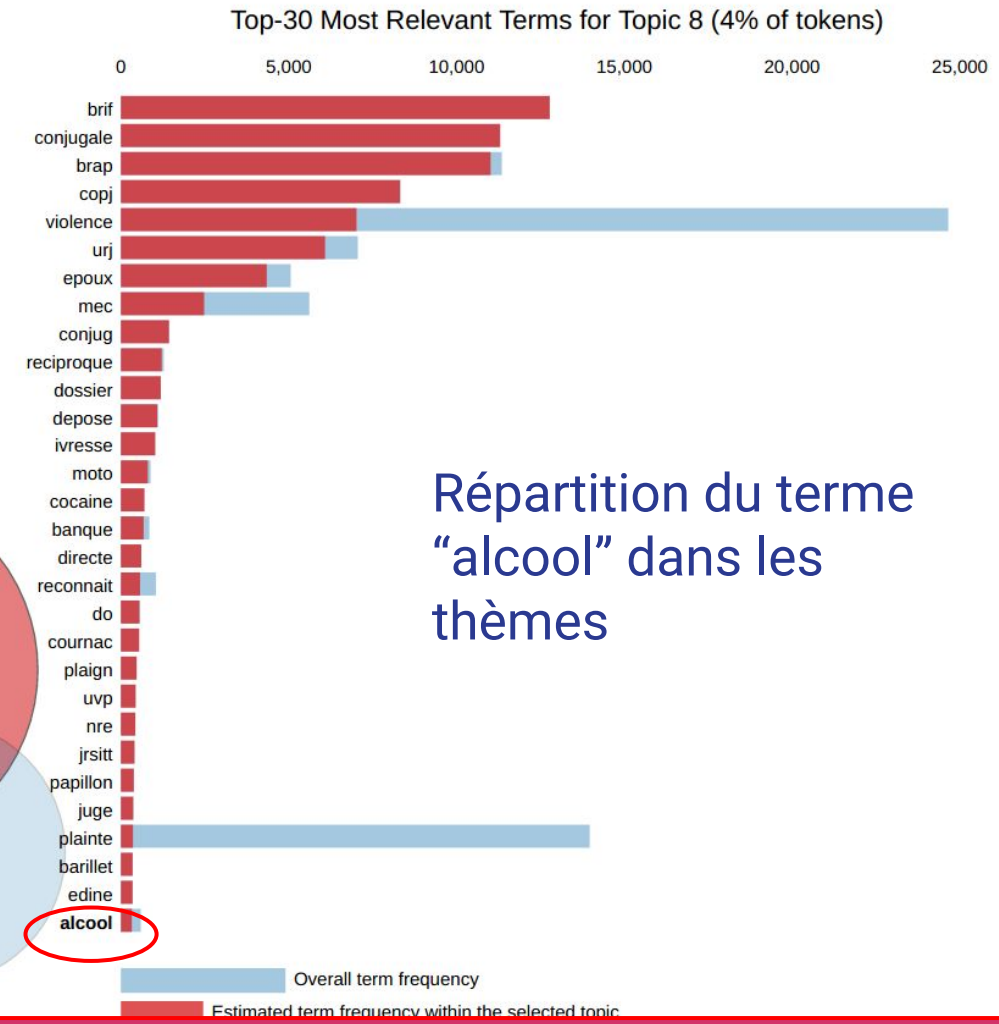


Répartition du terme “violence” dans les thèmes



Répartition du terme “menace” dans les thèmes





Répartition du terme “alcool” dans les thèmes

Remarque sur le clustering

Le clustering est rarement le but du clustering, on peut s'en servir à différentes fins :

- Soit en complément des **statistiques descriptives** pour explorer de façon plus avancée les données (ou des résultats comme dans l'exemple précédent sur les compétences recherchées dans les offres d'emploi)
- Soit en prétraitement d'une procédure supervisée, pour **réduire la dimension** (la LDA et la LSA, comme par ailleurs l'ACP, peuvent servir à réduire la dimension avant application d'un algorithme de classification, dans notre exemple cela nous sert à identifier un champ lexical dans le but d'établir des règles déterministes pour affecter un code 0 ou 1 aux champs textuels selon qu'ils indiquent un cas de violence familiale ou non).

Le clustering est sensible à l'échantillon, au prétraitement (normalisation, pondération pour le texte, les réseaux), aux variables retenues ou leurs transformées (feature engineering), tous ces paramètres doivent donc être choisis en lien avec le but poursuivi.

Clustering de données textuelles en grande dimension

Environnement informatique

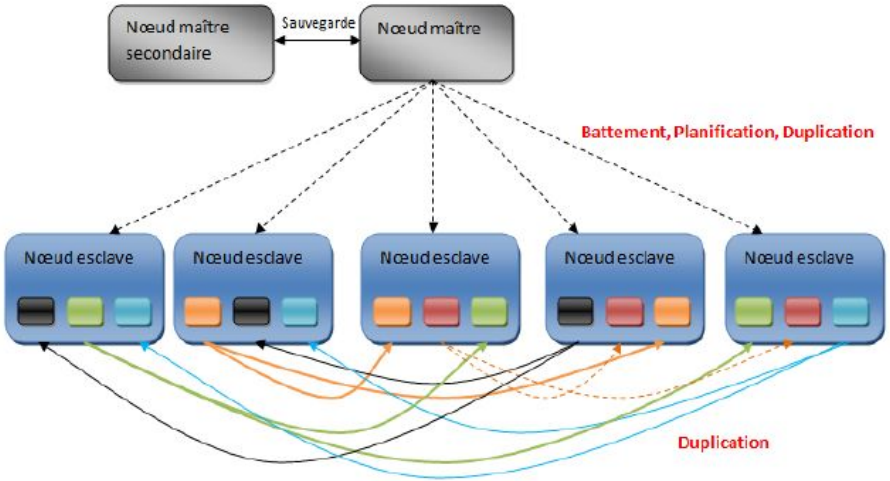
Architecture

À la base des systèmes dits « big data », il y a un principe central : la **distribution**, à la fois des **données** et des **traitements**, sur un ensemble de machines formant un cluster.

Le stockage des données brutes s'appuie sur un **système de fichiers distribués**. Dans les faits, les fichiers sont donc découpés en blocs de taille constante, répartis et répliqués de manière homogène sur l'ensemble des machines dédiées au stockage.

Cette répartition va permettre le traitement en parallèle des données et donc **un gain de temps proportionnel au nombre de machines utilisées**.

Architecture



Le traitement est transmis au **noeud maître** supervisant le cluster qui aura la charge de le transmettre à son tour aux différents nœuds de traitement mais également de s'assurer du bon déroulement des opérations.

En effectuant les opérations directement au niveau des nœuds **on gagne le temps de déplacement des données**.

La réplication permet en outre une **tolérance à la panne** d'une ou plusieurs machines, d'autant plus probable que le nombre de machines est grand. Les architectures « big data » sont donc conçues pour être **robustes**.

Architecture

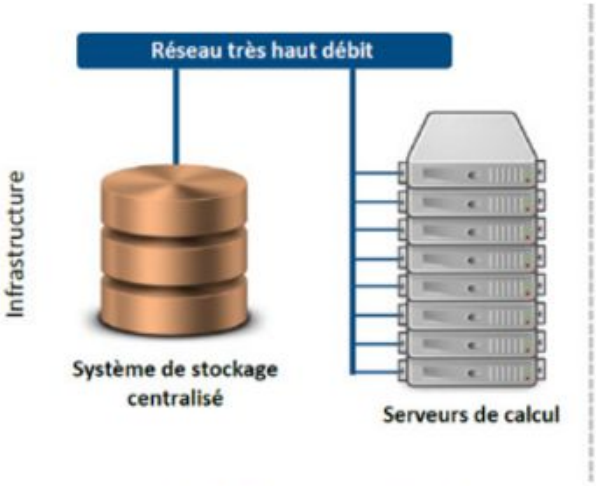


Fig. 1 – Cluster de calcul HPC

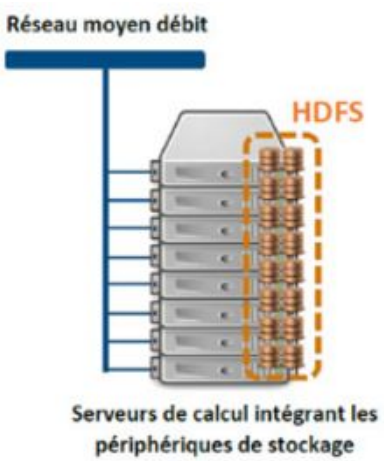


Fig. 2 – Cluster Hadoop

Cette technologie ne doit pas être confondue avec la parallélisation d'un calcul sur les cœurs d'un micro-processeur par exemple, au sens où dans ce dernier cas **chacun des nœuds impliqués dans le calcul a accès à l'intégralité des données** et c'est le calcul seul qui est distribué entre différentes unités de calcul (par exemple différentes itérations indépendantes d'un même algorithme).

Architecture

Cette technologie, plus ancienne (apparue dans les années 60 et popularisée dans les années 90), est caractérisée par un grand nombre de serveurs dont les processeurs sont mobilisés pour effectuer les calculs des utilisateurs (calculs pouvant être très complexes), un **système de stockage de données centralisé** permettant la sollicitation simultanée par de nombreux serveurs de calcul et un **réseau très haut débit** reliant les serveurs entre eux ainsi qu'au système de stockage central.

On parle de cluster de calcul ou HPC (**High Performance Computing**, calcul haute performance).

Architecture

Le choix entre ces deux architectures repose sur des compromis : si l'on souhaite appliquer des traitements simples ou parallélisables sur un énorme volume de données, il **sera moins coûteux de laisser les données en place et d'aller effectuer le traitement localement** (les fichiers distribués sur des machines distinctes avec leurs propres capacités de calcul), pour ne déplacer vers le serveur maître qu'un résultat agrégé.

Faire transiter cette énorme quantité de données vers les serveurs de calcul entraînerait d'importants coûts en termes d'infrastructures de réseaux.

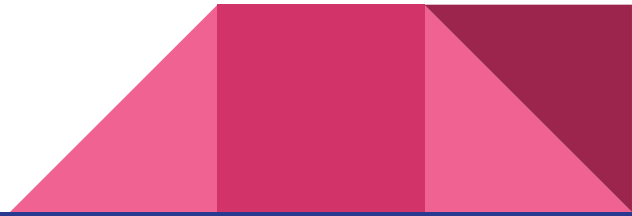
A l'inverse, si les données ne sont pas suffisamment importantes, ou que les traitements sont beaucoup plus sophistiqués, les coûts fixes d'une architecture Big Data pourrait la rendre moins attrayante.

Remarque : les architectures distribuées pour les données massives tendent vers une **technologie hybride mettant en commun des capacités de stockages et de calcul**.

Le passage à l’échelle - scalability

L’architecture Big Data repose sur un principe flexible puisqu’en cas d’accroissement sensible des données, il suffit de rajouter des machines, ni l’architecture informatique dans sa globalité ni les traitements ne doivent être refondus : on dit que le système **passe à l’échelle**.

Hadoop est la solution historique de **stockage (avec HDFS) et de traitement (MapReduce)** des Big Data. Plus récemment est apparu un nouveau challenger, Spark, qui est devenu la solution de référence pour le traitement de données massives.



Map Reduce

MapReduce est la première implémentation pour les big data du principe de **parallélisation des traitements appliquée aux fichiers distribués**. Elle repose sur deux fonctions principales, **map** et **reduce**, qui sont appliquées parfois à de multiples reprises.

La première décrit une **transformation que l'on applique aux valeurs d'une collection de données au format clé / valeur** ; la deuxième applique une **opération à toutes les valeurs d'une même clé**.

Prenons l'exemple du dénombrement des nucléotides composant un brin d'ADN : «AGTCGGGGCT». L'objet en entrée est donc une séquence, et on souhaite obtenir en sortie une table avec les différents symboles et le nombre d'occurrences de chacun d'entre eux dans la séquence initiale : (A,1), (C,2), (G,5), (T,2).

Map Reduce

La démarche « naturelle » consiste à prendre le premier symbole de la séquence, de noter dans la table de sortie son nom et d'initialiser un compteur à 1, puis de revenir à la séquence, d'identifier le second et s'il correspond au premier, d'incrémenter le compteur correspondant…

Cette démarche est assez simple à décrire, mais demande en pratique de **parcourir tout ou partie de la séquence à chaque itération**.

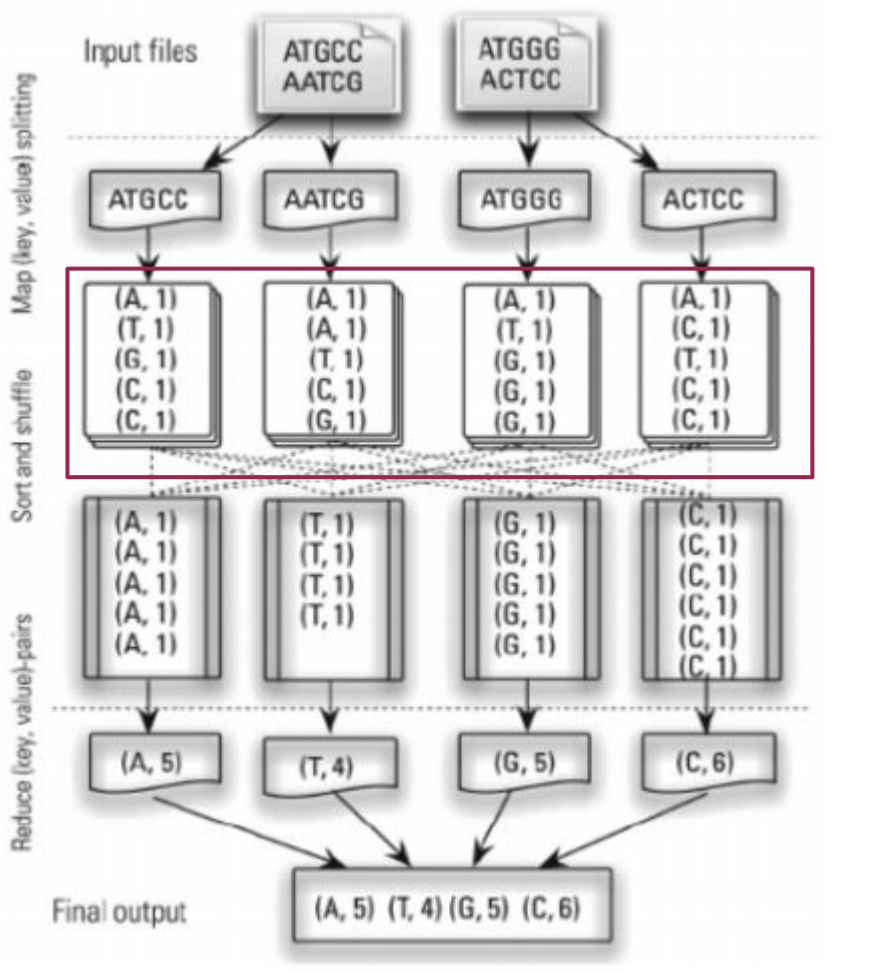
Cela n'est pas problématique si la séquence est aussi élémentaire que celle prise dans cet exemple… mais le devient si elle correspond au génome humain, constitué de quelques milliards de nucléotides.

Map Reduce

La même opération réalisée en Map Reduce, correspond aux opérations suivantes :

- lors du chargement des données, la séquence a été coupée en plusieurs sous-séquences chargées sur différentes machines ou nœuds

- la première étape (« Map ») consiste alors à appliquer au niveau de chaque noeud et pour chaque élément une fonction dont le résultat est un couple constitué d'une clé et d'une valeur. Ici, la **clé correspond au symbole lu**, et **la valeur à l'occurrence 1** : on a donc une liste de couples (A,1), (G,1), (T,1), (A,1),..

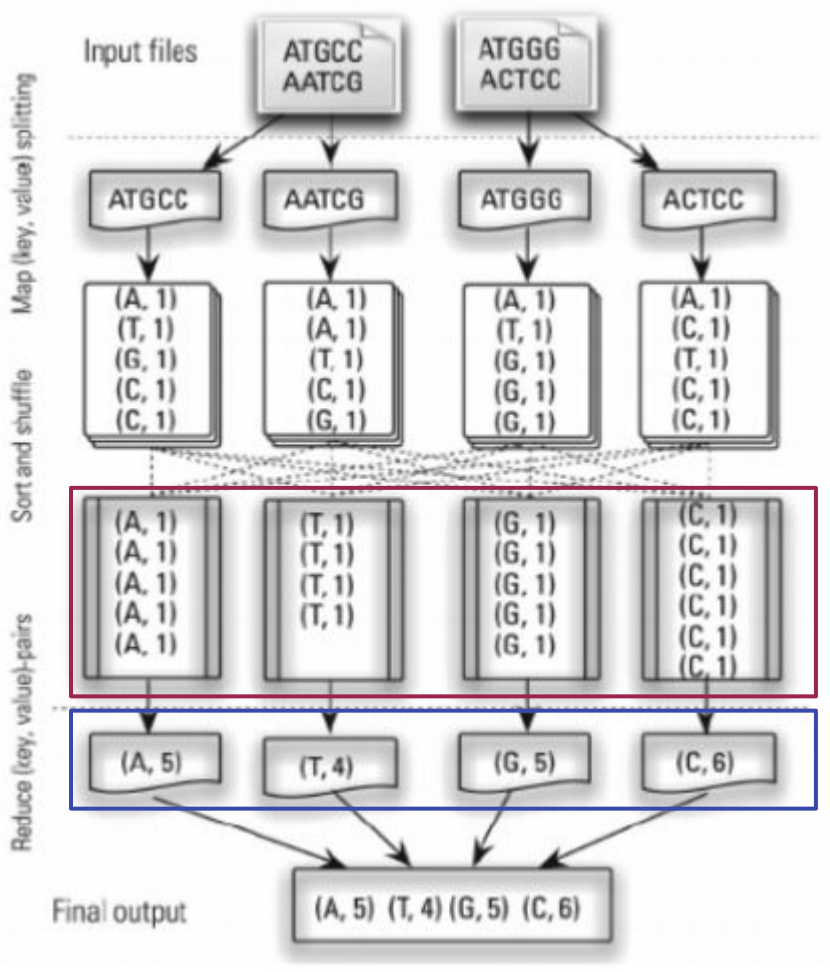


Map Reduce

- Dans une deuxième étape de triage («**shuffling and sorting**»), on **rassemble sur un même nœud toutes les paires correspondant à une même clé**.

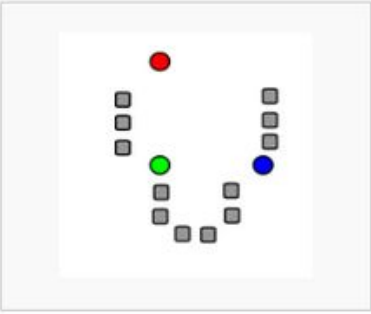
- Finalement, on applique une fonction d'**agrégation** (« **Reduce** ») qui combine (ici on somme tout simplement) les valeurs correspondantes à chaque clé.

On aboutit dans cet exemple à quatre paires constituées d'un symbole et du nombre d'occurrences de ce symbole sur l'ensemble de la séquence.

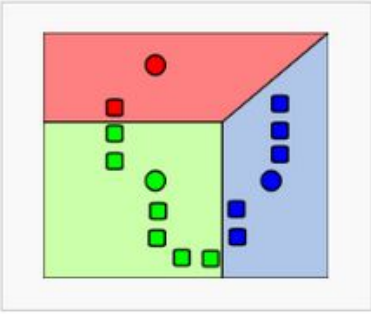


Algorithmes

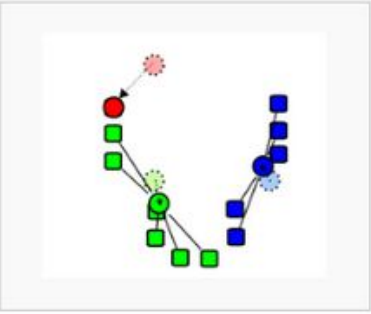
K-means



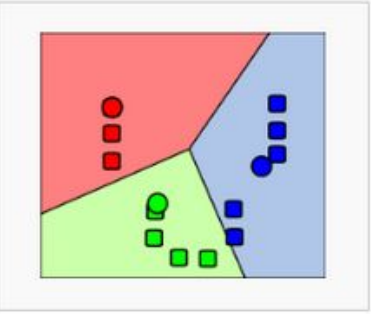
1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).



2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.



3. The centroid of each of the *k* clusters becomes the new mean.



4. Steps 2 and 3 are repeated until convergence has been reached.

Les K-means

L’étape la plus intensive en calculs est l’étape de **calcul des distances**. A chaque étape on doit calculer **nxk** distances (entre chaque observation et chacun des k centroïdes).

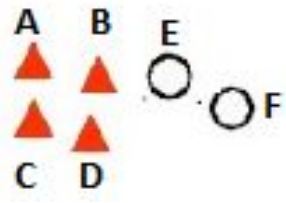
Le calcul d’un objet aux centroïdes est **indépendant** du calcul d’un autre objet aux centroïdes.

Par conséquent les calculs des distances d’objets différents aux centres peuvent être **parallélisés**.

Référence : Parallel K-Means Clustering Based on MapReduce, Weizhong Zhao, Huifang Ma, Quing He, 2009

Les k-means distribués

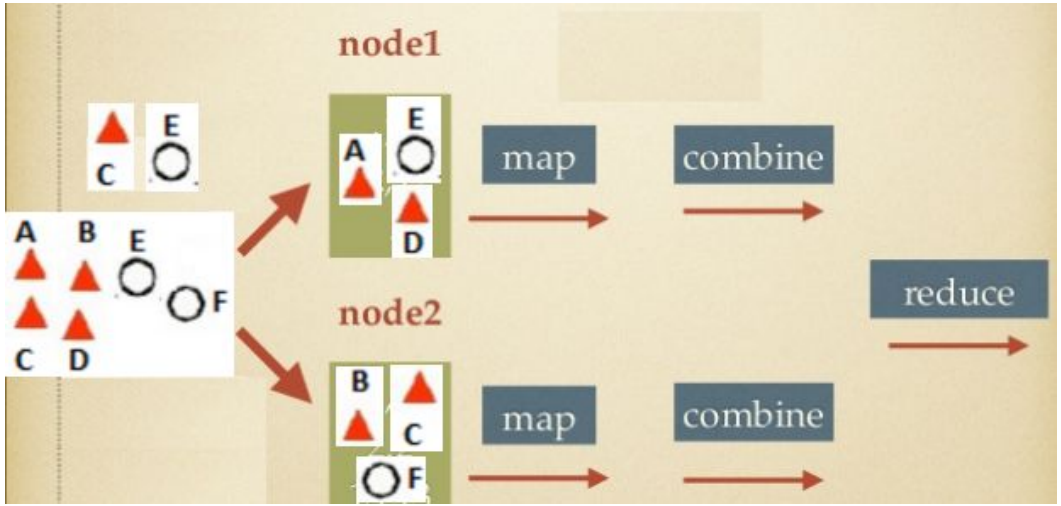
Admettons qu'on ait 6 objets et qu'on cherche à retrouver 2 regroupements : ABCD d'un côté et EF de l'autre.



On fixe 2 centroïdes aléatoirement au départ, par exemple C et E que l'on stocke avec les labels C1 et C2.

Au départ on va répartir aléatoirement les 6 observations sur 2 noeuds, par exemple AED et BCF.

Les k-means distribués



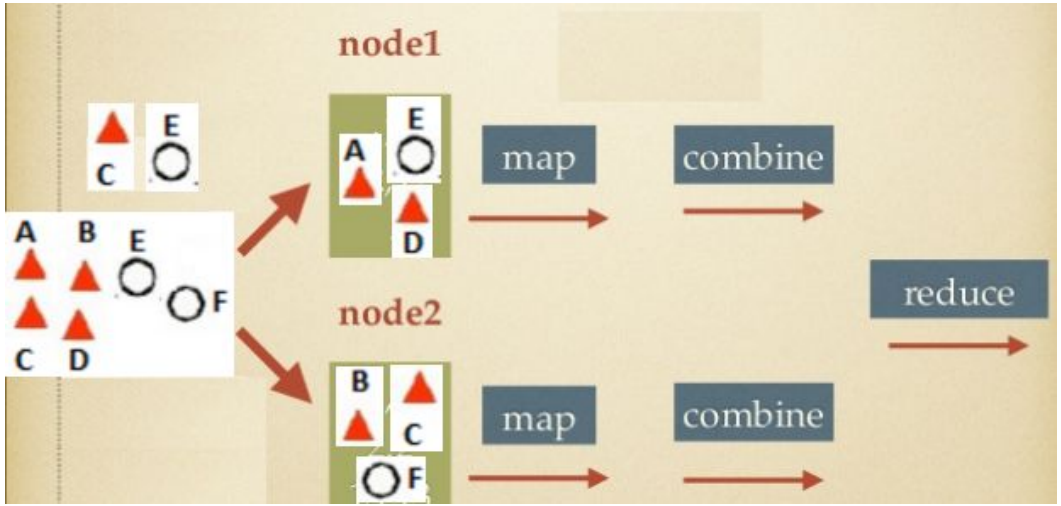
L'opération **map** consiste à calculer pour chaque observation d'un noeud la distance aux centres (ici C et E, labellisés C1 et C2), et donc **assigner à chaque observation le centre le plus proche**.

L'output de l'opération map est un couple clé valeur avec comme clé le centre et comme valeur l'étiquette de l'observation et ses coordonnées,

pour le noeud 1 :

A -> (C1, {A,(xA,yA)}) ,
E -> (C2, {E,(xE,yE)}),
D -> (C1, {D,(xD,yD)})

Les k-means distribués



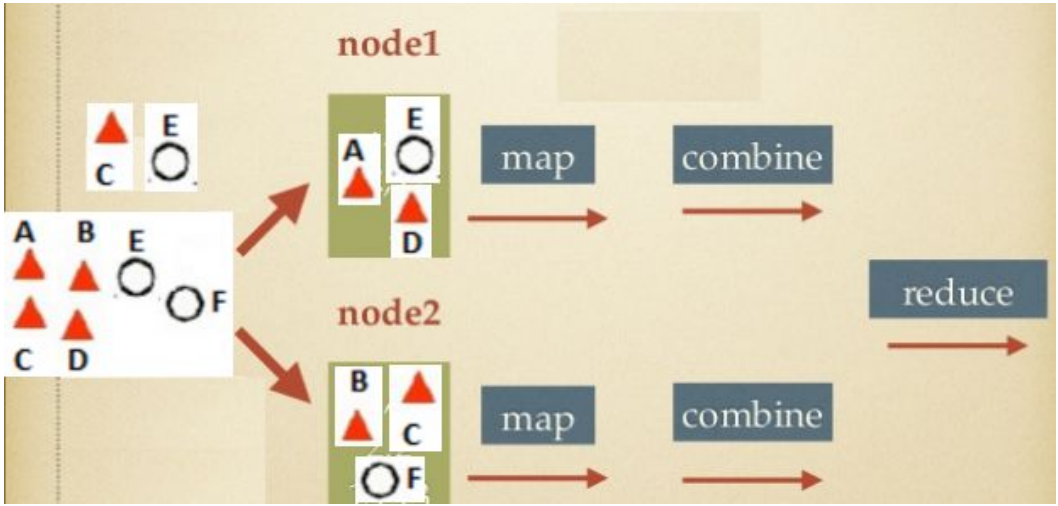
L'opération **combine** vise à réunir les couples admettant la même clé.

Donc dans le noeud 1, les couples (C1,A) et (C1,D) par exemple, ce qui fournit le couple clé valeur :

(C1, {(xA+xD,yA+yD),{A,D,2}}) et (C2, {(xE,yE),{E,1}}) pour le noeud 1.

La valeur comprend la somme des coordonnées des observations allouées à ce centre (utile pour la réactualisation du calcul du centroïde), les étiquettes et le nombre d'observations.

Les k-means distribués



L'opération **reduce** vise à réunir les couples admettant la même clé répartis sur des noeuds distincts.

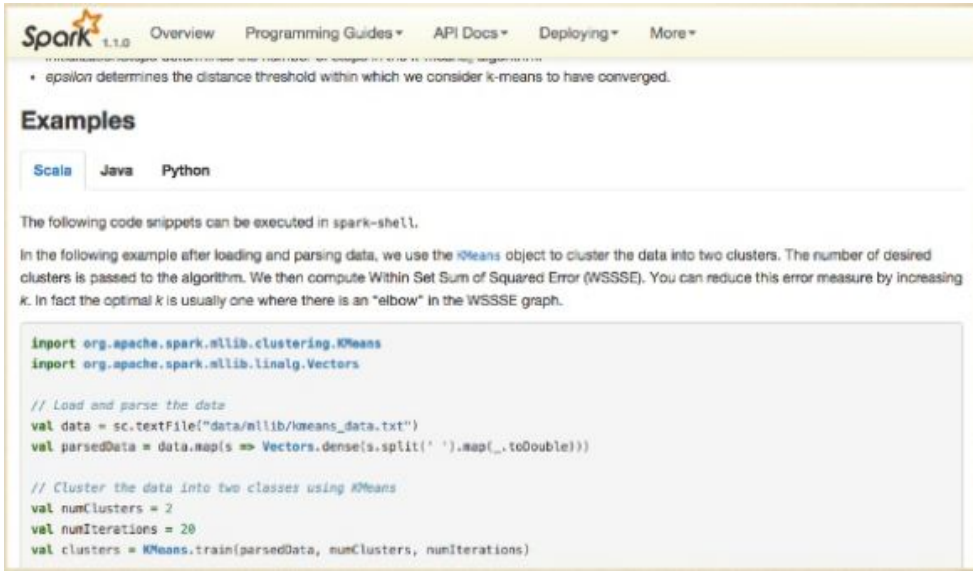
Après l'opération combine on a (C1, {(xA+xD,yA+yD),{A,D,2}}) pour le noeud 1 et (C1, {(xC,yC),{C,1}}) pour le noeud 2.

Ici l'opération de réduction consiste à sommer les coordonnées (xA+xD)+xC, (yA+yD)+yC d'une part, et les fréquences d'autre part 2+1, et de diviser les sommes des coordonnées par les fréquences :

(C1, {((xA+xD+xC)/3,(yA+yD+yC)/3),{A,D,C}})

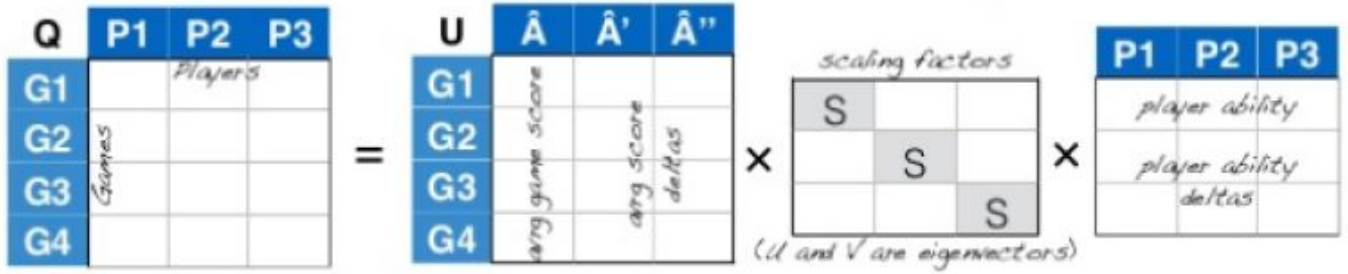
Ces nouvelles coordonnées sont celles du nouveau centroïde, représentant le cluster des observations (A,D,C).

Les k-means distribués



Les 3 opérations map, combine et reduce se succèdent jusqu'à convergence.

Latent semantic analysis



- Inverted index = doc. eigenvectors × singular values × term eigenvectors

Latent semantic analysis

On décompose la matrice documents termes (DTM, mxn) en une matrice U mxn, S diagonale nxn et V nxn : DTM = USV’ (**décomposition en valeurs singulières**)

On fait une hypothèse de **faible rang** (k) sur la matrice diagonale, on considère que l’essentiel de l’information peut être résumé par un petit nombre de “directions sémantiques principales”.

On détermine le rang k<n (nombre de termes), par une règle du coude sur la distance de Frobenius entre la matrice de départ et son approximation, ou un pourcentage de variance expliquée.

k peut être interprété comme le **nombre de concepts latents**. Et DTM peut être approchée par le produit de U’ mxk, S’ kxk et V’ kxn.

Latent semantic analysis distribuée

Si le nombre de termes est raisonnable, on va obtenir la décomposition en valeurs singulières en passant par la détermination des valeurs propres et vecteurs propres de la matrice carrée : **DTM' x DTM**, **de taille nxn, où n le nombre de termes peut être très inférieur à m le nombre de documents** (sur des jeux de données comme Wikipédia par exemple).

Il existe une **relation simple entre les valeurs singulières de A et les valeurs propres de 'AA** (les secondes sont le carré des premières), et entre vecteurs singuliers de A et vecteurs propres de 'AA et de AA'.

On peut utiliser l'algorithme de la bibliothèque ARPACK pour calculer les valeurs et vecteurs propres les plus importants. **La multiplication d'un vecteur par la matrice DTM est la seule opération de grande ampleur** (et peut être distribuée). On fera référence à DTM' x DTM par la matrice A dans la suite.

Algorithme - principe de base

L'algorithme de **Lanczos** permet de se ramener à la matrice H tridiagonale et semblable à A (donc admettant les mêmes valeurs propres : $H = U^{-1} A U$).

On sait qu'il existe une matrice de cette forme lorsque A est symétrique.

En réalité on construit **une suite de matrices H_m dont les valeurs propres convergent rapidement vers celles de A et sont facilement calculables** (les plus grandes convergent le plus vite, on n'a pas forcément besoin de tout le spectre en général).

$$\begin{pmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & \beta_3 & \alpha_3 & \beta_4 & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \beta_m \\ & & & & \beta_m & \alpha_m \end{pmatrix}$$

Algorithme - principe de base

Soit u_1 un vecteur initial de norme unité, on crée, pour k=2 à m la suite (u_k) orthonormée telle que : $\beta_{k+1}u_{k+1} = Au_k - \alpha_k u_k - \beta_k u_{k-1}$

où :

- $v_k = Au_k - \beta_k u_{k-1}$,
- $\alpha_k = (v_k, u_k)$,
- $v_k = v_k - \alpha_k u_k$
- $\beta_{k+1} = ||v_k||$
- $u_{k+1} = v_k/\beta_{k+1}$

$$H_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & \beta_3 & \alpha_3 & \beta_4 & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \beta_m \\ & & & & \beta_m & \alpha_m \end{pmatrix}$$

On a $U_m^{-1}AU_m = H_m$, avec **H_m tridiagonale de termes α_k et β_k et dont les valeurs propres convergent rapidement vers celles de A**

Algorithme - principe de base

La méthode dite de **décomposition QR** permet de calculer les valeurs propres d'une matrice A (il existe différentes approches pour calculer Q, R non uniques), et est d'autant plus efficace sur la matrice de Hessenberg calculée précédemment.

On décompose A = QR avec Q **orthogonale** et R **triangulaire supérieure**.

Il vient $RQ = Q^{-1}AQ$, **RQ est donc semblable à A** et a donc les mêmes valeurs propres.

On opère la décomposition QR de RQ , ce qui fournit une nouvelle matrice semblable à A : RQ = Q'R' = A. Il vient : **R'Q' semblable à A également**.

En itérant le processus on génère **une suite de matrices semblables à A**.

Algorithme - principe de base

On note $A^* = \lim R_k Q_k = \lim R_k$ en montrant que $\lim Q_k = I$

La suite A_k converge donc vers une matrice A^* **triangulaire supérieure** (et semblable à A) dont les **éléments diagonaux sont les valeurs propres** de A.

Algorithme

$A^{(0)} = A$

Décomposition QR de $A^{(0)} \Rightarrow A^{(0)} = Q^{(0)} R^{(0)}$

$A^{(1)} = R^{(0)} Q^{(0)}$

Décomposition QR de $A^{(1)} \Rightarrow A^{(1)} = Q^{(1)} R^{(1)}$

$A^{(2)} = R^{(1)} Q^{(1)}$

.....

Décomposition QR de $A^{(k-1)} \Rightarrow A^{(k-1)} = Q^{(k-1)} R^{(k-1)}$

$A^{(k)} = R^{(k-1)} Q^{(k-1)}$

.....

Si la limite existe,

$A^{(\infty)} = \lim_{k \to \infty} A^{(k)}$

est triangulaire supérieure et semblable à A.

(les valeurs propres sont ses éléments diagonaux)

Algorithme - principe de base

$$A = \begin{pmatrix} -183 & -48.4064584445 & -20.5079721005 & -31.1967604278 \\ 692.8001154734 & 183.1579425466 & 86.1545159544 & 100.4673262120 \\ 0 & 0.3272285441 & -11.7853328556 & 23.9080122256 \\ 0 & 0 & -5.4857473093 & 11.8273903090 \end{pmatrix}$$

$$Q^{(0)} = \begin{pmatrix} -0.25539 & -0.07440 & -0.21603 & 0.93946 \\ 0.96684 & -0.01965 & -0.05706 & 0.24815 \\ 0 & 0.99704 & -0.01724 & 0.07499 \\ 0 & 0 & 0.97457 & 0.22410 \end{pmatrix} \quad R^{(0)} = \begin{pmatrix} 716.562 & 189.447 & 88.5350 & 105.103 \\ 0 & 0.32820 & -11.9177 & 24.1837 \\ 0 & 0 & -5.62891 & 11.9258 \\ 0 & 0 & 0 & 0.02191 \end{pmatrix}$$

$$A^{(1)} = \begin{pmatrix} 0.1643922323 & 31.239616571 & -64.704608513 & 750.38221617 \\ 0.3173181897 & -11.888849194 & 23.755415435 & 4.6073277768 \\ 0 & -5.6122249602 & 11.719547613 & 2.2504716426 \\ 0 & 0 & 0.0213496146 & 0.0049093434 \end{pmatrix}$$

Algorithme - principe de base

$$A = \begin{pmatrix} -183 & -48.4064584445 & -20.5079721005 & -31.1967604278 \\ 692.8001154734 & 183.1579425466 & 86.1545159544 & 100.4673262120 \\ 0 & 0.3272285441 & -11.7853328556 & 23.9080122256 \\ 0 & 0 & -5.4857473093 & 11.6273903090 \end{pmatrix}$$

$$\underbrace{\begin{pmatrix} -0.46000 & 0.87550 & -0.14668 & -0.01950 \\ -0.88792 & -0.45357 & 0.07599 & 0.01010 \\ 0 & -0.16064 & -0.97742 & -0.12993 \\ 0 & 0 & -0.13177 & 0.99128 \end{pmatrix}}_{Q^{(1)}} \underbrace{\begin{pmatrix} -0.35737 & -3.81395 & 8.67135 & -349.268 \\ 0 & 33.6780 & -69.3768 & 654.497 \\ 0 & 0 & -0.16202 & -111.913 \\ 0 & 0 & 0 & -14.8720 \end{pmatrix}}_{R^{(1)}}$$

$$A^{(2)} = \begin{pmatrix} 3.5508638700 & -0.0280170419 & 37.311652637 & -347.38040814 \\ -29.903327829 & -3.7141310160 & -15.876691726 & 658.14383638 \\ 0 & 0.0269989710 & 14.905521505 & -110.91593560 \\ 0 & 0 & 1.9597556587 & -14.742354359 \end{pmatrix}$$

Algorithme - principe de base

$$A = \begin{pmatrix} -183 & -48.4064584445 & -20.5079721005 & -31.1967604278 \\ 692.8001154734 & 183.1579425466 & 86.1545159544 & 100.4673262120 \\ 0 & 0.3272285441 & -11.7853328556 & 23.9080122256 \\ 0 & 0 & -5.4857473093 & 11.6273903090 \end{pmatrix}$$

$$\underbrace{\begin{pmatrix} -1.00000 & 1.6e^{-16} & 0.00000 & 0.00000 \\ -1.6e^{-16} & -1.00000 & -2.e^{-295} & 0.00000 \\ 0 & 2.e^{-295} & -1.00000 & -4.2e^{-42} \\ 0 & 0 & -4.2e^{-42} & 1.00000 \end{pmatrix}}_{Q^{(784)}} \underbrace{\begin{pmatrix} 3.81165 & -30.4272 & 50.5250 & 622.739 \\ 0 & -3.64206 & 3.62773 & 386.234 \\ 0 & 0 & -1.53262 & 169.212 \\ 0 & 0 & 0 & -1.36303 \end{pmatrix}}_{R^{(784)}}$$

$$A^{(785)} = \begin{pmatrix} -3.8116468558 & 30.427166103 & -50.524986826 & 622.73876473 \\ 5.970094e^{-16} & 3.6420581683 & -3.6277337538 & 386.23421835 \\ 0 & -3.59316e^{-295} & 1.5326169130 & 169.21232904 \\ 0 & 0 & 5.677704e^{-42} & -1.3630282256 \end{pmatrix}$$

Algorithme - principe de base

$$A = \begin{pmatrix} -183 & -48.4064584445 & -20.5079721005 & -31.1967604278 \\ 692.8001154734 & 183.1579425466 & 86.1545159544 & 100.4673262120 \\ 0 & 0.3272285441 & -11.7853328556 & 23.9080122256 \\ 0 & 0 & -5.4857473093 & 11.6273903090 \end{pmatrix}$$

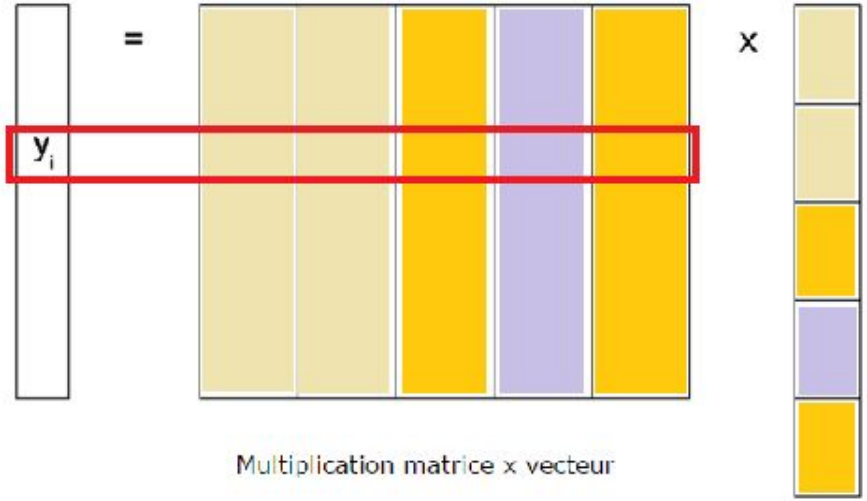
Valeurs propres trouvées:

$\lambda_1 = -3.81164685575223$
$\lambda_2 = 3.64205816828632$
$\lambda_3 = 1.53261691304088$
$\lambda_4 = -1.36302822557497$

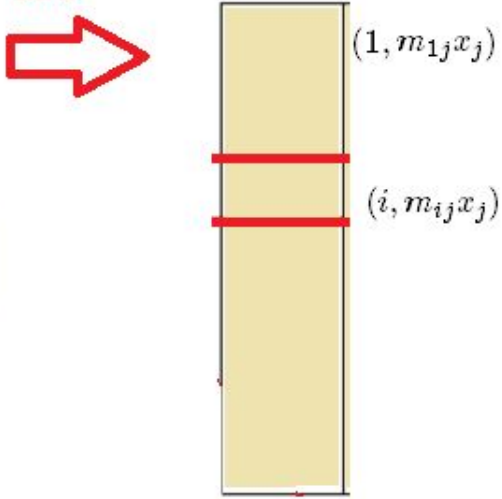
$$Q^{(784)} \begin{pmatrix} -1.00000 & 1.6e^{-16} & 0.00000 & 0.00000 \\ -1.6e^{-16} & -1.00000 & -2.e^{-295} & 0.00000 \\ 0 & 2.e^{-295} & -1.00000 & -4.2e^{-42} \\ 0 & 0 & -4.2e^{-42} & 1.00000 \end{pmatrix} \quad R^{(784)} \begin{pmatrix} 3.81165 & -30.4272 & 50.5250 & 622.739 \\ 0 & -3.64206 & 3.62773 & 386.234 \\ 0 & 0 & -1.53262 & 169.212 \\ 0 & 0 & 0 & -1.36303 \end{pmatrix}$$

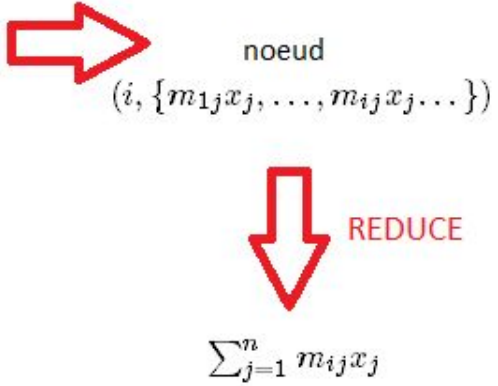
$$A^{(785)} = \begin{pmatrix} -3.8116468558 & 30.427166103 & -50.524986826 & 622.73876473 \\ 5.970094e^{-16} & 3.6420581683 & -3.6277337538 & 386.23421835 \\ 0 & -3.69316e^{-295} & 1.5326169130 & 169.21232904 \\ 0 & 0 & 5.677704e^{-42} & -1.3630282256 \end{pmatrix}$$

Distribution du calcul matriciel : y = Mx



Multiplication matrice x vecteur

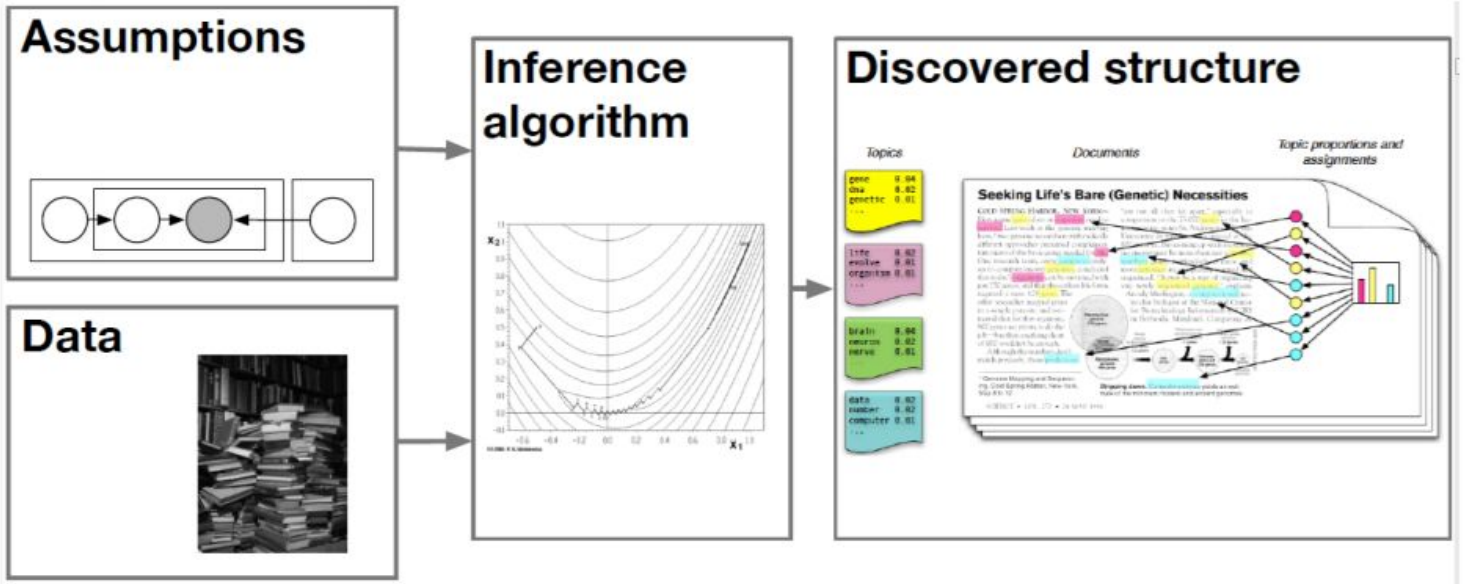




$$y = M \times x$$

$$y_i = \sum_{j=1}^{n} m_{ij}x_j$$

Latent Dirichlet allocation



LDA - rappel

La LDA extrait les **thèmes principaux** dans un grand corpus de textes

Chaque thème est décrit par une **liste ordonnée de mots représentatifs**

Chaque document est décrit par la **répartition des thèmes** qu'il traite

C'est une technique **non supervisée** qui peut être utilisée à titre exploratoire ou comme une matrice de pondération pour mesurer la similarité entre concepts, documents ou appliquer une technique supervisée (elle est alors assimilée à une technique de réduction de la dimension).

LDA - challenge

On cherche la distribution a posteriori jointe des pondérations des thèmes par document et mots par thème. La constante de normalisation est difficile à calculer à cause du couplage entre z, theta et phi, on a besoin d'une **méthode approchée**.

$$p(\theta, \phi, z|w; \alpha, \beta) = \frac{p(\theta, \phi, z, w; \alpha, \beta)}{\sum_z \int_\theta \int_\phi p(\theta, \phi, z, w; \alpha, \beta)\, d\phi\, d\theta}$$

$$= \frac{\prod_{t=1}^{T} p(\phi_t; \beta) \prod_{d=1}^{D} p(\theta_d; \alpha) \prod_{n=1}^{N_d} p(z_{d,n}|\theta_d)\, p(w_{d,n}|\phi_{z_{d,n}})}{\sum_z \int_\theta \int_\phi \prod_{t=1}^{T} p(\phi_t; \beta) \prod_{d=1}^{D} p(\theta_d; \alpha) \prod_{n=1}^{N_d} p(z_{d,n}|\theta_d)\, p(w_{d,n}|\phi_{z_{d,n}})\, d\phi\, d\theta}$$

LDA - estimation

Plus précisément, il y a deux façons de réaliser l'inférence approchée du modèle :

- L'**inférence bayésienne variationnelle** (également appelée VEM) qui estime une distribution plus simple mais proche (au sens de la divergence de Kullback-Leibler) de la distribution a posteriori (proposée initialement, 2003).

- L'**échantillonnage de Gibbs**, approche stochastique appartenant aux méthodes Markov Chain Monte Carlo (**MCMC**) qui construit une chaîne de Markov dont la distribution stationnaire est la distribution a posteriori (2004)

LDA

Les deux approches admettent leurs avantages et défauts.

L’inférence bayésienne variationnelle est une technique d’optimisation qui converge vers un **optimum local** et les formules de mise à jour sont longues et fastidieuses à dériver.

L’approche stochastique visite l’espace des variables latentes et permet donc de trouver un **optimal global**, elle est simple à implémenter mais la **convergence peut être plus longue**.

Dans la suite on s’intéresse à l’approche par échantillonnage de Gibbs

Échantillonnage de Gibbs - principe

On considère deux variables aléatoires x_1 et x_2 de probabilité jointe $p(x_1, x_2)$.

On connaît leur **probabilité marginale conditionnelle** $p(x_1|x_2)$ et $p(x_2|x_1)$.

On veut approximer leur **probabilité jointe** $p(x_1, x_2)$, qui est **difficile à calculer**

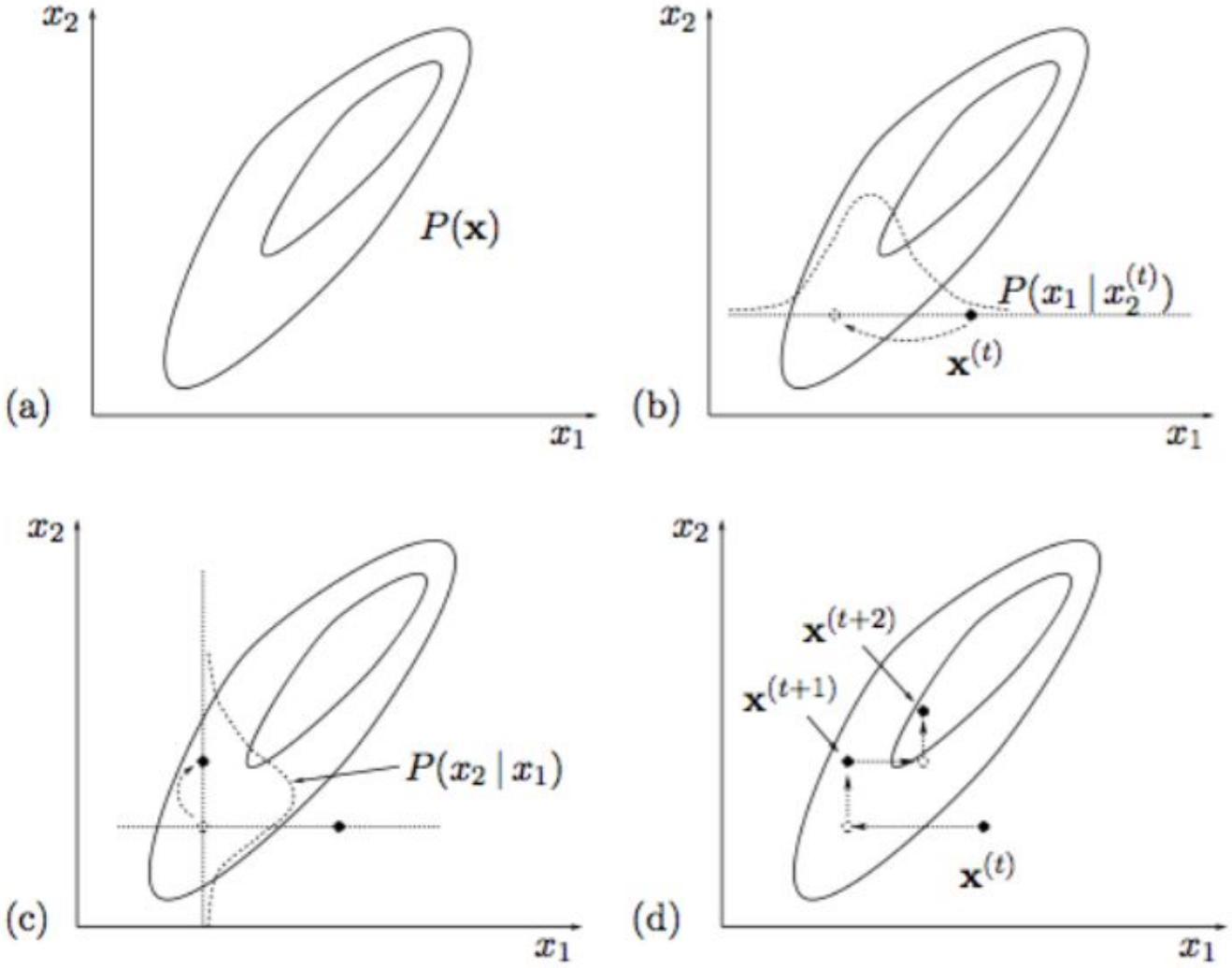
L'échantillonnage de Gibbs permet de générer successivement T échantillons $\{x^{(t)}\}$ avec $x^{(t)} = (x^{(t)}_1 , x^{(t)}_2)$, distribués suivant $p(x_1, x_2)$

Échantillonnage de Gibbs - principe

1. Initialiser aléatoirement $x_1^{(0)}$ et $x_2^{(0)}$
2. Construire l'échantillon $x^{(t+1)} = (x_1^{(t+1)}, x_2^{(t+1)})$:
   (a) Tirer aléatoirement $x_1^{(t+1)}$ suivant la probabilité marginale conditionnelle $p(x_1|x_2^{(t)})$
   (b) Tirer aléatoirement $x_2^{(t+1)}$ suivant la probabilité marginale conditionnelle $p(x_2|x_1^{(t+1)})$
3. Répéter jusqu'à ce que $t = T$

Les échantillons générés permettent d'approcher la distribution jointe $p(x_1,x_2)$:

$$p(x_1 = i, x_2 = j) \approx \frac{1}{T} \sum_{t=1}^{T} \delta(x_1^{(t)}, i) \times \delta(x_2^{(t)}, j)$$


x_2
$P(\mathbf{x})$
(a)
x_1
x_2
$P(x_1 \mid x_2^{(t)})$
$\mathbf{x}^{(t)}$
(b)
x_1
x_2
$P(x_2 \mid x_1)$
(c)
x_1
x_2
$\mathbf{x}^{(t+2)}$
$\mathbf{x}^{(t+1)}$
$\mathbf{x}^{(t)}$
(d)
x_1

LDA par échantillonnage de Gibbs

L'échantillonneur de Gibbs génère des échantillons $z^{(t)}=\{z^{(t)}_{d,n}\}_{d,n}$ où $z^{(t)}_{d,n}$ assigne pour chaque mot (indexé par n) de chaque document *d* un thème *j* avec la probabilité :

$$p(z_{d,n}=j|z_{-(d,n)}, w_{d,n}=k, w_{-(d,n)}) \propto \overbrace{\frac{\#\{z_{d,n'}=j\}_{-(d,n)}+\alpha}{\#\{z_{d,n'}\}_{-(d,n)}+T\alpha}}^{\text{Combien de fois le thème } j \text{ apparait dans le doc. } d \text{ ?}} \times \overbrace{\frac{\#\{z_{d',n'}=j, w_{d',n'}=k\}_{-(d,n)}+\beta}{\#\{z_{d',n'}=j, w_{d',n'}\}_{-(d,n)}+W\beta}}^{\text{Combien de fois le mot } k \text{ est associé au thème } j \text{ dans la collection ?}}$$

Combien de mots dans le doc. *d* ?

Combien de fois le thème *j* apparait dans la collection ?

A la fin de l'échantillonnage de Gibbs, on obtient donc les pondération des thèmes par document et des mots par thème (T est le nombre de thèmes et W la taille du vocabulaire) :

$$\hat{\theta}^{(t)}_{d,j}=\frac{\#\{z^{(t)}_{d,n'}=j\}+\alpha}{\#\{z^{(t)}_{d,n'}\}+T\alpha} \qquad \hat{\phi}^{(t)}_{j,k}=\frac{\#\{z^{(t)}_{d',n'}=j, w_{d',n'}=k\}+\beta}{\#\{z^{(t)}_{d',n'}=j, w_{d',n'}\}+W\beta}$$

LDA en grande dimension, un exemple

On **distribue chaque document sur P processeurs** (on répartit les mots des documents) :

- le nombre d'occurrences de mots alloués au thème k dans le document j $n_{k|j}$ est distribué
- mais chaque processeur possède **sa propre copie des poids des mots w dans le thème k** : $n_{w|k}$ (nombre d'occurrences de w allouées à k dans le corpus)
- et de **la répartition des thèmes dans le corpus** : n_k (nombre d'occurrences de mot allouées au thème k)

On rajoute l'indice p pour le processeur.

LDA en grande dimension, un exemple

Dans cette version approchée très simple, on effectue une LDA sur chaque processeur avec des **échantillonnages de Gibbs indépendants**.

Pour chaque processeur on va donc à une itération donnée, allouer le thème k au mot i du document j en tirant dans :

$$p(z_{ijp} = k|\mathbf{z}_p^{\neg ijp}, \mathbf{x}, \alpha, \beta) \propto (\alpha + n_{k|jp}^{\neg ijp})(\beta + n_{x_{ij}|kp}^{\neg ijp})(W\beta + n_{kp}^{\neg ijp})^{-1}$$

Après qu'on ait alloué z_p au niveau de chaque processeur (pour toutes les données dont il dispose), et donc modifié $n_{k|jp}$, $n_{w|kp}$ et n_{kp}, avant de faire un nouveau tirage on met à jour les comptes (opération **reduce**) :

$$n_{w|k} \leftarrow n_{w|k} + \sum_p (n_{w|kp} - n_{w|k}) \qquad n_{w|kp} \leftarrow n_{w|k} \qquad n_k = \sum_w n_{w|k}.$$

Passage à l'échelle de la LDA

Que l'on estime par inférence variationnelle ou échantillonnage de Gibbs, le passage à l'échelle est un challenge et un domaine de **recherche active**.

Plusieurs pistes ont été explorées et implémentées surtout du côté de l'inférence variationnelle : Collapsed Variational Bayes dans Mahout, online variational bayes dans Spark (l'étape E de l'algorithme EM peut être facilement distribuée)...

L'aspect purement séquentiel d'une chaine MCMC la rend difficile à paralléliser a priori mais l'approche proposée ici, **sans garantie théorique de convergence**, semble empiriquement performante (Distributed Inference for Latent Dirichlet Allocation, Newman et al., 2008 NIPS).

Questions ?

Contact : stephanie.combes@gmail.com