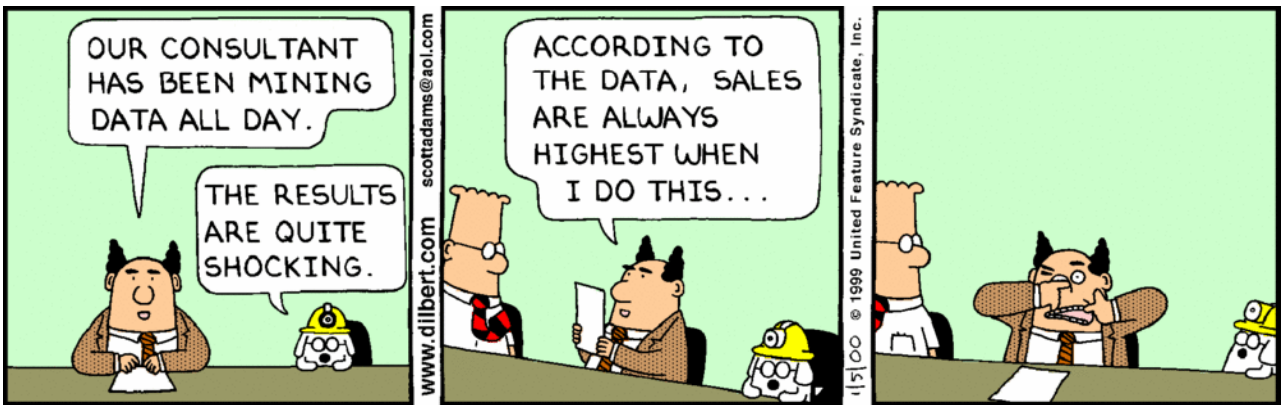
RCP 216 - INGÉNIERIE DE LA FOUILLE ET DE LA VISUALISATION DE DONNÉES MASSIVES

PROJET: NYSK DATA SET

Emeric HENRY - 27 février 2017



Introduction

Nous voulons réaliser une classification automatique d'un ensemble d'environ 10.000 articles publiés en mai 2011 et relatifs à l'affaire DSK à New-York.
Pour cela, nous devons d'abord choisir une méthode permettant une représentation numérique de chaque article, puis appliquer un algorithme de classification à la représentation obtenue.
Nous avons essayé différents algorithmes, et présentons dans ce rapport la méthode retenue, qui nous semble avoir donné les meilleurs résultats.

Le jeu de données

Le jeu de données « NYSK Data Set » est constitué de 10.421 articles en anglais publiés en ligne entre le 17 mai 2011 à 12h50 et le 26 mai 2011 à 15h50, heure de la côte est des USA. Ils ont été obtenus par une simple recherche sur internet, avec les mots clés dsk, strauss-kahn, strauss-khan.
Chaque article est constitué des informations suivantes:

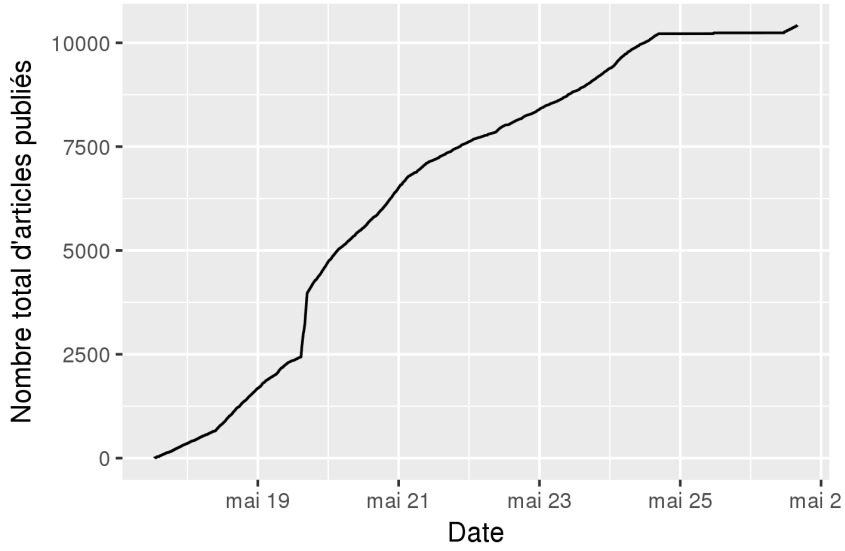
- un identifiant unique
- la source (l'organe de presse)
- l'url de publication de l'article
- le titre
- un résumé
- le texte entier
- la date et l'heure de publication

Ce jeu de données a fait l'objet d'une étude académique qui porte sur l'évolution du sentiment (positif ou négatif) en rapport avec un nombre donné de thématiques (8 dans l'article [2]). Pour cela, les auteurs ont utilisé une deuxième source de données, obtenue à partir des commentaires faits sur les articles vendus par Amazon, pour associer aux mots une tonalité positive, neutre ou négative. La méthode pour construire ces données de sentiment est présentée plus en détail dans l'article [3], mais elle n'y est pas appliquée au jeu de données NYSK.

Le rythme de publication des articles est sensiblement constant sur la période, avec deux exceptions très notables:

- le 19 mai vers 14h: un grand nombre d'articles est publié simultanément
- entre le 24 mai à 15h et le 26 mai à 12h, pratiquement aucun article n'est publié (ou en tout cas ne figure dans le jeu de données)

La figure ci-dessous retrace l'évolution du nombre d'articles dans le jeu de données en fonction du temps. On y voit les cassures de pente mentionnées le 19 mai et du 24 au 26 mai. On notera également qu'il n'y a pas d'interruption des publications la nuit. Le rythme de publication est d'environ 1 article par minute.



Méthode de représentation numérique des articles

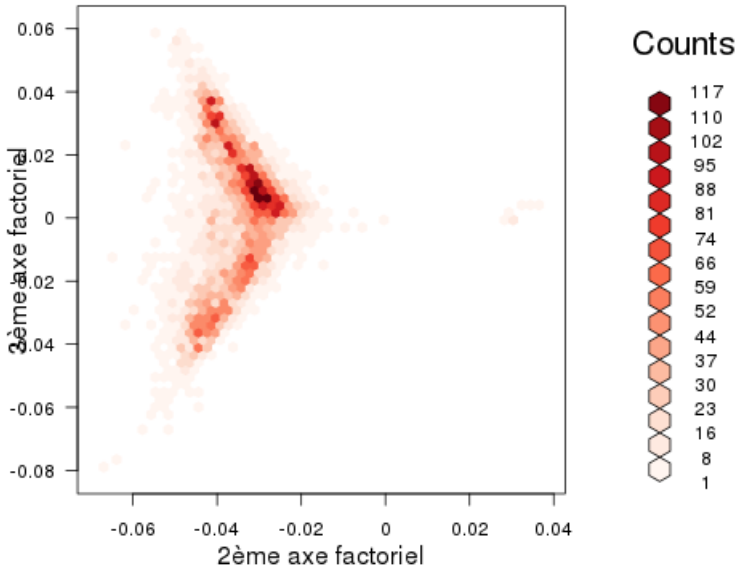
Analyse sémantique latente

La première méthode qui semble adaptée au problème est l'analyse sémantique latente (LSA), que nous avons vue en cours et en TP, et qui est implémentée dans Spark.
La fonction scala utilisée est `termDocumentMatrix` du fichier `ParseWikipedia.scala`. Mais nous l'avons légèrement réécrite: en effet, bien qu'elle prenne un `Set[String]` en argument pour les stop words, elle ne l'utilise pas. Nous avons donc modifié le code de manière à filtrer:

```
def termDocumentMatrix(docs: RDD[(String, Seq[String])], stopWords:
Set[String], numTerms: Int, sc: SparkContext):
(RDD[Vector], Map[Int, String], Map[Long, String], Map[String, Double]) = {
  val docTermFreqs = docs.mapValues(
    terms => {
      val termFreqsInDoc = terms.foldLeft(new HashMap[String, Int]()) {
        (map, term) => map += term -> (map.getOrElse(term, 0) + 1)
      }
      termFreqsInDoc
    }
  )
…

def termDocumentMatrix(docs: RDD[(String, Seq[String])], stopWords:
Set[String], numTerms: Int, sc: SparkContext):
(RDD[Vector], Map[Int, String], Map[Long, String], Map[String, Double]) = {
  val docTermFreqs = docs.mapValues(
    terms => {
      val termFreqsInDoc = terms.filter( lemma => ! (stopWords contains
lemma)).foldLeft(new HashMap[String, Int]()) {
        (map, term) => map += term -> (map.getOrElse(term, 0) + 1)
      }
      termFreqsInDoc
    }
  )
…
```

Nous avons également remplacé le filtre présent dans la fonction (isOnlyLetters, qui ne retient un mot que s'il est composé uniquement de lettres) par un filtre qui accepte un mot dès qu'il contient au moins une lettre. Ainsi des mots comme Eurostoxx50 sont conservés, ainsi que les mots composés comme Strauss-Kahn.
Nous avons essayé cette méthode: le texte de l'article est découpé en mots, qui sont lemmatisés, puis on calcule les fréquences TF-IDF de chaque lemme. Un article est alors décrit par chacune des valeurs TF-IDF pour tous les termes retenus (1.000).
En retenant 1.000 termes, on travaille dans un espace de dimension 1.000, ce qui nous amène à faire une Analyse en Composantes Principales (ACP) pour visualiser les données. Plus précisément, nous avons réalisé une décomposition en valeurs singulières (SVD), plus adaptée à la taille de la matrice (1.000 colonnes et 10.000 lignes).



Dans la figure ci-dessus, on a tracé la densité de points dans le plan formé par les 2èmes et 3èmes axes factoriels principaux (le 1er axe n'est pas intéressant, car c'est un facteur taille: la matrice TF-IDF n'a que des termes positifs).
On peut raisonnablement dire que des zones denses sont séparées par des zones moins denses, et que donc on arrivera à distinguer des classes.
Quantitativement, on peut estimer la séparabilité du nuage de points en faisant le ratio entre les pics et les creux de densité. Sur la figure ci-dessus, on arrive à un ratio de l'ordre de 2 (115 / 50).

Word2Vec associé à la LSA

Dans la démarche précédente, chaque terme était représenté par une coordonnée. La lemmatisation a pour conséquence que des formes grammaticales du même lemme sont regroupées. Mais on pourrait regrouper encore mieux si on arrivait à intégrer la notion de synonymes.
Pour cela, le modèle word2vec est particulièrement adapté. Des lemmes synonymes auront une représentation vectorielle très proche.
Un article sera alors représenté par une combinaison linéaire de tous les vecteurs correspondant à chacun de ses lemmes. La combinaison linéaire que nous avons choisie n'est pas la simple moyenne arithmétique sur les mots

$$\sum_{\text{mot}=1}^{M} \frac{1}{M}\,\vec{v}_{\text{m}} = \sum_{\text{lemme}=1}^{L} \frac{n_l}{M}\,\vec{v}_{\text{l}} = \sum_{\text{lemme}=1}^{L} \text{tf}_l\,\vec{v}_{\text{l}}$$

mais nous avons pondéré le vecteur correspondant au lemme par sa tf-idf:

$$\frac{\sum_{l=1}^{L} \text{tf-idf}_l\,\vec{v}_{\text{l}}}{\sum_{l=1}^{L} \text{tf-idf}_l}$$

Par rapport à la moyenne arithmétique simple, on donne moins de poids aux lemmes qui sont communs à beaucoup de documents. On renforce ainsi les lemmes rares dans l'ensemble des articles, qui sont plus discriminants.

Reconstruction du modèle Word2Vec

Le modèle word2vec fourni pendant les TPs a été construit avec le fichier `text8`, qui correspond aux premiers 100 Mo de Wikipédia. Le problème est que de nombreux mots présents dans nos articles n'y apparaissent pas, et par conséquent leur représentation vectorielle est impossible. Il s'agit de mots importants, comme « dsk », « imf » ou encore « lagarde ».
Nous avons donc reconstruit un modèle word2vec à partir d'un corpus de phrases constitué par

- le fichier `text8` (plus exactement, nous avons également reconstruit ce fichier, car il ne comporte pas de retour à la ligne. A partir du fichier `enwik8` disponible en ligne [4], nous avons extrait 827.233 paragraphes).
- nos données NYSK, qui comportent 10.421 articles, chacun traité comme un seul paragraphe.

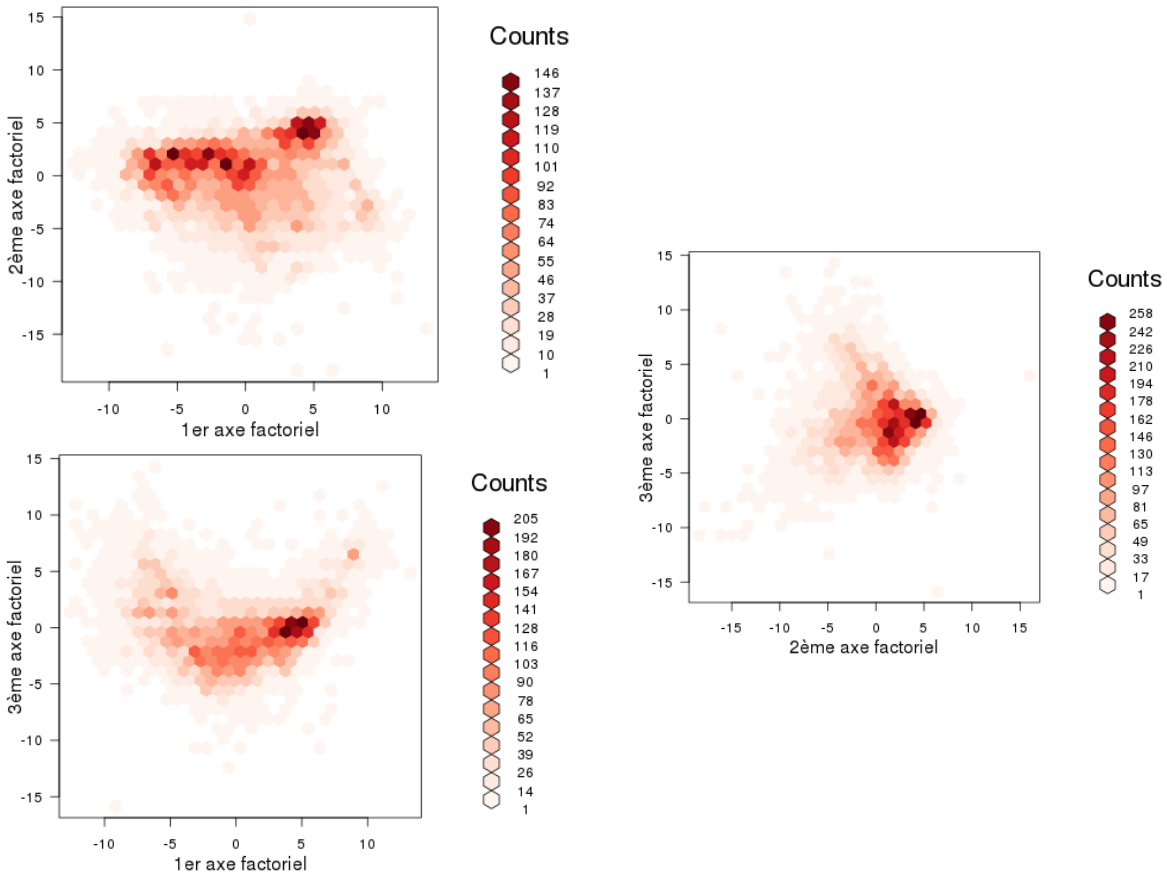
Ce corpus de phrases est très proche du corpus initial `text8`, car nos données NYSK ne représentent que 1,2 % du total. Nous ne perturbons donc pas trop l'apprentissage du vocabulaire général, tout en permettant au modèle de connaître les termes spécifiques de notre contexte.
Le programme Scala qui réalise cette construction du modèle est `w2vec.scala`. Il est invoqué comme suit:

```
spark-submit --class "w2vec" --jars lib/lsa.jar,lib/common.jar --
master yarn target/scala-2.10/nysk_2.10-1.0.jar
```

Visualisation du nuage de points

En résumé, nous procédons dans l'ordre à
- la lemmatisation du texte
- le calcul des tf-idfs
- la somme pour tous les lemmes d'un article de: tf-idfs * word2vec(lemme)

Et pour visualiser le nuage de points ainsi obtenu, nous le projetons sur les principaux plans factoriels obtenus par ACP.
Dans les figures ci-dessous, nous avons représenté les projections dans les 3 premiers plans factoriels. On voit en particulier dans le plan (1-2) qu'il n'est pas absurde de parler de classes différentes: des zones très denses sont séparées par des zones beaucoup moins denses. Quantitativement, les pics et les creux de densité sont dans un rapport d'environ 3 (145 / 45). C'est pour cette raison que nous pensons que la séparabilité en classes est meilleure avec ce type de représentation vectorielle, plutôt qu'avec la simple LSA.
Visuellement, on arrive a distinguer environ 3 ou 4 classes. Quand on appliquera un algorithme de classification, on tentera de construire un nombre de classe au moins supérieur. On pourra en effet regrouper à la main des classes similaires si on en a trop, alors que si on n'en a pas assez, on ne le saura pas.



Classification automatique

Pour la classification automatique, nous avons retenu l'algorithme des K-moyennes, avec un nombre de classes égal à 10.

Résultats pour la méthode LSA

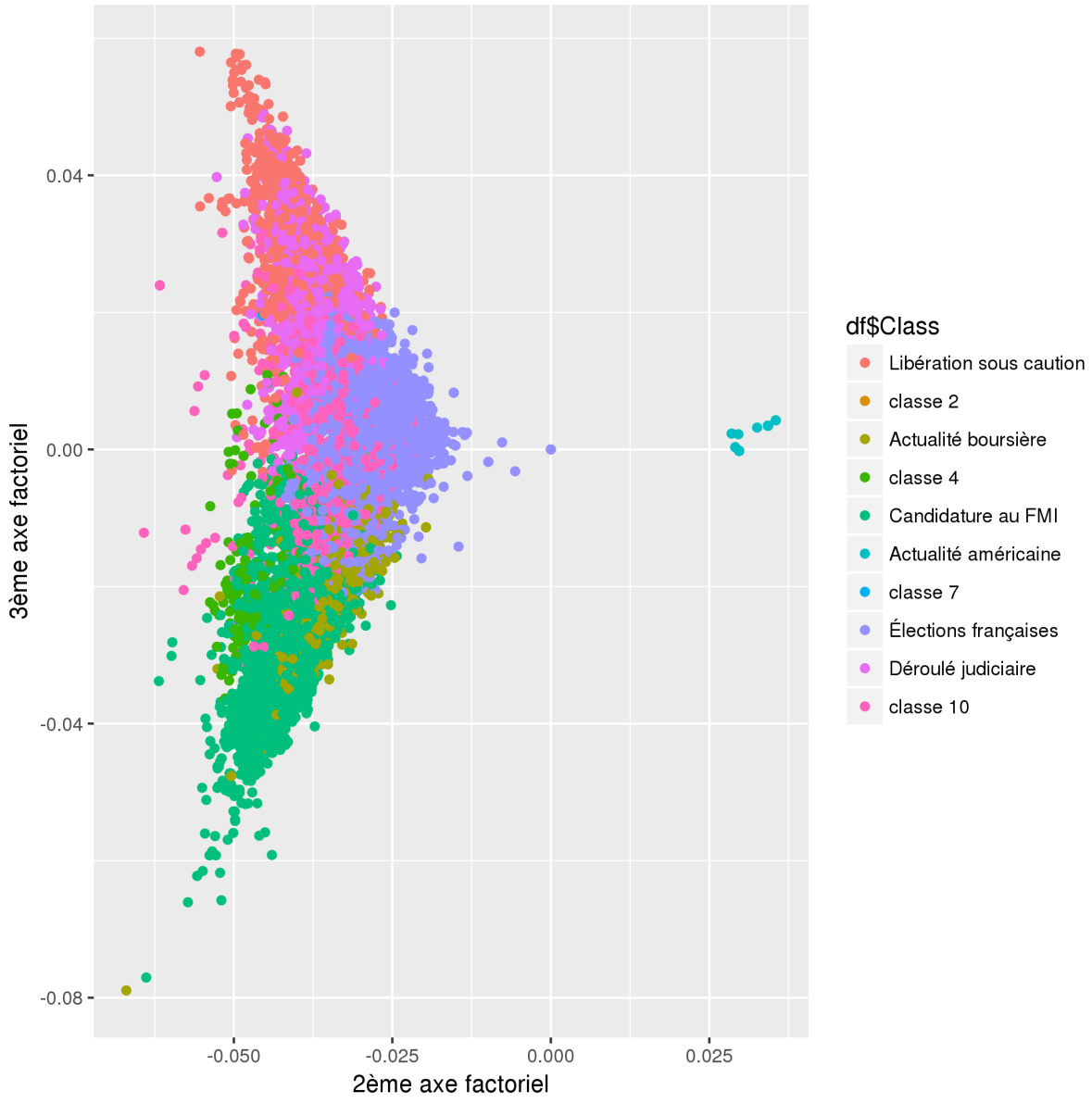
La méthode retourne les coordonnées des centres des classes. Dans l'espace vectoriel où nous travaillons, ces coordonnées s'interprètent comme une tf-idf pour un document fictif. Les valeurs les plus élevées des coordonnées correspondent aux termes les plus pertinents de ce document fictif.
On donne les 10 termes les plus représentatifs de chacune des 10 classes dans le tableau ci-dessous:

Classe	Lemmes	Interprétation
1	apartment,bail,sinclair,building,million,manhattan,release,court,guard,judge	libération sous caution
2	maintain,site,media,incident,rate,michael,eye,widely,secretary,sunday	—
3	percent,stock,market,index,price,trading,datum,fall,higher,investor	actualité boursière
4	reuter,product,news,comment,professional,mobile,index,com,capital,review	—
5	lagarde,european,minister,finance,candidate,europe,bank,economy,imf,emerge	candidatures de remplacement au FMI
6	journal,opinion,obama,geithner,scandal,politics,limit,tax,age,previous	actualité américaine
7	request,information,incident,rate,michael,eye,widely,secretary,sunday,launch	—
8	woman,obama,man,dsk,sarkozy,know,people,president,socialist,france	scandale et élections françaises
9	shapiro,dna,lawyer,room,brafman,hotel,evidence,maid,sofitel,bail	déroulé judiciaire
10	comment,email,com,news,article,search,twitter,facebook,business,subscribe	—

Les classes 1, 5, 8 et 9 ont une interprétation cohérente et nous intéressent particulièrement. Les autres sont

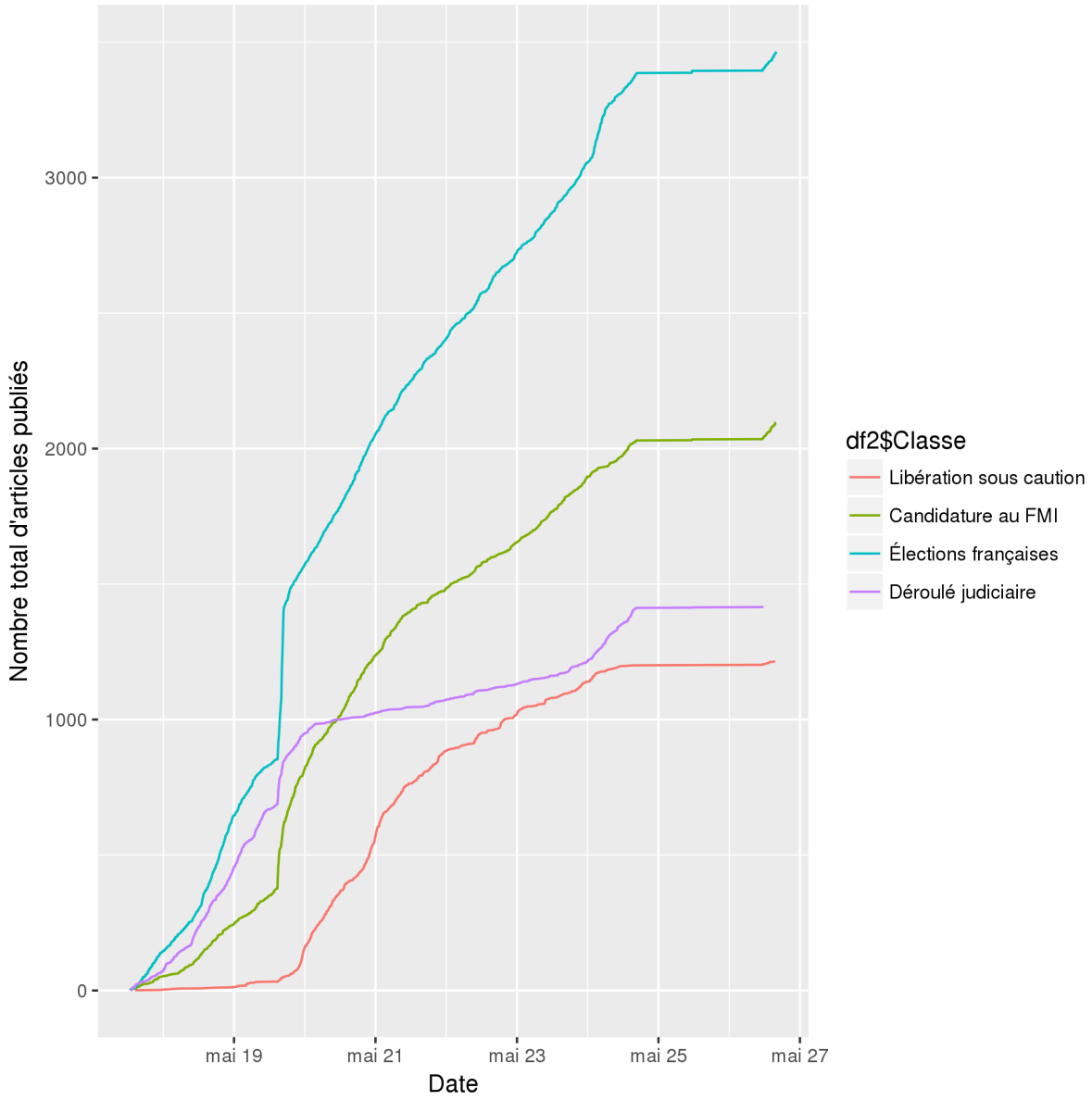
- soit difficiles à interpréter, comme par exemple la classe 10.
- soit éloignées du véritable sujet. Ainsi, les articles d'actualité boursière qui constituent la classe 3 peuvent tout à fait contenir le mot-clé « DSK » au détour d'une phrase, sans que le sujet principal de l'article ne porte sur l'affaire.

On peut visualiser les 10 classes dans le plan factoriel (2-3) de la section précédente:



Pour les 4 classes retenues traitant explicitement de notre sujet, le graphique ci-dessous totalise le nombre d'articles publiés pour chacune des classes.
On peut faire les remarques suivantes:

- globalement le nombre d'articles publiés augmente spectaculairement le 19 mai vers 10h. Cela correspond à l'inculpation de DSK par un grand jury.
- ce sont particulièrement les articles parlant des conséquences sur les élections françaises qui augmentent.
- après cette date, les articles commencent à parler d'une libération sous caution, alors qu'ils n'en parlaient pas avant



- enfin, après cette date les articles parlant du déroulé judiciaire sont moins nombreux, et ce sont plutôt les candidatures de remplacement au FMI qui sont évoquées.

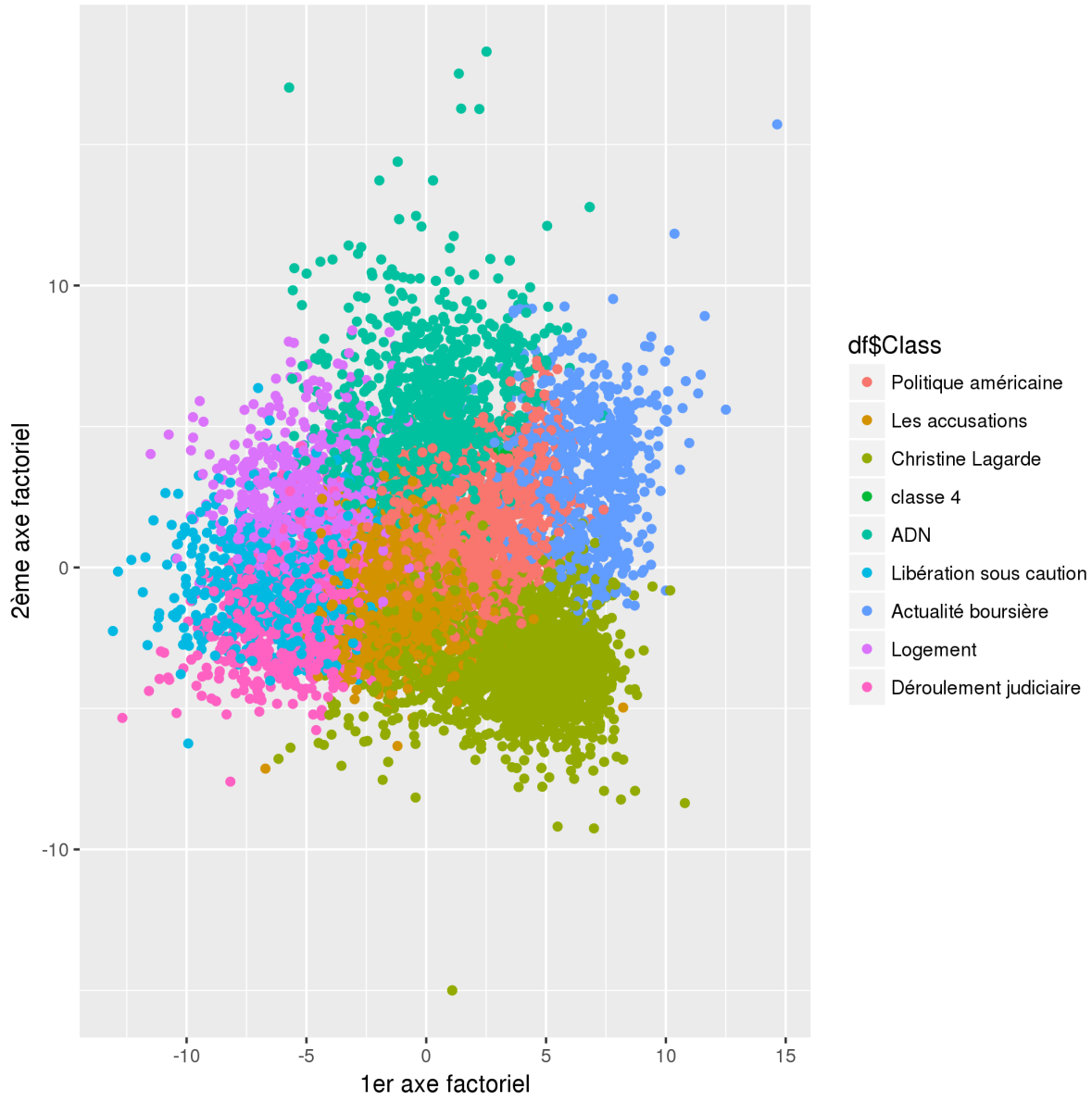
Résultats pour la méthode LSA + word2vec

De la même manière, nous utilisons les K-moyennes avec un nombre de classes égal à 10.
Les centres des classes sont maintenant des points dont les coordonnées sont abstraites. Pour donner un sens au centre de la classe, nous cherchons les 10 plus proches voisins du centre parmi les articles. Enfin, pour chaque article, nous regardons quels termes ont le tf-idf le plus élevé.
Les résultats sont donnés dans le tableau ci-dessous (les doublons ne sont pas reproduits):

Classe	Lemmes	Interprétation
1	obama,speech,white,opinion,subscribe, russia,dsk,political,history,level, russia,wednesday,prime,region,candidacy, average,lipsky,greece,social,rate, indian,obama,middle,pay,thursday, russia,tuesday,reform,biggest,economic, ireland,lagarde,japan,cost,obama, russia,greece,war,explain,wednesday, com,die,latest,class,student, middle,launch,growth,east,india,	politique américaine
2	dsk,woman,affair,socialist,plot, woman,dsk,france,know,lot, dsk,yesterday,paris,daily,sarkozy, post,employee,sport,scandal,comment, man,dsk,woman,scandal,admit, dsk,man,fact,public,michael, image,economist,innocence,law,know, shapiro,child,abuse,woman,man,	les accusations
3	lagarde,friday,board,minister,dervi, lagarde,friday,minister,dervi,finance, european,comment,europe,candidate,decision, lagarde,friday,minister,finance,dervi, lagarde,friday,board,minister,dervi,	christine lagarde
4	journal,opinion,obama,geithner,scandal,	—
5	click,track,dna,news,story, review,issue,national,print,profile, click,read,check,dsk,story, shapiro,morning,subscribe,jeffrey,program, reuter,product,news,comment,minute, network,newspaper,online,subscribe,social, visit,obama,content,comment,yesterday, dna,site,story,reporter,idea, content,today,obama,visit,comment,	l'ADN
6	bail,taylor,lawyer,release,judge, sinclair,jury,grand,code,bail, bail,taylor,court,lawyer,release, sinclair,jury,grand,code,bail, judge,sinclair,jury,grand,bail, bail,flight,scene,judge,taylor,	libération sous caution

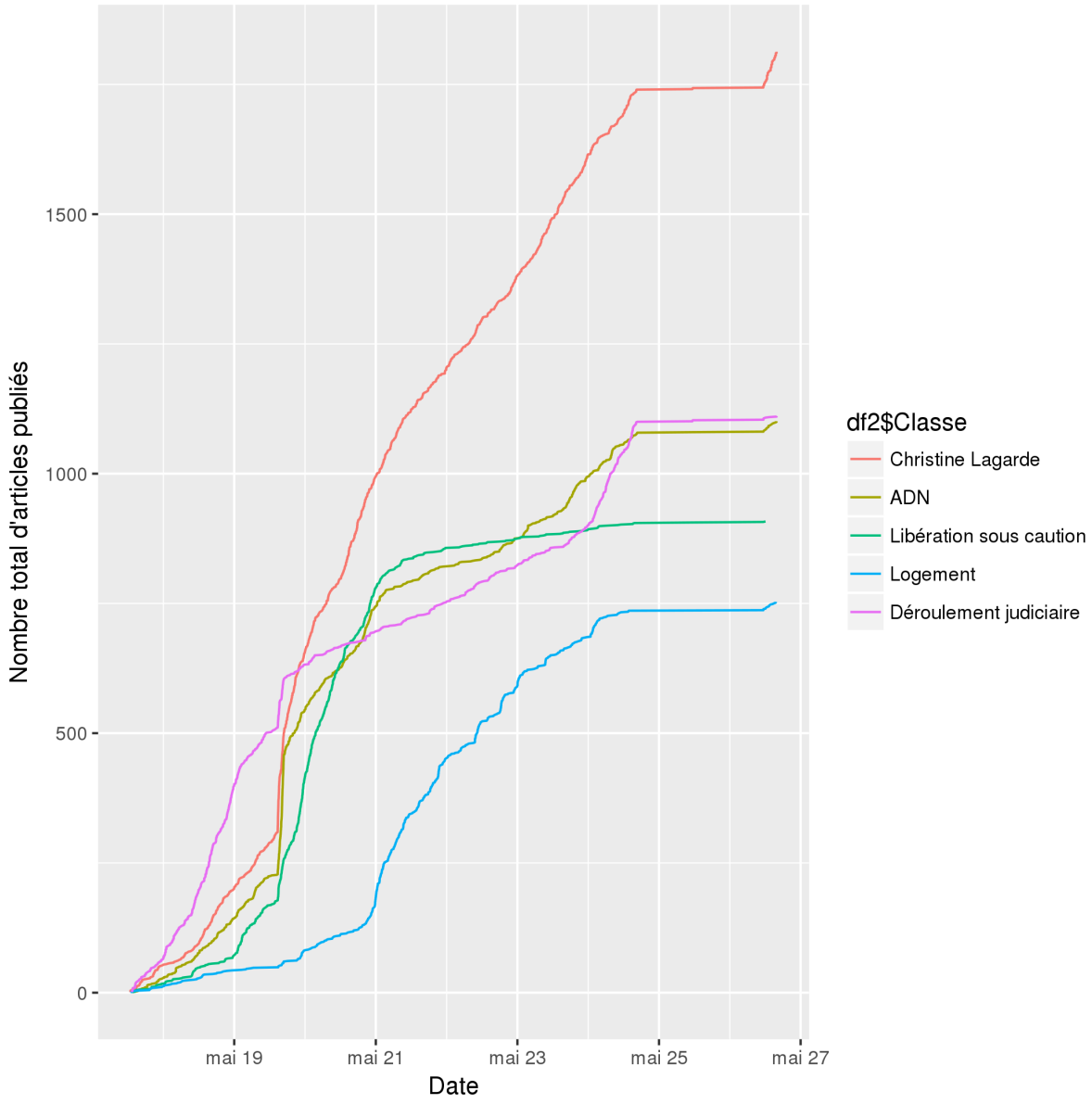
Classe	Lemmes	Interprétation
7	price,final,understand,sale,index, gain,stock,recent,share,index, debt,limit,percent,stock,exchange, ireland,index,price,market,bank, demand,price,research,investment,fall, price,eurozone,loan,bank,rise, datum,trade,level,sell,focus, demand,price,fall,capital,longer, billion,stock,price,firm,company,	actualité boursière
8	wealth,guard,expert,camera,house, building,afp,obama,apartment,house, temporary,taylor,building,effort,release, image,wealth,guard,expert,camera, apartment,building,visit,weekly,resident, location,house,bail,obus,building,	logement
9	source,socialist,reuter,situation,selection, tuesday,turkey,battle,board,finance, european,lagarde,resignation,reuter,economy, source,socialist,minister,selection,china, socialist,source,situation,selection,china,	—
10	shapiro,woman,client,bathroom,clean, advertise,forensic,prosecutor,authority,inside, authority,complex,phone,lawyer,assistant, shapiro,woman,client,bathroom,employee, shapiro,woman,accuser,encounter,lawyer, shapiro,woman,client,forensic,sofitel,	le déroulé judiciaire

L'interprétation des classes est sensiblement la même que les précédentes. On notera qu'une classe apparaît ici qui n'apparaissait pas précédemment: celle qui évoque les thématiques de logement.
La visualisation des classes dans le 1er plan factoriel est présentée sur la figure ci-dessous.



L'évolution des thématiques est présentée dans la figure ci-après.
On observe que:

- les premiers articles publiés concernent le déroulement judiciaire
- comme précédemment, le thème « Christine Lagarde » devient majoritaire au fil du temps
- la thématique « logement » n'apparaît qu'après le 21 mai, soit après que soit intervenue la libération sous caution.



Conclusions

Nous avons réalisé une classification automatique des articles du jeu de données NYSK, au moyen de deux représentations vectorielles différentes des articles: la méthode LSA, et le modèle word2vec (pondéré par le score tf-idf).
Les deux méthodes donnent des résultats sensiblement identiques, bien que les classifications résultantes soient légèrement différentes.
Parmi les conclusions quantitatives qu'on peut tirer de la classification et de son évolution dans le temps:

- les premiers articles évoquent le déroulé judiciaire

- ensuite le 19 mai apparaît la thématique d'une libération sous caution, suite à l'inculpation par un grand jury
- puis le 21 mai une thématique « logement »
- enfin, au fil du temps, les articles parlent progressivement de plus en plus de la candidature de Christine Lagarde au FMI

Références

[1] NYSK Data Set. http://archive.ics.uci.edu/ml/datasets/NYSK
[2] Dermouche M., Velcin J., Khouas L. & Loudcher S. A Joint Model for Topic-Sentiment Evolution over Time. https://pdfs.semanticscholar.org/7a9a/87a3fbd0575690aeb3643f84db829951e268.pdf
[3] Dermouche M., Khouas L., Velcin J. & Loudcher S. A Joint Model for Topic-Sentiment Evolution from Text. http://mediamining.univ-lyon2.fr/velcin/public/publis/SAC_2015_PREPRINT.pdf
[4] http://www.mattmahoney.net/dc/textdata