Université Claude Bernard Lyon I

Master Data Science

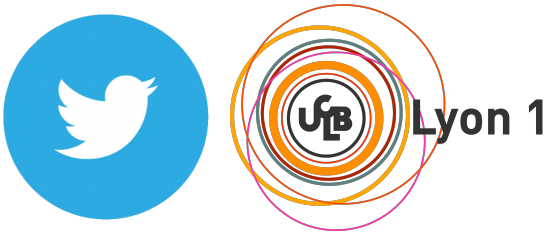
Projet Data Mining

Détection d'événements sur Twitter

Auteurs :
Gregory Howard *11207726*
Marine Ruiz *11208141*
Jules Sauvinet *p1412086*

Professeur :
Marc Plantevit

20 janvier 2017



Résumé

La détection d'évènement est un des sujets de recherche les plus importants dans l'analyse des réseaux sociaux. Les flux de données provenant des plates-formes de réseaux sociaux contiennent en général beaucoup d'informations, permettant notamment de pouvoir faire de la détection d'évènement. Nánmoins, une grande partie est bruitée, notamment pas des comptes Twitter entretenus par des robots qui empêchent de détecter des événements ou qui en créént de faux. Il est ainsi important de comprendre comment atténuer l'influence du bruit pour faire de la détection d'év́enements. Notre objectif a été de détecter des évènements à partir de tweets en exploitant des informations spatio-temporelles et textuelles en essayant d'écarter les tweets de robots.

Table des matières

1 Introduction

Notre objectif a été de détecter des évènements à partir de tweets en exploitant des informations spatio-temporelles et textuelles. Notre approche utilise la technique de détection d'événements multi-échelles dans les réseaux sociaux introduite dans l'article de recherche [1] et développé par Ahmed Anes Bendimerad et Aimene Belfodil. Nous avons adapté cette version à nos données à laquelle nous avons ajouté des mécanismes permettant de filtrer les tweets envoyés par des robots en amont du clustering permettant la détection d'évènements. Nous avons également développé une approche de détection spatio-temporelle d'évènements avec l'algorithme de clustering DBScan puis nous avons comparé les résultats des deux types de clustering.

2 Description de nos données

Source du fichier

Nous avons effectué notre détection d'évèvenent à partir d'un fichier de 1,7 Go comportant 10982005 tweets récupérés sur Twitter. Chaque tweet contient les informations suivantes :

- un identifiant
- la date et l'heure à laquelle le tweet a été posté
- la position de la personne quand le tweet a été posté (latitude et longitude)
- l'identifiant du lieu géographique où a été posté le tweet
- le lieu du tweet au moment où il a été posté
- l'ensemble des références et hashtags du tweet
- l'identifiant de la personne qui a posté le tweet
- le pseudo de la personne qui a posté le tweet
- le nom complet de la personne qui a posté le tweet
- l'identifiant du lieu de création du compte de la personne qui a posté le tweet
- l'information de la certification ou non du compte de l'utilisateur ayant posté le tweet
- le nombre de personnes qui suivent la personne qui a posté le tweet
- le nombre d'amis de la personne qui a posté le tweet
- le nombre de tweet déjà posté par la personne qui a posté le tweet

Pour le détection d'évènement, nous nous sommes servi de l'identifiant du tweet, du nom de la personne qui a posté le tweet, des hashtags, de la date et de la position d'envoi du tweet.

3 Le pré-traitement des données

Nous avons utiliser Knime pour effectuer un premier filtrage nous permettant d'augmenter la pertinence des évènements détectés. Nous supprimons les 0.5% d'utilisateur qui tweetent les plus. Cela nous permet d'homogénéiser nos données. D'une part, il y a une probabilité importante qu'un utilisateur qui poste une trés forte quantité de tweet soit un robot. D'autre part, nous nous sommes dit qu'un utilisateur réel ayant une activité sur Twitter bien supérieure à celle des autres utilisateur avait en moyenne moins de contenu descriptif d'évèvenement dans ses tweets et plus de "tweets

bruits", c'est à dire des tweets nous donnant aucune informations sur le déroulement d'un quelconque évèvenement (e.g "#lol", ou encore "#oklm avec mon refré"). Nous avons ensuite confirmé cette intuition empiriquement en observant notre jeu de données. Cette technique étant très appromative, nous ne l'utilisons que pour filtrer une quantité peu importante d'utilisateurs.
Nous avons joint le workflow Knime robotFilter.knwf permettant de faire ce filtrage à l'archive de rendu du projet.

La dimension géographique étant cruciale dans notre méthode de clustering, nous avons également filtré les tweets ne possédant pas de position géographique sur le lieu de leur envoi.
Nous n'avons á la fin plus que 1,600,000 de tweets et un fichier de 250Mo.

4 L'approche globale

Pour notre premier algorithme, nous faisons des sous-ensembles de tweets en découpant nos données par date. On a ainsi un ensemble de tweets par jour. On applique sur chaque ensemble journalier de tweets, l'algorithme de détection d'évènement multi-échelle.

Pour notre second algorithme, nous supprimons avant tout traitement, les comptes robots que nous sommes susceptibles d'avoir dans nos données. Il est fort probable qu'ils faussent notre clustering.

De plus, chacun de nos événements est décrit par 20 hashtags pertinents. Ces hashtags pertinents sont définis comme étant les plus récurrents dans les tweets contenus dans l'événement. Cependant, certains hashtags sont présents dans la description de beaucoup d'autres événements, ce qui les rend moins pertinents. Nous voulons les détecter et ne pas les prendre en compte lors du clustering afin d'avoir une vraie description de l'événement.

5 La détection d'évènement multi-ćhelle

La technique de détection d'évenement n'est autre que celle évoquée dans l'article [1].

L'algorithme calcule la matrice de similarité et construit les clusters.

6 Description des éléments utiles à la détection d'événements

Le tweet
On décrit un tweet par :

- son identifiant
- le nom du Tweetos
- les hashtags
- la date et l'heure à laquelle le Tweetos a tweeté
- sa position en latitude/longitude au moment du tweet
- La date

On découpe nos données par date c'est à dire par journée. Une date est caractérisée par une journée, un mois et une année.

La matrice de similarité
La matrice de similarité est une matrice triangulaire supérieure de taille (nombre de tweets) * (nombre de tweets). Décrire construction Citer l'article en parlant rapidement de la compression de Haar

Le clustering
Le clustering se fait à partir de la matrice de similarité. Décrire mieux l'algo
Citer le jar et expliquer rapidement ce qu'on maximise / minimise dans l'algorithme

L'événement
On décrit un événement par :

- une heure de début et une heure de fin qui permettent de calculer une durée
- une position centrale (longitude, latitude) calculée à partir de la moyenne de toutes les positions (moyenne des longitudes, moyenne des longitudes)
- le nombre de personnes différentes ayant posté des tweets
- la liste des 20 hashtags les plus importants permettant de le décrire

7 Algorithme de la détection d'événements

on traite ensuite tous les tweets d'une même journée
on enlève ceux qui sont régi par une loi Géométrique (a voir)
on fait du clustering sur les tweets et on renvoie les clusters pertinents sous forme d'événements

8 Les options sur le clustering

L'élasticité

On suppose qu'un hashtag est fréquent pour un ensemble de tweets donné s'il apparait plus de 20 fois.
Ce nombre est statique, ce qui ne permettrait pas de détecter de petits événements dans une journée où il y a eu peu de tweets. A l'inverse, on détecte beaucoup d'événements sur une journée où il y a eu beaucoup de tweets. Cela peut être intéressant, mais il est parfois plus judicieux de d'adapter au nombre de tweets postés dans une même journée. Ainsi, on définit un hashtag fréquent s'il apparait dans plus de 20Pour coder cette option nous avons définit un booléen, que l'on met à Vrai pour activer l'élasticité et à Faux pour ne pas l'activer. Les valeurs 20 et 20module d'éxécution.

La géolocalisation

La loi de Poisson géographique

9 La visualisation du résultat

L'utilité de la visualisation

Décrire

La source de données Décrire

L'organisation de la visualisation Décrire

10 Interprétation de nos résultats

11 Notre essai sur DBScan

12 Conclusion

Les combinaisons intéressantes Quand, comment

Acknowledgments

Nos remerciements vont à Marc Plantevit pour l'enseignement de son cours "Data Mining" à l'Université Claude Bernard Lyon I et son accompagnement durant ce projet.

Références

[1] X. Dong, M. D., Calabrese, and P. Frossard. Multiscale event detection in social media. *Data Min Knowl Disc*, June 2015. 2, 5