Université Claude Bernard Lyon I

Master Data Science

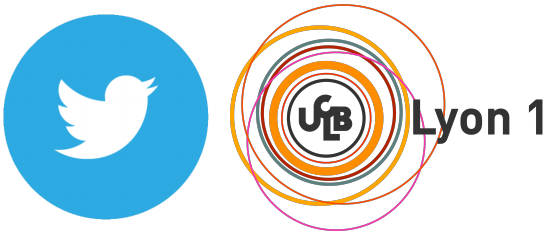
Projet Data Mining

Détection d'événements sur Twitter

Auteurs :
Gregory Howard *11207726*
Marine Ruiz *11208141*
Jules Sauvinet *11412086*

Professeur :
Marc Plantevit

21 janvier 2017



Résumé

La détection d'évènements est un des sujets de recherche les plus importants dans l'analyse des réseaux sociaux. Les flux de données provenant des plates-formes de ces réseaux contiennent, la plupart du temps, beaucoup d'informations. On peut traiter et analyser ces informations dans le but de faire de la détection d'évènement(s).
Néanmoins, une grande partie des données aiter est bruitée, notamment par des comptes Twitter entretenus par des robots. Ces comptes postent des tweets en masse avec une fréquence de publication constante, ce qui nous empêchent de donner une bonne description dún événement ou alors on peut créer de faux évènements.
Voila pourquoi il est important de comprendre comment atténuer l'influence du bruit pour faire de la détection évènements. 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.

Nous appelerons évèvenement (au sens évèvenement que nous souhaitons détecter) un phénomène physique survenant en un point et pendant une durée bien déterminé. Un événement est intéressant s'il suscite un nombre suffisant de tweets. Ces tweets peuvent se situer sur le lieu de l'évènement ou bien ailleurs, notamment si celui-ci est retransmis par les médias. Les tweets sur l'évènement peuvent se produire avant, pendant et après l'événement. Ainsi, la détection d'évènement doit réussir à intégrer ces dispersions pour localiser dans les temps et l'espace les évènements et c'est ce qui la rend complexe.

2 Description de nos données

Nous avons effectué notre détection d'évèvenents à partir d'un fichier de 1,7 Go comportant 10 982 005 tweets récupérés sur Twitter. Ces tweets ont été posté dans l'agglomération newyorkaise ou par des utilisateurs newyorkais du 21 juillet 2015 au 13 novembre 2015. 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'évènements, 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 utilisé Knime pour effectuer un premier filtrage des robots. Pour cela, nous classons dans l'ordre décroissant les utilisateurs en fonction du nombre de tweets qu'ils ont postés. Puis, nous supprimons les 0.5% premiers utilisateurs à partir de ce classement. Ce pré-traitement nous permet d'homogénéiser nos données mais aussi d'augmenter de façon évidente la pertinence des évènements détectés.

Nous avons basé ce choix de filtrage sur les deux critères suivants. En effet, d'une part, il y a une probabilité importante qu'un utilisateur qui poste en très grande quantité des tweets 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 utilisateurs, avait en moyenne moins de contenu descriptif d'évènevement dans ses tweets et plus de "tweets bruits", c'est à dire des tweets nous donnant aucune information sur le déroulement d'un quelconque évènevement (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 approximative, nous ne l'utilisons que pour filtrer les utilisateurs qui publient, vraiment de façon évidente, plus que tous les autres utilisateurs.

La dimension géographique étant cruciale dans notre méthode de clustering, nous avons également filtré les tweets ne possédant pas de position géographique dans le module Knime.
Nous avons donc donné en entra Knime notre fichier de 1,7 Go contenant 10 982 005 tweets et nous avons obtenu, à la fin du traitement, un fichier de 250Mo contenant 1 600 000 tweets.

Nous avons joint le workflow Knime robotFilter.knwf permettant de faire ce filtrage à l'archive de rendu du projet.

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 appliquons le premier algorithme mais nous supprimons avant tout traitement, les comptes robots que nous sommes susceptibles d'avoir dans nos données à l'aide du module Knime dit précédemment puis par autre filtre que nous décrirons plus tard. En effet, comme expliqué dans la partie précédente, 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.

MOI
MOI
MOI

6 Adaptation de l'algorithme de détection multi-échelle et intégration du filtrage des robots

6.1 Le tweet

Un tweet est décrit par :

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

Notre jeu de données de tweets s'étale sur 103 jours. Puisque nous voulons faire de la détection d'évènements (ponctuels) localisés - entre autres - dans le temps, nous avons découpé notre jeu de donnée initial en paquets jour par jour qui contiennent chacun plus de 15000 tweets.

6.2 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

6.3 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

6.4 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

a placer ou il faut
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

7 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

8 La détection dévènement avec DBScan

9 Résultats

10 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, 4