Exploitation des données du forum GOBAGES dans le cadre du projet FAUNA

Éric Quinton
eric.quinton@irstea.fr

UR *Écosystèmes aquatiques et changements globaux*
Document distribué sous licence CC-BY-SA
https://creativecommons.org/licenses/by-sa/3.0/fr/legalcode

Séminaire RBDD – 14 décembre 2017

Présentation du projet

- le forum gobages est un lieu d'échanges entre pêcheurs à la mouche
- objectif : rechercher en quoi les informations échangées peuvent refléter les évolutions de la pêche, notamment vis à vis de l'évolution des populations d'alose
- mode opératoire :
  - récupération de la base de données MySQL auprès des responsables du forum
    - engagement de notre part à anonymiser les données avant toute exploitation
  - transfert des informations dans une base de données PostgreSQL
  - traitement des données avec R, connexion directe avec la base PostgreSQL

Procédure utilisée pour anonymiser les données

- anonymisation des pseudos
  - basée sur l'utilisation de la fonction cryptographique de calcul d'empreinte sha256
    - le même pseudo donne toujours la même empreinte
    - fonction non réversible
  - empreinte limitée à une longueur de 8 caractères (suffisant pour le nombre de membres du forum)
  - permet de conserver la trace des différentes participations du même contributeur

Procédure utilisée pour anonymiser les données (2)

- anonymisation des dates
  - objectif : ne pas reconnaître un utilisateur à son rythme de vie (heure de connexion, par exemple)
  - précision mensuelle largement suffisante pour les traitements envisagés
  - toutes les dates sont ramenées au 1er jour du mois

```
update faunaforum.thread set
postusername = substring(digest(postusername, 'sha256')::
    varchar from 3 for 8),
dateline = date(date_part('year',dateline)::varchar || '-' ||
    date_part('month',dateline)::varchar || '-01')
```

Préparation des données - transformation en tableau des valeurs multiples

- le logiciel de gestion du forum stocke nativement les mots significatifs dans la base de données, en minuscules
- transformation en tableau PostgreSQL (une colonne peut contenir plusieurs valeurs)

```
update faunaforum.thread set keywords = string_to_array(
    keywords, ', ');
'alose, poisson' => ARRAY ['alose', 'poisson']
```

Fonctions natives PostgresSQL pour le traitement des tableaux

- unnest « explose » un tableau en autant de lignes que de valeurs :

```
unnest(ARRAY['alose','poisson'])
alose
poisson
```

Fonctions natives PostgresSQL pour le traitement des tableaux

- unnest « explose » un tableau en autant de lignes que de valeurs :

```
unnest(ARRAY['alose','poisson'])
alose
poisson
```

- Fonctions d'interrogation (dans le contexte du forum, elles sont ici toutes équivalentes)
  - any/some : permet de rechercher une valeur présente dans le tableau

```
select count(*) from faunaforum.thread
where 'alose' = any (keywords);
```

    - ★ any et some sont synonymes

Fonctions natives PostgresSQL pour le traitement des tableaux (2)

- @> – contient :

```
select count(*) from faunaforum.thread
where keywords @> array['alose']::varchar[];
```

- && – overlap : permet de rechercher si deux tableaux se superposent partiellement

```
select count(*) from faunaforum.thread
where keywords && array['alose']::varchar[];
```

- ces fonctions ne permettent pas une recherche partielle (opérateur like)

Mécanisme pour utiliser le comparateur like

```
select distinct on (threadid) keywords from (select *, unnest(
    keywords) keyword from faunaforum.thread) subreq
where keyword like '%alose%';
```

- une sous-requête est créée pour exploser chaque ligne en autant de lignes que de valeurs stockées dans *thread.keywords* (unnest)
- le résultat est réduit à une occurrence par clé (distinct on)
- affichage des mots-clés tels que stockés dans la base :

```
{switch,grammes,vends,canne,sierra,ideal,truite,lensemble,
    aloses,peche,20/22,valeur,sorties}
```

Exemple de requête complexe

```
/*
 * Nombre de nouveaux contributeurs par
 * annee pour les threads alose
 */
 with a as (
 select distinct extract (year from
     postfauna.dateline) as annee, username
from (select *, unnest(keywords) keyword
    from faunaforum.thread) subreq
join faunaforum.postfauna using (threadid)
where keyword like '%alose%')
select annee, count (distinct username) as
    nb_contrib
from a p1
where username not in (
select a1.username from a a1 where a1.annee
    < p1.annee
)
group by annee order by annee;
```

La première partie (with) sélectionne les contributeurs par année

annee	nb_contrib
2004	20
2005	62

La seconde compte les contributeurs par année qui ne figurent pas dans les années précédentes

Exploitation des données

- quelques chiffres :
  - environ 530 000 messages, répartis dans 54 000 fils de discussion, dans 18 forums différents
  - 7 600 contributeurs, dont 460 ont parlé d'aloses, dans 3 400 messages
- traitement statistique :
  - réalisé par Christophe Boschet avec R
  - 22 requêtes préparées par l'administrateur de bases de données et adaptées ensuite par le scientifique, lancées directement depuis RStudio
  - tableaux de données traités dans R avant calcul
    - suppression des fils de discussion sans intérêt pour le sujet