Data Mining

2. Text mining

Christophe Lalanne

Fall 2017

Analyse de données textuelles

Analyse d’email

Détection de spam

1

Analyse de données textuelles

Text mining

INTRO

Flux de données Twitter

TODO

- 140 caractères limite + hash tags
- analyse de sentiments
- Packages R : `{twitteR}`, `{streamR}`, `{sentiment}` (depr.), `{qdap}`, `{quanteda}`, ...

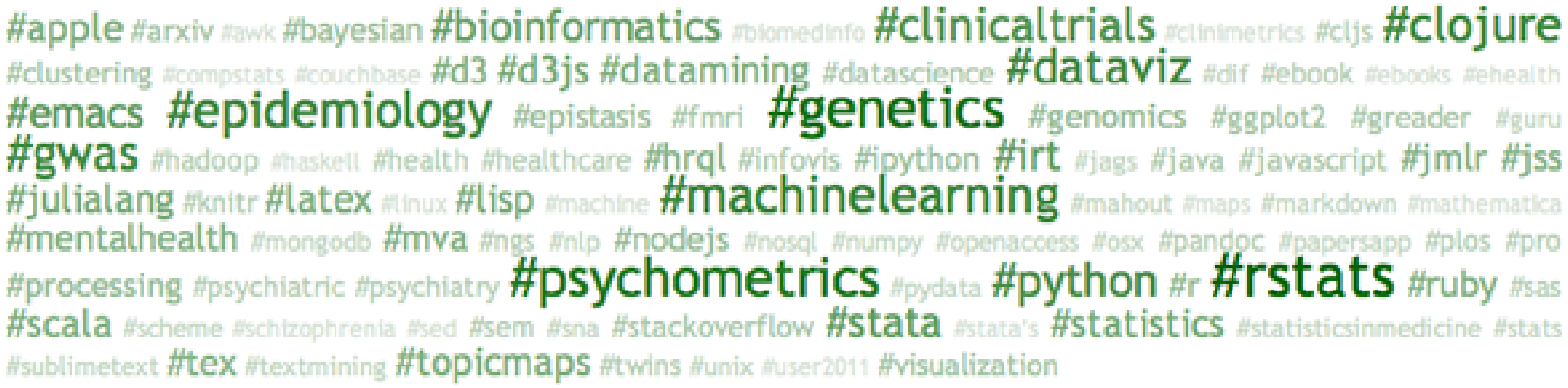
Tutoriel : https://sites.google.com/site/miningtwitter/references.

L'utilisation nécessite une autentification (OAuth).

Illustration

```
library(twitteR)
library(stringr)
tw = userTimeline("chlalanne", n = 1000)
find.tag = function(x)
  unlist(str_extract_all(x$getText(), "#[A-Za-z0-9]*"))
tags = lapply(tw, function(x) try(find.tag(x),
                silent = TRUE))
sort(table(unlist(tags)), decr = TRUE)
```

```
library(snippets)
wcl = table(unlist(tags))
names(wcl) = str_replace_all(names(wcl), "#", "")
cloud(wcl[wcl > 5])
```



Note : Le package `snippets` n'est plus disponible sur CRAN mais peut être installé depuis RForge.

Application

Text Mining the Complete Works of William Shakespeare

2

Analyse d’email

Analyse d'emails

Enron data set (enron.db, SQLite)

```
% sqlite enron.db
sqlite> .tables
Employee           EmployeeWithVars  MessageBase        Rec
EmployeeBase       Message           Recipient
sqlite> .schema Message
CREATE VIEW Message AS
SELECT
    mid,
    filename,
    datetime(unix_time, 'unixepoch') AS time,
    unix_time,
    subject,
    from_eid
FROM
    MessageBase;
```

```
sqlite> select * from Message limit 5;
1|taylor-m/sent/11|1998-11-13 04:07:00|910930020|Cd$ CME
2|taylor-m/sent/17|1998-11-19 07:19:00|911459940|Indemnif
3|taylor-m/sent/18|1998-11-19 08:24:00|911463840|Re: Inde
```

Importation de la base de données sous R :

```r
library(dplyr)
con <- src_sqlite("enron.db")
d <- tbl(con, "Message")
head(d, 3)
```

“Lazy” operation

```
y <- mutate(d, year = substr(time, 1, 4))
collect(y)
head(y, 3)
```

```
> summary(as.numeric(collect(select(y, year))[[1]]))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1998    2001    2001    2001    2001    2002
```

3

Détection de spam

Un problème supervisé

`ElemStatLearn::spam`

- 4601 mail classés en spam ou non
- fréquence relative de 57 mots-clés (pour chaque classe)
- `spam_names.txt`

```
if (george < 0.6) and (you > 1.5) then spam
  else email
```

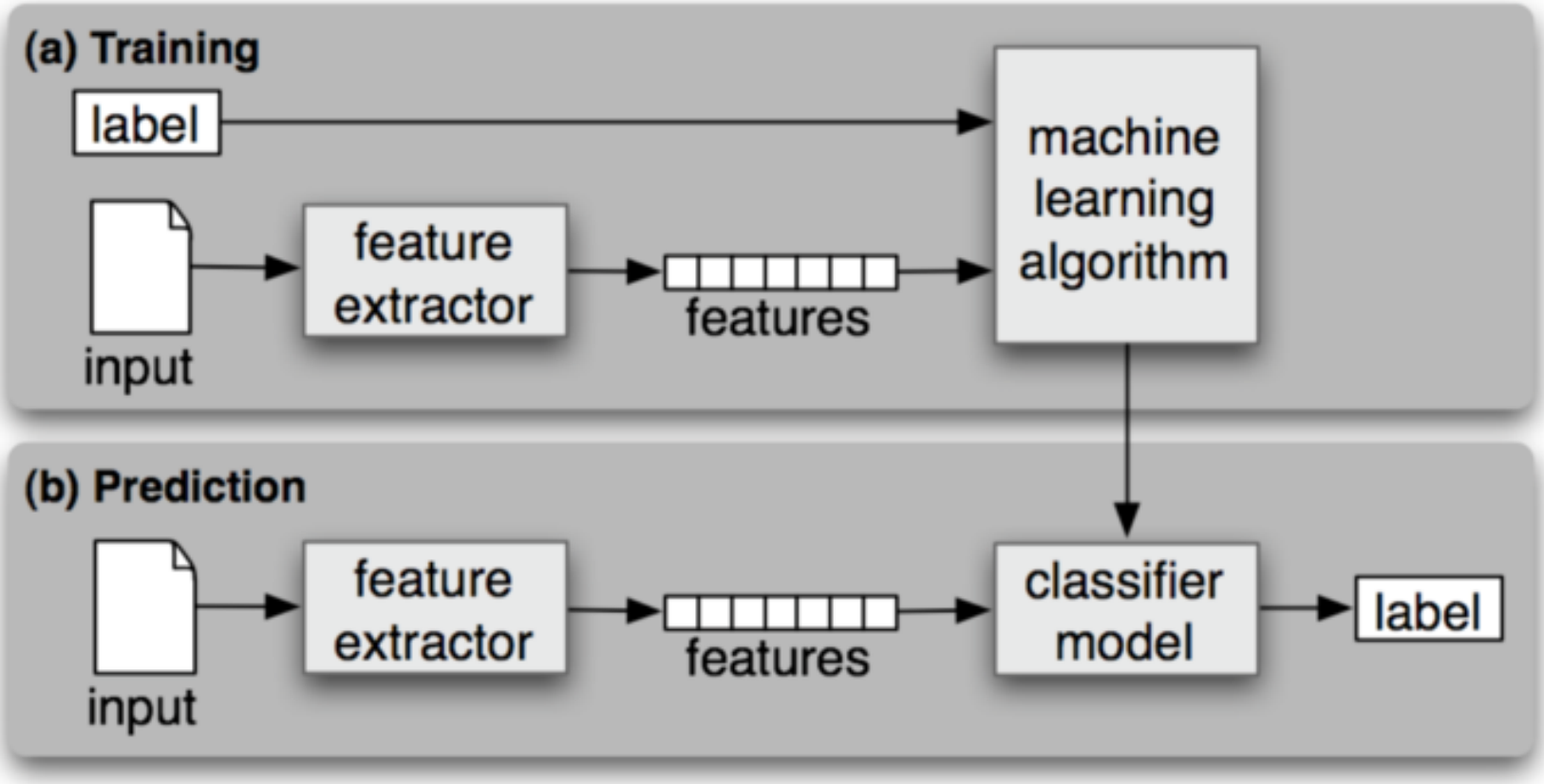


Figure 1: Source : NLTK documentation

Classifier naïf bayésien

$$\underset{c}{\text{argmax}}\, p(C = c) \prod_{i=1}^{n} p(F_i = f_i \mid C = c)$$

```r
data(spam, package = "ElemStatLearn")
library(klaR)

# set up a training sample
train.ind = sample(1:nrow(spam), ceiling(nrow(spam)*2/3))

# apply NB classifier
nb.res = NaiveBayes(spam ~ ., data = spam[train.ind,])
```

```
> # predict on holdout units
> nb.pred = predict(nb.res, spam[-train.ind,])

> # raw accuracy
> confusion.mat = table(nb.pred$class,
                        spam[-train.ind,"spam"])
> confusion.mat

       email spam
  email   519   34
  spam    420  560
> sum(diag(confusion.mat))/sum(confusion.mat)
[1] 0.7038487
```

References