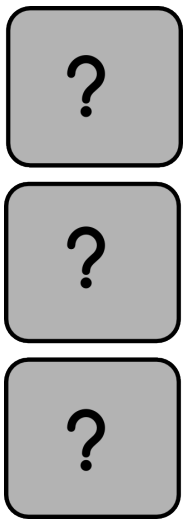
Machine Learning VI
Méthodes avancées

Nicolas Bourgeois

- chaque **topic** est une distribution de mots

- chaque **topic** est une distribution de mots
- chaque **document** est un mélange de quelques topics

- chaque **topic** est une distribution de mots
- chaque **document** est un mélange de quelques topics
- chaque **mot** est tiré au sort dans un topic

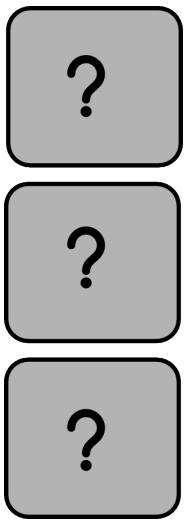



Sep
21

Foutage de gueule

Il paraît que le prof doc n'évalue pas les élèves.
Le prof doc passe un Capes ! Certificat d'Aptitude au Professorat de l'Enseignement du Second degré ! Comment peut-on enseigner sans évaluer ? Sans faire un suivi des élèves ? Là, j'aimerais bien qu'on m'explique….
J'enseigne et j'évalue mes élèves. Je suis prof, c'est mon boulot, faire passer des connaissances et m'assurer qu'elles sont comprises, acquises.
On l'entend partout, notre ministre le dit partout,

- Dans la réalité, on observe les documents



Sep
21

Foutage de gueule

Il paraît que le prof doc n'évalue pas les élèves.
Le prof doc passe un Capes ! Certificat d'Aptitude au Professorat de l'Enseignement du Second degré ! Comment peut-on enseigner sans évaluer ? Sans faire un suivi des élèves ? Là, j'aimerais bien qu'on m'explique....
J'enseigne et j'évalue mes élèves. Je suis prof, c'est mon boulot, faire passer des connaissances et m'assurer qu'elles sont comprises, acquises.
On l'entend partout, notre ministre le dit partout,

- Dans la réalité, on observe les documents
- Tout le reste constitue des **variables cachées**
- Nous cherchons à les retrouver, en inversant le processus génératif

```
6⟶liberte etat societe homme droit politique droi
economie selon monde pouvoir liberalisme justice se
7⟶triangle loys puzzle rectangle puzzles eleves z
maths dernier mathix emes reponses exemple isocele
8⟶documents republique document trace travail cor
place entreprise etats france problematique composi
9⟶jeux console jouer generation playstation stree
attends dois couleurs nouvelle premier semble live
10⟶eleves histoire premiere travail etaient ulysse
terre matiere niveau sujets possible princesse troi
11⟶opinion verite lire challenge opinions albums l
roman envie suite delivrer chapitre extraits person
12⟶ecole eleves dalton classe acte parents educati
```

```
le-data\bigbag\1-laviemoderne-1.txt 0 0 association 73
le-data\bigbag\1-laviemoderne-1.txt 1 1 populaire 84
le-data\bigbag\1-laviemoderne-1.txt 2 2 victime 5
le-data\bigbag\1-laviemoderne-1.txt 3 3 numerisme 82
le-data\bigbag\1-laviemoderne-1.txt 4 4 etude 82
le-data\bigbag\1-laviemoderne-1.txt 5 5 instructive 1
le-data\bigbag\1-laviemoderne-1.txt 6 6 enseignements 71
le-data\bigbag\1-laviemoderne-1.txt 7 7 soient 57
le-data\bigbag\1-laviemoderne-1.txt 8 8 scientifiques 1
le-data\bigbag\1-laviemoderne-1.txt 9 9 nouveaux 82
le-data\bigbag\1-laviemoderne-1.txt 10 10 vient 5
le-data\bigbag\1-laviemoderne-1.txt 11 11 publier 84
le-data\bigbag\1-laviemoderne-1.txt 12 12 afev 82
le-data\bigbag\1-laviemoderne-1.txt 13 13 numerique 82
le-data\bigbag\1-laviemoderne-1.txt 14 14 collegiens 1
```

Exercice

Exercice

Entraînez un topic model sur les éléments 500 à 1000 du dataset `fetch_20newsgroups`, pour une valeur de 10 topics. Affichez les 20 clefs principales des modèles fournis par le modèle.

Exercice

Codez une fonction de prétraitement pour enlever les scories, puis ré-entraînez le modèle et évaluez l'amélioration. Quelles autres pistes d'améliorations devraient être explorées ?

Résultat attendu (1)

Topic 0 : does god evidence don reason know think like use believe
Topic 1 : point right think people way just don law islam does
Topic 2 : cubs think suck good players time really numbers world league
Topic 3 : people said god say don just know like says did
Topic 4 : ax max a86 b8f pl 1t qq bhj qax bj
Topic 5 : like know use ve does want just good data bus
Topic 6 : game space shuttle play period blues ny power 12 14
Topic 7 : key keys encryption 20 chips 16 chip 15 10 use
Topic 8 : windows file nt sec ram swap use dos da disk
Topic 9 : db mov bh si cs byte al bl di maxbyte
Topic 0 : does god evidence don reason know think like use believe

Résultat attendu (2)

Topic 0 : windows know does believe think question people human good used
Topic 1 : government shuttle nasa space just encryption people attitude font satellite
Topic 2 : bits byte push picture loop offset east data west stuff
Topic 3 : evidence local like read people keys physical transfer company rate
Topic 4 : like really problem drives rotor problems hear lopez usual year
Topic 5 : cubs suck just like rights year people think printer
Topic 6 : just know good swap blues does think need drive power
Topic 7 : pick koresh like batf just thanks think know right space
Topic 8 : like people does just know said went going think good
Topic 9 : israel just want people islam israeli time little lebanese peace

Solution (1)

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation
from sklearn.datasets import fetch_20newsgroups
mini,maxi,n_features,n_components = 500,1000,1000,10
dataset = fetch_20newsgroups(shuffle=False,
    remove=('headers','footers','quotes'))
data_samples = dataset.data[mini:maxi]
tf_vectorizer = CountVectorizer(max_features=n_features,
    stop_words='english')
tf = tf_vectorizer.fit_transform(data_samples)
lda = LatentDirichletAllocation(n_components=n_components,
        learning_method='batch',max_iter=5)
lda.fit(tf)
tf_feature_names = tf_vectorizer.get_feature_names()
for j, topic in enumerate(lda.components_):
    message = "\nTopic␣{0}:␣".format(j)
    message += "␣".join([tf_feature_names[i]
        for i in topic.argsort()[:-11:-1]])
    print(message)
```

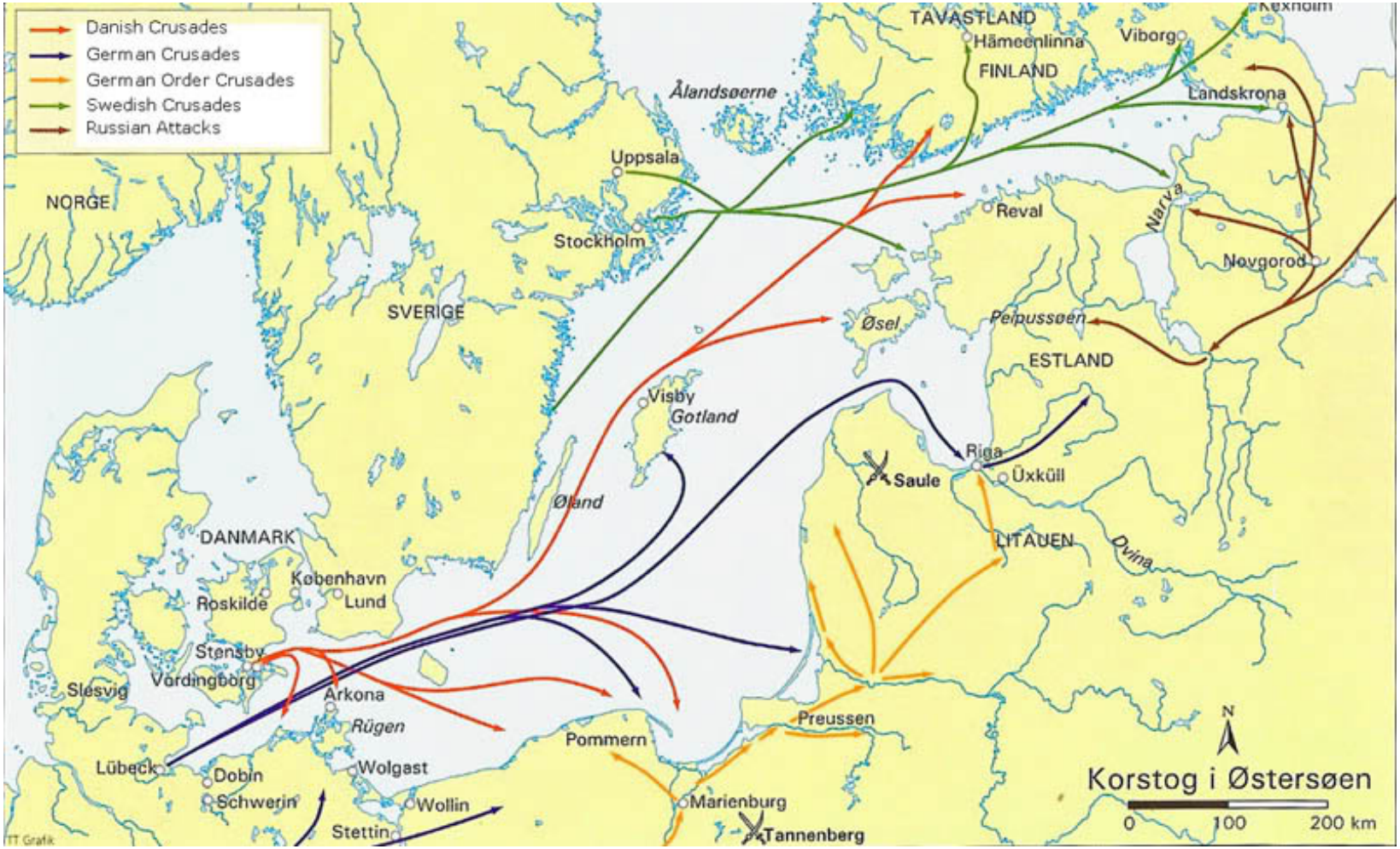
Solution (2)

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation
from sklearn.datasets import fetch_20newsgroups
mini,maxi,n_features,n_components = 500,1000,1000,10
dataset = fetch_20newsgroups(shuffle=False,
remove=('headers','footers','quotes'))
data_samples = dataset.data[mini:maxi]
for (i,t) in enumerate(data_samples):
    data_samples[i] = " ".join([x for x in t.split()
    if x.isalpha() and len(x)>3])
tf_vectorizer = CountVectorizer(max_features=n_features,
    stop_words='english')
tf = tf_vectorizer.fit_transform(data_samples)
lda = LatentDirichletAllocation(n_components=n_components,
        learning_method='batch',max_iter=5)
lda.fit(tf)
tf_feature_names = tf_vectorizer.get_feature_names()
for j, topic in enumerate(lda.components_):
    message = "\nTopic {0}: ".format(j)
    message += " ".join([tf_feature_names[i]
        for i in topic.argsort()[:-11:-1]])
    print(message)
```

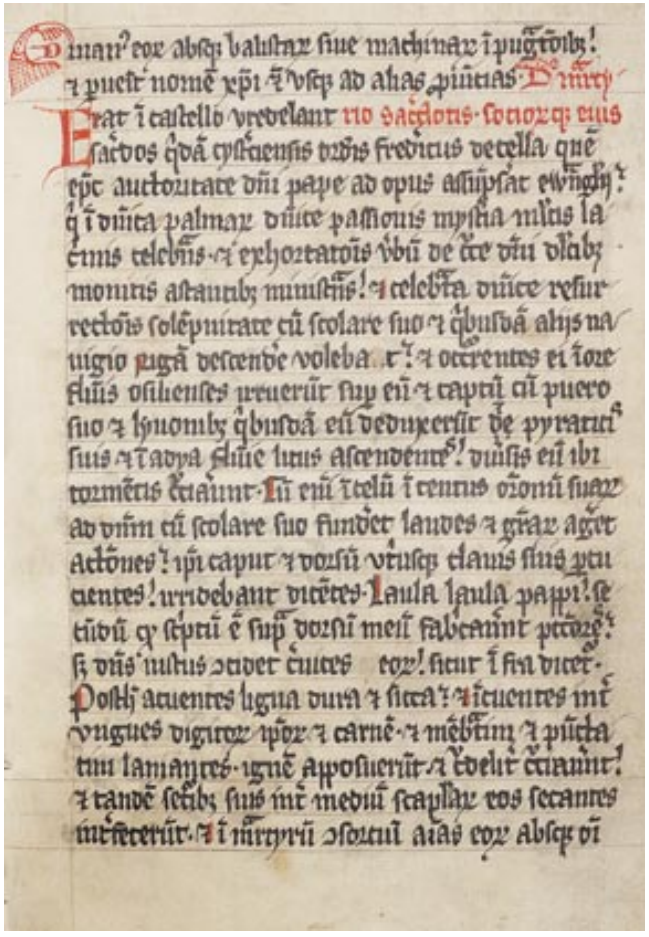
Qu’est-ce que c’est ?

Quelle est son origine ?

La Livonie, XII^{e}-XIII^{e} siècles



Le texte d'Henri



SCRIPTORES

RERUM GERMANICARUM

IN USUM SCHOLARUM

EX

MONUMENTIS GERMANIAE HISTORICIS

SEPARATIM EDITI

HEINRICI CHRONICON LIVONIAE

EDITIO ALTERA

RECOGNOVERUNT

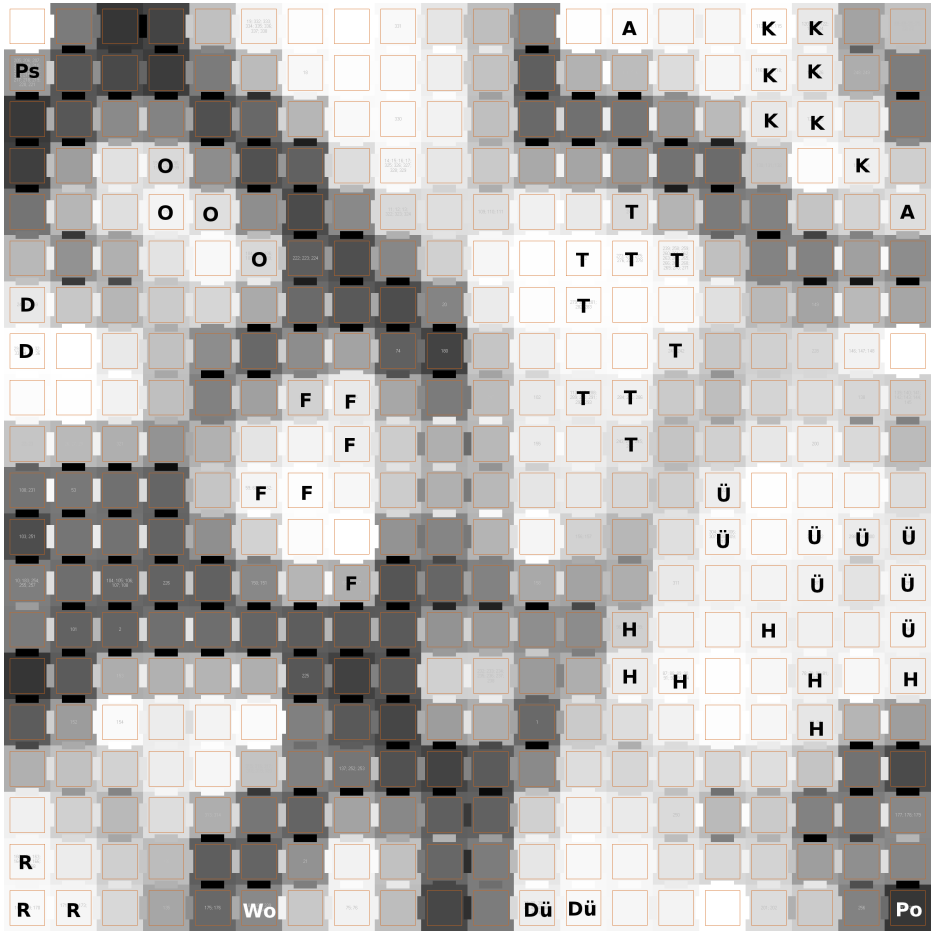
LEONID ARBUSOW (†) et ALBERTUS BAUER

HANNOVERAE

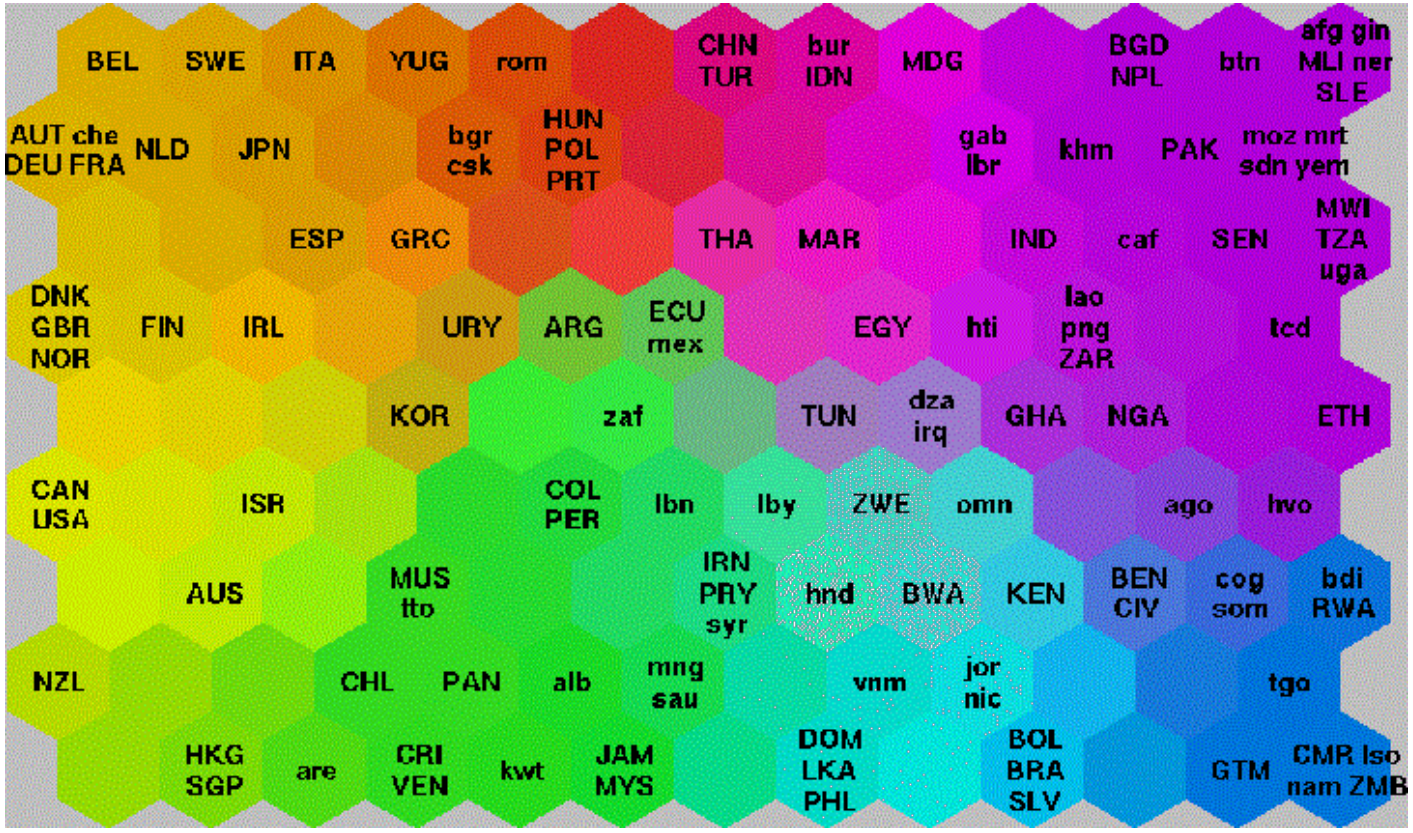
IMPENSIS BIBLIOPOLII HAHNIANI

1955

Carte de Kohonen



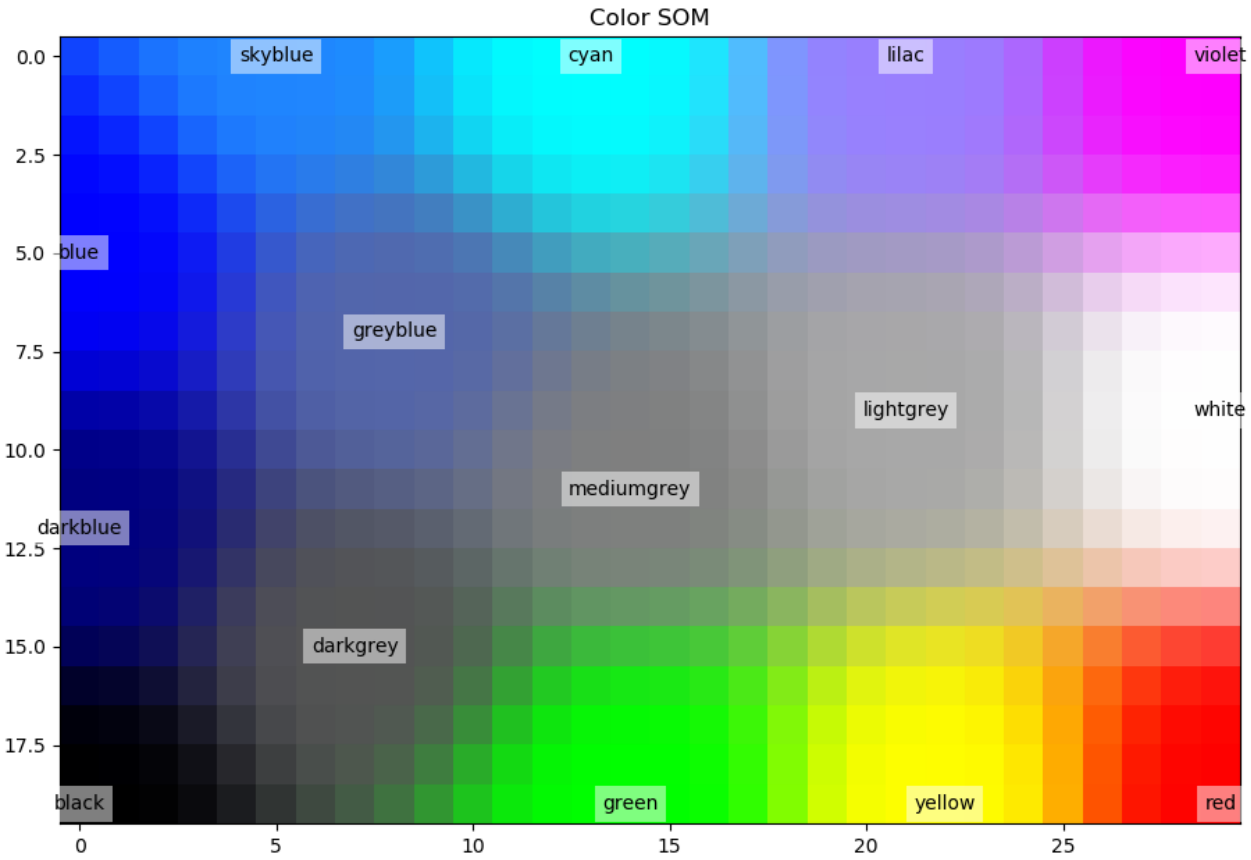
Kohonen Map


BEL
SWE
ITA
YUG
rom
CHN
TUR
bur
IDN
MDG
BGD
NPL
btn
afg gin
MLI ner
SLE
AUT che
DEU FRA
NLD
JPN
bgr
csk
HUN
POL
PRT
gab
lbr
khm
PAK
moz mrt
sdn yem
ESP
GRC
THA
MAR
IND
caf
SEN
MWI
TZA
uga
DNK
GBR
NOR
FIN
IRL
URY
ARG
ECU
mex
EGY
hti
lao
png
ZAR
tcd
KOR
zaf
TUN
dza
irq
GHA
NGA
ETH
CAN
USA
ISR
COL
PER
lbn
lby
ZWE
omn
ago
hvo
AUS
MUS
tto
IRN
PRY
syr
hnd
BWA
KEN
BEN
CIV
cog
som
bdi
RWA
NZL
CHL
PAN
alb
mng
sau
vnm
jor
nic
tgo
HKG
SGP
are
CRI
VEN
kwt
JAM
MYS
DOM
LKA
PHL
BOL
BRA
SLV
GTM
CMR lso
nam ZMB

Kohonen Map

```
from som1 import *
colors = np.array(
    [[0., 0., 0.],[0., 0., 1.],[0., 0., 0.5],[0.125, 0.529, 1.0],
    [0.33, 0.4, 0.67],[0.6, 0.5, 1.0],[0., 1., 0.],
    [1., 0., 0.],[0., 1., 1.],[1., 0., 1.],[1., 1., 0.],
    [1., 1., 1.],[.33, .33, .33],[.5, .5, .5],[.66, .66, .66]])
color_names = \
      ['black', 'blue', 'darkblue', 'skyblue',
       'greyblue', 'lilac', 'green', 'red',
       'cyan', 'violet', 'yellow', 'white',
       'darkgrey', 'mediumgrey', 'lightgrey']
som = SOM(20, 30, 3, 400)
som.train(colors)
image_grid = som.get_centroids()
mapped = som.map_vects(colors)
plt.imshow(image_grid)
plt.title('Color␣SOM')
for i, m in enumerate(mapped):
      plt.text(m[1], m[0], color_names[i], ha='center', va='center',
      bbox=dict(facecolor='white', alpha=0.5, lw=0))
plt.show()
```

Kohonen Map

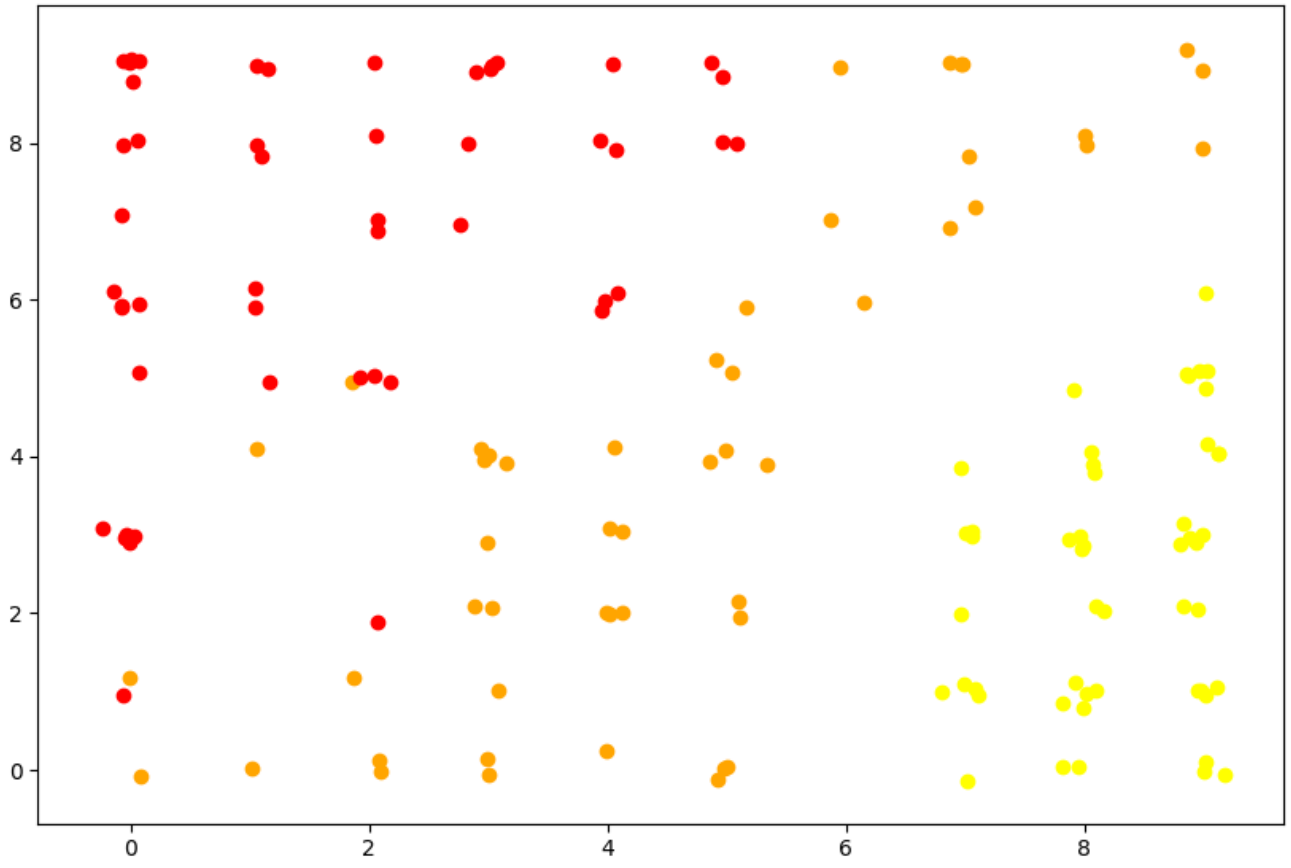

Color SOM
skyblue
cyan
lilac
violet
blue
greyblue
lightgrey
white
mediumgrey
darkblue
darkgrey
black
green
yellow
red
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
0
5
10
15
20
25

Exercice

Exercice

Installez tensorfow et placez `C4/som1.py` *dans le répertoire courant. Entraînez une carte auto-organisatrice sur* `C4/data1.csv` *avec une grille carrée* 10×10*. Affichez le résultat.*

Résultat attendu



Solution

```
from som1 import *
from sklearn.datasets import load_iris
from numpy.random import normal
iris = load_iris()
def color(x):
    return{2:'red',1:'orange',0:'yellow'}[x]
X,Y = iris.data, iris.target
som = SOM(10, 10, 4, 50)
som.train(X)
image_grid = som.get_centroids()
mapped = som.map_vects(X)
for i in range(len(Y)):
    plt.scatter(mapped[i][0]+normal(scale=0.1),
        mapped[i][1]+normal(scale=0.1),c=color(Y[i]))
plt.show()
```

Exercice

Exercice

Produisez un schéma de test semblable à celui que je vous ai montré à la séance 4 à partir des générateurs de la librairie datasets (et de matplotlib bien sûr).

Résultat attendu