INF5021 Machine Learning

Micro tutoriel Python

Lionel PREVOST, Directeur de Recherche LDR Lab

(lionel.prevost@esiea.fr)

TP ET PROJET

- Travaux pratiques réalisés avec scikit-learn, en langage Python
- Scikit-learn
  - Librairie Python pour effectuer de l'apprentissage automatique
  - Inclut la plupart des méthodes vues en classe
  - Projet ouvert (*open source*)
  - Site Web : `http://scikit-learn.org`
- Option 1 (**privilégiée**) : Anaconda (`https://www.anaconda.com/`)
  - Distribution contenant un ensemble de 1000+ librairies de calcul scientifique
  - Disponible sous Windows, Mac et Linux
  - Inclut Numpy, Scipy, matplotlib, scikit-learn, etc.

TP : créer et visualiser des données

```
#créer les données
from sklearn.datasets import make_blobs
X,y = datasets.make_blobs(n_samples=1000, n_features=2, centers=2, cluster_std=1.0,
center_box=(-10.0, 10.0))

#importer les librairies
import numpy
from matplotlib import pyplot

#visualiser les données
colors = numpy.array([x for x in "bgrcmyk"])
pyplot.scatter(X[:, 0], X[:, 1], color=colors[y].tolist(), s=10)
pyplot.show()
```

(*) https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_blobs.html

(…) entraîner et évaluer un modèle

```
from sklearn import svm, metrics

#entraîner un modèle (svm linéaire)
linsvm = svm.LinearSVC(C=10)
linsvm.fit(X, y)

#évaluer le modèle
ypred = linsvm.predict(X)

#taux d'erreur
err_train = 1 - metrics.accuracy_score (ypred, y)
print("Train error: %.3f" % err_train)

#matrice de confusion
metrics.confusion_matrix(y, ypred)
```

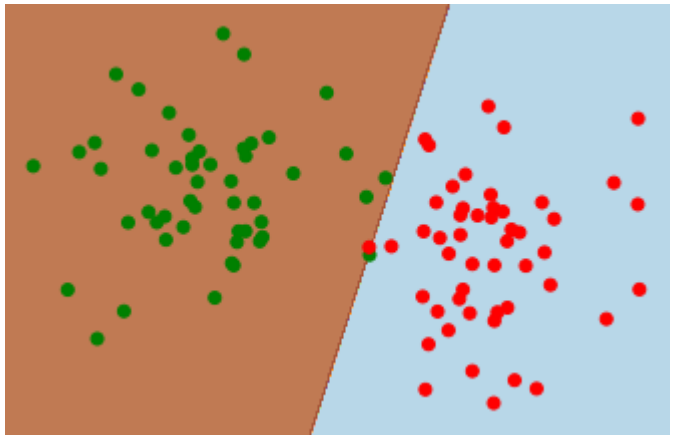
```
Out[109]:
Train error: 0.020
array([[49, 1],
       [ 1, 49]],
```

(*) https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

(…) : visualiser des frontières

```
# Créer un mesh
h = .02 # Espacement du mesh
x_min, x_max = X[:, 0].min()*1.1, X[:, 0].max()*1.1
y_min, y_max = X[:, 1].min()*1.1, X[:, 1].max()*1.1
xx, yy = numpy.meshgrid(numpy.arange(x_min, x_max, h),
numpy.arange(y_min, y_max, h))

#afficher les frontières
Y = model.predict(numpy.c_[xx.ravel(), yy.ravel()])
Y = Y.reshape(xx.shape)
pyplot.contourf(xx, yy, Y, cmap=pyplot.cm.Paired, alpha=0.8)
pyplot.scatter(X[:, 0], X[:, 1], cmap=pyplot.cm.Paired, color=colors[y].tolist())
pyplot.xlim(xx.min(), xx.max())
pyplot.ylim(yy.min(), yy.max())
pyplot.show()
```



(…) : cross-validation – grid search

```
from sklearn import model_selection

#validation croisée
X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y, train_size=0.7, test_size=0.3)
(voir aussi StratifiedShuffleSplit)

#grid search (apprentissage d'un SVM-RBF)
C_2d_range = [1e-2, 1, 1e2]
gamma_2d_range = [1e-1, 1, 1e1]
classifiers = []
for C in C_2d_range:
    for gamma in gamma_2d_range:
        rbfsvm = SVC(C=C, gamma=gamma)
        rbfsvm.fit(X2D, y)
        classifiers.append((C, gamma, rbfsvm))
```