Data Mining

Grid, Boosting / Xarxes Neuronals aplicats amb scikit-learn amb noves dades

http://www.salle.url.edu

Enginyeria La Salle – Universitat Ramon Llull

Pràctica 3 (Week 13): Aprendre a categoritzar imatges de dígits i classificació de grups de notícies

Objectiu

Els principals objectius d'aquesta pràctica son:

- Extendre els coneixements del paquet de Python scikit-learn
- Agafar fluïdesa amb la selecció, normalització, preprocés i cerca d'atributs
- Conèixer les diferents implementacions de l'algorisme d'aprenentatge basat amb exemples IBL
- Treballar amb el concepte d'optimització d'un *learner*.

Requirements

La pràctica es pot fer en qualsevol sistema operatiu: Windows / Mac OS X / Linux però es necessita d'un intèrpret de Python. Si no esteu acostumats a l'entorn Python adreceu-vos a la guia penjada a l'e-study:
https://estudy.salle.url.edu/mod/resource/view.php?id=463912

Els recursos per fer la pràctica es troben a:
https://estudy.salle.url.edu/mod/resource/view.php?id=463966

El pou es troba a:
https://estudy.salle.url.edu/mod/assign/view.php?id=463967

Si teniu algun problema, comenceu una discussió a :
https://estudy.salle.url.edu/mod/forum/view.php?id=463960

També recomano l'ús d'un editor avançat de programació tals com:

Sublime Text Editor : `http://www.sublimetext.com/`

Notepad++ : `http://notepad-plus-plus.org/`

Temps estimat: 2 hores. Màxim 3 hores

Deadline: 03 de Febrer del 2020 a les 23:59:59 CET (No hi haurà pròrrogues)

Descripció

La pràctica proporciona els següents recursos:

- **Assignment3.pdf:** Enunciat de la pràctica
- **Recordeu que a través de pip o easy-install s'instalen els paquests numpy, scipy i scikit-learn**

Entrega:

- **Es demana un document en format PDF donant resposta a les qüestions plantejades en aquesta pràctica, acompanyat del codi Python emprat per la mateixa**

Funcionament:

(Recapitulació de la pràctica anterior --- Recordeu que estàvem en el Dígits Dataset)

1. **Fes servir validació creuada per estimar el nombre òptim de veïns K**
   - **Com es pot fer un procés de cerca de paràmetres?**
   - **Petit tutorial sobre GridSearchCV:**

http://scikit-learn.org/stable/modules/generated/sklearn.grid_search.GridSearchCV.html

```
#Import the required packages
import sklearn.grid_search
import sklearn.cross_validation

#Generate possible values of the exploration Grid
k = np.arange(20)+1
```

```
#Infer all the exploratory surface into parameters struct
parameters = {'n_neighbors' : k}

#Create Learner Factory
knearest=sklearn.neighbors.KNeighborsClassifier()

#Instantiate a GridSearch with the a) Learner Factory b) Exploratory parameters c)
CrossValidation param
clf = sklearn.grid_search.GridSearchCV(knearest,parameters,cv=10)

#Perform exploratory grid search over TrainingData
clf.fit(X_train,Y_train)

#Compute Test Accuracy with the already defined function (it has to be adapted)
compute_test(x_test=X_Test,y_test = Y_test, clf = clf, cv =10)

#Obtrain the point of the grid that yielded the best train-accuracy
clf.best_params_['n_neighbors']
```

- Ara que ja tens el mètode d'aprenentatge, ja pots fer una cerca sobre:
  - Mètode de transformació de l'entrada
  - Nombre de dimensions d'entrada del classificador
  - Nombre de veïns a emprar
  - Esquema de ponderació de l'algorisme
  - Et pots ajudar de:

➢ Fes una cerca de paràmetres per tenir el millor classificador possible, printa o visualitza gràficament la evolució de la cerca i mostra els resultats i com hi has arribat.

2. Fer servir un altre learner (és recomana Xarxes Neuronals i algun mètode Ensemble)

➢ Petit tutorial sobre Neural Networks:

http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

```
from sklearn.neural_network import MLPClassifier

#Make a 2-hidden-layer Neural Network with 100 Rectifier Units
with a learning_rate of 0.02 e an 10 iterations
nn = MLPClassifier(hidden_layer_sizes(100,100,),
activation='relu',solver='sgd',learning_rate='constant',
learning_rate_init=0.02, n_iter=10)

#Fit the data
nn.fit(X_train, Y_train)
```

Per a ensemble methods:

http://scikit-learn.org/stable/modules/ensemble.html

The following example shows how to fit an AdaBoost classifier with 100 weak learners:

```
from sklearn.ensemble import AdaBoostClassifier

clf = AdaBoostClassifier(n_estimators=100)
#Fit the data
clf.fit(X_train, Y_train)
```

3. **Treballar amb un altre conjunt de dades**
   - **Es recomana treballar amb:**

     https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html

     o

     https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.html#sklearn.datasets.fetch_20newsgroups

   - (És important el punt 5.5.2.2 que parla de la conversió de text amb vectors)
4. **Conclusions (ampliades)**
   - Reporta les conclusions de la pràctica discutint, almenys, els següents apartats amb exemples i judicis

   a) **Com afecta el Learner a l'eficàcia dels resultats**

   b) **Com canvia l'eficàcia del learner en funció del problema**

   c) **Valoracions de NN o Ensemble Method?**

Felicitats! No només has après Data Mining amb Scikit-Learn sinó que també has après a reconèixer dígits, classificar textos, entendre diferents *learners* i treballar en processos de cerca!