第 2 章

Python 入門 (2)

2.1　機械学習を行うクラスの使い方

Scikit-learn[*1]は、識別・回帰・クラスタリング・次元削減などのツールが実装されたパッケージです。各アルゴリズムはクラスとして設計されていて、以下の共通した基本仕様からなります。

- コンストラクタ：クラスの初期化
  引数はアルゴリズムのパラメータ。
- fit() メソッド：学習
  引数は学習データと正解ラベル。必要に応じてデータに依存したパラメータ。
- predict() メソッド：予測
  学習済みのインスタンスに対して、予測対象のデータを引数として与えると、結果を返す。

上記仕様を満たした 1-NN 法を実装したクラスの例を以下に示します。

```
class NN(object):
    """simple 1-NN classifier"""
    def __init__(self):
        self.X = None
        self.y = None

    def fit(self, X, y):
        self.X = X
        self.y = y

    def predict(self, x):
        return y[np.argmin(np.linalg.norm(
                    self.X - np.tile(x,(self.y.size,1)), axis=1))]
```

2.2　交差確認法

1 つ抜き法の実装は、行列から位置を指定して要素を削除する np.delete を用いて識別したいデータを除いた学習データを作成し、fit メソッド、predict メソッドを順に呼んで識別を行います。

```
clf = NN()
for i in range(y.size):
    x = X[i]
    X2 = np.delete(X, i, axis=0)
    y2 = np.delete(y, i)
```

[*1] http://scikit-learn.org/

```
clf.fit(X2,y2)
print(clf.predict(x), end =' ')
```

10-fold CV のように学習データを分割するときは、クラスバランスを考えて分割する必要があり、コードが複雑になります。sklearn.model_selection.cross_val_score は、識別器・特徴ベクトル・正解ラベル・分割数を与えるだけで、クラスバランスを考慮した分割を行って、評価した値を返します。

実践演習 2-1

教科書 2.2 節から 2.5 節の手順を Python で実行せよ。ただし、データのスケール調整については正規化ではなく標準化（平均値 0、標準偏差 1）を行え。

実践演習 2-2

上記手順に、主成分分析（2 次元に変換）とデータ可視化の処理を加えよ。