

# Scikit-learn ゼミ

## 1.3. Stochastic Gradient Descent

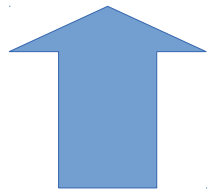
### 1.3.1 Classification

新納浩幸

# Stochastic Gradient Descent (SGD)

## 確率的勾配降下法

機械学習手法ではモデルのパラメータを求める際に、期待損失を最小化するようなパラメータを求めることが多い、最急勾配法を利用して求める



ある種の条件が満たされると、効率的に求めることができる、オンラインで可能、、、SGD

線型 SVM やロジスティック回帰のパラメータ推定に利用可能

# 利用可能な状況

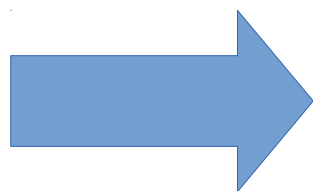
期待損失が各データの誤差の総和

例) 2乗誤差ならデータ全体の誤差は

$$E_N = \sum_{i=1}^N E_i = \sum_{i=1}^N |f(\mathbf{x}_i) - y_i|^2$$

最急降下法

$$\theta \leftarrow \theta - \eta \nabla E_N$$



SGD

$$\theta \leftarrow \theta - \eta_i \nabla E_i$$

# 特徴

## 長所

- ・ 効率的
- ・ 実装が容易

## 短所

- ・ メタパラメータが多い
- ・ 素性のスケールに sensitive

# SGDClassifier

パラメータの学習に SGD を使った分類器学習

分類器としては線型 SVM と ロジスティック回帰 を紹介



loss="hinge"



loss="log"

Multi-class も大丈夫、one vs. other 法を利用

動かしても学習できた分類器自体はあまり変化ない  
大規模、スパースデータのために学習が高速なのが  
特徴

# 確認

```
from sklearn import datasets
from sklearn.linear_model import SGDClassifier

iris = datasets.load_iris()
X=iris.data
Y=iris.target

clf = SGDClassifier(loss="hinge", penalty="l2")
clf.fit(X, Y)
print clf.score(X, Y)
```

使い方は上記

この例では正解率はめちゃくちゃ . . .

データがランダムに入っていないとダメ . . .

適当にシャッフルすると少しよくなるけど SVM の方がよい

データが小さいときは使わないほうがよいと思います