回帰:リコメンド

趙新宇

回帰を用いて物件価格

- 特徴量の要素は一つの場合
  例:「家の値段」 特徴量:部屋の数

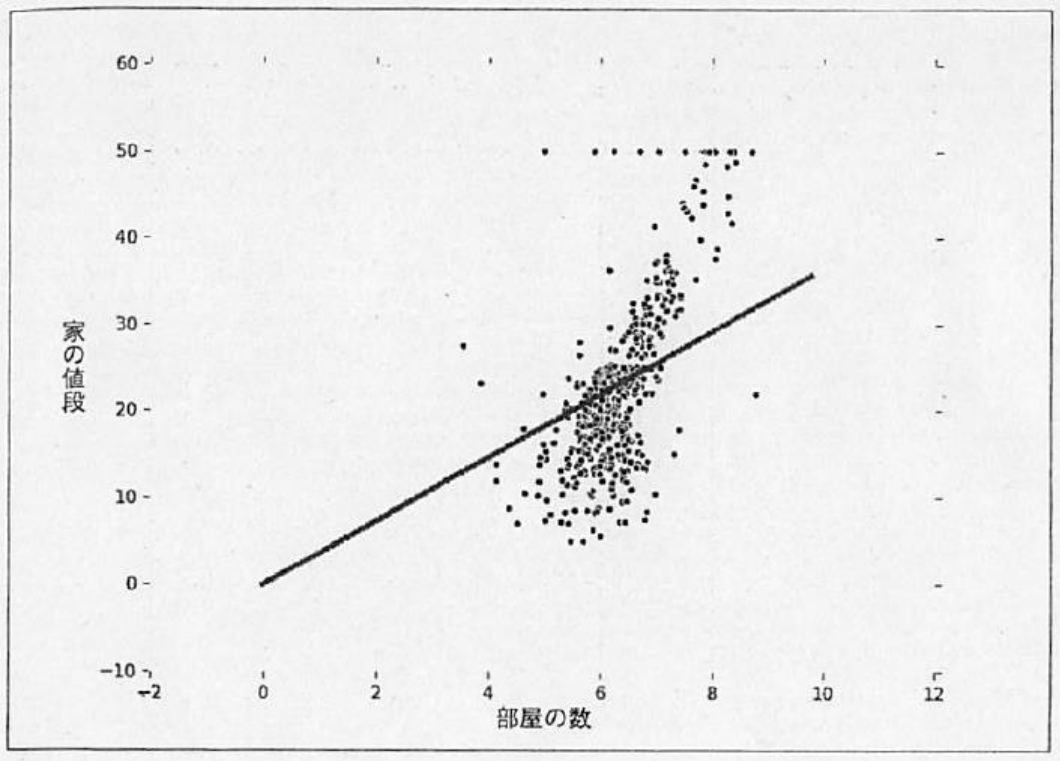
```
from sklearn.datasets import load_boston
boston = load_boston()

from matplotlib import pyplos as plt
plt.scatter(boston.data[:,5],boston.target,color='r')

import numpy as np
x=boston.data[:,5]
x=np.array([[v] for v in x])
```

y = boston.target

slope,_,_,_ = np.linalg.lstsq(x,y)



バイアス項を追加

```
x = boston.data[:,5]
x = np.array([[v,1] for v in x]) # [v]の代わりに[v,1]を使用
y = boston.target
slope,_,_,_ = np.linalg.lstsq(x,y)

(slope,bias),total_error,_,_=np.linalg.lstsq(x,y)
rmse = np.sqrt(total_error[0]/len(x)
```

ここで求める値は、平均二乗平方根誤差（root mean squared error:RMSE）と言います。

平均二乗誤差の平方根 (RMSE)定義

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{n} e(t)^2}$$

RMSE
式のキー

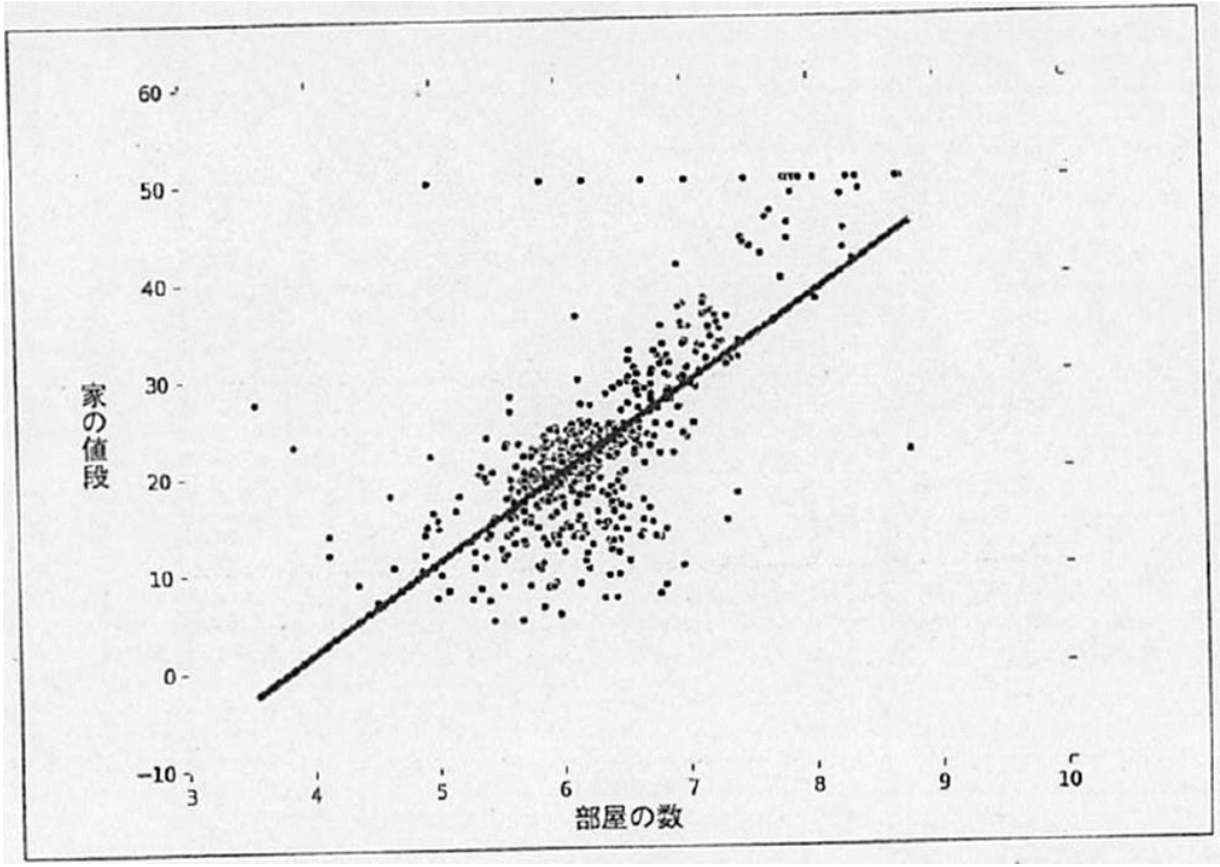
$e(t) = V(t) - P(t)$

$V(t)$ = 実績値

$P(t)$ = 予測値

n = 事後的期間の数

バイアス項を用いなかった場合、RMSEは7.6であるのに対して、バイアス項を追加した場合は6.6に改善されます。



多次元回帰

特徴量の要素は一つではないの場合

```
x = boston.data
x = np.array([[np.concatenate(v,[1]) for v in boston.data])
y = boston.target
s,total_error,_,_ = np.linalg.lstsq(x,y)
```

結果を見ると、RMSEの値は4.7になりました。

回帰における交差検定

- Kflodクラスを用いて10分割の交差検定を行い、線形回帰の汎化に対する能力について評価を行います。

```
from sklearn.cross_validation import Kfold
Kf = Kfold(len(x), n_folds=10)
err = 0
for train,tesr in kf:
    lr.fit(x[train],y[train](
    p = map(lr.predict, x[test])
    e = p-y[test]
    err += np.sum(e*e)
rmse_10cv = np.sqrt(err/len(x))
Print('RMSE on 10-fold CV: {} ' .format(rmse_10cv))
```

回帰における交差検定

- 交差検定を用いて、結果は5.6になります

- 交差検定を用いた場合のほうが、値段を予測する汎化能力について
、より正しい評価を行っていると言えます

罰則付き回帰

- 最小二乗法による回帰から派生した手法で重要なまのは、罰則付き回帰（penalized regression）と呼ばれるものです

- 「罰則」の意味は、パラメータが過剰に適合することに対して、罰則を追加するとです。

L1、L2罰則項

- 回帰で用いる罰則項として、L1罰則項、L2罰則項があります。
- L1罰則項は係数の絶対値の和を用います。「Lasso回帰」
- L2罰則項は係数の二乗和を用います。「Ridge回帰」
- この二つを合わせるのは「Elastic net」と呼ばれます。

L1、L2罰則項

$$\vec{b}^* = \arg\min_{\vec{b}} (y - X\vec{b})^2$$

$$\vec{b}^* = \arg\min_{\vec{b}} (y - X\vec{b})^2 + \lambda \sum_i |b_i|$$

$$\vec{b}^* = \arg\min_{\vec{b}} (y - X\vec{b})^2 + \lambda \sum_i b_i^2$$

scikit-learnのLassoとElastic netを使用する

- from sklearn.linear_model import ElasiticNet
- en = ElasticNet(fit_intercept=True, alpha=0.5)

- 訓練誤差は5.0（前は4.6）
- 交差検定を用いた誤差5.4（5.6）

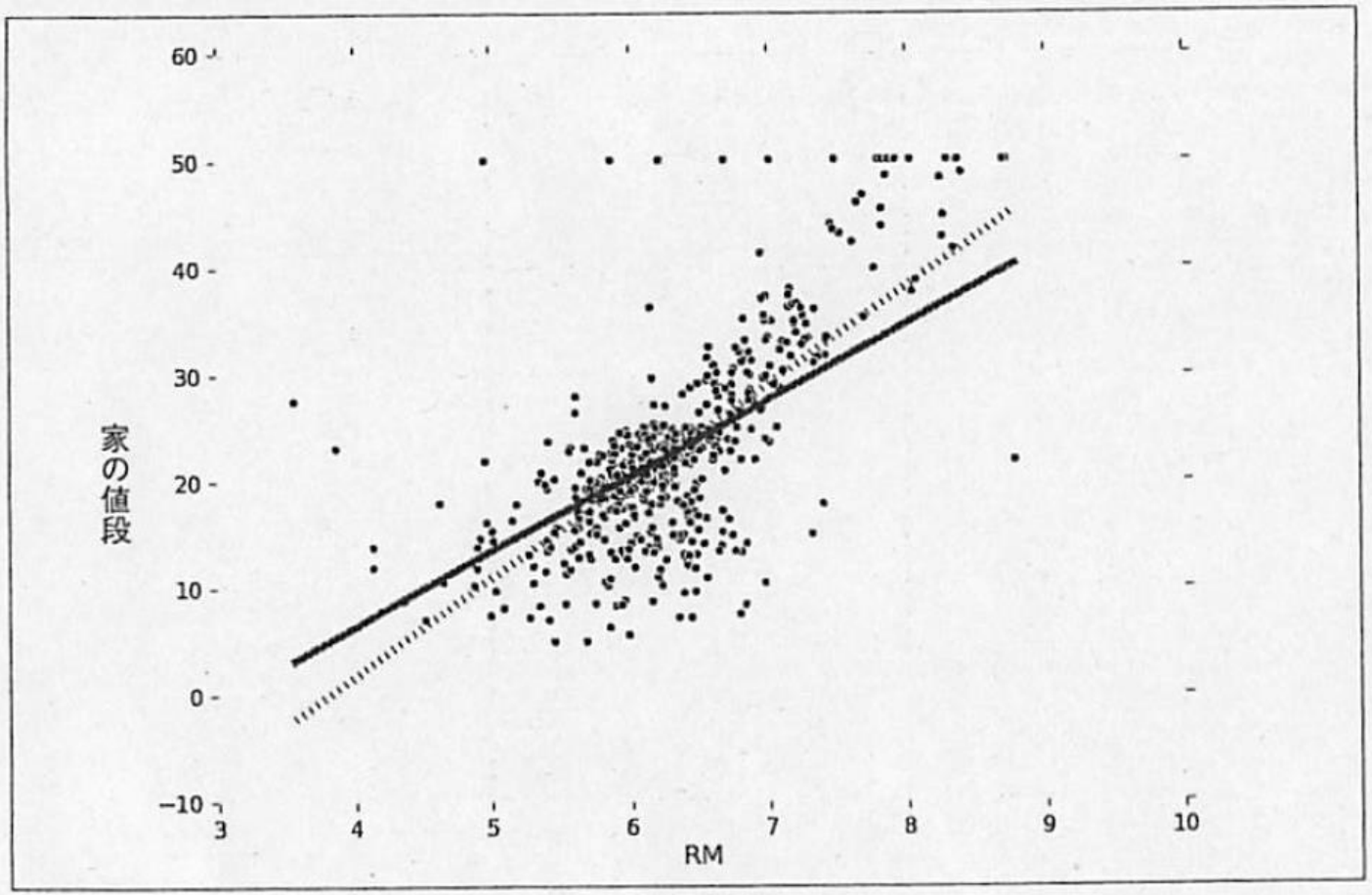


図7-3

PがNより大きい場合

- Pは特徴量の数
- Nはサンプルデータの数

- PがNより大きい場合、訓練誤差が0になるような回帰係数を求めることができる。
- しかし、訓練誤差が0であるということは、汎化能力があることを意味しません。実際、汎化能力は極めて低くなります。

ハイパラメータの賢い設定方法

- パラメータは大きい値を設定した場合、それは未学習の可能性が高くなります。極端な場合として、学習の結果、全ての係数が0になります
- 極端に小さい値を設定した場合、それは過学習であり、最小二乗法に近づくことになります。過学習の状態では汎化能力が低くなります。

- 一般的な解決策は交差検定を用いて、パラメータについて取り得る候補を用意し、その中から交差検定を用いて最適な値を一つ選び出します

ハイパラメータの賢い設定方法

- 汎化能力を評価し最適なパラメータを選択するためには、二段階の交差検定を用いなければならない。

- 一段階目で行う交差検定は、汎化能力を評価するため。
- 二段階目で行う交差検定は、最適なパラメータを得るために用います。