先端データ解析論レポート第二回

48196635 桑原亮介

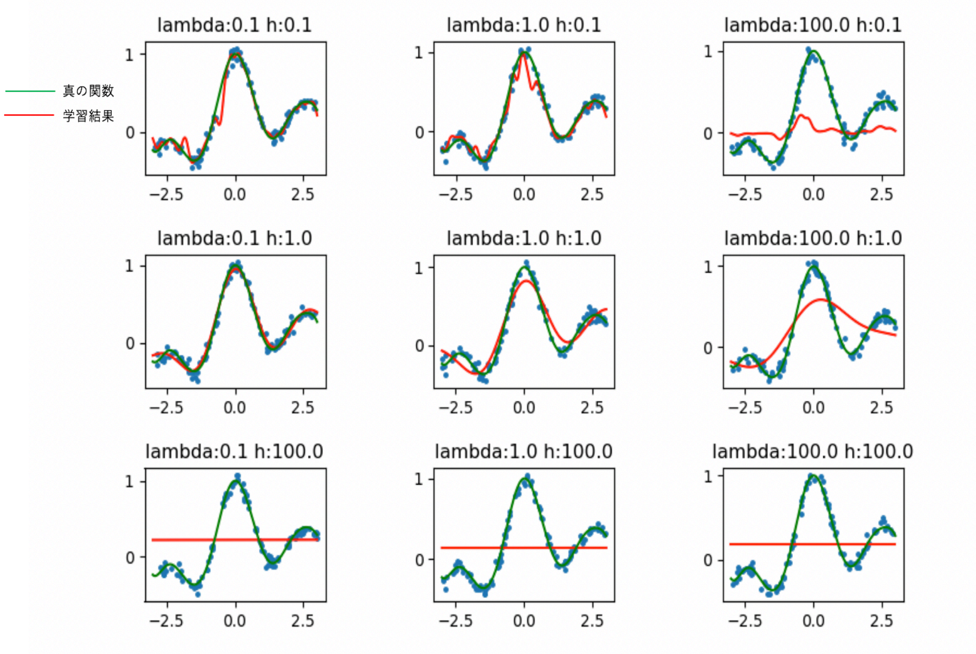
2019 年 4 月 22 日

1　宿題 1

目的：ガウスカーネルモデルに対する l_2-正則化を用いた最小二乗回帰の交差確認を実装し、正則化パラメタとガウス幅を決定する。

方法 : ガウス幅/正則化パラメタをそれぞれ $\{0.1, 1.0, 100.0\}$ に設定した際の計 9 パターンで予測式を立て、正解データとの最小二乗誤差のもっとも低い組み合わせを決定する。

結果：ガウス幅 1.0、正則パラメタ 0.1 の時に誤差が最小となった。



実装コード：

```
import numpy as np
import matplotlib.pyplot as plt
```

```python
from sklearn.utils import shuffle
np.random.seed(0)
ans=10**9
ans_h=10**9
ans_l=10**9



#データを生成する
def generate_sample(xmin=-3,xmax=3,sample_size=50):
  x=np.linspace(start=xmin,stop=xmax,num=sample_size)
  pix=np.pi*x
  target=np.sin(pix)/pix+0.1*x
  noise=0.05*np.random.normal(loc= 0.,scale=1.,size=sample_size)
  return x,target+noise

#ガウスカーネル
def calc_design_matrix(x_train,x_test,y_train,h,lam):
  global theta
  #theta を算出
  k=np.exp(-(x_train[None]-x_train[:,None])**2/(2*h**2))
  theta=np.linalg.solve(k.T.dot(k)+lam*np.identity(len(k)),k.T.dot(y_train[:,None]))
  #データからカーネル関数設計
  K = np.exp(-(x_train[None]-x_test[:,None])**2/(2*h**2))
  return K.dot(theta) #予測値を返す

#二乗誤差
def sqared_error(y,t):
  y=y.flatten()
  t=t.flatten()
  return np.sum(((y-t)**2))

def cross_validation(x,y,n,h,lam,i):
  index=x.size//n
  tmp=0
  x,y=shuffle(x,y)

  for i in range(n):
    #データ分割
    x_test=np.hstack((x[:index*i],x[index*(i+1):]))
    x_train=np.array(x[i*index:(i+1)*index])
    y_test=np.hstack((y[:index*i],y[index*(i+1):]))
    y_train=y[i*index:(i+1)*index]
    #予測値&誤差を返す
    idx=np.argsort(x_test)
    x_test=x_test[idx]
    y_test=y_test[idx]
    prediction=calc_design_matrix(x_train,x_test,y_train,h,lam)
```

```
    truth=calc_design_matrix(x_test,x_test,y_test,1,0)
    error=sqared_error(y_test,prediction)
    print(" fold-",i,":",error)
    tmp+=error
  ax=fig.add_subplot(3,3,j)
  ax.set_title("lambda:{} h:{}".format(lam,h))
  ax.scatter(x_train,y_train,s=6)
  ax.plot(x_test,prediction,c='red')
  ax.plot(x_test,truth,c='green')

  print(" *****結果*****;",tmp/n)
  return tmp/n

x, y = generate_sample(sample_size=1000)
H=[0.1,1.0,100.0]
L=[0.1,1.0,100.0]

axes=[]
fig=plt.figure(figsize=(9,6))
fig.subplots_adjust(hspace=0.6, wspace=0.6)
j=1
for h in H:
  for l in L:
    print("h-value:",h)
    print("l-value:",l)
    tmp=cross_validation(x,y,10,h,l,j)
    if ans>tmp:
      ans=tmp
      ans_l=l
      ans_h=h
    j+=1
print('最適解は h:',ans_h,"l:",ans_l,"誤差:",ans)
plt.show()
```

2 宿題 2

目的：線形モデル $f_\theta(x) = \sum_{j=1}^{b} \theta_j \phi_j(x)$ を用いた l_2 正則化回帰に対する一つ抜き交差確認による二乗誤差は次式で解析的に計算できることを示す。

$$\frac{1}{n}\left(\tilde{H}^{-1}H\mathbf{y}\right)^2 \tag{1}$$

証明:

まず、i 番目のデータを抜かない場合の $\hat{\theta}$ は、

$$\hat{\theta} = \left(\Phi^{\mathrm{T}}\Phi + \lambda I\right)^{-1}\Phi\mathbf{y} \tag{2}$$

と表せる。
この $\hat{\theta}$ から i 番目のデータを抜いて学習したパラメータ $\hat{\theta}_i$ を示すには、単位行列 I から i 行目を抜いた行列 A_i を $\Phi, \mathbf{y}$ にそれぞれ左から掛ける。すると、下記のように表すことができる。

$$\hat{\theta}_i = \left(\left(A_i \Phi^{\mathrm{T}}\right) A_i \Phi + \lambda I\right)^{-1} \left(A_i \Phi\right)^{\mathrm{T}} A_i \mathbf{y} \tag{3}$$

ここで、B_i を b_{ii} 成分のみが 1 の行列とすることで $A_i^{\mathrm{T}} A_I = I - B_i$ と表すことができるので、代入すると次のようになる。

$$\begin{aligned}\hat{\theta}_i &= \left(\Phi^{\mathrm{T}} \left(I - B_i\right) \Phi + \lambda I\right)^{-1} \Phi^{\mathrm{T}} \left(I - B_i\right) \mathbf{y} \\ &= \left(\Phi^{\mathrm{T}} \Phi - \Phi^{\mathrm{T}} B_i \Phi + \lambda I\right)^{-1} \left(\Phi \mathbf{y} - \Phi^{\mathrm{T}} B_i \mathbf{y}\right)\end{aligned}$$

ここで、B_i は b_i 成分のみが 1 の行列なので、$\Phi^{\mathrm{T}} B_i \Phi = \phi_i \phi_i^{\mathrm{T}}$ であり、$\Phi^{\mathrm{T}} \Phi + \lambda I = U$ とおくと

$$\hat{\theta}_i = \left(\Phi^{\mathrm{T}} \Phi + \lambda I - \phi_i \phi_i^{\mathrm{T}}\right)^{-1} \left(\Phi \mathbf{y} - \Phi^{\mathrm{T}} B_i \mathbf{y}\right) = \left(U - \phi_i \phi_i^{\mathrm{T}}\right)^{-1} \left(\Phi \mathbf{y} - \phi_i y_i\right)$$

さらに、Sherman- Morrison-Woodbury の公式を用いて、下記のように変形することができる。

$$\begin{aligned}\hat{\theta}_i &= \left(U^{-1} + \tfrac{U^{-1} \phi_i \phi_i U^{-1}}{1 - \phi_i U^{-1} \phi_i}\right) \left(\Phi \mathbf{y} - \phi_i y_i\right) \\ &= \tfrac{U^{-1} - U^{-1} \phi_i U^{-1} \phi + U^{-1} \phi_i \phi_i U^{-1}}{1 - \phi_i U^{-1} \phi_i} = \tfrac{U^{-1}\left(\Phi^{\mathrm{T}} \mathbf{y} - \phi_i y_i\right)}{1 - \phi_i U^{-1} \phi_i}\end{aligned}$$

この $\hat{\theta}_i$ を $y_i - \phi_i \hat{\theta}_i$ に代入すると、下記のように表すことができる。

$$y_i - \phi_i \hat{\theta}_i = y_i - \phi_i \left(\tfrac{U^{-1}\left(\Phi^{\mathrm{T}} \mathbf{y} - \phi_i y_i\right)}{1 - \phi_i U^{-1} \phi_i}\right) = \tfrac{1}{1 - \phi_i U^{-1} \phi_i} \left(y_i - \phi_i U^{-1} \Phi \mathbf{y}\right)$$

以上より、交差確認による二乗誤差は、

$$\tfrac{1}{n} \textstyle\sum_i^n \left(y_i - \phi_i \hat{\theta}_i\right)^2 = \tfrac{1}{n} \sum_i^n \tfrac{1}{1 - \phi_i U^{-1} \phi_i} \left(y_i - \phi_i U^{-1} \Phi \mathbf{y}\right)^2$$

となる。ここで、

$$H = I - \Phi U^{-1} \Phi^{\mathrm{T}} \tag{4}$$

また、$\tilde{H}$ を行列 H と同じ対角成分を持ち、非対角成分は 0 の行列であるとすると、$\frac{1}{1 - \phi_i^{\mathrm{T}} U^{-1} \phi_i}$ は行列 H の i 行目の対角成分の逆数となることから

$$\tfrac{1}{n} \textstyle\sum_i^n \left(y_i - \phi_i \hat{\theta}_i\right)^2 = \tfrac{1}{n} \left(\tilde{H}^{-1} H \mathbf{y}\right)^2$$

と示すことができる。