1.2 ベイズ曲線フィッティング

平成 28 年 9 月 10 日

概 要

PRML の「1.2.5 曲線フィッティング再訪」「1.2.6 ベイズ曲線フィッティング」についての実装と考察

目 次

1　1.2.5 曲線フィッティング

1.1　問題設定

曲線フィッティングを行うと

$$p(t|x,\mathbf{w},\beta)=N(t|y(x,\mathbf{w}),\beta^{-1})$$

となる. これは, 平均がモデル関数値で精度 β の誤差を含むというもの.
N 個の訓練データ集合 $\mathbf{x},\mathbf{t}$ について, $\mathbf{t}$ の尤度関数は

$$p(\mathbf{t}|\mathbf{x},\mathbf{w},\beta)=\prod_{n=1}^{N}N(t_n|y(x_n,\mathbf{w}),\beta^{-1}) \quad (1.61)$$

となる.
また W の事前分布としては, 共役事前分布として

$$p(\mathbf{w}|\alpha)=N(W|\mathbf{0},\alpha^{-1}I) \quad (1.65)$$

を選ぶ.
また, ここでは多項式曲線フィッティングについて考える.

1.2　アルゴリズム

尤度関数 (1.61) の対数, つまり対数尤度関数は

$$\begin{aligned}\ln p(\mathbf{t}|\mathbf{x},\mathbf{w},\beta) &= \ln\prod_{n=1}^{N}\left(\left(\frac{\beta}{2\pi}\right)^{1/2} exp\left\{-\frac{\beta}{2}\left(t_n-y(x_n,\mathbf{w})\right)^2\right\}\right)\\ &= -\frac{\beta}{2}\sum_{n=1}^{N}\{t_n-y(x_n,\mathbf{w})\}^2+\frac{N}{2}\ln\beta-\frac{N}{2}\ln 2\pi \quad (1.62)\end{aligned}$$

ここで尤度関数最大化が二乗和誤差最小化と等価であることが分かる.
また, 事後分布はベイズの定理より尤度関数 $\times$ 事前分布に比例する.

$$p(\mathbf{w}|\mathbf{x},\mathbf{t},\alpha,\beta)\propto p(\mathbf{t}|\mathbf{x},\mathbf{w},\beta)p(\mathbf{w}|\alpha) \quad (1.66)$$

これより, 事後分布の対数をとると対数尤度に定数項と事前分布の対数を加えたものとなる.

$$\ln p(\mathbf{w}|\mathbf{x},\mathbf{t},\alpha,\beta)=-\frac{\beta}{2}\sum_{n=1}^{N}\{t_n-y(x_n,\mathbf{w})\}^2-\alpha\|\mathbf{w}\|^2+const \quad (1.67)$$

2　1.2.6 ベイズ曲線フィッティング

2.1　問題設定

新たな入力 x に対する目標値 t の予測分布を考えたい. ここでは α は固定値として

$$p(t|x,\mathbf{x},\mathbf{t})=\int p(t|x,\mathbf{w})p(\mathbf{w}|\mathbf{x},\mathbf{t})d\mathbf{w} \quad (1.68)$$

となる.

2.2　アルゴリズム

まず, β については, (1.62) を β で微分することにより MAP 解が

$$1/\beta = \frac{1}{N}\sum_{n=1}^{N}\{t_n - y(x_n, \mathbf{w})\}^2 \quad (1.63)$$

となることが分かる.
詳細は 3.3 節で記されるが

$$p(t|x, \mathbf{x}, \mathbf{t}) = N(t|m(x), s^2(x)) \quad (1.69)$$

ここで平均と分散は

$$m(x) = \beta\boldsymbol{\phi}(x)^T S \sum_{n=1}^{N}\boldsymbol{\phi}(x_n)t_n \quad (1.70)$$

$$s^2(x) = \beta^{-1} + \boldsymbol{\phi}(x)^T S \boldsymbol{\phi}(x) \quad (1.71)$$

また

$$S^{-1} = \alpha I + \beta\sum_{n=1}^{N}\boldsymbol{\phi}(x_n)\boldsymbol{\phi}(x_n)^T \quad (1.72)$$

$$\phi_i(x) = x^i \ (i = 0, \ldots, M)$$

である.

ベイズ曲線フィッティング

1. 訓練データから β, S を計算する.
2. 新たな入力から $m(x), s^2(x)$ を計算する.

2.3　コード

予測分布の平均, 分散を求めるプログラム (predictive.py)

```
#モデル選択多項式
M=20
def phi(x):
        V=np.zeros(M+1)
        for m in range(M+1):
                V[m]=np.power(x,m)
        return V

"""尤度関数最大化によりWを求める，bも求める"""
A=np.zeros((M+1,M+1))
W=np.zeros(M+1)
T=np.zeros(M+1)

#二乗和誤差の最小化
for i in range(M+1):
        for j in range(M+1):
                for n in range(N):
                        A[i][j]+=np.power(x[n],i+j)
        for n in range(N):
                T[i]+=np.power(x[n],i)*t[n]

#線形方程式(AW=T)を解くことでパラメータWを求める
```

```
W=np.linalg.solve(A,T)

#求まったパラメータからモデル関数を作り
def model_f(x):
        sum=0.0
        for m in range(M+1):
                sum+=W[m]*x**m
        return sum

#最尤推定によるパラメータb
b_inv=0.0
for n in range(N):
        b_inv+=np.power((model_f(x[n])-t[n]),2)
b=N/b_inv

#固定値のパラメータa
a=50

"""予測分布について"""
#Sの計算
S_inv=a*np.identity(M+1)
sum=np.zeros((M+1,M+1))
for n in range(N):
        sum+=np.outer(phi(x[n]),phi(x[n]))
sum*=b
S_inv+=sum
S=np.linalg.inv(S_inv)

#new_x,new_tについて
new_x=pi
new_t=original_f(new_x)

#平均m,分散s^2の計算
sum=np.zeros(M+1)
for n in range(N):
        sum+=phi(x[n])*t[n]
m=b*np.dot(phi(new_x),np.dot(S,sum))

s2=1.0/b + np.dot(phi(new_x),np.dot(S,phi(new_x)))

#表示
print("(a,b)=(",a,b,")")
print("new(x,t)=(",new_x,new_t,")")
print("(m,s^2)=(",m,s2,")")
```

3 結果

α は固定値として, 訓練データ集合から β, 新しい入力 new_x から m, s^2 を計算する．ここで $\alpha = 50, new\_x = \pi$ とした.
データの精度は $1/(0.3)^2 = 11.1111$ であったが, データから計算された精度パラメータ β の MAP 解は $\beta = 10.8591$ となり正解に近くなっている.
また, 新しい入力 $new\_x = \pi$ に元の関数を代入すると, $new\_t = -1.0$ となるが, 対して予測分布の平均は $m = -1.0898$ となり正解に近くなっている.
そして, 予測分布の分散は $s^2 = 0.0929$ はデータの分散 $(0.3)^2 = 0.09$ と近くなっている.
ただし, これは $\alpha = 50$ とした結果である.

α	β	s^2	new_x	new_t	m
50	10.859	0.0929	π	-1.0	-1.09
1	-	0.0929	-	-	-2.19
5	-	0.0928	-	-	-1.95
10	-	0.0930	-	-	1.42
100	-	0.0929	-	-	-1.74
500	-	0.0928	-	-	-1.32

表 1: a と m, σ^2 の関係

a によらずは s^2 安定して正解と近くなっているが, m はかなりぶれる. これは $\mathbf{w}$ の値が一定にならないことに由来する.

4 まとめ

1. 実行ごとに m が大きく変化し, もともと m が正解に近くならない. これらは, 超パラメータ α の最適化が必要となると考える.