

4.5 ベイズロジスティック回帰

平成 28 年 9 月 11 日

概 要

PRML の「4.5 ベイズロジスティック回帰」についての実装と考察.

目 次

1	問題設定	2
2	4.5.1 ラプラス近似	2
2.1	アルゴリズム	2
2.2	コード	2
2.3	結果	3
2.3.1	二乗和誤差	3
2.3.2	交差エントロピー誤差	3
3	4.5.2 予測分布	4
3.1	アルゴリズム	4
3.2	コード	5
3.3	結果	5
3.3.1	二乗和誤差	5
3.3.2	交差エントロピー誤差	6
4	まとめ	6

1 問題設定

ロジスティック回帰のベイズ的な取扱いについて考える。ここで問題となるのは、事後分布と予測分布が直接積分できない形になってしまうことである。そこで、ベイズロジスティック回帰問題にラプラス近似を適用することを考える。

2 4.5.1 ラプラス近似

2.1 アルゴリズム

事後分布のガウス分布表現を探索するということなので、事前分布にもガウス分布を共役なものとして用いる。

$$p(\mathbf{w}) = N(\mathbf{w}|\mathbf{m}_0, S_0) \quad (4.140)$$

ここで、 \mathbf{m}_0, S_0 は超パラメータである。これより、事後分布は

$$p(\mathbf{w}|\mathbf{t}) \propto p(\mathbf{w})p(\mathbf{t}|\mathbf{w}) \quad (4.141)$$

となる。ここで、尤度関数には、ロジスティック回帰ということので

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^N y_n^{t_n} \{1 - y_n\}^{1-t_n} \quad (4.89)$$

を用いる。ただし、

$$y_n = \sigma(\mathbf{w}^T \phi_n) \quad (4.87)$$

事後分布の対数をとると、

$$\ln p(\mathbf{w}|\mathbf{t}) = -\frac{1}{2}(\mathbf{w} - \mathbf{m}_0)^T S_0^{-1}(\mathbf{w} - \mathbf{m}_0) + \sum_{n=1}^N \{t_n \ln y_n + (1 - t_n) \ln (1 - y_n)\} + \text{const} \quad (4.142)$$

となる。これから MAP 解を見つける (4.3.3 反復重み付け最小二乗 を参照)。これを事後分布の平均とする。そして、共分散は負の対数尤度における 2 階微分行列の逆行列によって以下の式で与えられる。

$$S_N^{-1} = -\nabla \nabla \ln p(\mathbf{w}|\mathbf{t}) = S_0^{-1} + \sum_{n=1}^N y_n(1 - y_n) \phi_n \phi_n^T \quad (4.143)$$

これらより、事後分布は

$$q(\mathbf{w}) = N(\mathbf{w}|\mathbf{w}_{MAP}, S_N) \quad (4.144)$$

となる。

2.2 コード

事後分布のパラメータを求める (posterior2.py)

```
"""パラメータの決定"""
def sig(z):
    try:
        return 1/(1+np.exp(-z))
    except:
        return 0
#MAP解を見つける(誤差関数が交差エントロピー誤差関数のとき)
norm=1
```

```

W_new=np.zeros(3)
R=np.zeros((N,N))
i=0
while i!=10:#norm>10**(-3):
    W_old=W_new
    y=[sig(dot(W_old,x[n,:])) for n in range(N)]
    for n in range(N):
        R[n,n]=y[n]*(1-y[n])
    W_new=W_old-dot(inv(dot(x.T,dot(R,x))),dot(x.T,y-t))
    #z=dot(x,W_old)-dot(inv(R),y-t)
    #W_new=dot(inv(dot(x.T,dot(R,x))),dot(x.T,dot(R,z)))
    norm=np.linalg.norm(W_new-W_old)
    i+=1

#事後分布の平均を求める
m_N=W_new
#事後分布の共分散を求める
S_0=np.zeros((3,3))
alpha=10**(-3)
for m in range(3):
    S_0[m,m]=alpha

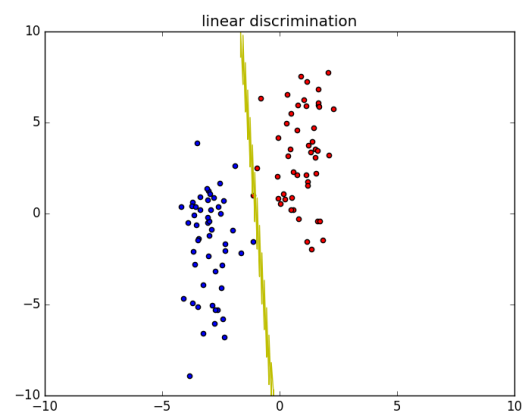
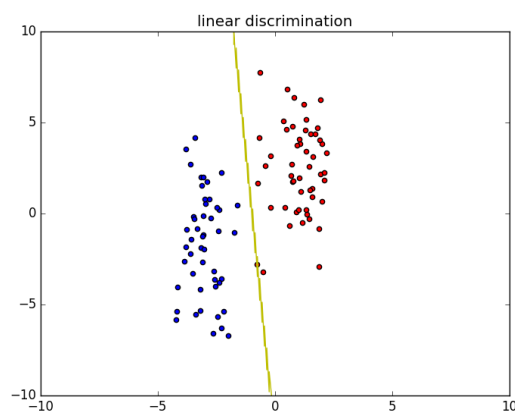
S_N=inv(S_0)
for n in range(N):
    y_n=sig(dot(m_N,x[n,:]))
    S_N+=(y_n*(1-y_n)*outer(x[n,:],x[n,:]))
S_N=inv(S_N)

```

2.3 結果

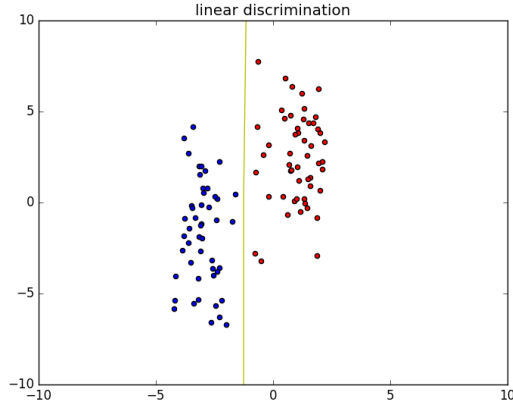
2.3.1 二乗和誤差

左は $t \in \{0, 1\}$ として, $\mathbf{w}^T \phi = 0.5$ の決定境界を描いた. 右は $t \in \{-1, 1\}$ として, $\sigma(\mathbf{w}^T \phi) = 0.5$ の決定境界を描いた.



2.3.2 交差エントロピー誤差

左は $t \in \{0, 1\}$ として, $\sigma(\mathbf{w}^T \phi) = 0.5$ の決定境界を描いた.



3 4.5.2 予測分布

3.1 アルゴリズム

新たな特徴ベクトル $\phi(\mathbf{x})$ が与えられた際のクラス C_1 に対する予測分布は, 事後分布 $p(\mathbf{w}|\mathbf{t})$ に関して周辺化して得られる. 事後分布 $p(\mathbf{w}|\mathbf{t})$ はそれ自体がガウス分布による近似であるので, 予測分布は

$$p(C_1|\phi, \mathbf{t}) = \int p(C_1|\phi, \mathbf{w})p(\mathbf{w}|\mathbf{t})d\mathbf{w} \simeq \int \sigma(\mathbf{w}^T\phi)q(\mathbf{w})d\mathbf{w} \quad (4.145)$$

$a = \mathbf{w}^T\phi$ とすると

$$\sigma(a) = \int \delta(a - \mathbf{w}^T\phi)\sigma(a)da \quad (4.146)$$

となる. そして

$$\int \sigma(\mathbf{w}^T\phi)q(\mathbf{w})d\mathbf{w} = \int \sigma(a)p(a)da \quad (4.147)$$

が得られる. ここで

$$p(a) = \int \delta(a - \mathbf{w}^T\phi)\sigma(\mathbf{w})d\mathbf{w} \quad (4.148)$$

これより, この平均は

$$\mu_a = E[a] = \int p(a)ada = \int q(\mathbf{w})\mathbf{w}^T\phi d\mathbf{w} = \mathbf{w}_{MAP}\phi \quad (4.149)$$

となり, 共分散は

$$\begin{aligned} \sigma_a^2 &= \text{var}[a] = \int p(a)\{a^2 - E[a]^2\}da \\ &= \int q(\mathbf{w})\{(\mathbf{w}^T\phi)^2 - (\mathbf{m}_N^T\phi)^2\}d\mathbf{w} = \phi^T S_N \phi \end{aligned} \quad (4.150)$$

となる. これより,

$$p(C_1|\mathbf{t}) = \int \sigma(a)p(a)da = \int \sigma(a)N(a|\mu_a, \sigma_a^2)da \quad (4.151)$$

となる. これは直接解析できない. そこで, ロジスティックシグモイド関数 $\sigma(a)$ とプロビット関数の逆関数 $\Phi(\lambda a)$ ($\lambda^2 = \pi/8$) との高い類似性を用いて近似を得る.

$$\int \Phi(\lambda a)N(a|\mu, \sigma^2)da = \Phi\left(\frac{\mu}{(\lambda^{-2} + \sigma^2)^{1/2}}\right) \quad (4.152)$$

を用いることによって,

$$\int \sigma(a) N(a|\mu, \sigma^2) da \simeq \sigma(\kappa(\sigma^2)\mu) \quad (4.153)$$

ここで,

$$\kappa(\sigma^2) = (1 + \pi\sigma^2/8)^{-1/2} \quad (4.154)$$

となる. 最終的に, 予測分布として

$$p(C_1|\phi, t) = \sigma(\kappa(\sigma_a^2)\mu_a) \quad (4.155)$$

が得られる. これは $a = \mathbf{w}^T \phi(\mathbf{x})$ という形で \mathbf{x} に依存する.

3.2 コード

予測分布を求める (test2.py)

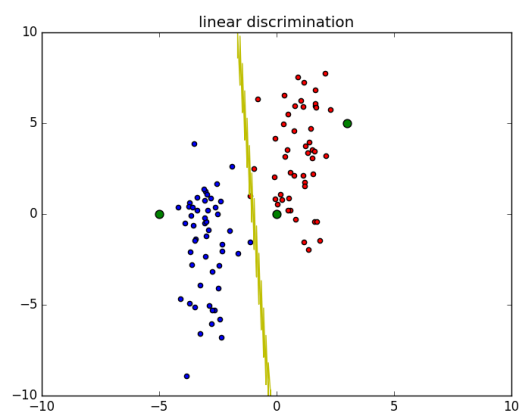
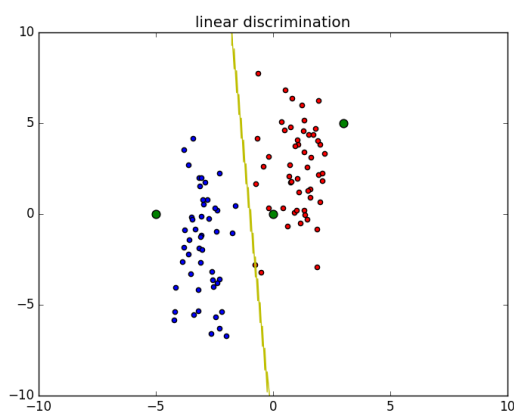
```
"""予測分布を求める"""
new_x=np.array([[ -5,0],[0,0],[3,5]])
for n in range(3):
    phi=np.array([1,new_x[n,0],new_x[n,1]])
    m_a=dot(m_N,phi)
    sig_a=dot(phi,dot(S_N,phi))
    kap_a=1/sqrt(1+pi*sig_a**2/8)
    p=sig(kap_a*m_a)
    print(p)
```

3.3 結果

新しい点 (緑色) がクラス C_1 (赤色) に属す確率を返す.

3.3.1 二乗和誤差

左は $t \in \{0, 1\}$ として, $\mathbf{w}^T \phi = 0.5$ の決定境界を描いた. 右は $t \in \{-1, 1\}$ として, $\sigma(\mathbf{w}^T \phi) = 0.5$ の決定境界を描いた.

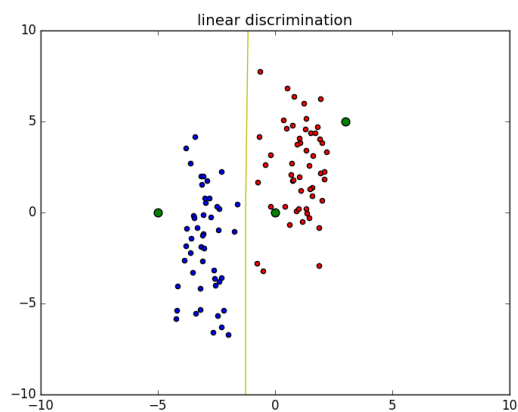


(x,y)	(-5,0)	(0,0)	(3,5)
左	0.156	0.603	0.860
右	0.420	0.6068	0.802

表 1: 近似予想分布

3.3.2 交差エントロピー誤差

左は $t \in \{0, 1\}$ として, $\sigma(W^T \Phi) = 0.5$ の決定境界を描いた.



(x,y)	(-5,0)	(0,0)	(3,5)
左	0.000	1.000	1.000

表 2: 近似予想分布

これからもわかるように, W の最適化を途中で切っているため問題が生じている. ほかの点でも実験したところ step 関数のように傾きが急になっていた.

4 まとめ

大体うまくいった. ただ, 確率の尺度が正しいのかは疑問.