R でベイス線形回帰

宇佐美雅紀

2014/12/01

1 はじめに

この資料は、第８回 機械学習勉強会で勉強した、ベイス線形回帰を R 言語を使って実装する例を説明するものです。参考にしたのは、以下のブログです。

Mimanca qualche giovedi?
R でベイズ線形回帰の予測分布 (2009-07-09 付)
`http://d.hatena.ne.jp/n_shuyo/20090709/predictive`

ソースコードは、丸ごとコピペです。宇佐美のオリジナリティはありません。
コンセプトは、「他人のふんどしで相撲をとる」です。

2 PRML 3.3.2 章 予測分布のおさらい

教科書の 3.3.2 章で解説している予測分布の式をおさらいします。予測分布の式は以下のようになります (教科書 3.58 式)。

$$p(t|\mathbf{x},\mathbf{t},\alpha,\beta) = N(t|\mathbf{m}_N^T\phi(\mathbf{x}),\sigma_N^2(\mathbf{x})) \tag{1}$$

ここで、予測分布の分散 $\sigma_N^2(x)$ は、以下の式で与えられます (教科書 3.59 式)。

$$\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \phi(\mathbf{x})^T\mathbf{S}_N\phi(x) \tag{2}$$

また、$\mathbf{m}_N$、$\mathbf{S}_N^{-1}$ はそれぞれ教科書 3.53 式、3.54 式から以下のように与えられます。

$$\mathbf{m}_N = \beta\,\mathbf{S}_N\boldsymbol{\phi}^T\mathbf{t} \tag{3}$$

$$\mathbf{S}_N^{-1} = \alpha\,\mathbf{I} + \beta\,\boldsymbol{\phi}^T\boldsymbol{\phi} \tag{4}$$

基底関数はガウス基底関数を使用します (教科書 3.4 式)。ここで μ_j は入力空間における基底関数の位置を表し、パラメータ s は空間的な尺度を表す、と教科書には書いてますがイマイチよくわかりません。

$$\phi_j(x) = exp\left\{-\frac{(x-u_j)^2}{2s^2}\right\} \tag{5}$$

念のためガウス分布の定義もおさらいします (教科書 1.47 式)。μ は平均、σ^2 が分散。

$$N(x|\mu,\sigma^2)=\frac{1}{(2\pi\sigma^2)^{1/2}}exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\} \tag{6}$$

3 R のソースコード

```
#####################################################
#
# Usage: R --vanilla --slave < linear_reqgression.r
#       http://d.hatena.ne.jp/n_shuyo/20090709/predictive
#
#####################################################

M <- 9          # number of basis function
alpha <- 2      # hyper parameter α
beta <- 25      # hyper parameter β
Lattice <- 30   # number of graph's lattice
s<-0.1
u_i<-0.5

# training data
xlist <- seq(0, 1, length=25)
tlist <- sin(2*pi*xlist) + rnorm(length(xlist), sd=0.2)
D0 <- data.frame(x=xlist, t=tlist)

predictive <- function(D) {
    # design matrix
    phi <- function(x) sapply(x,function(x){exp(-(x-seq(0,1,length=9))^2/(2*s*s))})
    PHI <- t(phi(D$x))

    # convariance matrix & means
    S_N_inv <- alpha * diag(9) + beta * t(PHI) %*% PHI
    S_N <- solve(S_N_inv)
    m_N <- beta * S_N %*% t(PHI) %*% D$t

    # regression function
    y <- function(x)(t(phi(x)) %*% m_N)
    plot(y, xlim=c(0,1), ylim=c(-1.2, 1.2))
    par(new=T)
    plot(D, xlim=c(0,1), ylim=c(-1.2, 1.2), ylab="")

    # predictive distribution
    var_N <- function(x) {1/beta + (t(phi(x)) %*% S_N %*% phi(x))[1]}
    function(x,t) {
        mapply(function(x,t)dnorm(t,m=(t(m_N) %*% phi(x))[1], s=var_N(x), log=T), x, t)
    }
}
draw_dist <- function(p){
    x <- seq(0, 1, length=Lattice)
    t <- seq(-1.5, 1.5, length=Lattice*2)
    z <- outer(x, t, p)
    persp(x, t, z, theta=0, phi=60, shade=0.4)
}
```

```
p <- predictive(D0[sample(length(D0$x))[1:1],])
draw_dist(p);
p <- predictive(D0[sample(length(D0$x))[1:2],])
draw_dist(p);
p <- predictive(D0[sample(length(D0$x))[1:4],])
draw_dist(p);
p <- predictive(D0)
draw_dist(p);
```

4 解説

4.1 計画行列

ガウス基底関数をおさらいします。

$$\phi_j(x) = exp\left\{-\frac{(x-u_j)^2}{2s^2}\right\} \tag{7}$$

計画行列のおさらいです。

$$\boldsymbol{\phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \cdots & \cdots & \cdots & \cdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix} \tag{8}$$

```
# design matrix
phi <- function(x) sapply(x,function(x){exp(-(x-seq(0,1,length=9))^2/(2*s*s))})
PHI <- t(phi(D$x))
```

計画行列に相当する部分のソースコードです。

phi の後半部分は、ガウス基底関数そのままであることがわかります。ガウス基底関数の μ_j 部分が seq(0,1,length=9) に変更されています。これは、パラメータ μ_j を [0,1] で 9 等分にすることを意味しています。これでガウス基底関数部分はスカラー値 x を与えると、ベクトル値が返る関数になります。sapply は、リストに関数を適用して行列を返す関数なので、phi にリストを与えると行列が返ります。

phi(D$x) で、データフレーム D の x 列を phi 関数に適用して、計画行列を作ります。t 関数は行列を転置してくれる関数です。

4.2 共分散&平均行列

共分散 S_N と平均行列 M_N をおさらいします。

$$\mathbf{m}_N = \beta\ \mathbf{S}_N \boldsymbol{\phi}^T \mathbf{t} \tag{9}$$

$$\mathbf{S}_N^{-1} = \alpha\ \mathbf{I} + \beta\ \boldsymbol{\phi}^T \boldsymbol{\phi} \tag{10}$$

```
    # convariance matrix & means
    S_N_inv <- alpha * diag(9) + beta * t(PHI) %*% PHI
    S_N <- solve(S_N_inv)
    m_N <- beta * S_N %*% t(PHI) %*% D$t
```

共分散 S_N と平均行列 M_N に相当する部分のソースコードです。
diag は単位行列を作る関数です。solve は逆行列を求める関数です。
ハイパーパラメータの α と β は、ソースコードの最初で定数として定義されています。

4.3 予測分布

予測分布の式をおさらいします。

$$p(t|\mathbf{x}, \mathbf{t}, \alpha, \beta) = N(t|\mathbf{m}_N^T \phi(\mathbf{x}), \sigma_N^2(\mathbf{x})) \tag{11}$$

予測分布の分散 $\sigma_N^2(x)$ の式をおさらいします。

$$\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \phi(\mathbf{x})^T \mathbf{S}_N \phi(x) \tag{12}$$

```
# predictive distribution
var_N <- function(x) {1/beta + (t(phi(x)) %*% S_N %*% phi(x))[1]}
function(x,t) {
    mapply(function(x,t)dnorm(t,m=(t(m_N) %*% phi(x))[1], s=var_N(x), log=T), x, t)
}
```

予測分布の式に相当する部分のソースコードです。
分散の関数は、式そのままです。
dnorm 関数は、正規分布の確率密度です。