第5章　径向基函数(RBF)网络

5.3 RBF网络的正交最小平方学习算法

5.4 广义RBF网络及其学习算法

5.3 RBF网络的正交最小平方学习算法

(i). 回归模型

考虑RBF网络实现从 R^n 到 R 的映射 $f(x)$：

$$f(x) = \lambda_0 + \sum_{j=1}^{M} \lambda_j \phi(\| x - c_j \|).$$

这里假定核函数 $\phi(\bullet)$ 与中心 c_j 是给定的，所求的参数为 λ_j。注意，由于一般函数可能存在常数项，因此我们在RBF组合中增加了一个常数项。为了与正交最小平方(OLS)法联系起来，我们将RBF网络看成下面的回归模型：

$$y(t) = \sum_{j=1}^{M} p_j(t)\theta_j + \varepsilon(t),$$

其中 $y(t)$ 为期望输出，$p_j(t)$ 称为回归算子，是数据变量 $x(t)$ 的函数：$p_j(t) = p_j(x(t)) = \phi(\| x(t) - c_j \|^2)$

5.3 RBF网络的正交最小平方学习算法

另外，$\varepsilon(t)$ 为误差变量，并假定它与回归算子$p_j(t)$是不相关（或独立）的。现在我们假定有一组中心 $c_1, c_2, \cdots, c_M$ （注意这里存在着特别情况$\{c_1, c_2, \cdots, c_M\} \subset \{x_1, x_2, \cdots, x_N\}$），则形成了$M$ 个备选的回归算子 $p_1(t), p_2(t), \cdots, p_M(t)$ ，如何从其中选出最有用的来，剔除掉多余的，这就是我们下面要讨论的问题。

将前面的回归方程写成向量矩阵形式，则有

$$y = P\theta + E,$$

其中 $y = [y(1), y(2), \cdots, y(N)]^T$ 。

5.3 RBF网络的正交最小平方学习算法

$$P=[p_1,p_2,\cdots,p_M]=(p_{ij})_{N\times M}, \qquad p_{ij}=\phi(\|x_i-c_j\|)$$

$$\theta=[\theta_1,\theta_2,\cdots,\theta_M]^T, \qquad E=[\varepsilon(1),\varepsilon(2),\cdots,\varepsilon(N)]^T$$

注意这里取 $x(i)=x_i$ 。

(ii). QR分解与回归方程的解

当 M 较大时，回归算子 $p_j(t)=\phi(\|x(t)-c_j\|)$ 一般具有较强的相关性。然而，它们对输入数据所形成了矢量 p_j 是不相关的，即是线性独立的。我们现在通过QR分解求出这些矢量(P 的列向量)所展线性空间的一组正交基，并依此分析回归算子的重要性。为此，我们将回归矩阵 P 进行QR分解： $P=WA$

5.3 RBF网络的正交最小平方学习算法

其中 $W=(w_{ij})_{N\times M}=[w_1,w_2,\cdots,w_M]$ 是各列正交的矩阵，

即满足：

$$W^TW=H,\quad H=diag[h_1,h_2,\cdots,h_M],\quad h_i>0$$

而 $A=(a_{ij})_{M\times M}$ 是如下的一个上三角矩阵：

$$A=\begin{bmatrix} 1 & a_{12} & a_{13} & \cdots & a_{1,M-1} & a_{1M} \\ 0 & 1 & a_{23} & \cdots & a_{2,M-1} & a_{2M} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & a_{M-1,M} \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

5.3 RBF网络的正交最小平方学习算法

根据QR分解，我们有：

$$L(p_1, p_2, \cdots, p_M) = L(w_1, w_2, \cdots, w_M)$$

从而可将回归方程改写为：

$$y = Wg + E$$

根据正交最小均方误差$\| E \|^2$得出：

$$\hat{g} = H^{-1} W^T y$$

或 $$\hat{g}_i = w_i^T y / (w_i^T w_i), \qquad 1 \le i \le M.$$

进一步，我们得到 θ 应满足的方程为：

$$A\theta = \hat{g}.$$

5.3 RBF网络的正交最小平方学习算法

由于 A 是上三角矩阵，很容易从上式求出 θ。

(iii). QR分解的实现

对于 P 的QR分解，我们可以采用传统的Gram-Schmit方法：

$$w_1 = p_1, \qquad k = 2,3,\cdots,M$$

$$w_k = p_k - \sum_{i=1}^{k-1} a_{ik} w_i$$

$$a_{ik} = w_i^T p_k / (w_i^T w_i), \qquad 1 \le i \le k$$

另外，也可以采用Householder变换得到对 P 的一个类似的QR（正交-三角）分解。可以参阅有关矩阵的书籍。

5.3 RBF网络的正交最小平方学习算法

(iv). 重要回归算子的选择

根据 $y = W\hat{g} + E$， 我们定义 y 的能量为

$$
\begin{aligned}
y^T y &= (W\hat{g} + E)^T (W\hat{g} + E) \\
&= \hat{g}^T W^T W\hat{g} + \hat{g}^T W^T E + E^T W\hat{g} + E^T E \\
&= \hat{g}^T W^T W\hat{g} + E^T E \quad (\text{注意在}LMS\text{下}W^T E = 0) \\
&= \sum_{i=1}^{M} \hat{g}_i^2 w_i^T w_i + E^T E
\end{aligned}
$$

这样便可得到各个正交“回归”矢量 w_i 在回归方程中所占的“比重”，我们定义其为：

$$
r_i = \hat{g}_i^2 w_i^T w_i / y^T y, \qquad i = 1, 2, \cdots, M.
$$

5.3 RBF网络的正交最小平方学习算法

根据这些“比重” r_i ，我们下面选择重要回归算子的方法：

Step1：对于 $1 \le i \le M$ ，定义并计算

$$w_1^{(i)} = p_i$$

$$g_1^{(i)} = (w_1^{(i)})^T y / ((w_1^{(i)})^T w_1^{(i)})$$

$$r_1^{(i)} = (g_1^{(i)})^2 (w_1^{(i)})^T w_1^{(i)} / y^T y$$

寻找 $r_1^{(i_1)} = \max\{r_1^{(i)}, i = 1,2,\cdots,M\}$

并令 $w_1 = w_1^{(i_1)} = p_{i_1}$

Step2：$k \ge 2$ ，对于 $1 \le i \le M, i \ne i_1, i_2, \cdots, i_{k-1}$ ，计算

5.3 RBF网络的正交最小平方学习算法

$$a_{jk}^{(i)} = w_j^T p_i / (w_j^T w_j), \qquad 1 \le j \le k-1$$

$$w_k^{(i)} = p_i - \sum_{j=1}^{k-1} a_{jk}^{(i)} w_j \quad (\Rightarrow (w_k^{(i)}) w_j = 0)$$

$$g_k^{(i)} = (w_k^{(i)})^T y / ((w_k^{(i)})^T w_k^{(i)})$$

$$r_k^{(i)} = (g_k^{(i)})^2 (w_k^{(i)})^T w_k^{(i)} / y^T y$$

选择 $r_k^{(i_k)} = \max\{r_k^{(i)}, i = 1,2,\cdots,M, i \ne i_1,\cdots,i_{k-1}\}$

并令 $w_k = w_k^{(i_k)} = p_{i_k} - \sum_{j=1}^{k-1} a_{kj}^{i_k} w_j$

直到满足条件

$$1 - \sum_{j=1}^{k} r_j^{(i_j)} < \rho \qquad (\rho > 0)$$

为止，其中ρ 为容差度。

5.3 RBF网络的正交最小平方学习算法

通过上述算法，我们便选择出了满足容差性的最重要的一些 w_i，进一步便可得到所涉及的 p_i，这样便找到了重要的回归算子，并依此可以简化RBF网络。

最后我们需要讨论以下容差度的选择 ρ。显然，它是平衡精度与网络复杂度的一个重要参数。实际中可以根据问题需要的精度进行分析和估计。

5.4 广义RBF网络及其学习算法

- 高斯径向基函数的推广

（1）简单中心径向基函数

$$R_j(x) = e^{-\|x-c_j\|^2/\sigma^2}$$

（2）带宽度的径向基函数

$$R_j(x) = e^{-\|x-c_j\|^2/\sigma_j^2}$$

5.4 广义RBF网络及其学习算法

(3) 带协方差矩阵的径向基函数

$$R_j(x) = e^{-(x-c_j)^T \Sigma_j^{-1} (x-c_j)}$$

(4) 高斯密度函数型的径向基函数

$$R_j(x) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_j|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1} (x-\mu_j)\}$$

5.4 广义RBF网络及其学习算法

- 广义RBF网络的输出（单输出单元）

$$y = f(x) = \sum_{j=1}^{m} \lambda_j R_j(x),$$

其中

$$R_j(x) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_j|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j)\}.$$

- 与高斯混合模型的区别

$$P(x|\Theta_k) = \sum_{i=1}^{k} \alpha_i q(x|m_i,\Sigma_i),$$

$$q(x|m_i,\Sigma_i) = R_i(x),\ \alpha_i \geq 0, \sum_{i=1}^{k} \alpha_i = 1.$$

$$\Theta_k = \{\alpha_i, m_i, \Sigma_i\}_{i=1}^{k}$$

5.4 广义RBF网络及其学习算法

广义RBF网络的隐单元个数

针对不同的问题，如何选择RBF网络的隐单元（即径向基函数）的个数是广义RBF网络设计的核心问题。隐单元个数不能太小，会影响网络的学习能力，但也不能太大，会影响网络的推广能力。从隐单元的接受域上考虑，若输入样本空间是由 k 个高斯分布组成的混合体，那么选取 k 隐单元最为合理。那么隐单元个数的选取就成为高斯混合模型对于输入样本集的模型选择问题。

5.4 广义RBF网络及其学习算法

传统的模型选择方法

对于一个可能的 k ，根据输入样本集：

$$\mathrm{X}=\{x_1,x_2,\cdots,x_N\}\leftarrow \mathrm{X}\times Y=\{(x_i,\hat{y}_i)\}_{i=1}^{N}$$

建立高斯混合模型 $P(x|\Theta_k)$ ，可通过EM算法得到参数估计（最大似然估计）：

$$E-step:\quad p(j|x_t)=\frac{\alpha_j q(x_t|m_j,\Sigma_j)}{\sum_{i=1}^{k}\alpha_i q(x_t|m_i,\Sigma_i)}.$$

5.4 广义RBF网络及其学习算法

$$M-step:$$

$$\alpha_j^+ = \frac{1}{N}\sum_{t=1}^{N} p(j \mid x_t);$$

$$m_j^+ = \frac{1}{\sum_{t=1}^{N} p(j \mid x_t)} \sum_{t=1}^{N} p(j \mid x_t) x_t;$$

$$\Sigma_j^+ = \frac{1}{\sum_{t=1}^{N} p(j \mid x_t)} \sum_{t=1}^{N} p(j \mid x_t)(x_t - m_j^+)(x_t - m_j^+).$$

5.4 广义RBF网络及其学习算法

- 给定 k 的可行范围 $[k_{\min}, k_{\max}]$，对于每个 k 施行EM算法得到参数估计值。
- 根据一种选择准则函数的最小值来选择最优的 k 。例如下面两个常用的准则（函数）：

(1) AIC(Akaike信息准则)

$$AIC = -2L + 2N_p$$

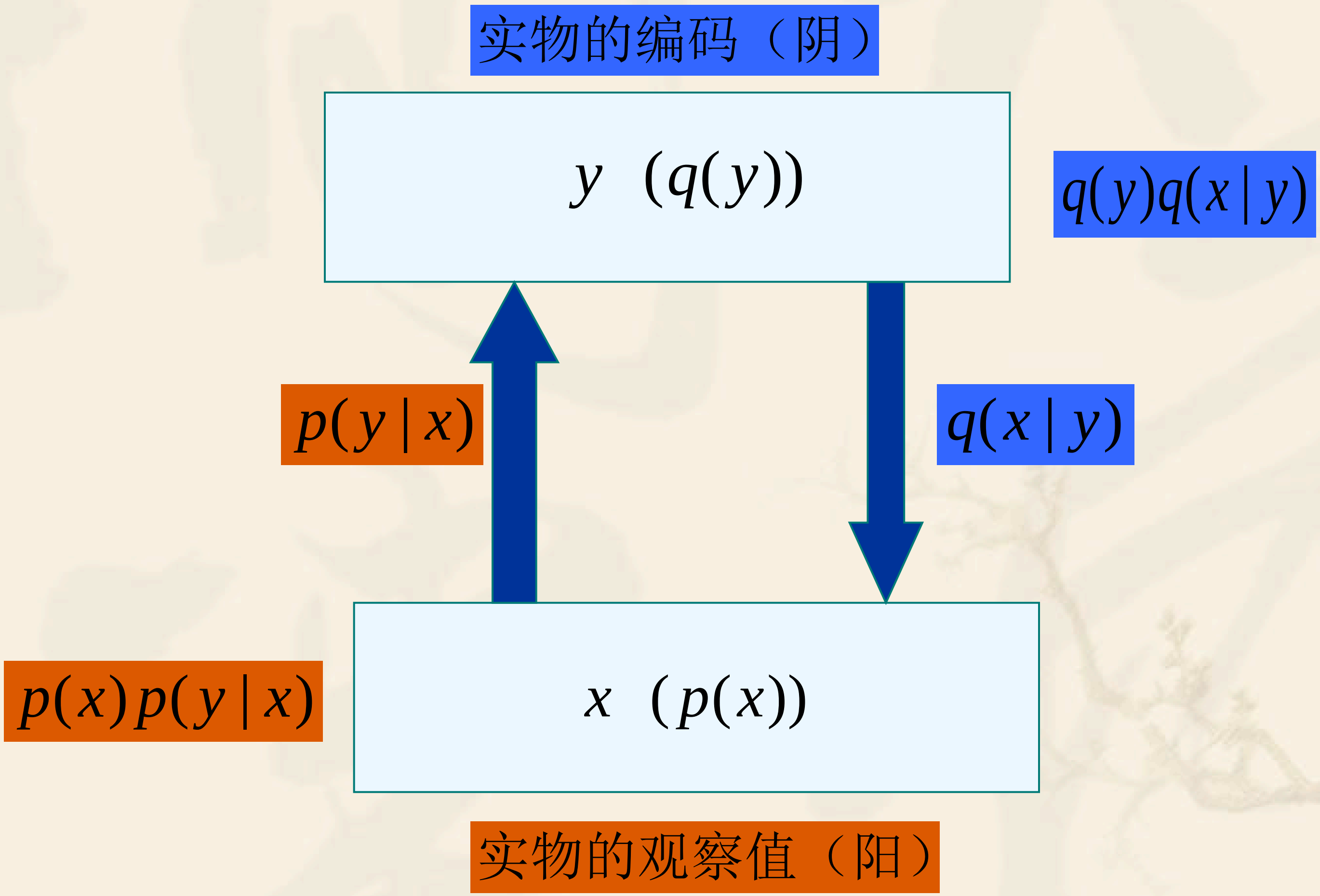
（N_p 为被估计的参数个数）

(2) BIC(贝叶斯推理准则)

$$BIC = -2L + 2N_p \log N$$

5.4 广义RBF网络及其学习算法

- 自动模型选择方法：BYY学习系统 (Xu, 1995, 2001, 2002)



5.4 广义RBF网络及其学习算法

和谐学习原则

$p(x)q(y|x),\ q(y)q(x|y)$ 尽可能的相等，从数学表示为下面的和谐泛函达到最大值：

$$H(p\|q)=\int p(y|x)p(x)\ln[q(x|y)q(y)]dxdy$$

该和谐泛函来自于阴机和阳机的KL散度：

$$K(p\|q)=\int p(y|x)p(x)\ln[\frac{p(y|x)p(x)}{q(x|y)q(y)}]dxdy$$
$$=-H(p(x,y))-H(p\|q).$$

5.4 广义RBF网络及其学习算法

与有限混合模型相对应的双向结构的BYY学习系统：

$$x \in R^n, \qquad y \in \{1,2,\cdots,k\} \subset R$$

$$q(y=j) = q(j) = \alpha_j \quad \text{(混合比例系数)}$$

$$q(x \mid y=j) = q(x \mid \theta_j) \text{（分量）}$$

$$p(y=j \mid x) = \frac{\alpha_j q(x \mid \theta_j)}{q(x \mid \Theta_k)}, \quad q(x \mid \Theta_k) = \sum_{j=1}^{k} \alpha_j q(x \mid \theta_j)$$

$$D_x = \{x_t\}_{t=1}^{N} \sim p(x) = q(x \mid \Theta_{k^*}^*) = \sum_{j=1}^{k^*} \alpha_j^* q(x \mid \theta_j^*).$$

5.4 广义RBF网络及其学习算法

- 有限混合模型上的和谐泛函与函数

$$H(p\|q)=E_{p(x)}\left[\sum_{j=1}^{k}\frac{\alpha_j q(X|\theta_j)}{\sum_{i=1}^{k}\alpha_i q(X|\theta_i)}\ln[\alpha_j q(X|\theta_j)]\right];$$

在样本$D=\{x_t\}_{t=1}^{N}$给定下的估计值———和谐函数：

$$J(\Theta_k)=\frac{1}{N}\sum_{t=1}^{N}\sum_{j=1}^{k}\frac{\alpha_j q(x_t|\theta_j)}{\sum_{i=1}^{k}\alpha_i q(x_t|\theta_i)}\ln[\alpha_j q(x_t|\theta_j)].$$

- 高斯混合模型的和谐函数只需要令

$$q(x|\theta_j)=q(x|m_j,\Sigma_j)=\frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_j|^{\frac{1}{2}}}e^{-\frac{1}{2}(x-m_j)^T\Sigma_j^{-1}(x-m_j)}$$

5.4 广义RBF网络及其学习算法

- 和谐函数 $J(\Theta_k)$ 可以作为一种有效的模型选择准则进行有限混合模型的建模，因为它在正确的模型尺度上（即 $k=k^*$ ）达到最大。

- 根据BYY和谐学习原则， 若 $k>k^*$ ， 则可以通过梯度或其它学习算法得到和谐函数的极大值点， 实现自动模型选择功能， 即将多余高斯分量的混合比例系数逼为零， 其高斯余分量与数据中的真实分量相匹配。其它情况下也可由于自适应模型选择（增长型贪婪算法和动态模型选择算法）。

5.4 广义RBF网络及其学习算法

- Batch-way 梯度学习算法(Ma, Wang, and Xu, 2004)
- 共轭和自然梯队学习算法(Ma et al., 2005)
- 快速不动点学习算法(Ma and He, 2008)
- Adaptive梯度学习算法(Ma and Wang, 2006)
- 模拟退火型学习算法(Ma and Liu, 2007)：该算法是基于后向结构的BYY 学习系统，其形式是一种确定型EM模拟退火算法，但具有自动模型选择功能

5.4 广义RBF网络及其学习算法

广义RBF网络的参数学习：LMS算法

根据前面所选择的隐单元数，网络的输出则为：

$$y = f(x) = \sum_{j=1}^{m} w_j R_j(x \mid \mu_j, \Sigma_j).$$

这样可将从前面算法得到的均值和协方差矩阵做为初始值统一执行LMS算法，来最小化整体均方误差：

$$E = \frac{1}{2}\sum_{t=1}^{N}(\hat{y}_t - y_t)^2 = \frac{1}{2}\sum_{t=1}^{N}(\hat{y}_t - \sum_{j=1}^{k}\omega_j R_j(x_t))^2$$

5.4 广义RBF网络及其学习算法

❖ 参数的迭代公式

$$\Delta\omega_j = \eta\sum_{t=1}^{N}(\hat{y}_t - y_t)R_j(x_t)$$

$$\Delta\mu_j = \eta\sum_{t=1}^{N}(\hat{y}_t - y_t)\omega_j R_j(x_t)\Sigma_j^{-1}(x_t - \mu_j)$$

$$\Delta\mathrm{vec}[B_j] = \frac{\eta}{2}\sum_{t=1}^{N}(\hat{y}_t - y_t)\omega_j R_j(x_t)\frac{\partial(B_j B_j^T)}{\partial B_j}\mathrm{vec}[\Sigma_j^{-1}(x_t - \mu_j)(x_t - \mu_j)^T\Sigma_j^{-1} - \Sigma_j^{-1}]$$

其中 $\Sigma_j = B_j B_j^T$，而其它运算可参加文献(Ma and Wang, 2006)。

参考文献

- Jinwen Ma and Jianfeng Liu, The BYY annealing learning algorithm for Gaussian mixture with automated model selection, **Pattern Recognition**, vol.40, pp:2029-2037, 2007.
- Jinwen Ma and Le Wang, BYY harmony learning on finite mixture: adaptive gradient implementation and a floating RPCL mechanism, **Neural Processing Letters**, vol.24, no.1, pp: 19-40, 2006.
- Jinwen Ma, Taijun Wang, and Lei Xu, A gradient BYY harmony learning rule on Gaussian mixture with automated model selection, **Neurocomputing**, vol.56, pp: 481-487, 2004.

参考文献

- Kai Huang, Le Wang, and Jinwen Ma, Efficient training of RBF networks via the BYY automated model selection learning algorithms, , **Lecture Notes in Computer Science**, vol.4491, pp: 1183-1192，2007.
- Lei Li and Jinwen Ma, A BYY scale-incremental EM algorithm for Gaussian mixture learning, **Applied Mathematics and Computation**, vol.205, pp: 832-840, 2008.
- Jinwen Ma, Jianfeng Liu and Zhijie Ren, Parameter estimation of Poisson mixture with automated model selection through BYY harmony learning, **Pattern Recognition**, vol. 42(2009), pp: 2659-2670.