

# 马氏链蒙特卡洛算法及其应用

yfreedomliTHU THU EE

**摘要:** 本文首先利用 Metropolis-Hastings 算法对二维高斯分布的相关系数进行估计,估计误差在 1%以内,并且就实验结果分析了分布选取对估计结果的影响。接着,本文基于 RJMCMC 原理设计了一个简单的实验。然后,本文实现了基于 RJMCMC 和模拟退火算法的 RBF 模型选择算法,并基于 AIC,BIC,MDL 准则就计算精度,运行效率和数值性能进行了分析对比,同时用训练数据对模型进行训练及就待测数据得到测试结果。最后,将 MCMC 和 RBF 模型结合,实现了基于 MCMC 的 RBF 模型选择算法,同样就 AIC,BIC,MDL 准则进行的对比分析,并训练得到测试结果。对比表明, RJSA 在时间效率和精确度上均优于基于 MCMC 的 RBF,更适宜用于 RBF 学习。

**关键词** MCMC, M-H 算法, RBF 模型, RJMCMC, 模拟退火算法, AIC 准则, BIC 准则, MDL 准则

## Markov Chain Monte Carlo Algorithm and Its Application

Li Yinfeng

(Electronic Engineering, Tsinghua University)

**Abstract** In this paper, the correlation coefficient of two-dimensional Gaussian distribution is estimated by using Metropolis-Hastings algorithm with the estimation error within 1%. And the influence of the distribution selection on the estimation results is analyzed based on the experimental results. Then, we design an experiment based on RJMCMC principle. Next, this paper implements the RBF model selection algorithm based on RJMCMC and simulated annealing algorithm and compares the calculation precision, the operation efficiency and the numerical performance with different criteria(AIC, BIC and MDL). At the same time, it uses the training data to train the model and get the test results with test data. Finally, combining the MCMC and RBF model, the MCMC-based RBF model selection algorithm is implemented, and the comparison and analysis of the AIC, BIC and MDL criteria are also carried out, and the training results are obtained. The comparison shows that RJSA is superior to MCMC-based RBF in terms of time efficiency and accuracy, and is more suitable for RBF learning.

**Key words** MCMC, M-H algorithm, RBF model, RJMCMC, simulated annealing algorithm, AIC criterion, BIC criterion, MDL criterion

## 1 引言(Introduction)

### 1.1 MCMC

马氏链蒙特卡洛 (Markov Chain Monte Carlo, MCMC) 方法, 是马氏链理论的一个重要应用。从1950 年萌芽, 马氏链蒙特卡洛方法在实践中不断发展, 逐渐成长为一个颇具分量的理论分支, 广泛应用于各种学科领域 (如信息科学、物理、化学、生物学、金融、材料等) 的科学计算, 展示出越来越强大的威力。MCMC方法由Metropolis提出, 之后Hastings对MCMC加以推广, 形成了MH算法。但是在实验中发现, 普通的MH算法对迭代次数和提议分布都有一定的依赖。

本文使用MH算法对二维高斯分布进行采样, 由采样结果对二维高斯分布的相关系数进行估计。就问题本文提出了二维均匀分布和与原分布类似的二维正态分布, 并针对不同的分布对结果的影响进行了分析。

### 1.2基于MCMC的径向基函数 (RBF) 网络模型选择方法

RBF 网络是一种经典的神经网络结构。由于 RBF 函数能够逼近任意的非线性函数, 其通常被应用于处理系统内的难以解析的规律性, 具有良好的泛化能力, 并有很快的学习收敛速度, 已成功应用于非线性函数逼近、时间序列分析、数据分类、模式识别、信息处理、图像处理、系统建模、控制和故障诊断等。经典的RBF模型选择方法有很多种, 如AIC方法[5]、MDL方法[6]、BIC 方法[7]以MAP方法[8]。其中基于RJ (Reversible Jump) MCMC[9]及模拟退火算法 (Simulated Annealing algorithm, SA) 的RBF模型选择方法, 可以对RBF网络进行更好的学习。

## 2 二维高斯分布相关系数的估计(Correlation coefficient of 2D Gauss distribution)

### 2.1 概述

已知如下二维高斯分布:

$$\mathcal{N}\left\{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \middle| \begin{pmatrix} 5 \\ 10 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ -1 & 4 \end{pmatrix}\right\}$$

通过理论计算可以得到相关系数 $\rho = 0.5$ .

## 2.2 Metropolis–Hastings算法

---

### Algorithm 1 Metropolis-Hastings Algorithm

---

Initialize S

**for**  $i = 1$  **to** N **do**

    Generate Y from proposal distribution  $Q(X)$

    Evaluate the acceptance probability

$$r = \min\left\{1, \frac{p(Y)Q(Y,X)}{p(X)Q(X,Y)}\right\}$$

    Generate  $u$  from Uniform distribution  $[0,1]$

    if  $u < r$  then  $X^{(i)} = Y$

**end for**

---

MH算法通过生成的均匀分布的数来决定是否接受Y中的数，对于给定的仿真次数，程序需要循环执行N次，故时间复杂度为 $O(N)$ 。

## 2.3 实验结果及分析

(a) 提议分布为均匀分布：

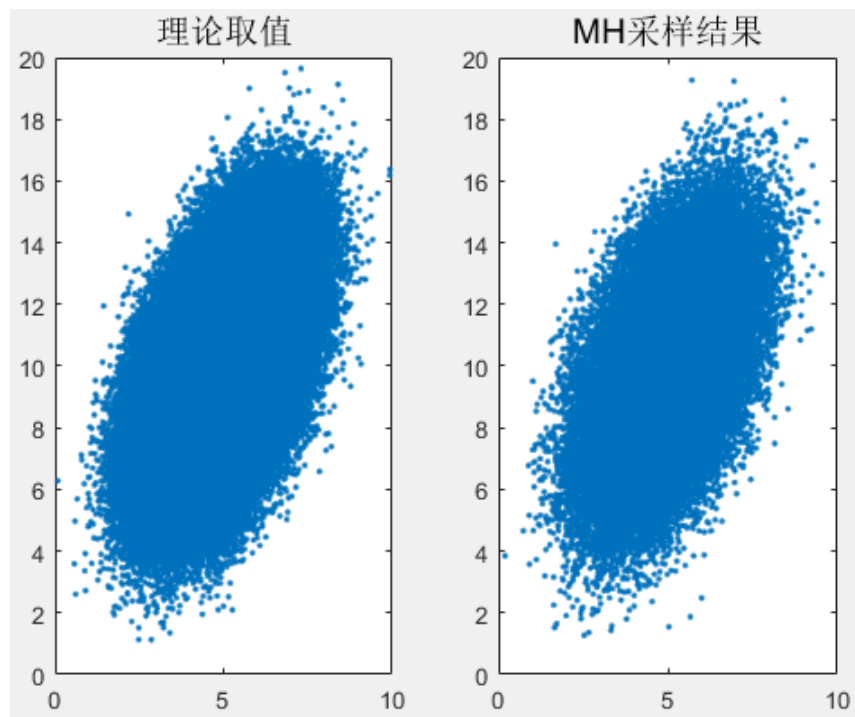


图1

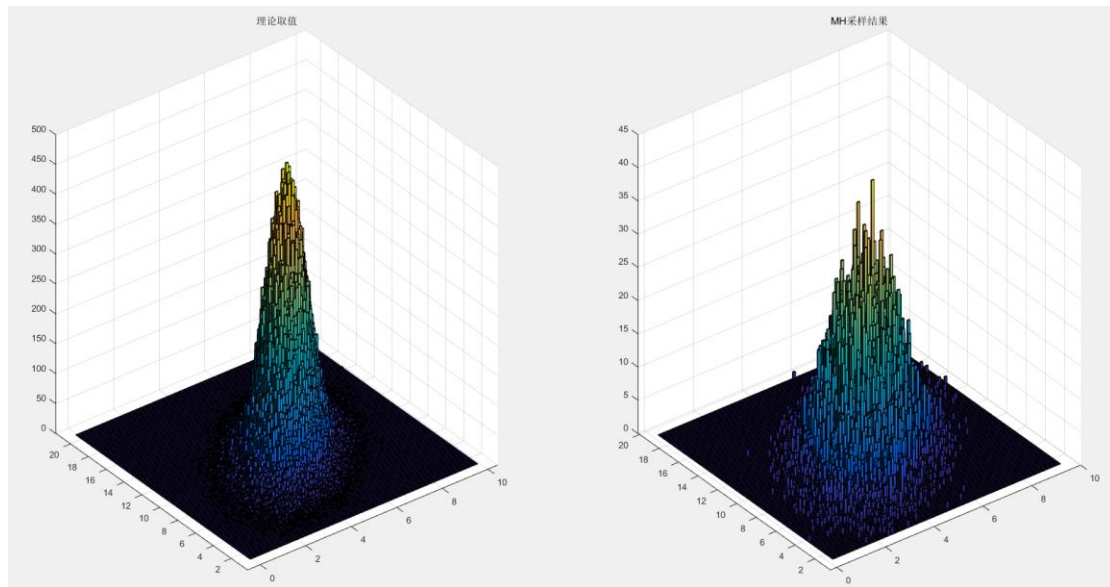


图2

图1，图2分别为在均匀分布下，仿真500000次后采样结果与理论分布的对比图。此时相关系数的估计值为0.5003，和理论值的误差在1%以内，能够较好地拟合理论分布。在此基础上，将提议分布换为二维正太分布，进行进一步的对比。

(b) 提议分布为二维高斯分布：

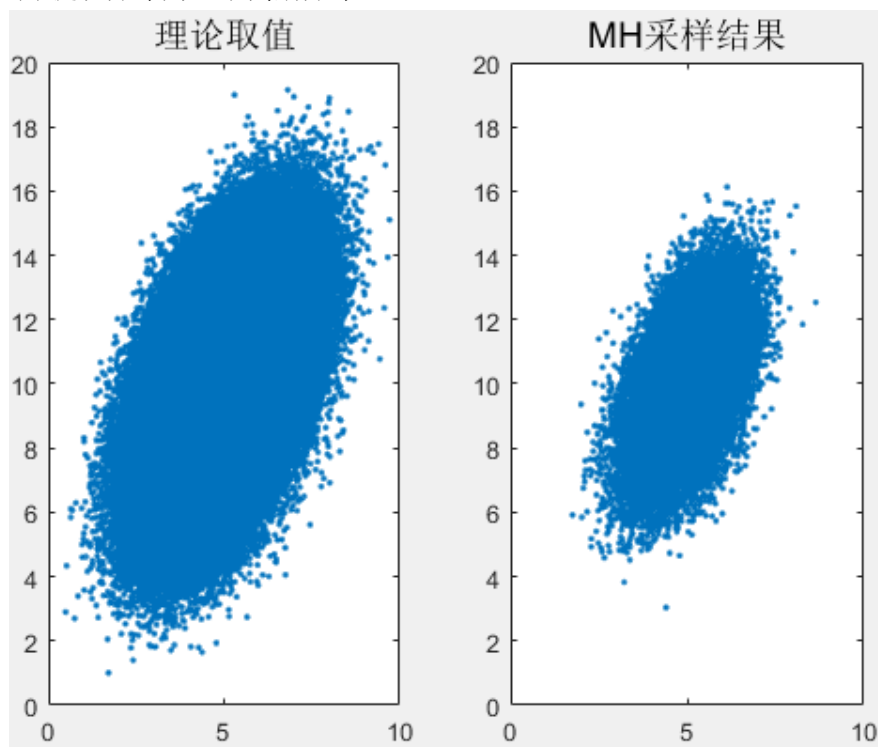


图3

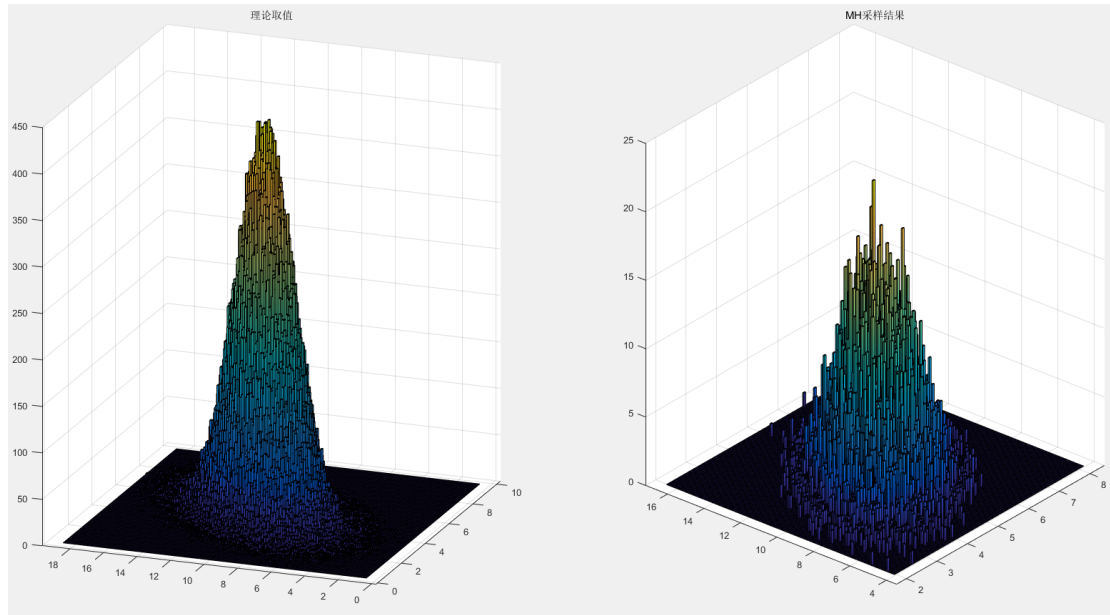


图4

图3，图4是二维高斯分布（和目标分布十分类似），仿真500000次后采样结果和理论分布的对比图。此时相关系数的估计值为0.4980，和理论误差值的误差仍在1%以内。但是需要注意的是，仿真时发现不同高斯分布的选取对采样结果以及相关值估计影响较大。当选取的二维高斯分布逼近目标分布是，能得到较好的估计，但是如果随便选取，由于受限于采样区域，得到的相关系数估计值精度将下降。

### 3 基于MCMC的径向基函数网络模型选择方法(Model Selection Method of Radial Basis Function Networks Based on MCMC)

#### 3.1 概述

RBF 网络是一种经典的神经网络结构，RBF 模型可以简单表示为以下形式 [4]：

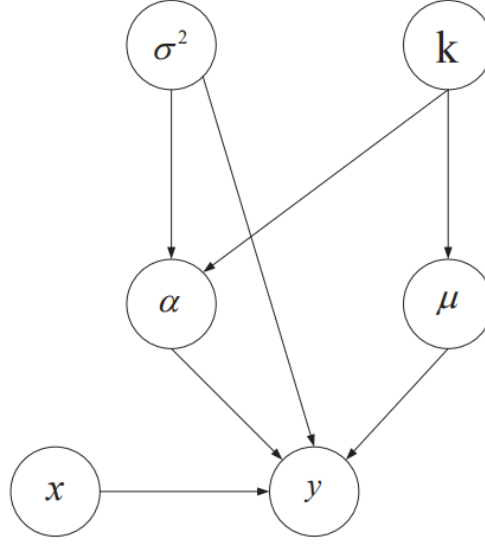
$$\mathbf{y}_{N \times c} = \mathbf{D}(\boldsymbol{\mu}_{k \times d}, \mathbf{x}_{N \times d}) \boldsymbol{\alpha}_{(1+d+k) \times c} + \mathbf{n}_{N \times c}$$

$\mathbf{y}$ 为观测样本， $N$ 为样本点数， $c$ 为每个样本点的维度； $\mathbf{x}$ 为 $N$ 个观测样本点对应的变量， $d$ 为每个变量的维度；矩阵 $\mathbf{D}$ 有如下形式：

$$\begin{pmatrix} 1 & \mathbf{x}_{1,1} & \dots & \mathbf{x}_{1,d} & \phi(\mathbf{x}_1, \boldsymbol{\mu}_1) & \dots & \phi(\mathbf{x}_1, \boldsymbol{\mu}_k) \\ 1 & \mathbf{x}_{2,1} & \dots & \mathbf{x}_{2,d} & \phi(\mathbf{x}_2, \boldsymbol{\mu}_1) & \dots & \phi(\mathbf{x}_2, \boldsymbol{\mu}_k) \\ \vdots & \vdots & & \vdots & & & \vdots \\ 1 & \mathbf{x}_{N,1} & \dots & \mathbf{x}_{N,d} & \phi(\mathbf{x}_N, \boldsymbol{\mu}_1) & \dots & \phi(\mathbf{x}_N, \boldsymbol{\mu}_k) \end{pmatrix}$$

$\phi(\mathbf{x}, \boldsymbol{\mu})$ 表示RBF函数， $k$ 为RBF函数的个数， $\boldsymbol{\mu}$ 为函数的中心， $\boldsymbol{\alpha}$ 为加权系数， $\mathbf{n}$ 是高斯白噪声， $\mathbf{n}_i \sim N(0, \sigma_i^2), i = 1, 2, \dots, c$ 。RBF模型的参数集合可用 $\mathcal{M}_s = (\boldsymbol{\mu}, \boldsymbol{\alpha}, \sigma^2, k)$ 表示。

RBF模型的基本结构如下：



### 3.2 基于RJMCMC原理的实验设计

#### 3.2.1 RJMCMC理论

MCMC算法有一定的局限性，例如无法实现不同维度之间的跳转。RJMCMC基于原有MCMC算法，在当前状态下，下一状态能够生成或者跳转到与当前维度不同的状态，本文认为这一特性为神经网络的实现提供了重要的思路和方法。

和MCMC类似，RJMCMC的提议也不是被立即接收的，文献[9]提出了RJMCMC的过渡内核：

$$P(x, B) = \sum_m \int_B q_m(x, dx') \alpha_m(x, x') + s(x) I(x \in B) \quad (1)$$

我们设

$$\xi(A \times B) = \xi(B \times A) = \lambda \{(\theta^{(1)}, u^{(1)}): \theta^{(1)} \in A, \theta^{(2)}(\theta^{(1)}, u^{(1)}) \in B\},$$

通过运算可以得到不同维度空间的接收判据如下：

$$\min \left\{ 1, \frac{p(2, \theta^{(2)}|y) j(2, \theta^{(2)}) q_2(u^{(2)}) \left| \frac{\partial(\theta^{(2)}, u^{(2)})}{\partial(\theta^{(1)}, u^{(1)})} \right|}{p(1, \theta^{(1)}|y) j(1, \theta^{(1)}) q_1(u^{(1)})} \right\} \quad (2)$$

#### 3.2.2 实验设计

RJMCMC算法往往和神经网络相结合。作为第二问RJSA算法的探索，本文基于RJMCMC算法和RBF模型设计了对函数  $y = 1.1(1 - 5x + 2x^3)e^{-2.2x}$  的拟合。通过训练值训练基于RJMCMC的RBF神经网络，然后用测试值拟合结果，画出图像进行对比。算法部分由于是RJSA的一部分，算法分析将在后面进行。

### 3.2.3 实验结果

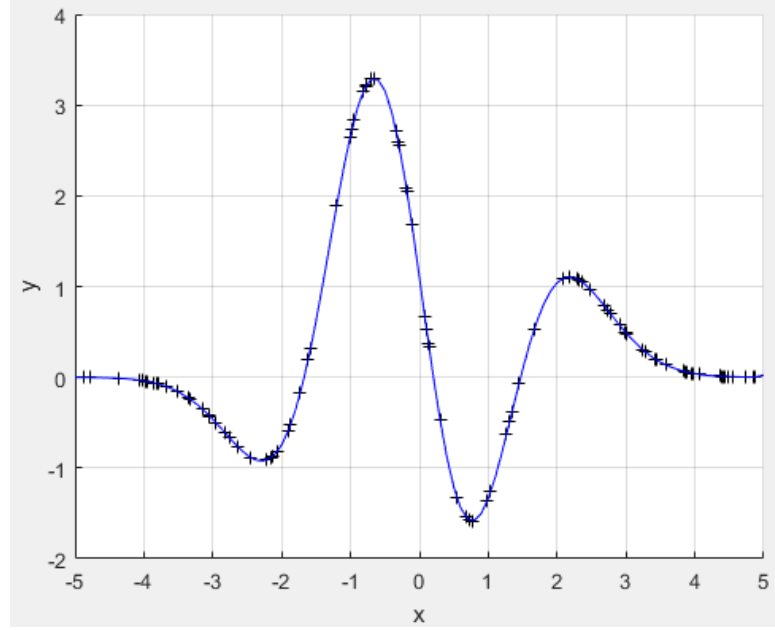


图5

图5是拟合的结果，黑色为训练的标准值，蓝色为由测试结果模拟的曲线，可以看出二者拟合得很好。

## 3.3 基于RJCMCMC和模拟退火算法的RBF模型选择

### 3.3.1 RJSA基本理论

RJSA由Nando de Freitas等人1999年提出[4]，可以对RBF网络进行更好的学习。RJSA的基本思想是随机初始化模型的参数，然后通过构造出中间的概率，由RJCMCMC算法的判断依据，在马尔科夫链之间跳转。输入训练数据得到相应的模型参数。

### 3.3.2 方法实现

假设输入 $x$ 和输出 $y$ 满足模型：

$$\begin{aligned} y_t &= b + \beta' x_t + n_t \quad k = 0 \\ y_t &= \sum_{j=1}^k a_j \phi(\|x_t - \mu_j\|) + b + \beta' x_t + n_t \quad k \geq 1 \end{aligned}$$

则进一步可以转化为题干比较简单的描述，在此不再赘述。RJSA算法中对于每一次马氏链状态的转移，都有中可能的操作(birth, death, split, merge, update)。五种操作的可能性 $b_k, d_k, s_k, m_k, u_k$ ，且满足和为1。文献[4]的作者在算法实现时规定每个概率都为0.2。

(a) birth过程是否接受数据主要取决于 $A_{birth}$ ,

$$A_{birth} = \min \{1, r_{birth}\} \quad r_{birth} = \left[ \prod_{i=1}^c \left( \frac{y'_{1:N,i} P_{i,k} y_{1:N,i}}{y'_{1:N,i} P_{i,k+1} y_{1:N,i}} \right)^{\left( \frac{N}{2} \right)} \right] \frac{\Im \exp(-C)}{k+1} \quad (3)$$

其中正交投影矩阵 $P_{ik}$ 取决于:

$$\mathbf{I}_N - \mathbf{D}(\hat{\boldsymbol{\mu}}_{1:k}, \mathbf{x}) [\mathbf{D}'(\hat{\boldsymbol{\mu}}_{1:k}, \mathbf{x}) \mathbf{D}(\hat{\boldsymbol{\mu}}_{1:k}, \mathbf{x})]^{-1} \mathbf{D}'(\hat{\boldsymbol{\mu}}_{1:k}, \mathbf{x})$$

(b) death过程是否接受数据也可由下式决定:

$$r_{death} = \left[ \prod_{i=1}^c \left( \frac{\mathbf{y}'_{1:N,i} \mathbf{P}_{i,k} \mathbf{y}_{1:N,i}}{\mathbf{y}'_{1:N,i} \mathbf{P}_{i,k-1} \mathbf{y}_{1:N,i}} \right)^{\left(\frac{N}{2}\right)} \right] \frac{k \exp(\mathcal{C})}{\mathfrak{F}} \quad (4)$$

(c) split过程会由当前的 $\mu$ 按以下方式生成 $\mu_1, \mu_2$ ,

$$\begin{aligned} \mu_1 &= \mu - u_{ms} \zeta^* \\ \mu_2 &= \mu + u_{ms} \zeta^* \end{aligned} \quad (5)$$

是否接受数据由下式决定:

$$r_{split} = \left[ \prod_{i=1}^c \left( \frac{\mathbf{y}'_{1:N,i} \mathbf{P}_{i,k} \mathbf{y}_{1:N,i}}{\mathbf{y}'_{1:N,i} \mathbf{P}_{i,k+1} \mathbf{y}_{1:N,i}} \right)^{\left(\frac{N}{2}\right)} \right] \frac{k \zeta^* \exp(-\mathcal{C})}{k+1} \quad (6)$$

(d) merge过程用 $\mu$ 替代原有的 $\mu_1, \mu_2$ ,  $\mu$ 按以下方式生成:

$$\mu = \frac{\mu_1 + \mu_2}{2}$$

是否接受数据由下式决定:

$$r_{merge} = \left[ \prod_{i=1}^c \left( \frac{\mathbf{y}'_{1:N,i} \mathbf{P}_{i,k} \mathbf{y}_{1:N,i}}{\mathbf{y}'_{1:N,i} \mathbf{P}_{i,k-1} \mathbf{y}_{1:N,i}} \right)^{\left(\frac{N}{2}\right)} \right] \frac{k \exp(\mathcal{C})}{\zeta^*(k-1)} \quad (7)$$

(e) update过程是否接受数据也可由下式决定:

$$\min \left\{ 1, \left[ \prod_{i=1}^c \left( \frac{\mathbf{y}'_{1:N,i} \mathbf{P}_{i,k} \mathbf{y}_{1:N,i}}{\mathbf{y}'_{1:N,i} \mathbf{P}_{i,k}^* \mathbf{y}_{1:N,i}} \right)^{\left(\frac{N}{2}\right)} \right] \right\} \quad (8)$$

模拟退火过程接受判据类似:

$$A_{SA} = \min \left\{ 1, \frac{\pi^{(1/T_i-1)}(\mathbf{z}^*)}{\pi^{(1/T_i-1)}(\mathbf{z})} \right\} \quad (9)$$

RBF网络参数 $\alpha$ 有上述迭代得到的矩阵即可算出

$$\hat{\boldsymbol{\alpha}}_{1:m,i} = [\mathbf{D}'(\boldsymbol{\mu}_{1:k}, \mathbf{x}) \mathbf{D}(\boldsymbol{\mu}_{1:k}, \mathbf{x})]^{-1} \mathbf{D}'(\boldsymbol{\mu}_{1:k}, \mathbf{x}) \mathbf{y}_{1:N,i} \quad (10)$$

高斯白噪声方差:

$$\begin{aligned} \hat{\sigma}_i^2 &= \frac{1}{N} (\mathbf{y}_{1:N,i} - \mathbf{D}(\hat{\boldsymbol{\mu}}_{1:k}, \mathbf{x}) \hat{\boldsymbol{\alpha}}_{1:m,i})' \\ &\quad \times (\mathbf{y}_{1:N,i} - \mathbf{D}(\hat{\boldsymbol{\mu}}_{1:k}, \mathbf{x}) \hat{\boldsymbol{\alpha}}_{1:m,i}) \\ &= \frac{1}{N} \mathbf{y}'_{1:N,i} \mathbf{P}_{i,k} \mathbf{y}_{1:N,i} \end{aligned} \quad (11)$$

如此便可得到模型的所有参数

$$\mathcal{M}_s = (\mu, \alpha, \sigma^2, \mathbf{k})$$



### 3.3.2 方法实现

---

#### Algorithm 2 RJSA Algorithm

---

```

Initialize S (k,μ)
for i = 1 to N do
    Generate u from Uniform distribution [0,1]
    Set the temperature with a cooling schedule
    if u<=0.2 then birth move using (3)
    else if u<=0.4 then death move using (4)
    else if u<=0.6 then split move using (6)
    else if u<=0.8 then merge move using (7)
    else update using (8)
    end if
    Evaluate the SA ratio using (9)

    Generate u from Uniform distribution [0,1]
    if u <= SA ratio then  $X^{(t)} = Y$ 
end for
Compute the coefficients  $\alpha_{1:m}$  using (10)

```

---

### 3.3.3 算法分析

使用MATLAB仿真的时候，发现split和merge如果要在循环中得到产生满足条件的均匀分布值u，有可能会循环较长的时间，我的改进是取消该循环，直接放到最后判断，如果不满足条件，接受概率置为0，不接受数据。这样能减少无效循环，缩短运行时间。

修改之后。每次大的循环，里面并没有嵌套循环，时间复杂度为O(N)。

### 3.3.4 基于RJSA的AIC,BIC,MDL的对比

AIC, BIC, MDL都是RBF模型选择的标准，对于RJSA算法，则简化为

$$\mathcal{P}_{\text{AIC}} = \xi \text{ 和 } \mathcal{P}_{\text{BIC}} = \mathcal{P}_{\text{MDL}} = \frac{\xi}{2} \log(N)。$$

文献[4]的算法还提出了修改方式：

$$\text{AIC准则: } C=c+1, \text{BIC和MDL准则: } C=\frac{c+1}{2} \log(N)。$$

由于AIC, BIC, MDL是基于算法的RBF模型选择，所以就RJSA算法而言，比较不同准则时，保持原有RJSA算法的参数不变，修改C的值即可。

本文对于对误差的估计采用的是训练结果相对于提供的y的相对偏移量来进行衡量，即 $\varepsilon = \frac{\|y-y_1\|}{\|y\|} * 100\%$ ，可以看出该式能较好地反映误差大小。就AIC, BIC&MDL

准则用RJSA算法仿真结果如下（均采用data1）

仿真100次马氏链转移（20次仿真取平均）

	$\varepsilon$	时间
AIC	3.0237%	15.03s
BIC&MDL	3.1001%	14.72s

可以看出基于同一算法的不同准则最终仿真得到的结果无论是误差还是时间基本相同，所以本文认为不同准则的选取对于RJSA算法的影响比较有限。为了进一步探索RJSA算法自身的优越性，结合前面的工作，本文对传统的MCMC进行了改造，使马氏链和RJSA的马氏链一致，以便于在不同的维度间转移，探索基于MCMC算法的RBF与RJSA的结果有何差异。

### 3.3.5 基于MCMC的RBF与RJSA算法的对比

---

#### Algorithm 3 MCMC RBF Algorithm

---

```

Initialize S (k,μ)
for i = 1 to N do
    new a μ from RBF
    Evaluate the acceptance probability using(8)

    Generate u from Uniform distribution [0,1]
    if u < r then substitute μ with new μ
end for
Compute the coefficients  $\alpha_{1:m}$  using (10)

```

---

为了与RJSA进行比较，同样用data1仿真20次，保证RBF函数相同，结果如下：

	$\varepsilon$	时间
RJMCMCSA	3.0348%	14.9s
MCMCRBF	4.1226%	38.3s

可以看出RJSA相比于普通的MCMC算法，在时间和精度上都有一定的提升，能对RBF网络进行更好的学习。

在对RJSA进行仿真时，发现将马氏链的状态数设为100和1000得到的结果并没有大的区别，这意味着RJSA算法在经过100次马氏链转移后就收敛到了比较好的结果，收敛性较好。

**注：所有实验统一仿真环境：win10操作系统，MATLAB2016a。**

## 4 结论(Conclusion)

MH算法是最基本的MCMC方法，对低维度能够进行高效率的采样，但是对提议分布有着一定的要求，提议分布越接近目标分布得到的效果越好。一般而言，利用均匀分布也能得到较好的采样结果。对于维度变化，维度较高的采样，传统的MCMC方法不是很有效，通常对于可变维度采样RJMC方法，对于高维度，则多采用吉布斯采样。

本文使用RJSA算法实现了基于MCMC的RBF神经网络模型，用给定的数据进行训练并测试。并且和变换的MCMCRBF算法进行了比较，得出以下结论：

- (1) RJSA能在较短的时间内得到较为精确的结果，能够更好的帮助RBF网络学习；
- (2) RJSA算法的收敛性较强，在有限的迭代次数下能有较好的收敛；
- (3) AIC, BIC, MDL准则对RJSA算法的影响有限，上述是选择的准则，算法对RBF的实际影响更大；
- (4) 与MCMCRBF算法对比可知，RJSA算法和精度均更好，所以最后的测试数据以RJSA得到的数据为准。

## 5 致谢(Acknowledgement)

感谢professor OU在课堂上马尔科夫过程及MH算法的相关讲解，使我对MCMC有了最基本的认识。

感谢yg同学在RJMC知识上对我的指导，让我对算法有了初步的了解。

感谢wsy同学在误差分析和参数调校上对我的指导，让我在核函数的选取和参数调整，以及数据分析有了更好的理解。

## 参考文献

- [1] 林元烈, 应用随机过程. 清华大学出版社, 2002.
- [2] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
- [3] J. S. Liu, *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.
- [4] C. Andrieu, N. D. Freitas, and A. Doucet, "Reversible jump mcmc simulated annealing for neural networks," in *Sixteenth Conference on Uncertainty in Artificial Intelligence*, pp. 11–18, 2000.
- [5] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.
- [6] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, no. 2, pp. págs. 15–18, 1978.
- [7] J. Rissanen, "Stochastic complexity," *Journal of the Royal Statistical Society*, vol. 49, no. 3, pp. 223–239, 1987.

- 
- [8] P. M. Djuric, “Asymptotic map criteria for model selection,” *IEEE Transactions on Signal Processing*, vol. 46, no. 10, pp. 2726–2735, 2002.
  - [9] P. J. Green, “Reversible jump markov chain monte carlo computation and bayesian model determination,” *Biometrika*, vol. 82, no. 4, pp. 711–732, 1995.
  - [10] C. Andrieu, J. D. Freitas, and A. Doucet, “Robust full bayesian learning for neural networks,” 1999.
  - [11] C. Andrieu, P. M. Djurić, and A. Doucet, “Model selection by mcmc computation.,” *Signal Processing*, vol. 81, no. 1, pp. 19–37, 2001.