证券研究报告 发布时间：2017-01-11

证券研究报告/投资策略报告

金融工程：HMM 指数择时研究之理论篇

报告摘要:

本篇报告是 HMM 指数择时系列研究报告的第二篇——理论篇，我们在上一篇实战篇的基础上，首先回顾了 HMM 三大算法：概率计算算法、学习算法及预测算法；同时，我们也提供了五个案例解析。其次提出了基于无监督学习 HMM 评估方法，得到 dist 是一个很好的评估模型估计误差、状态估计精度及状态估计稳定性的指标。最后，我们根据 dist 值进一步对上一篇报告中的观测变量日收益率指标进行反思，同时把日 $MA_{short/long}$ 指标和日收益率指标进行对比，根据 dist 的大小与对 N 的敏感度，得出当训练长度足够长时，日 $MA_{short/long}$ 指标较日收益率指标更适合作为 HMM 的观测变量。

相关报告

《金融工程：基金仓位估测模型之一》
2016-06-17

《金融工程之择时：期现回复 S 型动量指标》
2016-06-23

《金融工程：偏度调整的收益率 MACD 择时信号》
2016-07-13

《金融工程：偏度调整的收益率 MACD 择时信号(二)》
2016-07-25

《金融工程：基金仓位估测模型之二—误差修正》
2016-08-04

《金融工程：个股及基金绩效分析与风险归因分析》
2016-08-26

《风险预测、纯因子组合及基于收益的风格分析》
2016-09-26

《金融工程：HMM 指数择时研究之实战篇》
2016-09-26

《FOF：从大类资产配置到股票型基金的选取》
2016-12-01

《大类资产配置（1）：短期长期风险度量和超额收益的边际效益递减》
2016-12-17

证券分析师：陈亚龙
执业证书编号：S0550516050001

研究助理：肖承志
执业证书编号：S0550116080014
021-2036 1264 xiaocz@nesc.cn

目录

图表目录

1. HMM 算法介绍

我们在上一篇报告《金融工程:HMM 指数择时研究之实战篇》中已介绍了 HMM 模型以及如何用 HMM 模型进行择时，但关于模型算法的部分还未详细介绍。本篇报告的第 1 章我们将详细地介绍 HMM 的三个主要算法:前向后向算法、Baum-Welch 算法和 Viterbi 算法。

首先，我们简要回顾一些重要的符号及其含义。

表 1：符号及含义

符号	含义
N	隐状态个数
M	观测变量是离散型分布的情况下，观测变量结果个数
$P(X)$	事件X发生的概率
$P(X\|Y)$	事件Y发生的条件下事件X发生的概率
$O(x^n)$	与x^n同阶，(计算数量上)复杂度与x^n相当
$\alpha_t(i)$	状态为q_i的前向概率
$\beta_t(i)$	状态为q_i的后向概率
$\mathbf{N}(\mu,\sigma^2)$	均值为μ、方差为σ^2的正态分布
$\mathbf{U}[a,b]$	区间$[a,b]$上的均匀分布
$\mathbf{1}_{\{X=x\}}$	示性函数，$\mathbf{1}_{\{X=x\}}=\begin{cases}1, 当X=x时\\0, 当X\neq x时\end{cases}$
$\boldsymbol{Q}$	隐状态集合，$\boldsymbol{Q}=\{q_1,q_2,\dots,q_N\}$，
$\boldsymbol{V}$	观测变量集合，当观测变量是离散型分布：$\boldsymbol{V}=\boldsymbol{V}_M=\{v_1,v_2,\dots,v_M\}$; 当观测变量是连续型分布(以正态分布为例)：$\boldsymbol{V}=\boldsymbol{V}_\infty=(-\infty,+\infty)=\mathbb{R}$
$\boldsymbol{I}$	长度为T的隐状态序列，$\boldsymbol{I}=(i_1,i_2,\dots,i_T)$
$\boldsymbol{O}$	长度为T的观测序列，$\boldsymbol{O}=(o_1,o_2,\dots,o_T)$
$\boldsymbol{A}$	状态转移概率矩阵，$\boldsymbol{A}=[a_{ij}]_{N\times N}$，其中$a_{ij}=P(i_{t+1}=q_j\|i_t=q_i)$，$i,j=1,2,\dots,N$， 表示在$t$时刻处于状态$q_i$的条件下在$t+1$转移到状态$q_j$的概率
$\boldsymbol{B}$	观测概率分布向量，$\boldsymbol{B}=[b_j(x)]_{N\times 1}$， 当$x$是离散型分布：$b_j(x)=P(x=v_k\|i_t=q_j)=b_{jk}$，$x\in\boldsymbol{V}_M$，$j=1,2,\dots,N$，$k=1,2,\dots,M$; 当$x$是连续型分布(以正态分布为例)：$b_j(x)=\frac{1}{\sqrt{2\pi}\sigma_j}\exp\left(-\frac{(x-\mu_j)^2}{2\sigma_j^2}\right)$，$x\in\boldsymbol{V}_\infty$，$j=1,2,\dots,N$
$\boldsymbol{\pi}$	初始状态概率向量，$\boldsymbol{\pi}=(\pi_1,\pi_2,\dots,\pi_N)$，其中$\pi_i=P(i_1=q_i)$，$i=1,2,\dots,N$
$\boldsymbol{\lambda}$	隐马尔可夫模型参数，$\boldsymbol{\lambda}=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$
$\gamma_t(i)$	给定模型$\boldsymbol{\lambda}$和观测序列$\boldsymbol{O}$，在时刻t处于状态q_i的概率，$\gamma_t(i)=P(i_t=q_i\|\boldsymbol{O},\boldsymbol{\lambda})$
$\xi_t(i,j)$	给定模型$\boldsymbol{\lambda}$和观测序列$\boldsymbol{O}$，在时刻t处于状态q_i且在$t+1$处于状态q_j的概率， $\xi_t(i,j)=P(i_t=q_i,i_{t+1}=q_j\|\boldsymbol{O},\boldsymbol{\lambda})$
Q	EM 算法的 E 步，求Q函数，$Q(\boldsymbol{\lambda},\bar{\boldsymbol{\lambda}})=\sum_{\boldsymbol{I}}\log P(\boldsymbol{O},\boldsymbol{I}\|\boldsymbol{\lambda})P(\boldsymbol{O},\boldsymbol{I}\|\bar{\boldsymbol{\lambda}})$

数据来源：东北证券

其次，回顾 HMM 的两个基本假设:

- **齐次马尔可夫性假设**,即假设马尔可夫链在任意时刻t的状态只依赖于前一个时刻的状态，而与其他时刻的状态及观测无关，也与时刻t无关。即

$$P(i_t|i_{t-1},o_{t-1},\dots,i_1,o_1)=P(i_t|i_{t-1}),\qquad t=1,2,\dots,T$$

- **观测独立假设性**，即假设任意时刻的观测到的变量的分布只依赖于该时刻

的状态与其他时刻的状态和观测值无关。即

$$P(o_t|i_T,o_T,i_{T-1},o_{T-1},\dots,i_t,o_t,\dots,i_1,o_1)=P(o_t|i_t)$$

最后，回顾隐马尔可夫模型研究的三个基本问题：

- **概率计算问题**：给定模型$\boldsymbol{\lambda}=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$和观测序列$\boldsymbol{O}=(o_1,o_2,\dots,o_T)$，如何评估模型与观测序列之间的匹配度？换言之，计算概率$P(\boldsymbol{O}|\boldsymbol{\lambda})$。
- **学习问题**：已知观测序列$\boldsymbol{O}=(o_1,o_2,\dots,o_T)$，估计模型参数$\boldsymbol{\lambda}=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$，使得在该模型下观测序列概率$P(\boldsymbol{O}|\boldsymbol{\lambda})$最大？即用极大似然估计方法估计参数。
- **预测问题**：已知模型$\boldsymbol{\lambda}=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$和观测序列$\boldsymbol{O}=(o_1,o_2,\dots,o_T)$，求使得条件概率$P(\boldsymbol{I}|\boldsymbol{O})$最大的$\boldsymbol{I}=(i_1,i_2,\dots,i_T)$？即通过观测序列找出最有可能对应的隐状态序列。该问题也称为**解码问题**。

下面，我们将重点介绍与这三个问题相关的三类算法：概率计算算法，学习算法以及预测算法。以下内容大多来自《统计学习方法》(李航著)一书，感兴趣的读者可以阅读相关章节。

1.1. 概率计算算法

本节介绍计算观测概率序列概率$P(\boldsymbol{O}|\boldsymbol{\lambda})$的前向(forward)与后向(backward)算法。先介绍概念上可行但计算上不可行的直接计算法。

1.1.1. 直接计算算法

给定模型$\boldsymbol{\lambda}=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$和观测序列$\boldsymbol{O}=(o_1,o_2,\dots,o_T)$，计算观测序列$\boldsymbol{O}$出现的概率$P(\boldsymbol{O}|\boldsymbol{\lambda})$。最直接的方法是按照概率公式直接计算。通过列举所有可能的长度为T的状态序列$\boldsymbol{I}=(i_1,i_2,\dots,i_T)$，求各个状态序列$\boldsymbol{I}$与观测序列$\boldsymbol{O}=(o_1,o_2,\dots,o_T)$的联合概率$P(\boldsymbol{O},\boldsymbol{I}|\boldsymbol{\lambda})$，然后对所有可能的状态序列求和，得到$P(\boldsymbol{O}|\boldsymbol{\lambda})$。

给定$\boldsymbol{\lambda}$，状态序列$\boldsymbol{I}=(i_1,i_2,\dots,i_T)$出现的概率是

$$P(\boldsymbol{I}|\boldsymbol{\lambda})=\pi_{i_1}a_{i_1i_2}a_{i_2i_3}\dots a_{i_{T-1}i_T}$$

给定$\boldsymbol{\lambda}$和状态序列$\boldsymbol{I}=(i_1,i_2,\dots,i_T)$，观测序列$\boldsymbol{O}=(o_1,o_2,\dots,o_T)$的出现概率

$$P(\boldsymbol{O}|\boldsymbol{\lambda},\boldsymbol{I})=b_{i_1}(o_1)b_{i_2}(o_2)\dots b_{i_T}(o_T)$$

给定$\boldsymbol{\lambda}$，$\boldsymbol{O}$和$\boldsymbol{I}$同时出现的联合概率为

$$P(\boldsymbol{O},\boldsymbol{I}|\boldsymbol{\lambda})=P(\boldsymbol{O}|\boldsymbol{\lambda},\boldsymbol{I})P(\boldsymbol{I}|\boldsymbol{\lambda})=\pi_{i_1}b_{i_1}(o_1)a_{i_1i_2}b_{i_2}(o_2)\dots a_{i_{T-1}i_T}b_{i_T}(o_T)$$

然后，对所有可能的状态序列$\boldsymbol{I}$求和，得到观测序列$\boldsymbol{O}$的概率$P(\boldsymbol{O}|\boldsymbol{\lambda})$，即

$$P(\boldsymbol{O}|\boldsymbol{\lambda})=\sum_{\boldsymbol{I}}P(\boldsymbol{O}|\boldsymbol{\lambda},\boldsymbol{I})P(\boldsymbol{I}|\boldsymbol{\lambda})=\sum_{i_1,i_2,\dots,i_T}\pi_{i_1}b_{i_1}(o_1)a_{i_1i_2}b_{i_2}(o_2)\dots a_{i_{T-1}i_T}b_{i_T}(o_T)$$

但是上述公式计算量很大，是$O(TN^T)$阶的，当序列长度T很大的时候，这种算法不可行。下面介绍计算观测序列概率$P(\boldsymbol{O}|\boldsymbol{\lambda})$的有效算法：前向-后向算法(forward-backward algorithm)。

1.1.2. 前向算法

定义给定隐马尔可夫模型$\boldsymbol{\lambda}$，定义到时刻t部分观察序列为$o_1,o_2,\dots,o_t$且状态为q_i的概率为**前向概率**，记作

$$\alpha_t(i)=P(o_1,o_2,\dots,o_t,i_t=q_i|\boldsymbol{\lambda})$$

可以递推地求得前向概率$\alpha_t(i)$及观测序列概率$P(\boldsymbol{O}|\boldsymbol{\lambda})$。

算法(观测序列概率的前向算法)

输入：隐马尔可夫模型$\boldsymbol{\lambda}$，观测序列$\boldsymbol{O}$;

输出：观测序列概率$P(\boldsymbol{O}|\boldsymbol{\lambda})$。

步骤

（1） **初值**

$$\alpha_1(i) = \pi_i b_i(o_1), \qquad i = 1,2,\dots,N$$

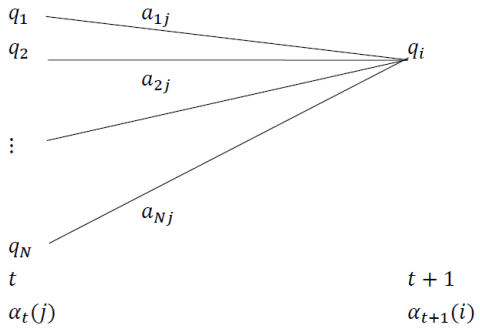
（2） **递推**对$t = 1,2,\dots,T-1,$

$$\alpha_{t+1}(i) = \left[\sum_{j=1}^{N} \alpha_{t+1}(j)\, a_{ji}\right] b_i(o_{t+1}), \qquad i = 1,2,\dots,N$$

（3） **终止**

$$P(\boldsymbol{O}|\boldsymbol{\lambda}) = \sum_{i=1}^{N} \alpha_T(i)$$

图 1：前向概率的递推公式



数据来源：东北证券，《统计学习方法》(李航著)

前向算法，步骤(1)初始化前向概率，是初始时刻的状态$i_1 = q_i$和观测o_1的联合概率。步骤(2)是前向概率的递推公式，计算到时刻$t+1$部分观测序列为$o_1, o_2, \dots, o_t, o_{t+1}$且在时刻$t+1$处于状态$q_i$的前向概率，如图 1 所示。在步骤(2)式子的方括弧里，既然$\alpha_t(j)$是到时刻t观测到$o_1, o_2, \dots, o_t$并在时刻t处于状态q_j的前向概率，那么乘积$\alpha_t(j)a_{ji}$就是到时刻t观测到$o_1, o_2, \dots, o_t$并在时刻t处于状态q_j而在时刻$t+1$到达状态q_i的联合概率。对这个乘积在时刻t的所有可能的N个状态q_j求和，其结果就是到时刻t观测为$o_1, o_2, \dots, o_t$并在时刻$t+1$处于状态q_i的联合概率。方括弧里的值与观测概率$b_i(o_{t+1})$的乘积恰好是到时刻$t+1$观测到$o_1, o_2, \dots, o_t, o_{t+1}$并在时刻$t+1$处于状态$q_i$的前向概率$\alpha_{t+1}(i)$。步骤(3)给出$P(\boldsymbol{O}|\boldsymbol{\lambda})$的计算公式。因为

$$\alpha_T(i) = P(o_1, o_2, \dots, o_T, i_T = q_i|\boldsymbol{\lambda})$$

所以

$$P(\boldsymbol{O}|\boldsymbol{\lambda}) = \sum_{i=1}^{N} \alpha_T(i)$$

前向算法实际是基于“状态序列的路径结构”递推计算$P(\boldsymbol{O}|\boldsymbol{\lambda})$的算法。前向算法高效的关键是其局部计算前向概率，然后利用路径结构将前向概率“递推”到全局，

得到$P(\boldsymbol{O}|\boldsymbol{\lambda})$。具体地，在时刻$t=1$，计算$\alpha_1(i)$的$N$个值$(i=1,2,\dots,N)$；在各个时刻$t=1,2,\dots,T-1$，计算$\alpha_{t+1}(i)$的$N$个值$(i=1,2,\dots,N)$，而且每个$\alpha_{t+1}(i)$的计算利用前一时刻$N$个$\alpha_t(j)$。减少计算量的原因在于每一次计算直接引用前一个时刻的计算结果，避免重复计算。容易推出，这样利用前向概率计算$P(\boldsymbol{O}|\boldsymbol{\lambda})$的计算量是$O(N^2T)$阶的，而不是直接计算的$O(TN^T)$阶。

1.1.3. 后向算法

定义给定隐马尔可夫模型$\boldsymbol{\lambda}$，定义在时刻t状态为q_i的条件下，从$t+1$到T的部分观测序列为$o_{t+1},o_{t+2},\dots,o_T$的概率为**后向概率**，记作

$$\beta_t(i)=P(o_{t+1},o_{t+2},\dots,o_T|i_t=q_i,\boldsymbol{\lambda})$$

可以用倒向递推的方法求得后向概率$\beta_t(i)$及观测序列概率$P(\boldsymbol{O}|\boldsymbol{\lambda})$。

算法(观测序列概率的后向算法)

输入：隐马尔可夫模型$\boldsymbol{\lambda}$，观测序列$\boldsymbol{O}$;

输出：观测序列概率$P(\boldsymbol{O}|\boldsymbol{\lambda})$。

步骤

（1） **终值**

$$\beta_T(i)=1,\qquad i=1,2,\dots,N$$

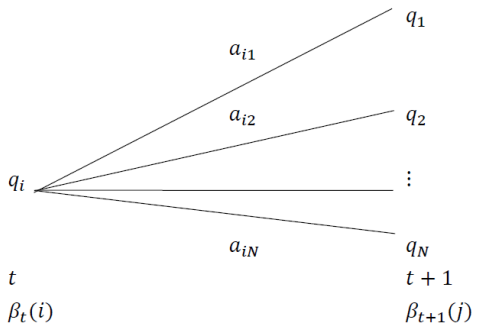
（2） **倒向递推**对$t=T-1,T-2,\dots,1$

$$\beta_t(i)=\sum_{j=1}^{N}a_{ij}\,b_j(o_{t+1})\beta_{t+1}(j),\qquad i=1,2,\dots,N$$

（3） **终止**

$$P(\boldsymbol{O}|\boldsymbol{\lambda})=\sum_{i=1}^{N}\pi_i b_i(o_1)\,\beta_1(i)$$

图 2: 后向概率的递推公式



数据来源：东北证券，《统计学习方法》(李航著)

步骤(1)初始化后向概率，对最终时刻的所有状态q_i规定$\beta_T(i)=1$。步骤(2)是后向概率的倒向递推公式。如图 2 所示，为了计算在时刻t状态为q_i的条件下时刻$t+1$之后的观测序列为$o_{t+1},o_{t+2},\dots,o_T$的后向概率$\beta_t(i)$，只需考虑在时刻$t+1$所有可能的$N$个状态$q_j$的转移概率(即$a_{ij}$项)，以及在此状态下的观测$o_{t+1}$的观测概率(即$b_j(o_{t+1})$项)，然后考虑状态$q_j$之后的观测序列的后向概率(即$\beta_{t+1}(j)$项)。步骤(3)求$P(\boldsymbol{O}|\boldsymbol{\lambda})$的思路与步骤(2)一致，只是初始概率$\pi_i$代替转移概率。

利用前向概率和后向概率的定义将观测序列概率$P(\boldsymbol{O}|\boldsymbol{\lambda})$统一写成

$$P(\boldsymbol{O}|\boldsymbol{\lambda}) = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)\, a_{ij}\, b_j(o_{t+1})\beta_{t+1}(j), t = 1,2,\dots,T-1$$

此式当$t=1$和$t=T-1$时分别为 1.1.2（3）和 1.1.3（3）。

1.1.4. 一些概率与期望值的计算

利用前向概率和后向概率，可以得到关于单个状态和两个状态概率的计算公式。

1. **给定模型$\boldsymbol{\lambda}$和观测$\boldsymbol{O}$，在时刻$\boldsymbol{t}$处于状态$\boldsymbol{q_i}$的概率**。记

$$\gamma_t(i) = P(i_t = q_i|\boldsymbol{O},\boldsymbol{\lambda})$$

可以通过前向后向概率计算。事实上，

$$\gamma_t(i) = P(i_t = q_i|\boldsymbol{O},\boldsymbol{\lambda}) = \frac{P(i_t = q_i,\boldsymbol{O}|\boldsymbol{\lambda})}{P(\boldsymbol{O}|\boldsymbol{\lambda})}$$

由前向概率$\alpha_t(i)$和后向概率$\beta_t(i)$定义可知:

$$\alpha_t(i)\beta_t(i) = P(i_t = q_i,\boldsymbol{O}|\boldsymbol{\lambda})$$

于是得到:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(\boldsymbol{O}|\boldsymbol{\lambda})} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}$$

2. **给定模型$\boldsymbol{\lambda}$和观测$\boldsymbol{O}$，在时刻$\boldsymbol{t}$处于状态$\boldsymbol{q_i}$且在$\boldsymbol{t+1}$处于状态$\boldsymbol{q_j}$的概率**。记

$$\xi_t(i,j) = P(i_t = q_i, i_{t+1} = q_j|\boldsymbol{O},\boldsymbol{\lambda})$$

可以通过前向后向概率计算:

$$\xi_t(i,j) = \frac{P(i_t = q_i, i_{t+1} = q_j,\boldsymbol{O}|\boldsymbol{\lambda})}{P(\boldsymbol{O}|\boldsymbol{\lambda})} = \frac{P(i_t = q_i, i_{t+1} = q_j,\boldsymbol{O}|\boldsymbol{\lambda})}{\sum_{i=1}^{N}\sum_{j=1}^{N}P(i_t = q_i, i_{t+1} = q_j,\boldsymbol{O}|\boldsymbol{\lambda})}$$

而

$$P\left(i_t = q_i, i_{t+1} = q_j,\boldsymbol{O}|\boldsymbol{\lambda}\right) = \alpha_t(i)\, a_{ij}\, b_j(o_{t+1})\beta_{t+1}(j)$$

所以

$$\xi_t(i,j) = \frac{\alpha_t(i)\, a_{ij}\, b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)\, a_{ij}\, b_j(o_{t+1})\beta_{t+1}(j)}$$

3. **将$\boldsymbol{\gamma_t(i)}$和$\boldsymbol{\xi_t(i,j)}$对各个时刻$\boldsymbol{t}$求和，可以得到一些有用的期望值**:

（1） 在观测$\boldsymbol{O}$下状态i出现的期望值

$$\sum_{t=1}^{T}\gamma_t(i)$$

（2） 在观测$\boldsymbol{O}$下由状态i转移的期望值

$$\sum_{t=1}^{T-1}\gamma_t(i)$$

（3） 在观测$\boldsymbol{O}$下由状态i转移到状态j的期望值

$$\sum_{t=1}^{T-1}\xi_t(i,j)$$

1.2. 学习算法

隐马尔可夫模型的学习，根据训练数据是包括观测序列和对应的状态序列还是只有观测序列，可以分为监督学习和非监督学习。本节首先介绍监督学习算法，而后介绍非监督学习算法——Baum-Welch 算法(也就是 EM 算法，关于 EM 算法的内容，请参见《统计学习方法》(李航著)的第 9 章)。

1.2.1. 监督学习方法

假设已给训练数据包含S个长度相同的观测序列和对应的状态序列$\{(\boldsymbol{O}_1,\boldsymbol{I}_1),(\boldsymbol{O}_2,\boldsymbol{I}_2),\dots,(\boldsymbol{O}_s,\boldsymbol{I}_s)\}$，那么可以利用极大似然法来估计隐马尔可夫模型的参数。具体方法如下。

1. **转移概率a_{ij}的估计**

设样本中时刻t处于状态i时刻$t+1$转移到状态j的频数为A_{ij}，那么状态转移概率a_{ij}的估计是

$$\hat{a}_{ij}=\frac{A_{ij}}{\sum_{j=1}^{N}A_{ij}},\qquad i=1,2,\dots,N;\ j=1,2,\dots,N$$

2. **观测概率$\boldsymbol{b_j(x)}$的估计**

当x取值是离散情况：设样本中状态为j并且观测为k的频数是B_{jk}，那么状态为j观测为k的概率b_{jk}的估计是

$$\hat{b}_{jk}=\frac{B_{jk}}{\sum_{k=1}^{M}B_{jk}},\qquad j=1,2,\dots,N;\ k=1,2,\dots,M$$

当$\boldsymbol{x}$取值是连续情况(以多元正态分布为例)，观测正态分布的均值和协方差

$$\hat{\boldsymbol{\mu}}_j=\frac{\sum_{t=1}^{T}\boldsymbol{o}_t\mathbf{1}_{\{q_t=j\}}}{\sum_{t=1}^{T}\mathbf{1}_{\{q_t=j\}}},\qquad j=1,2,\dots,N;$$

$$\hat{\boldsymbol{\Sigma}}_j=\frac{\sum_{t=1}^{T}(\boldsymbol{o}_t-\hat{\boldsymbol{\mu}}_j)(\boldsymbol{o}_t-\hat{\boldsymbol{\mu}}_j)^{\mathrm{T}}\mathbf{1}_{\{q_t=j\}}}{\sum_{t=1}^{T}\mathbf{1}_{\{q_t=j\}}},\qquad j=1,2,\dots,N;$$

3. **初始状态概率$\boldsymbol{\pi_i}$的估计$\hat{\boldsymbol{\pi}}_i$为S个样本中初始状态为$\boldsymbol{q_i}$的频率**

由于监督学习需要使用训练数据，而人工标注训练数据往往代价很高，有时就会利用非监督学习的办法。

1.2.2. Baum-Welch 算法

假设给定训练数据只包含S个长度为T的观测序列$\{\boldsymbol{O}_1,\boldsymbol{O}_2,\dots,\boldsymbol{O}_s\}$而没有对应的状态序列，目标是学习隐马尔可夫模型$\boldsymbol{\lambda}=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$的参数。我们将观测序列数据看作观测数据$\boldsymbol{O}$，状态序列数据看作不可预测的隐数据$\boldsymbol{I}$，那么隐马尔可夫模型事实上是一个含有隐变量的概率模型

$$P(\boldsymbol{O}|\boldsymbol{\lambda})=\sum_{I}P(\boldsymbol{O}|\boldsymbol{I},\boldsymbol{\lambda})\,P(\boldsymbol{I}|\boldsymbol{\lambda})$$

它的参数学习可以由 EM 算法实现。

1. **确定完全数据的对数似然函数**

所有观测数据写成$\boldsymbol{O}=(o_1,o_2,\dots,o_T)$，所有隐数据写成$\boldsymbol{I}=(i_1,i_2,\dots,i_T)$，完全数据是$(\boldsymbol{O},\boldsymbol{I})=(o_1,o_2,\dots,o_T,i_1,i_2,\dots,i_T)$。完全数据的对数似然函数是$\log P(\boldsymbol{O},\boldsymbol{I}|\boldsymbol{\lambda})$。

2. **EM 算法 E 步**：求Q函数$Q(\boldsymbol{\lambda},\bar{\boldsymbol{\lambda}})$

$$Q(\boldsymbol{\lambda},\bar{\boldsymbol{\lambda}})=\sum_{\boldsymbol{I}}\log P(\boldsymbol{O},\boldsymbol{I}|\boldsymbol{\lambda})P(\boldsymbol{O},\boldsymbol{I}|\bar{\boldsymbol{\lambda}})$$

其中，$\bar{\boldsymbol{\lambda}}$是隐马尔可夫模型参数的当前估计值，$\boldsymbol{\lambda}$是要极大化的隐马尔可夫模型参数。

$$P(\boldsymbol{O},\boldsymbol{I}|\boldsymbol{\lambda})=\pi_{i_1}b_{i_1}(o_1)a_{i_1i_2}b_{i_2}(o_2)\dots a_{i_{T-1}i_T}b_{i_T}(o_T)$$

于是函数$Q(\boldsymbol{\lambda},\bar{\boldsymbol{\lambda}})$可以写成:

$$Q(\boldsymbol{\lambda},\bar{\boldsymbol{\lambda}})=\sum_{\boldsymbol{I}}\log\pi_{i_1}P(\boldsymbol{O},\boldsymbol{I}|\bar{\boldsymbol{\lambda}})+\sum_{\boldsymbol{I}}\left(\sum_{t=1}^{T-1}\log a_{i_ti_{t+1}}\right)P(\boldsymbol{O},\boldsymbol{I}|\bar{\boldsymbol{\lambda}})$$

$$+\sum_{\boldsymbol{I}}\left(\sum_{t=1}^{T}\log b_{i_t}(o_t)\right)P(\boldsymbol{O},\boldsymbol{I}|\bar{\boldsymbol{\lambda}})$$

式中求和都是对所有训练数据的序列总长度T进行的。

3. **EM 算法的 M 步**：极大化Q函数$Q(\boldsymbol{\lambda},\bar{\boldsymbol{\lambda}})$，求模型参数$\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi}$

由于要极大化的参数在上式中单独地出现在 3 个项中，所以只需对各项分别极大化。

（1） 第 1 项可以写成

$$\sum_{\boldsymbol{I}}\log\pi_{i_1}P(\boldsymbol{O},\boldsymbol{I}|\bar{\boldsymbol{\lambda}})=\sum_{i=1}^{N}\log\pi_iP(\boldsymbol{O},i_1=i|\bar{\boldsymbol{\lambda}})$$

注意到π_i满足约束条件$\sum_{i=1}^{N}\pi_i=1$，利用拉格朗日乘子法，写出拉格朗日函数:

$$\sum_{i=1}^{N}\log\pi_iP(\boldsymbol{O},i_1=i|\bar{\boldsymbol{\lambda}})+\gamma(\sum_{i=1}^{N}\pi_i-1)$$

对其求偏导数并令结果为 0

$$\frac{\partial}{\partial\pi_i}\left[\sum_{i=1}^{N}\log\pi_iP(\boldsymbol{O},i_1=i|\bar{\boldsymbol{\lambda}})+\gamma\left(\sum_{i=1}^{N}\pi_i-1\right)\right]=0$$

得

$$P(\boldsymbol{O},i_1=i|\bar{\boldsymbol{\lambda}})+\gamma\pi_i=0$$

对i求和得到γ

$$\gamma=-P(\boldsymbol{O}|\bar{\boldsymbol{\lambda}})$$

代入即得

$$\pi_i=\frac{P(\boldsymbol{O},i_1=i|\bar{\boldsymbol{\lambda}})}{P(\boldsymbol{O}|\bar{\boldsymbol{\lambda}})}$$

（2） 第 2 项可以写成

$$\sum_{\boldsymbol{I}}\left(\sum_{t=1}^{T-1}\log a_{i_ti_{t+1}}\right)P(\boldsymbol{O},\boldsymbol{I}|\bar{\boldsymbol{\lambda}})=\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T-1}\log a_{ij}\,P(\boldsymbol{O},i_t=i,i_{t+1}=j|\bar{\boldsymbol{\lambda}})$$

类似第 1 项，应用具有约束条件$\sum_{i=1}^{N}a_{ij}=1$的拉格朗日乘子法可以求出

$$a_{ij}=\frac{\sum_{t=1}^{T-1}P(\boldsymbol{O},i_t=i,i_{t+1}=j|\bar{\boldsymbol{\lambda}})}{\sum_{t=1}^{T-1}P(\boldsymbol{O},i_t=i|\bar{\boldsymbol{\lambda}})}$$

（3） 第 3 项改写成

$$\sum_{\boldsymbol{I}}\left(\sum_{t=1}^{T}\log b_{i_t}(o_t)\right)P(\boldsymbol{O},\boldsymbol{I}|\bar{\boldsymbol{\lambda}})=\sum_{j=1}^{N}\sum_{t=1}^{T}\log b_j(o_t)P(\boldsymbol{O},i_t=j|\bar{\boldsymbol{\lambda}})$$

同样用拉格朗日乘子法，观测变量是离散型分布时，约束条件是$\sum_{k=1}^{N}b_{jk}=1$。注意，只有在$o_t=v_k$时$b_j(o_t)$对b_{jk}的偏导数才不为 0，以$\mathbf{1}_{\{o_t=v_k\}}$表示。求得

$$b_{jk}=\frac{\sum_{t=1}^{T}P(\boldsymbol{O},i_t=j|\bar{\boldsymbol{\lambda}})\mathbf{1}_{\{o_t=v_k\}}}{\sum_{t=1}^{T}P(\boldsymbol{O},i_t=j|\bar{\boldsymbol{\lambda}})}$$

同理，观测变量是连续型分布(以多元正态分布为例)时，

$$\boldsymbol{\mu}_j=\frac{\sum_{t=1}^{T}\boldsymbol{o}_t\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}$$

$$\boldsymbol{\Sigma}_j=\frac{\sum_{t=1}^{T}\gamma_t(j)(\boldsymbol{o}_t-\boldsymbol{\mu}_j)(\boldsymbol{o}_t-\boldsymbol{\mu}_j)^{\mathrm{T}}}{\sum_{t=1}^{T}\gamma_t(j)}$$

1.2.3. Baum-Welch 模型参数估计公式

将前面各式中的概率分别用$\gamma_t(i)$，$\xi_t(i,j)$表示，则可将相应的公式写成:

$$a_{ij}=\frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}$$

$$\pi_i=\gamma_1(i)$$

当观测变量是离散型分布时，

$$b_{jk}=\frac{\sum_{t=1}^{T}\gamma_t(j)\mathbf{1}_{\{o_t=v_k\}}}{\sum_{t=1}^{T}\gamma_t(j)}$$

当观测变量是连续型分布(以多元正态分布为例)时，

$$\boldsymbol{\mu}_j=\frac{\sum_{t=1}^{T}\boldsymbol{o}_t\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}$$

$$\boldsymbol{\Sigma}_j=\frac{\sum_{t=1}^{T}\gamma_t(j)(\boldsymbol{o}_t-\boldsymbol{\mu}_j)(\boldsymbol{o}_t-\boldsymbol{\mu}_j)^{\mathrm{T}}}{\sum_{t=1}^{T}\gamma_t(j)}$$

上述公式即为 Baum-Welch 算法(Baum-Welch algorithm)，它是 EM 算法在隐马尔可夫模型学习中的具体实现，由 Baum 和 Welch 提出。但必须指出，上述得到的参数并非全局最优参数，而是局部最优的参数。这意味着，这些参数的估计，依赖于迭代的初值的选取。

算法(Baum-Welch)

输入：观测数据$\boldsymbol{O}=(o_1,o_2,\dots,o_T)$;

输出：隐马尔可夫模型参数。

（1） **初始化**对$n=0$，选取$a_{ij}^{(0)},b_{jk}^{(0)},\pi_i^{(0)}$，得到模型

$$\boldsymbol{\lambda}^{(0)}=(\boldsymbol{A}^{(0)},\boldsymbol{B}^{(0)},\boldsymbol{\pi}^{(0)})。$$

（2） **递推**对$n=1,2,\dots,$

$$a_{ij}{}^{(n+1)}=\frac{\sum_{t-1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}$$

$$\pi_i{}^{(n+1)}=\gamma_1(i)$$

当观测变量是离散型分布时

$$b_{jk}^{(n+1)}=\frac{\sum_{t=1}^{T}\gamma_t(j)\mathbf{1}_{\{o_t=v_k\}}}{\sum_{t=1}^{T}\gamma_t(j)}$$

当观测变量是连续型分布(以多元正态分布为例)时

$$\boldsymbol{\mu}_j^{(n+1)}=\frac{\sum_{t=1}^{T}\boldsymbol{o}_t\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}$$

$$\boldsymbol{\Sigma}_j^{(n+1)}=\frac{\sum_{t=1}^{T}\gamma_t(j)(\boldsymbol{o}_t-\boldsymbol{\mu}_j)(\boldsymbol{o}_t-\boldsymbol{\mu}_j)^T}{\sum_{t=1}^{T}\gamma_t(j)}$$

右端各值按观测$\boldsymbol{O}=(o_1,o_2,\dots,o_T)$和模型$\boldsymbol{\lambda}^{(n)}=(\boldsymbol{A}^{(n)},\boldsymbol{B}^{(n)},\boldsymbol{\pi}^{(n)})$计算。

（3） **终止**给定误差ε，当$Q^{(n+1)}-Q^{(n)}<\varepsilon$时停止，即$Q$函数收敛，得到模型参数$\boldsymbol{\lambda}^{(n+1)}=(\boldsymbol{A}^{(n+1)},\boldsymbol{B}^{(n+1)},\boldsymbol{\pi}^{(n+1)})$。

1.3. 预测算法

下面介绍隐马尔可夫模型预测的两种算法：近似算法与 Viterbi 算法。这两种算法比较相近，但又有本质的区别。Viterbi 算法的本质是寻找一条最优路径，然后输出这条路径上的每个状态；而近似算法则是从所有路径的概率角度出发，计算每个时刻最有可能的状态是哪个，因而无法考虑全局的最优性。更进一步的分析和举例请见 1.4.4 和 1.4.5。

1.3.1. 近似算法

近似算法的想法是，在每个时刻t选择在该时刻最有可能出现的状态i_t^*，从而得到一个状态序列$\boldsymbol{I}^*=(i_1^*,i_2^*,\dots,i_T^*)$，将它作为预测的结果。

给定隐马尔可夫模型$\boldsymbol{\lambda}$和观测序列$\boldsymbol{O}$，在时刻t处于状态q_i的概率$\gamma_t(i)$是

$$\gamma_t(i)=\frac{\alpha_t(i)\beta_t(i)}{P(\boldsymbol{O}|\boldsymbol{\lambda})}=\frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}$$

且满足$\sum_{j=1}^{N}\gamma_t(j)=1$。在每一时刻$t$最有可能的状态$i_t^*$是

$$i_t^*=\arg\max_{1\le i\le N}[\gamma_t(i)],\qquad t=1,2,\dots,T$$

从而得到状态序列$\boldsymbol{I}^*=(i_1^*,i_2^*,\dots,i_T^*)$。由于近似算法常用$\gamma_t$表示，有时我们也称这个算法叫 gamma 算法。

近似算法的优点是计算简单，其缺点是不能保证预测的状态序列整体是最有可能的状态序列。一方面，上述方法得到的状态序列中有可能存在转移概率为 0 的相邻状态(即对某些i,j，$a_{ij}=0$)，因此预测的状态可能有实际不发生；另一方面，近似算法是孤立地在每个时刻计算“期望”意义上的概率最大的状态，并没有考虑条件概率，也即没有考虑前后状态之间的关联，这样即便每个时刻都是“期望”意义上的最优状态，组成的路径也不见得是最优的(一个简单的类比是大数定律在小样本上不一定行得通)。尽管如此，近似算法仍然是有用的。

1.3.2. Viterbi 算法

Viterbi 算法实际是用动态规划解隐马尔可夫模型预测问题，即用动态规划(dynamic programming)求概率最大路径。这时一条路径对应着一个状态序列。

根据动态规划原理，最优路径具有这样的特性：如果最优路径在时刻t通过节点i_t^*，那么这一路径从结点i_t^*到终点i_T^*的部分路径，对于从i_t^*到i_T^*的所有可能的部分路

径来说，必须是最优的。因为假如不是这样，那么从i_t^*到i_T^*就有另一条更好的部分路径存在，如果把它和从i_1^*到达i_t^*的部分路径连接起来，就会形成一条比原来的路径更优的路径，这是矛盾的。依据这一原理，我们只需从时刻$t=1$开始，递推地计算在时刻t状态为i的各条部分路径的最大概率，直至得到时刻$t=T$状态为i的各条路径的最大概率。时刻$t=T$的最大概率即为最优路径的概率P^*，最优路径的终结点i_T^*也同时得到。之后，为了找出最优路径的各个节点，从终结点i_T^*开始，由后向前逐步求得结点$i_{T-1}^*,\dots,i_1^*$，得到最优路径$\boldsymbol{I}^*=(i_1^*,i_2^*,\dots,i_T^*)$。这就是 Viterbi 算法。

首先导入两个变量δ和ψ。定义在时刻t状态为i的所有单个路径$(i_1,i_2,\dots,i_t)$中概率最大值为

$$\delta_t(i)=\max_{i_1,i_2,\dots,i_{t-1}} P(i_t=i,i_{t-1},\dots,i_1,o_t,\dots,o_1|\boldsymbol{\lambda}),\qquad i=1,2,\dots,N$$

有定义可得变量δ的递推公式:

$$\begin{aligned}&\delta_{t+1}(i)\\&=\max_{i_1,i_2,\dots,i_t} P(i_{t+1}=i,i_t,\dots,i_1,o_{t+1},\dots,o_1|\lambda)\\&=\max_{1\le j\le N}\left[\delta_t(j)a_{ji}\right]b_i(o_{t+1}),\qquad i=1,2,\dots,N,\qquad t=1,2,\dots,T-1\end{aligned}$$

定义在时刻t状态为i的所有单个路径$(i_1,i_2,\dots,i_{t-1},i)$中概率最大的路径的第$t-1$个节点为

$$\psi_t(i)=\arg\max_{1\le j\le N}\left[\delta_{t-1}(j)a_{ji}\right],\qquad i=1,2,\dots,N$$

算法(Viterbi)

输入：模型$\boldsymbol{\lambda}=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$和观测$\boldsymbol{O}=(o_1,o_2,\dots,o_T)$;

输出；最优路径$\boldsymbol{I}^*=(i_1^*,i_2^*,\dots,i_T^*)$。

（1） **初始化**

$$\delta_1(i)=\pi_i b_{io_1},\qquad i=1,2,\dots,N$$

$$\psi_1(i)=0,\qquad i=1,2,\dots,N$$

（2） **递推**对$t=2,3,\dots,T$

$$\delta_t(i)=\max_{1\le j\le N}\left[\delta_{t-1}(j)a_{ji}\right]b_i(o_t),\qquad i=1,2,\dots,N$$

$$\psi_t(i)=\arg\max_{1\le j\le N}\left[\delta_{t-1}(j)a_{ji}\right],\qquad i=1,2,\dots,N$$

（3） **终止**

$$P^*=\max_{1\le j\le N}\delta_T(i)$$

$$i_T^*=\arg\max_{1\le j\le N}[\delta_T(i)]$$

（4） **最优路径回溯**对$t=T-1,T-2,\dots,1$

$$i_t^*=\psi_{t+1}(i_{t+1}^*)$$

求得最优路径$\boldsymbol{I}^*=(i_1^*,i_2^*,\dots,i_T^*)$。

1.4. 案例解析

前三节，我们已经分别介绍了 HMM 相关的三类算法，下面我们以五个例子再来让读者了解 HMM 以及这些算法的应用。

1.4.1. 盒子和球模型——原问题

假设有 3 个盒子，每个盒子里都装有红白两种颜色的球。第一个盒子里红白球数均为 5，第二个盒子里红球有 4 个白球有 6 个，第三个盒子里红球有 7 个白球有 3 个。

表 2: 各盒子和红白球数

	盒子一	盒子二	盒子三
红球数	5	4	7
白球数	5	6	3

数据来源：东北证券，《统计学习方法》(李航著)

按照下面的方法抽球，产生一个球的颜色的观测序列：开始，以概率 0.2，0.4 和 0.4 从 3 个盒子里选取 1 个盒子，从这个盒子里随机抽出 1 个球，记录其颜色后，放回。然后，从当前盒子随机转移到下一个盒子，规则是：如果当前盒子是盒子一，那么下一个盒子是盒子一、盒子二和盒子三的概率分别是 0.5，0.2 和 0.3；如果当前是盒子二，那么下一个盒子是盒子一、盒子二和盒子三的概率分别是 0.3，0.5 和 0.2；如果当前是盒子三，下一个盒子是盒子一、盒子二和盒子三的概率分别是 0.2，0.3 和 0.5；确定转移的盒子后，再从这个盒子里随机抽出 1 个球，记录其颜色，放回。如此下去，重复进行 3 次，得到一个球的颜色的观测序列：

$$\boldsymbol{O} = (\text{红，白，红})$$

在这个过程中，观察者只能观测到球的颜色的序列，观测不到球是从哪个盒子取出的，即观测不到盒子的序列。

在这个例子中有两个随机序列，一个是盒子的序列（状态序列），一个是球的颜色的观测序列（观测序列）。前者是隐蔽的，只有后者是可观测的。这是一个隐马尔可夫模型的例子，根据所给条件，可以明确状态集合、观测集合、序列长度以及模型的三要素。

盒子对应状态，状态的集合是

$$\boldsymbol{Q} = \{\text{盒子一，盒子二，盒子三}\}，N = 3$$

球的颜色对应观测的集合是

$$\boldsymbol{V} = \{\text{红，白}\}，M = 2$$

状态序列和观测序列长度$T = 3$

初始概率分布为

$$\boldsymbol{\pi} = (0.2,0.4,0.4)$$

状态转移概率分布为

$$\boldsymbol{A} = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}$$

观测概率分布为

$$\boldsymbol{B} = \begin{bmatrix} 0.5 & 0.5 \\ 0.4 & 0.6 \\ 0.7 & 0.3 \end{bmatrix}$$

1.4.2. 盒子和球模型——前向算法

考虑 1.4.1 中盒子和球模型：$\boldsymbol{\lambda} = (\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$，$\boldsymbol{Q} = \{1,2,3\}$，观测集合$\boldsymbol{V} = \{\text{红，白}\}$，

$\boldsymbol{A}=\begin{bmatrix}0.5 & 0.2 & 0.3\\ 0.3 & 0.5 & 0.2\\ 0.2 & 0.3 & 0.5\end{bmatrix}$, $\boldsymbol{B}=\begin{bmatrix}0.5 & 0.5\\ 0.4 & 0.6\\ 0.7 & 0.3\end{bmatrix}$, $\boldsymbol{\pi}=(0.2,0.4,0.4)$, $T=3$, $\boldsymbol{O}=$(红,白,红)。

用前向算法计算$P(\boldsymbol{O}|\boldsymbol{\lambda})$。

首先，计算初值:

$$\alpha_1(1)=\pi_1 b_1(o_1)=0.10$$
$$\alpha_1(2)=\pi_2 b_2(o_1)=0.16$$
$$\alpha_1(3)=\pi_3 b_3(o_1)=0.28$$

然后，递推计算:

$$\alpha_2(1)=\left[\sum_{i=1}^{3}\alpha_1(i)\,a_{i1}\right]b_1(o_2)=0.154\times 0.5=0.077$$

$$\alpha_2(2)=\left[\sum_{i=1}^{3}\alpha_1(i)\,a_{i2}\right]b_2(o_2)=0.184\times 0.6=0.1104$$

$$\alpha_2(3)=\left[\sum_{i=1}^{3}\alpha_1(i)\,a_{i3}\right]b_3(o_2)=0.202\times 0.3=0.0606$$

$$\alpha_3(1)=\left[\sum_{i=1}^{3}\alpha_2(i)\,a_{i1}\right]b_1(o_3)=0.08374\times 0.5=0.04187$$

$$\alpha_3(2)=\left[\sum_{i=1}^{3}\alpha_2(i)\,a_{i2}\right]b_2(o_3)=0.08878\times 0.4=0.035512$$

$$\alpha_3(3)=\left[\sum_{i=1}^{3}\alpha_2(i)\,a_{i3}\right]b_3(o_3)=0.07548\times 0.7=0.052836$$

最后，终止:

$$P(\boldsymbol{O}|\boldsymbol{\lambda})=\sum_{i=1}^{3}\alpha_3(i)=0.130218$$

1.4.3. 盒子和球模型——后向算法

考虑1.4.1中盒子和球模型:$\boldsymbol{\lambda}=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$, $\boldsymbol{Q}=\{1,2,3\}$,观测集合$\boldsymbol{V}=\{$红，白$\}$,

$\boldsymbol{A}=\begin{bmatrix}0.5 & 0.2 & 0.3\\ 0.3 & 0.5 & 0.2\\ 0.2 & 0.3 & 0.5\end{bmatrix}$, $\boldsymbol{B}=\begin{bmatrix}0.5 & 0.5\\ 0.4 & 0.6\\ 0.7 & 0.3\end{bmatrix}$, $\boldsymbol{\pi}=(0.2,0.4,0.4)$, $T=3$, $\boldsymbol{O}=$(红,白,红)。

用后向算法计算$P(\boldsymbol{O}|\boldsymbol{\lambda})$。

首先，终值均设为 1:

$$\beta_3(1)=\beta_3(2)=\beta_3(3)=1$$

然后，倒向递推计算:

$$\beta_2(1)=\sum_{j=1}^{3}a_{1j}b_j(o_3)\beta_3(j)=0.54\times 1=0.54$$

$$\beta_2(2)=\sum_{j=1}^{3}a_{2j}\,b_j(o_3)\beta_3(j)=0.49\times 1=0.49$$

$$\beta_2(3)=\sum_{j=1}^{3}a_{3j}\,b_j(o_3)\beta_3(j)=0.57\times 1=0.57$$

$$\beta_1(1)=\sum_{j=1}^{3}a_{1j}\,b_j(o_2)\beta_2(j)=0.2451$$

$$\beta_1(2)=\sum_{j=1}^{3}a_{2j}\,b_j(o_2)\beta_2(j)=0.2622$$

$$\beta_1(3)=\sum_{j=1}^{3}a_{3j}\,b_j(o_2)\,\beta_2(j)=0.2277$$

最后，终止:

$$P(\boldsymbol{O}|\boldsymbol{\lambda})=\sum_{i=1}^{3}\pi_i b_i(o_1)\beta_1(i)=0.130218$$

1.4.4. 盒子和球模型——近似算法

考虑1.4.1中盒子和球模型:$\boldsymbol{\lambda}=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$,$\boldsymbol{Q}=\{1,2,3\}$,观测集合$\boldsymbol{V}=\{$红，白$\}$,

$\boldsymbol{A}=\begin{bmatrix}0.5 & 0.2 & 0.3\\ 0.3 & 0.5 & 0.2\\ 0.2 & 0.3 & 0.5\end{bmatrix}$, $\boldsymbol{B}=\begin{bmatrix}0.5 & 0.5\\ 0.4 & 0.6\\ 0.7 & 0.3\end{bmatrix}$, $\boldsymbol{\pi}=(0.2,0.4,0.4)$, $T=3$, $\boldsymbol{O}=$(红,白,红)。

试用近似算法求最优状态序列$\boldsymbol{I}^*=(i_1^*,i_2^*,i_3^*)$。

首先，计算$P(\boldsymbol{O}|\boldsymbol{\lambda})$，可以利用前向-后向算法得到$P(\boldsymbol{O}|\boldsymbol{\lambda})=0.130218$。

然后，在$t=1$时，对每一状态i，$i=1,2,3$，计算$\gamma_1(i)$:

$$\gamma_1(1)=\frac{\alpha_1(1)\beta_1(1)}{P(\boldsymbol{O}|\boldsymbol{\lambda})}=\frac{0.1\times 0.2451}{0.130218}=0.1882$$

$$\gamma_1(2)=\frac{\alpha_1(2)\beta_1(2)}{P(\boldsymbol{O}|\boldsymbol{\lambda})}=\frac{0.16\times 0.2622}{0.130218}=0.3222$$

$$\gamma_1(3)=\frac{\alpha_1(3)\beta_1(3)}{P(\boldsymbol{O}|\boldsymbol{\lambda})}=\frac{0.28\times 0.2277}{0.130218}=0.4896$$

从而

$$i_1^*=\arg\max_{1\le i\le 3}[\gamma_1(i)]=3$$

同理，在$t=2$时：$\gamma_2(1)=0.3193$，$\gamma_2(2)=0.4154$，$\gamma_2(3)=0.2653$，从而

$$i_2^*=\arg\max_{1\le i\le 3}[\gamma_2(i)]=2$$

在$t=3$时：$\gamma_3(1)=0.3215$，$\gamma_3(2)=0.2727$，$\gamma_3(3)=0.4058$，从而

$$i_3^*=\arg\max_{1\le i\le 3}[\gamma_3(i)]=3$$

因此

$$\boldsymbol{I}^*=(i_1^*,i_2^*,i_3^*)=(3,2,3)$$

1.4.5. 盒子和球模型——Viterbi 算法

考虑1.4.1中盒子和球模型：$\boldsymbol{\lambda}=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$，$\boldsymbol{Q}=\{1,2,3\}$，观测集合$\boldsymbol{V}=\{$红，白$\}$，

$\boldsymbol{A}=\begin{bmatrix}0.5 & 0.2 & 0.3\\ 0.3 & 0.5 & 0.2\\ 0.2 & 0.3 & 0.5\end{bmatrix}$，$\boldsymbol{B}=\begin{bmatrix}0.5 & 0.5\\ 0.4 & 0.6\\ 0.7 & 0.3\end{bmatrix}$，$\boldsymbol{\pi}=(0.2,0.4,0.4)$，$T=3$，$\boldsymbol{O}=$(红,白,红)。

试用 Viterbi 算法求最优状态序列$\boldsymbol{I}^*=(i_1^*,i_2^*,i_3^*)$。

如图 3 所示。

首先，初始化。在$t=1$时，对每一状态i，$i=1,2,3$，求状态为i观测结果$o_1=$红的概率，记为$\delta_1(i)$，则

$$\delta_1(1)=\pi_1 b_1(o_1)=0.10,$$
$$\delta_1(2)=\pi_2 b_2(o_1)=0.16,$$
$$\delta_1(3)=\pi_3 b_3(o_1)=0.28,$$

记$\psi_1(i)=0$，$i=1,2,3$。

然后，在$t=2$时，对每个状态i，$i=1,2,3$，求在$t=1$时状态为j观测结果$o_1=$红且在$t=2$时状态为i观测结果$o_2=$白的路径的最大概率，记为$\delta_2(i)$，则

$$\delta_2(i)=\max_{1\leq j\leq 3}[\delta_1(j)a_{ji}]b_i(o_2)$$

同时，对每个状态i，$i=1,2,3$，记录概率最大路径的前一个状态j：

$$\psi_2(i)=\arg\max_{1\leq j\leq 3}[\delta_1(j)a_{ji}]$$

计算：

$$\begin{aligned}&\delta_2(1)\\&=\max_{1\leq j\leq 3}[\delta_1(j)a_{j1}]b_1(o_2)\\&=\max_j\{0.10\times 0.5,0.16\times 0.3,0.28\times 0.2\}\times 0.5\\&=0.028\end{aligned}$$

$$\psi_2(1)=3$$
$$\delta_2(2)=0.0504,\quad \psi_2(2)=3$$
$$\delta_2(3)=0.042,\quad \psi_2(3)=3$$

同样，在$t=3$时，

$$\delta_3(i)=\max_{1\leq j\leq 3}[\delta_2(j)a_{ji}]b_i(o_3)$$

$$\psi_3(i)=\arg\max_{1\leq j\leq 3}[\delta_2(j)a_{ji}]$$

$$\delta_3(1)=0.00756,\quad \psi_3(1)=2$$
$$\delta_3(2)=0.01008,\quad \psi_3(2)=2$$
$$\delta_3(3)=0.0147,\quad \psi_3(3)=3$$

接着，以P^*表示最优路径的概率，则

$$P^*=\max_{1\leq i\leq 3}\delta_3(i)=0.0147$$

最优路径的终点是i_3^*：

$$i_3^*=\arg\max_{1\leq i\leq 3}\delta_3(i)=3$$

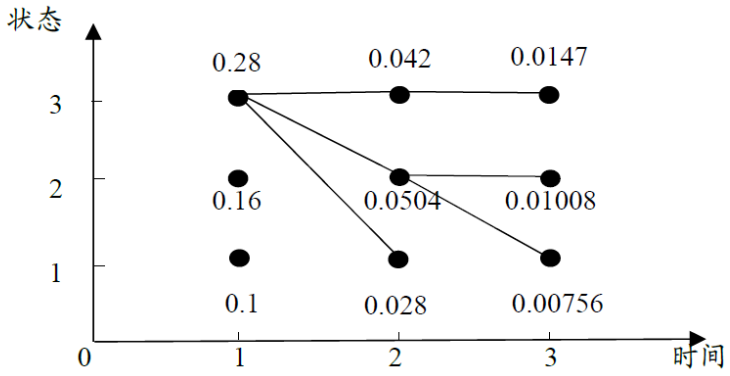
最后，最优路径的终点回溯，找到i_2^*，i_1^*：

在$t=2$时，$i_2^*=\psi_3(i_3^*)=\psi_3(3)=3$

在$t=1$时，$i_1^*=\psi_2(i_2^*)=\psi_2(3)=3$

于是求得最优路径，即最优状态序列$\boldsymbol{I}^*=(i_1^*,i_2^*,i_3^*)=(3,3,3)$。

图 3: 求最优路径



数据来源：东北证券，《统计学习方法》(李航著)

注 1: 从 1.4.4 和 1.4.5 的计算过程可以看出，gamma 算法的结果是输出每个时刻下“按照路径概率求和”意义上的概率最大的那个状态，而 Viterbi 算法是通过动态规划原理寻找最优的一条路径，这时该条路径对应着一个状态序列即为所求。因此，Viterbi 算法和 gamma 算法的结果不完全相同。从图 3 来看总共有 3 条路径(倒着看)，分别是(3,3,3)、(3,2,2)和(3,2,1)，其中路径(3,3,3)的概率最高为 0.0147；但若固定$t=2$，去计算每个状态出现的概率，则会得到状态 2 才是最有可能的状态。

注 2: 如果要评价这两个算法哪个更优，实在难以权衡。Viterbi 算法的好处是明显的——求得的隐状态序列一定是全局最优的，但缺点也是显著的，这个算法对终值特别敏感，也就是说这个算法求出来的终值的状态是不稳定的；gamma 算法的缺点是无法求得全局最优的隐状态序列，但相比之下求出的隐状态却对终值不那么敏感——这是由算法决定的，对每条路径的概率求和本身就是把权重分散的过程(可以视为 Viterbi 算法对全局最优的路径赋权是 1，非全局最优路径赋权为 0)，因此某个观测值的概率变化对整体概率求和的影响是有限的，而 Viterbi 算法则把它放大至极。因此，从静态的观点来看，Viterbi 算法无疑是最优的解法，但若动态地看，gamma 算法可以在不完全失真的前提下，保证状态的稳定性。

2. HMM 评估方法探究

在本章中，我们将围绕 HMM 的评估方法做一定的探究。首先，我们先谈一下动机，我们为什么要做评估方法的研究。在前面 1.2 的预测算法中我们已经介绍针对 HMM 有监督学习和无监督学习(Baum-Welch 算法，也就是 EM 算法)这两种预测方法。至于监督学习方面，由于我们已然知道真实的分类状态是什么，所以我们仅需要估计$\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi}$即可。而估计这些参数仅仅需要用到简单的统计频率去估计概率就可以完成。从回归或者预测的角度这样的建模未必比得上一些经典的监督学习方法如 SVM 和决策树(这里并非鼓吹一些看起来高大上的方法，而是 HMM 的假设很强，使用起来不如一些无分布的假设来的方便和准确)。然而，至于无监督学习方面，上述两种方法就派不上任何用场了。而且 EM 算法对于含有隐变量的联合分布建模有

着与生俱来的优势。但是，由于我们不知道隐变量是什么，也就无法对其进行很好的刻画或者分类，而且 EM 算法一直被诟病也是其迭代算法不能保证收敛到全局最优点。那么，对于 HMM 使用者而言，有两个问题是无法回避的。我们将这两个问题做了稍加具体的描述。

问题一：给定参数$\boldsymbol{\lambda}=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$，模拟一组隐状态序列和观测变量序列，那么在给定隐状态个数N的前提下，如何评估 HMM 模型拟合出的结果(与真实结果的差距)?

问题二：给定参数$\boldsymbol{\lambda}=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$，如何评价模型的预测能力和模型的稳定性?

这两个问题我认为是在使用 HMM 过程中必须要解决的最重要的两个问题，一方面由于 Baum-Welch 算法得到的是一个局部最优解，那么如何评估这个局部最优解和全局最优解的差距呢？这里我们提出用“最小距离”这一指标去分析；另一方面对于预测(状态)而言，按照时间窗口滚动预测的方式每天更新最新的状态组成的状态序列是一个局部最优路径，那么这个局部最优路径和全局最优路径差距有多大呢？下面我们将分拟合和预测两个部分分别构建评估方法进行探究。

2.1. HMM 拟合结果的评估标准

首先，我们定义两个评价指标：参数估计误差(Parametersestimation error)和状态估计精度(State estimationaccuracy)。

定义给定 HMM 参数$\boldsymbol{\lambda}=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$，模拟一组长度为$T$的隐状态序列$\boldsymbol{I}=(i_1,i_2,\dots,i_T)$和观测序列$\boldsymbol{O}=(o_1,o_2,\dots,o_T)$。在给定隐状态个数$N$、观测变量分布的前提下通过观测序列$\boldsymbol{O}$训练出 HMM 参数$\hat{\boldsymbol{\lambda}}=(\hat{\boldsymbol{A}},\hat{\boldsymbol{B}},\hat{\boldsymbol{\pi}})$，并估计出相应隐状态$\hat{\boldsymbol{I}}=(\hat{\imath}_1,\hat{\imath}_2,\dots,\hat{\imath}_T)$。称$\hat{\boldsymbol{\lambda}}$与$\boldsymbol{\lambda}$中各个参数之差的$L_2$范数为 HMM 的参数估计误差，即

$$\mathrm{err}_{\boldsymbol{A}}=\frac{\|\boldsymbol{A}-\hat{\boldsymbol{A}}\|_{L_2}}{\|\boldsymbol{A}\|_{L_2}}=\frac{\left(\sum_{i,j=1}^{N}(a_{ij}-\hat{a}_{ij})^2\right)^{\frac{1}{2}}}{\left(\sum_{i,j=1}^{N}a_{ij}^2\right)^{\frac{1}{2}}}$$

$$\mathrm{err}_{\boldsymbol{\pi}}=\frac{\|\boldsymbol{\pi}-\hat{\boldsymbol{\pi}}\|_{L_2}}{\|\boldsymbol{\pi}\|_{L_2}}=\frac{\left(\sum_{i=1}^{N}(\pi_i-\hat{\pi}_i)^2\right)^{\frac{1}{2}}}{\left(\sum_{i=1}^{N}\pi_i^2\right)^{\frac{1}{2}}}$$

当观测变量是离散型时，

$$\mathrm{err}_{\boldsymbol{B}}=\frac{\|\boldsymbol{B}-\hat{\boldsymbol{B}}\|_{L_2}}{\|\boldsymbol{B}\|_{L_2}}=\frac{\left(\sum_{j=1}^{N}\sum_{k=1}^{M}(b_{jk}-\hat{b}_{jk})^2\right)^{\frac{1}{2}}}{\left(\sum_{j=1}^{N}\sum_{k=1}^{M}b_{jk}^2\right)^{\frac{1}{2}}}$$

当观测变量是正态分布时，$\boldsymbol{\mu}$的估计误差和$\boldsymbol{\sigma}$的估计误差分别为

$$\mathrm{err}_{\boldsymbol{\mu}}=\frac{\|\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}\|_{L_2}}{\|\boldsymbol{\mu}\|_{L_2}}=\frac{\left(\sum_{j=1}^{N}(\mu_j-\hat{\mu}_j)^2\right)^{\frac{1}{2}}}{\left(\sum_{j=1}^{N}\mu_j^2\right)^{\frac{1}{2}}}$$

$$\mathrm{err}_{\boldsymbol{\sigma}}=\frac{\|\boldsymbol{\sigma}-\hat{\boldsymbol{\sigma}}\|_{L_2}}{\|\boldsymbol{\sigma}\|_{L_2}}=\frac{\left(\sum_{j=1}^{N}(\sigma_j-\hat{\sigma}_j)^2\right)^{\frac{1}{2}}}{\left(\sum_{j=1}^{N}\sigma_j^2\right)^{\frac{1}{2}}}$$

此时

$$\mathrm{err}_{\boldsymbol{B}}=(\mathrm{err}_{\boldsymbol{\mu}}^2+\mathrm{err}_{\boldsymbol{\sigma}}^2)^{\frac{1}{2}}$$

$\mathrm{err}_{\boldsymbol{A}}$、$\mathrm{err}_{\boldsymbol{B}}$和$\mathrm{err}_{\boldsymbol{\pi}}$分别称为$\boldsymbol{A}$、$\boldsymbol{B}$和$\boldsymbol{\pi}$的参数估计误差。称估计隐状态$\hat{\boldsymbol{I}}$和真实状态$\boldsymbol{I}$的重合度为 HMM 的状态估计精度，即

$$\mathrm{accu}=\frac{\sum_{t=1}^{T}\mathbf{1}_{\{i_t=\hat{\imath}_t\}}}{T}$$

从上面两个指标的定义来看，HMM 的参数估计误差越小越好，越小则估计的$\hat{\boldsymbol{\lambda}}$越接近真实的$\boldsymbol{\lambda}$；HMM 的状态估计精度越大越好，越大则代表估计的隐状态和真实的状态重合度越高，误差越小。但是在实际操作中，由于我们无法知道真实的$\boldsymbol{\lambda}$，也就无法直接计算这两个指标。不过如果我们能通过回归找到关于这两个指标很好的解释变量，我们便可以拿这些解释变量来估计这两个指标，比如我们将在 2.1.2 节提到“最小距离”这一指标可以作为很好的解释变量。

2.1.1. 模拟参数

定义好这两个指标后，接下来我们随机生成 1000 组参数$\boldsymbol{\lambda}$。然后在每个参数下，通过模拟观测变量序列来训练$\hat{\boldsymbol{\lambda}}$，这里假设观测变量服从一维正态分布。具体可以参照按照以下步骤进行。

（1） 给定隐状态个数N，随机生成参数$\boldsymbol{\lambda}=(\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi})$，其中$\boldsymbol{B}$的参数即$\boldsymbol{\mu}=(\mu_1,\mu_2,\dots,\mu_N)$，$\boldsymbol{\sigma}=(\sigma_1,\sigma_2,\dots,\sigma_N)$。

（2） 模拟长度为T的状态序列$\boldsymbol{I}$和观测序列$\boldsymbol{O}$：根据$\boldsymbol{A}$和$\boldsymbol{\pi}$可以模拟$\boldsymbol{I}=(i_1,i_2,\dots,i_T)$；对于每个$t=1,2,\dots,T$，由$i_t(i_t=q_i)$独立产生一个服从$\mathbf{N}(\mu_i,\sigma_i^2)$的随机变量$o_t$，从而形成观测变量$\boldsymbol{O}=(o_1,o_2,\dots,o_T)$。

（3） 根据观测序列$\boldsymbol{O}$训练模型得到$\hat{\boldsymbol{\lambda}}=(\hat{\boldsymbol{A}},\hat{\boldsymbol{B}},\hat{\boldsymbol{\pi}})$，然后利用 Viterbi 算法得到全局最优的隐状态估计$\hat{\boldsymbol{I}}=(\hat{\imath}_1,\hat{\imath}_2,\dots,\hat{\imath}_T)$。

（4） 根据定义，计算 HMM 的参数估计误差和状态估计精度。

（5） 重复做 1000 次步骤(1)~(4)，上述过程不改变N和T。

注 3：关于随机生成$\boldsymbol{\mu}$：由于平移性，所以我们可以设定一个μ_i的取值范围，并固定$\boldsymbol{\mu}$一个中心点(比如 0)，故只要考虑μ_i和μ_j之间的距离大小即可。

注 4：理论上，在计算精度的时候，默认了我们知道$\boldsymbol{I}$和$\hat{\boldsymbol{I}}$的隐状态的含义相同，即$\boldsymbol{Q}=\{q_1,q_2,\dots,q_N\}$和$\hat{\boldsymbol{Q}}=\{\hat{q}_1,\hat{q}_2,\dots,\hat{q}_N\}$是同一个集合且对应下标位置的元素相同，否则计算没有意义。而在实际操作中，我们无法保证这点，因而我们首先需要去确认和识别$\hat{\boldsymbol{I}}$的状态的含义，或者说建立$\hat{\boldsymbol{I}}$中的元素和$\boldsymbol{I}$中元素的一一对应的关系，然后把$\hat{\boldsymbol{I}}$中的元素替换成$\boldsymbol{I}$中对应的元素，最后才能比较$\boldsymbol{I}$和$\hat{\boldsymbol{I}}$的重合度。

回到我们之前说的，我们的目的是寻找一些能够较好的解释参数估计误差和状态估计精度的指标来代替这两个评价指标。首先我们想到的就是去定义一个“距离”的概念。

2.1.2. “最小距离”与参数估计误差、状态估计误差的关系

如何定义两个分布之间的距离呢？直观上，当两个正态分布密度函数重合面积越大时，我们认为这两个分布的“距离”越近，反之则认为越远。因此，用面积来度量这两个分布的距离不失为一个不错的方法。然而每次都通过计算面积去判断无疑是增加了计算量，有没有其他更简单又准确的度量方法呢？我们认为还有一种办法可以用“均值差/标准差”这一指标来定义“距离”。该思想借鉴了统计距离中的“马氏距离”，不过通常的“马氏距离”定义的是点和点或者点和分布之间的距离。下面我们给出具体的分布和分布之间的距离的定义。

定义给定两个多元正态分布$\mathbf{N}(\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1)$和$\mathbf{N}(\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2)$，称这两个分布之间的“**距离**”的平方为均值差的平方与协方差的比值，即

$$\mathrm{dist}_{12}=[\min\{(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^T\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2),(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1)\}]^{1/2}$$

同理，给定一族正态分布$\{\mathbf{N}(\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1),\mathbf{N}(\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2),\dots,\mathbf{N}(\boldsymbol{\mu}_N,\boldsymbol{\Sigma}_N)\}$，称其中任意两个分布之间的“**距离**”的最小值为这一族正态分布的“**最小距离**”，即

$$\mathrm{dist}=\min_{1\le i,j\le N,i\ne j}\mathrm{dist}_{ij}=\min_{1\le i,j\le N,i\ne j}\left[(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)\right]^{1/2}$$

特别地，当变量为一维正态分布时，

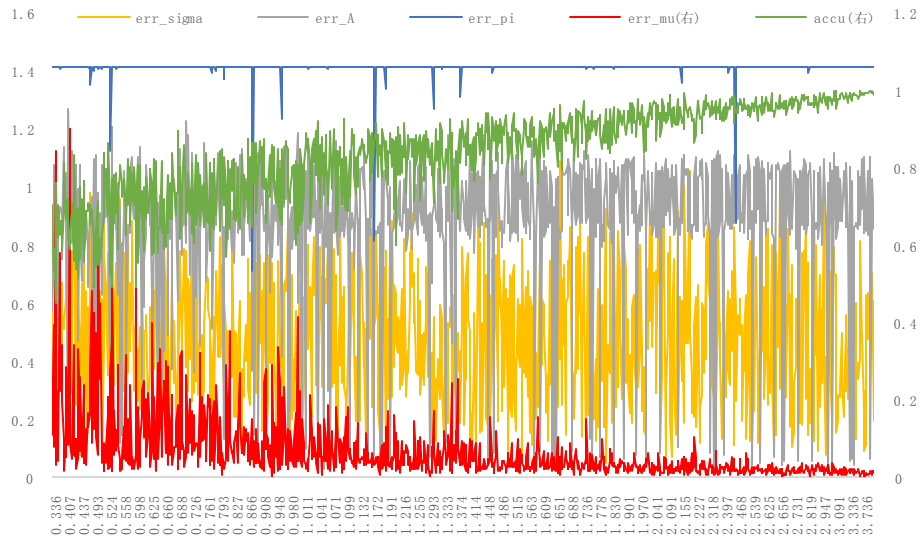
$$\mathrm{dist}=\min_{1\le i,j\le N,i\ne j}\mathrm{dist}_{ij}=\min_{1\le i,j\le N,i\ne j}\frac{|\mu_i-\mu_j|}{\sigma_i}$$

定义好了“最小距离”这一度量指标，我们将通过模拟，来探究其与两个评价指标之间的关系。

首先，我们先给定$N=3$，$T=1000$，$\boldsymbol{A}=\begin{bmatrix}0.7 & 0.15 & 0.15\\ 0.15 & 0.7 & 0.15\\ 0.15 & 0.15 & 0.7\end{bmatrix}$，$\mu_2=0$，$\boldsymbol{\pi}=(\frac{1}{3},\frac{1}{3},\frac{1}{3})$。然后，我们取$\mu_1\sim\mathbf{U}[-10,-1]$，$\mu_3\sim\mathbf{U}[1,10]$，$\sigma_1,\sigma_2,\sigma_3\sim\mathbf{U}[1,9]$。计算各参数估计误差和状态估计精度。如此做模拟 1000 次。得到如下的结果：

图 4 的横坐标是 dist 值从大到小排列，纵坐标是$\mathrm{err}_{\boldsymbol{\mu}}$、$\mathrm{err}_{\boldsymbol{A}}$、$\mathrm{err}_{\boldsymbol{B}}$、$\mathrm{err}_{\boldsymbol{\pi}}$和 accu 值。

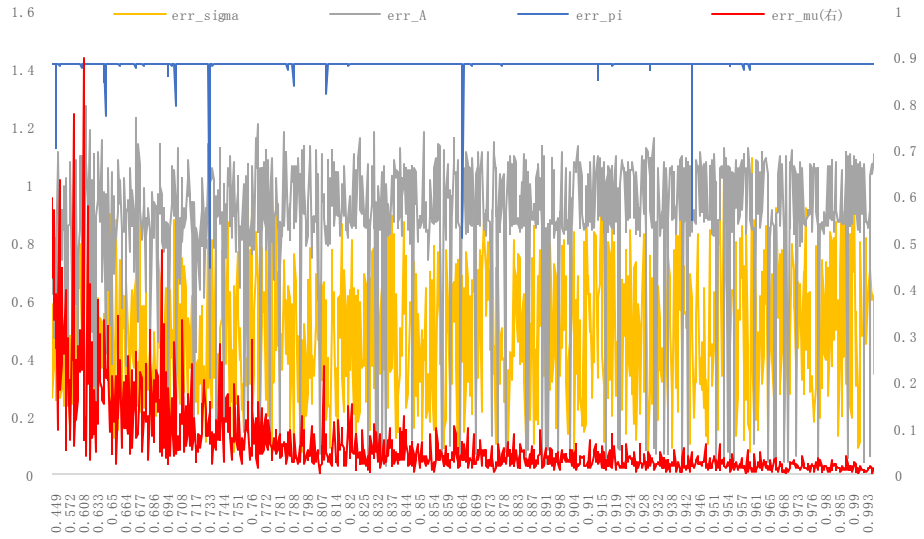
图 4：参数估计误差和 accu 随 dist 的走势



数据来源：东北证券

图 5 横坐标是 accu 值从小到大排列，纵坐标是err_{μ}、err_{A}、err_{B}和err_{π}值。

图 5：参数估计误差随 accu 的走势



数据来源：东北证券

表 3 是 dist、accu 和参数估计误差的线性相关系数，其中括号内的数字是其回归系数对应的 p-value 值，该数值代表假设“回归系数为零”的概率，而没有括号的默认其 p-value 都为 0，即拒绝原假设，接受回归系数不为零的假设。

表 3：dist、accu 和参数估计误差的线性相关性

	err_{μ}	err_{σ}	err_{A}	err_{π}	accu
dist	**-0.467**	-0.073(0.020)	0.056(0.074)	0.026(0.405)	**0.819**
accu	**-0.710**	0.106(0.001)	0.093(0.003)	0.044(0.167)	

数据来源：东北证券

表 4 是不同的 dist 分位数与相应参数估计误差和 accu 的关系，其中括号前的数字表示对应项的均值，括号内的数字表示对应项标准差。

表 4：不同 dist 分位数下参数估计误差和 accu 的均值和标准差

dist	err_{μ}	err_{σ}	err_{A}	err_{π}	Accu
0-25%:2.732(0.488)	**0.022(0.012)**	0.457(0.221)	0.894(0.260)	1.411(0.034)	0.962(0.024)
25-50%:1.701(0.203)	**0.038(0.025)**	0.480(0.234)	0.903(0.240)	1.414(0.001)	0.894(0.044)
50-75%:1.135(0.136)	**0.070(0.060)**	0.480(0.203)	0.860(0.257)	1.410(0.041)	0.812(0.067)
75-100%:0.633(0.150)	**0.142(0.137)**	0.497(0.206)	0.854(0.215)	1.409(0.048)	0.704(0.086)

数据来源：东北证券

结合表 3、表 4、图 4 和图 5，我们总结如下：

（1）dist、accu 与err_{μ}呈现一定的负线性相关性：即“最小距离”或状态估计精度越大，均值估计误差越小。dist 与err_{μ}的相关系数为-0.467；accu 与err_{μ}的相关系数为-0.710。

（2）dist 与 accu 呈现较强的正线性相关性：即“最小距离”越大，状态估计精度越大。dist 与 accu 的相关系数为 0.819。

（3）dist 与err_{σ}、err_{A}、err_{π}的相关性较低，分别为-0.073、0.056 和 0.026；accu 与err_{σ}、err_{A}、err_{π}的相关性较低，分别为 0.106、0.093 和 0.044。

（4）在不同的 dist 分位数下，err_{σ}、err_{A}、err_{π}的均值和标准差都比较稳定，可以判断是具有同分布噪声产生的误差。

（5）虽然 dist 指标仍然是一个未知变量，但我们可以用$\hat{\lambda}$来计算$\widehat{dist}$，而 dist 和$\widehat{dist}$的线性相关性达到 0.89。

从上述分析中，我们得到一个有用的结论：**虽然我们用 Baum-Welch 算法估计的参数$\hat{\lambda}=(\hat{A},\hat{B},\hat{\pi})$并不一定是全局最优，但是我们能用 dist 指标来评价$\hat{\lambda}$拟合结果的好坏。**如果 dist 越大，一方面 accu 这个指标越大，说明预测真实状态的准确率越高；另一方面，$\hat{\lambda}$的误差估计中关于均值的误差估计是随着 dist 增大显著地衰减，这说明估计的参数收敛到真实的均值。err_{σ}、err_{A}和err_{π}并没有随着 dist 增大而收敛，表现出在某个均值附近震荡，且具有同方差特性。这一点基本可以判定是噪声产生的。这也说明了我们很难准确地估计真实的$\boldsymbol{\sigma}$、$\boldsymbol{A}$和$\boldsymbol{\pi}$，因此我们只能尽可能去使得均值收敛，也就是我们可以最大化 dist 来选择$\hat{\lambda}$。

以上分析的结果是基于固定$\boldsymbol{A}$和$\boldsymbol{\pi}$，对均值和方差的分析得到的结论。我们同样也做了固定$\boldsymbol{A}$和$\boldsymbol{B}$的情况下，研究$\boldsymbol{\pi}$的影响以及固定$\boldsymbol{B}$和$\boldsymbol{\pi}$的情况下，研究$\boldsymbol{A}$的影响

2.1.3. 初始分布$\boldsymbol{\pi}$的影响

下面我们将研究初始分布$\boldsymbol{\pi}$对参数估计误差和状态估计误差的影响。**结论是初始概率分布对参数误差估计的相关性非常小，可以忽略。**

给定$N=3$，$T=1000$，$\boldsymbol{A}=\begin{bmatrix}0.7 & 0.15 & 0.15\\ 0.15 & 0.7 & 0.15\\ 0.15 & 0.15 & 0.7\end{bmatrix}$，$\mu_2=0$，$\mu_3=-\mu_1$，$\sigma_1,\sigma_2,\sigma_3=1$。按照$\mu_1=1,2,3,4,5$，(即$dist=1,2,3,4,5$)分别模拟 100 次$\boldsymbol{\pi}$。得到如下的结果：

表 5：accu 和参数估计误差的线性相关性(固定A和B)

	err_μ	err_σ	err_A	err_π
accu (dist =1)	-0.445	-0.120	-0.284	0.074
accu (dist =2)	-0.471	-0.111	0.008	-0.013
accu (dist =3)	-0.039	-0.184	0.179	0.004
accu (dist =4)	0.003	-0.103	-0.071	-0.023
accu (dist =5)	-0.011	-0.340	-0.180	-0.074

数据来源：东北证券

表 6：不同 dist 值下各参数估计误差和 accu 的均值和标准差(固定A和B)

	err_μ	err_σ	err_A	err_π	accu
dist =1	0.247(0.123)	0.198(0.105)	0.754(0.269)	1.228(0.410)	0.600(0.037)
dist =2	0.062(0.037)	0.116(0.053)	0.863(0.279)	1.177(0.470)	0.835(0.020)
dist =3	0.027(0.011)	0.088(0.036)	0.939(0.182)	1.268(0.421)	0.940(0.008)
dist =4	0.016(0.006)	0.074(0.033)	0.941(0.184)	1.179(0.496)	0.980(0.004)
dist =5	0.012(0.005)	0.067(0.034)	0.907(0.258)	1.202(0.495)	0.995(0.002)

数据来源：东北证券

从表 5 和表 6，我们能得到以下结论：不同 dist 值下，accu 和各参数估计误差的相关性并不一样。具体来说:

（1）从表 5 来看，在 dist 较小时，err_μ与 accu 呈现一定的负线性相关性，而在 dist $>$ 3后则几乎没有线性关系，这是由于此时err_μ几乎收敛，其误差在非常小的范围内变动。这点也能从表 6 之中得到印证。

（2）而err_σ则呈现不一样的关系。从表 5 看，dist $\leq$ 4时，accu 与err_σ呈现很弱的负线性相关性，而当dist $=$ 5时负线性相关性显著增大；从表 6 上看，dist 增大的过程中，err_σ的均值和标准差都在减小。综上说明dist $=$ 5时 accu 增大的同时err_σ减小更为显著，即对$\boldsymbol{\sigma}$的估计也越精确。

（3）err_A则没有诸如上述的“单调性”，不仅相关性在变化而且相关系数的绝对值也在变化。从表 5 来看，在不同 dist 下 accu 与err_A并没有一致的相关性，相关系数的绝对值也不大，说明更偏向于噪声产生的震荡。从表 6 来看，err_A的均值随 dist 增大呈现增大的趋势，说明$\boldsymbol{A}$的估计偏差在增大，这点似乎与我们期望的相反。标准差也体现了震荡的趋势。我们认为$\boldsymbol{A}$的估计误差仍然难以有明显的下降，可能是由于$\boldsymbol{A}$的参数较多(有 9 个)，此时每个参数的噪声叠加起来对结果产生了一个比较大的误差。这个解释和误差与 accu 的没有明确的线性相关性这点也可以相互印证。

（4）err_π比err_A体现出更多的噪声性，一方面从表 5 上看，不同 dist 下 accu 与err_π几乎没有线性相关性，另一方面从表 6 来看，不同 dist 下err_π几乎是同分布。这两者都说明了err_π是一个标准差比较大的噪声。

从上面的分析中，我们得到了和 2.1.2 类似的结论，这说明了初始分布$\boldsymbol{\pi}$对参数估计误差和状态估计误差几乎没有影响，或者说知道初值是均匀分布或随机分布对估计参数并没有任何好处。这也许是因为初值的影响只是改变了前几个状态的和观测变量，但由于序列长度足够就会收敛到稳定的分布。之所以没有任何好处讲的更具体一些，是因为均与分布的情况下，err_π为代表的参数估计误差的均值更大一些，但标准差较小；随机分布的情况下，err_π为代表的参数估计误差的均值小一些，但

标准差扩大了。这就说明了，真实的初始分布是什么其实并不重要，因为无论是均匀分布或者随机分布的情况下都难以控制好估计误差，不同的情况下的误差就是不同均值方差的正态分布罢了。

2.1.4. 转移概率矩阵A的影响

下面我们将研究对转移概率矩阵$\boldsymbol{A}$对参数估计误差和状态估计误差的影响。**结论是不同A下，err_μ和err_σ与 accu 呈现较强的负相关，err_π和err_A但与 accu 相关性很低，基本可以认为是噪声引起的。**

给定$N=3$，$T=1000$，$\mu_2=0$，$\mu_3=-\mu_1$，$\sigma_1,\sigma_2,\sigma_3=1$，$\boldsymbol{\pi}=(\frac{1}{3},\frac{1}{3},\frac{1}{3})$。按照$\mu_1=1,2,3,4,5$，(即$\mathrm{dist}=1,2,3,4,5$)分别模拟 100 次$\boldsymbol{A}$。得到如下的结果：

表 7：accu 和参数估计误差的线性相关性(固定B和π)

	err_μ	err_σ	err_A	err_π
accu (dist =1)	**-0.786**	-0.258	0.107	0.058
accu (dist =2)	**-0.886**	**-0.454**	0.094	-0.035
accu (dist =3)	**-0.864**	**-0.532**	-0.013	0.012
accu (dist =4)	**-0.912**	**-0.557**	-0.001	0.114
accu (dist =5)	**-0.979**	**-0.786**	-0.030	0.316

数据来源：东北证券

表 8：不同 dist 值下各参数估计误差和 accu 的均值和标准差(固定B和π)

	err_μ	err_σ	err_A	err_π	accu
dist =1	0.432(0.245)	0.264(0.127)	0.757(0.227)	1.404(0.070)	0.555(0.121)
dist =2	0.274(0.257)	0.267(0.159)	0.776(0.276)	1.413(0.004)	0.765(0.133)
dist =3	0.159(0.247)	0.260(0.256)	0.765(0.337)	1.409(0.053)	0.891(0.145)
dist =4	0.074(0.174)	0.182(0.285)	0.771(0.350)	1.413(0.009)	0.953(0.115)
dist =5	0.056(0.175)	0.180(0.409)	0.743(0.348)	1.414(0.003)	0.973(0.096)

数据来源：东北证券

从表 7 和表 8，我们能得到以下结论：err_μ和err_σ与 accu 的负线性相关性很高，且 dist 取值越大二者与 accu 的负线性相关性就越强；err_A和err_π与 accu 的相关性并无明显的规律。具体来说：

（1）从表 7 来看，无论 dist 取值如何，err_μ与 accu 的负线性相关性很高(最低也有-0.786)，而且随 dist 增大相关系数的绝对值越大，负相关性越强，最高达到-0.979。另一方面从表 8 来看，当$\mathrm{dist}>2$以后均值明显的小于标准差，这意味着，err_μ有明显的“右偏厚尾”的特性，大部分的值其实在均值左侧。这也说明了err_μ收敛的速度很快。这是非常好的结果，也就是说无论$\boldsymbol{A}$的取值如何，当 dist 取值较大的时候，都可以比较准确的估计出$\boldsymbol{\mu}$的取值，而且此时估计状态的准确率也非常高。

（2）err_σ也呈现相似的关系。从表 7 看，accu 与err_σ负线性相关性随着 dist 增大而显著增强，而当$\mathrm{dist}=5$时负线性相关性最大为-0.786；从表 8 上看，dist 增大的过程中，err_σ的均值呈减小趋势和但标准差在显著增大。这种情况和err_μ也非常相似，都是“右偏厚尾”的特性。从而err_σ均值也有明显的收敛特性。综上说明 dist 取值较大的时候，可以比较准确的估计$\boldsymbol{\sigma}$的取值，而且此

时估计状态的准确率也较高。

（3）$\text{err}_{\boldsymbol{A}}$则没有诸如上述的“单调性”，不仅相关性在变化而且相关系数的绝对值也在变化。从表 7 来看，在不同 dist 下 accu 与$\text{err}_{\boldsymbol{A}}$并没有一致的相关性，相关系数的绝对值很小，说明基本都是噪声产生的震荡。从表 8 来看，$\text{err}_{\boldsymbol{A}}$的均值随 dist 增大过程中保持稳定，但标准差略有扩大，说明$\boldsymbol{A}$的估计偏差在略微增大。这点虽然与我们期望相反，但从另一方面体现了$\boldsymbol{A}$的估计误差难以大幅下降的特点以及相对稳定的特性。

（4）$\text{err}_{\boldsymbol{\pi}}$与$\text{err}_{\boldsymbol{A}}$一样体现出很高的噪声性。一方面从表 7 上看，dist 较小时 accu 与$\text{err}_{\boldsymbol{\pi}}$几乎没有线性相关性，dist 较大时 accu 与$\text{err}_{\boldsymbol{\pi}}$呈现一定的正线性相关性另。一方面从表 8 来看，不同 dist 下$\text{err}_{\boldsymbol{\pi}}$的均值稳定，但标准差大小不同，但总体变化幅度不大。因此，在初始分布是均匀分布的情况下，虽然不能收敛，$\text{err}_{\boldsymbol{\pi}}$表现较为稳定。

2.1.5. 参数误差分析小结

通过前面的分析，2.1.2~2.1.4 都有相似的结论，即 dist 值很好地刻画了分布的距离，同时对$\boldsymbol{\mu}$和$\boldsymbol{\sigma}$的参数估计误差有很强的负相关性。但通过对比 2.1.3 和 2.1.4 我们得到：对于不同的$\boldsymbol{A}$而言，其$\text{err}_{\boldsymbol{\mu}}$和$\text{err}_{\boldsymbol{\sigma}}$整体上随着 dist 值的增大显著地衰减；对于同一个$\boldsymbol{A}$而言，dist 值从小变大对$\text{err}_{\boldsymbol{\mu}}$的衰减一开始比较显著，后续增大衰减的显著性明显降低(因为已经收敛)，而 dist 值从小变大对$\text{err}_{\boldsymbol{\sigma}}$的衰减一开始比较不显著，后续增大衰减显著。而无论哪种情况，$\text{err}_{\boldsymbol{\pi}}$与$\text{err}_{\boldsymbol{A}}$均表现明显的噪声性，难以收敛。

2.2. HMM 预测结果的评估标准

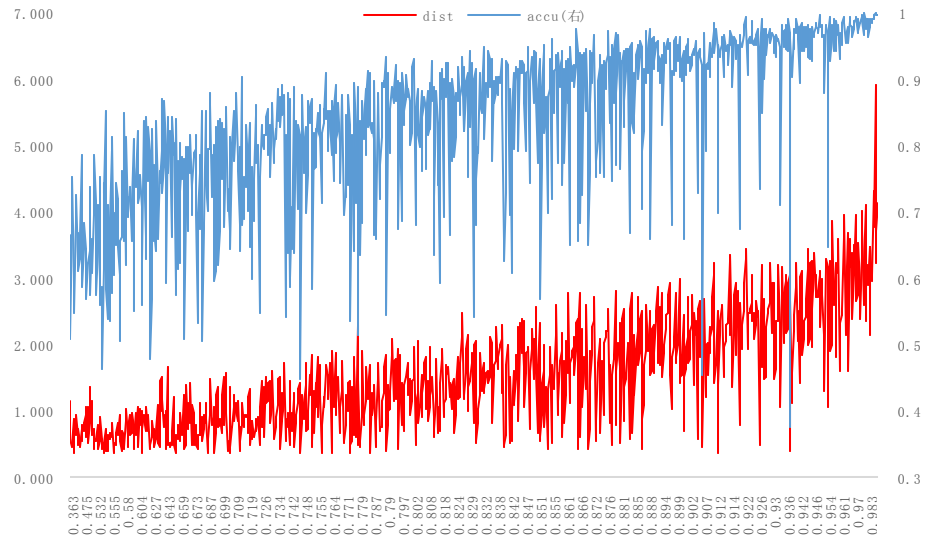
上一节中，我们介绍了如何评价 Baum-Welch 算法估计 HMM 的参数与真实参数之间的关系。在本节中，我们将再添加一个评价指标——状态估计稳定性(State estimation robustness)，用来度量 Viterbi 算法的全局最优性和实际使用时候的局部最优性的差距。

定义给定一组长度为T的观测序列$\boldsymbol{O}=(o_1,o_2,\dots,o_T)$，在给定隐状态个数$N$、观测变量分布的前提下通过观测序列$\boldsymbol{O}$训练出 HMM 参数$\hat{\boldsymbol{\lambda}}=(\hat{\boldsymbol{A}},\hat{\boldsymbol{B}},\hat{\boldsymbol{\pi}})$。分别用两种途径预测隐状态。第一种：把$\boldsymbol{O}$一次性代入 Viterbi 算法中，预测出全局最优的隐状态序列$\boldsymbol{I}^*=(i_1^*,i_2^*,\dots,i_T^*)$。第二种：按照时间窗口滚动输入预测，$t=1,2,\dots,T$，依次把$\boldsymbol{O}_t=(o_1,o_2,\dots,o_t)$代入 Viterbi 算法并预测$t$时刻状态，从而形成一个局部最优序列$\boldsymbol{I}=(i_1,i_2,\dots,i_T)$。称$\boldsymbol{I}$和$\boldsymbol{I}^*$的重合度为 HMM 的状态估计稳定性，即

$$\text{robu}=\frac{\sum_{t=1}^{T}\mathbf{1}_{\{i_t=i_t^*\}}}{T}$$

加入这个指标后，我们便有了预测评估标准。同理，我们也可以研究 robu 与 dist 和 accu 之间的关系。为了便于比较，我们模拟的参数和 2.1.2 中的参数是一致的。

图 6：robu 与 dist 和 accu 的走势



数据来源：东北证券

表 9：robu 与 dist 和 accu 的线性相关性

	dist	accu
robu	0.714	0.731

数据来源：东北证券

结合表 9 和图 6，我们得到如下结论：dist 和 accu 与 robu 之间具有显著的正线性相关性，即“最小距离”越大，HMM 状态估计稳定性越高；状态估计精度越大，状态估计稳定性越高。

2.3. 本章小结

我们再回顾一下这一章里我们研究的动机和结果。

第一，对于一个 HMM 模型而言，其优势在于可以对含有隐变量的数据通过 Baum-Welch 算法(也就是 EM 算法)进行无监督学习。而由于 EM 的迭代算法无法收敛到全局最优解，那么我们自然关心的这个局部最优解和全局最优解的差距有多大？即我们如何来估计一个模型的解$\hat{\boldsymbol{\lambda}} = (\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}, \hat{\boldsymbol{\pi}})$的误差？为此，我们构建了一个指标——“最小距离”dist——一族分布中任意两个分布之间距离的最小值。通过在 2.1.2~2.1.4 中的分析，我们知道 dist 值可以有效的评估与均值$\hat{\boldsymbol{\mu}}$和标准差$\hat{\boldsymbol{\sigma}}$ (即$\hat{\boldsymbol{B}}$)的估计误差，即 dist 取值越大，$\hat{\boldsymbol{\mu}}$和$\hat{\boldsymbol{\sigma}}$能快速收敛到$\boldsymbol{\mu}$和$\boldsymbol{\sigma}$，但对转移矩阵$\hat{\boldsymbol{A}}$和$\hat{\boldsymbol{\pi}}$的估计误差却束手无策。

第二，从时间序列的角度上看，Viterbi 算法推测的隐状态实际上是全局最优的隐状态，一方面这个状态是不可回测的，另一方面如果按照时间推移逐个输出每个时刻的隐状态从而组成一个局部最优的隐状态序列，那么这个局部最优的隐状态序列和全局最优的隐状态序列的差距有多大？为此，我们构建的评价指标为状态估计稳定性 robu——即是局部最优序列和全局最优序列在每个位置上的重合度。我们发现，robu 这个指标和状态估计精度 accu 和“最小距离”dist 均有较高的正相关性。其实，拟合结果的评估和预测结果的评估完全是两个维度的问题，即我们完全可以不用 dist 指标来估计 robu，只要独立计算每个 HMM 模型的 robu 值，并且在众多的

HMM 模型之中挑一个 robu 值较高的即可。但结合 dist 值之后，dist 便把拟合和预测这两个问题给关联起来了，其实二者都可以和分布的距离扯上关系，而且通过这样的分布距离的辅助筛选，逻辑上更加具有说服力，这样的模型才更具有稳定性。

第三，上一篇报告的择时模型出炉之后，许多读者的问题多是围绕 HMM 参数 N 展开的，因为我们把 N 当作一个拟合的参数，通过择时策略最大化收益回撤比的方式去求 N。但之后我们进一步的研究，N 应该有一个合理的范围(否则 N 太大可能会造成过度拟合)，而这个范围其实也与 dist 值息息相关。试想：把一堆数据按照 Kmeans 聚类(最小化点到中心距离的一种无监督学习的方法)方式分 N 堆，那么每一堆数据的中心点之间距离的最小值一定不会随着 N 的增大而增大的。这个性质可以用均值不等式证明。其含义是随着分堆数 N 的增大，这些数据会被分的越来越细，越来越密，以至于相邻两堆数据的临界点容易判错，甚至这两类数据其实没有显著的区别。这就是过度分类，此时其 dist 值一定会变小。也就是说，我们可以通过设定 dist 值下界作为模型的一个过滤器。甚至，我们可以通过 dist 指标来甄别观测变量的好坏。

3. 基于 dist 指标的反思和初试

在上一篇报告之中，我们用了 return 序列作为单一观测变量。现在我们结合第 2 章中对 HMM 评估方法来计算这个基于日收益率序列的 HMM 的 dist 值。

从表 10 我们可以立马算出该模型的 dist 值几乎为 0，因为状态 5 和状态 11 的均值几乎相同。也就是说，这个模型的拟合误差可以非常大！同样地，我们不用计算也可以知道模型的稳健性不高。这引起了我们对 return 这个观测指标及对选择隐状态个数 N 的一个反思：一、return 是不是一个好的 HMM 观测指标？二、N 的选择是否跟观测变量有关，但是否有客观评价的标准？

表 10：基于日收益率的 HMM 参数：均值和方差

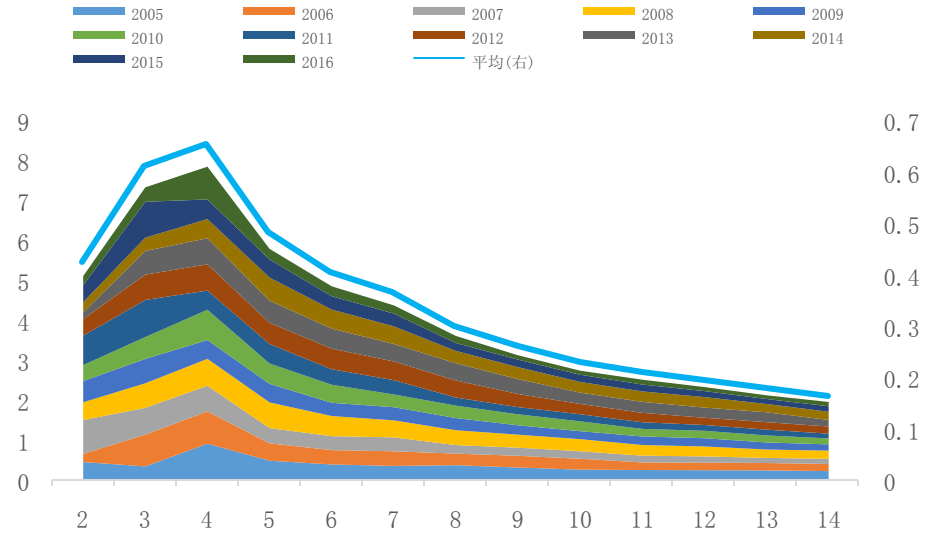
状态	均值(%)	方差(%)
0	-0.04	0.0011
1	2.44	0.0156
2	-3.32	0.0105
3	1.39	0.0273
4	-1.41	0.0052
5	0.33	0.0021
6	-1.13	0.1021
7	1.32	0.0055
8	8.59	0.0075
9	-0.13	0.0040
10	-1.37	0.0284
11	0.33	0.0022
12	-0.98	0.0028
13	0.72	0.0140

数据来源：东北证券，Wind

3.1. 反思：关于 return 指标分类显著性探究

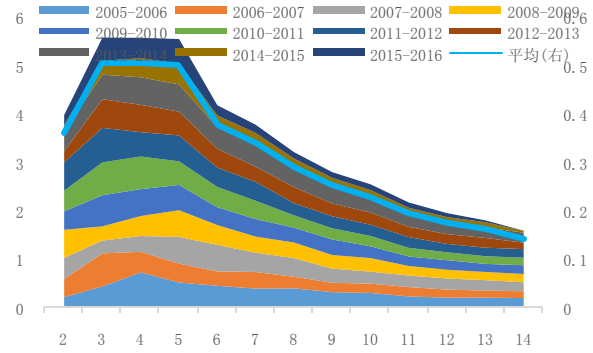
为了研究 return 这个指标适不适合做观测指标，我们以上证综指为例，选取的时间区间是 2005 年至 2016 年，训练长度为分别为一年至五年不等。在固定训练长度时，不同的年份，不同的N的取值下，一方面研究各个 HMM 模型的 dist 值的大小变化呈现何种规律；另一方面，我们也将看到选取不同长度的样本，模型的 dist 值的大小如何变化。

图 7：一年训练长度的上证综指日收益率 HMM 的 dist 随N的变化堆叠图



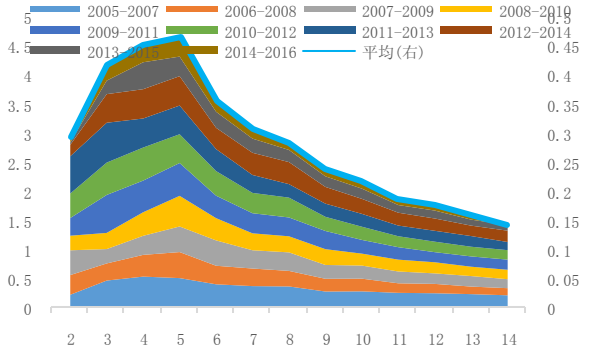
数据来源：东北证券，Wind

图 8：两年训练长度的上证综指日收益率 HMM 的 dist 随N的变化堆叠图



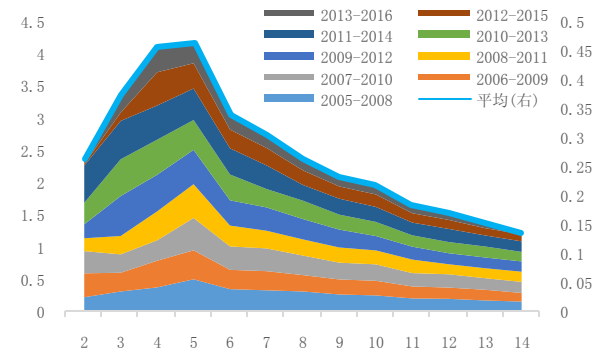
数据来源：东北证券，Wind

图 9：三年训练长度的上证综指日收益率 HMM 的 dist 随N的变化堆叠图



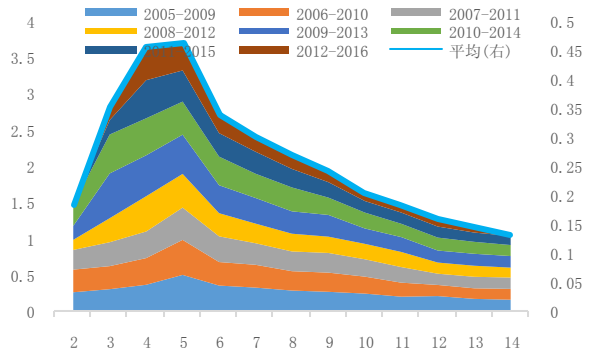
数据来源：东北证券，Wind

图 10：四年训练长度的上证综指日收益率 HMM 的 dist 随N的变化堆叠图



数据来源：东北证券，Wind

图 11：五年训练长度的上证综指日收益率 HMM 的 dist 随N的变化堆叠图



数据来源：东北证券，Wind

从图 7~图 11 可以看出，一方面固定训练长度的情况下 dist 值随着N的增大先快速增大后缓慢减小，综合个年份的情况来看，最优的N的取值为 3~5；另一方面随着训练长度的增大，曲线在最优值在$N=4$或 5的平均值也跟着下降但最终达到稳定，平均值曲线收敛。如一年训练长度时，综合各年的 dist 平均值$N=4$时最高为 0.653，两年训练长度时，综合各年的 dist 平均值$N=3$或 4时最高为 0.505，三年训练长度时，综合各年的 dist 平均值$N=5$时最高为 0.467，四年训练长度时，综合各年的 dist 平均值$N=5$时最高为 0.462，五年训练长度时，综合各年的 dist 平均值$N=5$时最高为 0.462。也就是说，当训练长度较长的时，模型的 dist 值较为稳定。

因此，我们得出 return 这个指标可能不太适合应用于N比较大的情形，而且 dist 值对N有较为明显的依赖性，因为N越大，dist 越小，分类越不显著。下面我们将会看到，选取不同的观测指标，我们能做到对N的低依赖性，dist 值也比使用 return 的情况更大，分类更加显著。另外，我们也将在后面看到，return 之所以不太适用的原因还跟它的分布是尖峰厚尾而非正态分布有关。

3.2. 初试：一个新指标的分类显著性探究

我们认为日收益率之所以不适合作为观测变量的原因是其变化太快，并且没有规律，呈现的是如噪声一样的平稳序列，如图 12；另一方面，其分布呈现“尖峰厚尾”，如图 14，那么在N增大的过程中其势必因为无法显著区分各个状态。因此，是否应该找一个相对“慢”一些的指标呢？下面我们尝试了一个新的指标：

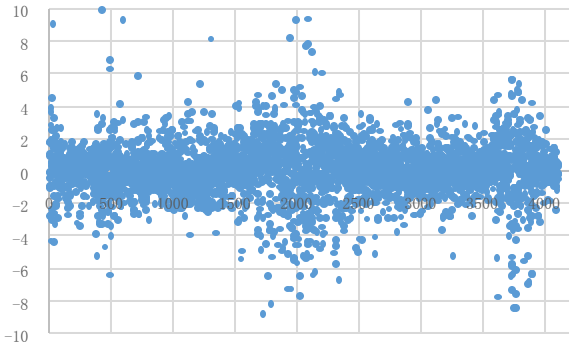
$$\mathrm{MA}_{\mathrm{short\ /long}}(t)=\frac{\mathrm{MA}_{\mathrm{short}}(t)}{\mathrm{MA}_{\mathrm{long}}(t)}-1$$

其中$\mathrm{MA}_{\mathrm{short}}$表示短期 MA，$\mathrm{MA}_{\mathrm{long}}$表示长期 MA。

这个指标表现为短期 MA 相对长期 MA 的增长率，是一个变化速率指标，该指标和 ROC 相似。图 13 显示的是$\mathrm{MA}_{20/120}$这一指标在 2000 年至 2016 年以来的散点图。通过对比，我们明显看出日收益率变化“快”，而$\mathrm{MA}_{20/120}$变化“慢”。从图 14 和图 15 的比较，虽然二者都通过了 ADF 检验，但我们也可以看出一些端倪：$\mathrm{MA}_{20/120}$更“宽”一些，更分散一些；return 太“窄”，更集中一些。所以，我们完全有理由相信$\mathrm{MA}_{20/120}$这个指标在N较大的时候，仍然具有比较好的分类效果。

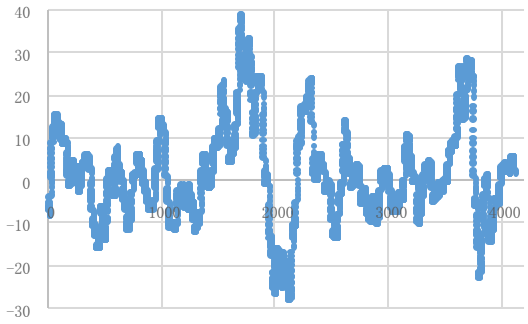
注 5： 图 12 和图 13 的纵坐标、图 14 和图 15 的横坐标轴均是以%为单位。

图 12：2000-2016 年上证综指日收益率散点图



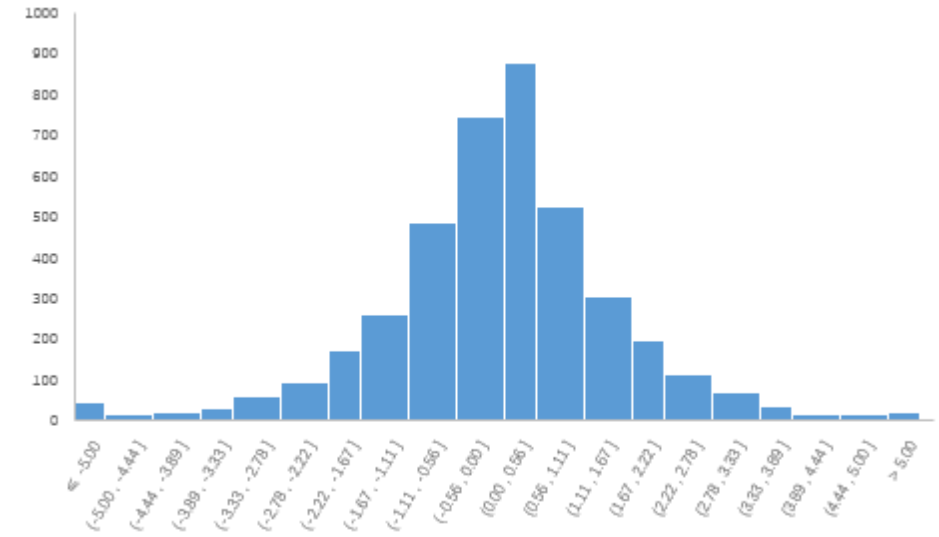
数据来源：东北证券，Wind

图 13：2000-2016 年上证综指$MA_{20/120}$散点图



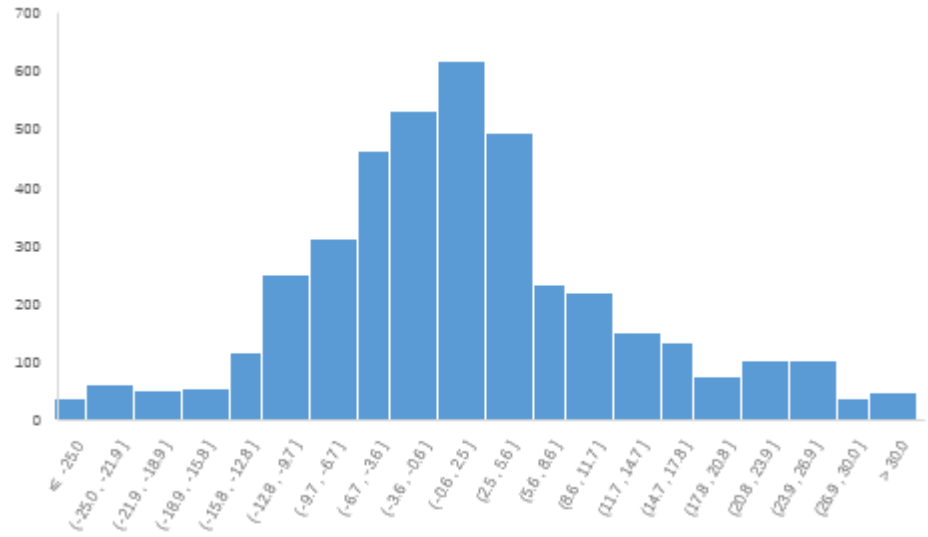
数据来源：东北证券，Wind

图 14：2000-2016 年上证综指日收益率直方图



数据来源：东北证券，Wind

图 15：2000-2016 年上证综指$MA_{20/120}$直方图

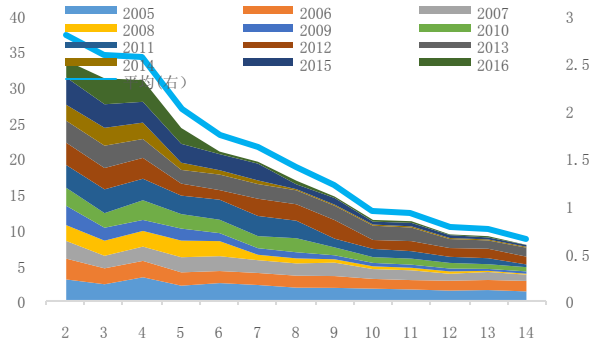


数据来源：东北证券，Wind

下面我将看到以$MA_{20/120}$这一指标为观测变量的 HMM 模型的 dist 值与N之间的关系。从图 16~图 19 可以看出在固定训练长度的前提下，dist 值随N的增大呈现递减趋势，但训练长度越长，衰减幅度减小，尤其是当训练长度达到七年的时候，$N>3$后 dist 值变化非常小。也就是说 dist 值与N的相关性非常低，并且此时 dist 值几乎都在 1 以上，可见此时的分类显著性非常高。

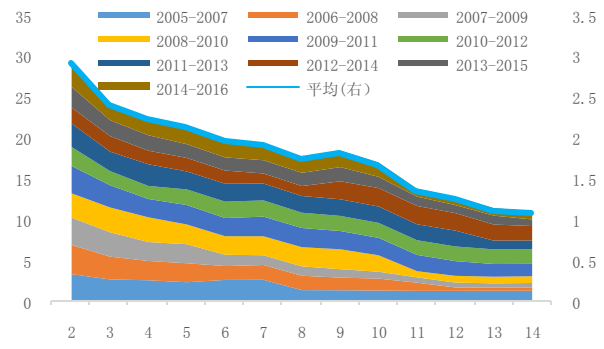
通过对比图 16~图 19 和图 7~图 11，我们能够直观的发现不同的指标对N的敏感程度是不一样的，而且对训练长度的敏感程度也不一样。对变化"快"的 return 而言，训练长度越长，dist 随N的变化规律基本上保持稳定，只会在$N=3$~5局部时候分类比较显著；对变化"慢"的$MA_{20/120}$而言，训练长度越长，dist 越是对所有N都显著。从这层意义上而言，$MA_{20/120}$的分类显著性上比 return 要好，而且更稳定。

图 16: 一年训练长度的上证综指$MA_{20/120}$HMM 的 dist 随N的变化堆叠图



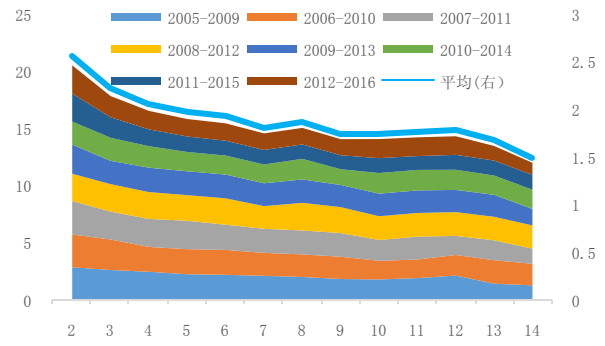
数据来源：东北证券，Wind

图 17：三年训练长度的上证综指$MA_{20/120}$HMM 的 dist 随N的变化堆叠图



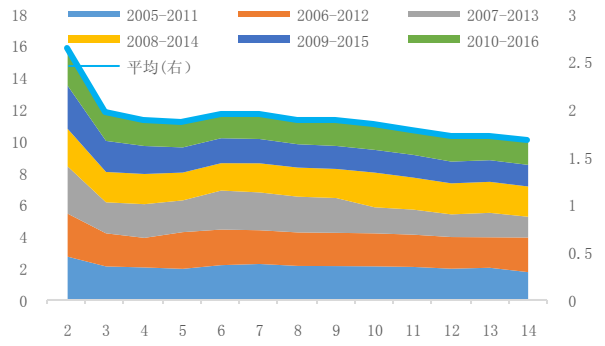
数据来源：东北证券，Wind

图 18: 五年训练长度的上证综指$MA_{20/120}$HMM 的 dist 随N的变化堆叠图



数据来源：东北证券，Wind

图 19: 七年训练长度的上证综指$MA_{20/120}$HMM 的 dist 随N的变化堆叠图



数据来源：东北证券，Wind

3.3. 本章小结

在本章中，我们试图借助第 2 章研究得到的评估指标 dist 值对 return 和$MA_{20/120}$做一个分类显著性的研究。我们得到的结论是：

第一，从时间序列的散点图来看，return 指标变化无规律，$MA_{20/120}$指标变化具有趋势性；return 是平稳序列，$MA_{20/120}$短期来看不是平稳序列，长期来看是平稳序列。

第二，从二者的分布来看，return 指标“尖峰厚尾”，$MA_{20/120}$指标更“矮平”一些。虽然二者都无法通过正态性检验，但$MA_{20/120}$比 return 更像正态一些。

第三，从对N的敏感性而言，无论观测样本长度如何变化，return 指标均并不适合N比较大的情形，几乎只对$N=3\sim5$分类比较显著，对N敏感。而随着观测样本长度的增大，对N的敏感性降低，且适合N比较大的情形。

第四，对第三个结论的解释是：由于 return 的样本过于集中在均值附近，导致分类过多时，分类的显著性肯定下降，所以分类数稳定在$3\sim5$类合适，样本数量的提升只会加剧集中程度；而$MA_{20/120}$的样本较为分散，当样本数量不够多时，势必分布有偏，导致分类不显著，但随着样本数量的增加，样本分布更加趋近于整体分布，使得分类的显著性提高。

分析师简介:

陈亚龙：策略分析师，复旦大学世界经济硕士，2014年加入东北证券研究所。
肖承志：研究助理，同济大学应用数学硕士，2016年加入东北证券研究所。

重要声明

本报告由东北证券股份有限公司(以下称“本公司”)制作并仅向本公司客户发布，本公司不会因任何机构或个人接收到本报告而视其为本公司的当然客户。

本公司具有中国证监会核准的证券投资咨询业务资格。

本报告中的信息均来源于公开资料，本公司对这些信息的准确性和完整性不作任何保证。报告中的内容和意见仅反映本公司于发布本报告当日的判断，不保证所包含的内容和意见不发生变化。

本报告仅供参考，并不构成对所述证券买卖的出价或征价。在任何情况下，本报告中的信息或所表述的意见均不构成对任何人的证券买卖建议。本公司及其雇员不承诺投资者一定获利，不与投资者分享投资收益，在任何情况下，我公司及其雇员对任何人使用本报告及其内容所引发的任何直接或间接损失概不负责。

本公司或其关联机构可能会持有本报告中涉及到的公司所发行的证券头寸并进行交易，并在法律许可的情况下不进行披露；可能为这些公司提供或争取提供投资银行业务、财务顾问等相关服务。

若本公司客户(以下称“该客户”)向第三方发送本报告，则由该客户独自为此发送行为负责。提醒通过此途径获得本报告的投资者注意，本公司不对通过此种途径获得本报告所引起的任何损失承担任何责任。

分析师声明

作者具有中国证券业协会授予的证券投资咨询执业资格，并在中国证券业协会注册登记为证券分析师。本报告遵循合规、客观、专业、审慎的制作原则，所采用数据、资料的来源合法合规，文字阐述反映了作者的真实观点，报告结论未受任何第三方的授意或影响，特此声明。

投资评级说明

股票投资评级说明	买入	未来 6 个月内，股价涨幅超越市场基准 15%以上。
	增持	未来 6 个月内，股价涨幅超越市场基准 5%至 15%之间。
	中性	未来 6 个月内，股价涨幅介于市场基准-5%至 5%之间。
	减持	在未来 6 个月内，股价涨幅落后市场基准 5%至 15%之间。
	卖出	未来 6 个月内，股价涨幅落后市场基准 15%以上。
行业投资评级说明	优于大势	未来 6 个月内，行业指数的收益超越市场平均收益。
	同步大势	未来 6 个月内，行业指数的收益与市场平均收益持平。
	落后大势	未来 6 个月内，行业指数的收益落后于市场平均收益。

东北证券股份有限公司

中国吉林省长春市
自由大路1138号
邮编：130021
电话：4006000686
传真：(0431)85680032
网址：http://www.nesc.cn

中国北京市西城区
锦什坊街28号
恒奥中心D座
邮编：100033
电话：(010)63210800
传真：(010)63210867

中国上海市浦东新区
源深路305号
邮编：200135
电话：(021)20361009
传真：(021)20361258

中国深圳南山区
大冲商务中心1栋2号楼24D
邮编：518000

机构销售

华北地区
销售总监李航
电话：(010) 63210896
手机：136-5103-5643
邮箱：lihang@nesc.cn

华东地区
销售总监袁颖
电话：(021) 20361100
手机：136-2169-3507
邮箱：yuanying@nesc.cn

华南地区
销售总监邱晓星
电话：(0755)33975865
手机：186-6457-9712
邮箱：qiuxx@nesc.cn