Contents

Acknowledgement I

Abstract V

1 Introduction 1
- 1.1 Background of HMM . . . 1
- 1.2 High-Order HMM . . . 2

2 First-Order HMM 3
- 2.1 EM Algorithm for First-Order HMM . . . 4
  - 2.1.1 E-Step . . . 4
  - 2.1.2 M-Step . . . 4
- 2.2 Viterbi Algorithm for First-Order HMM . . . 5
  - 2.2.1 Derivation of Max-sum Algorithm . . . 5
  - 2.2.2 Derivation of Viterbi Algorithm . . . 6

3 High-Order HMM 7
- 3.1 Model Setting-up . . . 7
- 3.2 Transforming into first order . . . 8
- 3.3 Parameter Learning- EM Algorithm . . . 9
  - 3.3.1 E-Step: Forward-Backward Algorithm subject to constraints . . . 9
  - 3.3.2 M-Step: Minimizing Expectation of Log-Likelihood Function . . . 13
- 3.4 Numerical Problems . . . 13
  - 3.4.1 Adjustments on α . . . 13
  - 3.4.2 Adjustments on β . . . 14
  - 3.4.3 Computation of c_i . . . 14
- 3.5 Adjusted Viterbi Algorithm . . . 15

4 Appendix 17
- 4.1 Reestimation on Parameters of High-Order HMM . . . 17
  - 4.1.1 Derivation of Log-Likelihood Function . . . 17

知乎 @华尔街老兵

4.1.2 EM Algorithm: E-Step 18
4.1.3 EM Algorithm: M-Step 18

知乎 @华尔街老兵

第一章　引言

1.1　隐马尔可夫模型背景

隐马尔可夫模型最初应用于语音识别的领域。在 1970 和 1980 年代，语音识别的研究者逐渐从传统的距离度量模型转移到一些由隐马尔可夫模型衍生出的统计模型。在这个过程中，Leonard E. Baum 做出了非常大的贡献，尤其是利用递归降低计算复杂度的 Baum-Welch 算法[2]。HMM 的结构在数学上和现实意义上都很巧妙：在数学上，有很多条件独立（d-Seperation）可以用来估计参数，同时也避免了观测值激活 v-structure 破坏独立条件；在现实意义上，它的结构和很多物理过程，比如发音、音乐演奏、证券交易都有相似的直觉。HMM 在使用、处理序列数据，尤其是变长度的序列数据任务中的便捷性让它在真实生活中有很多应用场景。

在半个世纪的发展中，研究者根据不同的应用需求 HMM 衍生出了不少改进模型，比如允许隐藏状态保持一段时间不变的隐半马尔科夫模型（HSMM），考虑到观测序列自回归效应的自回归隐马尔可夫模型（Auto-regressive HMM）等，并针对它们的特点改写了一阶 HMM 的参数学习方法。图1.1展示了一些较为知名的改进模型。

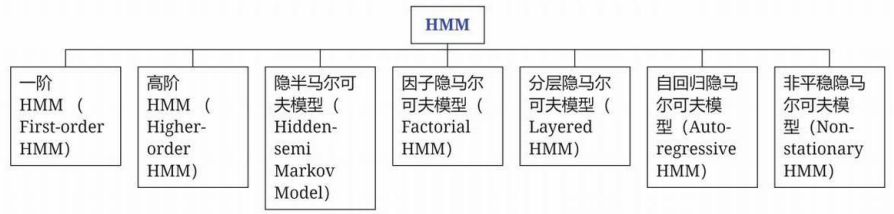

图 1.1: 隐马尔可夫模型的分类

除了在工业界的应用，在过去的 20 年内，也有很多研究将 HMM 用于金融时间序列的预测任务，不过它们大多数都是用的是一阶 HMM。比如 Gupta 和 Dhingra[6] 基于 HMM，用 MAP（Maximum a Posteriori）方法来预测下一交易日的股票价格。Erlwein[5] 给出了 HMM 在大宗商品和利率预测上的方法，并且提供了一个基于 HMM 来解决资产组合优化问题的框架。

知乎 @华尔街老兵

1.2 高阶隐马尔可夫模型

虽然一阶隐马尔可夫模型具有很优良的性能，也有非常易于理解的参数学习方法，但是它只能够捕捉隐藏状态短期的相关性。同时，它对系统长时间停留在某个状态的建模能力不太理想 [3]。为了解决这两个问题，高阶马尔可夫链和半马尔科夫过程分别被用来代替一阶 HMM 的马尔可夫链。本文将会聚焦高阶隐马尔可夫模型，简单说明它捕捉长期相关性的能力以及它部分解决一阶 HMM 对常驻状态情形建模能力欠佳的潜力。

事实上，尽管高阶 HMM 无法像隐半马尔科夫模型一样完全解决状态常驻概率建模上的弱点，它也受到了很多关注。Xiong 和 Mamon[14] 将高阶 HMM 微调后用在了气温预测上，并且用调整后的模型对一些天气衍生品进行了分析；在投资领域，Siu 等人 [10] 在使用高阶 HMM 对资产价格建模后，利用还原出的状态转移过程来衡量资产组合的风险;Zhu 等人 [16] 使用了考虑自回归的高阶 HMM（HO-HMMAR）来研究资产价格的波动，并用其解决了最优资产配置问题。

本文按照下面顺序组织：第二部分简要介绍一阶 HMM 及其参数学习的 EM（Expectation Maximization）算法，然后导出对应估计隐藏变量序列的 Viterbi 算法。第三部分先建立高阶 HMM 的模型，然后利用一个变换算法将高阶 HMM 转换成一阶 HMM，并根据变换方式重新给出相应的 EM 算法和 Viterbi 算法。第四部分通过实验说明高阶 HMM 在金融时间序列预测中的良好性能。最后一部分进行总结。

知乎 @华尔街老兵

第二章　一阶隐马尔可夫模型

一阶的隐马尔可夫模型是一个将有限状态和一组观测联系在一起的随机过程。它具有一组参数：$\Theta = \{\boldsymbol{A}, \phi, \pi\}$，其中 $\boldsymbol{A} = (a_{i,j})_{i,j}$ 用于表示隐藏状态的转移概率,$a_{i,j} = p(Z_n = j|Z_{n-1} = i)$，$\phi$ 是输出分布 $p(O_t|Z_t;\phi)$ 的参数，π 是初始状态的分布 $\pi_i = p(Z_0 = i)$

隐马尔可夫模型的隐藏状态一般没有具体意义。不过在实际应用中，它们可能会产生含义，比如 Zhang[15] 在金融市场数据中发现当使用三个隐藏状态时，它们会分别识别出上涨、横盘波动以及下跌三种情形。在本文中，用 $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$ 表示 N 个观测，其中 $\boldsymbol{x}_t = (x_{t,1}, x_{t,2}, \ldots, x_{t,K})$，也就是说观测数据是一个 K 维向量。于是，一个高斯隐马尔可夫模型在状态 i 下的输出概率就是一个高斯分布：

$$p(\boldsymbol{x}_t) = g(\boldsymbol{x}_t; \mu_i, \Sigma_i) \tag{2.1}$$

这里 $c_{i,k}$ 是在状态 i 时第 k 个混合部分的参数，$g(\boldsymbol{x}_t; \mu_{i,k}, \Sigma_{i,k})$ 时相应的多元正态分布密度函数：

$$g(\boldsymbol{x}_t; \mu_{i,k}, \Sigma_{i,k}) = \frac{1}{(2\pi)^{K/2} \det \Sigma_{i,k}{}^{1/2}} \exp[-\frac{1}{2}(\boldsymbol{x}_t - \mu_{i,k})' \Sigma_{i,k}^{-1} (\boldsymbol{x}_t - \mu_{i,k})] \tag{2.2}$$

因此利用贝叶斯网络的 d-sep 性质可以将整个模型所有随机变量的分布写为：

$$\begin{aligned} p(\mathbf{X}, \mathbf{Z}|\Theta) &= p(\mathbf{z}_1|\boldsymbol{\pi}) \left(\prod_{i=2}^{N} p(\mathbf{z}_i|\mathbf{z}_{i-1}; \mathbf{A}) \right) \prod_{i=1}^{N} p(\mathbf{x}_i|\mathbf{z}_i; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= \left(\prod_{k=0}^{K-1} \pi_k^{\tilde{z}_1} \right) \cdot \left(\prod_{i=2}^{N} \prod_{k=0}^{K-1} \prod_{j=0}^{K-1} A_{j,k}^{\tilde{z}_{i-1}\tilde{z}_i} \right) \cdot \left(\prod_{i=1}^{N} \prod_{k=0}^{K-1} \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}, \boldsymbol{\Sigma})^{\tilde{z}_i} \right) \end{aligned} \tag{2.3}$$

总体而言，本文的高斯隐马尔可夫模型需要估计的参数就是：$\lambda = \{\pi, \boldsymbol{A}, c_{i,k}, \mu_{i,k}, \Sigma_{i,k}\}$，$i = 0, 1, \ldots, K-1$。一阶隐马尔可夫模型有两种参数学习方法：1. 最小化观测值的对数似然函数 $\ln p(\boldsymbol{X})$ 来得到准确的最大似然估计 2. 最小化所有变量（观测值和隐藏变量）的似然函数 $\ln(\boldsymbol{X}, \boldsymbol{Z})$ 来得到近似的最大似然估计。后者常见于各种在线学习算法（On-line Learning）[4]

通过求解 $p(\boldsymbol{X})$ 的最大值点来进行最大似然估计的方法主要有两种：1. 通过 EM 算法不断提升对数似然函数的下界 2. 数值最优化求解。其中 EM 算法因为它较小的算力需求而得到更多关注。不过，随着高性能计算的兴起，直接使用数值算法来优化似然函数的方法越来越受欢迎。其中的一个原因是数值算法可以通过添加动量来避免陷入局部最优解。同时，在判别分析框架下的 HMM 可以利用更多样的损失函数来适应不同应用，比如使用最小分类误

知乎 @华尔街老兵

2.2 一阶 HMM 的 Viterbi 算法

Viterbi 算法最早由 [11] 提出，它在隐马尔可夫模型中可以用于“解码”模型的马尔可夫链，找到一个出现概率最高的马氏链的实现。事实上，在隐马尔可夫模型中使用 Viterbi 算法相当于使用 max-sum 算法来找到使得 $p(\mathbf{z}_1, \ldots, \mathbf{z}_N)$ 最大的序列 $\{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$。

为了使用 max-sum 算法，首先需要将一阶 HMM 的有向图表示改写为一个无向图（马尔可夫随机场）的形式。

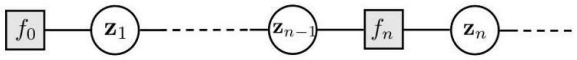

图 2.1: 以 MRF（Markov Random Field）方法表示的隐马尔可夫模型

其中，因子 $f_0(\tilde{z}_1)$ 是这个 MRF 的叶节点。因子具体形式是

$$f_n(\tilde{z}_{n-1}, \tilde{z}_n) = p(\tilde{z}_n|\tilde{z}_{n-1})p(\mathbf{x}_n|\tilde{z}_n) \tag{2.4}$$

$$f_0(\tilde{z}_1) = p(\tilde{z}_1)p(\mathbf{x}_1|\tilde{z}_1) \tag{2.5}$$

2.2.1 max-sum 算法导出

max-sum 算法的想法是非常直接：相对于直接在 $\{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ 上寻找使得 $p(\mathbf{Z})$ 最大的解，可以利用 MRF 上的因子将 $p(\mathbf{Z})$ 拆开，然后将一部分 $\arg\max$ 运算提前进行。事实上，对于一个因子为 $\{\psi_{1,2}(x_1, x_2), \ldots, \psi_{N-1,N}(x_{N-1}, x_N)\}$ 的链状 MRF，max-sum 算法可以将 $\mathcal{O}(K^N)$ 的时间复杂度降低到 $\mathcal{O}(KN)$。从公式上看：

$$\begin{aligned}\max_{x_1,\ldots,x_n} p(x_1, x_2, \ldots, x_N) =& \frac{1}{Z}\max_{x_1}\cdots\max_{x_N}\left[\psi_{1,2}(x_1, x_2)\ldots, \psi_{N-1,N}(x_{N-1}, x_N)\right] \\ =& \frac{1}{Z}\max_{x_1}\left[\psi_{1,2}(x_1, x_2)\left[\cdots\max_{x_N}\psi_{N-1,N}(x_{N-1}, x_N)\right]\right]\end{aligned}$$

计算的具体过程如下：上式并未显式写出 $x_N \to \psi_{N-1,N}$ 的信息传递过程 $\mu_{x_N\to\psi_{N-1,N}}$，这是因为当叶节点是一个随机变量时，实际的 max-sum 运算并不涉及到它，于是本文规定

$$\mu_{x_N\to\psi_{N-1,N}} := 1$$

然后求得最大值运算结果 $\mu_{\psi_{x_{N-1,N}}\to x_{N-1}}(x_N) = \max_{x_N}\psi_{N-1,N}(x_{N-1}, x_N)$。再次，需要将信息从 x_{N-1} 传递到 $\psi_{N-2,N-1}$，由于是链状 MRF，$\mu_{x_{N-1}\to\psi_{N-2,N-1}}(x_{N-1}) = \mu_{\psi_{x_{N-1,N}}\to x_{N-1}}(x_{N-1})$

在更加一般的树状 MRF 中，可以很容易将上面的过程扩展为一个所谓的“max-prod”算法：

$$\mu_{f\to x}(x) = \max_{\mathbf{y}\in Nb(f)\setminus\{x\}} f(x, \mathbf{y})\prod_{y_1,\ldots,y_M}\mu_{y_i\to f}(y_i)$$

知乎 @华尔街老兵

$$\mu_{x\to f}(x)=\prod_{i\in Nb(x)\setminus f}\mu_{f_i\to x}(x)$$

在实际的算法设计中，为了避免过多因子相乘而超出数值精度，又注意到 log（自然对数函数）是一个单调递增函数，于是利用

$$\log\left(\max_{\mathbf{x}}p(\mathbf{x})\right)=\max_{\mathbf{x}}\ln p(\mathbf{x})$$

就可以把求 $\arg\max_{\mathbf{Z}}p(\mathbf{Z})$ 的问题转化为 $\arg\max_{\mathbf{Z}}\ln p(\mathbf{Z})$ 的问题，这样本文就得到了概率图模型中常用的 max-sum 算法（对数函数将前面的“max-prod”转换为了“max-sum”）

$$\mu_{f\to x}(x)=\max_{\mathbf{y}\in Nb(f)\setminus\{x\}}f(x,\mathbf{y})\sum_{y_1,\dots,y_M}\mu_{y_i\to f}(y_i)\tag{2.6}$$

$$\mu_{x\to f}(x)=\sum_{i\in Nb(x)\setminus f}\mu_{f_i\to x}(x)\tag{2.7}$$

其中 $Nb(f)$ 表示与因子 f 在图上邻接的随机变量。

由叶节点传递出的初始信息与前面在“max-prod”方法中的规定类似：

$$\mu_{x\to f}(x)=0$$
$$\mu_{f\to x}(x)=\log f(x)$$

2.2.2 Viterbi 算法导出

在前面的2.4式中，本文已经给出了一阶 HMM 的 MRF 因子图表示。现在直接应用上一节的 max-sum 算法就可以得到使得 $p(\mathbf{z}_1,\dots,\mathbf{z}_N)$ 最大的 $\{\mathbf{z}_1,\dots,\mathbf{z}_N\}$ 序列。

HMM 的链状结构可以给出：

$$\mu_{z_n\to f_{n+1}}(\mathbf{z}_n)=\mu_{f_n\to z_n}(\mathbf{z}_n)\tag{2.8}$$

参考图2.1又可以得到：

$$\mu_{f_n\to z_n}(\mathbf{z}_n)=\max_{z_n}\left[\log f_n(\mathbf{z}_{n-1},\mathbf{z}_n)+\mu_{z_{n-1}\to f_n}(\mathbf{z}_{n-1})\right]\tag{2.9}$$

令 $\omega(\mathbf{z}_n)=\mu_{f_n\to z_n}(\mathbf{z}_n)$，结合2.8式 2.9式，那么 max-sum 算法的递归就是：

$$\omega(\mathbf{z}_n)=\log p(\mathbf{x}_n|\mathbf{z}_n)+\max_{\mathbf{z}_{n-1}}\left[\log p(\mathbf{z}_n|\mathbf{z}_{n-1})+\omega(\mathbf{z}_{n-1})\right]\tag{2.10}$$

利用2.5式，得到递归的起始公式：

$$\omega(\mathbf{z}_1)=\log p(\mathbf{z}_1)+\log p(\mathbf{x}_1|\mathbf{z}_1)\tag{2.11}$$

利用上面两个公式，就可以在进行 EM 算法收敛后，从 MRF 表示的一阶 HMM 的末端，也就叶节点 $\mathbf{z}_1$ 开始逐步向前计算，得到一个最大化似然函数的序列 $\{\mathbf{z}_1,\dots,\mathbf{z}_N\}$。

知乎 @华尔街老兵

第三章　高阶隐马尔可夫模型

在这一章，本文介绍 Hardar, Uri 等人 [7] 的方法将高阶 HMM 转换为一阶 HMM。然后基于他们的方法重新推导 Welch 和 Lyold R[13] 非常知名的 Baum-Welch 算法。

3.1　模型建立

本文假设隐藏状态 $\{\mathbf{z}_t\}_{t=1}^N$ 是 r 阶的齐次马尔可夫过程，隐藏状态 $\mathcal{S} = (0, 1, \ldots, K-1)$. 这个高阶马尔可夫过程满足：

$$p(\mathbf{z}_t|\{\mathbf{z}_i\}_{i<t}) = p(\mathbf{z}_t|\{\mathbf{z}_i\}_{i=t-r,\ldots,t-1}) \tag{3.1}$$

本文使用齐次过程，因此上式右端的转移概率对所有 $t = 1, 2, \ldots, N$ 均成立。

同时，本文假设生成观测值 $\{\mathbf{x}_t\}_{t=1}^N$ 的输出概率为：

$$p(\mathbf{x}_n|\{\mathbf{x}_i\}_{i\leq n}, \{\mathbf{z}_i\}_{i\leq n}) = p(\mathbf{x}_n|\{\mathbf{z}_i\}_{i=n-m+1,\ldots,n}) \tag{3.2}$$

也就是观测值 $\mathbf{x}_n$ 由从第 n 期开始向前共 m 个隐藏状态生成。

于是，可以将一阶 HMM 的参数重写。（隐藏状态 i 有两种表示方式，第一种是通过第 i 元为 1 其余为 0 的向量 $\boldsymbol{z}_n$ 表示，第二种是用一个标量 $\tilde{z}_n = i$ 表示。后面本文将在不引起误解的情况下使用波浪线 ˜ 进行随机变量的标量表示，使用数学粗体 $\boldsymbol{x}$ 进行随机变量的向量表示，使用斜体粗体 $\boldsymbol{\mu}$ 表示参数向量）

- $|\mathcal{S}|^{r+1}$ 个转移概率

$$A_{i_r\ldots i_0} = p(\tilde{z}_n = i_0|\tilde{z}_{n-1} = i_1, \ldots, \tilde{z}_{n-r} = i_r) \tag{3.3}$$

- $|\mathcal{S}|^m$ 个观测值的概率分布

$$p_{i_0\ldots i_{m-1}}(\mathbf{X}_n) = P(\mathbf{X}_n|\tilde{z}_n = i_0, \ldots, \tilde{z}_{n-m+1} = i_{m-1}) \tag{3.4}$$

- $|\mathcal{S}|^\nu$ 个初始状态概率

$$\tilde{\pi}_{i_1\ldots i_\nu} = P(\tilde{z}_1 = i_1, \tilde{z}_0 = i_2 \ldots, \tilde{z}_{2-\nu} = i_\nu) \tag{3.5}$$

其中 $\nu = \max\{r, m\}$，$i_0, i_1, \ldots, i_\nu \in \mathcal{S}$

知乎 @华尔街老兵

由于初始分布没有参数上的限制，3.37式就和一阶 HMM 的2.11式没有太大区别。由于转换后的高阶 HMM 仍然是链状的 MRF，因此

$$\mu_{q_n \to f_{n+1}}(\mathbf{q}_n) = \mu_{f_n \to q_n}(\mathbf{q}_n) \tag{3.38}$$

仍然成立，于是再次得到递归：

$$\begin{aligned}\omega(\mathbf{q}_n) = \mu_{f_n \to q_n}(\mathbf{q}_n) &= \max_{\mathbf{q}_{n-1}} \left[\log p(\mathbf{x}_n|\mathbf{q}_n) + \log p(\mathbf{q}_n|\mathbf{q}_{n-1}) + \mu_{q_{n-1} \to f_n}(\mathbf{q}_{n-1})\right] \\ &= \max_{q_{n-1}} \left[\log p(\mathbf{x}_n|\mathbf{q}_n) + \log p(\mathbf{q}_n|\mathbf{q}_{n-1}) + \omega(\mathbf{q}_{n-1})\right]\end{aligned} \tag{3.39}$$

3.39式与2.10式的一点区别在于：由于冗余的隐藏状态表示，最大化式中第二项会出现 $K^{\nu-r}$ 个解。但是实际上，部分的转移是无法实现的（$\tilde{q}_{n-1} = i$ 到 $\tilde{q}_n = j$ 的转移要求 $(i,j) \in \mathcal{E}_\nu(K)$）。因此需要将无法转移至 $\mathbf{q}_n$ 的 $\mathbf{q}_{n-1}$ 从求最大值的范围中剔除。

知乎 @华尔街老兵

第四章　附录

4.1　高阶 HMM 参数的重新估计

4.1.1　对数似然函数的导出

本文利用马尔可夫随机场的 d-sep 假设可以将 HMM 的观测值的分布 $p(\boldsymbol{X})$ 写为：

$$p(\mathbf{X}|\Theta) = p(\mathbf{q}_1|\boldsymbol{\pi}) \prod_{t=2}^{N} p(\mathbf{q}_t|\mathbf{q}_{t-1}, \mathbf{A}) \cdot \prod_{t=1}^{N} p(\mathbf{x}_t|\mathbf{q}_t, \boldsymbol{\phi}) \tag{4.1}$$

在高斯 HMM 的假设下，本文将不同的分布具体写为：

- 起始分布

$$p(\mathbf{q}_1|\boldsymbol{\pi}) = \prod_{i=0}^{K^\nu - 1} \pi_i^{q_{1,i}}$$

- 转移概率

$$p(\mathbf{q}_n|\mathbf{q}_{n-1}, \mathbf{A}) = \prod_{(j,k)\in\mathcal{E}_\nu(K)} A_{j,k}^{q_{n-1,j}q_{n,k}}$$

其中的 $\mathcal{I}_r$ 表示转移概率不为零的 (j,k) 组合。若 $(j,k) \notin \mathcal{I}_r$，那么概率为 0. 由于实际中并不会出现这种情况，所以并不需要将其写入似然函数中。

- 输出概率

$$p(\mathbf{x}_n|\mathbf{q}_n, \Phi) = \prod_{i=0}^{K^\nu - 1} N(\mathbf{x}_n|\mu_k, \Sigma_k)^{q_{n,i}}$$

其中 k 满足 $k = \lfloor \frac{i}{K^{\nu-m}} \rfloor$，每个 μ_k 对应 $K^{\nu-m}$ 种冗余状态表示（当 $r \le m$ 时无表示冗余，该表达也自然变为每个参数对应一种隐藏状态），$N(\cdot|\mu_i, \Sigma_i)$ 是隐藏状态为 i 时的正态分布密度函数。

于是对数似然函数就是：

$$\begin{aligned}\log p(\mathbf{X}|\Theta) = &\sum_{i=0}^{K^\nu-1} q_{1,i} \log \pi_i + \sum_{n=2}^{N} \sum_{(j,k)\in\mathcal{I}_r} q_{n-1,j}q_{n,k} \log A_{j,k} + \\ &\sum_{t=1}^{N} \sum_{i=0}^{K^\nu-1} q_{t,i} \left[-\frac{1}{2}(\mathbf{x}_t - \mu_k)'\Sigma_i^{-1}(\mathbf{x}_t - \mu_k) - \frac{n}{2}\log(\det \Sigma_k) \right] + const\end{aligned}$$

知乎 @华尔街老兵

其中 μ_k, Σ_k 的 k 与上面输出概率的记号一致，用于对应可能有的冗余状态表示。对数似然函数的期望：

$$\begin{aligned} Q(\Theta, \Theta^{old}) =& \mathbb{E}\big[\log p(\mathbf{X}|\Theta)\big] \\ =& \sum_{i=1}^{K^\nu} \gamma(q_{1,i}) \log \pi_i + \sum_{n=2}^{N} \sum_{(j,k)\in\mathcal{E}_\nu(K)} \xi(q_{n-1,j}, q_{n,k}) \log A_{j,k} + \\ & \sum_{t=1}^{N} \sum_{i=0}^{K^\nu-1} \gamma(q_{t,i}) \left[-\frac{1}{2}(\mathbf{x}_t - \mu_k)' \Sigma_k^{-1} (\mathbf{x}_t - \mu_k) - \frac{n}{2} \log(\det \Sigma_k) \right] \end{aligned}$$

之后在 $\sum_i \pi_i = 1$ 和 $\sum_j A_{i,j} = K^{\nu-r}, \forall i$ 限制下求解 Q 的最大值点即可。

为了推导上的方便，本文不严谨地规定允许 $\log 0 = -\infty$ 出现在推导式中，并规定：$-\infty \times 0 = 0$。这样的规定并不影响优化求解的结果，因为若 $(i \to j)$ 的转移不可能发生，即 $A_{i,j} = 0, \log A_{i,j} = -\infty$，那么 $\xi(q_{n-1,i}, q_{n,j}) = 0$，于是按照规定 $\xi(q_{n-1,i}, q_{n,j}) \log A_{i,j} = 0$。这与不进行规定并将求和限制在 $(i,j) \in \mathcal{E}_\nu(K)$ 时一致，但是在后续的代码复现中将带来便捷。于是，就可以将上式改写为

$$\begin{aligned} Q(\Theta, \Theta^{old}) =& \mathbb{E}\big[\log p(\mathbf{X}|\Theta)\big] \\ =& \sum_{i=1}^{K^\nu} \gamma(q_{1,i}) \log \pi_i + \sum_{n=2}^{N} \sum_{j=0}^{K^\nu-1} \sum_{k=0}^{K^\nu-1} \xi(q_{n-1,j}, q_{n,k}) \log A_{j,k} + \\ & \sum_{t=1}^{N} \sum_{i=0}^{K^\nu-1} \gamma(q_{t,i}) \left[-\frac{1}{2}(\mathbf{x}_t - \mu_k)' \Sigma_k^{-1} (\mathbf{x}_t - \mu_k) - \frac{n}{2} \log(\det \Sigma_k) \right] \end{aligned}$$

4.1.2 EM 算法-E 步骤

E 步骤需要求出对数似然函数的期望（在高斯 HMM 中，只需要求出充分统计量的的期望）。本步骤涉及到 Baum-Welch 算法的改写，在正文部分介绍。

记 $\gamma(q_{n,k}) = \mathbb{E}\big[q_{n,k}|\mathbf{X}\big]$，$\xi(q_{n-1,j}, q_{n,k}) = \mathbb{E}\big[q_{n-1,j}q_{n,k}|\mathbf{X}\big]$

4.1.3 EM 算法-M 步骤

在 M 步，最大化对数似然函数的期望。

通过拉格朗日乘数法可以得到：

$$\hat{\pi}_k = \frac{\gamma(q_{1,k})}{\sum_{j=0}^{K^\nu-1} \gamma(q_{1,j})} \tag{4.2}$$

$$\hat{A}_{i,j} = \frac{\sum_{n=2}^{N} \xi\left(q_{n-1,i}, q_{n,j}\right)}{\sum_{n=2}^{N} \sum_{k=0}^{K^\nu-1} \xi\left(q_{n-1,i}, q_{n,k}\right)} \tag{4.3}$$

部分转移概率应当为 0 的 (i,j) 对应的 $A_{i,j}$ 会因为 $\xi(q_{n,k}, q_{n+1}, \tilde{k}) = 0, \forall k \in \mathcal{I}_r(i)$ 从而计算结果为 0。

知乎 @华尔街老兵

参考文献

[1] Pierre Baldi and Yves Chauvin. Smooth on-line learning algorithms for hidden markov models. *Neural Computation*, 6(2):307–318, 1994.

[2] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning*, chapter 13, pages 610–635. Springer, 233 Spring Street, New York, NY 10013, USA, 2006.

[4] Vasilios V Digalakis. Online adaptation of hidden markov models using incremental estimation algorithms. *IEEE Transactions on Speech and Audio Processing*, 7(3):253–261, 1999.

[5] Christina Erlwein. *Applications of hidden Markov models in financial modelling.* PhD thesis, Brunel University, 5 2008.

[6] Aditya Gupta and Bhuwan Dhingra. Stock market prediction using hidden markov models. In *2012 Students Conference on Engineering and Systems*, pages 1–4, 2012.

[7] Uri Hadar et al. High-order hidden markov models-estimation and implementation. In *2009 IEEE/SP 15th Workshop on Statistical Signal Processing*, pages 249–252. IEEE.

[8] Xiaodong He and Li Deng. A new look at discriminative training for hidden markov models. *Pattern Recognition Letters*, 28(11):1285–1294, 2007. Advances on Pattern recognition for speech and audio processing.

[9] C. Rathinavelu and Li Deng. The trended hmm with discriminative training for phonetic classification. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96*, volume 2, pages 1049–1052 vol.2, 1996.

[10] T.K. Siu, W.K. Ching, E. Fung, M. Ng, and X. Li. A high-order markov-switching model for risk measurement. *Computers & Mathematics with Applications*, 58(1):1–10, 2009.

[11] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

知乎 @华尔街老兵

[12] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

[13] Lloyd R Welch. Hidden markov models and the baum-welch algorithm. *IEEE Information Theory Society Newsletter*, 53(4):10–13, 2003.

[14] Heng Xiong and Rogemar Mamon. A self-updating model driven by a higher-order hidden markov chain for temperature dynamics. *Journal of Computational Science*, 17:47–61, 2016.

[15] Mengqi Zhang, Xin Jiang, Zehua Fang, Yue Zeng, and Ke Xu. High-order hidden markov model for trend prediction in financial time series. *Physica A: Statistical Mechanics and its Applications*, 517:1–12, 2019.

[16] Dong-Mei Zhu, Jiejun Lu, Wai-Ki Ching, and Tak-Kuen Siu. Discrete-time optimal asset allocation under higher-order hidden markov model. *Economic Modelling*, 66:223–232, 2017.

[17] 高惠璇. 应用多元统计分析. 北京大学出版社, 2004.

知乎 @华尔街老兵