第十章、含时概率图模型

赵涵

2022 年 2 月 13 日

这一章我们介绍概率图模型与马尔可夫链的结合，马尔可夫模型是一种概率图模型。我们知道，机器学习模型可以从频率派和贝叶斯派两个方向考虑，在频率派的方法中的核心是优化问题，而在贝叶斯派的方法中，核心是积分问题，也发展出来了一系列的积分方法如变分推断，MCMC 等。概率图模型最基本的模型可以分为有向图（贝叶斯网络）和无向图（马尔可夫随机场）两个方面，例如GMM，在这些基本的模型上，如果样本之间存在关联，可以认为样本中附带了时序信息，从而样本之间不独立全同分布的，这种模型就叫做动态模型，隐变量随着时间发生变化，于是观测变量也发生变化。根据状态变量的特点，可以分为：

1. HMM，状态变量（隐变量）是离散的；
2. Kalman滤波，状态变量是连续的，线性的；
3. 粒子滤波，状态变量是连续，非线性的。

接下来我们一一进行介绍。

1 隐马尔可夫模型

隐马尔可夫模型（Hidden Markov Model）用概率图表示（1）： 上图表示了四个时刻的隐变量变化。用参数$\lambda = (\pi, A, B)$来表示，其中π是开始的概率分布，A为状态转移矩阵，B 为发射矩阵。使用o（observation）来表示观测变量，O为观测序列，$V = \{v_1, v_2, \ldots, v_M\}$ 表示观测的值域，i_t 表示状态变量，I 为状态序列，$Q = \{q_1, q_2, \ldots, q_N\}$表示状态变量的值域。定义$A = (a_{ij} = p(i_{t+1} = q_j | i_t = q_i))$ 表示状态转移矩阵，$B = (b_j(k) = p(o_t = v_k | i_t = q_j))$ 表示发射矩阵。在HMM 中，有两个基本假设：

1. 齐次Markov假设（未来只依赖于当前）：

$$p(i_{i+1} | i_t, i_{t-1}, \ldots, i_1, o_t, o_{t-1}, \ldots, o_1) = p(i_{t+1} | i_t) \tag{1}$$

2. 观测独立假设：

$$p(o_t | i_t, i_{t-1}, \ldots, i_1, o_{t-1}, \ldots, o_1) = p(o_t | i_t) \tag{2}$$

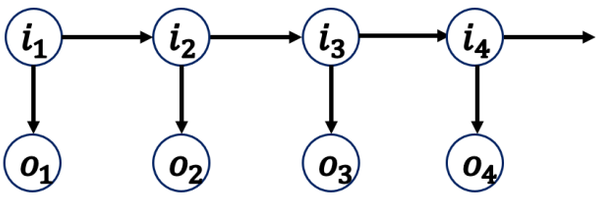

Figure 1: 隐马尔可夫模型的概率图表示。

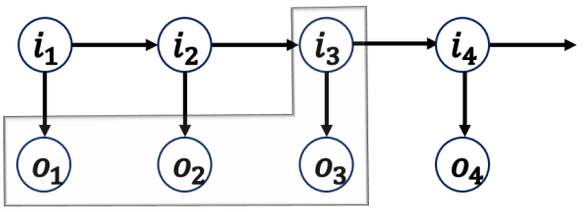

Figure 2: 记号α的含义。

HMM要解决三个问题：

1. 评估（Evaluation）：$p(O|\lambda)$，Forward-Backward算法
2. 学习（Learning）：$\lambda = \arg\max_\lambda p(O|\lambda)$，EM算法
3. 解码（Decoding）：$I = \arg\max_I p(I|O,\lambda)$，Virtbi算法
   - 预测问题：$p(i_{t+1}|o_1, o_2, \ldots, o_t)$
   - 滤波问题（filtering）：$p(i_t|o_1, o_2, \ldots, o_t)$

1.1 评估

$$p(O|\lambda) = \sum_I p(I, O|\lambda) = \sum_I p(O|I,\lambda)p(I|\lambda) \tag{3}$$

对于隐变量的分布，我们有：

$$p(I|\lambda) = p(i_1, i_2, \ldots, i_t|\lambda) = p(i_t|i_1, i_2, \ldots, i_{t-1}, \lambda)p(i_1, i_2, \ldots, i_{t-1}|\lambda) \tag{4}$$

根据齐次Markov假设：

$$p(i_t|i_1, i_2, \ldots, i_{t-1}, \lambda) = p(i_t|i_{t-1}) = a_{i_{t-1}i_t} \tag{5}$$

所以我们把隐变量的分布，进行化简：

$$p(I|\lambda) = \pi_1 \prod_{t=2}^{T} a_{i_{t-1}i_t} \tag{6}$$

又因为：

$$p(O|I,\lambda) = \prod_{t=1}^{T} b_{i_t}(o_t) \tag{7}$$

所以，我们有：

$$p(O|\lambda) = \sum_I \pi_{i_1} \prod_{t=2}^{T} a_{i_t-1,i_t} \prod_{i=1}^{T} b_{i_t}(o_t) \tag{8}$$

我们看到，上面的式子中的求和符号是对所有的状态变量求和，单个求和都是对N 次连加，所以复杂度为$O(N^T)$。下面，我们引入一个记号：

$$\alpha_t(i) = p(o_1, o_2, \ldots, o_t, i_t = q_i|\lambda) \tag{9}$$

这里从图（2），可以看出α 的意义。 所以：

$$\alpha_T(i) = p(O, i_T = q_i|\lambda) \tag{10}$$

我们可以看到：

$$p(O|\lambda) = \sum_{i=1}^{N} p(O, i_T = q_i|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{11}$$

对$\alpha_{t+1}(j)$：

$$\begin{aligned}
\alpha_{t+1}(j) &= p(o_1, o_2, \ldots, o_{t+1}, i_{t+1} = q_j|\lambda) && (12)\\
&= \sum_{i=1}^{N} p(o_1, o_2, \ldots, o_{t+1}, i_{t+1} = q_j, i_t = q_t|\lambda) && (13)\\
&= \sum_{i=1}^{N} p(o_{t+1}|o_1, o_2, \ldots, i_{t+1} = q_j, i_t = q_i|\lambda) p(o_1, \ldots, o_t, i_t = q_i, i_{t+1} = q_j|\lambda) && (14)
\end{aligned}$$

利用观测独立假设：

$$\begin{aligned}
\alpha_{t+1}(j) &= \sum_{i=1}^{N} p(o_{t+1}|i_{t+1} = q_j) p(o_1, \ldots, o_t, i_t = q_i, i_{t+1} = q_j|\lambda) && (15)\\
&= \sum_{i=1}^{N} p(o_{t+1}|i_{t+1} = q_j) p(i_{t+1} = q_j|o_1, \ldots, o_t, i_t = q_i, \lambda) p(o_1, \ldots, o_t, i_t = q_i|\lambda) && (16)\\
&= \sum_{i=1}^{N} b_j(o_t) a_{ij} \alpha_t(i) && (17)
\end{aligned}$$

上面利用了齐次Markov假设得到了一个递推公式，这个算法叫做前向算法。还有一种算法叫做后向算法，引入第二个记号：

$$\beta_t(i) = p(o_{t+1}, o_{t+2}, \ldots, o_T|i_t = i, \lambda) \tag{18}$$

所以，我们有：

$$\begin{aligned}
p(O|\lambda) &= p(o_1, \ldots, o_T|\lambda) && (19)\\
&= \sum_{i=1}^{N} p(o_1, o_2, \ldots, o_T, i_1 = q_i|\lambda) && (20)\\
&= \sum_{i=1}^{N} p(o_1, o_2, \ldots, o_T|i_1 = q_i, \lambda) \pi_i && (21)\\
&= \sum_{i=1}^{N} p(o_1|o_2, \ldots, o_T, i_1 = q_i, \lambda) p(o_2, \ldots, o_T|i_1 = q_i, \lambda) \pi_i && (22)\\
&= \sum_{i=1}^{N} b_i(o_1) \pi_i \beta_1(i) && (23)
\end{aligned}$$

最后一个等式，依旧利用了观测独立假设。对于$beta_1(i)$，我们有：

$$\begin{aligned}
\beta_t(i) &= p(o_{t+1},\ldots,o_T|i_t=q_i) && (24)\\
&= \sum_{j=1}^{N} p(o_{t+1},o_{t+2},\ldots,o_T,i_{t+1}=q_j|i_t=q_i) && (25)\\
&= \sum_{j=1}^{N} p(o_{t+1},\ldots,o_T|i_{t+1}=q_j,i_t=q_i)p(i_{t+1}=q_j|i_t=q_i) && (26)\\
&= \sum_{j=1}^{N} p(o_{t+1,\ldots,o_T}|i_{t+1}=q_i)a_{ij} && (27)\\
&= \sum_{j=1}^{N} p(o_{t+1}|o_{t+2},\ldots,o_T,i_{t+1}=q_j)p(o_{t+2},\ldots,o_T|i_{t+1}=q_j)a_{ij} && (28)\\
&= \sum_{j=1}^{N} b_j(o_{t+1})a_{ij}\beta_{t+1}(j) && (29)
\end{aligned}$$

第三个等号，我们去掉了i_t，因为在链式结构下，给定了i_{t+1}，就使得前后两个随机变量产生了独立性。这是由后往前迭代，所以叫后向算法（β_T可以默认为1）。

1.2 学习

为了学习得到参数的最优值，在MLE中：

$$\lambda_{MLE} = \arg\max_{\lambda} p(O|\lambda) \tag{30}$$

我们采用EM算法（在这里也叫Baum Welch 算法），用上标表示迭代：

$$\theta^{t+1} = \arg\max_{\theta} \int_z \log p(X,Z|\theta)p(Z|X,\theta^t)dz \tag{31}$$

其中，X是观测变量，Z是隐变量序列。于是：

$$\begin{aligned}
\lambda^{t+1} &= \arg\max_{\lambda} \sum_{I} \log p(O,I|\lambda)p(I|O,\lambda^t) && (32)\\
&= \arg\max_{\lambda} \sum_{I} \log p(O,I|\lambda)p(O,I|\lambda^t) && (33)
\end{aligned}$$

第二个等式，利用了$p(O|\lambda^t)$与λ 无关，且观测变量集合O也与λ无关。将Evaluation中的结果代入，有：

$$\sum_{I} \log p(O,I|\lambda)p(O,I|\lambda^t) = \sum_{I}[\log \pi_{i_1} + \sum_{t=2}^{T} \log a_{i-1,i_t} + \sum_{t=1}^{T} \log b_{i_t}(o_t)]p(O,I|\lambda^t) \tag{34}$$

对π^{t+1}：

$$\begin{aligned}
\pi^{t+1} &= \arg\max_{\pi} \sum_{I}[\log \pi_{i_1} p(O,I|\lambda^t)] && (35)\\
&= \arg\max_{\pi} \sum_{I}[\log \pi_{i_1} \cdot p(O,i_1,i_2,\ldots,i_T|\lambda^t)] && (36)
\end{aligned}$$

上面的式子中，对$i_1,i_2,\ldots,i_T$ 求和可以将这些参数消掉：

$$\pi^{t+1} = arg\max_{\pi} \sum_{i_1}[\log \pi_{i_1} \cdot p(O,i_{i_1}|\lambda^t)] \tag{37}$$

上面的式子还有对π的约束$\sum_i \pi_i = 1$。定义Lagrange函数：

$$L(\pi, \eta) = \sum_{i=1}^{N} \log \pi_{i_1} \cdot p(O, i_1 = q_i|\lambda^t) + \eta(\sum_{i=1}^{N} \pi_i - 1) \tag{38}$$

于是有：

$$\frac{\partial L}{\partial \pi_i} = \frac{1}{\pi_i} p(O, i_1 = q_i|\lambda^t) + \eta = 0 \tag{39}$$

对上式求和：

$$\sum_{i=1}^{N} p(O, i_1 = q_i|\lambda^t) + \pi_i \eta = 0 \Rightarrow \eta = -p(O|\lambda^t) \tag{40}$$

化简得到：

$$\pi_i^{t+1} = \frac{p(O, i_1 = q_i|\lambda^t)}{p(O|\lambda^t)} \tag{41}$$

1.3 译码

Decoding问题表述为：

$$I = \arg\max_I p(I|O, \lambda) \tag{42}$$

我们需要找到一个序列，其概率最大，这个序列就是在参数空间中的一个路径，可以采用动态规划的思想。定义：

$$\delta_t(j) = \max_{i_1, \ldots, i_{t-1}} p(o_1, \ldots, o_t, i_1, \ldots, i_{t-1}, i_t = q_i) \tag{43}$$

于是：

$$\delta_{t+1}(j) = \max_{1 \le i \le N} \delta_t(i) a_{ij} b_j(o_{t+1}) \tag{44}$$

这个式子就是从上一步到下一步的概率再求最大值。记这个路径为：

$$\psi_{t+1}(j) = \arg \max_{1 \le i \le N} \delta_t(i) a_{ij} \tag{45}$$

1.4 HMM总结

HMM是一种动态模型，是由混合树形模型和时序结合起来的一种模型（类似GMM+Time）。可以记为，一个模型，两个假设，三个问题。对于类似HMM的这种状态空间模型，普遍的除了学习任务（采用EM）外，还有推断任务，推断任务包括：

1. 译码Decoding：$p(z_1, z_2, \ldots, z_t|x_1, x_2, \ldots, x_t)$
2. 似然概率：$p(X|\theta)$
3. 滤波：$p(z_t|x_z, \ldots, x_t)$，可以记为：

$$p(z_t|x_{1:t}) = \frac{p(x_{1:t}, z_t)}{p(x_{1:t})} = C\alpha_t(z_t) \tag{46}$$

4. 平滑：$p(z_t|x_1, \ldots, x_T)$，可以记为：

$$p(z_t|x_{1:T}) = \frac{p(x_{1:T}, z_t)}{p(x_{1:T})} = \frac{\alpha_t(z_t) p(x_{t+1:T}|x_{1:t}, z_t)}{p(x_{1:T})} \tag{47}$$

根据概率图的条件独立性，有：

$$p(z_t|x_{1:T}) = \frac{p(x_{1:T}, z_t)}{p(x_{1:T})} = C\alpha_t(z_t)\beta_t(z_t) \tag{48}$$

这个算法叫做前向后向算法。

5. 预测：$p(z_{t+1}z_{t+2}|x_1,\ldots,x_t), p(x_{t+1},x_{t+2}|x_1,\ldots,x_t)$：

$$p(z_{t+1}|x_{1:t}) = \sum_{z_t} p(z_{t+1},z_t|x_{1:t}) = \sum_{z_t} p(z_{t+1}|z_t)p(z_t|x_{1:t}) \tag{49}$$

$$p(x_{t+1}|x_{1:t}) = \sum_{x_{t+1}} p(x_{t+1},z_{t+1}|x_{1:t}) = \sum_{z_{t+1}} p(x_{t+1}|z_{t+1})p(z_{t+1}|x_{1:t}) \tag{50}$$

2 卡曼滤波——线性动态系统

HMM模型适用于隐变量是离散的值的时候，对于连续隐变量的HMM，常用线性动态系统描述线性高斯模型的状态变量，使用粒子滤波来表述非高斯非线性的状态变量。LDS 又叫卡尔曼滤波，线性体现在上一时刻和这一时刻的隐变量以及隐变量和观测之间：

$$\begin{aligned} z_t &= Az_{t-1} + B + \epsilon & (51)\\ x_t &= Cz_t + D + \delta & (52)\\ \epsilon &\sim N(0,Q) & (53)\\ \delta &\sim N(0,R) & (54) \end{aligned}$$

套用HMM中的几个参数：

$$\begin{aligned} p(z_t|z_{t-1}) &\sim N(Az_{t-1}+B,Q) & (55)\\ p(x_t|z_t) &\sim N(Cz_t+D,R) & (56)\\ z_1 &\sim N(\mu_1,\Sigma_1) & (57) \end{aligned}$$

在含时的概率图中，除了对参数估计的学习问题外，在推断任务中，包括译码，证据概率，滤波，平滑，预测问题，LDS更关心滤波这个问题：$p(z_t|x_1,x_2,\ldots,x_t)$。类似HMM 中的前向算法，我们需要找到一个递推关系。

$$p(z_t|x_{1:t}) = p(x_{1:t},z_t)/p(x_{1:t}) = Cp(x_{1:t},z_t) \tag{58}$$

对于$p(x_{1:t},z_t)$：

$$\begin{aligned} p(x_{1:t},z_t) &= p(x_t|x_{1:t-1},z_t)p(x_{1:t-1},z_t) & (59)\\ &= p(x_t|z_t)p(x_{1:t-1},z_t) & (60)\\ &= p(x_t|z_t)p(z_t|x_{1:t-1})p(x_{1:t-1}) & (61)\\ &= Cp(x_t|z_t)p(z_t|x_{1:t-1}) & (62) \end{aligned}$$

可以看到，右边除了只和观测相关的常数项，还有一项是预测任务需要的概率。对这个值：

$$\begin{aligned} p(z_t|x_{1:t-1}) &= \int_{z_{t-1}} p(z_t,z_{t-1}|x_{1:t-1})dz_{t-1} & (63)\\ &= \int_{z_{t-1}} p(z_t|z_{t-1},x_{1:t-1})p(z_{t-1}|x_{1:t-1})dz_{t-1} & (64)\\ &= \int_{z_{t-1}} p(z_t|z_{t-1})p(z_{t-1}|x_{1:t-1})dz_{t-1} & (65) \end{aligned}$$

我们看到，这又化成了一个滤波问题。于是我们得到了一个递推公式：

- $t=1$，$p(z_1,x_1)$，称为update 过程，然后计算$p(z_2|x_1)$，通过上面的积分进行，称为prediction过程。

- $t=2$，$p(z_2|x_2,x_1)$和$p(z_3|x_1,x_2)$。

我们看到，这个过程是一个Online的过程，对于我们的线性高斯假设，这个计算过程都可以得到解析解。

1. Prediction：

$$p(z_t|x_{1:t-1})=\int_{z_{t-1}}p(z_t|z_{t-1})p(z_{t-1}|x_{1:t-1})dz_{t-1}=\int_{t-1}N(Az_{t-1}+B,Q)N(\mu_{t-1},\Sigma_{t-1})dz_{t-1} \tag{66}$$

其中第二个高斯分布是上一步的Update 过程，所以根据线性高斯模型，直接可以写出这个积分：

$$p(z_t|x_{1:t-1})=N(A\mu_{t-1}+B,Q+A\Sigma_{t-1}A^T) \tag{67}$$

2. Update：

$$p(z_t|x_{1:t})\propto p(x_t|z_t)p(z_t|x_{1:t-1}) \tag{68}$$

同样利用线性高斯模型，也可以直接写出这个高斯分布。

3 粒子滤波

Kalman滤波根据线性高斯模型可以求得解析解，但是在非线性，非高斯的情况，是无法得到解析解的，对这类一般的情况，我们叫做粒子滤波，我们需要求得概率分布，需要采用采样的方式。

我们希望应用Monte Carlo方法来进行采样，对于一个概率分布，如果我们希望计算依这个分布的某个函数$f(z)$ 的期望，可以利用某种抽样方法，在这个概率分布中抽取N 个样本，则$E[f(z)]\simeq\frac{1}{N}\sum_{i=1}^{N}f(z_i)$。但是如果这个概率十分复杂，那么采样比较困难。对于复杂的概率分布，我们可以通过一个简单的概率分布$q(z)$作为桥梁（重要性采样）：

$$E[f(z)]=\int_z f(z)p(z)dz=\int_z f(z)\frac{p(z)}{q(z)}q(z)dz=\sum_{i=1}^{N}f(z_i)\frac{p(z_i)}{q(z_i)} \tag{69}$$

于是直接通过对$q(z)$采样，然后对每一个采样的样本应用权重就得到了期望的近似，当然为了概率分布的特性，我们需要对权重进行归一化。

在滤波问题中，需要求解$p(z_t|x_{1:t})$，其权重为：

$$w_t^i=\frac{p(z_t^i|x_{1:t})}{q(z_t^i|x_{1:t})},i=1,2,\ldots,N \tag{70}$$

于是在每一个时刻t，都需要采样N个点，但是即使采样了这么多点，分子上面的那一项也十分难求，于是希望找到一个关于权重的递推公式。为了解决这个问题，引入**序列重要性采样**（Sequential importance sampling,SIS）。

3.1 序列重要性采样

在SIS中，解决的问题是$p(z_{1:t}|x_{1:t})$，权重为：

$$w_t^i\propto\frac{p(z_{1:t}|x_{1:t})}{q(z_{1:t}|x_{1:t})} \tag{71}$$

根据LDS中的推导：

$$\begin{aligned} p(z_{1:t}|x_{1:t}) \propto p(x_{1:t}, z_{1:t}) &= p(x_t|z_{1:t}, x_{1:t-1})p(z_{1:t}, x_{1:t-1}) && (72)\\ &= p(x_t|z_t)p(z_t|x_{1:t-1}, z_{1:t-1})p(x_{1:t-1}, z_{1:t-1}) && (73)\\ &= p(x_t|z_t)p(z_t|z_{1:t-1})p(x_{1:t-1}, z_{1:t-1}) && (74)\\ &\propto p(x_t|z_t)p(z_t|z_{1:t-1})p(z_{1:t-1}|x_{1:t-1}) && (75)\end{aligned}$$

于是分子的递推式就得到了。对于提议分布的分母，可以取：

$$q(z_{1:t}|x_{1:t}) = q(z_t|z_{1:t-1}, x_{1:t})q(z_{1:t-1}|x_{1:t-1}) \tag{76}$$

所以有：

$$w_t^i \propto \frac{p(z_{1:t}|x_{1:t})}{q(z_{1:t}|x_{1:t})} \propto \frac{p(x_t|z_t)p(z_t|z_{t-1})p(z_{1:t-1}|x_{1:t-1})}{q(z_t|z_{1:t-1}, x_{1:t})q(z_{1:t-1}|x_{1:t-1})} = \frac{p(x_t|z_t)p(z_t|z_{t-1})}{q(z_t|z_{1:t-1}, x_{1:t})}w_{t-1}^i \tag{77}$$

我们得到的对权重的算法为：

1. $t-1$时刻，采样完成并计算得到权重；
2. t时刻，根据$q(z_t|z_{1:t-1}, x_{1:t})$ 进行采样得到z_t^i。然后计算得到N个权重；
3. 最后对权重归一化。

SIS算法会出现权值退化的情况，在一定时间后，可能会出现大部分权重都逼近0的情况，这是由于空间维度越来越高，需要的样本也越来越多。解决这个问题的方法有：

1. 重采样，以权重作为概率分布，重新在已经采样的样本中采样，然后所有样本的权重相同，这个方法的思路是将权重作为概率分布，然后得到累积密度函数，在累积密度上取点（阶梯函数）。
2. 选择一个合适的提议分布，$q(z_t|z_{1:t-1}, x_{1:t}) = p(z_t|z_{t-1})$，于是就消掉了一项，并且采样的概率就是$p(z_t|z_{t-1})$，这就叫做生成与测试方法。

采用重采样的SIS算法就是基本的粒子滤波算法。如果采用上面那样选择提议分布，这个算法叫做SIR算法。

4 条件随机场

我们知道，分类问题可以分为硬分类和软分类两种，其中硬分类有SVM，PLA，LDA等。软分类问题大体上可以分为概率生成和概率判别模型，其中较为有名的概率判别模型有Logisti 回归，生成模型有朴素贝叶斯模型。Logistic 回归模型的损失函数为交叉熵，这类模型也叫对数线性模型，一般地，又叫做最大熵模型，这类模型和指数族分布的概率假设是一致的。对朴素贝叶斯假设，如果将其中的单元素的条件独立性做推广到一系列的隐变量，那么，由此得到的模型又被称为动态模型，比较有代表性的如HMM，从概率意义上，HMM也可以看成是GMM 在时序上面的推广。

我们看到，一般地，如果将最大熵模型和HMM 相结合，那么这种模型叫做最大熵Markov模型（MEMM），如图（3）： 这个图就是将HMM的图中观测变量和隐变量的边方向反向，应用在分类中，隐变量就是输出的分类，这样HMM 中的两个假设就不成立了，特别是观测之间不是完全独立的。

HMM是一种生成式模型，其建模对象为$p(X, Y|\lambda)$，根据HMM的概率图，$p(X, Y|\lambda) = \prod_{t=1}^{T} p(x_t, y_t|\lambda, y_{t-1})$。我们看到，观测独立性假设是一个很强的假设，如果我们有一个文本样本，那么观测独立性假设就假定

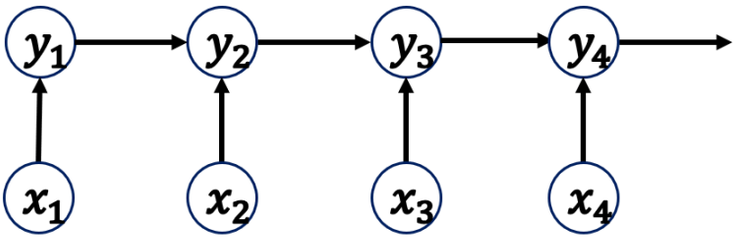

Figure 3: 条件随机场的概率图表示。

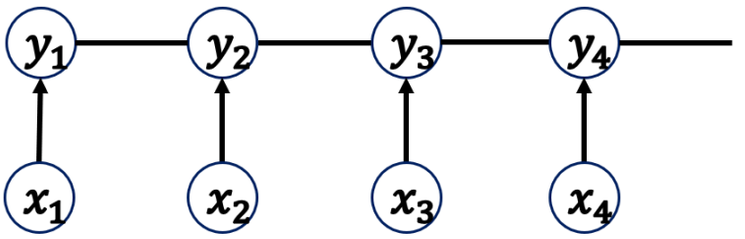

Figure 4: 线性链条件随机场。

了所有的单词之间没有关联。在MEMM中，建模对象是$p(Y|X,\lambda)$，我们看概率图，给定y_t，x_t, x_{t-1}是不独立的，所以观测独立假设就不成立了。根据概率图，$p(Y|X,\lambda) = \prod_{t=1}^{T} p(y_t|y_{t-1}, X, \lambda)$。

MEMM的缺陷是其必须满足局域的概率归一化（Label Bias Problem），我们看到，在上面的概率图中，$p(y_t|y_{t-1}, x_t)$，这个概率如果$p(y_t|y_{t-1})$ 常接近1，那么事实上，观测变量是什么就不会影响这个概率了。

对于这个问题，我们将y之间的有向边转为无向图（线性链条件随机场），这样就只要满足全局归一化了（破坏齐次Markov 假设），如图（4）。 线性链的CRF的概率密度函数为$p(Y|X) = \frac{1}{Z} \exp \sum_{t=1}^{T} (F_t(y_{t-1}, y_t, x_{1:T}))$，两两形成了最大团，其中$y_0$是随意外加的一个元素。作为第一个简化，我们假设每个团的势函数相同$F_t = F$。对于这个F，我们进一步，可以将其写为$F(y_{t-1}, y_t, X) = \Delta_{y_{t-1},X} + \Delta_{y_t,X} + \Delta_{y_t,y_{t-1},X}$ 这三个部分，分别表示状态函数已经转移函数，由于整体的求和，可以简化为$F(y_{t-1}, y_t, X) = \Delta_{y_t,X} + \Delta_{y_t,y_{t-1},X}$。我们可以设计一个表达式将其参数化：

$$\Delta_{y_t,y_{t-1},X} = \sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_t, X) \tag{78}$$

$$\Delta_{y_t,X} = \sum_{l=1}^{L} \eta_l g_l(y_t, X) \tag{79}$$

其中g, f叫做特征函数，对于y有S 种元素，那么$K \leq S^2, L \leq S$。代入概率密度函数中：

$$p(Y|X) = \frac{1}{Z} \exp \sum_{t=1}^{T} [\sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_t, X) + \sum_{l=1}^{L} \eta_l g_l(y_t, X)] \tag{80}$$

对于单个样本，将其写成向量的形式。定义：

$$y = (y_1, y_2, \ldots, y_T)^T \tag{81}$$

$$x = (x_1, x_2, \ldots, x_T)^T \tag{82}$$

$$\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_K)^T \tag{83}$$

$$\eta = (\eta_1, \eta_2, \ldots, \eta_L)^T \tag{84}$$

并且有$f=(f_1,f_2,\ldots,f_K)^T, g=(g_1,g_2,\ldots,g_L)^T$。于是：

$$p(Y=y|X=x)=\frac{1}{Z}\exp\sum_{t=1}^{T}[\lambda^T f(y_{t-1},y_t,x)+\eta^T g(y_t,x)] \tag{85}$$

不妨记：$\theta=(\lambda,\eta)^T, H=(\sum_{t=1}^{T} f, sum_{t=1}^{T} g)^T$：

$$p(Y=y|X=x)=\frac{1}{Z(x,\theta)}\exp[\theta^T H(y_t,y_{t-1},x)] \tag{86}$$

上面这个式子是一个指数族分布，Z是配分函数。

CRF需要解决下面几个问题：

1. Learning：参数估计问题，对N 个T 维样本，$\hat{\theta}=\arg\max_\theta \prod_{i=1}^{N} p(y^{(i)}|x^{(i)})$，这里用上标表示样本的编号。

2. Inference：

   (a) 边缘概率：$p(y_t|x)$；

   (b) 条件概率：一般在生成模型中较为关注，CRF中不关注；

   (c) MAP推断：$\hat{y}=\arg\max p(y|x)$。

先看边缘概率：边缘概率这个问题描述为，根据学习任务得到的参数，给定了$p(Y=y|X=x)$，求解$p(y_t=i|x)$。根据无向图可以给出：

$$p(y_t=i|x)=\sum_{y_{1:t-1},y_{t+1}:T} p(y|x)=\sum_{y_{1:t-1}}\sum_{y_{t+1:T}}\frac{1}{Z}\prod_{t'=1}^{T}\phi_{t'}(y_{t'-1},y_{t'},x) \tag{87}$$

我们看到上面的式子，直接计算的复杂度很高，这是由于求和的复杂度在$O(S^T)$，求积的复杂度在$O(T)$，所以整体复杂度为$O(TS^T)$。我们需要调整求和符号的顺序，从而降低复杂度。首先，将两个求和分为：

$$p(y_t=i|x)=\frac{1}{Z}\Delta_l\Delta_r \tag{88}$$

$$\Delta_l=\sum_{y_{1:t-1}}\phi_1(y_1,y_1,x)\phi_2(y_1,y_2,x)\cdots\phi_{t-1}(y_{t-2},y_{t-1},x)\phi_t(y_{t-1},y_t=i,x) \tag{89}$$

$$\Delta_r=\sum_{y_{t+1:T}}\phi_{t+1}(y_t=i,y_{t+1},x)\phi_{t+2}(y_{t+1},y_{t+2},x)\cdots\phi_T(y_{T-1},y_T,x) \tag{90}$$

对于Δ_l，从左向右，一步一步将y_t 消掉：

$$\Delta=\sum_{y_{t-1}}\phi_t(y_{t-1},y_t=i,x)\sum_{y_{t-2}}\phi_{t-1}(y_{t-2},y_{t-1},x)\cdots\sum_{y_0}\phi_1(y_0,y_1,x) \tag{91}$$

引入：

$$\alpha_t(i)=\Delta_l \tag{92}$$

于是：

$$\alpha_t(i)=\sum_{j\in S}\phi_t(y_{t-1}=j,y_t=i,x)\alpha_{t-1}(j) \tag{93}$$

这样我们得到了一个递推式。类似地，$\Delta_r=\beta_t(i)=\sum_{j\in S}\phi_{t+1}(y_t=i,y_{t+1}=j,x)\beta_{t+1}(j)$。这个方法和HMM中的前向后向算法类似，就是概率图模型中精确推断的变量消除算法（信念传播）。在进行各

种类型的推断之前，还需要对参数进行学习：

$$\hat{\theta} = \arg\max_{\theta} \prod_{i=1}^{N} p(y^{(i)}|x^{(i)}) \tag{94}$$

$$= \arg\max_{\theta} \sum_{i=1}^{N} \log p(y^{(i)}|x^{(i)}) \tag{95}$$

$$= \arg\max_{\theta} \sum_{i=1}^{N} [-\log Z(x^{(i)}, \lambda, \eta) + \sum_{t=1}^{T} [\lambda^T f(y_{t-1}, y_t, x) + \eta^T g(y_t, x)]] \tag{96}$$

上面的式子中，第一项是对数配分函数，根据指数族分布的结论：

$$\nabla_\lambda (\log Z(x^{(i)}, \lambda, \eta)) = E_{p(y^{(i)}|x^{(i)})} [\sum_{t=1}^{T} f(y_{t-1}, y_t, x^{(i)})] \tag{97}$$

其中，和η相关的项相当于一个常数。求解这个期望值：

$$E_{p(y^{(i)}|x^{(i)})} [\sum_{t=1}^{T} f(y_{t-1}, y_t, x^{(i)})] = \sum_{y} p(Y|x^{(i)} \sum_{t=1}^{T} f(y_{t-1}, y_t, x^{(i)})) \tag{98}$$

第一个求和号的复杂度为$O(S^T)$，重新排列求和符号：

$$E_{p(y^{(i)}|x^{(i)})} [\sum_{t=1}^{T} f(y_{t-1}, y_t, x^{(i)})] = \sum_{t=1}^{T} \sum_{y_{1:t-2}} \sum_{y_{t-1}} \sum_{y_t} \sum_{y_{t+1:T}} p(y|x^{(i)}) f(y_{t-1}, y_t, x^{(i)}) \tag{99}$$

$$= \sum_{t=1}^{T} \sum_{y_{t-1}} \sum_{y_t} p(y_{t-1}, y_t|x^{(i)}) f(y_{t-1}, y_t, x^{(i)}) \tag{100}$$

和上面的边缘概率类似，也可以通过前向后向算法得到上面式子中的边缘概率。于是：

$$\nabla_\lambda L = \sum_{i=1}^{N} \sum_{t=1}^{T} [f(y_{t-1}, y_t, x^{(i)}) - \sum_{y_{t-1}} \sum_{y_t} p(y_{t-1}, y_t|x^{(i)}) f(y_{t-1}, y_t, x^{(i)})] \tag{101}$$

利用梯度上升算法可以求解。对于η 也是类似的过程。

最后还有一个译码的问题，译码问题和HMM 中的Viterbi 算法类似，同样采样动态规划的思想一层一层求解最大值。

5 高斯网络

高斯图模型（高斯网络）是一种随机变量为连续的有向或者无向图。有向图版本的高斯图是高斯贝叶斯网络，无向版本的叫高斯马尔可夫网络。高斯网络的每一个节点都是高斯分布：$N(\nu_i, \Sigma_i)$，于是所有节点的联合分布就是一个高斯分布，均值为μ，方差为Σ。对于边缘概率，我们有下面三个结论：

1. 对于方差矩阵，可以得到独立性条件：$x_i \perp x_j \Leftrightarrow \sigma_{ij} = 0$，这个叫做全局独立性。
2. 我们看方差矩阵的逆（精度矩阵或信息矩阵）：$\Lambda = \Sigma^{-1} = (\lambda_{ij})_{pp}$，有定理：$x_i \perp x_j | (X - \{x_i, x_j\}) \Leftrightarrow \lambda_{ij} = 0$。因此，我们使用精度矩阵来表示条件独立性。
3. 对于任意一个无向图中的节点x_i，

$$x_i|(X - x_i) \sim N(\sum_{j \neq i} \frac{\lambda_{ij}}{\lambda_{ii}} x_j, \lambda_{ii}^{-1}) \tag{102}$$

也就是其他所有分量的线性组合，即所有与它有链接的分量的线性组合。

5.1 高斯贝叶斯网络GBN

高斯贝叶斯网络可以看成是LDS的一个推论，LDS 的假设是相邻时刻的变量之间的依赖关系，因此是一个局域模型，而高斯贝叶斯网络，每一个节点的父亲节点不一定只有一个，因此可以看成是一个全局的模型。根据有向图的因子分解：

$$p(x) = \prod_{i=1}^{p} p(x_i|x_{Parents(i)}) \tag{103}$$

对里面每一项，假设每一个特征是一维的，可以写成线性组合：

$$p(x_i|x_{Parents(i)}) = N(x_i|\mu_i + W_i^T x_{Parents(i)}, \sigma_i^2) \tag{104}$$

将随机变量写成：

$$x_i = \mu_i + \sum_{j \in x_{Parents(i)}} w_{ij}(x_j - \mu_j) + \sigma_i \epsilon_i, \epsilon_i \sim N(0,1) \tag{105}$$

写成矩阵形式，并且对w进行扩展：

$$x - \mu = W(x - \mu) + S\epsilon \tag{106}$$

其中，$S = diag(\sigma_i)$。所以有：$x - \mu = (\mathbb{I} - W)^{-1} S\epsilon$。由于：

$$Cov(x) = Cov(x - \mu) \tag{107}$$

可以得到协方差矩阵。

5.2 高斯马尔可夫网络GMN

对于无向图版本的高斯网络，可以写成：

$$p(x) = \frac{1}{Z} \prod_{i=1}^{p} \phi_i(x_i) \prod_{i,j \in X} \phi_{i,j}(x_i, x_j) \tag{108}$$

为了将高斯分布和这个式子结合，我们写出高斯分布和变量相关的部分：

$$\begin{aligned} p(x) &\propto \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)) && (109)\\ &= \exp(-\frac{1}{2}(x^T \Lambda x - 2\mu^T \Lambda x + \mu^T \Lambda \mu)) && (110)\\ &= \exp(\frac{1}{2} x^T \Lambda x + (\Lambda\mu)^T x) && (111) \end{aligned}$$

可以看到，这个式子与无向图分解中的两个部分对应，我们记$h = \Lambda\mu$为Potential Vector。其中和x_i相关的为：$x_i : -\frac{1}{2}\lambda_{ii} x_i^2 + h_i x_i$，与$x_i, x_j$相关的是：$x_i, x_j : -\lambda_{ij} x_i x_j$，这里利用了精度矩阵为对称矩阵的性质。我们看到，这里也可以看出，x_i, x_j构成的一个势函数，只和λ_{ij}有关，于是$x_i \perp x_j | (X - \{x_i, x_j\}) \Leftrightarrow \lambda_{ij} = 0$。

6 贝叶斯线性回归

我们知道，线性回归当噪声为高斯分布的时候，最小二乘损失导出的结果相当于对概率模型应用MLE，引入参数的先验时，先验分布是高斯分布，那么MAP的结果相当于岭回归的正则化，如果先验是拉普拉斯分布，那么相当于Lasso的正则化。这两种方案都是点估计方法。我们希望利用贝叶斯方法来求解参数的后验分布。

线性回归的模型假设为：

$$f(x) = w^T x \tag{112}$$

$$y = f(x) + \epsilon \tag{113}$$

$$\epsilon \sim N(0, \sigma^2) \tag{114}$$

在贝叶斯方法中，需要解决推断和预测两个问题。

6.1 推断

引入高斯先验：

$$p(w) = N(0, \Sigma_p) \tag{115}$$

对参数的后验分布进行推断：

$$p(w|X, Y) = \frac{p(w, Y|X)}{p(Y|X)} = \frac{p(Y|w, X)p(w|X)}{\int p(Y|w, X)p(w|X)dw} \tag{116}$$

分母和参数无关，由于$p(w|X) = p(w)$，代入先验得到：

$$p(w|X, Y) \propto \prod_{i=1}^{N} N(y_i|w^T x_i, \sigma^2) \cdots N(0, \Sigma_p) \tag{117}$$

高斯分布取高斯先验的共轭分布依然是高斯分布，于是可以得到后验分布也是一个高斯分布。第一项：

$$\prod_{i=1}^{N} N(y_i|w^T x_i, \sigma^2) = \frac{1}{(2\pi)^{N/2}\sigma^N} \exp(-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - w^T x_i)^2) \tag{118}$$

$$= \frac{1}{(2\pi)^{N/2}\sigma^N} \exp(-\frac{1}{2}(Y - Xw)^T(\sigma^{-2}I)(Y - Xw)) \tag{119}$$

$$= N(Xw, \sigma^2 I) \tag{120}$$

代入上面的式子：

$$p(w|X, Y) \propto \exp(-\frac{1}{2\sigma^2}(Y - Xw)^T\sigma^{-2}I(W - Xw) - \frac{1}{2}w^T\sigma_p^{-1}w) \tag{121}$$

假定最后得到的高斯分布为：$N(\mu_w, \Sigma_w)$。对于上面的分布，采用配分的方式来得到最终的分布，指数上面的二次项为：

$$-\frac{1}{2\sigma^2}w^T X^T X w - \frac{1}{2}w^T\Sigma_p^{-1}w \tag{122}$$

于是：

$$\Sigma_w^{-1} = \sigma^{-2}X^T X + \Sigma_p^{-1} = A \tag{123}$$

一次项：

$$\frac{1}{2\sigma^2}2Y^T X w = \sigma^{-2}Y^T X w \tag{124}$$

于是：

$$\mu_w^T\Sigma_w^{-1} = \sigma^{-2}Y^T X \Rightarrow \mu_w = \sigma^{-2}A^{-1}X^T Y \tag{125}$$

6.2 预测

给定一个$x^\star$，求解$y^\star$，所以$f(x^\star)=x^{\star^T}w$，代入参数后验，有$x^{\star^T}w\sim N(x^{\star^T}\nu_w,x^{\star^T}\Sigma x^\star)$，添上噪声项：

$$p(y^\star|X,Y,x^\star)=\int_w p(y^\star|w,X,Y,x^\star)p(w|X,Y,x^\star)dw \tag{126}$$

$$=\int_w p(y^\star|w,x^\star)p(w|X,Y)dw=N(x^{\star^T}\mu_w,x^{\star^T}\Sigma_w x^\star+\sigma^2) \tag{127}$$

7 高斯过程回归

将一维高斯分布推广到多变量中就得到了高斯网络，将多变量推广到无限维，就得到了高斯过程，高斯过程是定义在连续域（时间空间）上的无限多个高维随机变量所组成的随机过程。在时间轴上的任意一个点都满足高斯分布，将这些点的集合叫做高斯过程的一个样本。

- 对于时间轴上的序列ξ_t，如果$\forall n\in\mathbb{N}^+.t_i\in T$，有$\xi_{t_1-t_n}\sim N(\mu_{t_1-t_n},\Sigma_{t_1-t_n})$，那么$\{\xi_t\}_{t\in T}$ 是一个高斯过程。
- 高斯过程有两个参数（高斯过程存在性定理），均值函数$m(t)=E[\xi_t]$和协方差函数$k(s,t)=E[(\xi_x-E[\xi_s])(\xi_t-E[\xi_t])]$。

我们将贝叶斯线性回归添加核技巧的这个模型叫做高斯过程回归，高斯过程回归分为两种视角：

- 权空间的视角-核贝叶斯线性回归，相当于x 为t，在每个时刻的贝斯分布来源于权重，根据上面的推导，预测的函数依然是高斯分布。
- 函数空间的视角-高斯分布通过函数$f(x)$来体现。

7.1 核贝叶斯线性回归

贝叶斯线性回归可以通过加入核函数的方法来解决非线性函数的问题，将$f(x)=x^Tw$这个函数变为$f(x)=\phi(x)^Tw$（当然这个时候，Σ_p也要变为更高维度的），变换到更高维的空间，有：

$$f(x^\star)\sim N(\phi(x^\star)^T\sigma^{-2}A^{-1}\Phi^TY,\phi(x^\star)^TA^{-1}\phi(x^\star)) \tag{128}$$

$$A=\sigma^{-2}\Phi^T\Phi+\Sigma_p^{-1} \tag{129}$$

其中$\Phi=(\phi(x_1),\phi(x_2),\dots,\phi(x_N))^T$。为了求解$A^{-1}$，可以利用Woodbury Formula，$A=\Sigma_p^{-1},C=\sigma^{-2}I$：

$$(A+UCV)^{-1}=A^{-1}-A^{-1}U(C^{-1}+VA^{-1}U)^{-1}VA^{-1} \tag{130}$$

所以$A^{-1}=\Sigma_p-\Sigma_p\Phi^T(\sigma^2I+\Phi\Sigma_p\Phi^T)^{-1}\Phi\Sigma_p$ 也可以用另一种方法：

$$A=\sigma^{-2}\Phi^T\Phi+\Sigma_p^{-1} \tag{131}$$

$$\Leftrightarrow A\Sigma_p=\sigma^{-2}\Phi^T\Phi\Sigma_p+I \tag{132}$$

$$\Leftrightarrow A\Sigma_p\Phi^T=\sigma^{-2}\Phi^T\Phi\Sigma_p\Sigma^T+\Phi^T=\sigma^{-2}\Phi^T(k+\sigma^2I) \tag{133}$$

$$\Leftrightarrow \Sigma_p\Phi^T=\sigma^{-2}A^{-1}\Phi^T(k+\sigma^2I) \tag{134}$$

$$\Leftrightarrow \sigma^{-2}A^{-1}\Phi^T=\Sigma_p\Phi^T(k+\sigma^2I)^{-1} \tag{135}$$

$$\Leftrightarrow \phi(x^\star)^T\sigma^{-2}A^{-1}\Phi^T=\phi(x^\star)^T\Sigma_p\Phi^T(k+\sigma^2I)^{-1} \tag{136}$$

上面的左边的式子就是变换后的均值，而右边的式子就是不含A^{-1}的式子，其中$k = \Phi\Sigma_p\Phi^T$。根据A^{-1}得到方差为：

$$\phi(x^\star)^T\Sigma_p\phi(x^\star) - \phi(x^\star)^T\Sigma_p\Phi^T(\sigma^2 I + k)^{-1}\Phi\Sigma_p\phi(x^\star) \tag{137}$$

上面定义了：

$$k = \Phi\Sigma\Phi^T \tag{138}$$

我们看到，在均值和方差中，含有下面四项：

$$\phi(x^\star)^T\Sigma_p\Phi^T, \phi(x^\star)^T\Sigma_p\phi(x^\star), \phi(x^\star)^T\Sigma_p\Phi^T, \Phi\Sigma_p\phi(x^\star) \tag{139}$$

展开后，可以看到，有共同的项：$k(x, x') = \phi(x)^T\Sigma_p\phi(x')$。由于$\Sigma_p$是正定对称的协方差矩阵，所以，这是一个核函数。

对于高斯过程中的协方差：

$$\begin{aligned} k(t,s) &= Cov[f(x), f(x')] && (140)\\ &= E[\phi(x)^T ww^T\phi(x')] && (141)\\ &= \phi(x)^T E[ww^T]\phi(x') && (142)\\ &= \phi(x)^T\Sigma_p\phi(x') && (143) \end{aligned}$$

我们可以看到，这个就对应着上面的核函数。因此我们看到$\{f(x)\}$组成的组合就是一个高斯过程。

7.2 函数空间的观点

相比权重空间，我们也可以直接关注f这个空间，对于预测任务，这就是类似于求：

$$p(y^\star|X, Y, x^\star) = \int_f p(y^\star|f, X, Y, x^\star)p(f|X, Y, x^\star)df \tag{144}$$

对于数据集来说，取$f(X) \sim N(\mu(X), k(X, X)), Y = f(X) + \epsilon \sim N(\mu(X), k(X, X) + \sigma^2 I)$。预测任务的目的是给定一个新数据序列$X^\star = (x_1^\star, \ldots, x_M^\star)^T$，得到$Y^\star = f(X^\star + \epsilon)$。我们可以写出：

$$\begin{pmatrix} Y \\ f(X^\star) \end{pmatrix} \sim N\left(\begin{pmatrix} \mu(X) \\ \mu(X^\star) \end{pmatrix} \begin{pmatrix} k(X,X) + \sigma^2 I & k(X, X^\star) \\ k(X^\star, X) & k(X^\star, X^\star) \end{pmatrix} \right) \tag{145}$$

根据高斯分布的方法：

$$x = \begin{pmatrix} x_a \\ x_b \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix} \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix} \right) x_b|x_a \sim N(\mu_{b|a}, \Sigma_{b|a}) \tag{146}$$

$$\mu_{b|a} = \Sigma_{ba}\Sigma_{aa}^{-1}(x_a - \mu_a) + \mu_b \tag{147}$$

$$\Sigma_{b|a} = \Sigma_{bb} - \Sigma_{ba}\Sigma_{aa}^{-1}\Sigma_{ab} \tag{148}$$

可以直接写出：

$$p(f(X^\star)|X, Y, X^\star) = p(f(X^\star)|Y) \tag{149}$$

$$N(k(X^\star, X)[k(X, X) + \sigma^2 I]^{-1}(Y - \mu(X)) + \mu(X^\star), k(X^\star, X^\star) - k(X^\star, X)[k(X, X) + \sigma^2 I]^{-1}k(X, X^\star)) \tag{150}$$

所以对于$Y = f(X^\star) + \epsilon$：

$$N(k(X^\star, X)[k(X, X) + \sigma^2 I]^{-1}(Y - \mu(X)) + \mu(X^\star), \tag{151}$$

$$k(X^\star, X^\star) - k(X^\star, X)[k(X, X) + \sigma^2 I]^{-1}k(X, X^\star) + \sigma^2 I) \tag{152}$$

我们看到，函数空间的观点更加简单易于求解。

References