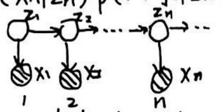
HMM $\theta=(\pi,A,B)$

latent Z_n (时刻n)

$A=[a_{ij}]$ 由状态i转移j的概率

$B=[b_{jk}]$ 发射概率 观测变量的概率

$p(x_n|z_n)$ $p(x_n=k|z_n=j)=B_j(k)$



$\alpha_n(i)$ 时,模型处于i 且观测序列 $x_1,x_2\cdots x_n$ 已经发生的条件下 状态序列的概率

forward-backward

前 $\alpha_n(i)=P(x_1,\cdots,x_n,z_n=i|\theta)$

$\alpha_1(i)=\pi_i\cdot B_i(x_1)$ sum-product

$\alpha_{n+1}(j)=(\sum_{i=1}^{D}\alpha_n(i)a_{ij})B_j(x_{n+1})$

后 $\beta_n(i)=P(x_{n+1},x_{n+2}\cdots x_N|z_n=i,\theta)$

$\beta_N(i)=1$ (一定出现) max-product

$\beta_{n-1}(i)=\sum_{j=1}a_{ij}\cdot B_j(x_n)\cdot\beta_n(j)$

$p(z_n=i|X,\theta)=\frac{\alpha_n(i)\beta_n(i)}{P(X|\theta)}\rightarrow\sum_i\alpha_N(i)\beta_N(i)$

$P(z_{n+1},z_n|X,\theta)=\frac{\alpha_{n-1}(i)B_i(x_n)a_{ii}\beta_n(i)}{P(X|\theta)}$

MDP $<S,R,P,\gamma>$

state x, next x', action u, policy $\pi(u|x)$ (与t无关)

转移概率 $p(x'|u,x)$

reward $r(x,u)$ discount $\gamma\in(0,1)$

$R_T^{\pi}(x_t)=E[\sum_{\tau=0}^{T}\gamma^{\tau}r_{t+\tau}|u_{t+\tau}=\pi(x_{t+\tau})]$

目标 $\pi^*=\arg\max_{\pi}R_T^{\pi}(x_t)$

additive $R(x_t,x_{t+1}\cdots)=r(x_t)+\gamma r(x_{t+1})+\gamma^2 r(x_{t+2})+\cdots$

state value $V(x)$ 从x状态开始R的期望

$V(x)=E_\pi[\sum_{\tau=0}^{\infty}\gamma^{\tau}R_{t+\tau}|x_t=x]=r(x,u)+\gamma\sum_{x'\in S}P(x'|x,u)V(x')$

$Q_\pi(s,u)=E_\pi[r_{t+1}+\gamma V(x_{t+1},u_{t+1})|x,u]$

$P^\pi(x'|x)=\sum_{u\in A}\pi(u|x)P(x'|x,u)$

$r^\pi(x)=\sum_{u\in A}\pi(u|x)r(u,x)$

$Q(s,u)=r(s,u)+\gamma\sum_{x'\in S}P(x'|s,u)\cdot V_\pi(x')$

$V_\pi(s)=\sum_{u\in A}\pi(u|s)q_\pi(s,u)$

$V_\pi(s)=\sum_{a\in A}\pi(a|s)(r(s,a)+\gamma\sum_{s'\in S}P(s'|s,a)\cdot V_\pi(s'))$

终点 $V=0$ $V_*(s)=\max V_\pi(s)$

$q_*(s,a)=r(s,a)+\gamma\sum_{s'\in S}p(s'|s,a)\max_{a'}q(s',a')$

监督: regression classification detection

KNN. SVM. DT. NN DNN

无监督: Clustering, data Dimension reduction

K-means. GMM. PCA. ICA. NMF. GAN

K-Means

1.随机找K个点 2.把每个点分给最近的中心 3.重新计算中心点

目标 $J=\sum_{n=1}^{N}\sum_{k=1}^{K}r_{nk}\|x_n-\mu_k\|^2$

I. 固定μ_k min J $r_{nk}\to$ 0 or 1   II. 固定r_{nk} min μ_k

$\mu_k=\frac{\sum_n r_{nk}x_n}{\sum_n r_{nk}}$

GMM $p(x)=\sum_{k=1}^{K}\pi_k N(x|\mu_k,\Sigma_k)$ $\pi_k\to$ 混合系数

对每个观测数据x_n都有latent z_n 聚类标签

E: $\gamma(z_{nk})=\frac{\pi_k N(x_n|\mu_k,\Sigma_k)}{\sum_j\pi_j N(x_n|\mu_j,\Sigma_j)}=P(z_k=1|x_n)$

指定x_n属于第k个高斯分布的概率

$\ln p(X|\pi,\mu,\Sigma)=\sum_{n=1}^{N}\ln\{\sum_{k=1}^{K}\pi_k N(x_n|\mu_k,\Sigma_k)\}$

↳ maximum 求导

$\mu_k^{new}=\frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})x_n$ $N_k=\sum_{n=1}^{N}\gamma(z_{nk})$

$\Sigma_k^{new}=\frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})(x_n-\mu_k^{new})(x_n-\mu_k^{new})^T$

$\pi_k^{new}=\frac{N_k}{N}$

complete data $\{X,Z\}$

$p(X,Z|\mu,\Sigma,\pi)=\prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k^{z_{nk}}N(x_n|\mu_k,\Sigma_k)^{z_{nk}}$

$E(z_{nk})=\gamma(z_{nk})$

$E_Z[\ln p(X,Z|\mu,\Sigma,\pi)]=\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{nk})\{\ln\pi_k+\ln N(x_n|\mu_k,\Sigma_k)\}$

Mixture of Bernoulli

$p(x|z,\mu)=\prod_{k=1}^{K}p(x|\mu_k)^{z_k}$, $p(z|\pi)=\prod_{k=1}^{K}\pi_k^{z_k}$

$\ln p(X|\mu,\pi)=\sum_{n=1}^{N}\ln\{\sum_{k=1}^{K}\pi_k p(x_n|\mu_k)\}$

$\ln p(X,Z|\mu,\pi)=\sum_{n=1}^{N}\sum_{k=1}^{K}z_{nk}\{\ln\pi_k+\sum_{i=1}^{D}[x_{ni}\ln\mu_{ki}+(1-x_{ni})\ln(1-\mu_{ki})]\}$

E: $\gamma(z_{nk})=E(z_{nk})=\frac{\sum_{z_{nk}}z_{nk}(\pi_k p(x_n|\mu_k))^{z_{nk}}}{\sum_{z_{nj}}[\pi_j p(x_n|\mu_j)]^{z_{nj}}}$

$=\frac{\pi_k p(x_n|\mu_k)}{\sum_{j=1}^{K}\pi_j p(x_n|\mu_j)}$

$E_Z[\ln p(X,Z|\mu,\pi)]=\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma(z_{nk})\{\cdots\}$

M: $N_k=\sum_{n=1}^{N}\gamma(z_{nk})$, $\pi_k=\frac{N_k}{N}$

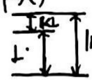
$\bar{x}_k=\frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})x_n$ $\mu_k=\bar{x}_k$

Ⓔ 隐变量后验 $\gamma(z)$

Ⓜ 更新模型参数(根据E())

EM通用算法 求 $\ln p(X|\theta)$

$\ln p(X|\theta)=L(q,\theta)+KL(q\|p)$ (非负)



$L(q,\theta)=\sum_Z q(Z)\ln(\frac{p(X,Z|\theta)}{q(Z)})$

$KL(q\|p)=-\sum_Z q(Z)\ln(\frac{p(Z|X,\theta)}{q(Z)})\geq 0$

两个概率分布差异

⇒最大化$L(q,\theta)$ → 对latent变量后验评估下界

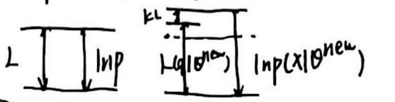
E: Fix θ^{old} $L(q,\theta^{old})$ max ⇐ $KL(q\|p)=0$ ($p(Z|X,\theta^{old})=q(Z)$)

M: Fix $q(Z)$. $L(q,\theta^{old})$ max ⇐ θ更新使得KL>0

$L(q,\theta)=\sum q(Z)\ln p(X,Z|\theta)-\sum_Z q(Z)\ln q(Z)$

其中 $q(Z)=p(Z|X,\theta^{old})$

$\Rightarrow L(q,\theta)=Q(\theta,\theta^{old})+const$



Ex EM for Linear Regression

$\ln p(t|\alpha,\beta)=\ln p(t|w,\beta)+\ln p(w|\alpha)$

E → $E[\ln p(t,w|\alpha,\beta)]=\frac{M}{2}\ln(\frac{\alpha}{2\pi})-\frac{\alpha}{2}E(w^Tw)+\frac{N}{2}\ln(\frac{\beta}{2\pi})-\frac{\beta}{2}\sum_{n=1}^{N}E[(t_n-w^T\phi_n)^2]$

M → $\alpha=\frac{M}{E(w^Tw)}=\frac{M}{m_N^Tm_N+Tr(S_N)}$

$\frac{1}{\beta}=\frac{1}{N}\sum_{i=1}^{N}(t_n-m_N^T\phi_n)^2$

补 MDP 求解

Bellman $V^*(s)=\max_a\sum_{s'}P(s'|s,a)[R(s,a,s')+\gamma V^*(s')]$

$\pi^*(s)=\arg\max V^*(s)$

① value iteration

$V^T(s')=r(s,a)+V^{T-1}(s')p(s'|s,a)$ 选择方向最大的

② policy iteration

$V^{\pi^T}\to\pi'$

HMM + EM

E-step $\alpha_t(z)=\sum_{z'}\alpha_{t-1}(z')A_{z'z}p(x_t|\theta_z)$

$\beta_t(z)=\sum_{z'}A_{zz'}p(x_{t+1}|\theta_{z'})\beta_{t+1}(z')$

状态边缘概率

$\gamma_t(z)=p(z_t=z|X,\theta)\propto\alpha_t(z)\beta_t(z)$

状态转移边缘概率

$\xi_t(z,z')=p(z_t=z,z_{t+1}=z'|X,\theta)\propto\alpha_t(z)A_{zz'}p(x_{t+1}|\theta_{z'})\beta_{t+1}(z')$

M-step: 更新模型参数 $\theta=\{\pi,A,\theta_z\}$

$\pi(z)=\gamma_1(z)$

$A_{zz'}=\frac{\sum_{t=1}^{T-1}\xi_t(z,z')}{\sum_{t=1}^{T}\gamma_t(z)}$

Ex. $p(x|\theta_i)=\theta_i^x(1-\theta_i)^{1-x}\Rightarrow\theta_i$

$p(x,z|\theta)=\prod_{i=1}^{k}[p(z_i)p(x|\theta_i)]^{z_i}$

M: $Q(\theta|\theta^{old})=E_{Z|X,\theta^{old}}[\log p(X,Z|\theta)]$

对θ求导

MAP+EM

E一样

M. $Q_{map}(\theta|\theta^{old})=E[\log p(X,Z|\theta)]+\ln p(\theta)$

SVM $y(x)=w^T\phi(x)+b$

$t_n y(x_n)>1$

dis(point→line) $=\frac{t_n y(x_n)}{\|w\|}$

目标 $\min\frac{\|w\|^2}{2}$ s.t. $t_n(w^T\phi+b)\geq 1$

Lagrange

$L(w,b,a)=\frac{\|w\|^2}{2}-\sum_{n=1}^{N}a_n(t_n(w^T\phi(x_n)+b)-1)$

导⇒ $w=\sum_{n=1}^{N}a_nt_n\phi(x_n)$ $\sum_{n=1}^{N}a_nt_n=0$

Dual $\tilde{L}(a)=\sum_{n=1}^{N}a_n-\frac{1}{2}\sum_n\sum_m a_na_mt_nt_mk(x_n,x_m)$

s.t. $a_n\geq 0$ $\sum_{n=1}^{N}a_nt_n=0$ $\phi(x_n)^T\phi(x_m)$

quadratic programming.

solve a_n

Max Margin Classifier $\lambda>0$

$\arg\min\sum_n E_\infty(y(x_n)t_n-1)+\lambda\|w\|^2$

↳ $\begin{cases}0 & \text{if } z\geq 0\\ \infty & \text{else}\end{cases}$

Soft / overlap

~~t_ny~~ $t_ny(x_n)\geq 1-\varepsilon_n$

$\min\frac{\|w\|^2}{2}+C\sum_{n=1}^{N}\varepsilon_n$

$L(w,b,a)=\frac{\|w\|^2}{2}+C\sum\varepsilon_n-\sum a_n(t_ny(x_n)-1+\varepsilon_n)-\sum\mu_n\varepsilon_n$

$\frac{\partial L}{\partial w}=0\Rightarrow w=\sum a_nt_n\phi(x_n)$

$\frac{\partial L}{\partial b}=0\Rightarrow\sum a_nt_n=0$

$\frac{\partial L}{\partial\varepsilon_n}=0\Rightarrow a_n=C-\mu_n$

KTT Condition

$\mu_n\geq 0$ $\varepsilon_n\geq 0$ $\mu_n\varepsilon_n=0$

$a_n\geq 0$ $t_ny(x_n)\geq 1-\varepsilon_n$ $a_n(t_ny(x_n)-1+\varepsilon_n)=0$

$\frac{\partial L}{\partial w}=\frac{\partial L}{\partial b}=\frac{\partial L}{\partial\varepsilon_n}=0$

NN

error funct: & cross entropy 分类

$E(w) = -\sum_{n=1}^{N}(t_n \ln y_n + (1-t_n)\ln(1-y_n))$

$y = \sigma(a)$ sigmoid

regression $y = a$

$E(w) = \frac{1}{2}\sum_{n=1}^{N}(y(x_n, w) - t_n)^2$

multiclass

$E(w) = -\sum_n \sum_k t_{kn} \ln y_k(x_n, w)$ $\to \frac{\exp(a_k(x,w))}{\sum_j \exp(a_j(x,w))}$ softmax

$w^{\tau+1} = w^{\tau} + \Delta w^{\tau} = w^{\tau} - \eta \nabla E(w^{\tau})$

$z_i \xrightarrow{w_{ji}} a_j\ z_j \to a_k$ $E = \frac{1}{2}\sum(y_k - t_k)^2$

$a_j = \sum_i w_{ji} z_i$ $z_j = h(a_j)$ $\frac{\partial a_j}{\partial w_{ji}} = z_i$

$a_k = \sum_j w_{kj} z_j$ $y_k = a_k$

$\frac{\partial E_n}{\partial a_j} = \sum_k \frac{\partial E_n}{\partial a_k}\frac{\partial a_k}{\partial a_j} = \sum_k (y_k - t_k) w_{kj} \cdot h'(a_j)$

$E_n = -\sum_k t_k \ln y_k + (1-t_k)\ln(1-y_k)$

$\frac{\partial E_n}{\partial y_k} = \frac{y_k - t_k}{y_k(1-y_k)}$ $\frac{\partial y_k}{\partial a_k} y_k(1-y_k), y_k = \sigma(a_k)$

$\frac{\partial E_n}{\partial w_{kj}} = \delta_k z_j$ $\frac{\partial E_n}{\partial w_{ji}} = \delta_j z_i$

$H = \nabla\nabla E = \sum_{n=1}^{N} \nabla y_n \nabla y_n + \sum_{n=1}^{N}(y_n - t_n)\nabla\nabla y_n$

NN-regression

$E(w) = -\ln p(w|t) = \frac{\alpha}{2}w^T w + \frac{\beta}{2}\sum_{n=1}^{N}(y(x_n, w) - t_n)^2 + C$

$\nabla E(w) = \alpha w + \beta \sum_{n=1}^{N}(y_n - t_n)\nabla_w y(x_n w)$

$A = \nabla\nabla E(w) = \alpha I + \beta H$

$w_{MAP} \leftarrow w^{new} = w^{old} - A^{-1}\nabla E(w)$

$q(w) = N(w|w_{MAP}, A^{-1})$

$y(x,w) \simeq y(x, w_{MAP}) + g^T_{MAP}(w - w_{MAP})$

$p(t|x, w, \beta) = N(t|y(x,w), \beta^{-1})$

$p(t|x, D, \alpha, \beta) = N(t|y(x, w_{MAP}), g^T_{MAP}A^{-1}g_{MAP} + \beta^{-1})$

$g = \nabla_w y(x, w)$

NN-classification

$E(w) = -\ln p(w|t) = \frac{\alpha}{2}w^T w - \sum_{n=1}^{N}(t_n \ln y_n - (1-t_n)\ln(1-y_n))$

$\nabla_w E = \alpha w + \sum_{n=1}^{N}(y_n - t_n) g_n$

$A = \nabla\nabla E(w) = \alpha I + H$

$w_{MAP} \leftarrow w^{new} = w^{old} - A^{-1}\nabla E(w)$

$p(t|x, D) = \int p(t|x, w) q(w|D) dw$

预 $p(t|x, w_{MAP})$ $y(x,w) \simeq y_{MAP}(x) + g^T(w - w_{MAP})$

ML Logistic regression

$p(C_0|\phi) = y(\phi) = \sigma(w^T\phi), p(C_1|\phi) = 1 - p(C_0|\phi)$

$p(t|w) = \prod_{n=1}^{N} y_n^{t_n}(1-y_n)^{1-t_n}$

$E(w) = -\ln p(t|w) = -\sum_{n=1}^{N}(t_n \ln y_n + (1-t_n)\ln(1-y_n))$

$\nabla E(w) = \sum_{n=1}^{N}(y_n - t_n)\phi_n$ $H = \nabla\nabla E = \sum_{n=1}^{N} y_n(1-y_n)\phi_n\phi_n^T$

$w_{ML} \leftarrow w^{new} = w^{old} - H^{-1}\nabla E(w)$

$q(w) = N(w|w_{ML}, H^{-1})$

预 $E(t) = y$ $var(t) = y(1-y)$

$E(w^T\phi) = w_{ML}\phi$ $var(w^T\phi) = \phi^T H^{-1}\phi$

Evaluation

$p(D) = \int p(D|\theta)p(\theta)d\theta$

$\ln p(D) = \ln p(D|\theta_{MAP}) + \ln p(\theta_{MAP}) + \frac{M}{2}\ln(2\pi)$

where $A = -\nabla\nabla \ln p(D|\theta_{MAP})p(\theta_{MAP})$ $-\frac{1}{2}\ln|A|$

$= -\nabla\nabla \ln p(\theta_{MAP}|D)$

MAP

$p(w) = N(w|m_0, S_0)$ $p(w|t) \propto p(w)p(t|w)$

$E(w) = -\ln p(w|t) = \frac{1}{2}(w - m_0)^T S_0^{-1}(w - m_0) - \sum_{n=1}^{N}(t_n \ln y_n + (1-t_n)\ln(1-y_n))$

$\nabla E(w) = S_0^{-1}(w - m_0) + \sum_{n=1}^{N}(y_n - t_n)\phi_n$

$H = \nabla\nabla E(w) = S_0^{-1} + \dots$

$w_{MAP} \leftarrow w^{new} = w^{old} - H^{-1}\nabla E(w)$

$q(w) = N(w|w_{MAP}, H^{-1})$

Discrimination

$y(x) = w^T x$ $E(w) = \frac{1}{2}(XW - T)^T(XW - T)$

$\tilde{w} = (X^T X)^{-1}X^T T$

Fisher

$y = w^T x$ $w \propto m_2 - m_1$

$s_1^2 = \sum_{n \in C_1}(y_n - m_1)^2$ $s_2^2 = \sum_{n \in C_2}(y_n - m_2)^2$

$J(w) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$ objective $= \frac{w^T S_B w}{w^T S_W w}$

$S_B = (m_2 - m_1)(m_2 - m_1)^T$

$S_W = \sum_{n \in C_1}(x_n - m_1)(x_n - m_1)^T + \sum_{n \in C_2}(x_n - m_2)(x_n - m_2)^T$

$w \propto S_W^{-1}(m_2 - m_1)$

learn $p(\theta|train, model) = p(t|m, \theta)p(\theta|m)$

prediction $p(test|train, model)$

$= \int p(test|M, \theta)\, p(\theta|train, M)\, d\theta$

evaluation $p(train|M) =$

$\int p(train|M, \theta)\, p(\theta|M)\, d\theta$

Reinforcement Learning for MDP

1. What is the Bellman equation? How to solve the Bellman equation?

$v_\pi(s) = E[r_{t+1} + \gamma v_\pi(S_{t+1})|S_t = s]$. Analytical sol: $V = (I - \gamma\mathcal{P})^{-1}R$ using Guassian, can also be solved by dynamic programming, Monte-Carlo method, and temporal difference method.

What are the differences between policy iteration and value iteration? What are their advantages and disadvantages respectively?

- **Policy iteration:** Policy evaluation + policy enhancement. (Attach equations to explain) Policy evaluation in the policy iteration uses the Bellman expectation equation to obtain the state value function of a policy, which is a dynamic planning process; while the value iteration directly uses the Bellman optimization equation for dynamic planning to obtain the final optimal state value.
- **Value iteration:** (Attach equations to explain) Only one round of value updates in the policy evaluation, then do policy enhancements based directly on the updated values. Finally use T-policy (refer to MDP) to obtain the optimal policy.

Policy Iteration	Value Iteration
Complex algorithm	Simple algorithm
Cheaper to compute	Expensive to compute
Faster convergence	Slower convergence

3. What is the model-free reinforcement learning? How to achieve the mode-free reinforcement learning? Please use specific examples to illustrate your points.

Methods that agent can only learn through data that comes from interacting with the enviroment is called model-free reinforcement learning (Sarsa and Q-learning).

Application: Complex enviroment like E-sport games.

4. What are the differences between on-line and off-line RL? What are their advantages and disadvantages respectively? Please use specific examples to illustrate your points.

Online RL: Learning from the data that sampled from the environment by the current policy, and once the policy is updated, the data points will be deprecated. (E.g. Sarsa, policy is greedy)

Offline RL: Use experience recall pool to store and reuse the historical samples(E.g. Q-learning). Offline RL can better make use of historical data and obtain lower complexity of sample, i.e. use smaller amount of data to achieve convergence.

$\Phi = \begin{pmatrix} \varphi_0(x_1) & \varphi_1(x_1) & \dots \\ \varphi_0(x_2) & \dots \end{pmatrix}$

$tr(ABC) = tr(CAB) = tr(BCA)$

$\frac{\partial tr(AB)}{\partial A} = B^T$

$\frac{\partial tr(xx^T\Sigma^{-1})}{\partial \Sigma^{-1}} = xx^T$

$\frac{\partial \ln|A|}{\partial A} = (A^T)^{-1}$

$\frac{\partial \ln|\Sigma^{-1}|}{\partial \Sigma^{-1}} = \Sigma$

$y = Ax + v$ $v \to N(0, Q)$ $x \sim N(\mu, \Sigma)$

$x = m + u$ $u \to N(0, L)$ $p(x|y) = N(x|m, L)$

$\Rightarrow \begin{cases} L^{-1} = A^T Q^{-1} A + \Sigma^{-1} \\ Lm = A^T Q^{-1} y + \Sigma^{-1}\mu \end{cases}$

$p(y|x) = N(y|Ax, Q)$

$p(y') = \int p(y'|x)p(x|y)dx = N(y'|Am, ALA^T + Q)$

$p(y) = \int p(y|x)p(x)dx = N(y|A\mu, A\Sigma A^T + Q)$

$t = y(x, w) + \epsilon \to (0, \beta^{-1})$

$p(t|x, w, \beta) = N(t|y(x, w), \beta^{-1})$

likelihood $= \prod_n N(t_n|w^T\phi(x_n), \beta^{-1})$

$\ln p(t|w, \beta) = \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \beta E_D(w)$

$E_D(w) = \frac{1}{2}\sum(t_n - w\phi(x_n))^2$ $w_{ML} = (\Phi^T\Phi)^{-1}\Phi^T t$

$E_D(w) + \lambda E_W(w)$ $w = (\lambda I + \Phi^T\Phi)^{-1}\Phi^T t$

Bayesian Linear regression

$p(w) = N(w|m_0, S_0)$

$p(w|t) = N(w|m_N, S_N)$

$\begin{cases} S_N^{-1} m_N = \beta\Phi^T t + S_0^{-1} m_0 \\ S_N^{-1} = \beta\Phi^T\Phi + S_0^{-1} \end{cases}$

预 $p(t|t, \alpha, \beta)$

$= \int p(t|w, \beta)p(w|t, \alpha, \beta)dw$

$p(t|x, t, \alpha, \beta) = N(t|m_N^T\phi(x), \sigma_N^2(x))$

$\sigma_N^2(x) = \frac{1}{\beta} + \phi(x)^T S_N \phi(x)$

$\ln p(X|\mu, \Sigma) = -\frac{ND}{2}\ln(2\pi) - \frac{N}{2}\ln|\Sigma| - \frac{1}{2}\sum_{n=1}^{N}(x_n - \mu)^T\Sigma^{-1}(x_n - \mu)$

$N(x|\mu, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}}\frac{1}{|\Sigma|^{\frac{1}{2}}}\exp(-\frac{1}{2}(x - \mu)^T\Sigma^{-1}(x - \mu))$

cross-entropy 监. + 无

likelihood

ML与MAP考虑先验

EM局部 MDP stationary时间独立