## 矩阵求导 Matrix cookbook

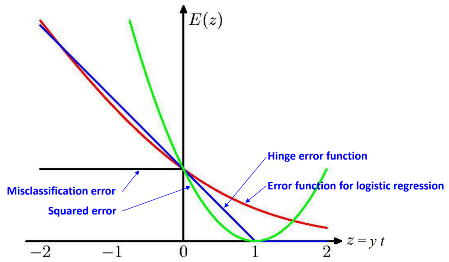
$\partial(\boldsymbol{X}^T) = (\partial\boldsymbol{X})^T, \partial(\boldsymbol{X}^{\text{-}1}) = -\boldsymbol{X}^{\text{-}1}(\partial\boldsymbol{X})\boldsymbol{X}^{\text{-}1}$

$\frac{\partial a^{\mathrm{T}}Xb}{\partial X} = ab^{\mathrm{T}}, \frac{\partial (Xb+c)^{\mathrm{T}}D(Xb+c)}{\partial X} = (D+D^{\mathrm{T}})(Xb+c)b^{\mathrm{T}}, \frac{\partial\,\mathrm{Tr}(AX^{\mathrm{T}}B)}{\partial X} = BA$

$\frac{\partial\,\mathrm{Tr}(AXBX^{\mathrm{T}}C)}{\partial X} = A^{\mathrm{T}}C^{\mathrm{T}}XB^{\mathrm{T}} + CAXB, \frac{\partial a^{\mathrm{T}}X^{-1}b}{\partial X} = -X^{-\mathrm{T}}ab^{\mathrm{T}}X^{-\mathrm{T}}$

## 误差函数 Error Function

| Name | Equation | Application |
|---|---|---|
| MAE/L1 Loss | $\frac{1}{N}\sum\lvert y_i - \hat{y}_i\rvert$ | / |
| MSE/L2 Loss | $\frac{1}{N}\sum(y_i - \hat{y}_i)^2$ | Linear R |
| Cross-Entropy | $\sum y_i \ln(y_i) + (1-y_i)\ln(y_i)$ | Logistic R/C |
| Hinge Loss | $\max(0, 1-\hat{y}y)$ | SVM |
| KL Divergence | $D_{\mathrm{KL}}(p\Vert q) = -\sum_x p(x)\ln\frac{q(x)}{p(x)}$ | EM |
| Neg Likelihood | $-\ln p$ | Any |




## 多元高斯分布 Multivariate Gaussian

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

$\ln p(\boldsymbol{x}|\mu,\sigma^2) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi)$

$\ln p(\boldsymbol{t}|\boldsymbol{x},\boldsymbol{w},\beta) = -\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n,\boldsymbol{w})-t_n\}^2 + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi)$

## 神经网络回归最大后验 Neural Networks Regression MAP

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

$\nabla E(\boldsymbol{w}) = \alpha\boldsymbol{w} + \beta\sum_{n=1}^{N}(y_n-t_n)\nabla_{\boldsymbol{w}}y(\boldsymbol{x},\boldsymbol{w})_n$

$\boldsymbol{A} = \nabla\nabla E(\boldsymbol{w}) = \alpha\boldsymbol{I} + \beta\boldsymbol{H}, \boldsymbol{H}$: Hessian of Sum of Error function

$\boldsymbol{w}_{\mathrm{MAP}} \leftarrow \boldsymbol{w}^{\mathrm{new}} = \boldsymbol{w}^{\mathrm{old}} - \boldsymbol{A}^{-1}\nabla E(\boldsymbol{w}), q(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_{\mathrm{MAP}},\boldsymbol{A}^{-1})$

$y(\boldsymbol{x},\boldsymbol{w}) \simeq y(\boldsymbol{x},\boldsymbol{w}_{\mathrm{MAP}}) + \boldsymbol{g}_{\mathrm{MAP}}^T(\boldsymbol{w}-\boldsymbol{w}_{\mathrm{MAP}})$

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

$p(t|\boldsymbol{x},D,\alpha,\beta) = \mathcal{N}(t|y(\boldsymbol{x},\boldsymbol{w}_{\mathrm{MAP}}),\boldsymbol{g}_{\mathrm{MAP}}^T\boldsymbol{A}^{-1}\boldsymbol{g}_{\mathrm{MAP}}+\beta^{-1})$

$\boldsymbol{g} = \nabla_{\boldsymbol{w}}y(\boldsymbol{x},\boldsymbol{w})$

## 神经网络分类最大后验 Neural Networks Classification MAP

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

$\nabla E(w) = \alpha w + \sum_{n=1}^{N}(y_n-t_n)\boldsymbol{g}_n, \boldsymbol{A} = \nabla\nabla E(\boldsymbol{w}) = \alpha\boldsymbol{I} + \boldsymbol{H}$

$w_{\mathrm{MAP}} \leftarrow w^{\mathrm{new}} = w^{\mathrm{old}} - A^{-1}\nabla E(w), q(w) = \mathcal{N}(w|w_{\mathrm{MAP}},A^{-1})$

$p(t|\boldsymbol{x},\mathcal{D}) \simeq p(t|\boldsymbol{x},\boldsymbol{w}_{\mathrm{MAP}}), y(\boldsymbol{x},\boldsymbol{w}) \simeq y_{\mathrm{MAP}}(\boldsymbol{x}) + \boldsymbol{g}^T(\boldsymbol{w}-\boldsymbol{w}_{\mathrm{MAP}})$

## 最大边界分类器 Maximum Margin Classifier

Origin: $\mathrm{argmax}_{w,b}\left\{\frac{1}{\|w\|}\min_n[t_n(w^{\mathrm{T}}\phi(x_n)+b)]\right\}$

Dual: $\mathrm{argmin}_{w,b}\frac{1}{2}\|w\|^2$ subject to $t_n(w^{\mathrm{T}}\phi(x_n)+b) \ge 1, n = 1\ldots N$

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

Constraints: $a_n = 0$ or $a_n \ne 0 \cap t_n(w^{\mathrm{T}}\phi(x_n)+b) = 1$

Let $\frac{\partial L}{\partial w} = \frac{\partial L}{\partial b} = 0, w = \sum_{n=1}^{N}a_nt_n\phi(x_n), \sum_{n=1}^{N}a_nt_n = 0$

Classifier: $y(x) = \sum_{n=1}^{N}a_nt_nk(x,x_n) + b, k(x,x_n) = \phi(x)^{\mathrm{T}}\phi(x_n)$

quadratic: $\tilde{L}(a) = \sum_{n=1}^{N}a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}a_na_mt_nt_mk(x_n,x_m), \sum_{n=1}^{N}a_nt_n = 0$

## 软边界分类器 Soft Margin Classifier

$t_ny(x_n) \ge 1-\xi_n$. Minimize: $C\sum_{n=1}^{N}\xi_n + \frac{1}{2}\|w\|^2$.

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

KKT conditions: $\mu_n \ge 0, \xi_n \ge 0, \mu_n\xi_n = 0$

$a_n \ge 0, t_ny(x_n) \ge 1-\xi_n, a_n(t_ny(x_n)-1+\xi_n) = 0$

Let $\frac{\partial L}{\partial w} = \frac{\partial L}{\partial b} = \frac{\partial L}{\partial \xi_n} = 0,$

$w = \sum_{n=1}^{N}a_nt_n\phi(x_n), \sum_{n=1}^{N}a_nt_n = 0, a_n = C-\mu_n$

## $\nu$-支持向量机 $\nu$-SVM

$\tilde{L}(a) = \sum_{n=1}^{N}a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}a_na_mt_nt_mk(x_n,x_m), \sum_{n=1}^{N}a_nt_n = 0$

subject to $0 \le a_n \le \frac{1}{N}, \sum_{n=1}^{N}a_nt_n = 0, \sum_{n=1}^{N}a_n \ge \nu$

## 相关向量机 Relevance Vector Machine

$p(t|x,w,\beta) = \mathcal{N}(t|y(x),\beta^{-1}), y(x) = \sum_{n=1}^{N}w_nk(x,x_n) + b$

$p(t|X,w,\beta) = \prod_{n=1}^{N}p(t_n|x_n,w,\beta^{-1})$

$p(w|\alpha) = \prod_{n=1}^{N}\mathcal{N}(w_i|0,\alpha_i^{-1})$

## K-均值聚类 K-means

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

## 高斯混合模型 Gaussian Mixture Model

PDF: $p(x) = \sum_{k=1}^{K}\pi_k\mathcal{N}(x|\mu_k,\Sigma_k)$

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

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

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

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

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

log-likelihood: $\ln p(X|\mu,\Sigma,\pi) = \sum_{n=1}^{N}\ln\left\{\sum_{k=1}^{K}\pi_k\mathcal{N}(x|\mu_k,\Sigma_k)\right\}$

## 伯努利混合模型 Mixtures of Bernoulli Distributions

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

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

**E:** $\gamma(z_{nk}) = \mathbb{E}[z_{nk}] = \frac{\pi_kp(x_n|\mu_k)}{\sum_{j=1}^{K}\pi_jp(x_n|\mu_j)}$

**M:** $N_k = \sum_{n=1}^{N}\gamma(z_{nk}), \pi_k = \frac{N_k}{N}, \mu_k = \overline{x}_k = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(z_{nk})x_n$

## 通用 EM 算法 EM Algorithm in General

Difficult to optimize $p(X|\theta)$, easy to optimize $p(X,Z|\theta)$.

$\ln p(X|\theta) = \mathcal{L}(q,\theta) + \mathrm{KL}(q\Vert p) = \sum_Z q(Z)\ln\frac{p(X,Z|\theta)}{q(Z)} - \sum_Z q(Z)\ln\frac{p(Z|X,\theta)}{q(Z)}$

$\mathrm{KL}(q\Vert p) \ge 0 \Rightarrow \mathcal{L}(q,\theta) \le \ln p(X|\theta)$

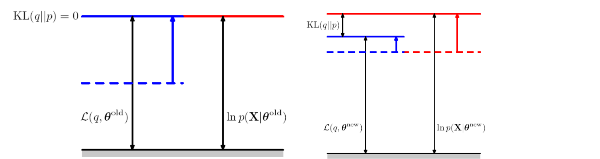
**E:** Fix $\theta^{\mathrm{old}}$, $\mathcal{L}(q,\theta^{\mathrm{old}})$ maximized when $\mathrm{KL}(q\Vert p) = 0, q(Z) = p(Z|X,\theta^{\mathrm{old}})$

**M:** Fix $q(Z)$, $\mathcal{L}(q,\theta^{\mathrm{old}})$ maximized by updating $\theta$ making $\mathrm{KL}(q\Vert p) > 0$

Q function:$Q(\theta,\theta^{\mathrm{old}}) = \sum_Z p(Z|X,\theta^{\mathrm{old}})\ln p(X,Z|\theta)$

$\theta^{\mathrm{new}} = \mathrm{argmax}_\theta Q(\theta,\theta^{\mathrm{old}})$

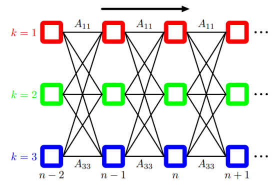
$$\begin{aligned}\mathcal{L}(q,\boldsymbol{\theta}) &= \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{\mathrm{old}})\ln p(\mathbf{X},\mathbf{Z}|\boldsymbol{\theta}) - \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{\mathrm{old}})\ln p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{\mathrm{old}})\\ &= \mathcal{Q}(\boldsymbol{\theta},\boldsymbol{\theta}^{\mathrm{old}}) + \mathrm{const} \quad \Longleftarrow \boxed{q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{\mathrm{old}})}\end{aligned}$$




## 隐马尔可夫模型 Hidden Markov Model

PDF: $p(\{x_N\},\{z_N\}) = p(z_1)\left[\prod_{n=2}^{N}p(z_n|z_{n-1})\right]\prod_{n=1}^{N}p(x_n|z_n)$

$p(z_n = k|z_{n-1} = j) = A_{jk}, p(x_t|z_n = k) = \mathcal{N}(x_t|\mu_k,\Sigma_k), p(z_1 = k) = \pi_k$




Latent State Lattice
transitions between latent states
col: one of $z_n$
row: one state of $z_n$

**E:** forward: $\alpha(z_n) = p(x_{1\dots N}, z_n) = p(x_n|z_n)\sum_{z_{n-1}} \alpha(z_{n-1})p(z_n|z_{n-1})$

back: $\beta(z_n) = p(x_{n+1\dots N}|z_n) = \sum_{z_{n+1}} \beta(z_{n+1})p(x_{n+1}|z_{n+1})p(z_{n+1}|z_n)$

$\gamma(z_{nk}) = \sum_z \gamma(z) z_{nk}$

$\xi(z_{n-1,j}, z_{nk}) = \sum_z \gamma(z) z_{n-1,j} z_{nk}$

$\gamma(z_n) = p(z_n|\mathbf{X}) = \frac{p(\mathbf{X}|z_n)p(z_n)}{p(\mathbf{X})} = \frac{\alpha(z_n)\beta(z_n)}{p(\mathbf{X})}, p(\mathbf{X}) = \sum_{z_n} \alpha(z_n)\beta(z_n)$

$x = \xi(z_{n-1}, z_n) = \frac{\alpha(z_{n-1})p(\mathbf{x}_n|z_n)p(z_n|z_{n-1})\beta(z_n)}{p(\mathbf{X})}$

$p(\mathbf{X}) = \sum_{z_n} \alpha(z_n)\beta(z_n) = \sum_{z_N} \alpha(z_N)$

**M:** $Q(\theta, \theta^{\text{old}}) = \sum_Z p(Z|X, \theta^{\text{old}}) \ln p(X, Z|\theta) = \sum_{k=1}^K \gamma(z_{1k}) \ln \pi_k + \sum_{n=2}^N \sum_{j=1}^K \sum_{k=1}^K \xi(z_{n-1,j}, z_{nk}) \ln A_{jk} + \sum_{n=2}^N \sum_{k=1}^K \gamma(z_{nk}) \ln p(x_n|\phi_k)$

$\pi_k = \frac{\gamma(z_{1k})}{\sum_{j=1}^K \gamma(z_{1j})}, A_{jk} = \frac{\sum_{n=2}^N \xi(z_{n-1,j}, z_{nk})}{\sum_{l=1}^K \sum_{n=2}^N \xi(z_{n-1,j}, z_{nl})}, \mu_k = \frac{\sum_{n=1}^N \gamma(z_{nk}) x_n}{\sum_{n=1}^N \gamma(z_{nk})}$

$\Sigma_k = \frac{\sum_{n=1}^N \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^{\mathrm{T}}}{\sum_{n=1}^N \gamma(z_{nk})}$ or $B_{km} = \frac{\sum_{n=1}^N \gamma(z_{nk})|_{x_n = m}}{\sum_{l=1}^K \sum_{n=1}^N \gamma(z_{nl})|_{x_n = m}}$

Trans $\begin{bmatrix} .6 & .3 \\ .4 & .7 \end{bmatrix}$, Emit $\begin{bmatrix} .8 & .1 \\ .2 & .9 \end{bmatrix}$, $\pi = [.5, .5], Z : \{\text{bull, bear}\}, X : \{\text{up, up, down}\}$

Evaluation: Sum-Product

$\alpha(z_1) = \begin{bmatrix} .8 \\ .1 \end{bmatrix} \circ \begin{bmatrix} .5 \\ .5 \end{bmatrix} = \begin{bmatrix} .4 \\ .05 \end{bmatrix}, \alpha(z_2) = \begin{bmatrix} .8 \\ .1 \end{bmatrix} \circ \begin{bmatrix} .6\times.4+.3\times.05 \\ .4\times.4+.7\times.05 \end{bmatrix} = \begin{bmatrix} .204 \\ .0195 \end{bmatrix}$

$\beta(z_3) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \beta(z_2) = \begin{bmatrix} .2\times.6\times1+.9\times.4\times1 \\ .2\times.3\times1+.9\times.7\times1 \end{bmatrix} = \begin{bmatrix} .48 \\ .69 \end{bmatrix}, \beta(z_1) = \begin{bmatrix} .8\times.6\times.48+.1\times.4\times.69 \\ .8\times.3\times.48+.1\times.7\times.69 \end{bmatrix}$

Decoding: Max-Product

$\delta(z_1) = \begin{bmatrix} .8 \\ .1 \end{bmatrix} \circ \begin{bmatrix} .5 \\ .5 \end{bmatrix}, \delta(z_2) = \begin{bmatrix} .8 \\ .1 \end{bmatrix} \circ \begin{bmatrix} .4\times.6 \\ .4\times.4 \end{bmatrix}, \delta(z_3) = \begin{bmatrix} .2 \\ .9 \end{bmatrix} \circ \begin{bmatrix} .192\times.6 \\ .192\times.4 \end{bmatrix}$

Predict: $p(x_{N+1}|X) = \frac{1}{p(X)} \sum_{z_{N+1}} p(x_{N+1}|z_{N+1}) \sum_{z_N} p(z_{N+1}|z_N)\alpha(z_N)$

## 马尔可夫决策过程 Markov Decision Process

T-policy: $\pi_T(x) = \text{argmax}_u [r(x, u) + \int V_{T-1}(x')p(x'|u, x)\ \mathrm{d}x']$

T-value: $V_T(x) = \gamma \max_u [r(x, u) + \int V_{T-1}(x')p(x'|u, x)\ \mathrm{d}x']$

Value iteration: $\hat{V} \leftarrow \gamma \max_u \left[r(x, u) + \int \hat{V}_{T-1}(x')p(x'|u, x)\ \mathrm{d}x'\right]$

Policy: $R_T^\pi(x_t) = E\left[\sum_{\tau=1}^T \gamma^\tau r_{t+\tau} | u_{t+\tau} = \pi(z_{1:t+\tau-1} u_{1:t+\tau-1})\right]$

Bellman equation: Optimize $v_*(s) = \max_{a \in \mathcal{A}} q_*(s, a)$

$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(s, a) q_\pi(s, a)$

$v_\pi(s) = E[r_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s]$

$q_\pi(s, a) = E[r_{t+1} + \gamma v_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a]$

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

## 常用分布 Distributions

Beta: $\text{Beta}(\mu|a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \mu^{a-1} (1-\mu)^{b-1}$

Dirichlet: $\text{Dir}(\mu|\alpha) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\dots\Gamma(\alpha_K)} \prod_{k=1}^K \mu_k^{\alpha_k - 1}, \alpha_0 = \sum_{k=1}^K \alpha_k$

Student-t: $\text{St}(x|\mu, \lambda, \nu) = \frac{\Gamma(\frac{\nu}{2}+\frac{1}{2})}{\Gamma(\frac{\nu}{2})} \left(\frac{\lambda}{\pi\nu}\right)^{\frac{1}{2}} \left(1 + \frac{\lambda(x-\mu)^2}{\nu}\right)^{-\frac{\nu}{2}-\frac{1}{2}}$

Gamma: $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} \mathrm{d}t$

## 计算题 Numerical Calculation

1. **Linear Regression for Least Square**:

$y = ax + b$ and three points $\{(1, 0), (3, 3), (5, 4)\}$ using pseudo-inverse.

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 3 & 1 \\ 5 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 3 \\ 4 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 \\ 3 & 1 \\ 5 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 4 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 3 & 5 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 3 & 1 \\ 5 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 1 & 3 & 5 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 3 \\ 4 \end{bmatrix}$$

2. **Maximum Margin Classifier**:

$\mathcal{C}_1$={(1,2),(2,2)} and $\mathcal{C}_2$={(4,4),(5,6)} find support vectors and decision boundary function.

Support vectors are {(2,2),(4,4)} (can be decided by checking whether deletion will change the boudary), the decision boundary is trivial.

3. **Clustering**:

{(1,2),(2,2),(4,4),(5,6)} for 2-class clustering

Randomly pick 2 center [(1.5,2) and (4.5,5.5), for this problem to achieve 1 iteration convergence] then calculate distance of each data point to assign the class. Clac mean coordinate of data points as the new center of cluster.

### 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: $\mathcal{V} = (I - \gamma\mathcal{P})^{-1} R$ 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.

## 复习课

$\ln p(X, Z|\theta) = \ln p(Z|X, \theta) + \ln p(X|\theta)$

EM:

$\ln p(x|\theta) = \ln(\sum p(x, z|\theta))$, z: latent

Jason inequivalence: $>= \sum_z p(z|x, \theta^{\text{old}}) \ln p(x, z|\theta) = Q(\theta, \theta^{\text{old}})$

$\theta^* = \text{argmax}\ Q(\theta, \theta^{\text{old}})$

KL divergence:

$\int q(z) \ln \frac{p(x,z|\ \theta)}{q(z)} dz - \int q(z) \ln \frac{p(z|x,\theta)}{q(z)} dz = \ln p(x|\theta)$

$\int q(z) \ln \frac{p(x,z|\theta)}{q(z)} dz = Q(\theta, \theta^{\text{old}}) + \text{const}$

non-convex EM, not converge to global maximum or may not converge

Policy iteration: $R_T^\pi(x_n) = \sum_{t=1}^T r(x_{n+t}, a_{n+t}), a_{n+t} = \pi(x_{n+t})$

# Contributors (in no particular order)

Site Fan, Jiachen Xiao, Kangrui Chen, Linghu Gong, Liang Kuang