机器学习

一. 基本概念与评估指标.

- 留出法 (hold-out): 直接划分 train-valid (若干次划分, 重复实验求平均)
- 交叉验证法 (cross-validation): K互斥子集划分, 每次取K-1个子集训练, 1个验证, 重复K次, 取均值. K=10! ← 留一法 (K=样本数, 准确但计算开销高)

真实 \ 预测	正	反
正	TP	FN
反	FP	TN

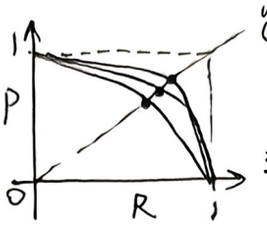
(假正例) 误判

混淆矩阵.

$$Precision = P = \frac{TP}{TP+FP}$$

$$Recall = R = \frac{TP}{TP+FN}.$$

P-R曲线



P=R: 平衡点.

平衡点可取值度量分类器的性能

$$\frac{1}{F1} = \frac{1}{P} + \frac{1}{R} \Rightarrow F1 = \frac{2PR}{P+R}. \quad F_\beta = \frac{(1+\beta^2)PR}{\beta^2 P + R}$$

$$\begin{cases} \beta > 1 \to R\uparrow & (\beta=\infty, F_\beta = R) \\ \beta < 1 \to P\uparrow. & (\beta=0, F_\beta = P) \end{cases}$$

[Definition] ML: 通过经验 (experience) 提升性能表现 (学习系统).

分类:
- 监督: label — classification / regression $x \longrightarrow y$
- 无监督: special info — clustering (interesting patterns)
- 强化: max reward — 环境交互. label exists, credit assignment.

二. 贝叶斯分类:

Naive Bayesian Classifier:

Given $X = (X_1, \cdots, X_p)^T$, predict ω. $\left(\max P(\omega|X) \propto P(X|\omega)P(\omega)\right)$.

Eg. X = (Refund = No, Married, Income = 120K)

1. compute $P(Refund = ?|No/Yes)$, $P(Marital\ status = ?|No/Yes)$, sample mean μ_{ij} & variance σ_{ij}^2 of Income|No/Yes.
2. compute $P(X|No) \overset{Independence}{=} P(Refund=No|No) \times P(Married|No) \times P(Income = 120k|No)$, $P(X|Yes)$

   $P(Income = 120k|No)$ → $\frac{1}{\sqrt{2\pi}\sigma_{ij}} \exp\left(-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}\right)$
3. compare $P(X|No)P(No)$ and $P(X|Yes)P(Yes)$
4. the bigger one is the output.

三. MLE

Regression model: $y = f(x,\omega) + \varepsilon$, $\varepsilon \sim N(0, \sigma^2)$. $p(y|x,\omega,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(y - f(x,\omega))^2}$

$$L(D,\omega,\sigma) = \prod_{i=1}^{n} p(y_i|x_i,\omega,\sigma), \quad \omega^* = \arg\max L$$

$$l(D,\omega,\sigma) = \log L = \sum_{i=1}^{n} \log p(y_i|x_i,\omega,\sigma) = -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y - f(x,\omega))^2 + C(\sigma).$$

$$l'(D,\omega,\sigma) = 0 \Leftrightarrow \sum_{i=1}^{n}(y - f(x,\omega))^2 = RSS(f) = 0. \quad J_n = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 = MSE$$

RSS: Residual Sum of Squares 残差平方和

四. 线性回归与分类.

(x_i, y_i), $f = \sum_{i=1}^{m} w_i x_i = w^T x$, $\omega = [w_1\ w_2 \cdots w_m]^T$.

$$MSE = J_n = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \omega^T x_i)^2 \Rightarrow J_n(\omega) = (y - X^T\omega)^T(y - X^T\omega)$$

$$\nabla J_n = -2X(y - X^T\omega) = 0 \Rightarrow Xy = XX^T\omega \Rightarrow \omega = (XX^T)^{-1}Xy.$$

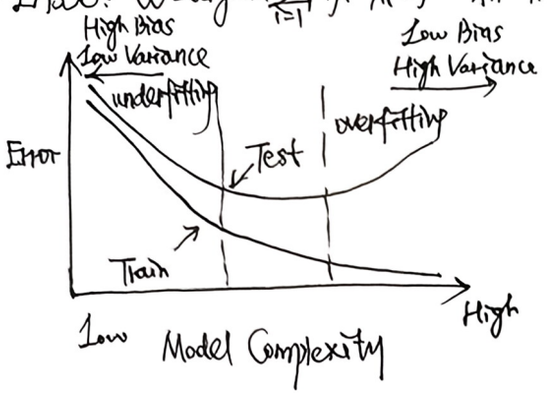
$$\hat{y} = X^T\omega = X^T(XX^T)^{-1}Xy.$$

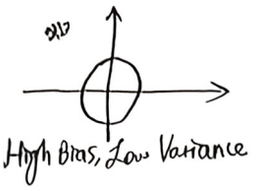
Ridge: $w^* = \arg\min \sum_{i=1}^{n} (y_i - x_i^T w)^2 + \lambda \sum_{j=1}^{p} w_j^2$. $\Leftrightarrow a^* = \arg\min \sum_{i=1}^{n} (y_i - x_i^T w)^2$ subject to $\sum_{j=1}^{p} w_j^2 \le t$.

(拉格朗日乘数法) $J_n = (y - X^T w)^T (y - X^T w) + \lambda w^T w$.

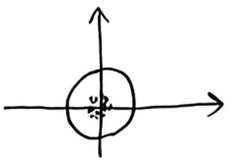
对 w 求导令 $\nabla J_n = -2X(y - X^T w) + 2\lambda w = 0 \Rightarrow (XX^T + \lambda I) w = Xy \Rightarrow w^* = (XX^T + \lambda I)^{-1} Xy$.

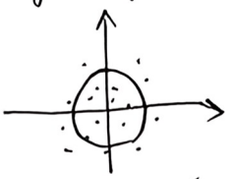
LASSO: $\hat{w} = \arg\min \sum_{i=1}^{n} (y_i - x_i^T w)^2 + \lambda \|w\|_1$, Sparse Model





large λ: over-regularized (underfitting) high bias





small λ: under-regularize (overfitting) high variance

Classifier:

regression: $f(x) = w^T x \in \mathbb{R}$.

sigmoid / logistic function $\sigma(x) = \frac{1}{1+e^{-x}} \in (0,1)$

$p(y_i = \pm 1 \mid x_i, a) = \sigma(y_i a^T x_i) = \frac{1}{1+e^{-y_i a^T x_i}}$

MLE: $L = \prod_{i=1}^{n} \sigma(y_i a^T x_i)$

$l = \log L = \sum_{i=1}^{n} \log \sigma(y_i a^T x_i) = \sum_{i=1}^{n} \log \frac{1}{1+e^{-y_i a^T x_i}} = -\sum_{i=1}^{n} \log(1+e^{-y_i a^T x_i}) = E(a)$. Gradient Desc

$\min J(w)$. $w_{n+1} = w_n - \gamma \nabla J(w_n)$.

Multi-Class Classifier.

- One vs Rest: 多个one分类器
- One vs One: 每两个类别作为一个分类器 $\frac{K(K-1)}{2}$ 个二分类器
- ECOC: 多分类问题转化为多个二分类问题. 海明头距离最优: 任意两个类别二元码之间海明距离最大

码长为9. 类别为4: A={000000000}, B={000111111}, C={111000111}, D={111111000}.

类别 \ 分类器	f_1	f_2	…	f_9
A				
B				
C				
D				

测试样本: {1, 0, 1, 0, …}

← 计算海明距离. 选择距离最小的作为预测结果

2.

SVM: $f(x)=w^Tx+b=0$.

$f(x_p)=w^T(x-\gamma\frac{w}{\|w\|})+b=0$. $f(x)=w^Tx+b=r\|w\|$. $\Rightarrow r=\frac{f(x)}{\|w\|}$

$\gamma=y\cdot r=y\frac{w^Tx+b}{\|w\|}$. geometrical margin

Maximum Margin Classifier: $\max_{w,b}\gamma=\max_{w,b}\frac{y(w^Tx+b)}{\|w\|}$, s.t. $\gamma_i\geq\gamma$.

fix $y(w^Tx+b)=1$. $\max_{w,b}\gamma\Leftrightarrow\min_{w,b}\frac{1}{2}\|w\|^2$ ~~s.t. $\gamma_i=\frac{y_i(w^Tx_i+b)}{\|w\|}\geq\gamma$~~

$\gamma_i=\frac{y_i(w^Tx_i+b)}{\|w\|}\geq\gamma\Leftrightarrow y_i(w^Tx_i+b)\geq 1$

MMC: $\min_{w,b}\frac{1}{2}\|w\|^2$, s.t. $y_i(w^Tx_i+b)\geq 1$.

支持向量：margin上的点，非支持向量的移动不影响超平面。

Slack variables $\min_{w,b,\xi}\frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\xi_i$, s.t. $y_i(w^Tx_i+b)\geq 1-\xi_i$, $\xi_i\geq 0$. 松弛。

C↑. ξ↓、overfitting.

C↓ ξ↑ underfitting

核方法：将映射性不可分问题且避免维数灾难 核函数：直接计算高维内积。

五、神经网络

1. 感知机 $f(x)=w^Tx+b$. predict wrong, set $w_{t+1}=w_t+xy$. $(w_t^Txy>0)$

OR: $f(x)=x_1+x_2+0.5$, AND: $f(x)=x_1+x_2-1.5$, XOR: X

$\mathcal{L}(w,b)=\sum_{x\in M}-y_i(wx_i+b)$., $\nabla\mathcal{L}=\sum_{x_i\in M}-x_iy_i$ $w_{k+1}=w_k+\eta_k\sum x_iy_i$ (gradient descent) learning rate

2. 反向传播：

forward: $z=f(\sum_i w_{kj}f(\sum_i w_{ji}x_i+w_{j0})+w_{k0})$

Activation func 激活函数

$\Delta w=-\eta\Delta J=-\eta\frac{\partial J}{\partial w}$. $w_{k+1}=w_k-\eta\frac{\partial J}{\partial w}$. $\frac{\partial J}{\partial w_{11}^1}=\frac{\partial J}{\partial\hat{y}}\cdot\frac{\partial\hat{y}}{\partial z^2}\cdot\frac{\partial z^2}{\partial a_1^1}\cdot\frac{\partial a_1^1}{\partial z_1^1}\cdot\frac{\partial z_1^1}{\partial w_{11}^1}$

例: Input (x_1,x_2). hidden (h_1,h_2) output $\hat{y}$, $f=$ sigmoid. $\mathcal{L}=MSE=\frac{1}{2}(y-\hat{y})^2$.

$x=[x_1,x_2]=[0.5,-0.8]$, $y=1$. $w^1=\begin{bmatrix}w_{11}^1 & w_{12}^1\\ w_{21}^1 & w_{22}^1\end{bmatrix}=\begin{bmatrix}0.1 & 0.3\\ -0.2 & 0.4\end{bmatrix}$, $b^1=[0.1,-0.1]$, $w^2=[0.2,-0.5]$, $b^2=0.05$.

forward: $z_1^1=w_{11}^1x_1+w_{12}^1x_2+b_1^1=0.31$. $a_1^1=\sigma(0.31)=0.58$

$z_2^1=w_{21}^1x_1+w_{22}^1x_2+b_2^1=-0.27$. $a_2^1=\sigma(-0.27)=0.43$.

$z^2=w_1^2a_1^1+w_2^2a_2^1+b^2=-0.05$, $a^2=\sigma(-0.05)=\hat{y}=0.49$ $(\hat{y}=\frac{1}{1+e^{-z}})$

backward: $\frac{\partial\mathcal{L}}{\partial\hat{y}}=\hat{y}-y=-0.51$. $\frac{\partial\hat{y}}{\partial z^2}=\hat{y}(1-\hat{y})=0.25$. $\delta^2=\frac{\partial\mathcal{L}}{\partial\hat{y}}\cdot\frac{\partial\hat{y}}{\partial z^2}=-0.13$

$\frac{\partial\mathcal{L}}{\partial w_1^2}=\delta^2\cdot a_1^1=-0.07$, $\frac{\partial\mathcal{L}}{\partial w_2^2}=\delta^2\cdot a_2^1=-0.06$. $\frac{\partial\mathcal{L}}{\partial b^2}=\delta^2=-0.13$.

$\frac{\partial z^2}{\partial a_1^1}=w_1^2=0.2$, $\frac{\partial z^2}{\partial a_2^1}=w_2^2=-0.5$, $\frac{\partial a_1^1}{\partial z_1^1}=a_1^1(1-a_1^1)=0.24$, $\frac{\partial a_2^1}{\partial z_2^1}=0.25$.

$\delta_1^1=\delta^2\times w_1^2\times\frac{\partial a_1^1}{\partial z_1^1}=-0.01$, $\delta_2^1=\delta^2\times w_2^2\times\frac{\partial a_2^1}{\partial z_2^1}=0.02$.

$\frac{\partial\mathcal{L}}{\partial w_{11}^1}=\delta_1^1\times x_1$, $\frac{\partial\mathcal{L}}{\partial w_{21}^1}=\delta_1^1\times x_2$, $\frac{\partial\mathcal{L}}{\partial b_1^1}=\delta_1^1$, $\frac{\partial\mathcal{L}}{\partial w_{21}^1}=\delta_2^1\times x_1$, $\frac{\partial\mathcal{L}}{\partial w_{22}^1}=\delta_2^1\times x_2$, $\frac{\partial\mathcal{L}}{\partial b_2^1}=\delta_2^1$.

$w_{1\_new}^2=w_1^2-\eta\frac{\partial\mathcal{L}}{\partial w_1^2}$, $w_{2\_new}^2=w_2^2-\eta\frac{\partial\mathcal{L}}{\partial w_2^2}$, $b_{\_new}^2=b^2-\eta\frac{\partial\mathcal{L}}{\partial b^2}$

$w_{11}^1, w_{12}^1, w_{21}^1, w_{22}^1, b_1^1, b_2^1$ 同样更新.

激活函数：non-linear: sigmoid, tanh.

$\frac{e^x-e^{-x}}{e^x+e^{-x}} \in (-1,1) = 2\sigma(2x)-1$　　vanishing gradient ⇒ ReLU. dropout. (-overfitting)

六. 决策树：

ID3、$Entropy(S) = -\sum_{i=1}^{C} p_i \log p_i$.　$Gain(S,A) = Entropy(S) - \sum_{v} \frac{|S_v|}{|S|} Entropy(S_v)$.

例：

	ID	Age	Gender	Income	Buy?
	1	27	M	15W	X
	②	47	F	30W	✓
	3	32	M	12W	X
	4	24	M	45W	✓
**	⑤	45	M	30W	X
**	⑥	56	M	32W	✓
	7	31	M	15W	X
	⑧	23	F	30W	✓

Age: 20-30, 30-40, 40+

Income: 10-20, 20-40, 40+

$a \in \{Age, Gender, Income\}$

(4+,4-)

$Entropy(S) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = 1$.

Age: 20-30: 2+, 1-. $E = -\frac{1}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3} = 0.918$

30-40: 0+, 2-. $E = 0$

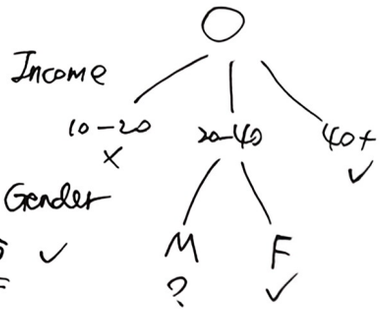
40+ : 2+, 1-. $E = 0.918$

$Gain(S, Age) = 1 - \left(\frac{3}{8}\times 0.918 + \frac{2}{8}\times 0 + \frac{3}{8}\times 0.918\right) = 0.312$.

用程序计算

$Gain(S, Gender) = 0.312$

$Gain(S, Income) = 0.595$



Income: 10-20: 0+, 3-. $E=0$

20-40: 3+, 1-. $E=0.811$

40+ : 1+, 0-. $E=0$

→ Age: 20-30: 1+, 0-　$E=0$

30-40: /

40+, 2+, 1-　$E=0.918$

$Gain = 0.811 - \left(\frac{1}{4}\times 0 + \frac{3}{4}\times 0.918\right) = 0$

Age: 20-30,

30-40: 无法划分

40+:

← Gender. M: 1+, 1-　$E=1$

F: 2+, 0-　$E=0$

$Gain = 0.811 - \frac{2}{4}\times 1 = 0.311$

CART (Classification and Regression Tree): partition the input space for continuous features (greedy method)

$\min\left[\min_{C_1}\sum_{x_i}(y_i-C_1)^2 + \min_{C_2}\sum_{x_i}(y_i-C_2)^2\right]$

feature 对 S 的 threshold

七. Bagging & Boosting / Ensemble Method

1. Bagging

选择决策树作为 base learner: 1. non-linear, 2. easy to use, 3. easy to overfit.

含义：Bagging 是一种并行的聚合法，通过重采样投票(分类)/取平均(回归)的方式改进结果

Random Forest: 1. random sampling

☆2. randomized feature selection at each split step

no pruning for overfitting

Vote for prediction

4.

2. Boosting.

定义：顺序聚合，每步提高被错分样本的权重。

AdaBoost: 1. $w_n^{(1)} = \frac{1}{N}$

2. for $m = 1, \cdots, M$:

3. fit classifier $y^{(m)}(x)$, $J_m = \sum_{n=1}^{N} w_n^{(m)} I(y^{(m)}(x_n) \neq t_n)$

4. $\varepsilon_m = \frac{J_m}{\sum_{n=1}^{N} w_n^{(m)}}$, $\alpha_m = \ln \frac{1-\varepsilon_m}{\varepsilon_m}$

5. update w_n^{m+1}

6. $Y_M(x) = \text{sign}(\sum_{m=1}^{M} \alpha_m y^{(m)}(x))$ #final model

Gradient Boosting Decision Tree (GBDT): predict remaining data & sum

八. KNN.

{ stored record, distance metrics, k } ⇒ { 1. compute the distance to other record; 2. find the k nearest neighbor; 3. vote }

九. Clustering

K-Means: 1. Randomly pick k data points as μ_k

2. Iterate until converge:

3. $\hat{z}_i = \arg\min_k \|x_i - \mu_k\|^2$, # assign each point to the cluster

4. $\mu_k = \frac{1}{N_k} \sum_{i=1}^{N_k} x_i^{(k)}$ # update seed point.

十. dimensionality reduction.

1. PCA. (unsupervised)

1st PC: $z_i^{(1)} = a_1^T x_i$, s.t. max Var($z^{(1)}$)

Solution: $Var(z^{(1)}) = E((z^{(1)} - \bar{z}^{(1)})^2) = \frac{1}{n} \sum_{i=1}^{n} (a_1^T x_i - a_1^T \bar{x})^2 = \frac{1}{n} \sum_{i=1}^{n} a_1^T (x_i - \bar{x})(x_i - \bar{x})^T a_1 = a_1^T S a_1$.

$\Rightarrow \max_{a_1} a_1^T S a_1$, s.t. $a_1^T a_1 = 1$. $\mathcal{L} = a_1^T S a_1 - \lambda(a_1^T a_1 - 1)$. $\frac{\partial \mathcal{L}}{\partial a_1} = 0 \Rightarrow \lambda a_1 = S a_1$

$\Rightarrow a_1$ is the eigenvector of S corresponds to the largest eigenvalue λ_1, $S = \frac{1}{n} \sum (x_i - \bar{x})(x_i - \bar{x})^T$

2nd PC: $\max_{a_2} a_2^T S a_2$, s.t. $a_2^T a_2 = 1$, $Cov(z^{(1)}, z^{(2)}) = 0$.

例: Given (90, 85, 80); (60, 65, 70); (85, 80, 75); (70, 75, 80); (95, 90, 85). $\bar{x} = (80, 79, 78)$

$\Rightarrow$ (10, 6, 2); (-20, -14, -8); (5, 1, -3); (-10, -4, 1); (15, 11, 7).

$\Rightarrow S = \frac{1}{5}\left(\begin{pmatrix} 10 \\ 6 \\ 2 \end{pmatrix}(10\ 6\ 2) + \begin{pmatrix} -20 \\ -14 \\ -8 \end{pmatrix}(-20\ -14\ -8) + \cdots\right) = \begin{pmatrix} 170 & 110 & 50 \\ 110 & 74 & 38 \\ 50 & 38 & 26 \end{pmatrix}$

$(S - \lambda_1 I) v_1 = 0$

$v_1 = \begin{pmatrix} 0.805 \\ 0.533 \\ 0.260 \end{pmatrix} = PC_1$

$|S - \lambda I| = \begin{vmatrix} 170-\lambda & 110 & 50 \\ 110 & 74-\lambda & 38 \\ 50 & 38 & 26-\lambda \end{vmatrix} = 0 \Rightarrow \lambda_1 = 258.875, \lambda_2 = 11.125, \lambda_3 = 0$ 投影: $x_i v_1$ 投影: $(x_i - \bar{x}) \cdot v_1$

2. LDA. (supervised).

Rayleigh quotient $= J(a) = \frac{a^T S_B a}{a^T S_w a}$, S_w: 类内散度, S_B: 类间散度

$\max J(a) \Leftrightarrow S_w^{-1} S_B$.

例:

ID	类别	数	语	
1	好	90	70	$\mu_1 = [85\ 65]$
2	好	85	65	
3	好	80	60	
4	差	60	85	$\mu_2 = [65\ 85]$
5	差	65	80	
6	差	70	80	

$\mu = [75\ 75]$

S_w: 好: $\begin{pmatrix} 5 & 5 \\ 0 & 0 \\ -5 & -5 \end{pmatrix}$. $S_1 = \sum_1 (X-\mu_1)(X-\mu_1)^T = \begin{pmatrix} 5 \\ 5 \end{pmatrix}(5\ 5) + \begin{pmatrix} 0 \\ 0 \end{pmatrix}(0\ 0) + \begin{pmatrix} -5 \\ -5 \end{pmatrix}(-5\ -5) = \begin{pmatrix} 50 & 50 \\ 50 & 50 \end{pmatrix}$

差: $\begin{pmatrix} -5 & 0 \\ 0 & 5 \\ 5 & -5 \end{pmatrix}$ $S_2 = \begin{pmatrix} 50 & -50 \\ -50 & 50 \end{pmatrix}$ $\Rightarrow S_w = \begin{pmatrix} 100 & 50 \\ 50 & 100 \end{pmatrix}$.

$S_B = \sum_{i=1}^{2} n_i (\mu_i - \mu)(\mu_i - \mu)^T = 3\begin{pmatrix} 10 \\ -10 \end{pmatrix}(10\ -10) + 3\begin{pmatrix} -10 \\ 10 \end{pmatrix}(-10\ 10) = \begin{pmatrix} 600 & -600 \\ -600 & 600 \end{pmatrix}$

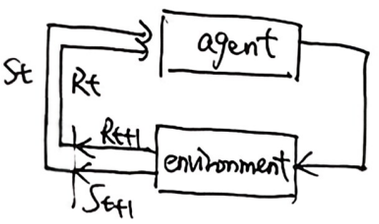
$S_B w = \lambda S_w w \Rightarrow \begin{pmatrix} 600 & -600 \\ -600 & 600 \end{pmatrix} w = \lambda \begin{pmatrix} 100 & 50 \\ 50 & 100 \end{pmatrix} w$

$S_w^{-1} S_B = \begin{pmatrix} 8 & -8 \\ -8 & 8 \end{pmatrix} \Rightarrow \lambda_1 = 16,\ \lambda_2 = 0.$

$w_1 = (1, -1)$

十一. 强化学习.

agent-environment interaction



(State, Action, Reward) = (S, A, R).

goal: optimal policy to max reward R.

- policy $\pi(s)$
- reward function
- value function $V_\pi(s)$

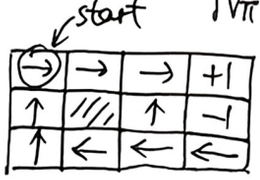
Markov Decision Process (MDP): $P_r(S_{t+1}=s', R_{t+1}=r \mid S_t, A_t, R_t, S_{t-1}, A_{t-1}, \dots, S_0, A_0) = P_r(S_{t+1}=s', R_{t+1}=r \mid S_t, A_t)$.

Bellman equation:

$$q_\pi(s,a) = \sum_{s',r} p(s',r \mid s,a)(r + \gamma V_\pi(s'))$$

$$V_\pi(s) = \sum_a (\pi(a \mid s) \sum_{s',r} p(s',r \mid s,a)(r + \gamma V_\pi(s')))$$

例:



0.812	0.868	0.912	+1
0.762		0.660	-1
0.705	0.655	0.611	0.388

$p = 0.8$ ✓

$p = 0.1 \leftarrow, 0.1 \rightarrow$

-0.04 each step, $\gamma = 1$

$V_\pi(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s',r \mid s,a)(r + \gamma V_\pi(s'))$

$= 0.8 \times 1 \times (-0.04 + 1 \times 0.868) + 0.1 \times 1 \times (-0.04 + 1 \times 0.812) + 0.1 \times 1 \times (-0.04 + 0.868)$

$q_\pi(s,a) \rightarrow \sum_{s',r} p(s',r \mid s,a)(r + \gamma V_\pi(s'))$

$= 1 \times (-0.04 + 1 \times 0.868)$

6.

DP: $V_\pi(S_t) = E_\pi[R_{t+1} + \gamma V(S_{t+1})]$

policy evaluation: $V_{k+1}(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)(r + \gamma V_k(s'))$ $\quad \{V_k(s)\} \Rightarrow \{V_\pi(s)\}$.

" improvement: $q_\pi(s, \pi'(s)) \geq V_\pi(s) \Leftrightarrow V_{\pi'}(s) \geq V_\pi(s)$.

$\pi'(s) = \arg\max_a \sum_{s',r} p(s',r|s,a)(r + \gamma V_\pi(s'))$: greedy

" $q_\pi(s', a)$

Monte Carlo: $q_\pi(s,a) = E[G_t]$. $V(S_t) \leftarrow V(S_t) + \alpha(G_t - V(S_t))$.

G_t = return at the end

exploration starts: first S-A pair

Temporal-Difference learning: $V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$

R_{t+1} = return in one step

Sarsa: ε-greedy: ε for exploration (all-actions)

$1-\varepsilon$ for exploitation ($\pi(s)$)

$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$

Q-learning: fully-greedy exploration. $\pi(s) = \arg\max_a Q(s,a)$

$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_{a'} Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$