1- Search : dfs - bfs - iterative depening - ucs

Greedy - A* $f(n)=g(n)+h(n)$

tree search vs graph search (add closed set)

A* optimal (tree) : $0 \le h(n) \le h^*(n)$

A* optimal (Graph) : $h(A) - h(C) \le cost(A\ to\ C)$

$(A>B) \vee (B>A) \vee (A \sim B)$

$(A>B) \wedge (B>C) \Rightarrow A>C$

$A>B>C \Rightarrow \exists p$ $[p, A; 1-p, C] \sim B$

$A \sim B \Rightarrow [p, A; 1-p, C] \sim [p, B; 1-p, C]$

$A>B \Rightarrow p \ge q \Leftrightarrow [p, A; 1-p, B] > [q, A; 1-q, B]$

2- CSP : { }

backtracking search (dfs improved by 2 things)

function BS(csp) returns s/f
    return RB

function RB(assignment, csp) returns s/f
    if assign is complete then return assign
MRV • var ← select - unassigned - var(variables[csp], assign, csp)
LCV • for each value in order-domain - value(var, assign, csp) do
        if value is consistent with assignment given constraint[csp] then
            add {var = value} to assign
            resut ← RB(assign, csp)
            if result ≠ fail then return result
            remov {var = value} from assign
filtering
    return fail

forward-checking - arc consistency ($x \to y$) - K-consistency Strong

Structure: independent subproblems - tree - nearly tree cutset

iterative improvment - Genetic

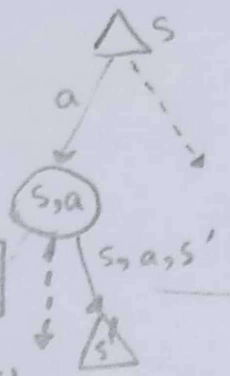
3- game tree : minimax - Alpa-beta pruning

def max-value(state, α, β):
    v = -∞
    for each successor:
        v = max(v, value(successor, α, β))
        if v > β return v
        α = max(α, v)
    return

def min-value(state, α, β):
    v = +∞
    for each successor:
        v = min(v, value(successor, α, β))
        v ≤ α return v
        β = min(β, v)
    return v

4. MDP: $S, A, T(s,a,s'), R(s,a,s')$, Start state, terminal

$\pi^*: S \to A$



$$V^*(s) = \max_a Q^*(s,a)$$

$$Q^*(s,a) = \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma V^*(s')\right]$$

$\to T(s,a,s') = P(s'|s,a)$

$R(s,a,s')$

$$V^*(s) = \max_a \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma V^*(s')\right]$$

value-iteration : $V_{k+1} \leftarrow$ " with k step

policy-evaluation: $V^{\pi}(s) = \sum_{s'} T(s,\pi(s),s')\left[R(s,\pi(s),s') + \gamma V^{\pi}(s')\right]$

$\to$ VE حل کردن با تکنیک

$k+1 \leftarrow k$

policy-extraction:

$$\pi^*(s) = \arg\max_a \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma V^*(s')\right]$$

$$\pi^*(s) = \arg\max_a Q^*(s,a)$$

policy-iteration :

1. Policy evalnation $\longrightarrow V^{\pi_i}_{k+1}(s) \leftarrow \sum_{s'} T(s,\pi_i(s),s')\left[R(s,\pi_i(s),s') + \gamma V^{\pi_i}_k(s)\right]$
2. policy improvement $\to \pi_{i+1}(s) = \arg\max_a \sum T(s,a,s')\left[R(s,a,s') + \gamma V^{\pi_i}(s')\right]$
3. repeat untill policy converges $\pi_i \stackrel{?}{=} \pi_{i+1}$

5-RL: Model-Based , model-free

Model-Based $\to$ learn T,R

مثل ما محاسبه می کنیم

model-free $\to$ passive-RL, active-RL

active-RL $\to$ Q-Value learning

passive-RL $\to$ direct, indirect

direct: میانگین مجموع r ها (تا آخر رفتن)

indirect: $V^{\pi}_{k+1}(s) \leftarrow \frac{1}{n}\sum_i \text{sample}_i$

$\to \text{sample}_i = R(s,\pi(s),s_i') + \gamma V^{\pi}_k(s_i')$

$\to$ TD : $\begin{cases} \text{sample of } V(s) = R(s,\pi(s),s') + \gamma V^{\pi}(s') \\ V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + \alpha\, \text{sample} \\ V^{\pi}(s) \leftarrow V^{\pi}(s) + \alpha(\text{sample} - V^{\pi}(s)) \end{cases}$

Q-value : $Q_{k+1}(s,a) \leftarrow \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma \max_{a'} Q_k(s',a')\right]$

$\text{sample} = R(s,a,s') + \gamma \max_{a'} Q(s',a')$

$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\, \text{sample}$

2,,

ε-greedy $\to$ ε : random
$\to$ $1-\varepsilon$: act on current π

} exploration function: $f(u,n) = u + K/n$

u: مقدار
K: ثابت
n: دفعات visit

$Q(s,a) \xleftarrow{\alpha} R(s,a,s') + \gamma \max_{a'} \underbrace{Q(s',a')}$

جایگزین کن با : $f(Q(s',a'), N(s',a'))$

$V(s) = \sum w_i f_i(s)$ } $Q(s,a) = \sum w_i f_i(s,a)$

difference $= \left[r + \gamma \max_{a'} Q(s',a') \right] - Q(s,a)$

$Q(s,a) \leftarrow Q(s,a) + \alpha$ difference } exact Qs

$w_i \leftarrow w_i + \alpha(\text{difference}) f_i(s,a)$ } Approximate Qs

---

6. probability and BNs : rep :

inference by enumeration

$P(Q, e_1 \dots e_k) = \sum_{h_1 \dots h_r} P(Q, h_1 \dots h_r, e_1 \dots e_k)$, $Z = \sum_q P(Q, e_1 \dots e_k)$

$P(Q | e_1 \dots e_k) = \frac{1}{Z} P(Q, e_1 \dots e_k)$

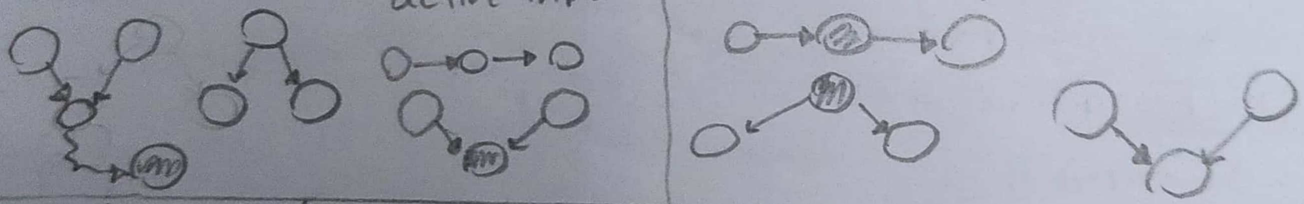
$X \perp\!\!\!\perp Y$: $P(x,y) = P(x)P(y)$ & $P(x|y) = P(x)$

$X \perp\!\!\!\perp Y | Z$: $P(x,y|z) = P(x|z)P(y|z)$ & $P(x|z,y) = P(x|z)$

7. independence

BN $\to P(x_1, \dots, x_n) = \prod_{i=1}^{n} P(x_i | \text{parents}(x_i))$

D-sepration : active tripler | inactive triple



---

8. inference and sampling:

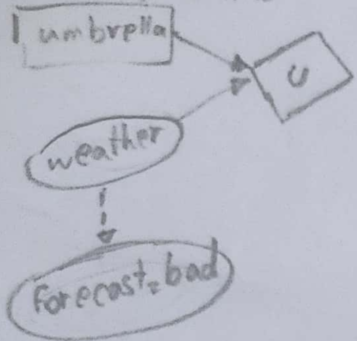
variable elimination: $P(Q | E_1 = e_1, \dots, E_k = e_k)$

- start with initial factors (instantiate evidence)
- while there are still hidden vars (not Q or e)
  - pick a H
  - join all factors that has H
  - sum out H
- join all remaining f and normalize

---

prior sampling
rejection "
likelihood " $\to$ w for each sample (فقط احتمالات e ها رو در هم ضرب میکنیم) $\to$ جمع وزن مربوط و نرمالایز
gibbs " $\to$ Consitent instantiation with e's $\to$ pick a non-e var and sample given others

9. DN & VPI and HMMs :



$EU(\underset{\text{action}}{\text{leave}}) = \sum_w P(w)\, U(\text{leave}, w)$

$MEU(\varnothing) = \max_a EU(a) = 70$

no evidence

$MEU(F = \text{bad}) = \max_a EU(a|\text{bad})$

$\sum P(w|\text{bad}) U(a,w)$

MEU یک average

VPI ≥ 0
جمع نیست.
ترتیب دونستن
مهم نیست

VPI is for knowing a random var (all possible states) − MEU (when not knowing that)

$VPI(E'|e) = \left(\sum_{e'} P(e'|e)\, MEU(e, e')\right) - MEU(e)$

MM: $P_\infty(X) = P_{\infty+1}(X) = \sum_x P(X|x) P_\infty(x)$ { $B(X_t) = P(X_t|e_{1:t})$

HMM: passage of time: $P(X_{t+1}|e_{1:t}) = B'(X_{t+1}) =$ ادامه باس

$\sum_{x_t} P(X_{t+1}|x_t)\, B(x_t)$

$P(x_{t+1}|e_{1:t+1}) = B(X_{t+1}) \propto_{X_{t+1}} P(e_{t+1}|X_{t+1})\, B'(X_{t+1})$

observation

---

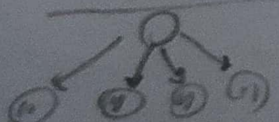
10 - PF & Hmm + naive bayes

PT ← $x' = \text{sample}(P(X'|x))$ → particle / جابه‌جا کردن

$B(X) \propto P(e|X)\, B'(X)$ → weight

O ← and normalize

Resample



$P(Y|F_{0,0} \dots F_{15,15}) \propto P(Y) \prod_{i,j} P(F_{i,j}|Y)$ →

برای همه حساب می‌کنیم و بعد نرمالایز.

$P_{LAP}(x) = \frac{C(x)+1}{\sum_x [C(x)+1]} = \frac{C(x)+1}{N+|X|}$

یک بار بیشتر همه رو دیدیم

$P_{LAP,k}(x) = \frac{C(x)+k}{N+k|X|}$

11 multiclass percep

① $\text{activation}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$; $>0 \to +1$; $<0 \to -1$

$y = y^* \to$ no change

else $\to w = w + y^* \cdot f$

binary perceptron

②

$y = \arg\max_y w_y \cdot f(x)$ { $w_y \to$ برای هر کلاس ; $w_y \cdot f(x) \to$ " "

update : { $w_y = w_y - f(x)$; $w_{y^*} = w_{y^*} + f(x)$

③

$z = w \cdot f(x)$; $+ \to 1$; $- \to 0$

$\phi(z) = \frac{1}{1+e^{-z}}$

Best w : $\max_w ll(w) = \max_w \sum \log P(y^{(i)}|x^{(i)}; w)$

$P(y^i = +1 | x^i; w) = \frac{1}{1+e^{-w \cdot f(x^i)}}$

$P(y^i = -1 | x^i; w) = 1 - \frac{1}{1+e^{-w \cdot f(x^i)}}$ 4

logistic regression

multi class regression: $z_1, z_2, z_3 \xrightarrow{\text{softmax}} \frac{e^{z_1}}{e^{z_1}+e^{z_2}+e^{z_3}}, \frac{e^{z_2}}{e^{z_1}+e^{z_2}+e^{z_3}}, \dots$

$$P(y^i \mid x^i; w) = \frac{e^{w_{y(i)} \cdot f(x^{(i)})}}{\sum_y e^{w_y \cdot f(x^{(i)})}}$$

Gradient Ascent:

$g(w_1, w_2)$

5

$w_1 \leftarrow w_1 + \alpha * \frac{\partial g}{\partial w_1}(w_1, w_2)$

$w_2 \leftarrow w_2 + \alpha * \frac{\partial g}{\partial w_2}(w_1, w_2)$

$w \leftarrow w + \alpha * \nabla_w g(w)$

$$\underbrace{\nabla_w g(w)}_{\text{gradient}} = \begin{bmatrix} \frac{\partial g(w)}{\partial w_1} \\ \frac{\partial g}{\partial w_2}(w) \end{bmatrix}$$

$$\max_w ll(w) = \max_w \underbrace{\sum_i \log P(y^i|x^i; w)}_{g(w)} \Rightarrow w \leftarrow w + \alpha * \sum_i \nabla \log$$

x_1 $z_1^{(1)}$ z_1^{n-1} $z_1^{(n)} = f_1(x)$ z_1

x_2 $z_2^{(1)}$ z_2^{n-1} z_2

$\vdots$ $\vdots$ $\vdots$ $\vdots$

x_L $z_{K^{(1)}}^{(1)}$ $z_{K^{(n-1)}}^{n-1}$ $f_{K(n)}$ z_3

لایه

softmax 5

$\frac{1}{1+e^{-z}}$ 1 تا 0

$\frac{e^z - e^{-z}}{e^z + e^{-z}}$ 1 تا -1

$g(z) = \max(0, z)$

$$z_i^{(k)} = g\left(\sum_j w_{i,j}^{(k-1,k)} z_j^{(k-1)}\right)$$