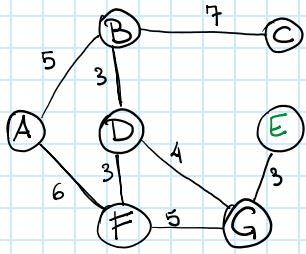
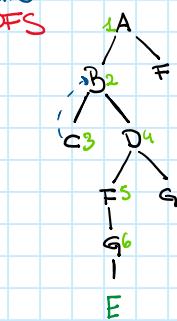


DFS & BFS

martedì 30 gennaio 2024 14:04

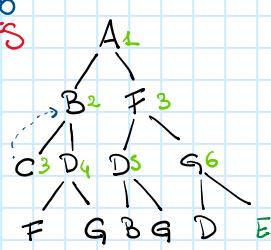


LIFO
DFS



Spazio: $O(d)$
Tempo: $O(b^d)$

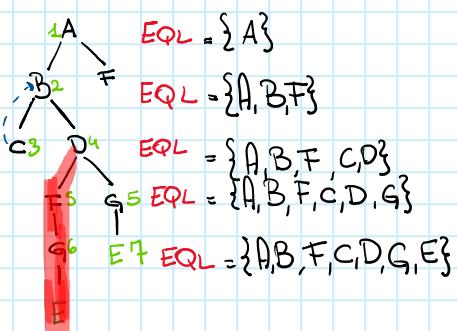
FIFO
BFS



EQL

UNA LISTA IN CUI METTERE TUTTI I NODI IN FRONTEIRA
IN MODO CHE VENGANO PRUNATI SE

DFS(EQL)



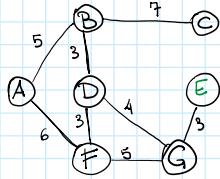
UCS

martedì 30 gennaio 2024 14:39

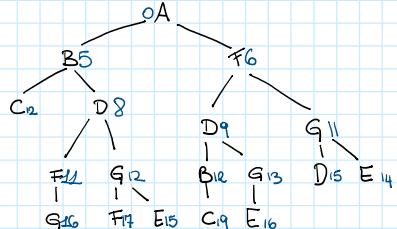
UNIFORMED COST SEARCH

$g(\text{nodo})$ = COSTO ACCUMULATO DA START

ESPLORANO IL NODO CON $g(\text{nodo})$ MINORE



UCS



OTTIMALITÀ

$$g(x_0 \rightarrow \dots \rightarrow x_n) = g(x_n)$$

Per es.

① UCS SCEGLIE V TRAMITE RECORDO ρ

② SIA $P \neq P^*$

Dimostrazione:

• PER IL PUNTO ② $\exists x$ IN FRONTIERA TRAMITE P^* SU P^* PER V

$$P^* = P_1^* \cup P_2^*$$

• P^* È OTTIMO QUINDI $g(P^*) = g(P_1^*) + \Delta_{P_2^*} < g(P)$

• $g(P_1^*) < g(P_1^*) + \Delta_{P_2^*} < g(P) \Rightarrow g(P_1^*) < g(P)$

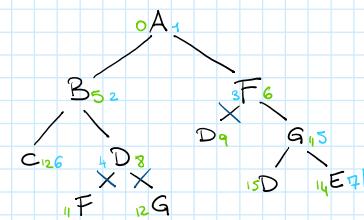
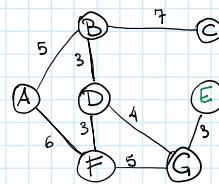
• $\Rightarrow g(x) < g(v)$

ExL (Excusion List)

Ogni volta che esplora un nodo

SE ! IN EXL → PRUNE
IN EXL → EXPAND & ADD TO EXL

Ucs con ExL



ExL = { }
ExL = {A}
ExL = {A, B}
ExL = {A, B, F}
ExL = {A, B, F, D}
ExL = {A, B, F, D, G}
ExL = {A, B, F, D, G, C}
ExL = {A, B, F, D, G, C, E}

È UNA DERIVAZIONE DELL'UCS

$$f(s) = g(s) + h(s) \rightarrow \text{SIIMA COSTO DA SPENDERE}$$

↓
COSTO ACCUMULATO

EURISTICA

EURISTICA \leftarrow AMMISSIBILITÀ ①
CONSISTENZA ②① UN'EURISTICA È AMMISSIBILE SE IL STATO $h(s)$ NON SOVRASTIMA IL COSTO MINIMO DA S-GOAL② SIANO GLI STATI V, U CONNESSI DA UN AZIONE a

$$h \text{ È CONSISTENTE SE } \forall V, U \text{ VALE: } \frac{h(V) \leq c(V, a, U) + h(U)}{\substack{\text{COSTO} \\ \text{V-GOAL}} \quad \substack{\text{COSTO} \\ \text{U-GOAL}}}$$

OTTIMALITÀ A*: 1 DATI V, U CONNESSI DA a

$$\begin{aligned} 2 & f(u) = g(u) + h(u) \text{ DED A*} \\ 3 & g(v) = g(v) + c(v, a, u) \\ 4 & f(v) = g(v) + c(v, a, u) + h(u) \xrightarrow{1} \\ 5 & \text{CONSISTENZA } c(v, a, u) + h(u) \geq h(v) \\ 6 & \text{ADD } g(v) \rightarrow g(v) + c(v, a, u) + h(u) \geq g(v) + h(v) \end{aligned}$$

$f(u)$ $f(v)$

OTTIMALITÀ A*: $f(p_m) = \overbrace{g(p)}^{\text{SMART-P}} + \overbrace{h(m)}^{\text{PREFAT}}$

DOTTESI:

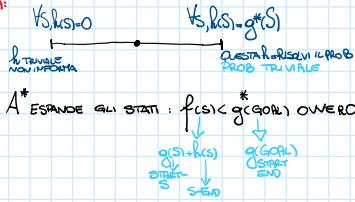
1: A* ESPANDE UN NODO GENERATO DA P

2: $P \neq P^*$

DIMOSTRAZIONE:

- 1: SEPARATION PROPERTY $\Rightarrow \exists x$ SUL CAMMINO OTTIMO: $P^* = P^*_1 + P^*_2$ PER V
- 2: $g(p) > g(p^*)$
- 3: $f(p^*) \geq f(P^*_1, V)$ CONSISTENZA h
- 4: $g(p) + h(v) > g(p^*) + h(v)$ SONNO $h(v)$ AL PUNTO 1
- 5: $f(p^*) > f(p^*, V) > f(p^*, x)$ PUNTO 3 E 4
- 6: $f(p^*) > f(p^*, x) \quad \checkmark$ NON PUÒ ESSERE SCELTO PRIMA X

PROGETTARE UN'EURISTICA:

Se $\forall s, h_1(s) \leq h_2(s) \Rightarrow h_2$ DOMINA h_1 DATE $h_{1(S)}, h_{2(S)}$ DOVE NESSUNA DOMINA L'ALTRA $\Rightarrow \forall s, h(s) = \max\{h_{1(S)}, h_{2(S)}\}$

RILASSAMENTO DI UN PROBLEMA

$$P \xrightarrow{\text{ORIGINALE}} \hat{P} \xrightarrow{\text{RILASSATO}}$$

costi \hat{P} MAI > P $\{c(s, a, v) \leq c(s, a, v)\}$

IDEA RILASSAMENTO

- COSTRUISSO \hat{P}, \hat{P}
- EXE A^* SU \hat{P} E TROVA COSTO OTTIMO DA OGNI NODO S PER ARRIVARE GOAL: $\hat{h}^*(s)$
- $h(s) = \hat{h}(s)$ EXE A*

RIMANE OTTIME

- ① $\forall s, \hat{h}^*(s) \leq \hat{c}(s, a, v) + \hat{h}(v)$
- ② $\hat{h}(v) \leq \hat{c}(s, a, v) + h(v)$
- ③ $\hat{c}(s, a, v) \leq c(s, a, v)$
- ④ h CONSISTENTE $h(s) \leq \hat{c}(s, a, v) + h(v) \leq c(s, a, v) + h(v)$

Focal Search

mercoledì 31 gennaio 2024 15:33

BISOGNA DARE FLESSIBILITÀ AD A* CHE DIVENTEREBBE INUSABILE

SUPPONIAMO DI AVERE h NON AMMISIBILE \hat{h}_F

NOOI $m \cdot f(m)$ IL COSTO OTTIMISTICO DA SPENDERE
 $\hat{h}_F(m)$ STIMA m AZIONI PER IL GOAL,

FOCAL SEARCH

- F : VISTA FRONTIERA DA A*

- $m_{best} = \arg \min_{m \in F} f(m)$

- $w > 1$

- $FOCAL \subseteq F ; FOCAL = \{m \in F \mid f(m) \leq w \cdot f(m_{best})\}$

REGOLA DI ESPANSIONE: SCEGLIE FOCAL MINIMIZZA \hat{h}_F , $m_{next} = \arg \min_{m \in FOCAL} \hat{h}_F(m)$

BSS: BOUNDED SUBOPTIMAL SEARCH

- SIA e IL GOAL DI FOCAL

- m^* PATH OTTIMO

- OPT COSTO OTTIMALE

- $f(m^*) \leq OPT$ f EURISTICA AMMISIBILE

- $f(m_{best}) \leq f(m^*)$ m_{best}

- $f(e) \leq w \cdot f(m_{best})$

$$g(e) = f(e) \leq w \cdot f(m_{best}) \leq w \cdot f(m^*) \leq w \cdot OPT$$

TRASHING: - FRONTIERA m_{best} TENDE A STARE A BASSA PROFONDITÀ \hat{h}_F ALTO
- NOI GENERATI TENDONO A NON ENTRARE FOCAL MOLTO TEMPO

FOCAL È UN'UNITÀ IN CUI VENNE RIMOSSO m_{best}

Adversarial Search

giovedì 1 febbraio 2024 10:35

SE \exists PIÙ AGENTI \Rightarrow Gioco

\forall AGENTE \exists GOAL: GOAL A \neq GOAL B

Gioco: -Lo m GIOCATORI

-RAZIONALI / e-RAZIONALI

-STRUCTURE: TURNI, AZIONI SIMULTANEE

-DETERMINISTICO / STOCASTICO

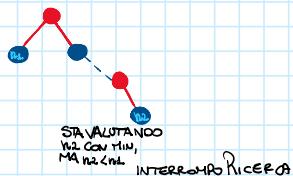
-SOMMA COSTANTE - GENERICA

-COOPERATIVA / COMPETITIVA

Alpha-Beta pruning

giovedì 1 febbraio 2024 15:17

PRUNING: TAGLIARE I RAMI INUTILI



- PRINCIPIO: DURANTE LA DFS V'HO DOVUTO TENERE TRACCIA IL VALORE DI NODO

- FINO A CHE LA DFS NON HA FINITO NON SAPPIAMO IL VALORE MA SAPPIAMO \rightarrow MAX(s) SCOPRISSO FIGLIO CON VAL \geq MAX(s) E ALMENO \geq MIN(q) " " " " " \geq MIN(q) E E' IL MASSIMO \times

- ALLORA USIAMO UN INTERVALLO I DI VALORI $[a, b]$ (QUANTO LA DFS HA SCOPERTO FINO A QUEL MOMENTO)
 (VALORE E' COMPRESO TRA $[a, b]$)
 ↑
 MAX MIN

QUESTI:
 1- ORDERING DELLE MOSSE ((ORDINARE DURANTE LE ITERAZIONI))
 2- SAREBBERE BELLO FOSSE COMPUTABILE. DARE UN PESO AD OGNI NODO

SOLUZIONI:
 DI VALUTAZIONI / CUTOFF
 TAB TRANSPOSITION
 ITERATIVE DEEPENING

VALUTAZIONE / CUTOFF.

Al posto del val Minimax fornisco una stima usando $V(s)$

Approccio: V e' CUTOFF TEST $CUT(S_d)$
 ↓
 NODO
 ↓
 DEPTH



HEURISTICA

FALSE MINIMAX
 TRUE $V(s)$

INCERTITUDINE: IL PROB E' TROPPO COMPLICATO
 TANTO DA NON COMPUTARLO

Lo STATO VIENE IDENTIFICATO DA UN INSIEME DI FEATURE

FUNZIONI BASATE SU: ESPERIENZA VP È UNA CLASSE DI EQUIVALENZA TRA STATI (SO CHE IN P HO VINTO 50%, PERDO 20%, PARIGRADO 30%)

COMBINAZIONE Chiamo $f_1(s), f_2(s)$; LE IN f PER UNO STATO $S \Rightarrow V(s) = w_1f_1(s) + w_2f_2(s) + \dots + w_nf_n(s)$

La funz $CUT(S_d)$
 STATO
 ↓
 DEPTH

STATO QUIESCENTE: GLI S CHE CAMBIANO POCO IL VALORE ($\exists s > 0 \Rightarrow |f(s) - f(s')| < s$)

QUIESCENT SEARCH: E' UN CUTOFF (RETURN $V(s)$) SOLO PER GLI STATI QUIETI

TRANS POSITION TABLE: HASH TABLE CHE V_S VALUTATO TIENE: VALORE DI V E MIGRA DA PROFONDO A RICERCA CHE HA VALUTATO S

$\alpha\beta$ PRUNING CON ID

$D(s_k, d_{MAX}, T)$
 ↓
 STATO
 ↓
 DEPTH
 ↓
 TRANSPPOSITION TABLE

V_S SELEZIONATO DA DFS: $S \notin T \Rightarrow$ VALUTA S; ADD $[S \in T, V \in v, j \in o, d \leq d_{MAX}]$ TO T

SET T
 []
 D se $d \geq d_{MAX}$ PRUNE SUBTREE E USA V
 D se $d \leq d_{MAX}$ ESPANDI NODO CON v

COMPLETA VALUTAZIONE E UPDATE T

FORWARD PRUNING: TIPICAMENTE UN GIOCATORE UN'UNICA VALUTA N POSSIBILI AZIONE

Scelgo n azion (random/p di val/both)

No more ID BUT Beam Search
 CUT \rightarrow Prune cut

Giochi stocastici

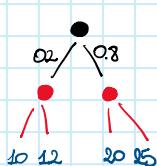
martedì 6 febbraio 2024 13:54

INTRODUO N : INCERTITUDINE: UN NUOVO GIOCATORE NATURA (N)

NE' UN NODO SPECIALE: - LA STRATEGIA E' FISSATA A PRIORI ED E' CONSCIUTA
 - LA STRATEGIA E' MISTA (LANCIO DI DADI)
 - NON HA PAY OFF, E' AGNOSTICO

EXPECTIMAX: UN MINIMAX CHE AL POSTO DI MIN E MAX PRENDE CIOE' LA SOMMA DEI VALORI PESATI CON LA PROB.

ESEMPIO

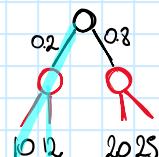


$$EV_1 = 0.2 \times 12 + 0.8 \times 25 = 22.4$$

SE PERO' \forall FOGLIA $10 \leq U_i \leq 30$

$$0.2 \cdot 12 + 0.8 \cdot 10 \leq EV_1 \leq 0.2 \cdot 12 + 0.8 \cdot 30$$

$$10.4 \leq EV_1 \leq 26.4$$



IDEA: TRASFORMO PROCESSO DI STIMA
 MA STIMO LA BONTA' DI AZIONE
 BEST STIMA

- CONSIDERO AZIONE a_i , STATO s
- SIMULO n A PARTIRE DA s CON AZIONE a_i
- LA MEDIA DEI PUNTI FINALI STIMA DEL VALORE
- RIPETO $\forall a_i$ COI CHE BEST STIMA

MCTS

- STIMO WINNING RATE DI a_i : CIOE' IL WIN RATE DI S^i
- SCEGLIO n AZIONI IN TEMPO LIMITATO (TOT N SIMULATION)
- APPROXIMO NAVIGUE $\frac{N}{n}$ V SI. OTTENGO $\underline{w}_i = \frac{\sum_{i=1}^n w_i}{\frac{n}{N}}$ OTTENGO: $a^* = \text{ARGMAX}_i \{ \underline{w}_i \}$

EXPLORATION VS EXPLOITATION

MULTI ARMED BANDIT (MAB)

- IN SLOT: VALORE E' $E[w_i]$
- POSSO AVERE N GIOCATE
- EXPLORATION: GIOCARE SU SLOT PER SCOPRIRE w_i
- EXPLOITATION: GIOCO SU SLOT: $\text{MAX}(w_i)$

$$U_{CB} = \underline{w}_i + c \sqrt{\frac{\log N}{N_i}}$$

PARAMETRO SMS_{UB}

EXPLOIT VS EXPLORE

MCTS-UCB

```

Vai EXE NLSM
Vai CALC UCB;  $\Lambda_{w_i}$ 
WHILE TIMELEFT > 0
    SCEGLI  $a_i$ : MAX(UCB $_i$ ) EXE MCTS-UCB
    Vai UPDATE UCB $_i$  e  $w_i$ 
RETURN  $a^*$ : MAX( $w_i$ )
    
```

CSP

martedì 6 febbraio 2024 16:06

CONSTRAINT SATISFACTION Problems

Lo STATO SHETTE DI ESSERE ATOMICO

Arrivo allo STATO GOAL MANIPOLANDO LO STATO

STRUTTURA: STATO \rightarrow INSIEME VAR = null

TRANSIZIONE \rightarrow ASSEGNARE UN VALORE

GOAL CHECK \rightarrow SODDISFARE INSIEME VINCOLI DI ALCUNE O TUTTI VARIABILI

SOLUZIONE $\forall \text{VAR} : \text{VAR} \neq \text{null}$

FORMALIZZAZIONE:

VARIABILI $X = \{x_1, x_2, \dots, x_n\}$

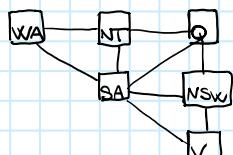
DOMINIO $D = \{D_1, D_2, \dots, D_n\}$

VINCOLI $C = \{C_1, C_2, \dots, C_n\}$

RAPPRESENTAZIONE VINCULO: $\langle \text{VARIABILI}, \text{RELAZIONE} \rangle$
↓
ELenco k-TUPLE
k ∈ VAR

ESEMPIO: $\langle (x_1, x_2, x_3), [(0,0,0), (1,0,0), (0,0,1), (0,1,0)] \rangle$ AL MAX UNA PUÒ ESSERE 1

ESEMPIO. GRAPH COLORING



VARIABILI = {VA, NT, Q, SA, NSW, V}

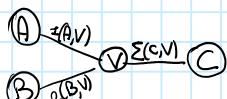
$\forall x, y : x \text{ CONNESSO } y \Rightarrow x \neq y$

DA CSP GENERICO A BINARIO

ESEMPIO: $A + B = C$

INTRODUO V AUSILIARIA: $D_V = D_A \times D_B$ [V È UNA TUPLA A 2]

INTRODUO VINCOLI EXTRA LEGENDA: $i(x, v)$ $x = V[j]$
 $\Sigma(x, v)$ $x = \sum_{k=0}^j V[k]$



METODI RISOLUTIVI

POTREI USARE DFS MA MEGLIO CONSTRAINT PROPAGATION [ASSEGNAZIONE \rightarrow VINCOLI \rightarrow ELIMINAZIONE VAL DA DOMINIO]

CONSISTENZA DI NODI [Pre-Processing]

SI APPLICA AI VINCOLI UNARI AD OGNI NODO ELIMINANDO DA OGNI NODO

ESITO:

- SE UN DOMINIO È VUOTO INSODDISFACENTE
- SE $\forall x \in D_x$ HA UN VAL RISOLTO
- E SE PROBLEMA SEMPLIFICATO

- SE UN DOMINIO È VUOTO Insoddisfacente
- SE $\forall x$ IL D_x HA UN VAL Risolto
- ELSE PROBLEMA SEMPLIFICATO

CONSISTENZA Archi

VINCOLI BINARI. Una var X_i è consistente per X_j se \exists val in D_i \exists in D_j un val che soddisfa il vincolo

Consistenza Percorso

Considera gruppo di 3 var

Dati A, B, C Assegnamento (A, B) consistente per vincoli tra $A \rightarrow B$ è val C :
 $A \rightarrow C$ cons
 $B \rightarrow C$ cons

Backtracking search/Min conflict

mercoledì 7 febbraio 2024 00:21

BACKTRACKING SEARCH [Algo CSP]

- Nodo di partenza: $T = \{\}$
- Estensione nodo: Assegna a x_i una var: $v \in D_i$
- Ricerca profondità
- L'assegnamento non conta
- Propagazione dei vincoli

Ordering Var & Val

FUNC SEL-VAR / SEL-VAL

h PER VARIABILI:

- Ordine lessico (statica)
- Random (dinamica)
- h di grado: Var. max vincoli con var. non assegnate
- Min val rimanenti: Var. min($D_{iH}(D_i)$)
- Least constraint value: Val minimizza l'esclusione dai domini adiacenti

FORWARD CHECKING

\forall assegnamento $x_i \leftarrow v$ consistenza ad Arcs $\forall x_j: x_j$ condivide vincolo x_i
MAINTAINING (AC-3 \forall assegnamento)

MIN CONFLICT

Idea:

- 1 Rnd var
- 2: Val: Minimizza i conflitti con il prossimo assegnamento

MDP

mercoledì 7 febbraio 2024 12:08

- No more Goal
- TIME Horizon
- STOCASTIC Ambient

Formalizzazione: $S = \{S_0, S_1, \dots, S_T\}$ SPAZIO FINITO DEGLI STATI

$S_0 \in S$ STATO INIZIALE

$S_T \in S$ STATO TERMINALE (LA SI TUTTA QUANDO SI FERMA L'EXE)

$A(s) = \{a_1, a_2, \dots, a_n\}$ AZIONI

Modello Transizione

Dato $s \in S$ e $a \in A \Rightarrow s' \in S$ = Prob Transizione $S \xrightarrow{s,a} S'$ [$P(s'|s, a)$]

$\forall s \in S$ AGENTE OTTIENE REWARD ADDITIVO $R(s, a, s')$

Orizzonte H quanti STEP DURA MDP

MARCOVIANITÀ

VEDIAMO UN PROCESSO STOCASTICO CON UNA SEQUENZA $s_0, s_1, s_2, \dots, s_t$ DOVE $s_t \in S$ IN TEMPO t

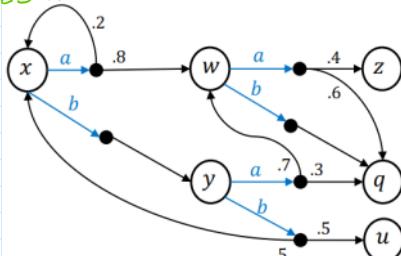
Proprietà di MARKOV Esito di un'azione dipende da $S_t: P(s_{t+1}|s_t, a_t) \rightarrow P(s_{t+1}|s_t, a_t)$

Quindi lo STATO CONTIENE TUTTE LE INFO PER PRENDERE LA BEST Decision

SOLUZIONE

CERCHIAMO UNA STRATEGIA [Policy], $\pi: S \rightarrow A$ ATTRIBUISCE UN a

ESEMPIO



STATO s	$\pi(s)$
x	b
w	a
z	-
y	a
q	-
u	-

DATA π DI MDP L'AGENTE:

- OSSERVA LO STATO s
- CONSULTA $\pi(s)$
- ESEGUE $\pi(s)$

BONTÀ π

- CONSENTE DI ARRIVARE GOAL
- ACCUMULA IL MINOR PESO

Prob

- NO GOAL - Reward Incerto

SOLUZIONE: MASSIMIZZARE VALORE ATTESO TRAMITE π^*

- π^* GENERA s_0, s_1, \dots, s_H CIOÈ $R_0, R_1, R_2, \dots, R_H$
- MASSIMIZZA $\sum_{k=0}^H R_k$

$$\pi^*: S \rightarrow A \quad \pi^* = \arg \max_{\pi} E_{\pi} [\sum_{k=0}^H R_k]$$

Reward scontato: PIÙ È LONTANO MENO FURO DE SO SCONTATO

$$U(s_0, s_1, \dots) = r_0 + \gamma r_1 + \gamma^2 r_2 + \dots = \sum_{k=0}^{\infty} \gamma^k r_k$$

Orizzonte FINITO/INFINITO

NEVER GO DEEPER THAN MAX DEEP
MA π^* NON È STAZIONARIA (Cambia in base al tempo)

Se H VALORE FINITO OGNI VOLTA CHE $H-t > 0$

IL FATTORE SCONTATO FA TENDERE LA SOMMATORIA A 0
 $\pi^*: S \rightarrow A$

DYNAMIC PROGRAMMING: P DIVENTA $P = (P_1, P_2, P_3, \dots, P_n)$
OTTIMALITÀ DI BELLMAN

DEFINIAMO π^*

$- V^*(s)$ VALORE S [SE DALLI IN AVANTI SEGUO π^* $V(s) = \max_{a,s'} P(s'|s,a) [R(s,a,s') + \gamma V^*(s')]$]

$- Q^*(s,a)$ VALORA COPPIA S & A

- AGENTE IN S, DECISO A
- DI TUTTE LE POLICY π^* SOMMA REWARD
- V SCORDATO

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

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

Prob S' | S, A Reward S, A Reward Futuro S' | S, A

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

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

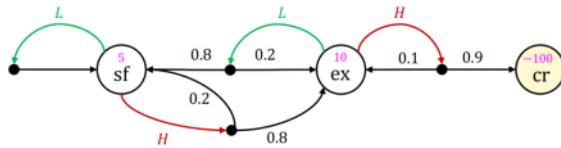
$$V^*(s) = \max_a \sum_{s'} P(s'|s,a) [R(s,a,s') + \gamma V^*(s')]$$

Value vs Policy Iteration

ESEMPIO

Running example

- $S = \{sf, ex, cr\}$, safe, exposed, critical (critical è terminale)
- $A = \{L, H\}$, low profile, high profile; applicabili solo negli stati di safe e exposed
- Reward: ogni transizione in sf vale 5, ogni transizione in ex vale 10, ogni transizione in cr vale -100



Interpretazione

- Se mantiene un profilo basso, l'agente resta al sicuro ma ottiene un reward basso
- Se rischia, adottando un profilo alto, si espone ma ottiene un reward alto
- Rischiare molto però può compromettere l'agente con una penalty elevata
- **Esempio:** attività di intelligence

s	a	s'	$P(s' s,a)$	$R(s,a,s')$
sf	L	sf	1	5
sf	L	ex	0	10
sf	L	cr	0	-100
sf	H	sf	0.2	5
sf	H	ex	0.8	10
sf	H	cr	0	-100
ex	L	sf	0.8	5
ex	L	ex	0.2	10
ex	L	cr	0	-100
ex	H	sf	0	5
ex	H	ex	0.1	10
ex	H	cr	0.9	-100
cr	L	sf	0	0
cr	L	ex	0	0
cr	L	cr	1	0
cr	H	sf	0	0
cr	H	ex	0	0
cr	H	cr	1	0

VALUE ITERATION

$$V_0(s) = 0 \quad \forall s \text{ AD ITERAZIONE } k+1 \text{ CALCOLA } V_{k+1}(s) \quad \forall s \text{ USANDO } V_k(s)$$

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} P(s'|s,a) [R(s,a,s') + \gamma V_k(s')]$$

CONVERGENZA VALUE ITERATION

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} P(s'|s,a) [R(s,a,s') + \gamma V_k(s')]$$

STRUTURA AD ALBERO $k \rightarrow k+1$



$V_{k+1}(s) > V_k(s)$

$V_{k+1}(s) \leq V_k(s)$

| REWARD OTTENGONO K+1-ESIME $\leq R_{MAX} \text{ e } \geq R_{MIN}$

POLICY EXTRACTION: $\pi^*(s) = \arg \max_a \sum_{s'} P(s'|s,a) [R(s,a,s') + \gamma V^*(s')]$

DIVARISSI [INEFFICIENTE]

APPROCCIO: ① π RANDOM

② Policy Evaluation: State Value Function $V^*(s)$ INDOTTA SU MDP

③ Policy Extraction: $V^*(s)$ APPENA CALCOLATA SI ESTRAE π^*

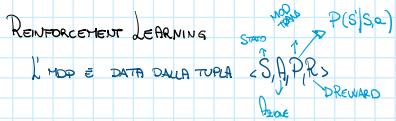
- Approccio:
- ① π RANDOM
 - ② Policy Evaluation: State Value Function $V^\pi(s)$ INQUITA SU MOP
 - ③ Policy Extraction: $V^\pi(s)$ APPENA CALCOLA SI ESTRAE π'
 - ④ $\pi' == \pi$
TRUE = Risolto
FALSE = Policy update $\pi = \pi'$

DATI π LA DEVO VALUTARE: $V^\pi(s)$ CHE π

$$V_{(s)}^\pi = Q(s, \underbrace{\pi(s)}_\alpha) + \sum_s P(s'|s, \alpha) [R(s, \alpha, s') + \gamma V^*(s')]$$

$$V_{k+1}^\pi(s) \leftarrow Q(s, \pi(s))$$

EXPECT-MAX



La TUPLA $\langle P, R \rangle$ SI CHIAMA MODELLO DI TRANSIZIONE + REWARD MODELLO

Ma se non avessimo il Modello Reinforcement Learning

Idea: Dato S_{start} non posso cercare π^* MA ESE UN ELEVATO NUMERO DI A POSSO APPROSSIMARE P, R E Poi Cercare π^*

EXPLORATION: Scelgo un'azione basandomi non sui R ma sulle info EXTRA che può dare

EXPLOITATION: Scelgo $A: \max(\hat{R})$ dato \hat{P}, \hat{R} trovati in EXPLORATION

RL PASSIVO.

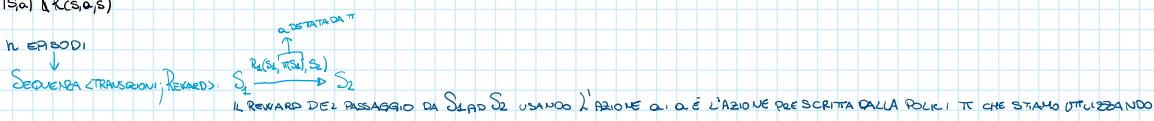
DATI UNA π FISSATA VOGLIO SAPERE LA BONTÀ DI π $[V^\pi(s)]$: $s =$ Stato raggiungibile con scatto y
 [Policy EVALUATION con $P(s'|s,a)$ $R(s,a,s')$]

ADAPTIVE DYNAMIC PROGRAMMING

Idea: ① Eszero
 ↓
 ② STIMO $P(s'|s,a)$ & $R(s,a,s')$ [Modello]

③ Policy Evaluation su $P(s'|s,a)$ ($R(s,a,s')$)

④ GENERA UN DATA-SET CON N EPISODI



⑤ STIMA MODELLO

$V(s,a)$ SI CALCOLANO: N VOLTE CHE IN S SI È ESEGUITA a $[\#(s,a)]$
 N VOLTE CHE DA S SI ARRIVA IN S' CON a $[\#(s,a,s')]$

$$\hat{P} = \frac{\#(s,a,s')}{\#(s,a)}$$

Modello transizione stimato

$$\hat{R} = V \text{ TRANSIZIONE SI SCOPRE } R(s,a,s') \text{ QUINDI } \hat{R}(s,a,s') = R(s,a,s')$$

$$\text{⑥ Policy Evaluation: } V^\pi(s) = Q(s, \pi(s)) = \sum \hat{P}(s|s, \pi(s)) [\hat{R}(s, \pi(s), s') + \gamma V^\pi(s')]$$

Questo Metodo È DETTO Model Based [Più episodi ho più \hat{P} -DP]

MA SE size(S) È ELEVATA STIMARE \hat{P} DIVENTA PROIBITIVO

Model Free

① Si genera DATASET con EXPLORATION = AOP

② Si estrae R ACCUMULATO A FINE EPISODIO $TotR_1^S, TotR_2^S, \dots, TotR_n^S$
REWARD ACCUMULATO
EPISODIO 1, 2, ... n

③ Divido la somma dei Tot. $\cdot \frac{K}{\# \text{VISITE}(S)}$ [MEDIA]

$$\hat{V}^\pi(s) = \frac{1}{k} \cdot \sum_i^K totR_i^S$$

[Più EPISODI HO PiÙ $\hat{V}^\pi(s)$ $\Rightarrow V^\pi(s)$]

Più Easy

Contro: - TENERE TRACCIA N VISITE V_S
 - STIMA V_S INDIPENDENTI [No More Bellman Update]

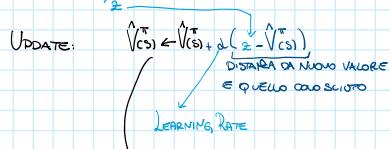
TEMPORAL DIFFERENCE:

Idea: UPDATE $\hat{V}^\pi(s)$ ITERATIVAMENTE

- Se l'agente si trova in s ha la stima $\hat{V}^\pi(s)$

- Ese $t(s)$ è ottenuto $s \rightarrow s'$ [uso questa transizione $s \rightarrow s'$ TO UPDATE $\hat{V}^\pi(s)$]

$$\hat{V}^\pi(s) = R + \gamma \hat{V}^\pi(s')$$



DISTANZA DA NUOVO VALORE
E QUELLO DAPOSSUTO

$$\hat{V}_{CS}^{\pi} \leftarrow \hat{V}_{CS}^{\pi} + \alpha (R_{CS, \pi, S'} + \gamma \hat{V}_{S'}^{\pi})$$

RIPETIZIONE ITERATIVA

$$\hat{V}_{CS}^{\pi} \leftarrow (1-\alpha) \hat{V}_{CS}^{\pi} + \alpha (R_{CS, \pi, S'} + \gamma \hat{V}_{S'}^{\pi})$$

$$\hat{V}_k^{\pi}(s) = (1-\alpha)\hat{V}_{k-1}^{\pi}(s) + \alpha z_k$$

$$\hat{V}_k^{\pi}(s) = (1-\alpha)[(1-\alpha)\hat{V}_{k-2}^{\pi}(s) + \alpha z_{k-1}] + \alpha z_k$$

$$\hat{V}_k^{\pi}(s) = (1-\alpha)[(1-\alpha)[(1-\alpha)\hat{V}_{k-3}^{\pi}(s) + \alpha z_{k-2}] + \alpha z_{k-1}] + \alpha z_k$$

$$\hat{V}_k^{\pi}(s) = (1-\alpha)[(1-\alpha)[(1-\alpha)[(1-\alpha)\hat{V}_{k-4}^{\pi}(s) + \alpha z_{k-3}] + \alpha z_{k-2}] + \alpha z_{k-1}] + \alpha z_k$$

$$\vdots$$

$$\hat{V}_k^{\pi}(s) = \underbrace{\alpha}_{\substack{\text{ITERAZIONE} \\ \text{RATE}}} \sum_{i=0}^{k-1} (1-\alpha)^i \underbrace{z_{k-i}}_{\substack{\text{REWARD} \\ \text{OSSERVATO}}} + (1-\alpha)^k \underbrace{\hat{V}_0^{\pi}(s)}_{\substack{\text{VALORE ELEVATO} \\ \text{AD ITERAZIONE}}} \quad \text{STIMA INIZIALE}$$

MEDIA MOBILE ESPONENZIALE: Più è grande il α meno è il peso

$$\hat{V}_{CS}^{\pi} = \alpha \sum_{i=0}^{k-1} (1-\alpha)^i \cdot z_{k-i}$$

VERSIONE FINALE TD

① $\hat{V}_{CS}^{\pi} = 0$ Inizializzo la stima a 0

② Osservo lo stato corrente, $s \in \pi(s)$, sette $R_{CS, \pi(s), S'}$; $s' \rightarrow [Se s' mai visto $\hat{V}(s') = 0$]$

③ $z = R_{CS, \pi(s), S'} + \gamma \hat{V}(s')$

④ Update $\hat{V}_{CS}^{\pi} = \hat{V}_{CS}^{\pi} + \alpha (z - \hat{V}(s))$

⑤ Go to 2

Converge? Si se α viene adattato e diventa più piccolo

ACTIVE REINFORCEMENT LEARNING

- Idea:
- ① STIMA $P(s'|s, a) \times R(s, a, s')$
 - ② STIMA π^*

Approccio Naïve: USO π_e PER ESPLOREZIONE \rightarrow STIMA \hat{P}, \hat{R} [come RL] \rightarrow STIMA π^*

Does not Work

2° Approccio Naïve: ESPLORO RND \rightarrow STIMA $\hat{P}, \hat{R} \rightarrow$ STIMA π^*

FUNZIONA MA PRENDE TANTO TEMPO $D[\pi^*; \hat{\pi}]$ REGRET

3° Approccio:

- ① SCELGO τ RND
- ② ESPLORO CON τ
- ③ STIMA \hat{P}, \hat{R}
- ④ π^* SU (\hat{P}, \hat{R})
- ⑤ $\pi = \pi^*$ GO TO 2

Pro: CONVERGE IN FRETTA

Contro: GREEDY; Non è garantita che $\pi = \pi^*$

PROBLEMA GREEDY

SOLUZIONE 1

- ① SCELGO τ RND
- ② ESPLORO CON PROB. A RANDOM E $(1-\epsilon)$ CON π E DEVE RIDURRE NEL TEMPO
- ③ STIMA \hat{P}, \hat{R}
- ④ π^* SU (\hat{P}, \hat{R})
- ⑤ $\pi = \pi^*$ GO TO 2

SOLUZIONE 2

Idea: SPOSTO TRADE-OFF DA PUNTO 2 A PUNTO 4

↓
DIVERGENT VALUE ITERATION

L'UTILITÀ VIENE ALTERATA AD $f(u, n) = u + \frac{V^*}{1+n} \rightarrow$ PARAMEETRO
 ↓
 VECCHIO REWARD

IL NUOVO BELLMAN UPDATE

$$V_{k+1}(s) \leftarrow \max_a \left(f \left(\sum_{s'} \hat{P}(s'|s, a) [\hat{R}(s, a, s') + \gamma V_k(s')] \right), \#(s, a) \right)$$

↓
 NON TRALASCIARE Q "CATTIVE"
 SENZA PROVARE

↓
 LASCA STORE
 LE CONFIDENT BAD A

LEARNING Policy Robuste

-RL-BAYESIANO
-CONTROLLO ROBUSTO

RL-BAYESIANO

AL POSTO DI UNA SOLA STIMA \exists TANTE POSSIBILI STIME

POSTERIOR P: DISTRIBUZIONE PROB SU $m_1, m_2, m_3 \dots$ (BELIEF)

DATA π, V_i^π Policy Evaluation V_m $\pi^* = \max_\pi \sum P_{(m)i} V_i^\pi$ COSTO ESAGERATO

CONTROLLO ROBUSTO

MIGLIOR DISPREZZIMMO

CONSIDERO WORST m_i : $\pi^* = \max_\pi (\min_{m_i} (V_i^\pi))$

IN MODO CHE SE È OTTIMA PER IL WORST È OTTIMA PER IL BEST

VOGLIO $\max(u) : u \leq V_i^\pi, \forall i$ IL MASSIMO DEI MINORI

ACTIVE MODEL FREE LEARNING

TEMPORAL DIFFERENCE APPLICATA AD ACTION VALUE $[Q(s,a)]$

$\pi^*(s) = \max(Q(s,a))$ Q-LEARNING

Q-LEARNING

- ① OSSEROV S, INIZIALIZZA $Q(s,a) = 0 ; \#(s,a) = 0 \quad \forall a$, SCEGLIO RND a
- ② ESE a; OSSERVA TRANS: VADO IN S' e $R(s,a,s')$
- ③ $\#(s,a)++$; SE S' MAI VISITATO $\Rightarrow Q(s',a') = 0 ; \#(s',a') = 0 \quad \forall a'$
- ④ UPDATE $Q(s,a)$
- ⑤ STEP 1

STEP 4: OSSERVAZIONE ($s \xrightarrow{a} s'$)

$$z = R + \gamma \max_a Q(s', a)$$

FATORE SCONTTO

$R(s,a,s')$ REWARD DA S' IN AVANT.

$$\text{UPDATE } Q(s,a) += \alpha(\#(s,a)) \underbrace{(z - Q(s,a))}_{\text{DISTANZA TRA OSSERVAZIONE}}$$

CONVERGE SE CON $t \rightarrow \infty$ OGNI (s,a) ESPLORATE ∞

IMPARO SENZA APPROPRIANDOMI SULLA TC MA NON LA USO PER VALUTARE: OFF-POLICY

⑤ SCELTA NEXT

$$\alpha \leftarrow \max_{\alpha'} f(Q(s', \alpha'), \#(s', \alpha'))$$

\downarrow e 5 non sono legata

ON POLICY
SARSA $[s, a, R, s']$ update Q-learning

$$Q(s, a) \leftarrow Q(s, a) + \alpha(\#(s, a)) [R + \gamma \max_{a'} (Q(s', a') - Q(s, a'))]$$

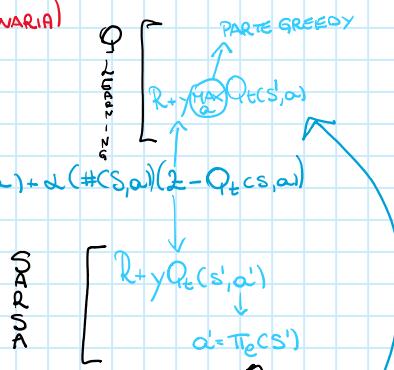
↑
TOLGO MAX
SCELGO a' CHE PRENDE IN s' PER DAVVERO

T UPDATE e EXPLORATION SONO LA STESSA Policy

Learning Policy (π_t) (Non-Stationaria)

UPDATE: $Q_t(s, a)$

$$Q_{t+1} = Q_t(s, a) + \alpha(\#(s, a))(R + \gamma \max_{a'} Q_t(s', a'))$$



On-Policy Vs Off-Policy

UPDATE SI BASA SU
 π_t

USA $\pi_t \neq \pi_t$: π_t è GREEDY

CONVERGENZA

Off-Policy: CONVERGE SEMPRE (per esplorazione infinita, $|E|$): $t \rightarrow \infty \quad \forall (s, a)$ esplorate ∞ volte

When $Q_t \rightarrow Q^*$ apprendimento ENDS π_t viene scartata e USA π_t greedy (π^*)

On-Policy: LA CONVERGENZA DI PENDE DA π_t

Dove essere GLE

$$\text{Greedy LIMIT} \xrightarrow{\pi_t \rightarrow \pi_t \text{ GREEDY}} \pi_t \rightarrow \pi^*$$

ESPLORAZIONE INFINTA $\rightarrow Q_t \rightarrow Q^*$

Quale uso?

Se ci interessa la performance in training?

si On-Policy

NO Off-Policy

LIMITI DI RL:
- ESPLORO MOLTO
- RAPPRESENTARE ESPlicitamente V e Q

LIMITI DI RL:

- ESPLORO MOLTO
- RAPPRESENTARE ESPlicitamente V e Q

Idea: GENERALIZZO LA SOLUZIONE [così non devo conoscere ogni stato
ma basta conoscere un subset delle
QUALITÀ DI S]
[cioè che ho imparato per s va bene per s'
 $s \in S$]

GENERALIZZARE:

$Q(s, a)$ viene frammentata $\theta_0 + \theta_1 f_1(s, a) + \theta_2 f_2(s, a) + \dots + \theta_m f_m(s, a)$

L'AGENTE NON CERCA DI STIMARE Q MA $\theta_0, \theta_1, \theta_2$ VETORE $\theta = \{\theta_0, \theta_1, \theta_2\}$

ESPlicito

USANDO TE CREO T (DATASET)

VRIGA

- TRANSIZIONE $s \xrightarrow{a} s'$
- $x_i = f(s, a), f_1(s, a), \dots, f_k(s, a)$ VETORE f
- $z_i = R + y^{\max}(s', a)$

Supervised Learning

$T = \langle (x_1, z_1), (x_2, z_2), \dots, (x_m, z_m) \rangle$

x_i è un vettore $\underbrace{\text{INPUT}}_{\text{OSSERVATORE}}$ z_i è uno scalare $\underbrace{\text{OUTPUT}}_{\text{OSSERVATO}}$

h approssima f e usiamo x_{new} senza avere output

Supervised Learning

$h(\text{POTESI})$ CLASSI: f LINEARI $z = \theta_0 + \theta_1 x_1 + \dots + \theta_m x_m$
POLINOMIALI DI GRADO N

SINUSOIDALI
ESPOZIONALI
ETC.

$$Q(s, a) = \theta_0 + \theta_1 s_1 + \dots + \theta_m s_m$$

APPROXIMAZIONE DI h
 $z = h(x) = \theta_0 + \theta_1 x_1 + \dots + \theta_m x_m$

REGRESSIONE LINEARE Multivariata

Devo minimizzare la perdita: $(z_i - h(x_i))^2$

RISOLUZIONE ESATA.

Ogni dato nel dataset in una bottiglia sola

$$E(\theta) = \frac{1}{N} \sum_{i=1}^N (z_i - h(x_i))^2$$

Lo voglio minimizzare

ERRORE

$$\begin{cases} \frac{\partial E(\theta)}{\partial \theta_0} = \frac{1}{N} \sum_{i=1}^N (z_i - \theta_0 - \theta_1 x_i)^2 \xrightarrow{\theta_0} -\frac{2}{N} \sum_{i=1}^N (z_i - \theta_0 - \theta_1 x_i) = 0 \\ \frac{\partial E(\theta)}{\partial \theta_1} = \frac{1}{N} \sum_{i=1}^N \frac{\partial}{\partial \theta_1} (z_i - \theta_0 - \theta_1 x_i)^2 \xrightarrow{\theta_1} -\frac{2}{N} \sum_{i=1}^N (z_i - \theta_0 - \theta_1 x_i) x_i = 0 \end{cases}$$

GRADIENTE

$$\begin{cases} \theta_0 = \frac{(\sum_{i=1}^N z_i - \theta_1 \sum_{i=1}^N x_i)}{N} \\ \theta_1 = \frac{N(\sum_{i=1}^N z_i x_i) - (\sum_{i=1}^N z_i)(\sum_{i=1}^N x_i)}{N(\sum_{i=1}^N x_i^2) - (\sum_{i=1}^N x_i)^2} \end{cases}$$

Secondo Metodo

ITERAZIONE UPDATE θ

ESEMPIO: $j < x_j, z_j$ e h

CALCOLO LOSS: $E(\theta) = \frac{1}{2} (z_j - h(x_j))^2$

\hookrightarrow LA DERIVATA $\frac{\partial E(\theta)}{\partial \theta_j} = \frac{1}{2} \frac{\partial (z_j - h(x_j))^2}{\partial \theta_j} = (z_j - h(x_j)) x_j$

$$\theta_j \leftarrow \theta_j - \alpha (z_j - h(x_j)) x_j$$

IL GRADIENTE SCENDE

θ-LEARNING Approssimato

① INIZIALIZZO I PARAMETRI $\theta_0, \dots, \theta_m$ E SCEGLIO AL A RDO

② ESEGUE S_{QPS}'

③ UPDATE $\theta_0, \dots, \theta_m$

$$\theta_i \leftarrow \theta_i - \alpha \frac{\partial E(\theta)}{\partial \theta_i} \rightarrow \theta_i + \alpha (z_j - Q(s, a)) f(s, a)$$

④ UPDATE α CON SECONDA

$$Residual + \gamma \max(Q', Q)$$