Search:

5 components: initial state, actions, transition-model, goal test, path cost

Tree search: Revisiting nodes is possible
→ good space complexity

Graph search: Explored set, no revisiting
→ good time complexity

Uninformed Search:

Criterion	BFS	UCS	DF	DL	ID
Complete	Yes[a]	Yes[a,b]	No	Yes	Yes[a]
Optimal	Yes[c]	Yes	No	No	Yes[c]
Time	$O(b^d)$	$O(b^{1+C^*/\varepsilon})$	$O(b^m)$	$O(b^l)$	$O(b^d)$
Space	$O(b^d)$	"	$O(bm)$	$O(bl)$	$O(bd)$

a: if b is finite — b: branching factor
b: if all step cost $\geq \varepsilon$ — d: depth of goal
c: all step cost identical — m: deepest node

Bidirectional Search: $b^{d/2} + b^{d/2} < b^d$
- goal must be well defined
- actions must be reversible

Informed Search:

GBF: $f(n) = h(n)$ — $h(n)$ is nonnegative and $h(goal) = 0$

Complete for graph search, tree search not
Optimality no, Time & space $O(b^m)$

A^*-Search: $f(n) = g(n) + h(n)$
Tree search: $h(n)$ admissible: $h(n) \leq g(n, goal)$
Graph search: $h(n)$ consistent: $\forall n\ h(n) \leq c(n, n') + h(n')$

Logical equivalences:

$\alpha \wedge (\beta \vee \gamma) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$

$\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$

$\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$ — $\alpha \vee \neg\alpha \equiv True$

$\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$

$\alpha \Rightarrow \beta \equiv \neg\alpha \vee \beta$

$\alpha \Leftrightarrow \beta \equiv (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$

Entailment: $\alpha \models \beta$ iff $M(\alpha) \subseteq M(\beta)$
Validity: α is valid iff $\alpha \equiv True$
Unsatisfiable: α is unsat. iff $\alpha \equiv False$

Conversion to CNF:
- Eliminate implications
- Move $\neg$ inwards
- Standardize variables
- Skolemization → get rid of $\exists$
- Drop $\forall$
- Distribute $\vee$ over $\wedge$

	Static environment	Dynamic environment
no actions	Bayesian networks	Hidden Markov Models
actions	decision networks (simple decisions)	Markov decision Process (complex decisions)

Constraint Satisfaction Problem:

CSP: Tuple (X, D, C)
Goal: Assign a value to each variable such that all C are satisfied

Convert n-Ary constraint into binary:
1. Replace constraint by new variable Z with domain of Z as a n-Tuple, which is restricted to satisfie constraint, e.g. ~~dom(Z) =~~
$dom(Z) = \{(z_1, z_2, z_3) \mid z_1 + z_2 = z_3 \wedge z_1 \in dom(A) \wedge z_2 \in dom(B), z_3 \in dom(C)\}$
2. Introduce new binary constraint to match the values of Z with the neighboring variables ($fst(Z) = A$)

Backtracking Search:

Variable Selection:
- Minimum Remaining Values: choose variable with smallest domain
- Degree Heuristic: Choose variable involved in highest number of constraints (highest degree)

Value Selection:
- Least Constraining Value: choose value that rules out the fewest choices for neighboring values

Inference techiques:
- Forward Checking: inconsistent values of neighboring variables are removed (after each assignment)
- Arc consistency algorithm: (after each assignment or as preprocessing: inconsistent values of all variables are removed

- after assigning variable X_j add each arc (X_i, X_j) to queue | queue removal (v_1, v_2)
- preprocessing: add all arcs to queue

if remove_inconsistent(X_i, X_j) then if size of Domain$(X_i) = 0$ {then return failure} for each X_k in Neighbors$[X_i] \setminus \{X_j\}$ add to queue

Inference Propositional Logic
- Forward- and Backward Chaining (only Horn Clause)
- Resolution (only CNF)

Horn clause: symbol or (conjunction) => symbol $(A \wedge B) \Rightarrow C$ ~~$(A \vee B) = C$~~ $A \vee$ ✓

Resolution:
$KB \models \alpha$ we show that $KB \wedge \neg\alpha$ is unsatisfiable
1. $KB \wedge \neg\alpha$ in CNF — $\boxed{\neg A}$ $\boxed{A \vee \neg B}$ → $\boxed{\neg B}$
2. Resolution Rule
3. - No new clauses to be added $\Rightarrow KB \not\models \alpha$
   - two clauses resolve to yield the empty clause $\Rightarrow KB \models \alpha$

Backward Chaining (First Order logic)
- Start with goal

eg. goals := $\{x_8 \leq y_8, y_8 \leq z_8\}$

$q' \leftarrow SUBST(\{x_8/7, z_8/3+9\}, x_8 \leq y_8)$
$= 7 \leq y_8$

Apply rule from KB that matches with q' — $\forall x_4\ x_4 \leq x_4 + 0$

$\theta' \leftarrow \{x_4/7, y_8/(x_4+0)\}$

new goals ← $\{y_8 \leq z_8\}$ (add goals from applied rule)

Probability:

Joint Probability: $P(X,Y) = P(X|Y)P(Y)$

Conditional Prob. $P(X|Y) = \frac{P(X,Y)}{P(Y)}$

Bayes' Theorem $P((Y=y)|(X=x)) = \frac{P((X=x)|(Y=y))P(Y=y)}{P(X=x)}$

$= \frac{\text{likelihood} \cdot \text{prior prob.}}{\text{evidence / normalization}} = \alpha P(X|Y)P(Y)$

$P(X,Y|Z) = P(X|Y,Z) \cdot P(Y|Z)$

Chain rule: $P(x_1,\dots x_n) = P(x_n|x_{n-1},\dots x_1) \cdot P(x_{n-1}|x_{n-2},\dots x_1),\dots,P(x_1) = \prod_{i=1}^{n} P(x_i|x_{i-1},\dots,x_1)$

Inference by Enumeration:

see graph up. right corner

Eg $P(B|j,m) = \alpha P(B,j,m) = \alpha \sum_e \sum_a P(B,e,a,j,m) = \dots$ Sum over all Hidden Variables

Inference by Variable Elimination:

storing results in f_i

$P(B|j,m) = \alpha \underbrace{P(B)}_{f_1(B)} \sum_e \underbrace{P(e)}_{f_2(E)} \sum_a \underbrace{P(a|B,e)}_{f_3(A,B,E)} \underbrace{P(j|a)}_{f_4(A)} \underbrace{P(m|a)}_{f_5(A)}$

-> Define axis for matrices to represent f_i:

$P(B|j,m) = \alpha f_1(B) \times \sum_e f_2(E) \times \sum_a f_3(A,B,E) \times f_4(A) \times f_5(A)$

$\times$ operator for pointwise product.

Sum out variables:

$f_6(B,E) = \sum_a f_3(A,B,E) \times f_4(A) \times f_5(A)$

-> Perform pointwise product, then sum over axis of A

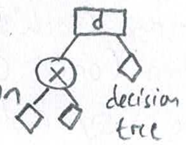
Simple Decisions:

$EU(a|e) = \sum_{s'} P(Result(a) = s'|a,e)U(s')$

If decision after a, sum needs to be split

$MEU(\alpha|e) = \max_a EU(a|e)$

↳ value of current best action



Optimal decision

$\pi^*(d_i|e) = \underset{d_i}{\operatorname{argmax}}\, EU(d_i|e)$

$MEU(d_{1:n}) = \max_{d_1} \sum_{x_1} \dots \sum_{x_n} \prod_{i=1}^{n} P(x_i|x_{1:i-1}, d_{1:i}) U(x_{1:n}, d_{1:n})$

Value of information := expected utility given the information at no charge − expected utility without the information

$VOI_e(E_j) = \left(\sum_k P(E_j = e_{jk}|e) MEU(\alpha_{e_{jk}}|e, E_j = e_{jk})\right) - MEU(\alpha|e)$

Learning:

$B(q) = -(q \log_2(q) + (1-q)\log_2(1-q))$

$B(0) = B(1) = 0 \quad B(\tfrac{1}{2}) = 1$

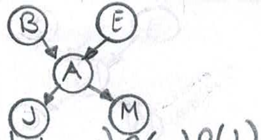
$Gain(A) = B\left(\frac{p}{p+n}\right) - \sum_{i=1}^{d} \frac{p_k + n_k}{p+n} B\left(\frac{p_k}{p_k+n_k}\right)$

A: Attribute taking d values

p and n number of positive and negative examples

p_k and n_k number of pos. and neg. examples of kth value of A

Bayesian Networks:

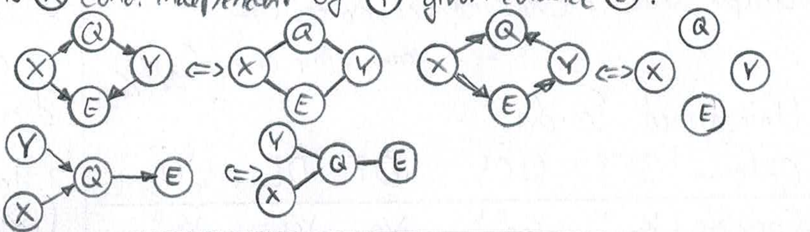


$P(x_1,\dots,x_n) = \prod_{i=1}^{n} P(x_i|parents(X_i))$

eg. $P(j,m,a,\neg b,\neg e) = P(j|a)P(m|a)P(a|\neg b,\neg e)P(\neg e)P(\neg b)$

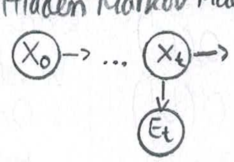
Graphical method for conditional independence

Is X cond. independent of Y given evidence E?



Hidden Markov Models:

$P(X_n = x_i) = \sum_{j=1}^{k} P(X_n = x_i|X_{n-1} = x_j)P(X_{n-1} = x_j)$



$\Leftrightarrow p_n = T p_{n-1}$ $\quad (p_i)_n = P(X_n = x_i)$

$T_{i,j} = P(X_n = x_i|X_{n-1} = x_j)$

dimension $|S| \times |S|^n$ ← order

↖ possible states of X

Observation model:

$P(E_t|X_t)$ $\quad (O_{i,j})_t = \begin{cases} P(e_t|X_t = x_i), & \text{if } i=j \\ 0 & \text{otherwise} \end{cases}$ $\quad P(x_t|e_{1:t})$

Filtering: $P(X_{t+1}|e_{1:t+1}) = \alpha P(e_{t+1}|X_{t+1}) \sum_{x_t} P(X_{t+1}|x_t) \cdot$

-> Matrix No. $(f_i)_{1:t} = P(X_t = x_i|e_{1:t})$

$f_{1:t+1} = \alpha O_{t+1} T^T f_{1:t} \qquad f_{1:1} = \alpha O_1 T p_0$

Prediction: $P(X_{t+k+1}|e_{1:t}) = \sum_{x_{t+k}} P(X_{t+k+1}|x_{t+k}) P(x_{t+k}|e_{1:t}) = T^{k+1} f_{1:t}$

Smoothing: $(0 \le k < t)$

$(b_i)_{k+1:t} \doteq P(e_{k+1:t}|X_k = x_i)\, b_{k+1:t} = T^T O_{k+1} b_{k+2:t}$

$P(X_k|e_{1:t}) = \alpha P(X_k|e_{1:k}) \times P(e_{k+1:t}|X_k) =$
$= \alpha f_{1:k} \times b_{k+1:t}$

Viterbi algorithm: with $\mu_{t+1} = \max_{x_1 \dots x_t} P(x_1,\dots x_t, X_{t+1}|e_{1:t+1})$

$\mu_{t+1}(X_{t+1}) = \alpha P(e_{t+1}|X_{t+1}) \max_{x_t} (P(X_{t+1}|X_t)\mu_t(X_t))$

$X_t = t \quad \mu_1(x_1) \xrightarrow{t_{11}} O_{11}\ \mu_2(x_2)$ $\qquad T = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}$

$X_t = f \quad \mu_1(\neg x_1) \xrightarrow{t_{22}} O_{22}\ \mu_2(\neg x_2)$ $\quad (t_{21}, t_{12})$

$\mu_2(X_2) = \alpha \begin{bmatrix} (O_{11})_2 \\ (O_{22})_2 \end{bmatrix} \times \begin{bmatrix} \max(t_{11} \cdot \mu_1(x_1), t_{12} \cdot \mu_1(\neg x_1)) \\ \max(t_{21} \cdot \mu_1(x_1), t_{22} \cdot \mu_1(\neg x_1)) \end{bmatrix}$

Init: $\mu_1(X_1) = f_{1:1}$

Making Complex Decisions:

Value Iteration: Bellman Update:

$U^{i+1}(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s,a) U^i(s')$

Optimal Policy: $\pi^*(s) = \underset{a \in A(s)}{\operatorname{argmax}} \sum_{s'} P(s'|s,a) U(s')$

Policy Iteration Policy Evaluation:

$U_i(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi_i(s)) U_i(s')$

Policy improvement:

$\pi_{i+1}(s) = \underset{a \in A(s)}{\operatorname{argmax}} \sum_{s'} P(s'|s,a) U_i(s')$

-> Split data on Attribute with highest gain

If all Examples are Positive (or negative)

↳ decision is yes (or no)