

## Search Algorithms

### • Informed Search Algos

- Greedy Best First Search -  $f(n) = h(n)$ . Complete in finite state spaces, but not optimal.
- A\* search -  $f(n) = g(n) + h(n)$ , where  $g(n)$  is path cost and  $h(n)$  is heuristic cost. Pick node with minimal value of  $f$  to be expanded.

$b$  is the branching factor;  $m$  is the maximum depth of the search tree;  $d$  is the depth of the shallowest solution, or is  $m$  when there is no solution;  $l$  is the depth limit.

<sup>1</sup> complete if is finite, and the state space either has a solution or is finite. <sup>2</sup> complete if all action costs are  $\geq \epsilon > 0$ , <sup>3</sup> cost-optimal if action costs are all identical; <sup>4</sup> if both directions are breadth-first or uniform-cost.

Criterion	BFS	UCS	DFS	DLS	IDS	Bi
Complete?	Yes <sup>1</sup>	Yes <sup>1,2</sup>	No	No	Yes <sup>1</sup>	Yes <sup>1,4</sup>
Optimal Cost?	Yes <sup>3</sup>	Yes	No	No	Yes <sup>3</sup>	Yes <sup>3,4</sup>
Time	$\mathcal{O}(b^d)$	$\mathcal{O}(b^{1+C^*/\epsilon})$	$\mathcal{O}(b^m)$	$\mathcal{O}(b^l)$	$\mathcal{O}(b^d)$	$\mathcal{O}(b^{d/2})$
Space	$\mathcal{O}(b^d)$	$\mathcal{O}(b^{1+C^*/\epsilon})$	$\mathcal{O}(bm)$	$\mathcal{O}(bl)$	$\mathcal{O}(bd)$	$\mathcal{O}(b^{d/2})$

### • Completeness & Optimality of A\* Search

- Admissible heuristic: one that never overestimates the cost to goal
- Consistent heuristic:  $\forall n$  and every successor  $n'$  of  $n$  generated by an action  $a$  we have  $h(n) \leq c(n, a, n') + h(n')$ .
- Admissible heuristics are guaranteed to return an optimal path
- Inadmissible heuristics *may* return optimal paths if the following conditions are satisfied
  - \* If there is even one cost-optimal path on which  $h(n)$  is admissible for all nodes  $n$  on the path, then that path will be found, no matter what the heuristic says for states off the path
  - \* If the optimal solution has cost  $C^*$ , and the second-best has cost  $C_2$ , and if  $h(n)$  overestimates some costs, but never by more than  $C^* - C_2$ , then A\* is guaranteed to return optimal cost solutions.

## Bayesian Networks

### • Syntax

- Links: For each node  $X_i$ 
  - \* Choose a minimal set of parents for  $X_i$  from  $\{X_1, \dots, X_{i-1}\}$
  - \*  $P(X_i | X_{i-1}, \dots, X_1) = P(X_i | \text{Parents}(X_i))$
  - \* For each parent, insert a link from parent to  $X_i$
  - \* Write down the CPTs or Conditional Probability Tables,  $P(X_i | \text{Parents}(X_i))$
- $\text{Parents}(X_i)$  directly influences  $X_i$ . No cycles and no redundancy by construction
- Causal mode: parents  $\leftrightarrow$  cause and child  $\leftrightarrow$  effect.  $P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{parents}(x_i))$

### • Conditional Independence in Bayesian Networks

- Each variable is conditionally independent of its non-descendants, given its parents.
  - \* Instead of full joint distribution, define a set of conditional independence properties.
  - \* The full joint distribution can be derived from those properties
- A variable is conditionally independent of all other nodes in the network, given its parents, children, and children's parents.

### • Efficient Representation of Conditional Distributions

- **Deterministic node:**  $X_i = f_i(\text{Parents}(X_i))$
- **Noisy-OR node:**  $P(X_i = 1 | \text{Parents}(X_i)) = 1 - \prod_{j: X_j = \text{true}} (q_j)$ , where  $q_j$  is the prob. that  $X_j$  is the cause of  $X_i$ 
  - \* All causes should be listed (use misc if some are unknown)
  - \* inhibition of each parent is independent of inhibition of any other parent
- Noisy logical relationships in which a variable depends on  $k$  parents can be described using  $\mathcal{O}(k)$  parameters instead of  $\mathcal{O}(2k)$  for the full conditional probability table

### • Inference in Bayesian Networks

- Inference: given some evidence, compute the posterior distribution over the unobserved variables
- **Exact inference:** compute the posterior distribution exactly
- Inference by enumeration: Sum up all the paths that are consistent with the evidence.  $P(X|e) = \alpha P(X, e) = \alpha \sum_y P(X, e, y)$ 
  - \* Variable elimination: basically memoization
  - \* pointwise product:  $f(X_1, \dots, X_j, Y_1, Y_k) \times g(Y_1, \dots, Y_k, Z_1, \dots, Z_l) = h(X_1, \dots, X_j, Y_1, \dots, Y_k, Z_1, \dots, Z_l)$
- **Approximation Inference:** They work by generating random events based on the probabilities in the Bayes net and counting up the different answers found in those random events. With enough samples, we can get arbitrarily close to recovering the true probability distribution—provided the Bayes net has no deterministic conditional distributions.

**Ablative Analysis:** Tries to explain the difference between some baseline performance (usually poorer) and current performance. Solution is to reduce complexity of the model by removing features one by one and see till we reach baseline performance.

## Probabilistic Reasoning over Time

- $X_t$ : State variables at time  $t$  which are not directly observed.
- $E_t$ : Evidence variables at time  $t$  which are directly observed.
- State Transition Model: How the world evolves.  $P(X_t|X_{0:t-1}) = P(X_t|X_{t-k:t-1})$
- Sensor / Observation Model: How the evidence variables get their values.  $P(E_t|X_{0:t}, E_{1:t-1}) = P(E_t|X_t)$
- Complete Joint Probability Distribution over all the variables, for any  $t$ .

$$P(X_{0:t}, E_{1:t}) = P(X_0) \prod_{k=1}^t P(X_k|X_{k-1})P(E_k|X_k)$$

- **Bayesian Network for Temporal Models** - Nodes: state and evidence variables, Edges: Indicate dependencies between nodes, Conditional Probability Tables: Time-homogeneous, i.e. same for all time
- **Inference in Temporal Models**
  - Filtering: Computing the belief state.  $P(X_t|e_{1:t})$ ; Prediction: Computing the belief state at a future time.  $P(X_{t+k}|e_{1:t})$
  - Smoothing: Computing the belief state at a past time.  $P(X_k|e_{1:t})$ , given all evidence up to time  $t$ , and  $0 \leq k < t$ .
  - Most likely explanation: The sequence of states that is most likely to have generated those observations  $\text{argmax}_{x_{1:t}} P(x_{1:t}|e_{1:t})$
- **Filtering and Prediction**
  - Filtering:  $P(X_{t+1}|e_{1:t+1}) = \alpha P(e_{t+1}|X_{t+1}) \sum_{x_t} P(X_{t+1}|x_t)P(x_t|e_{1:t})$  where  $\alpha$  is the normalization constant, made to ensure that the sum of probabilities is 1.
  - Prediction: Can be viewed as filtering without evidence.  $P(X_{t+k+1}|e_{1:t}) = \sum_{x_{t+k}} P(X_{t+k+1}|x_{t+k})P(x_{t+k}|e_{1:t})$
  - If you keep predicting further and further into the future, the probability of the state converges to a stationary distribution.  $P(X_{t+k+1}|e_{1:t}) = P(X_{t+k}|e_{1:t})$
- **Kalman Filters**
  - If the current distribution  $P(X_t|e_{1:t})$  is Gaussian and the transition model  $P(X_{t+1}|x_t)$  is linear-Gaussian, then  $P(X_{t+1}|e_{1:t}) = \int_{x_t} P(X_{t+1}|x_t)P(x_t|e_{1:t})dx_t$
  - If prediction  $P(X_{t+1}|e_{1:t})$  is Gaussian and the sensor model  $P(e_{1:t+1}|X_{t+1})$  is linear-Gaussian, then  $P(X_{t+1}|e_{1:t+1}) = \alpha P(e_{t+1}|X_{t+1})P(X_{t+1}|e_{1:t})$
  - If the current distribution is Gaussian and the transition model is linear-Gaussian, then prediction is  $P(X_{t+1}|e_{1:t}) = \int_{x_t} P(X_{t+1}|x_t)P(x_t|e_{1:t})dx_t$ .  $P(x_{t+1}|x_t) = \mathcal{N}(x_t + 1; Fx_t, \Sigma_x)$
  - Kalman Filters a big deal for inferences to be tractable.

## Utility Theory

- $A \succsim B$ :  $A$  is preferred to  $B$  or is indifferent to  $B$ ,  $A \sim B$ :  $A$  is indifferent to  $B$ ,  $A \succ B$   $A$  is preferred to  $B$ . Lotteries:  $[p, A; 1-p, B]$
- Axioms
  - Orderability: Exactly one of  $A \succ B$ ,  $B \succ A$ , or  $A \sim B$  holds
  - Transitivity: If  $A \succ B \wedge B \succ C \implies A \succ C$
  - Continuity: If  $A \succ B \succ C$ , then  $\exists p \in [0, 1]$  s.t.  $pA + (1-p)C \sim B$
  - Substitutability: If  $A \sim B$ , then  $A \succsim C \iff B \succsim C$
  - Monotonicity: If  $A \succ B \implies (p > q \iff [p, A; 1-p, B] \succ [q, A; 1-q, B])$
  - Decomposition:  $[p, A; 1-p, [q, B; 1-q, C]] \sim [p, A; (1-p)q, B; (1-p)(1-q), C]$
- **Utility Function and Expected Utility**
  - Existence of  $U$ :  $U(A) > U(B) \iff A \succ B$  and  $U(A) = U(B) \iff A \sim B$
  - Expected Utility:  $U([p_1, A_1; \dots; p_n, A_n]) = \sum_{i=1}^n p_i U(A_i)$
  - Utility function may not be unique
- **Multiattribute Utility Functions**
  - Strict Dominance:  $A \succ B$  for all attributes
  - Stochastic Dominance: If  $A_1 = S_1$  stochastically dominates  $A_2 = S_2$ , then for any monotonically nondecreasing utility function  $U(x)$ , the expected utility of  $A_1$  is at least as high as the expected utility of  $A_2$
- **Decision Networks**
  - Chance Nodes(ovals): Random variables
  - Decision Nodes(rectangles): Choice of Actions
  - Utility Nodes(diamonds): Value Nodes, Function of it's parents
  - Simple extension of Bayesian Networks
  - Sequential Decision Making
    - \* Utility over a finite horizon  $U_h([s_0, a_0, \dots, s_{N+k}]) = U_h([s_0, a_0, \dots, s_N])$
    - \* Utility over an infinite time horizon  $U([s_0, a_0, \dots]) = \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) \leq R_{\max}/(\gamma - 1)$
    - \* Policy  $[a = \pi(s_t)]$ : the function that generates the action, given the state
    - \* Expected utility of a policy:  $U^\pi(s) = E[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t), s_{t+1})]$

- Univariate Gaussian Distribution:  $N(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
- Multivariate Gaussian Distribution:  $N(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$ , where  $\Sigma$  is the covariance matrix, with  $\sigma_x^2$  and  $\sigma_y^2$  on the diagonal and 0 elsewhere.