

CS 486/686

Course Recap

Yuntian Deng

Lecture 22

Putting the term together: search, uncertainty, decisions, learning

The course in one picture



Search I – uninformed

Problem formulation

- States, actions, transition model, initial & goal states, path cost.
- Search tree explores states by expanding the **frontier**.

Pick the data structure for the frontier → pick the algorithm.

DFS vs BFS vs IDS

- **DFS**: stack → $O(bm)$ space, not complete.
- **BFS**: queue → complete, $O(b^d)$ time and space.
- **IDS**: DFS in increasing depth → BFS-completeness, DFS-space.

IDS combines the best of both.

Search II – heuristic + A*

The three frontier rules

- **LCFS:** $\min \text{cost}(n)$.
- **GBFS:** $\min h(n)$.
- **A*:** $\min f(n) = \text{cost}(n) + h(n)$.

Heuristic properties

- **Admissible** ($h \leq h^*$) \Rightarrow A* is optimal with multi-path pruning off.
- **Consistent** ($h(n) \leq c(n, n') + h(n')$) \Rightarrow safe to use multi-path pruning.
- Consistent \Rightarrow admissible (not the other way around).

Search strategy summary

Strategy	Frontier choice	Complete?	Space	Time
Depth-first	Last added	No	Linear	Exp
Breadth-first	First added	Yes	Exp	Exp
Lowest-cost first	min cost(n)	Yes	Exp	Exp
Greedy best-first	min $h(n)$	No	Exp	Exp
A*	min cost(n) + $h(n)$	Yes	Exp	Exp

A* dominates when the heuristic is informative enough to prune most of the tree.

CSPs + local search

Constraint satisfaction

- Generate-and-test is exponential — exploit problem structure.
- **Backtracking:** incremental, prune as soon as a constraint fails.
- **Arc consistency / AC-3:** shrink domains before/during search; complexity $O(cd^3)$.

Local search

- Trade completeness for very large state spaces.
- **Greedy descent / hill-climbing:** always move to best neighbour.
- Beats local minima with **random restarts + simulated annealing** (accept worse moves with prob $e^{-\Delta/T}$).

Probability + independence

The four rules

- **Product:** $P(A, B) = P(A | B) P(B)$
- **Sum:** $P(A) = \sum_b P(A, B=b)$
- **Chain:** $P(X_1, \dots, X_n) = \prod_i P(X_i | X_{<i})$
- **Bayes:** $P(H | e) \propto P(e | H) P(H)$

Independence


- **Unconditional:** $P(A, B) = P(A) P(B)$.
- **Conditional:** $P(A, B | C) = P(A | C) P(B | C)$.
- Tells us which terms drop out of the chain rule — the key to scaling up inference.


Bayesian networks + d-separation


Construction recipe

- Pick a variable order (causal often works best).
- Add each node; choose the smallest set of parents that makes it conditionally independent of the rest.
- Joint factorizes as $P(X_1, \dots, X_n) = \prod_i P(X_i \mid \text{pa}(X_i))$.

3 d-separation patterns

 **Chain** $A \rightarrow B \rightarrow C$: blocked by observing B .

 **Common cause** $A \leftarrow B \rightarrow C$: blocked by observing B .

 **Common effect** $A \rightarrow B \leftarrow C$: blocked by *not* observing B or descendants.

A path is blocked \Rightarrow the endpoints are conditionally independent.

Inference – VE & HMM filtering

Variable elimination

- **Define** a factor for each CPT.
- **Restrict** factors to evidence.
- **Multiply** factors sharing a variable; **sum out** the hidden one.
- **Normalize** to get a posterior.

HMM filtering

- Recursion: $P(S_t | o_{0:t}) \propto P(o_t | S_t) \sum_{s_{t-1}} P(S_t | s_{t-1}) P(s_{t-1} | o_{0:t-1})$
- One forward pass: $O(t \cdot |S|^2)$ time.

Decision networks

Three node types

- **Chance** nodes — like a Bayes net (CPTs).
- **Decision** nodes — rectangles, no CPT, chosen by us.
- **Utility** node — diamond, deterministic function of its parents.

Maximum expected utility

- Policy $\pi(d \mid \text{parents})$; MEU picks π^* maximizing $\mathbb{E}[U]$.
- Evaluate by enumeration or by running VE on the DN.
- **Value of information:** EU gain from observing a variable before deciding.

Markov decision processes

Setup

- States, actions, transition $P(s' | s, a)$, reward $R(s)$, discount γ .
- Policy $\pi : S \rightarrow A$; value $V^\pi(s) = \mathbb{E}[\sum_t \gamma^t R(s_t) | \pi]$.

Bellman + the two algorithms

- **Bellman optimality:** $V^*(s) = R(s) + \gamma \max_a \sum_{s'} P(s' | s, a) V^*(s')$.
- **Value iteration:** apply the Bellman update until convergence.
- **Policy iteration:** evaluate (linear system)
↔ improve (greedy) — converges fast.

Reinforcement learning

Two learning regimes

- **Passive ADP:** follow a fixed policy, estimate \hat{P} , \hat{R} , solve via PE.
- **Active ADP:** also pick actions — balance exploration vs exploitation (ϵ -greedy, softmax, optimistic init).

Model-free updates

- **Q-learning** (off-policy): $Q(s, a) \leftarrow Q(s, a) + \alpha [R + \gamma \max_{a'} Q(s', a') - Q(s, a)]$.
- **SARSA** (on-policy): use the action actually taken instead of $\max_{a'}$.
- ADP converges fast but needs a model; Q-learning is slower but model-free.

Supervised + unsupervised learning

Supervised

- **Classification** (discrete y) vs **regression** (continuous y).
- **Bias-variance**: underfit (high bias) ↔ overfit (high variance).
- **Cross-validation**: K -fold to pick the model complexity that minimizes held-out error.
- **ERM**: minimize empirical loss

$$\frac{1}{N} \sum_i \ell(\hat{y}^{(i)}, y^{(i)}).$$

Unsupervised

- **k-means**: alternate assign ↔ recompute centroids; pick k via elbow / silhouette.
- **PCA**: project onto top eigenvectors of the covariance matrix; maximize variance.
- **Autoencoders & GANs**: learn representations / generators with neural nets.

Decision trees + neural networks

Decision trees

- Greedy top-down split with **ID3**.
- Choose split that maximizes **information gain** $I = H(Y) - H(Y | X)$.
- Stop when pure / no features / no examples.
- Interpretable, great for small tabular data.

Neural networks

- Weighted sums + non-linear activations (sigmoid, ReLU).
- **Forward pass** computes loss; **backprop** computes gradients via chain rule (δ recursion).
- Train with **SGD** + momentum / Adam.
- Black box but unmatched for images, audio, text.

Notation cheat sheet

Symbols that recur across the four modules. Watch for the two collisions on the right.

Symbol	Meaning	First seen
γ	Discount factor for future rewards	L13 (MDPs)
π, π^*	Policy / optimal policy	L12–L15
$V(s), Q(s, a)$	State / state-action value functions	L13–L15
$H(p), IG$	Entropy / information gain ($IG = H_{\text{before}} - H_{\text{after}}$)	L6 (entropy) · L18 (IG)
\mathcal{H}	Hypothesis class (ERM)	L16
Σ	Covariance matrix (PCA)	L17
α	RL TD step size (L15); momentum coefficient (L21) — <i>different meanings</i>	L15 / L21
ε	Exploration probability in ε -greedy (L15); learning rate in SGD (L21) — <i>different meanings</i>	L15 / L21

Worth a row on your two-sided study sheet.

Final exam – logistics

What to bring

- Non-programmable calculator.
- One two-sided A4 study sheet (handwritten or typed).

Format (tentative)

- Around 8 problems spanning all four modules.
- ~84 marks total; 76+ marks count as 100%.
- Pass threshold: 42 marks.

Exact details will be confirmed on the course page before exam day.

Thanks for a great term — good luck on the final!