# Artificial Intelligence

CS 165A

Mar 12, 2020

Instructor: Prof. Yu-Xiang Wang

Today

→ Review

→ Responsible AI

# Announcement

- HW4 due on Friday 23:59 pm
  - We will grade on Saturday to provide feedback.

- Final: Next Monday 12:00 pm. Take home final on gradescope.
  - Open book. NO collaboration allowed. Your questions might be different from your peers.
  - Covers topics up to First Order Logic (but before FOL inference)
  - About 80% will be on topics after the midterm, 20% on earlier topics. (Note that you might be asked to apply ML or PGM on topics in the second half of the lecture!)
  - An additional 30 min will be given to everyone since you need to scan your solutions and upload to gradescope. So you have until 3:30 pm to submit.

# Tips for studying for the final

- Focus on the Lectures and HWs

- For any concepts that you are confused, check the textbook (Again, books are random access, you don't have to read chapters from the beginning to the end)
  - Stick to AIMA book for Search and Logic
  - Stick to the Sutton and Barto book for RL.

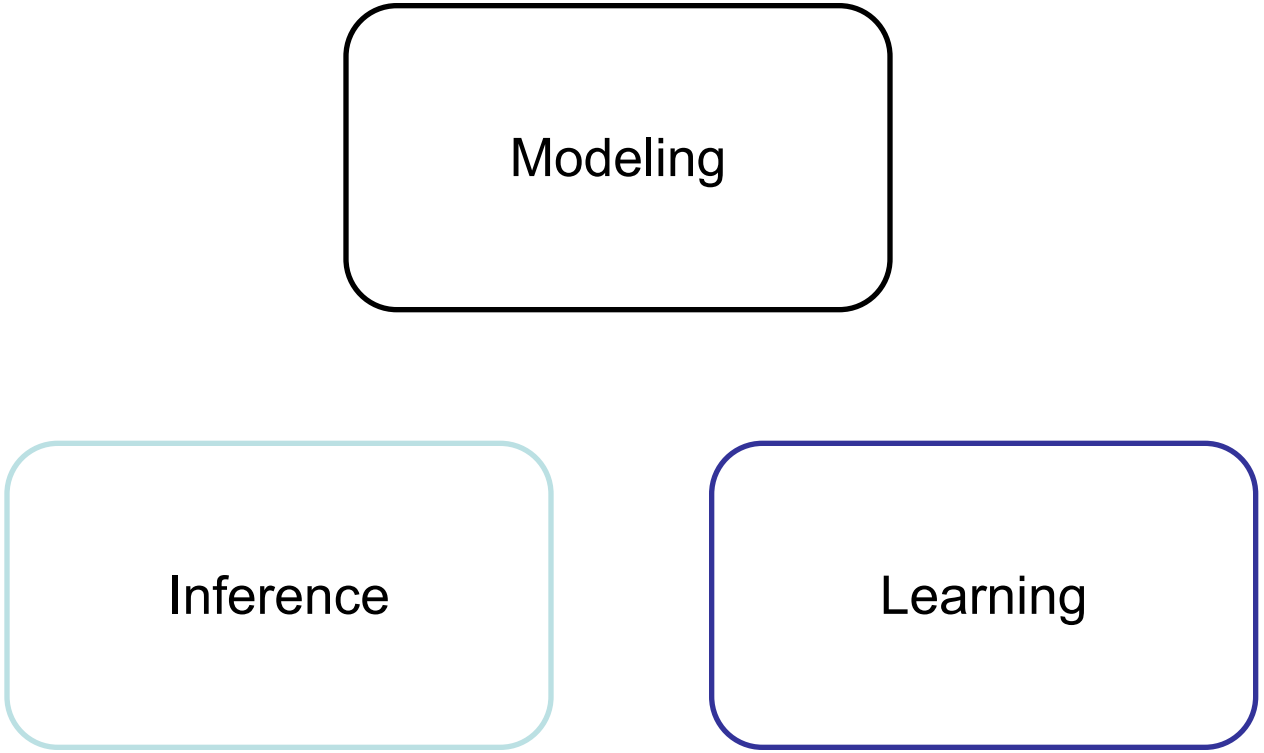
# We have come a long way

| Week | Topic | |
|---|---|---|
| 1 | Introduction and Course Overview<br>AI Problem Solving and Intelligent Agents | AI Overview |
| 2 | Machine Learning<br>Machine Learning | Machine Learning |
| 3 | Probabilistic Graphical Models<br>Probabilistic Graphical Models | Probabilistic Reasoning |
| 4 | Search: Problem solving with search<br>Search: Uninformed search algorithms | Search |
| 5 | Search: Informed search algorithms<br>Midterm Review | |
| 6 | Midterm<br>Search: Minimax Search and game playing | |
| 7 | RL: Markov Decision Processes<br>RL: Markov Decision Processes | Reinforcement Learning |
| 8 | RL: RL algorithms<br>RL: RL algorithms | |
| 9 | Logic: Logical agents<br>Logic: Propositional Logic / Inference | Logic |
| 10 | Logic:  First order Logic<br>Responsible AI: Review session | |
| 11 | **Final Exam**. 12:00 PM - 3:30 PM | |

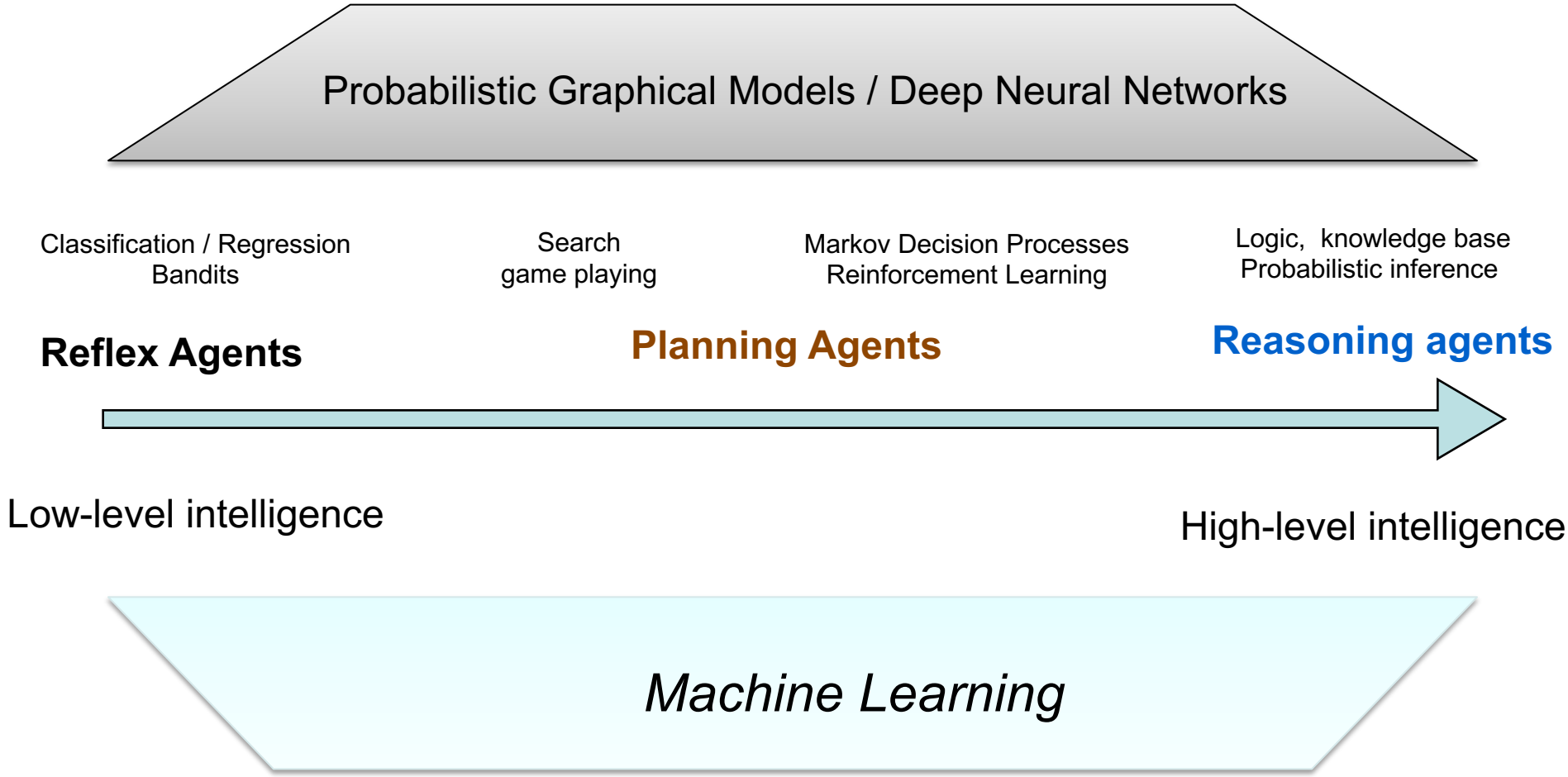
# Lecture 1-2: AI Overview

- Strong AI / Weak AI
- Turing Test
- AI for problem solving
- Rational agents
- Examples of AI in the real world
- Modelling-Inference-Learning Paradigm

# Modeling-Inference-Learning



# Structure of the course




(Again this idea is adapted from Percy Liang's teachings)

# Our view of AI

- So this course is about designing rational agents
  - Constructing $f$
  - For a given class of environments and tasks, we seek the agent (or class of agents) with the "best" performance
  - Note: Computational limitations make complete rationality unachievable in most cases
- In practice, we will focus on problem-solving techniques (ways of constructing $f$), not agents per se

# Different Ways of Looking at the AI

- Agent types / level of intelligence
  - Low-level: Reflex agents
  - Mid-level: Goal-based / Utility-based agents: planning agents
  - High-level: Knowledge-based: Logic agents

- Optimization view
  - Everything is an optimization problem

- Theoretical aspects
  - Time/space complexity
  - Algorithms and data structures
  - Statistical properties: # of free parameters / how easily can we learn them with data

# Optimization perspective of AI

- A rational agent $\max_{a_1,\ldots,a_T} \text{Utility}(a_1, \ldots, a_T)$
  - **Modelling tools:** Features / Hypothesis, PGM, State-space abstraction, agent categorization
  - **Constraints:** Computation, Data, Storage
- Discrete optimization $\min_{p \in \text{Paths}} \text{Distance}(p)$
  - **Algorithmic tool**: Search / Dynamic programming
- Continuous optimization $\min_{\theta \in \mathbb{R}^d} \text{TrainingError}(\theta)$
  - **Algorithmic tool**: Gradient descent / Stochastic gradient descent

# Different objectives to optimize in AI (first half of the course)

- PGM:
  - MLE: Maximize the log-likelihood function
  - Classifier / decision: max the posterior distribution

- Search and planning:
  - Find valid solutions with smallest path cost.

- Machine Learning
  - (Regularized) Empirical Risk Minimization (ERM):

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(\theta, (x_i, y_i)) + \lambda r(\theta)$$

# Different objectives to optimize in AI (second half of the course)

- Minimax Search / game playing
  - Maximize your worst-case pay-off (assuming your opponent plays optimally)

- Markov Decision Processes / RL
  - Maximize the cumulative expected reward of "decision policy"

- Logic / Knowledge based agent:
  - Solve a feasibility problem, find a "proof".
  - Determining "Valid, Satisfiable, Unsatisfiable"

# A lot of these problems are computationally / statistically hard, but so what?

- Get help from human:
  - Use model
  - Use abstractions at the right level
  - Use features
  - Use heuristic functions

- Get help from mathematics and computer science theory:
  - Solve an approx. solution
  - Approximate inference of a PGM
  - More iterations with less accuracy per SGD
  - A* Search
  - TD learning and Bootstrapping

# Lecture 3-4: Machine Learning

- Machine learning
  - Types of machine learning models
  - Focus on Supervised Learning ---- classifier agents.
- What is a feature vector?
  - Feature engineering,  feature extraction
- Hypothesis class and free parameters
  - How many are there? How to evaluate a classifier?
  - Error? On training data or on new data?
  - Overfitting,  underfitting?
- How to learn  (optimize)?
  - Surrogate losses, Gradient Descent, SGD

# Example of a feature vector of dimension 4

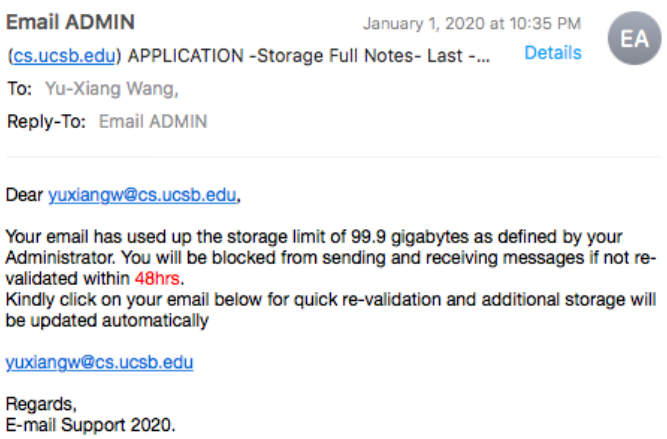
Contains hyperlinks

Proportion of misspelled words

| 1 | 0 | 0.0375 | 80 |
|---|---|---|---|

Whether the contact list

Length of the message



Email ADMIN January 1, 2020 at 10:35 PM

(cs.ucsb.edu) APPLICATION -Storage Full Notes- Last -... Details

To: Yu-Xiang Wang,

Reply-To: Email ADMIN

Dear yuxiangw@cs.ucsb.edu,

Your email has used up the storage limit of 99.9 gigabytes as defined by your Administrator. You will be blocked from sending and receiving messages if not re-validated within 48hrs.
Kindly click on your email below for quick re-validation and additional storage will be updated automatically

yuxiangw@cs.ucsb.edu

Regards,
E-mail Support 2020.

**Step 1 in Modelling**

**Feature extractor**:

Converting the object of interest to a vector of numerical values.

# Example: Linear classifiers

- Score(x) = $w_0 + w_1 * 1(\text{hyperlinks}) + w_2 * 1(\text{contact list}) + w_3 * \text{misspelling} + w_4 * \text{length}$

- A linear classifier: h(x) = 1 if Score(x) > 0 and 0 otherwise.

- Question: What are the "free-parameters" in a linear classifier?
  - If we redefine $\mathcal{Y} = \{-1, 1\}$
  - A compact representation:

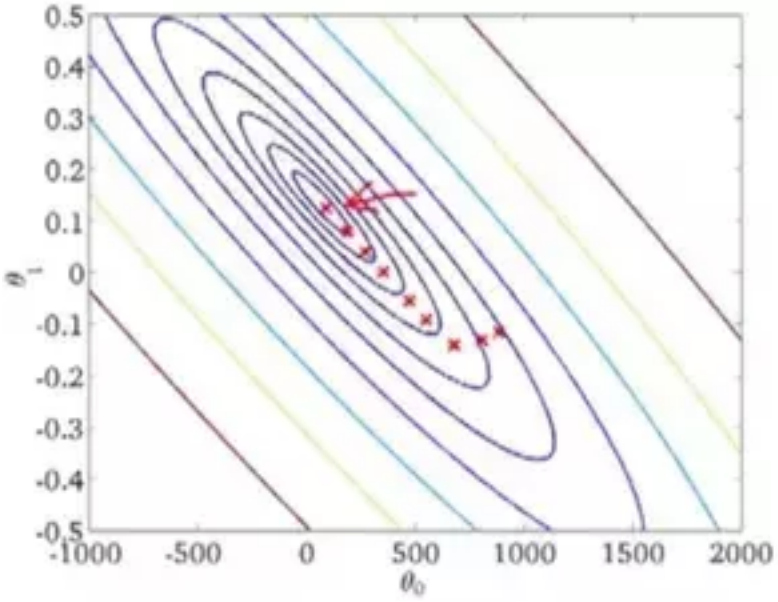
$$h(x) = \text{sign}(w^T [1; x])$$

# Just "relax": relaxing a hard problem into an easier one

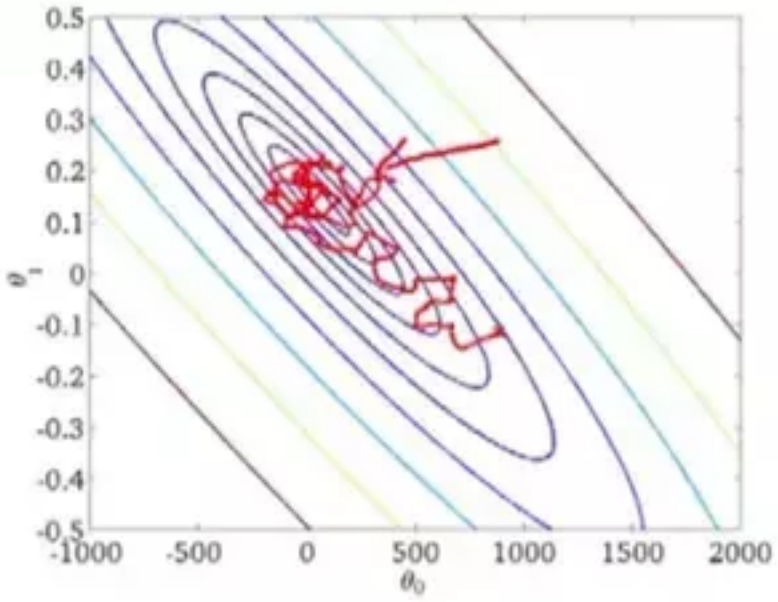
$$\min_{w \in \mathbb{R}^d} \text{Error}(w) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(\text{sign}(w^T x_i) \neq y_i)$$

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(w^T x_i, y_i).$$

# Illustration of GD vs SGD




**Batch Gradient Descent**




**Stochastic Gradient Descent**

**Observation:** With the time gradient descent taking one step.

SGD would have already moved many steps.

# One natural stochastic gradient to consider in machine learning

- Recall that

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(\theta, (x_i, y_i))$$

- Pick a **single** data point i uniformly at random
  - Use $\nabla_\theta \ell(\theta, (x_i, y_i))$
  - Show that this is an unbiased estimator!
  - Know which part of the expression is random!
  - Know how to apply the definition of expectation.

# Lecture 5-7: Probabilities and BayesNet

- Modelling the world with a joint probability distribution
  - Number of parameters?
- CPTs
  - Count number of independent numbers to represent a CPT
- Conditional, Marginal, Probabilistic Inference with Bayes Rule
- Read off conditional independences from the graph
  - d-separation
  - Bayes ball algorithm
  - Markov Blanket

# Tradeoffs in our model choices

**Fully Independent**

$P(X_1, X_2, \ldots, X_N)$

$= P(X_1)\, P(X_2) \ldots P(X_N)$

$O(N)$

**Fully general**

$P(X_1, X_2, \ldots, X_N)$

$O(e^N)$

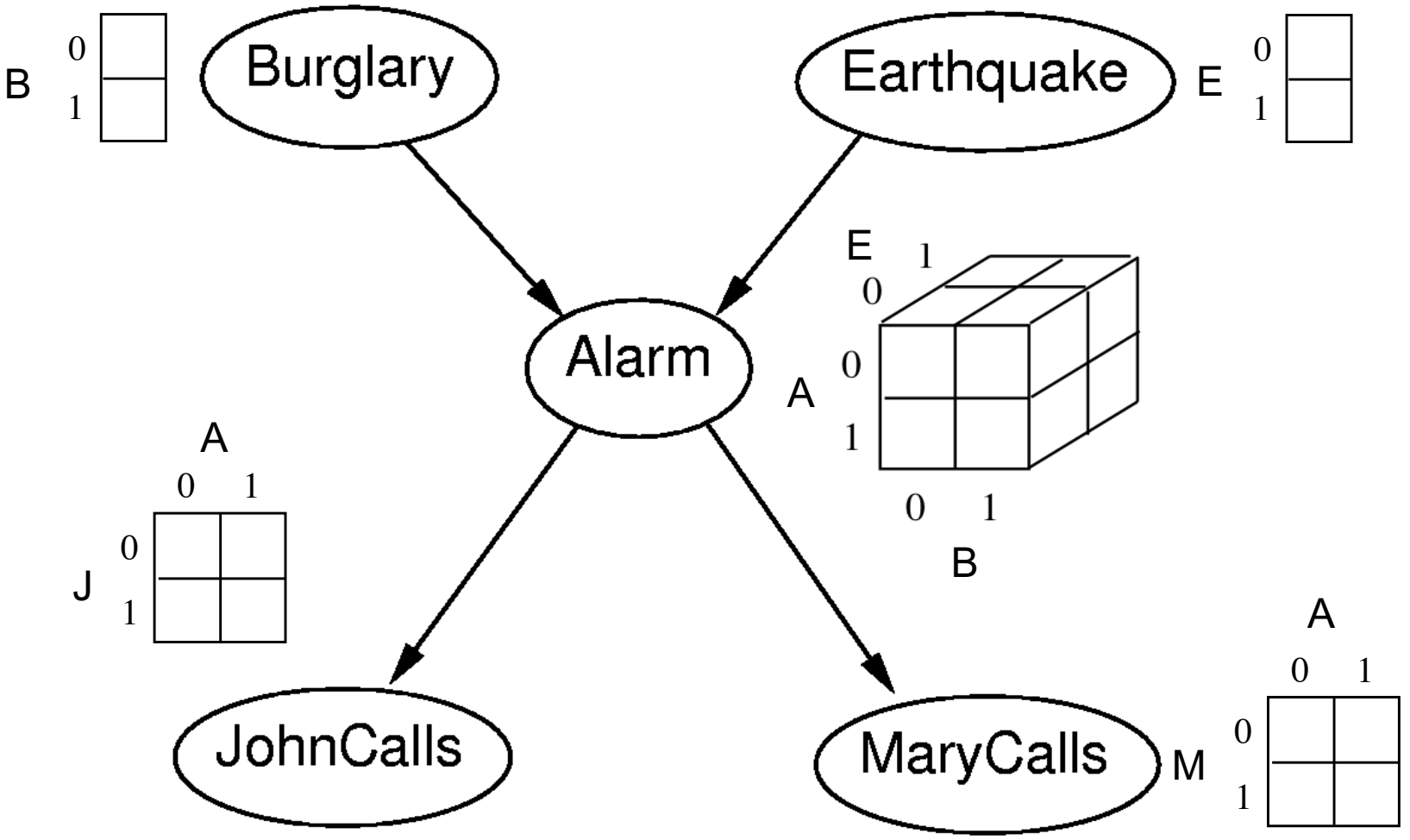
**Space / computation efficiency**

**Expressiveness**

**Idea:**

1. Independent groups of variables?
2. **Conditional independences?**

# What are the CPTs? What are their dimensions?

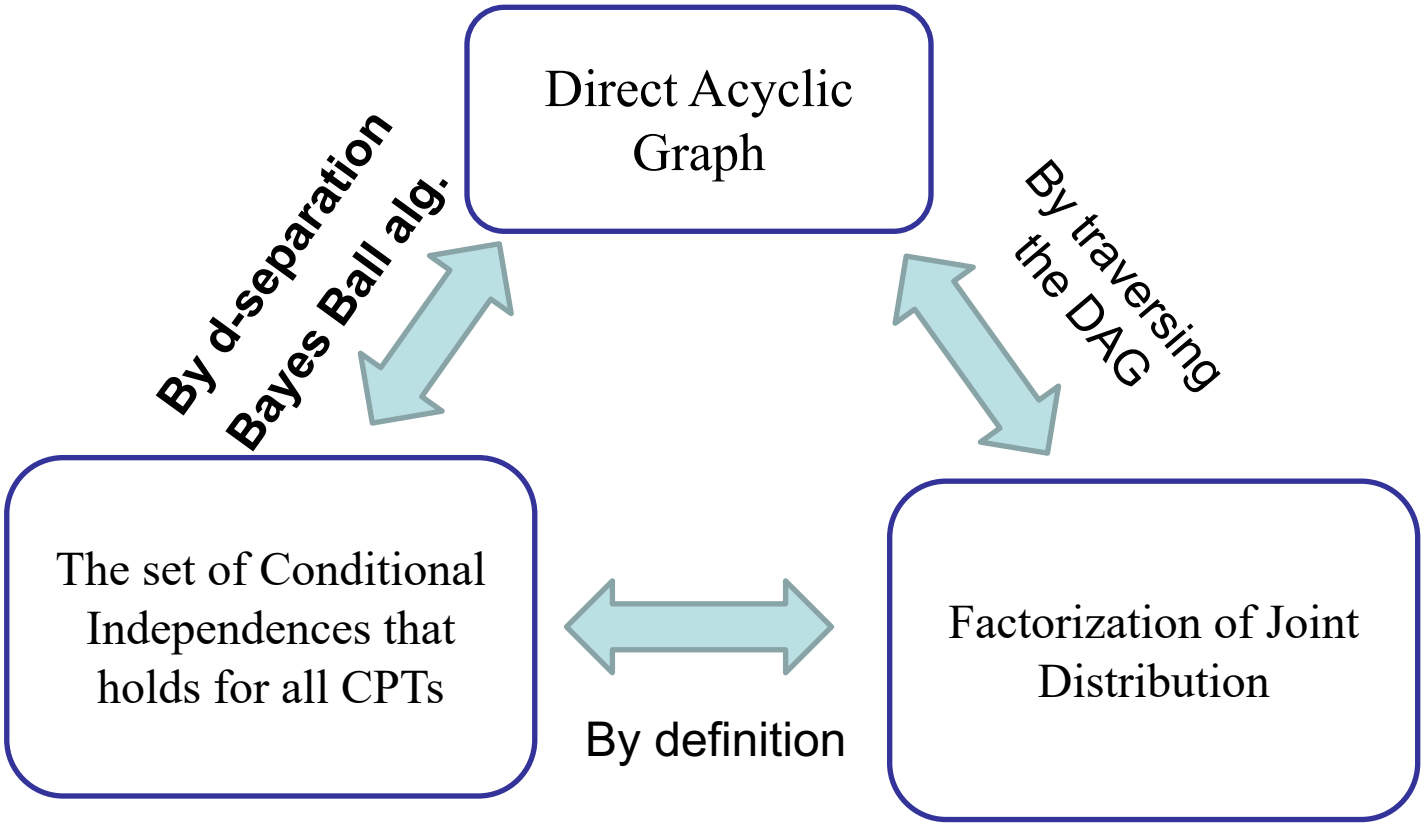



Question: How to fill values into these CPTs?

Ans: Specify by hands. Learn from data (e.g., MLE).

# Big question: Is there a general way that we can answer questions about conditional independences by just inspecting the graphs?

- Turns out the answer is "Yes!"



# What are the probabilistic graphical models for topics we learned in the second half?

- Expectimax
- MDP
- Bandits / Contextual Bandits
- Reinforcement Learning

# Lecture 7-10: Search

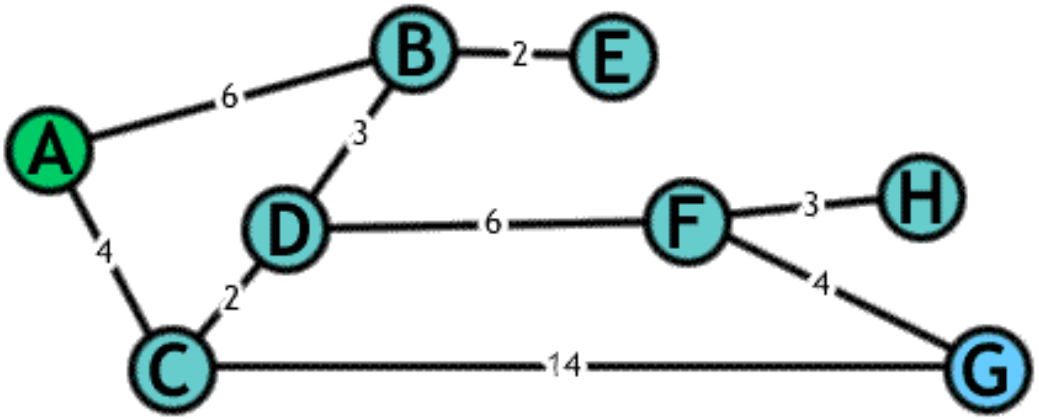
- Problem solving by search
  - Abstraction, problem formulation
  - State-space diagram
  - Count the number of states, number of actions.
- Uniformed Search algorithms
  - Four evaluation criteria
- Informed (heuristic) search
  - Admissible / consistent heuristics
  - Tree-search vs graph search
- **Minimax Search and Game playing**
  - Know how to do minimax / expectimax by hand!
  - Pruning

# Problem Formulation and Search

- Problem formulation
  - State-space description **< {S}, $S_0$, {$S_G$}, {O}, {g} >**
    - **S**: Possible states
    - **$S_0$**: Initial state of the agent
    - **$S_G$**: Goal state(s)
      - Or equivalently, a goal test **G(S)**
    - **O**: Operators  O: {S} => {S}
      - Describes the possible actions of the agent
    - **g**: Path cost function, assigns a cost to a path/action
- At any given time, which possible action $O_i$ is best?
  - Depends on the goal, the path cost function, the future sequence of actions….
- Agent's strategy:  Formulate, Search, and Execute
  - This is *offline* problem solving

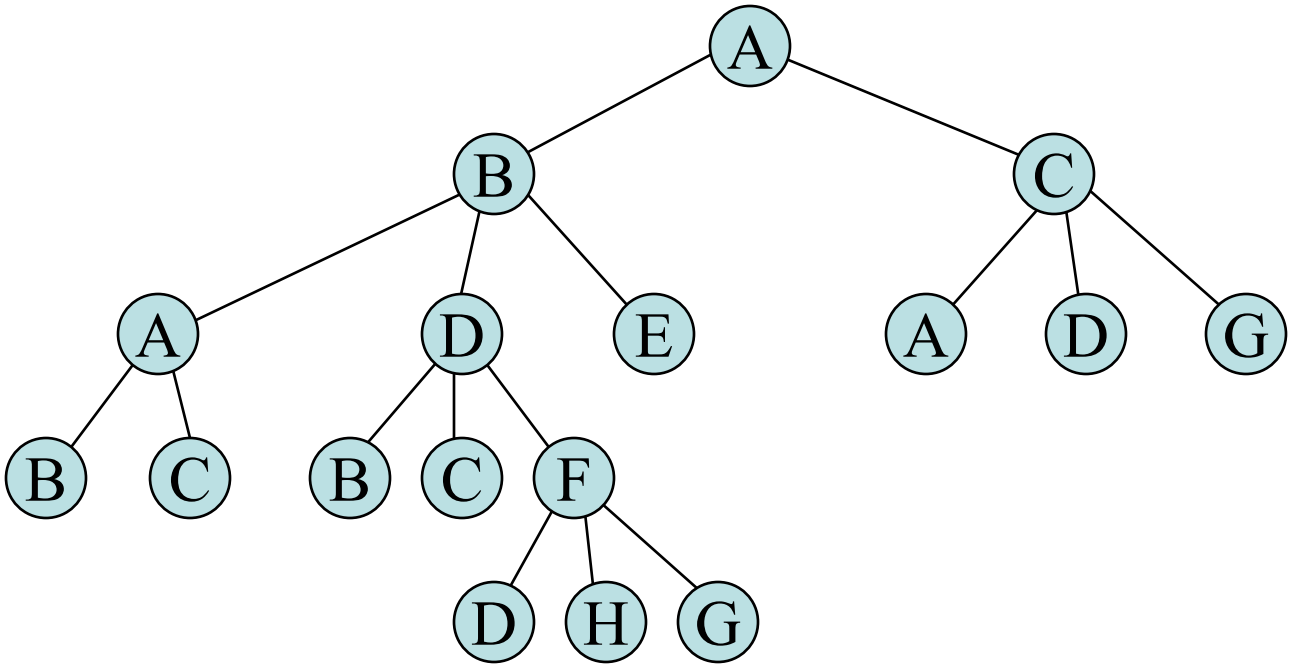
# State-Space Diagrams

- State-space description can be represented by a state-space diagram, which shows
  - States (incl. initial and goal)
  - Operators/actions (state transitions)
  - Path costs



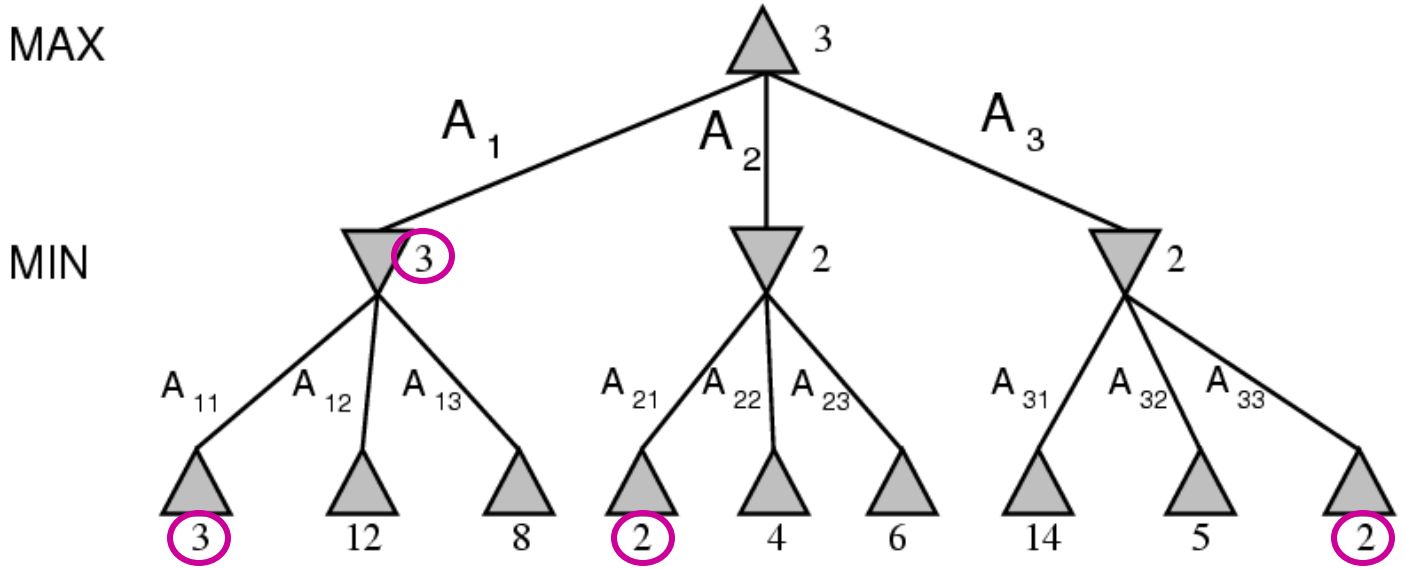
# State Space vs. Search Tree (cont.)

Search tree (partially expanded)



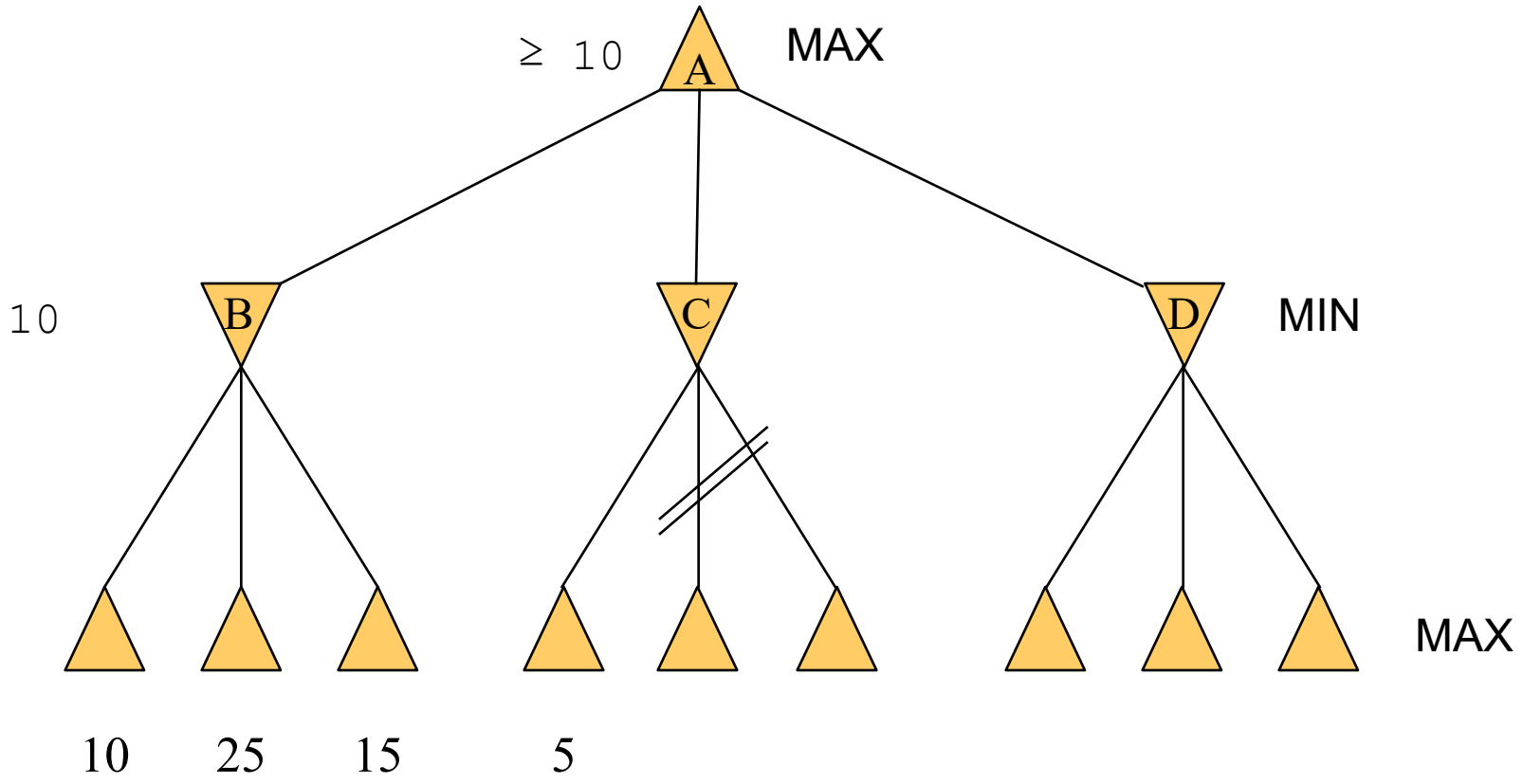
# Minimax example

Which move to choose?

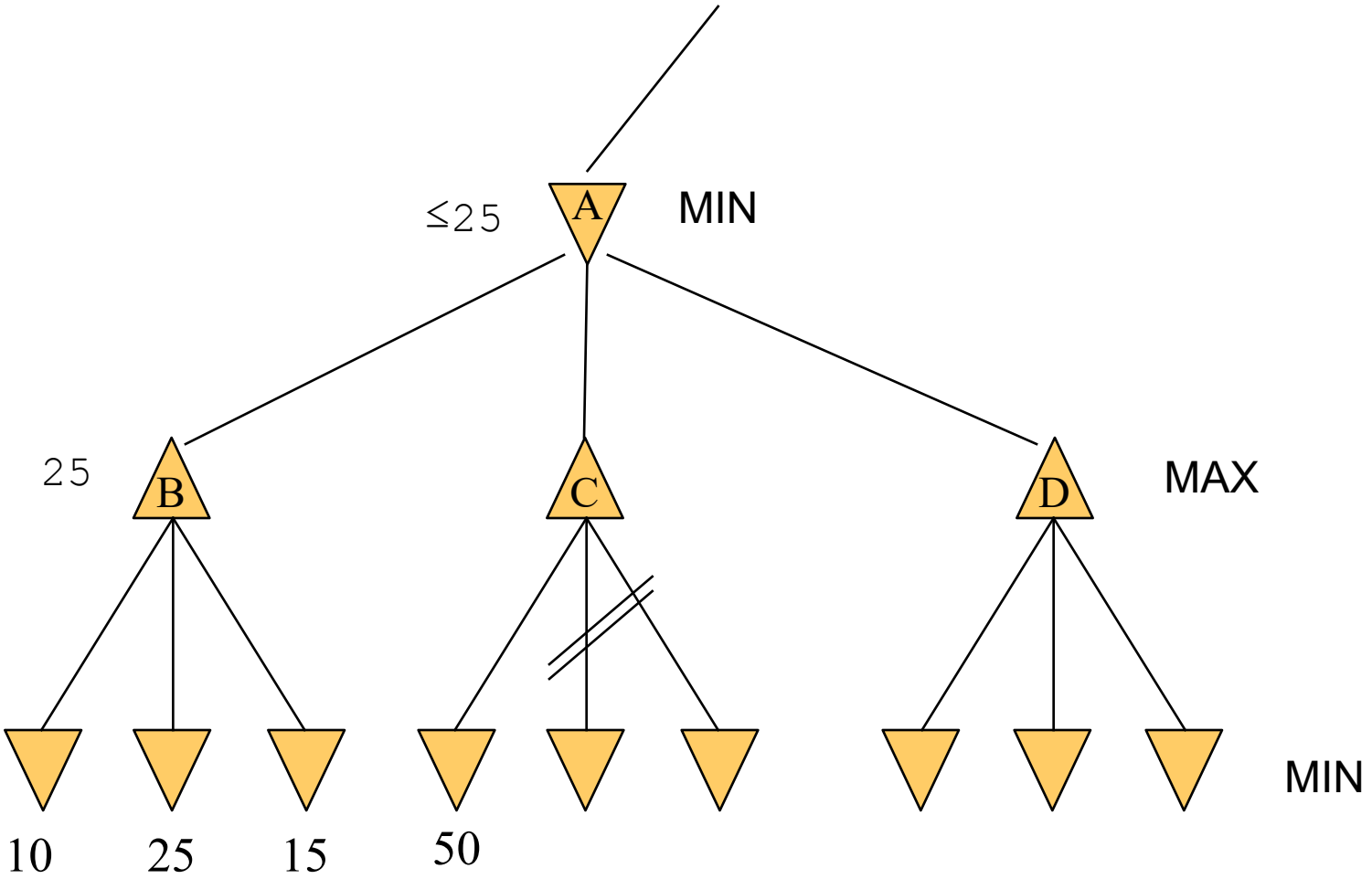



The **minimax decision** is move $\mathbf{A_1}$

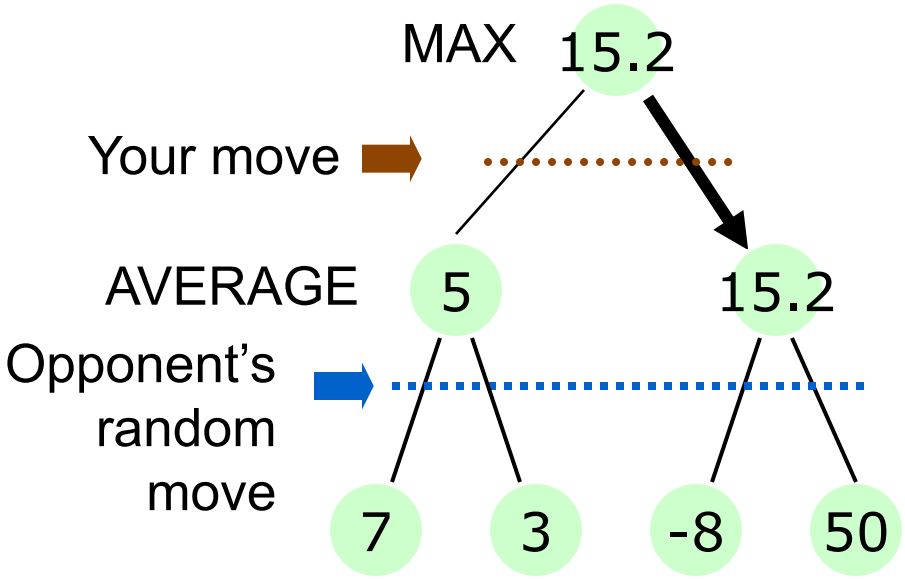
# Alpha pruning



# Beta pruning



# Expectimax




- Your opponent behave randomly with a given probability distribution,
- If you move left, your opponent will select actions with probability [0.5,0.5]
- If you move right, your opponent will select actions with [0.6,0.4]

# Lecture 11-16 Reinforcement Learning

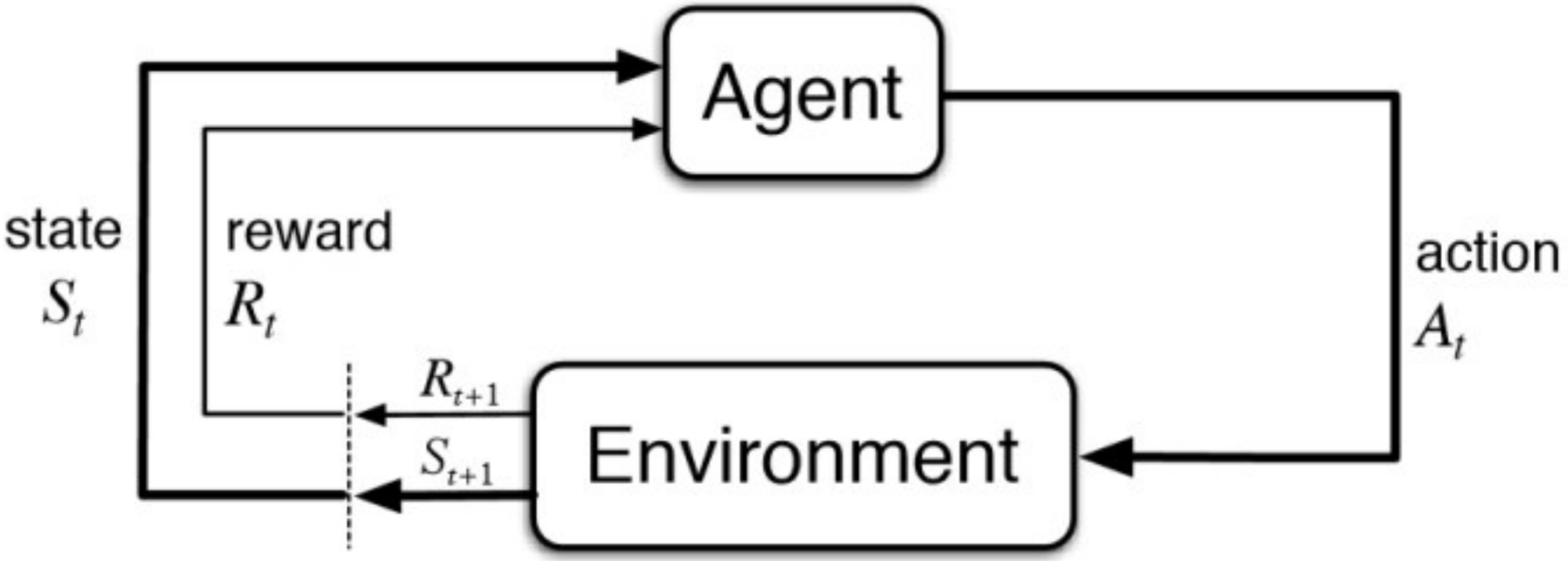
- Markov Decision Processes
  - New concepts: reward, value function, policy, transition dynamics
  - Bellman equations
  - Iterative algorithms for finding the optimal policy

- Bandits / Contextual bandits
  - The notion of regret
  - Explore-exploit

- Reinforcement Learning
  - Model-based learning
  - Model-free learning
  - Bootstrapping with Temporal Difference Learning

# Reinforcement learning problem setup

- State, Action, Reward
- Unknown reward function, unknown state-transitions.
- Agents might not even observe the state



# Reinforcement learning problem setup

- State, Action, Reward and Observation

$$S_t \in \mathcal{S} \quad A_t \in \mathcal{A} \quad R_t \in \mathbb{R} \quad O_t \in \mathcal{O}$$

- Policy:
  - When the state is observable: $\pi : \mathcal{S} \rightarrow \mathcal{A}$
  - Or when the state is not observable

$$\pi_t : (\mathcal{O} \times \mathcal{A} \times \mathbb{R})^{t-1} \rightarrow \mathcal{A}$$

- Learn the best policy that maximizes the expected reward
  - Finite horizon (episodic) RL: $\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}[\sum_{t=1}^{T} R_t]$ T: horizon
  - Infinite horizon RL: $\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}[\sum_{t=1}^{\infty} \gamma^{t-1} R_t]$ γ: discount factor

# Reinforcement learning is, arguably, the most general AI framework.

- RL: State, Action, Reward, Nothing is known.

- Simplified RL models:
  - iid state → Contextual bandits
  - No state, tabular action → Multi-arm bandits
  - iid state, no reward → Supervised Learning
  - Known dynamics / reward → Markov Decision Processes (Control/Cybernetics)
  - No reward / Unknown dynamics → System Identification

# Let us tackle different aspects of the RL problem one at a time

- Markov Decision Processes:
  - Dynamics are given no need to learn
- Bandits:  Explore-Exploit in simple settings
  - RL without dynamics
- Full Reinforcement Learning
  - Learning MDPs

# Tabular MDP

- Discrete State, Discrete Action, Reward and Observation

$$S_t \in \mathcal{S} \quad A_t \in \mathcal{A} \quad R_t \in \mathbb{R} \quad \sout{O_t \in \mathcal{O}}$$

- Policy:
  - When the state is observable: $\pi : \mathcal{S} \rightarrow \mathcal{A}$
  - ~~Or when the state is not observable~~

$$\sout{\pi_t : (\mathcal{O} \times \mathcal{A} \times \mathbb{R})^{t-1} \rightarrow \mathcal{A}}$$

- Learn the best policy that maximizes the expected reward
  - Finite horizon (episodic) MDP: $\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}[\sum_{t=1}^{T} R_t]$ T: horizon
  - Infinite horizon MDP: $\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}[\sum_{t=1}^{\infty} \gamma^{t-1} R_t]$

γ: discount factor

# Reward function and Value functions

- Immediate reward function r(s,a,s')
  - expected **immediate** reward

$$r(s,a,s') = \mathbb{E}[R_1 | S_1 = s, A_1 = a, S_2 = s']$$

$$r^{\pi}(s) = \mathbb{E}_{a \sim \pi(a|s)}[R_1 | S_1 = s]$$

- state value function: $V^{\pi}(s)$
  - expected **long-term** return when starting in *s* and following π

$$V^{\pi}(s) = \mathbb{E}_{\pi}[R_1 + \gamma R_2 + ... + \gamma^{t-1} R_t + ... | S_1 = s]$$

- state-action value function: $Q^{\pi}(s,a)$
  - expected **long-term** return when starting in *s*, performing *a,* and following π

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}[R_1 + \gamma R_2 + ... + \gamma^{t-1} R_t + ... | S_1 = s, A_1 = a]$$

# Bellman equations – the fundamental equations of MDP and RL

- An alternative, recursive and more useful way of defining the V-function and Q function
  - $V^{\pi}$ function Bellman equation

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

  - $Q^{\pi}$ function Bellman equation

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

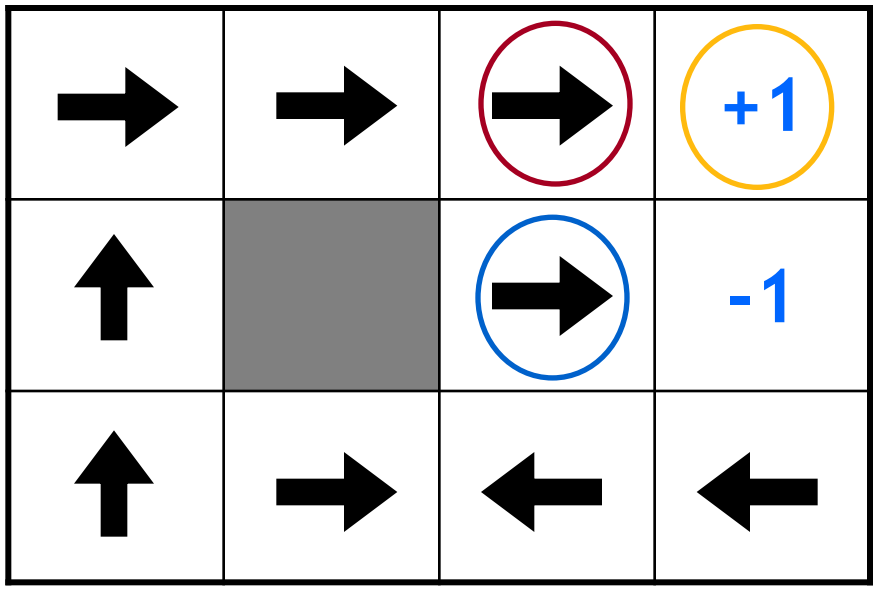
  - V* function Bellman (optimality) equation

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

  - Q* function Bellman (optimality) equation

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

# Let's work out the Value function for a specific policy




actions: UP, DOWN, LEFT, RIGHT

**UP**

80% move UP

10% move LEFT

10% move RIGHT

- reward +1 at [4,3], -1 at [4,2]
- reward -0.04 for each step

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

$$1.0 + 0.8 * (+1 - 0.04 + 0) + 0.1 * (-0.04 + V^{\pi}([3,2])) + 0.1 * (-0.04 + V^{\pi}([3,3]))$$

# Inference problem: given an MDP, how to compute its optimal policy?

- It suffices to compute its Q* function, because:

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

- It suffices to compute its V* function, because:

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

# How to calculate value functions given MDP parameters? Policy Iterations and Value Iterations

- What are these algorithms for?
  - Algorithms of computing the V* and Q* functions from MDP parameters

- Policy Iterations

$$\pi_0 \rightarrow^E V^{\pi_0} \rightarrow^I \pi_1 \rightarrow^E V^{\pi_1} \rightarrow^I \ldots \rightarrow^I \pi^* \rightarrow^E V^*$$

- Value iterations

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

- How do we make sense of them?
  - Recursively applying the Bellman equations until convergence.

# Multi-arm bandits: Problem setup

- No state. k-actions $a \in \mathcal{A} = \{1, 2, ..., k\}$

- You decide which arm to pull in every iteration

$$A_1, A_2, ..., A_T$$

- You collect a cumulative payoff of $\sum_{t=1}^{T} R_t$

- The goal of the agent is to maximize the expected payoff.
  - For future payoffs?
  - For the expected cumulative payoff?

# How do we measure the performance of an **online learning agent**?

- The notion of "Regret":
  - I wish I have done things differently.
  - Comparing to the best actions in the hindsight, how much worse did I do.

- For MAB, the regret is defined as follow

$$T \max_{a \in [k]} \mathbb{E}[R_t|a] - \sum_{t=1}^{T} \mathbb{E}_{a \sim \pi} \left[ \mathbb{E}[R_t|a] \right]$$

# Regret of different MAB algorithms

- Greedy $O(T)$
- ExploreFirst $O(T^{2/3}k^{1/3})$
- eps-Greedy $O(T^{2/3}k^{1/3})$
- Upper Confidence Bound: $O(T^{1/2}k^{1/2})$

# Off-policy evaluation under the Contextual bandits model

- Contexts:
  - $x_1, \ldots, x_n \sim \lambda$ drawn iid, possibly infinite domain
- Actions:
  - $a_i \sim \mu(a|x_i)$ Taken by a randomized "Logging" policy
- Reward:
  - $r_i \sim D(r|x_i, a_i)$ Revealed only for the action taken
- Value:
  - $v^{\mu} = \mathbb{E}_{x\sim\lambda}\mathbb{E}_{a\sim\mu(\cdot|x)}\mathbb{E}_D[r|x,a]$
- We collect data $(x_i, a_i, r_i)_{i=1}^n$ by the above processes.

# Off-policy Evaluation and Learning

**Off-policy evaluation**

Estimate the value of a fixed target policy $\pi$

$$v_\pi := \mathbb{E}_\pi[\text{Reward}]$$

**Off-policy learning**

Find $\pi \in \Pi$

that maximizes $v_\pi$

- Using data $(x_i, a_i, r_i)_{i=1}^n$
- often the policy $\mu$ or logged propensities $(\mu_i)_{i=1}^n$

# Algorithms for off-policy evaluation in contextual bandits

- Direct method

$$\hat{v}^{\pi}_{\mathrm{DM}} = \frac{1}{n} \sum_{i=1}^{n} \sum_{a \in \mathcal{A}} \hat{r}(x_i, a) \pi(a|x_i)$$

- Importance Sampling (Inverse Propensity Score)

$$\hat{v}^{\pi}_{\mathrm{IPS}} = \frac{1}{n} \sum_{i=1}^{n} \frac{\pi(a_i|x_i)}{\mu(a_i|x_i)} r_i$$

Question: Are these considered model-based or model-free RL?

# Revisit the dynamic programming approach

- Policy Evaluation

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

- Policy improvement

$$\pi'(s) = \arg\max_{a} Q^{\pi}(s,a)$$
$$= \arg\max_{a} \sum_{s'} P(s'|s,a)[r(s,a,s') + \gamma V_k^{\pi}(s')]$$

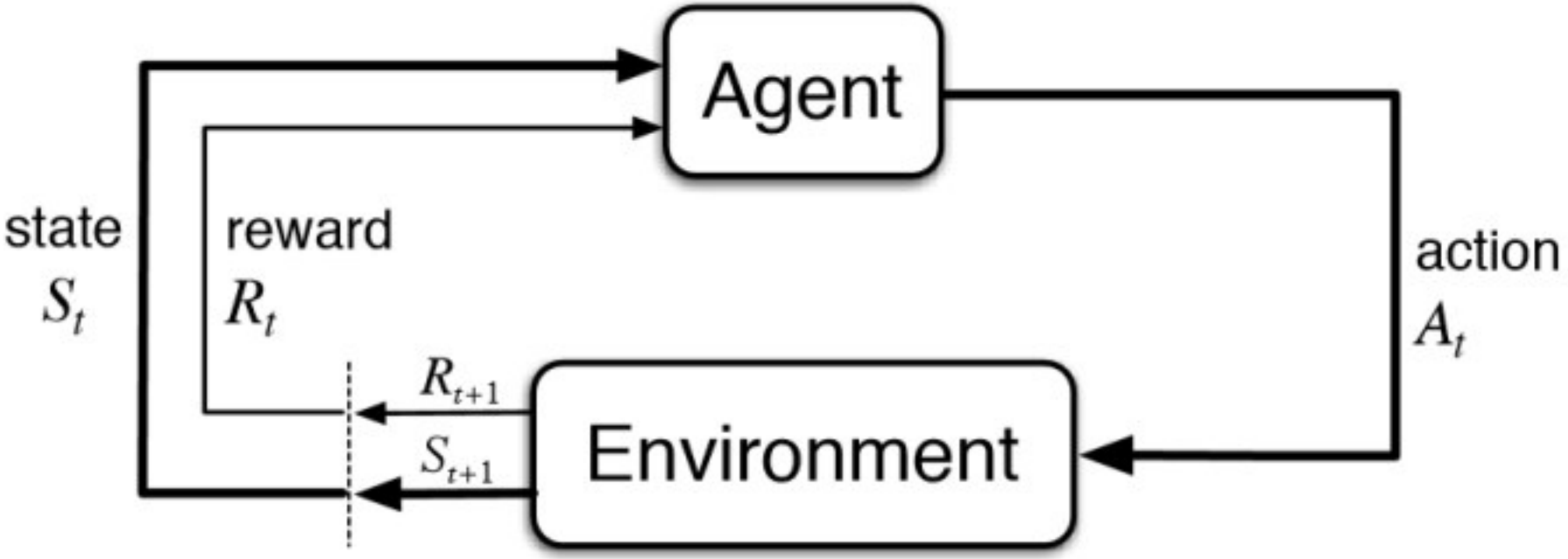
- Value iterations

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

***We do not have the MDP parameters in RL!**

# Reinforcement learning agents have "online" access to an environment

- State, Action, Reward
- Unknown reward function, unknown state-transitions.
- Agents can "act" and "experiment", rather than only doing offline planning.



# TD policy optimization (TD-control )

- SARSA (On-Policy TD-control)
  - Update the Q function by bootstrapping Bellman Equation

$$Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma Q(S',A') - Q(S,A)\right]$$

  - Choose the next A' using Q, e.g., eps-greedy.

- Q-Learning (Off-policy TD-control)
  - Update the Q function by bootstrapping Bellman Optimality Eq.

$$Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma \max_a Q(S',a) - Q(S,A)\right]$$

  - Choose the next A' using Q, e.g., eps-greedy, or any other policy.

# Q-Learning with function approximation

$$Q(s,a) = w_1 f_1(s,a) + w_2 f_2(s,a) + \ldots + w_n f_n(s,a)$$

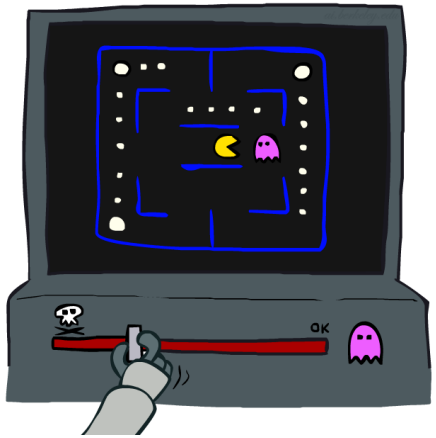
- Q-learning with linear Q-functions:

$$\text{transition} = (s, a, r, s')$$

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

$$Q(s,a) \leftarrow Q(s,a) + \alpha\,[\text{difference}] \qquad \text{Exact Q's}$$

$$w_i \leftarrow w_i + \alpha\,[\text{difference}]\, f_i(s,a) \qquad \text{Approximate Q's}$$



- Intuitive interpretation:
  - Adjust weights of active features
  - E.g., if something unexpectedly bad happens, blame the features that were on: disprefer all states with that state's features
- Formal justification: online least squares (Read the textbook!)

# Policy gradient

- Let's not worry about states, dynamics, Q function.
  - We might not even observe the true state.
  - Let's specify a class of parametrized policy and hope to compare to the best within this class

- Objective function to maximize: $J(\boldsymbol{\theta}) \doteq v_{\pi_{\boldsymbol{\theta}}}(s_0)$,

- Do SGD: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \widehat{\nabla J(\boldsymbol{\theta}_t)}$,

- Policy gradient theorem:

$$\nabla J(\boldsymbol{\theta}) = \sum_s d^\pi(s) \sum_a Q^\pi(s,a) \nabla_{\boldsymbol{\theta}} \pi(a|s,\boldsymbol{\theta})$$

*Note how this theorem is non-trivial… The first two terms depends on $\pi$, but we did not take the gradient w.r.t. them.

# Special case of policy gradient theorem in contextual bandits?

- Gradient of the IPS-estimator w.r.t. the parameter $\theta$ of the policy $\pi$

$$\hat{v}^{\pi}_{\mathrm{IPS}} = \frac{1}{n}\sum_{i=1}^{n}\frac{\pi(a_i|x_i)}{\mu(a_i|x_i)}r_i$$

- Connections to the policy gradient?

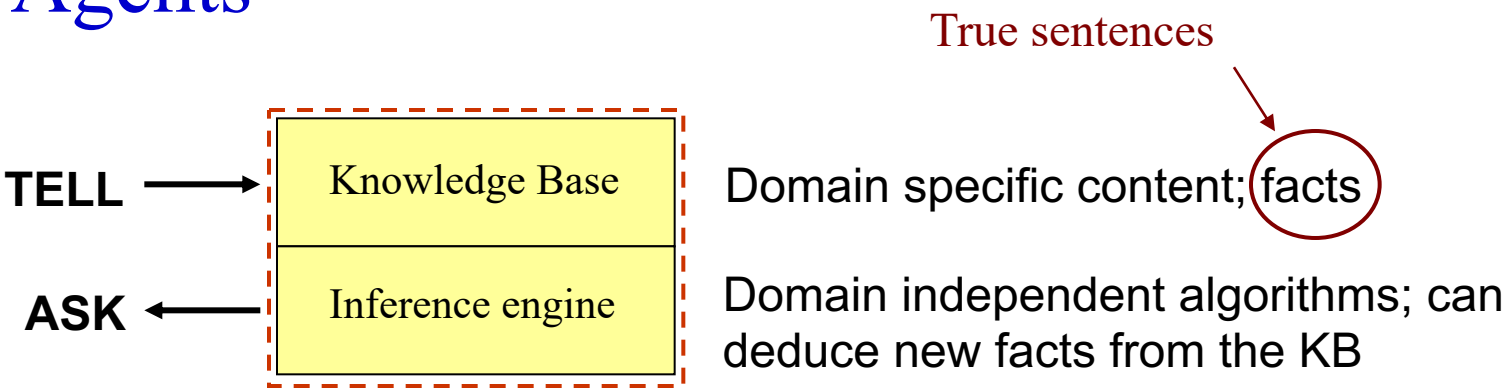
# Most important concepts in MDP / RL

- Make sure you understand
  - Problem setup, evaluation criteria
  - Definition of the policy, state, action, immediate reward, value, value function …
- Bellman equations
- Policy / Value iterations
  - Finding fixed points of Bellman equations
  - Finding eigenvector of a matrix
- SARSA, Q-Learning
  - SGD-style Stochastic simulation of Bellman equations

# Lecture 16-18: Logic

- Logic agent
- Knowledge Base
  - Tell operation
  - Ask operation
- Components of a formal mathematical logic system
  - Syntax, Semantics
- Inference Algorithms.

# KB Agents




- Need a formal logic system to work
- Need a data structure to represent known facts
- Need an algorithm to answer ASK questions

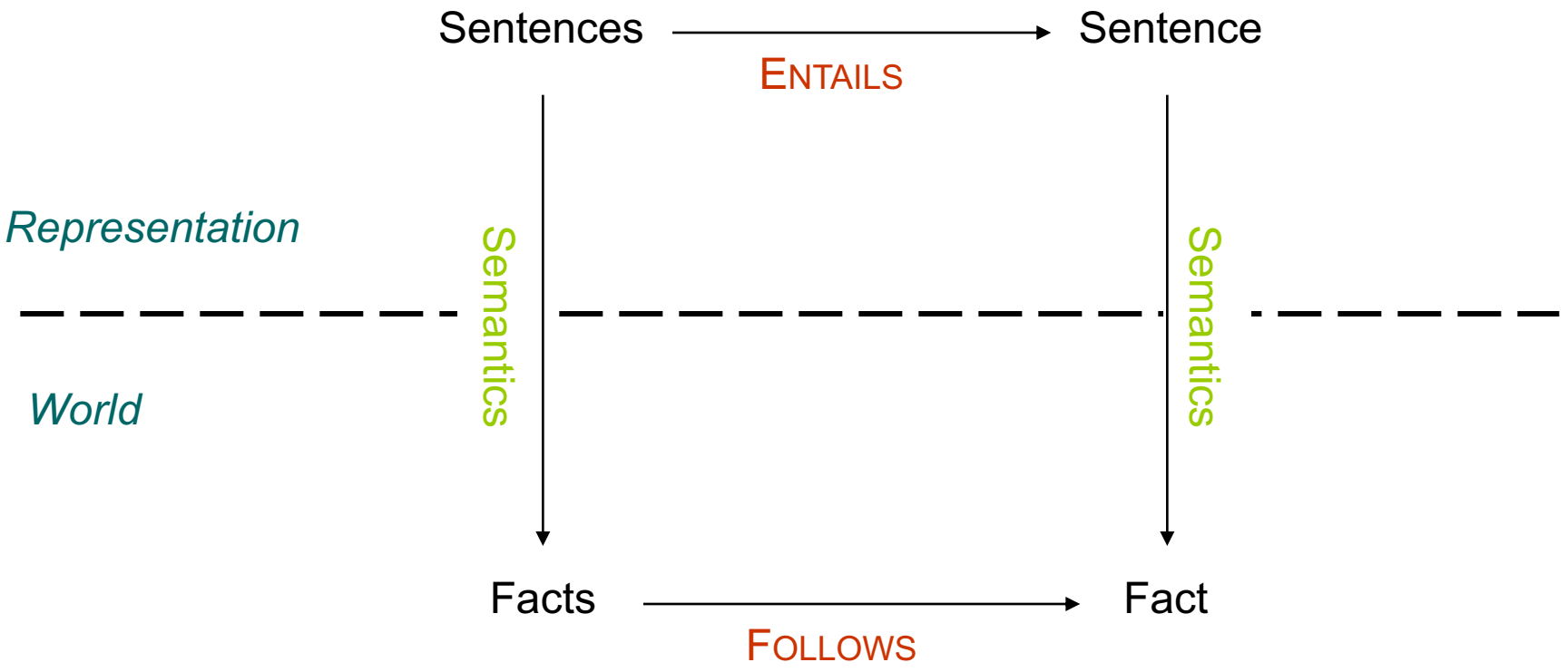
# Syntax and semantics

- Two components of a logic system

- Syntax --- How to construct sentences
  - The symbols
  - The operators that connect symbols together
  - A precedence ordering

- Semantics --- Rules the assignment of sentences to truth
  - For every possible worlds  (or “models” in logic jargon)
  - The truth table is a semantics

# Entailment




A is entailed by B, if A is true in all possible worlds consistent with B under the semantics.

# Inference procedure

- Inference procedure
  - Rules (algorithms) that we apply (often recursively) to derive truth from other truth.
  - Could be specific to a particular set of semantics, a particular realization of the world.

- Soundness and completeness of an inference procedure
  - Soundness: All truth discovered are valid.
  - Completeness: All truth that are entailed can be discovered.

# Propositional Logic

- Syntax:
  - *True, false,* propositional symbols
  - ( ) , ¬ (not), ∧ (and), ∨ (or), ⇒ (implies), ⇔ (equivalent)

- **Semantics**:
  - Five rules (the following truth table)

| $P$ | $Q$ | $\neg P$ | $P \wedge Q$ | $P \vee Q$ | $P \Rightarrow Q$ | $P \Leftrightarrow Q$ |
|---|---|---|---|---|---|---|
| *False* | *False* | *True* | *False* | *False* | *True* | *True* |
| *False* | *True* | *True* | *False* | *True* | *True* | *False* |
| *True* | *False* | *False* | *False* | *True* | *False* | *False* |
| *True* | *True* | *False* | *True* | *True* | *True* | *True* |

- **Inference rules**:
  - Modus Pronens etc.  Most important: Resolution

# Propositional logic agent

- Representation: Conjunctive Normal Forms
  - Represent them in a data structure:  a list, a heap, a tree?
  - Efficient TELL operation

- Inference: Solve ASK question
  - Use “Resolution” only on CNFs is Sound and Complete.
  - Equivalent to SAT, NP-complete, but good heuristics / practical algorithms exist

- Possible answers to ASK:
  - Valid, Satisfiable, Unsatisfiable

# First order logic

- More expressive language
  - Relations and functions of objects.
  - Quantifiers such as, All,  Exists.
- Easier to construct a KB.
  - Need much smaller number of sentences to capture a domain.
- Follow the same structure:  Symbols, Semantics
- Dedicated inference algorithms
- (FOL inference is not covered in the Final)

# Future of AI

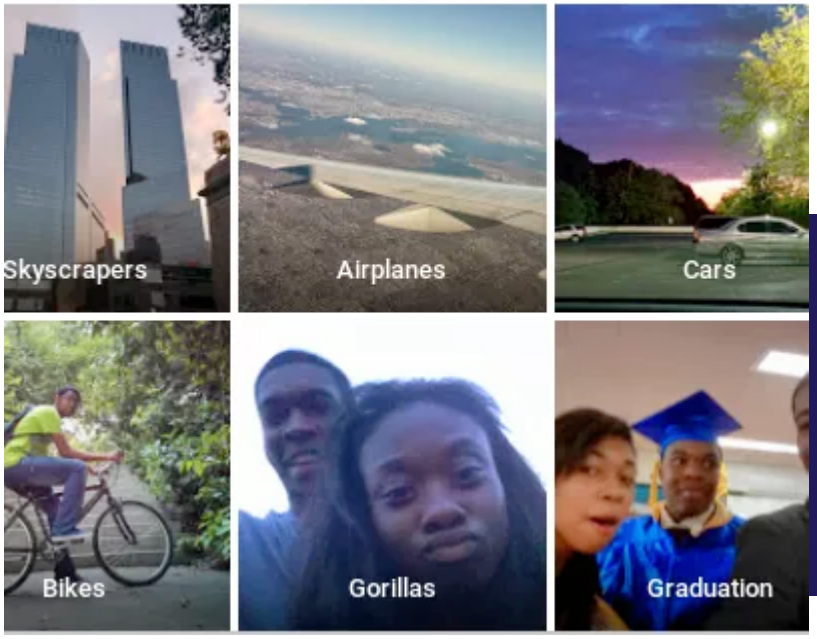
- More higher level intelligence
  - Logic is coming back
  - But more learning based than rule-based
- More stateful systems, more reinforcement learning
  - Causal modelling and reasoning
- More AI in the non-iid environment
  - Structured
  - Adversarial
- More forms of agent's perception
  - Weak supervision
  - Self-supervision (bootstrapping)

# The need for responsible AI: with great power comes great responsibility

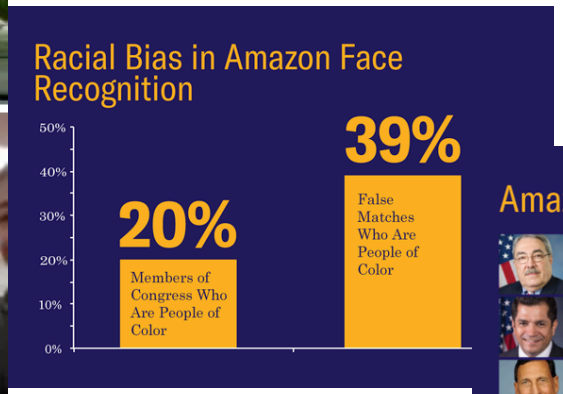
A face recognition system

- Technology is a double-bladed sword
- It matters who wields it and for what purpose

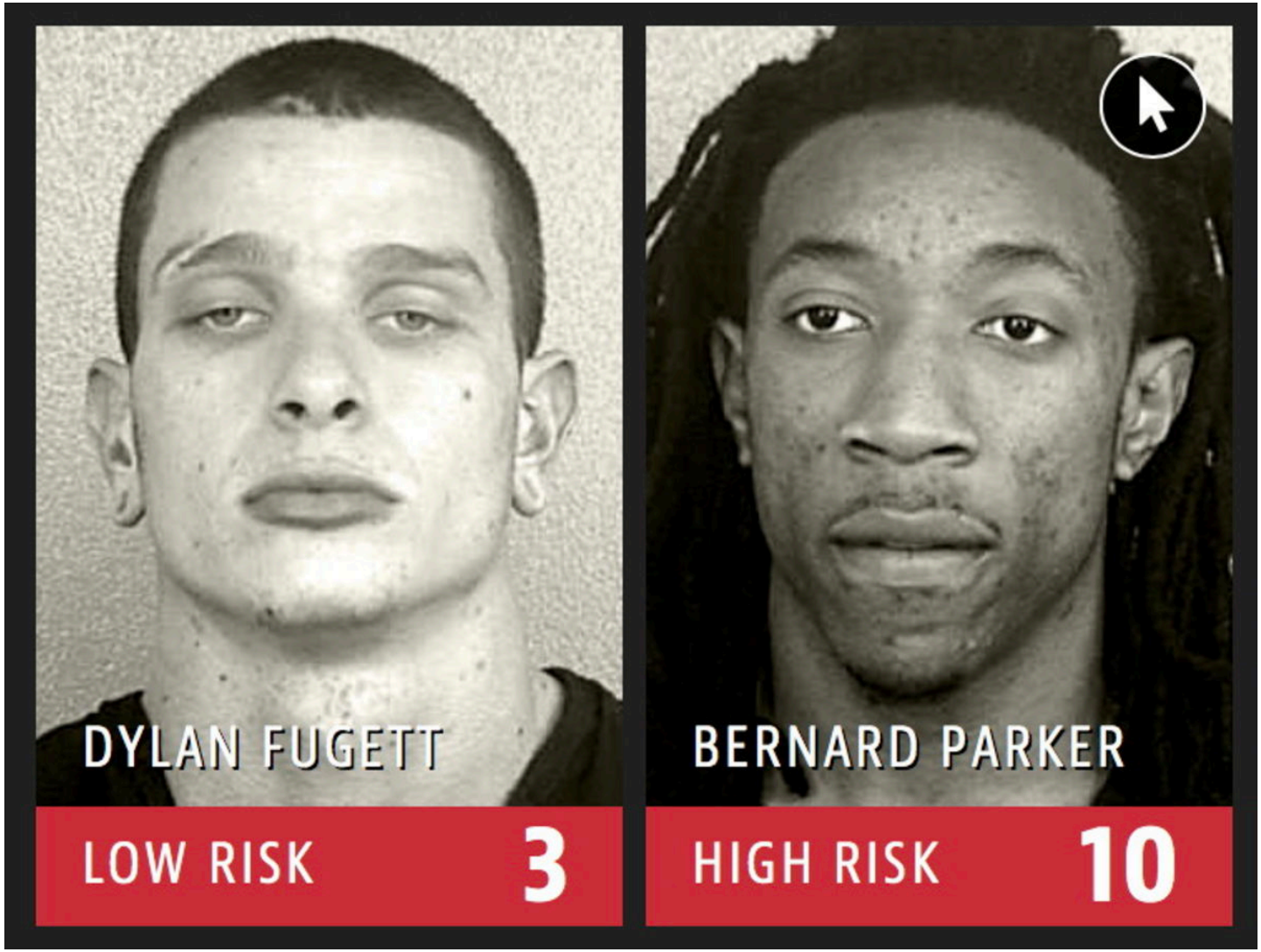
# Fairness challenges in AI systems / AI for decision making



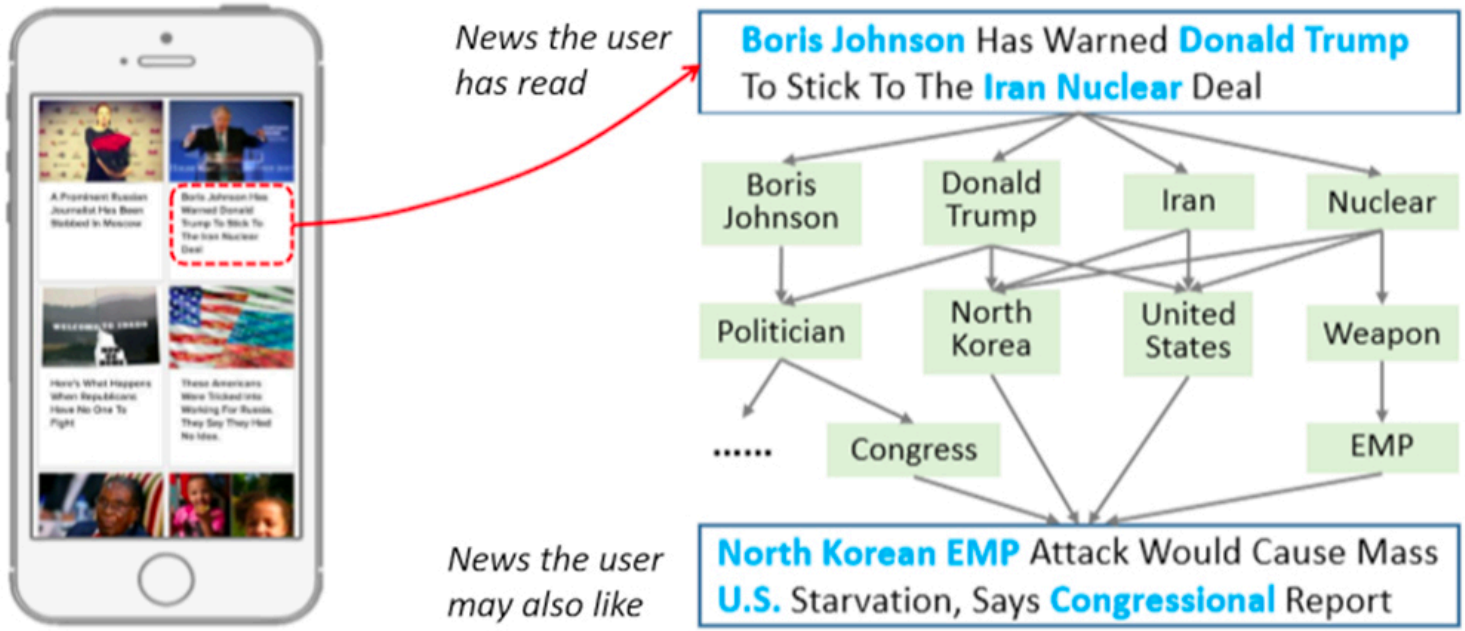

Google's image recognition system



# AI for predicting recidivism: "COMPAS" is used by courts… but is it biased?



# Polarizing effects of news recommendation




- Only what you like to read will be recommended to you.

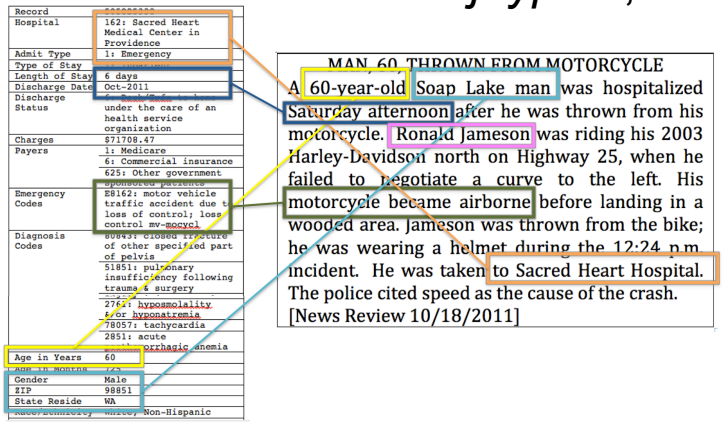
# Privacy issues in data collection and learning

"Robust De-anonymization of Large Datasets (How to Break Anonymity of the Netflix Prize Dataset)"

A. Narayanan & V. Shmatikov. *Security and Privacy, 2008*

- **Anonymization doesn't work!**
- **Need robust / provable approaches.**

Vijay Pandurangan. *tech.vijayp.ca*, 2014




"Only You, Your Doctor, and Many Others May Know"

L. Sweeney. *Technology Science, 2015*

# ML models memorize training datasets, even though they are generalizing well!

Membership Inference Attacks Against Machine Learning Models

Reza Shokri, Cornell Tech
Marco Stronati*, INRIA
Congzheng Song, Cornell
Vitaly Shmatikov, Cornell Tech

***Abstract*—We quantitatively investigate how machine learning models leak information about the individual data records on which they were trained. We focus on the basic membership inference attack: given a data record and black-box access to a model, determine if the record was in the model's training dataset. To perform membership inference against a target model, we make adversarial use of machine learning and train our own inference model to recognize differences in the target model's predictions on the inputs that it trained on versus the inputs that it did not train on.**

**We empirically evaluate our inference techniques on classification models trained by commercial "machine learning as a service" providers such as Google and Amazon. Using realistic datasets and classification tasks, including a hospital discharge dataset whose membership is sensitive from the privacy perspective, we show that these models can be vulnerable to membership inference attacks. We then investigate the factors that influence this leakage and evaluate mitigation strategies.**

*Security and Privacy, 2017*

**The Secret Sharer:**
**Measuring Unintended Neural Network Memorization & Extracting Secrets**

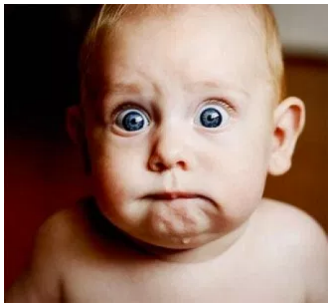
Nicholas Carlini, *University of California, Berkeley*
Chang Liu, *University of California, Berkeley*
Jernej Kos, *National University of Singapore*
Úlfar Erlingsson, *Google Brain*
Dawn Song, *University of California, Berkeley*

This paper presents *exposure*, a simple-to-compute metric that can be applied to any deep learning model for measuring the memorization of secrets. Using this metric, we show how to extract those secrets efficiently using black-box API access. Further, we show that unintended memorization occurs early, is not due to overfitting, and is a persistent issue across different types of models, hyperparameters, and training strategies. We experiment with both real-world models (e.g., a state-of-the-art translation model) and datasets (e.g., the Enron email dataset, which contains users' credit card numbers) to demonstrate both the utility of measuring exposure and the ability to extract secrets.

Finally, we consider many defenses, finding some ineffective (like regularization), and others to lack guarantees. However, by instantiating our own differentially-private recurrent model, we validate that by appropriately investing in the use of state-of-the-art techniques, the problem can be resolved, with high utility.

*USENIX Security 2019*

# Recent/upcoming legislations on privacy forces companies to revise their data practice

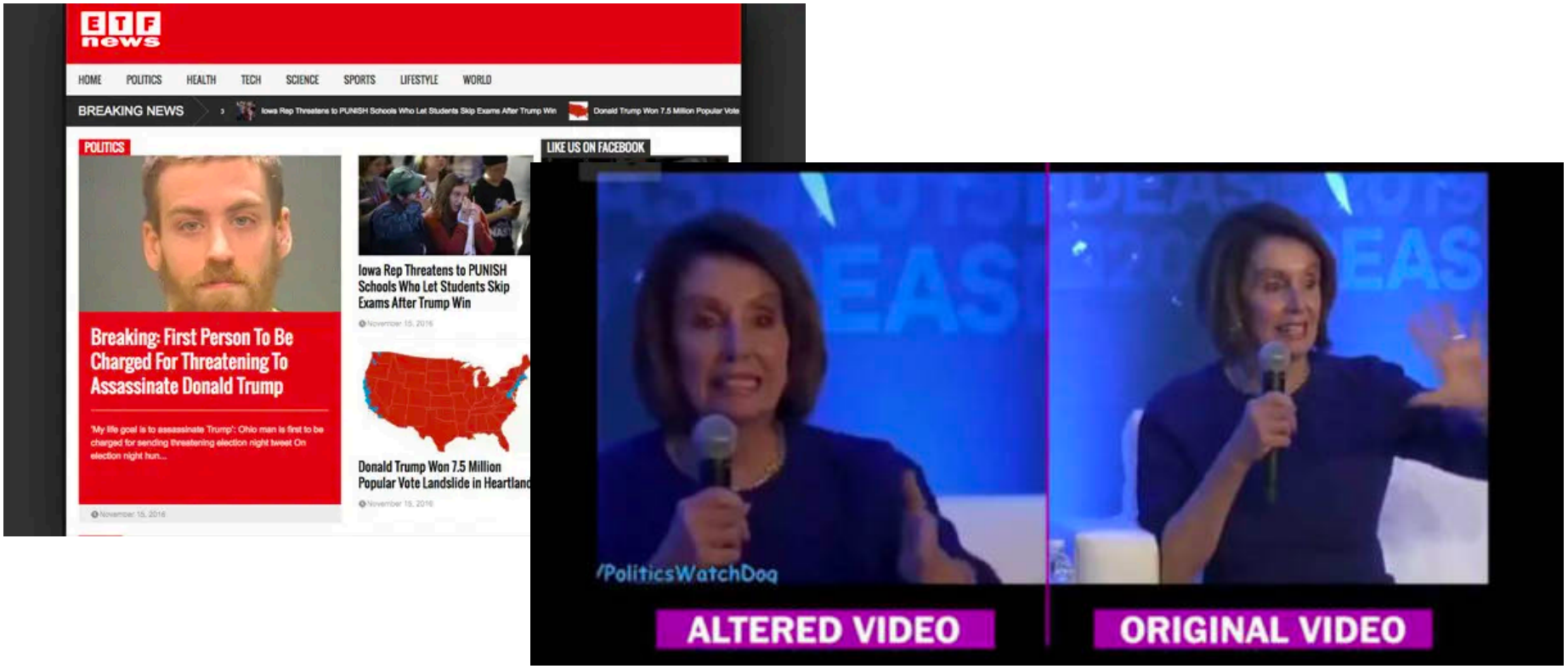


- I can't keep personal data for more than three weeks?

- I will have to delete all traces of a user upon request?

**How about my machine learning models trained on user data?**

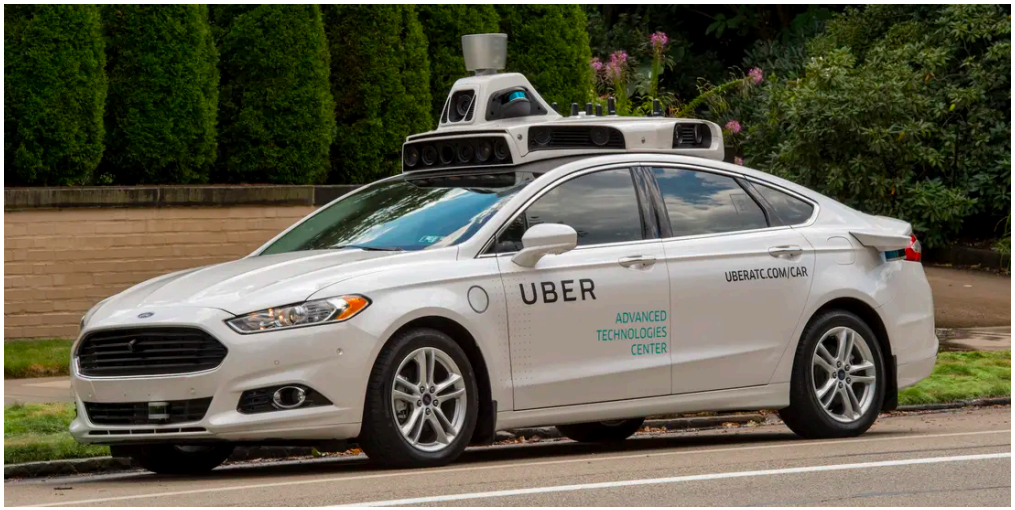
# Fake-news, fake voice, fake video

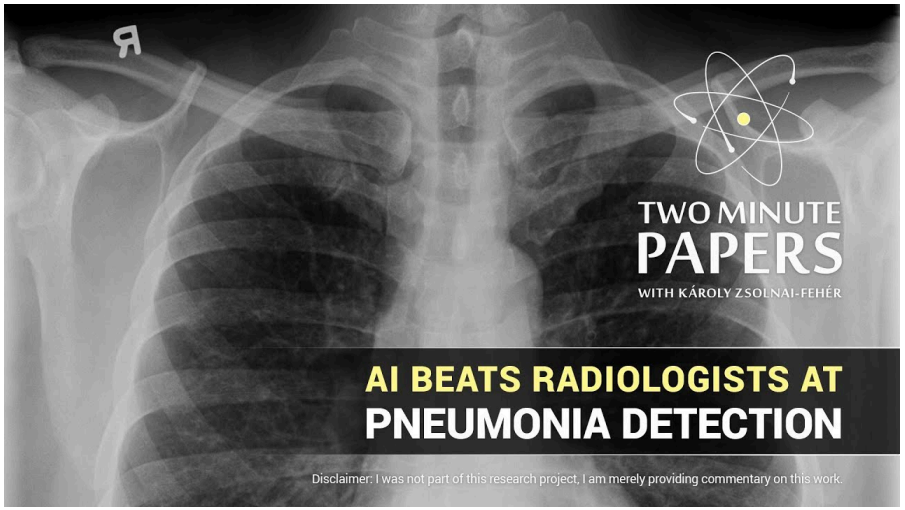



- How to tell if something is true or false?
- How to attribute a crime with factual evidence when people can just claim it's fake?

# Societal impacts of new technology

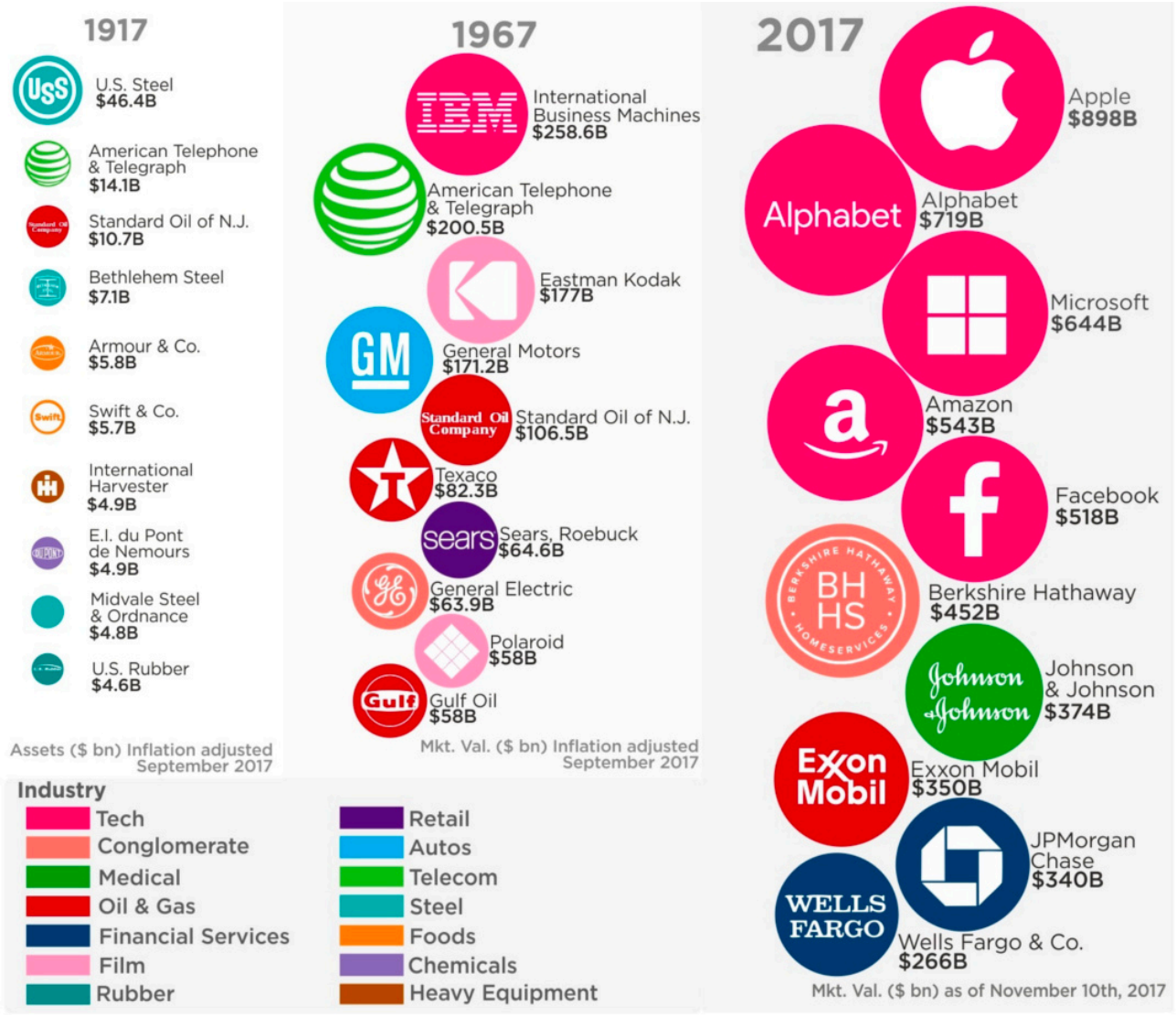
- Unemployment
  - Making people more productive. Less demand for labor.
- Specific tasks in jobs are being eliminated




- AI is also creating new jobs, but…
  - Can your grandpa learn how to code?

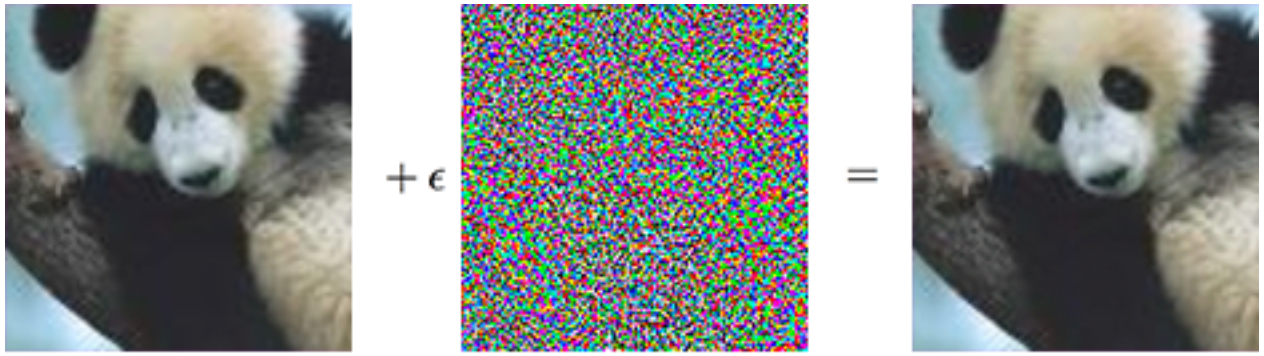
# Who are getting the largest piece of the technology pie?




**Source and Article:**
https://howmuch.net/articles/100-years-of-americas-top-10-companies
https://forbes.com

howmuch.net

# Safety issue in deploying AI




"panda"
57.7% confidence

"gibbon"
99.3% confidence

classified as
**Stop Sign**

misclassified as
**Max Speed 100**

# Final words

- With greater power comes great responsibility.
  - Ethics in AI,  Privacy, fairness, social impacts
  - Transparency, robustness, explanability
  - AI for good causes

- These are very complex issues
  - Are humans good decision makers? Are there implicit biases?
  - Can we explain our decisions
  - Should we regulate? How? To what extent?

- The future is in your hands. Be a good driver!

# Thank you

- It's my pleasure to work with you!
- I hope the course is / will be useful.

- AI Research at UCSB
  - Machine Learning Lab
  - Natural Language Processing Lab
  - Center for Responsible Machine Learning
  - Center for Information Technology and Society
  - The Mellichamp Initiative in Mind & Machine Intelligence
  - Data Science Initiative