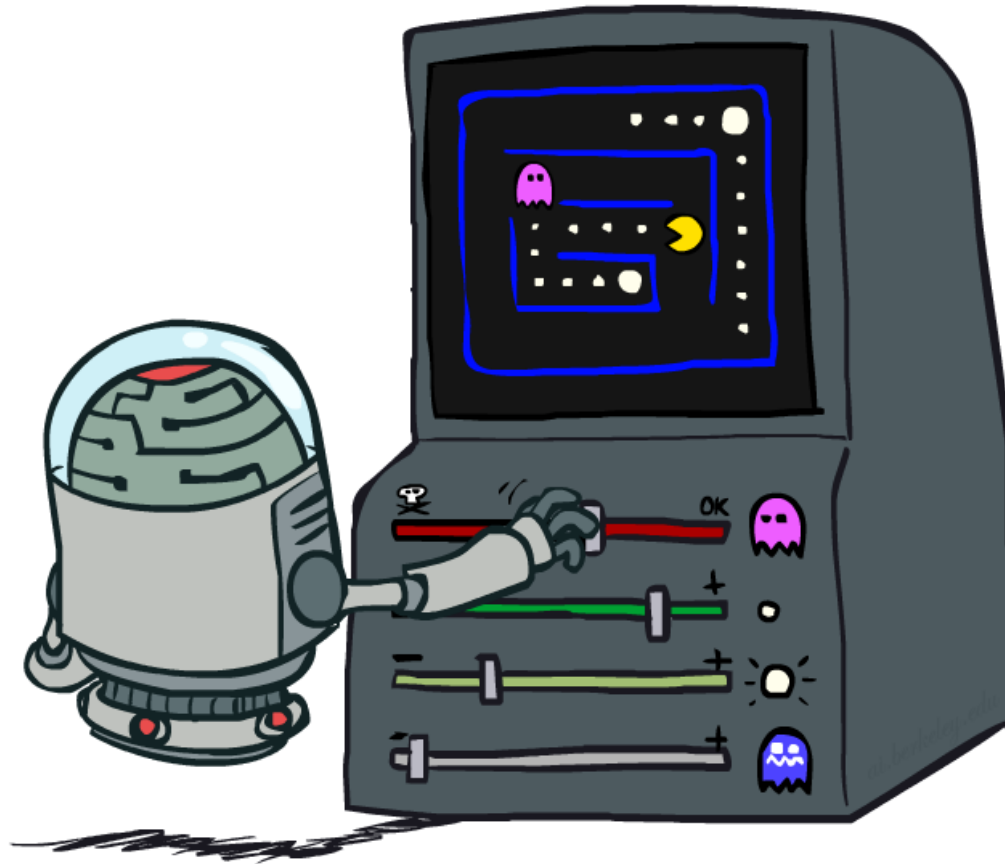# Reinforcement Learning Wrap-up
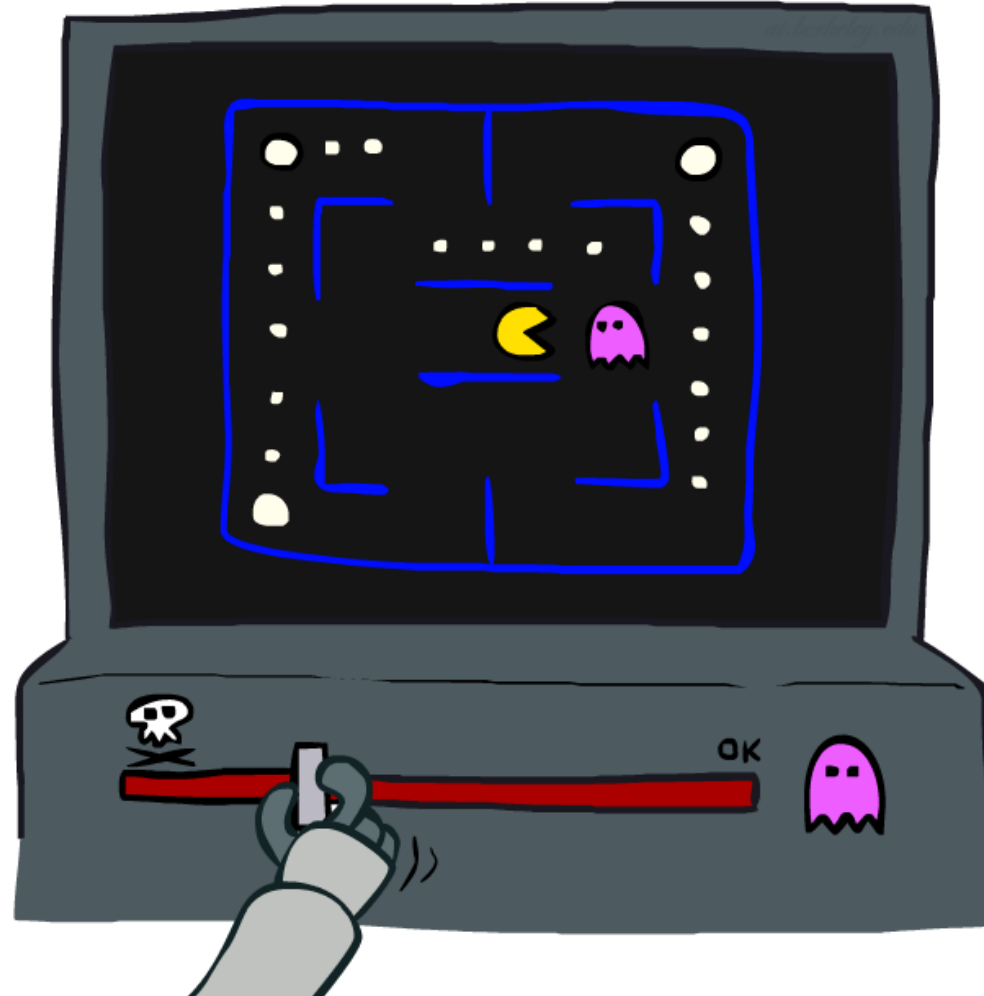

Slides courtesy of Dan Klein and Pieter Abbeel – University of California, Berkeley

Approximate Q-Learning

Generalizing Across States

- Basic Q-Learning keeps a table of all q-values

- In realistic situations, we cannot possibly learn about every single state!
  - Too many states to visit them all in training
  - Too many states to hold the q-tables in memory

- Instead, we want to generalize:
  - Learn about some small number of training states from experience
  - Generalize that experience to new, similar situations
  - This is a fundamental idea in machine learning, and we'll see it over and over again

[demo – RL pacman]
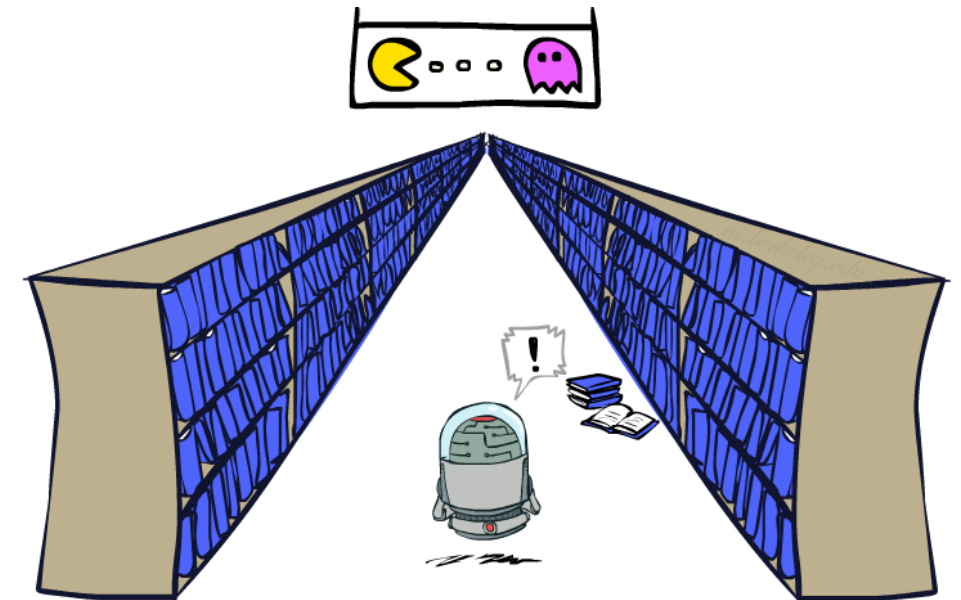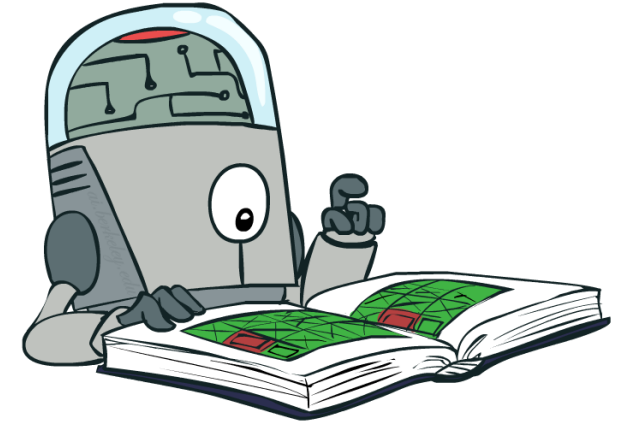
Example: Pacman

Let's say we discover
through experience
that this state is bad:
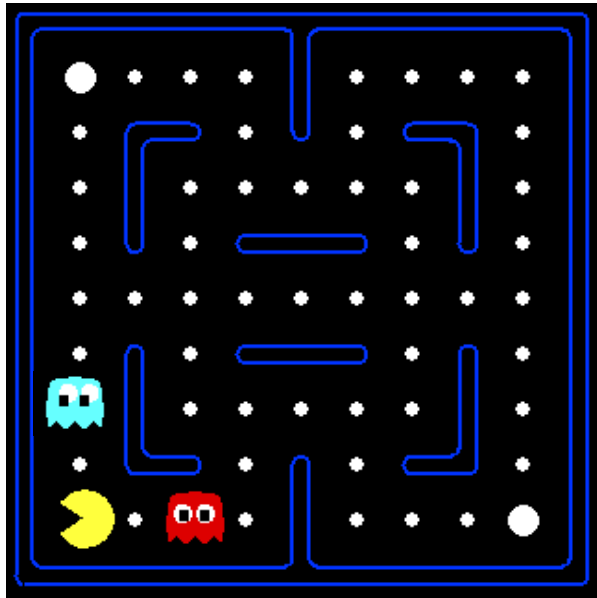
In naïve q-learning,
we know nothing
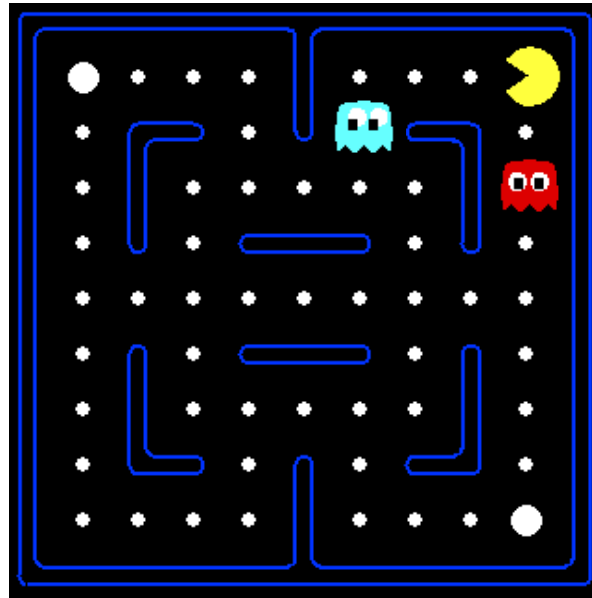about this state:

Or even this one!

Feature-Based Representations

- Solution: describe a state using a vector of features (properties)
  - Features are functions from states to real numbers (often 0/1) that capture important properties of the state
  - Example features:
    - Distance to closest ghost
    - Distance to closest dot
    - Number of ghosts
    - 1 / (dist to dot)^2
    - Is Pacman in a tunnel? (0/1)
    - …… etc.
    - Is it the exact state on this slide?
  - Can also describe a q-state (s, a) with features (e.g. action moves closer to food)

Linear Value Functions

- Using a feature representation, we can write a q function (or value function) for any state using a few weights:

$$V(s) = w_1 f_1(s) + w_2 f_2(s) + \ldots + w_n f_n(s)$$

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

- Advantage: our experience is summed up in a few powerful numbers

- Disadvantage: states may share features but actually be very different in value!

Approximate Q-Learning

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

- Q-learning with linear Q-functions:

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

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

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

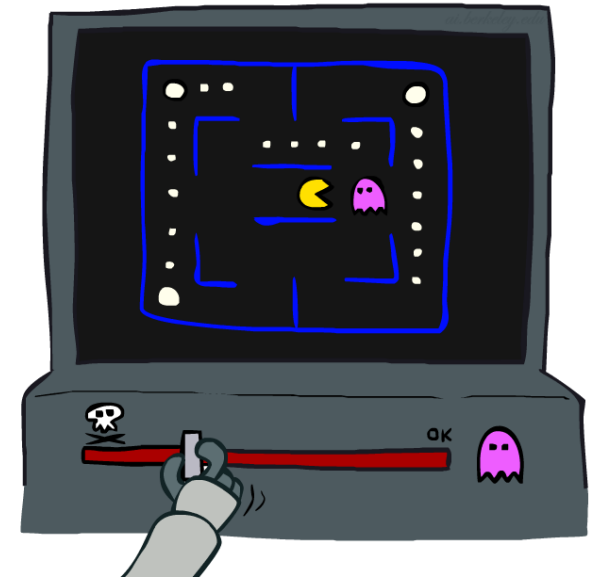$$w_i \leftarrow w_i + \alpha \, [\text{difference}] \, f_i(s,a) \qquad \text{Approximate Q's}$$

- 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

Example: Q-Pacman

$$Q(s, a) = 4.0 f_{DOT}(s, a) - 1.0 f_{GST}(s, a)$$

$f_{DOT}(s, \textsf{NORTH}) = 0.5$

$a = \textsf{NORTH}$
$r = -500$
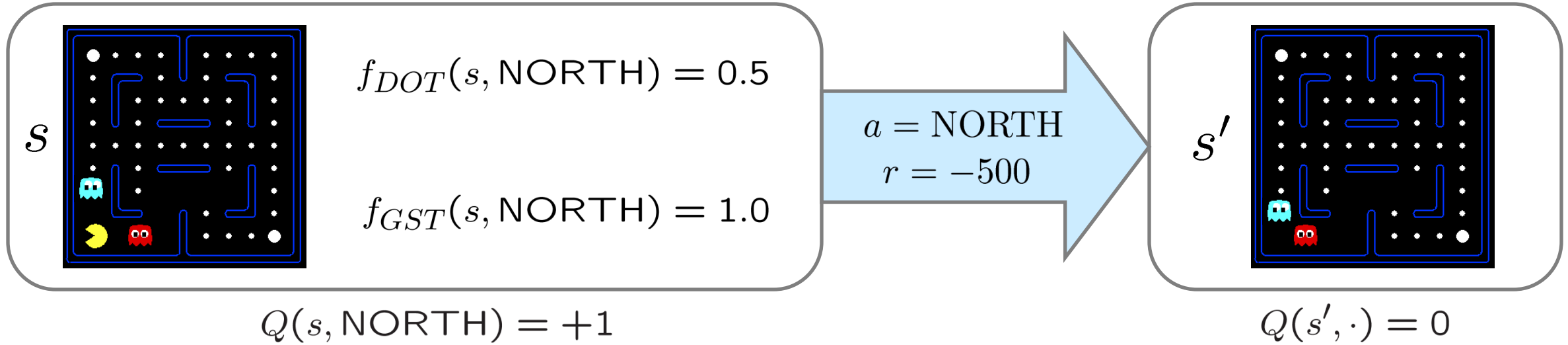
$f_{GST}(s, \textsf{NORTH}) = 1.0$

$Q(s, \textsf{NORTH}) = +1$

$r + \gamma \max_{a'} Q(s', a') = -500 + 0$

$Q(s', \cdot) = 0$

$\text{difference} = -501$

$w_{DOT} \leftarrow 4.0 + \alpha \, [-501] \, 0.5$
$w_{GST} \leftarrow -1.0 + \alpha \, [-501] \, 1.0$

$$Q(s, a) = 3.0 f_{DOT}(s, a) - 3.0 f_{GST}(s, a)$$

[Demo: approximate Q-learning pacman (L11D10)]

Video of Demo Approximate Q-Learning -- Pacman
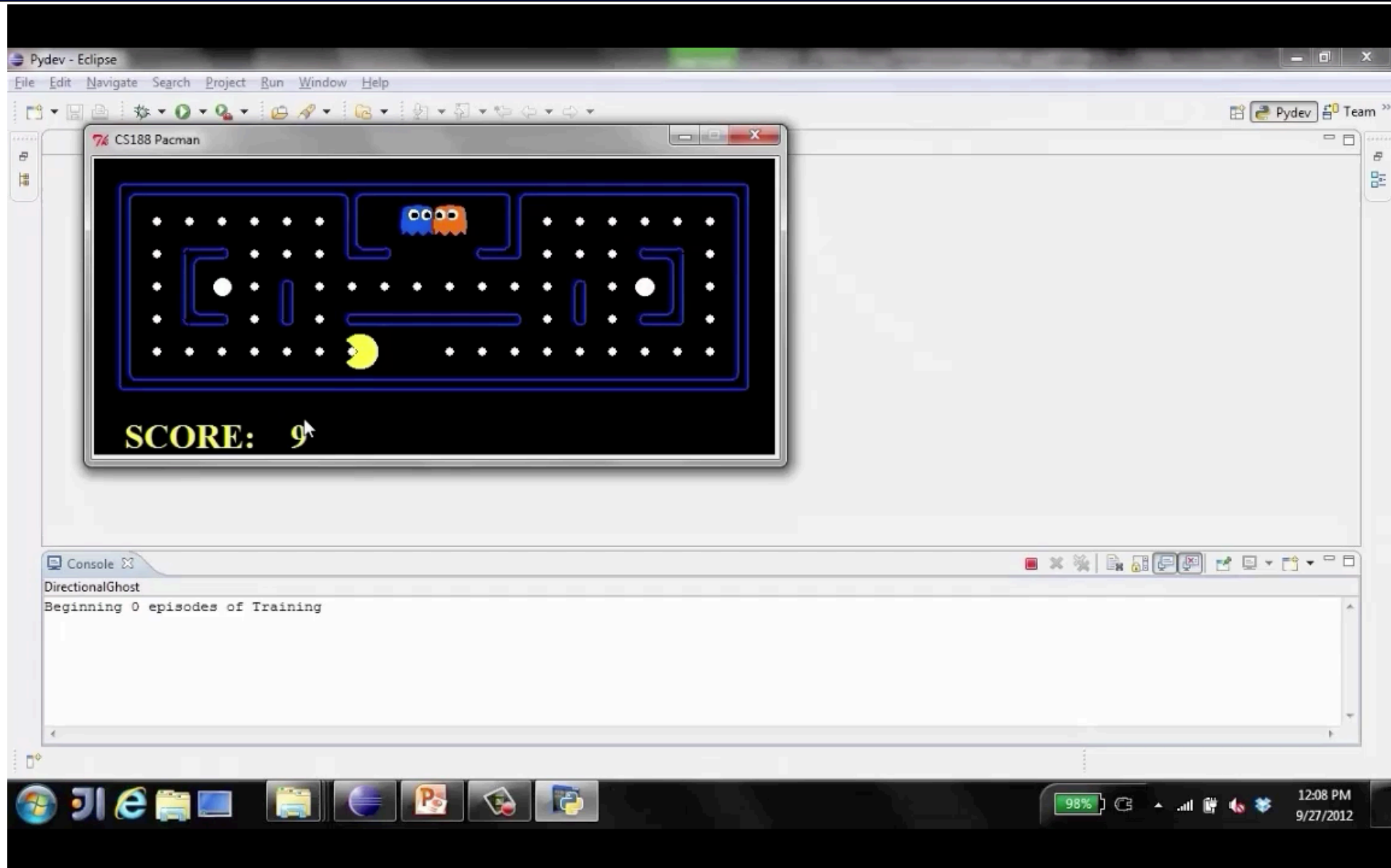
Linear Approximation: Regression*
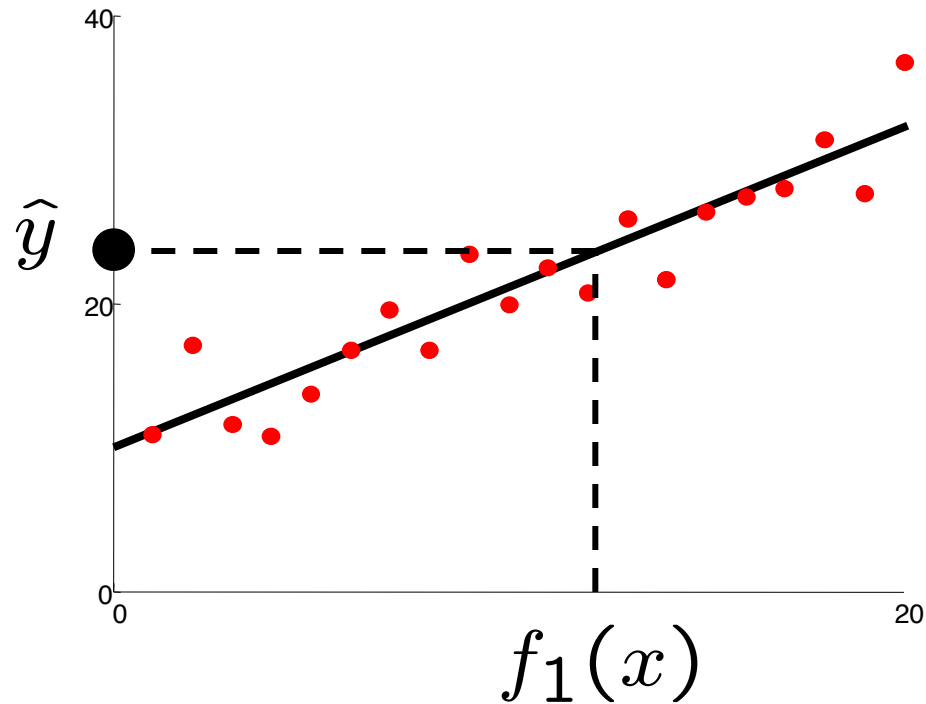
Prediction:
$$\hat{y} = w_0 + w_1 f_1(x)$$
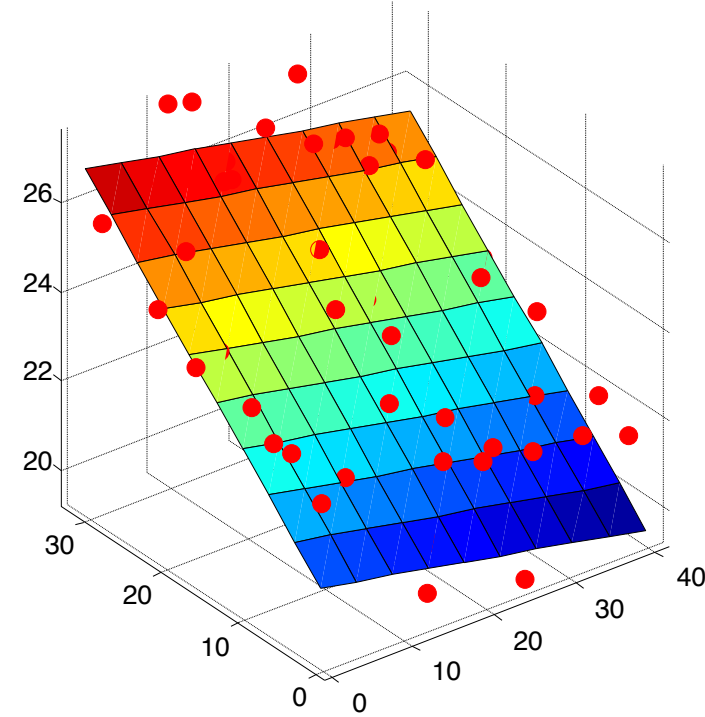
Prediction:
$$\hat{y}_i = w_0 + w_1 f_1(x) + w_2 f_2(x)$$

Optimization: Least Squares*

$$\text{total error} = \sum_i (y_i - \widehat{y}_i)^2 = \sum_i \left(y_i - \sum_k w_k f_k(x_i)\right)^2$$
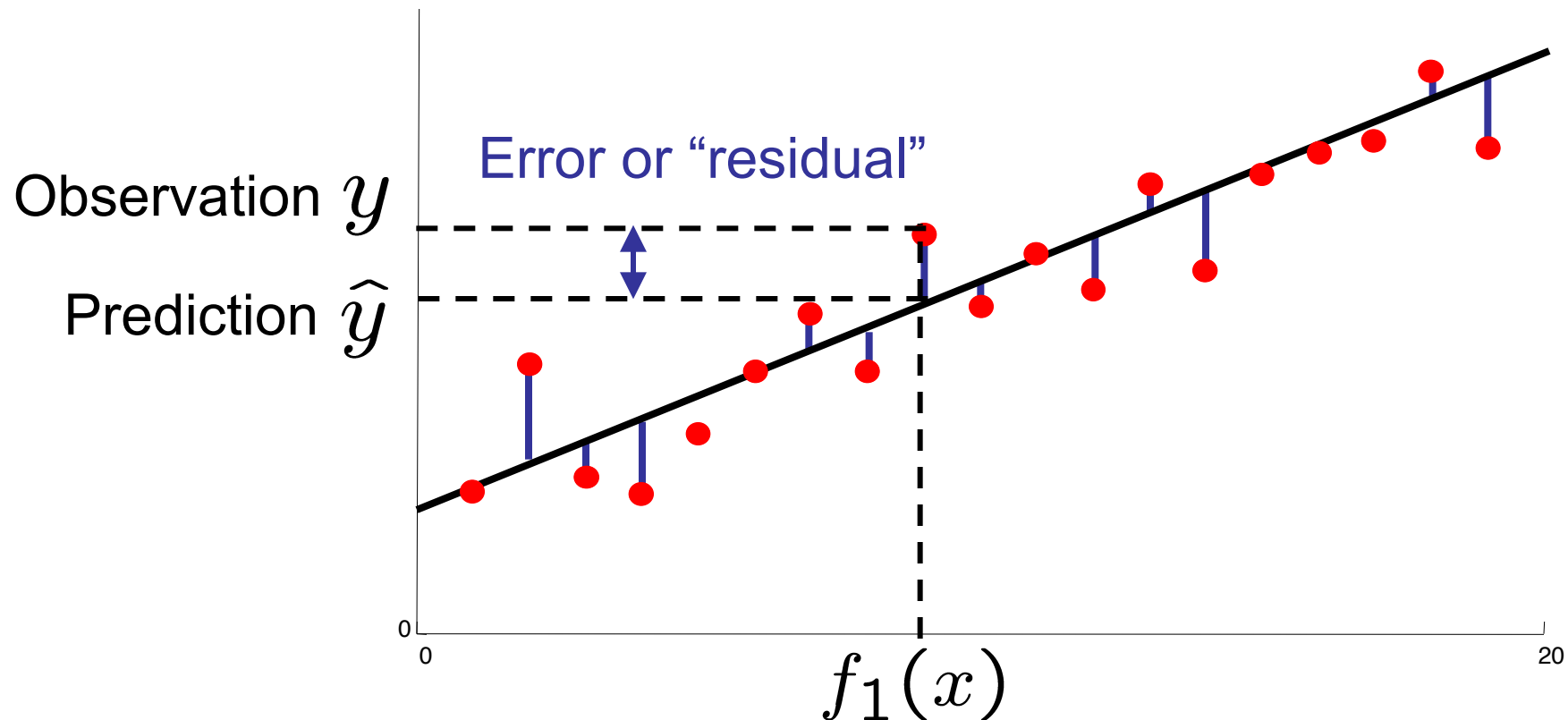
Minimizing Error*

Imagine we had only one point x, with features f(x), target value y, and weights w:

$$\text{error}(w) = \frac{1}{2}\left(y - \sum_k w_k f_k(x)\right)^2$$

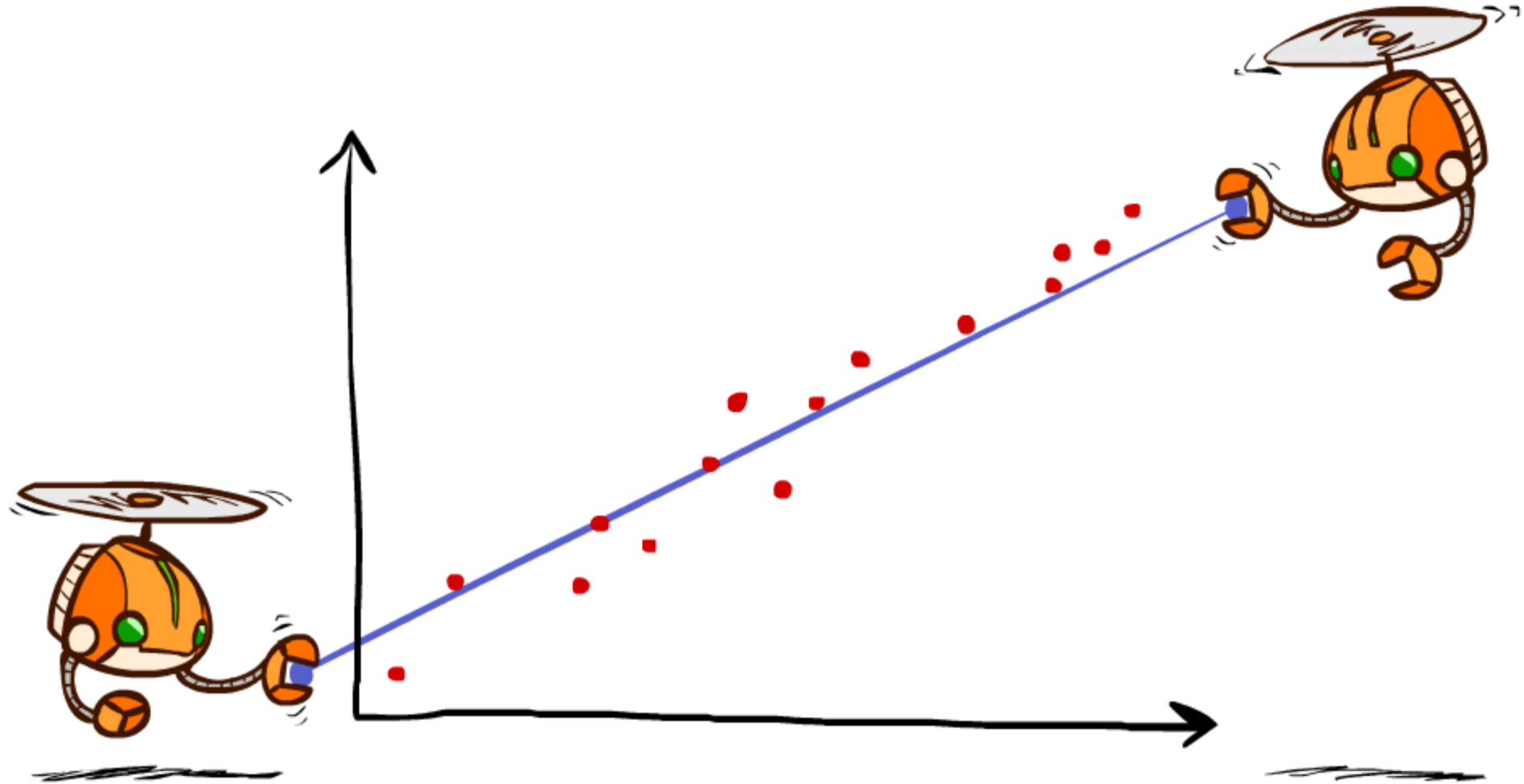$$\frac{\partial\ \text{error}(w)}{\partial w_m} = -\left(y - \sum_k w_k f_k(x)\right) f_m(x)$$

$$w_m \leftarrow w_m + \alpha\left(y - \sum_k w_k f_k(x)\right) f_m(x)$$
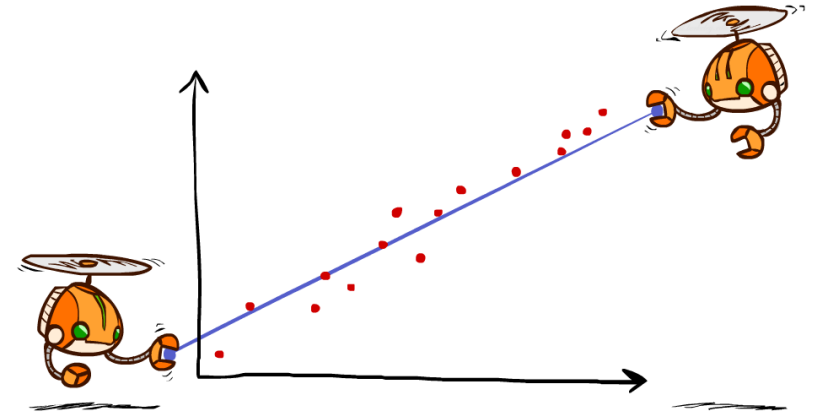
Approximate q update explained:

$$w_m \leftarrow w_m + \alpha\left[r + \gamma \max_a Q(s', a') - Q(s, a)\right] f_m(s, a)$$

"target"   "prediction"

Overfitting: Why Limiting Capacity Can Help*
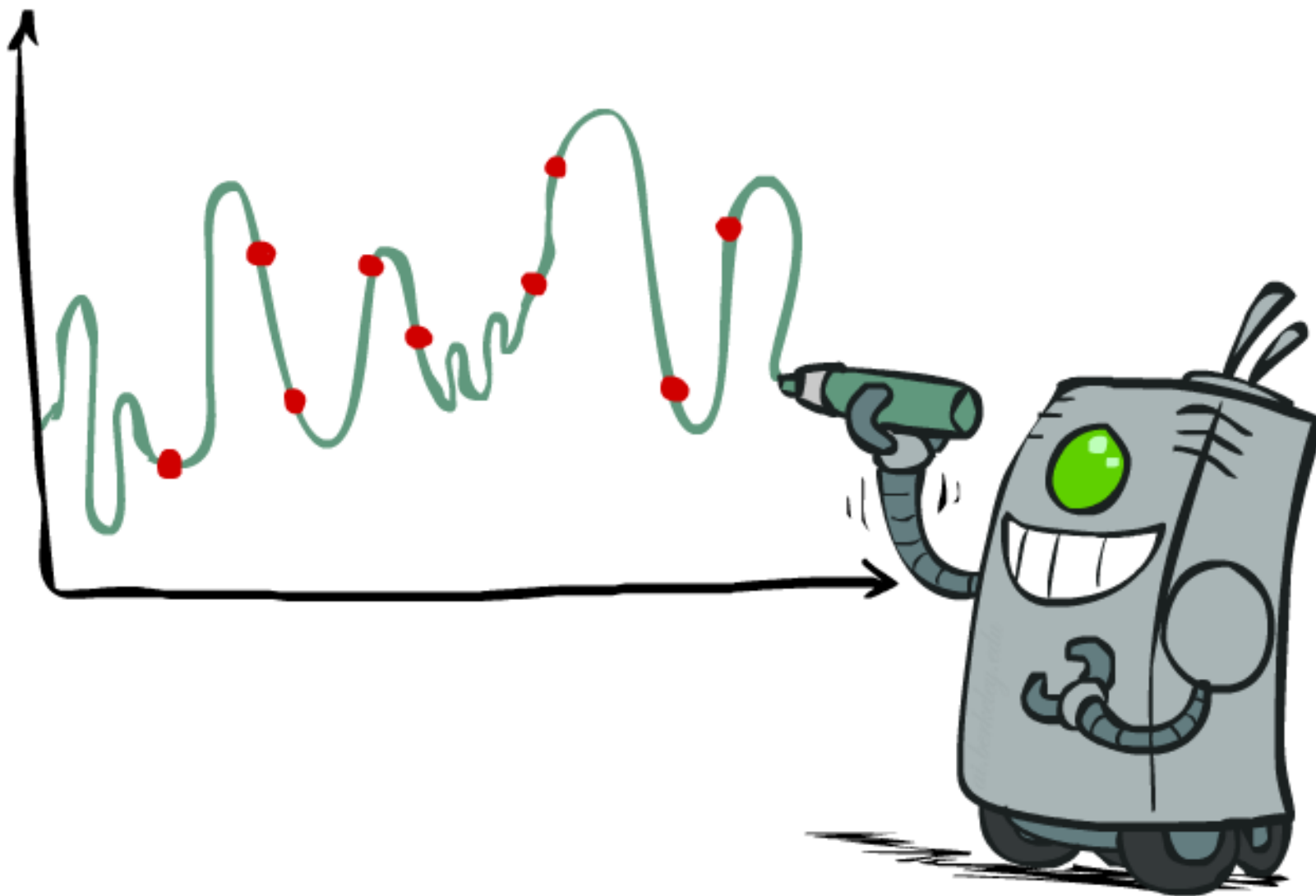
Policy Search
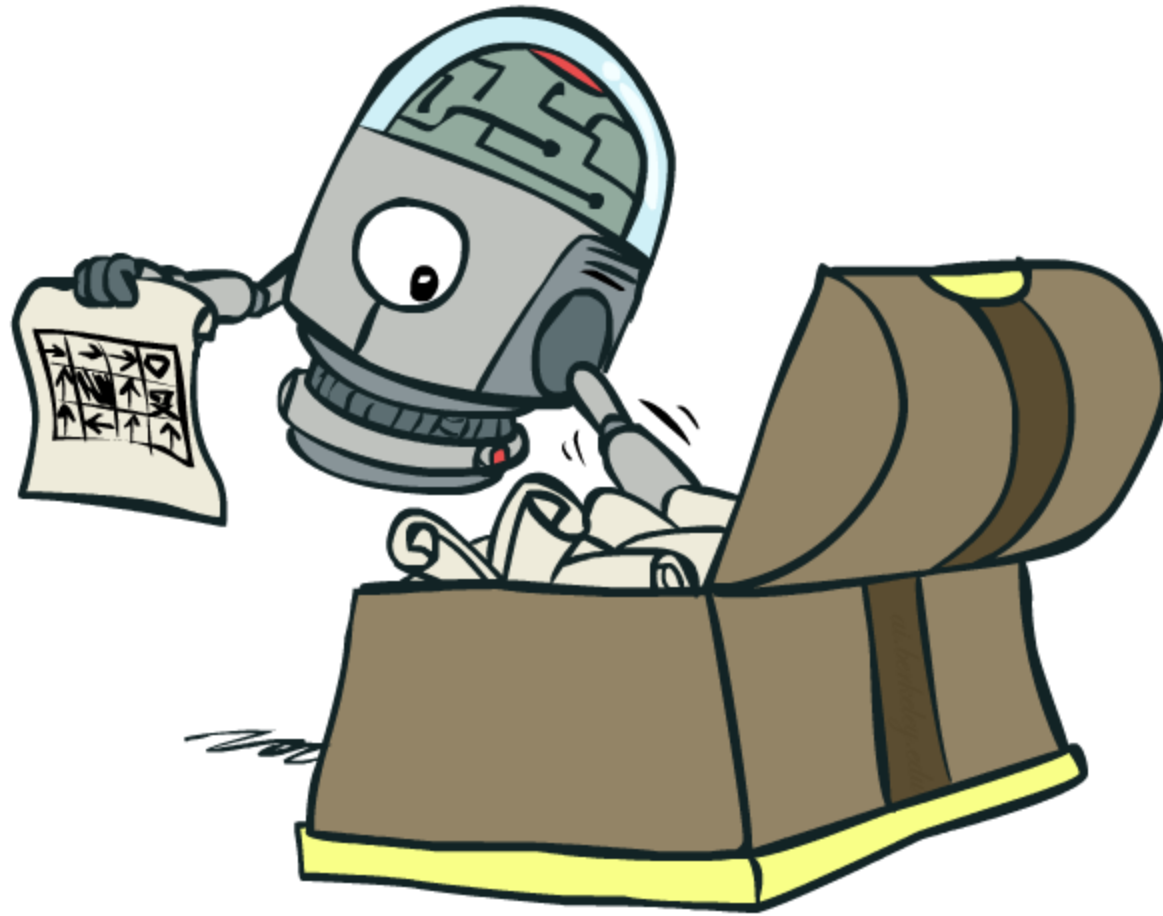
Policy Search

- Problem: often the feature-based policies that work well (win games, maximize utilities) aren't the ones that approximate V / Q best
  - E.g. your value functions from project 2 were probably horrible estimates of future rewards, but they still produced good decisions
  - Q-learning's priority: get Q-values close (modeling)
  - Action selection priority: get ordering of Q-values right (prediction)
  - We'll see this distinction between modeling and prediction again later in the course

- Solution: learn policies that maximize rewards, not the values that predict them

- Policy search: start with an ok solution (e.g. Q-learning) then fine-tune by hill climbing on feature weights

Policy Search

- Simplest policy search:
  - Start with an initial linear value function or Q-function
  - Nudge each feature weight up and down and see if your policy is better than before

- Problems:
  - How do we tell the policy got better?
  - Need to run many sample episodes!
  - If there are a lot of features, this can be impractical

- Better methods exploit look ahead structure, sample wisely, change multiple parameters…

Policy Search

[Video: HELICOPTER]
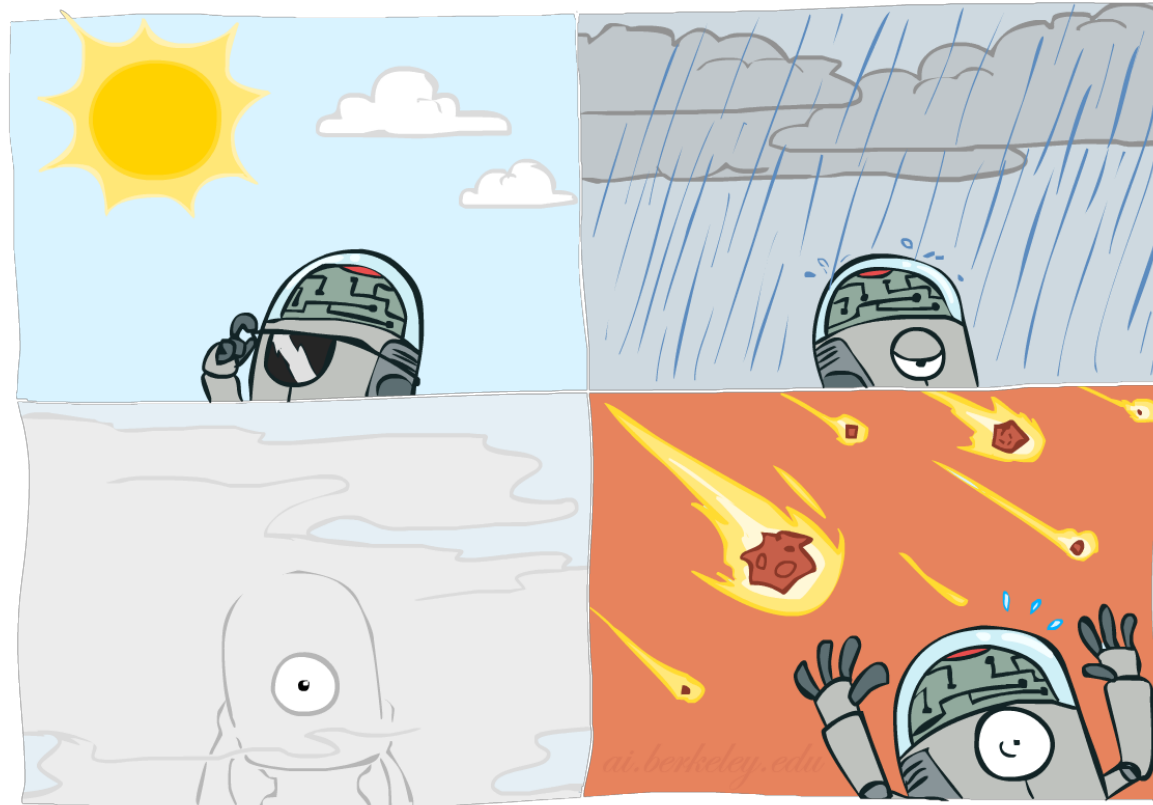
Probability

Slides courtesy of Dan Klein and Pieter Abbeel – University of California, Berkeley

[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at http://ai.berkeley.edu.]

Today

- Probability
  - Random Variables
  - Joint and Marginal Distributions
  - Conditional Distribution
  - Product Rule, Chain Rule, Bayes' Rule
  - Inference
  - Independence

- You'll need all this stuff A LOT for the next few weeks, so make sure you go over it now! Read AIMA 13.1-13.5.
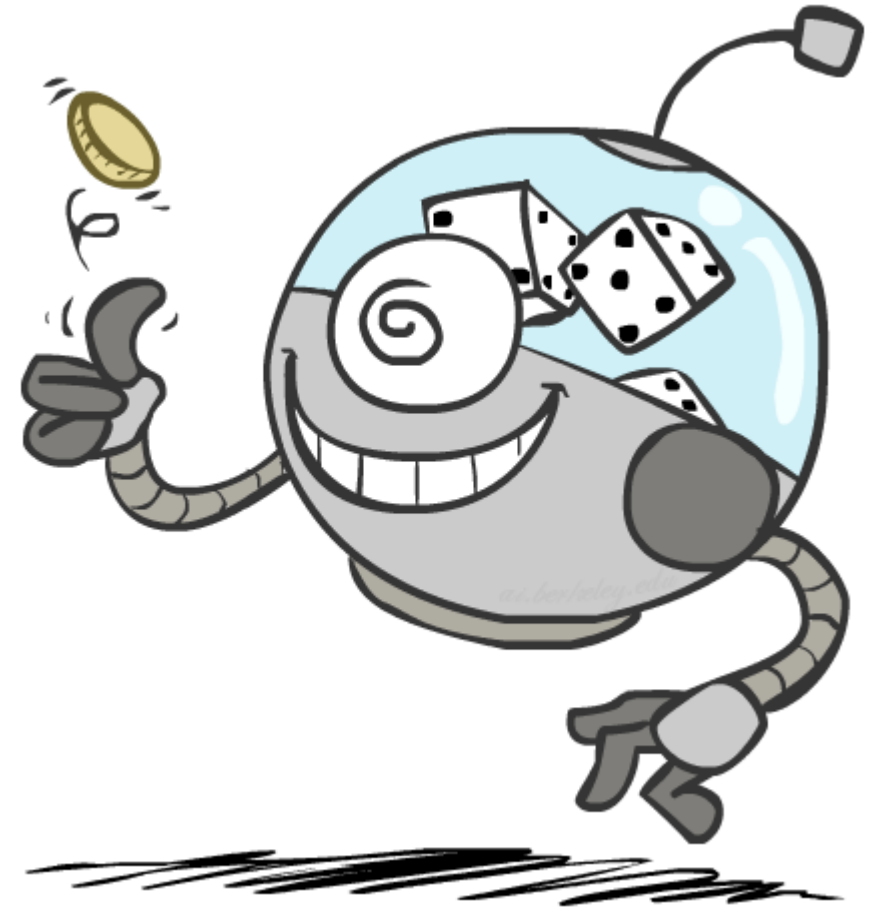
Uncertainty

- **General situation:**

  - **Observed variables (evidence)**: Agent knows certain things about the state of the world (e.g., sensor readings or symptoms)

  - **Unobserved variables**: Agent needs to reason about other aspects (e.g. where an object is or what disease is present)

  - **Model**: Agent knows something about how the known variables relate to the unknown variables

- **Probabilistic reasoning gives us a framework for managing our beliefs and knowledge**
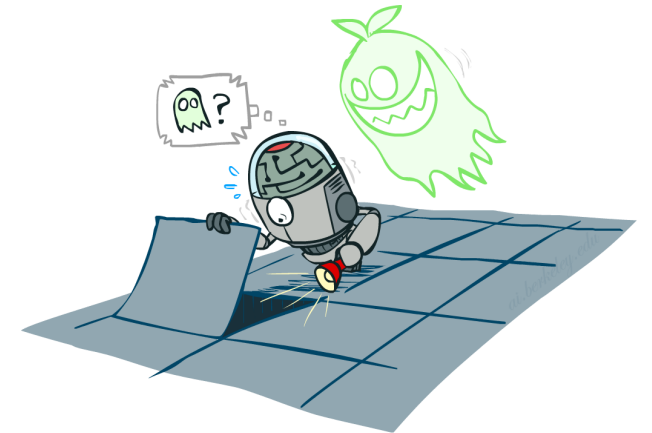
Random Variables

- A random variable is some aspect of the world about which we (may) have uncertainty

  - R = Is it raining?
  - T = Is it hot or cold?
  - D = How long will it take to drive to work?
  - L = Where is the ghost?

- We denote random variables with capital letters

- Like variables in a CSP, random variables have domains

  - R in {true, false}   (often write as {+r, -r})
  - T in {hot, cold}
  - D in [0, ∞)
  - L in possible locations, maybe {(0,0), (0,1), …}
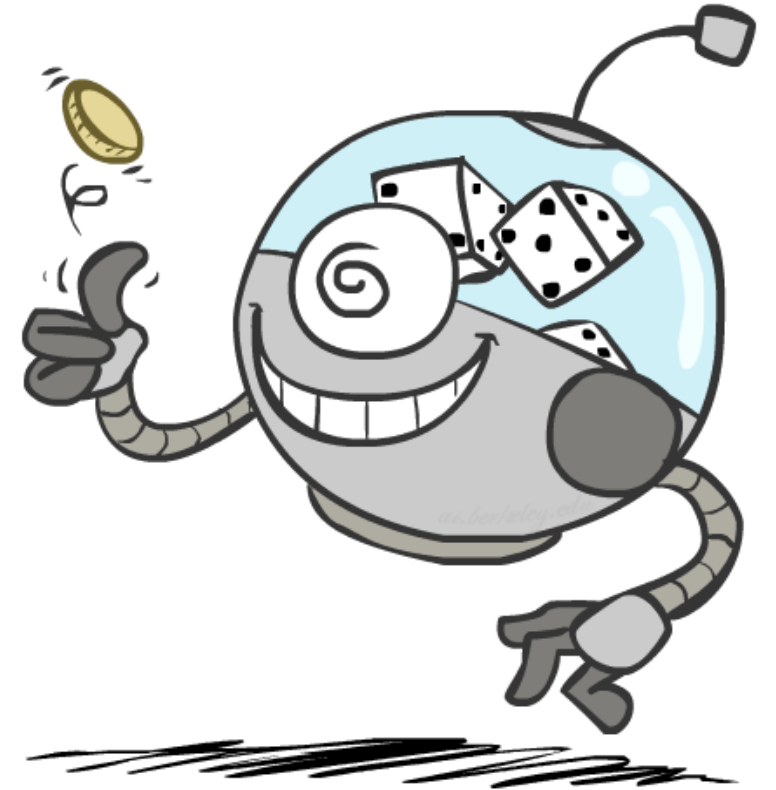
Probability Distributions

- Associate a probability with each value

  - Temperature:
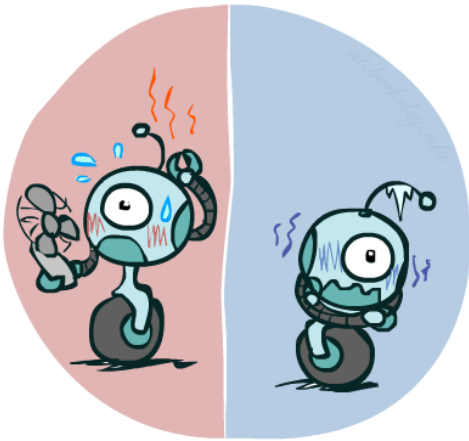
    $P(T)$

T	P
hot	0.5
cold	0.5

  - Weather:

    $P(W)$
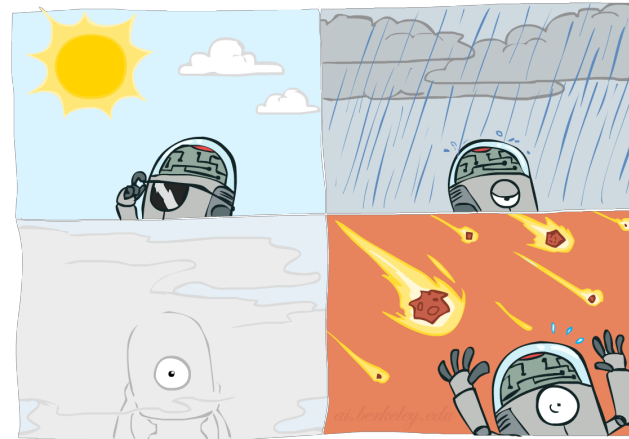
W	P
sun	0.6
rain	0.1
fog	0.3
meteor	0.0

Probability Distributions

- Unobserved random variables have distributions

$$P(T)$$

T	P
hot	0.5
cold	0.5

$$P(W)$$

W	P
sun	0.6
rain	0.1
fog	0.3
meteor	0.0

Shorthand notation:

$$P(hot) = P(T = hot),$$
$$P(cold) = P(T = cold),$$
$$P(rain) = P(W = rain),$$
$$\ldots$$

OK *if* all domain entries are unique

- A distribution is a TABLE of probabilities of values

- A probability (lower case value) is a single number

$$P(W = rain) = 0.1$$

- Must have: $\forall x \ \ P(X = x) \geq 0$ and $\sum_x P(X = x) = 1$

Joint Distributions

- A *joint distribution* over a set of random variables: $X_1, X_2, \ldots X_n$ specifies a real number for each assignment (or *outcome*):

$$P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n)$$

$$P(x_1, x_2, \ldots x_n)$$

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

- Must obey:
$$P(x_1, x_2, \ldots x_n) \geq 0$$

$$\sum_{(x_1, x_2, \ldots x_n)} P(x_1, x_2, \ldots x_n) = 1$$

- Size of distribution if n variables with domain sizes d?

  - For all but the smallest distributions, impractical to write out!

Probabilistic Models

- A probabilistic model is a joint distribution over a set of random variables

- Probabilistic models:
  - (Random) variables with domains
  - Assignments are called *outcomes*
  - Joint distributions: say whether assignments (outcomes) are likely
  - *Normalized:* sum to 1.0
  - Ideally: only certain variables directly interact

- Constraint satisfaction problems:
  - Variables with domains
  - Constraints: state whether assignments are possible
  - Ideally: only certain variables directly interact

Distribution over T,W

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

Constraint over T,W

T	W	P
hot	sun	T
hot	rain	F
cold	sun	F
cold	rain	T
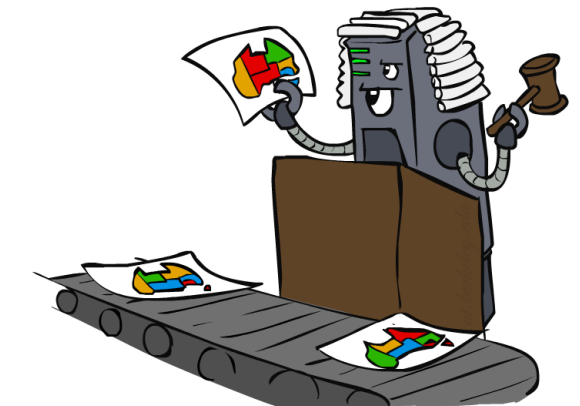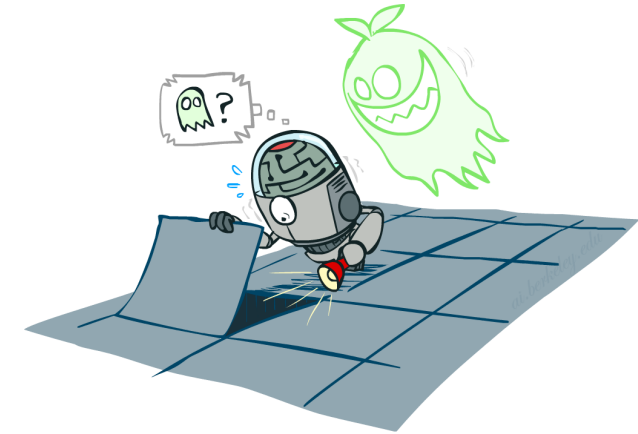
Events

- An *event* is a set E of outcomes

$$P(E) = \sum_{(x_1 \ldots x_n) \in E} P(x_1 \ldots x_n)$$

- From a joint distribution, we can calculate the probability of any event

  - Probability that it's hot AND sunny?

  - Probability that it's hot?

  - Probability that it's hot OR sunny?

- Typically, the events we care about are *partial assignments*, like P(T=hot)

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

Quiz: Events

- P(+x, +y) ?

- P(+x) ?

- P(-y OR +x) ?

$$P(X, Y)$$

X	Y	P
+x	+y	0.2
+x	-y	0.3
-x	+y	0.4
-x	-y	0.1

Marginal Distributions

- Marginal distributions are sub-tables which eliminate variables
- Marginalization (summing out): Combine collapsed rows by adding

$P(T, W)$
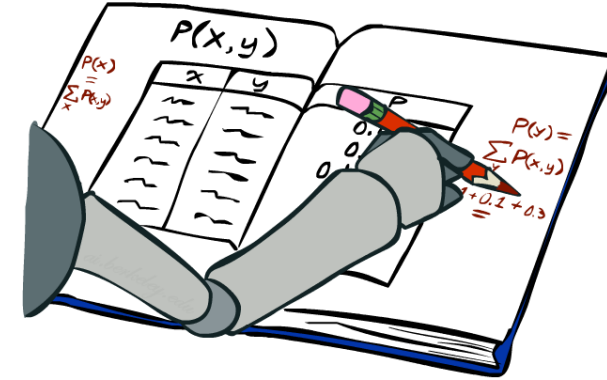
T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$$P(t) = \sum_s P(t, s)$$

$$P(s) = \sum_t P(t, s)$$

$P(T)$

T	P
hot	0.5
cold	0.5

$P(W)$

W	P
sun	0.6
rain	0.4

$$P(X_1 = x_1) = \sum_{x_2} P(X_1 = x_1, X_2 = x_2)$$

Quiz: Marginal Distributions

$P(X,Y)$

X	Y	P
+x	+y	0.2
+x	-y	0.3
-x	+y	0.4
-x	-y	0.1

$$P(x) = \sum_y P(x,y)$$

$$P(y) = \sum_x P(x,y)$$

$P(X)$

X	P
+x	
-x	

$P(Y)$

Y	P
+y	
-y	
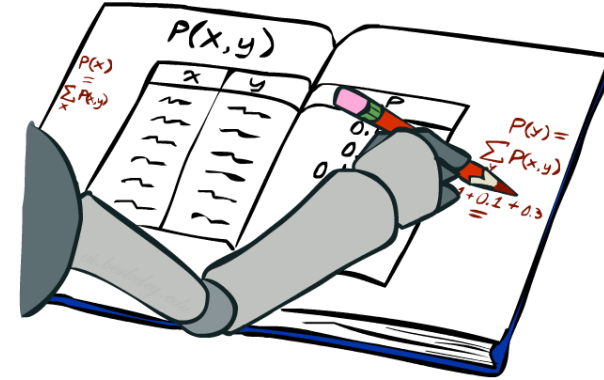
Conditional Probabilities

- A simple relation between joint and conditional probabilities
  - In fact, this is taken as the *definition* of a conditional probability

$$P(a|b) = \frac{P(a,b)}{P(b)}$$

$P(a,b)$

$P(a)$ $P(b)$

$P(T,W)$
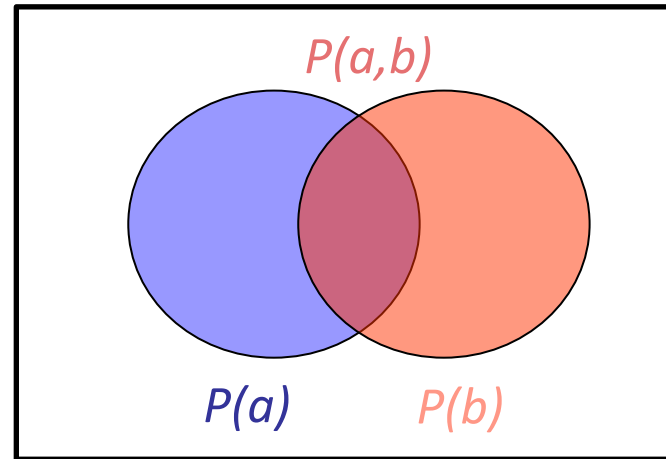
T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$$P(W = s | T = c) = \frac{P(W = s, T = c)}{P(T = c)} = \frac{0.2}{0.5} = 0.4$$

$$= P(W = s, T = c) + P(W = r, T = c)$$
$$= 0.2 + 0.3 \ = 0.5$$

Quiz: Conditional Probabilities

$P(X,Y)$

X	Y	P
+x	+y	0.2
+x	-y	0.3
-x	+y	0.4
-x	-y	0.1

- P(+x | +y) ?

- P(-x | +y) ?

- P(-y | +x) ?

Conditional Distributions

- Conditional distributions are probability distributions over some variables given fixed values of others

Joint Distribution

$P(W|T)$

$P(W|T = hot)$

W	P
sun	0.8
rain	0.2

$P(W|T = cold)$

W	P
sun	0.4
rain	0.6

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

Normalization Trick

$$P(W = s | T = c) = \frac{P(W = s, T = c)}{P(T = c)}$$

$$= \frac{P(W = s, T = c)}{P(W = s, T = c) + P(W = r, T = c)}$$

$$= \frac{0.2}{0.2 + 0.3} = 0.4$$

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$P(W | T = c)$

W	P
sun	0.4
rain	0.6

$$P(W = r | T = c) = \frac{P(W = r, T = c)}{P(T = c)}$$

$$= \frac{P(W = r, T = c)}{P(W = s, T = c) + P(W = r, T = c)}$$

$$= \frac{0.3}{0.2 + 0.3} = 0.6$$

Normalization Trick

$$P(W = s | T = c) = \frac{P(W = s, T = c)}{P(T = c)}$$

$$= \frac{P(W = s, T = c)}{P(W = s, T = c) + P(W = r, T = c)}$$

$$= \frac{0.2}{0.2 + 0.3} = 0.4$$

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

SELECT the joint probabilities matching the evidence

$\longrightarrow$

$P(c, W)$

T	W	P
cold	sun	0.2
cold	rain	0.3

NORMALIZE the selection
(make it sum to one)

$\longrightarrow$

$P(W | T = c)$

W	P
sun	0.4
rain	0.6

$$P(W = r | T = c) = \frac{P(W = r, T = c)}{P(T = c)}$$

$$= \frac{P(W = r, T = c)}{P(W = s, T = c) + P(W = r, T = c)}$$

$$= \frac{0.3}{0.2 + 0.3} = 0.6$$

Normalization Trick

$$P(T, W)$$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

SELECT the joint probabilities matching the evidence

$\longrightarrow$

$$P(c, W)$$

T	W	P
cold	sun	0.2
cold	rain	0.3

NORMALIZE the selection (make it sum to one)

$\longrightarrow$

$$P(W | T = c)$$

W	P
sun	0.4
rain	0.6

- Why does this work? Sum of selection is P(evidence)! (P(T=c), here)

$$P(x_1 | x_2) = \frac{P(x_1, x_2)}{P(x_2)} = \frac{P(x_1, x_2)}{\sum_{x_1} P(x_1, x_2)}$$

- P(X | Y=-y) ?

$P(X, Y)$

X	Y	P
+x	+y	0.2
+x	-y	0.3
-x	+y	0.4
-x	-y	0.1

SELECT the joint probabilities matching the evidence

→

NORMALIZE the selection (make it sum to one)

→

To Normalize

- (Dictionary) To bring or restore to a normal condition

All entries sum to ONE

- Procedure:
  - Step 1: Compute Z = sum over all entries
  - Step 2: Divide every entry by Z

- Example 1

W	P
sun	0.2
rain	0.3

Normalize

Z = 0.5

W	P
sun	0.4
rain	0.6

- Example 2

T	W	P
hot	sun	20
hot	rain	5
cold	sun	10
cold	rain	15

Normalize

Z = 50

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

Probabilistic Inference

- Probabilistic inference: compute a desired probability from other known probabilities (e.g. conditional from joint)

- We generally compute conditional probabilities
  - P(on time | no reported accidents) = 0.90
  - These represent the agent's *beliefs* given the evidence

- Probabilities change with new evidence:
  - P(on time | no accidents, 5 a.m.) = 0.95
  - P(on time | no accidents, 5 a.m., raining) = 0.80
  - Observing new evidence causes *beliefs to be updated*

Inference by Enumeration

- General case:
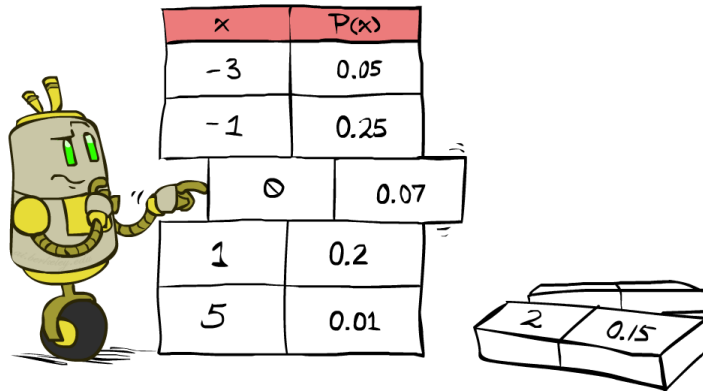 - Evidence variables:
  - Query* variable:
  - Hidden variables:

$$E_1 \ldots E_k = e_1 \ldots e_k$$
$$Q$$
$$H_1 \ldots H_r$$

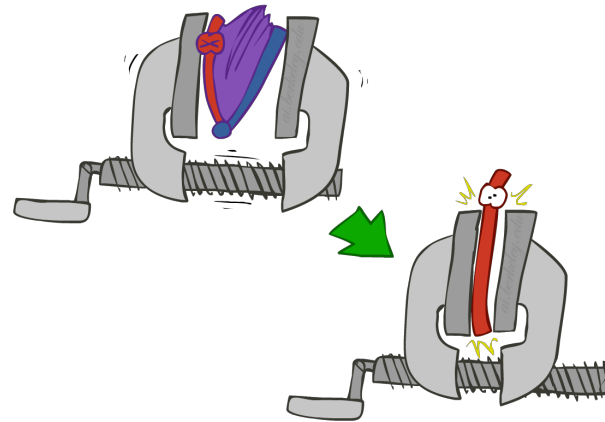$$\left.\right\} X_1, X_2, \ldots X_n$$

All variables

- We want:

Works fine with multiple query variables, too

$$P(Q|e_1 \ldots e_k)$$

- Step 1: Select the entries consistent with the evidence

- Step 2: Sum out H to get joint of Query and evidence

- Step 3: Normalize

$$\times \frac{1}{Z}$$

$$Z = \sum_q P(Q, e_1 \cdots e_k)$$

$$P(Q|e_1 \cdots e_k) = \frac{1}{Z} P(Q, e_1 \cdots e_k)$$

$$P(Q, e_1 \ldots e_k) = \sum_{h_1 \ldots h_r} P(\underbrace{Q, h_1 \ldots h_r, e_1 \ldots e_k}_{X_1, X_2, \ldots X_n})$$

Inference by Enumeration

- P(W)?

- P(W | winter)?

- P(W | winter, hot)?

S	T	W	P
summer	hot	sun	0.30
summer	hot	rain	0.05
summer	cold	sun	0.10
summer	cold	rain	0.05
winter	hot	sun	0.10
winter	hot	rain	0.05
winter	cold	sun	0.15
winter	cold	rain	0.20

Inference by Enumeration
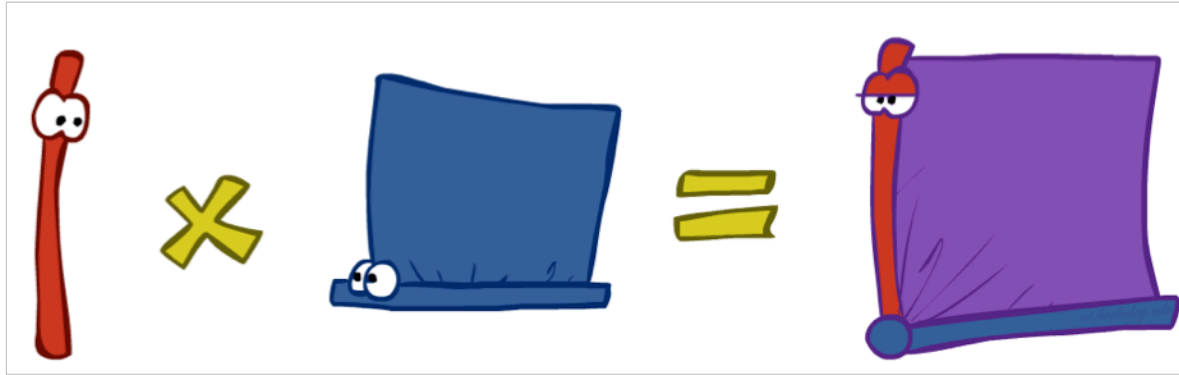
- Obvious problems:

    - Worst-case time complexity $O(d^n)$

    - Space complexity $O(d^n)$ to store the joint distribution

The Product Rule

- Sometimes have conditional distributions but want the joint

$$P(y)P(x|y) = P(x, y) \qquad \Longleftrightarrow \qquad P(x|y) = \frac{P(x, y)}{P(y)}$$
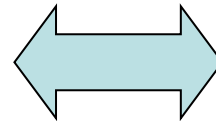
The Product Rule

$$P(y)P(x|y) = P(x,y)$$

- Example:

$P(W)$

R	P
sun	0.8
rain	0.2

$P(D|W)$

D	W	P
wet	sun	0.1
dry	sun	0.9
wet	rain	0.7
dry	rain	0.3

$P(D,W)$

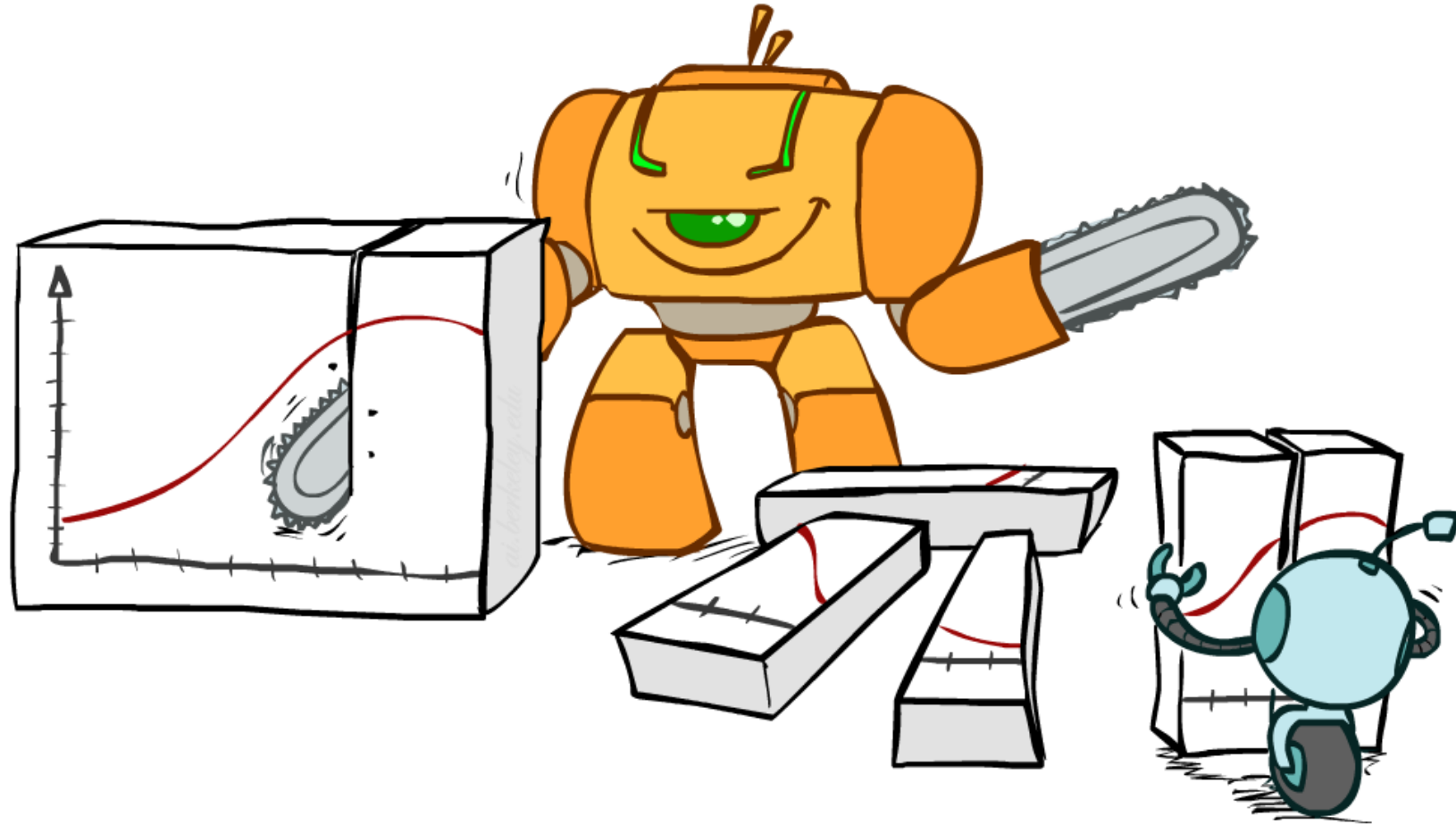| D	W	P
wet	sun	
dry	sun	
wet	rain	
dry	rain	

The Chain Rule

- More generally, can always write any joint distribution as an incremental product of conditional distributions

$$P(x_1, x_2, x_3) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2)$$

$$P(x_1, x_2, \ldots x_n) = \prod_i P(x_i|x_1 \ldots x_{i-1})$$

- Why is this always true?

Bayes' Rule

- Two ways to factor a joint distribution over two variables:

$$P(x, y) = P(x|y)P(y) = P(y|x)P(x)$$

That's my rule!

- Dividing, we get:

$$P(x|y) = \frac{P(y|x)}{P(y)}P(x)$$

- Why is this at all helpful?

  - Lets us build one conditional from its reverse
  - Often one conditional is tricky but the other one is simple
  - Foundation of many systems we'll see later (e.g. ASR, MT)

- In the running for most important AI equation!

Inference with Bayes' Rule

- Example: Diagnostic probability from causal probability:

$$P(\text{cause}|\text{effect}) = \frac{P(\text{effect}|\text{cause})P(\text{cause})}{P(\text{effect})}$$

- Example:
  - M: meningitis, S: stiff neck

$$\left.\begin{array}{l} P(+m) = 0.0001 \\ P(+s|+m) = 0.8 \\ P(+s|-m) = 0.01 \end{array}\right\} \text{Example givens}$$

$$P(+m|+s) = \frac{P(+s|+m)P(+m)}{P(+s)} = \frac{P(+s|+m)P(+m)}{P(+s|+m)P(+m) + P(+s|-m)P(-m)} = \frac{0.8 \times 0.0001}{0.8 \times 0.0001 + 0.01 \times 0.999}$$

- Note: posterior probability of meningitis still very small
- Note: you should still get stiff necks checked out! Why?

Quiz: Bayes' Rule

- Given:

$$P(W)$$

R	P
sun	0.8
rain	0.2

$$P(D|W)$$

D	W	P
wet	sun	0.1
dry	sun	0.9
wet	rain	0.7
dry	rain	0.3

- What is P(W | dry) ?

Next Time: Markov Models