

# Machine Learning & Data Mining

## **CS/CNS/EE 155**

### Lecture 17:

### The Multi-Armed Bandit Problem

# Announcements

- Lecture Tuesday will be Course Review
- Final should only take a 4-5 hours to do
  - We give you 48 hours for your flexibility
- Homework 2 is graded
  - We graded pretty leniently
  - Approximate Grade Breakdown:
    - 64: A    61: A-    58: B+    53: B    50: B-    47: C+    42: C    39: C-
- Homework 3 will be graded soon

# Today

- The Multi-Armed Bandits Problem
  - And extensions
- Advanced topics course on this next year

# Recap: Supervised Learning

- Training Data:  $S = \{(x_i, y_i)\}_{i=1}^N$   $x \in R^D$   
 $y \in \{-1, +1\}$
- Model Class:  $f(x | w, b) = w^T x - b$  **E.g., Linear Models**
- Loss Function:  $L(a, b) = (a - b)^2$  **E.g., Squared Loss**
- Learning Objective:  $\operatorname{argmin}_{w, b} \sum_{i=1}^N L(y_i, f(x_i | w, b))$

**Optimization Problem**

# But Labels are Expensive!

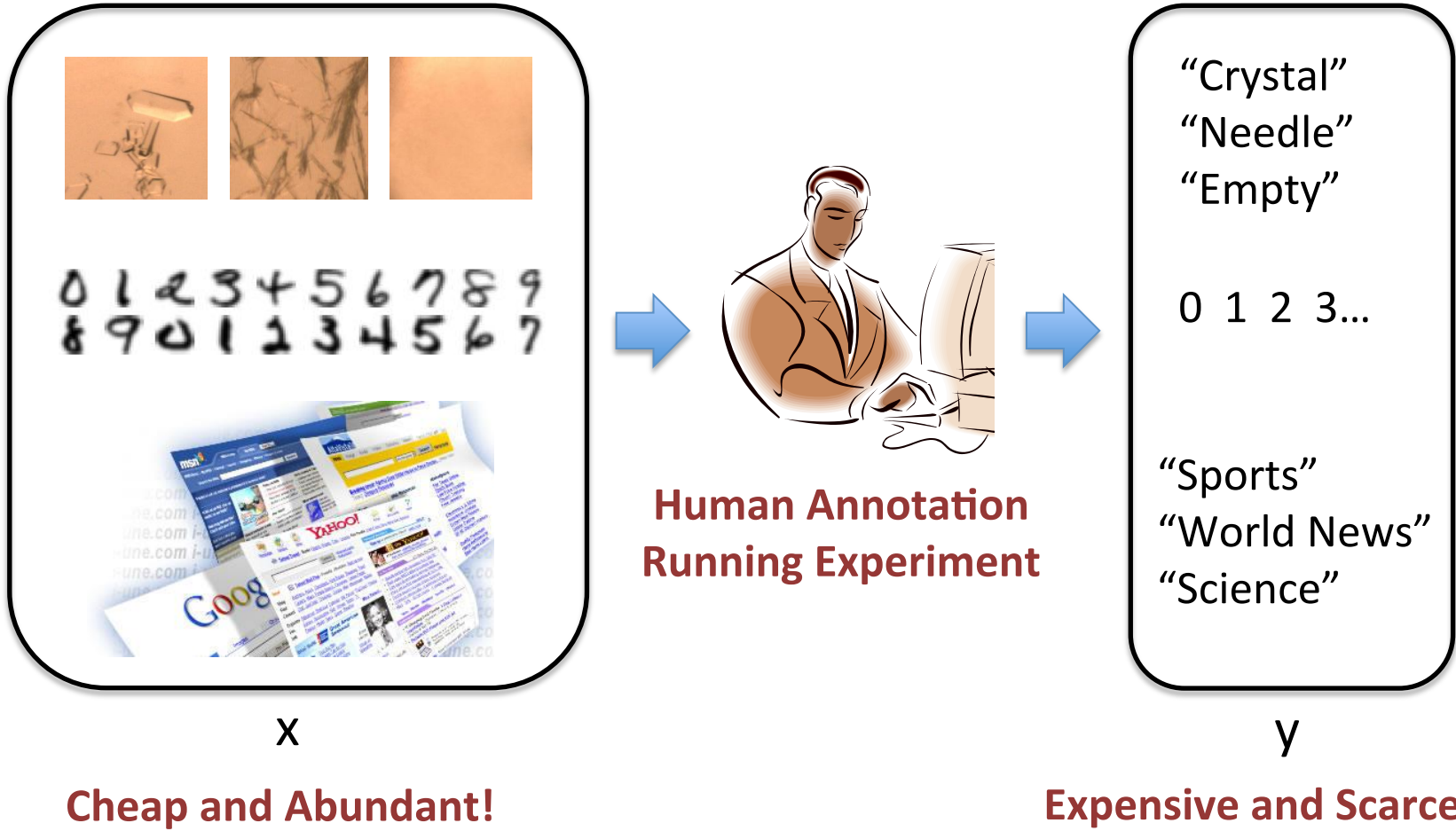


Image Source: <http://www.cs.cmu.edu/~aarti/Class/10701/slides/Lecture23.pdf>

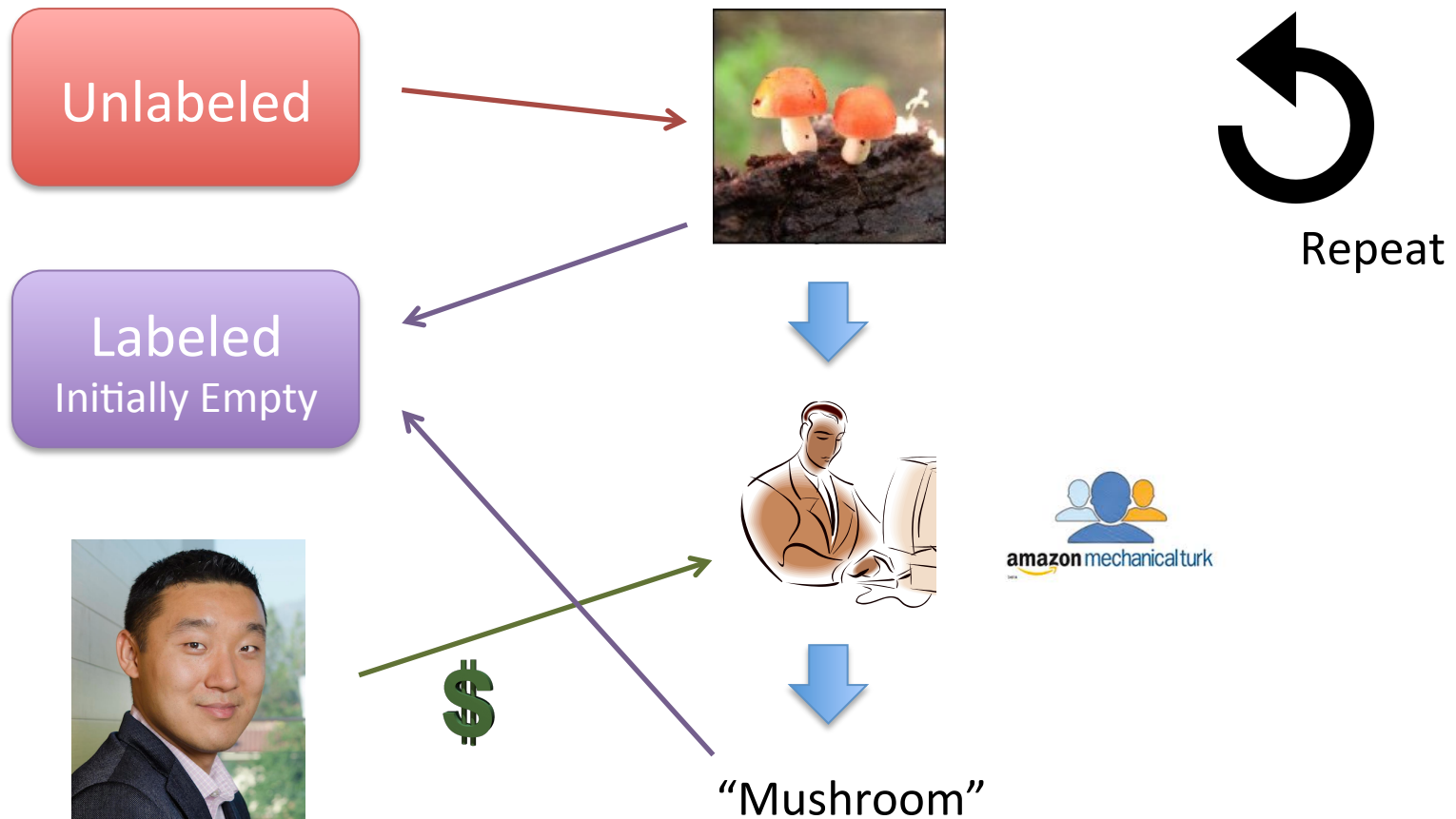
# Solution?

- Let's grab some labels!
  - Label images
  - Annotate webpages
  - Rate movies
  - Run Experiments
  - Etc...
- How should we choose?

# Interactive Machine Learning

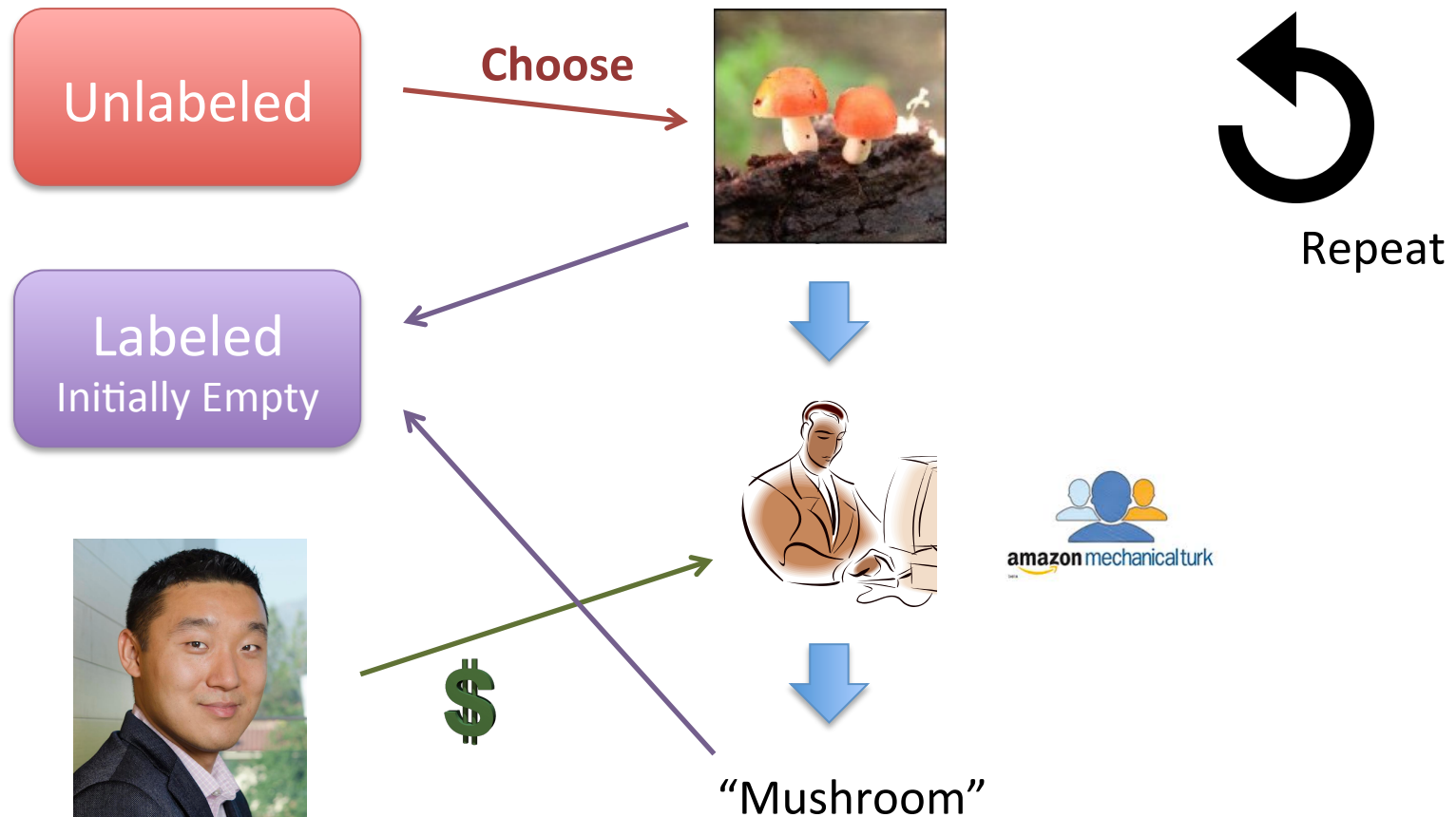
- Start with unlabeled data:
- Loop:
  - select  $x_i$
  - receive feedback/label  $y_i$
- How to measure cost?
- How to define goal?

# Crowdsourcing

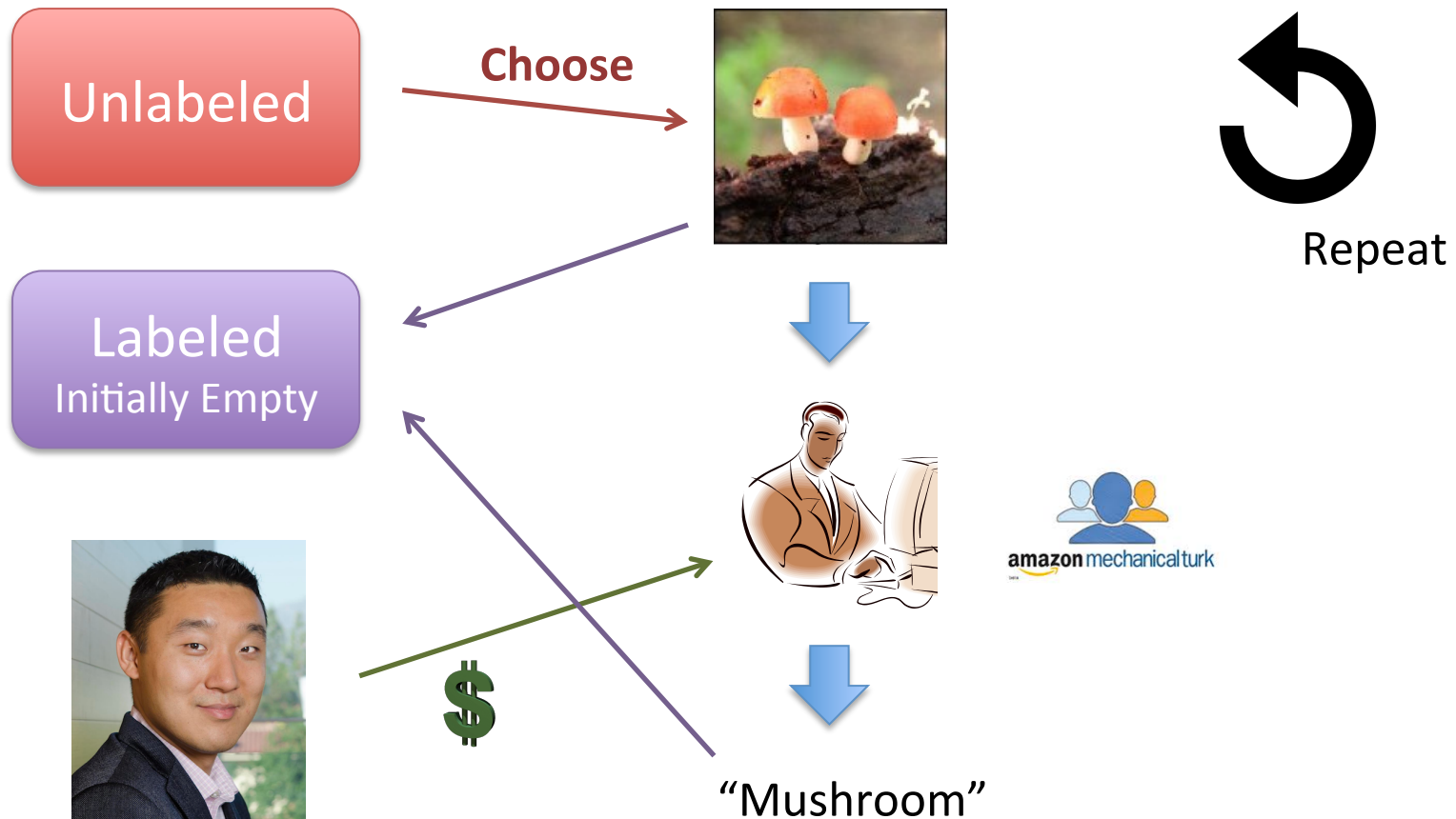




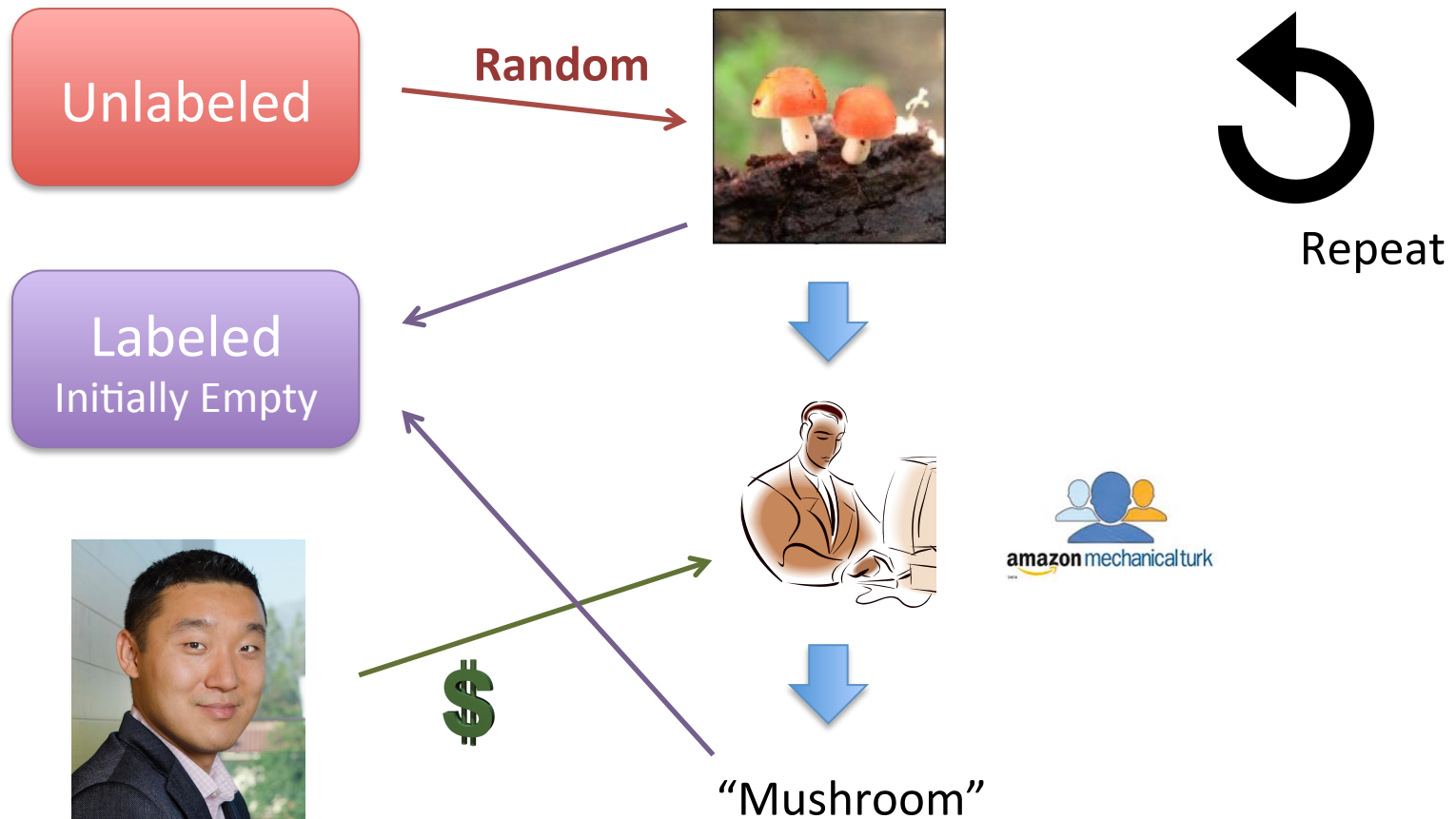
# Aside: Active Learning



# Goal: Maximize Accuracy with Minimal Cost



# Passive Learning

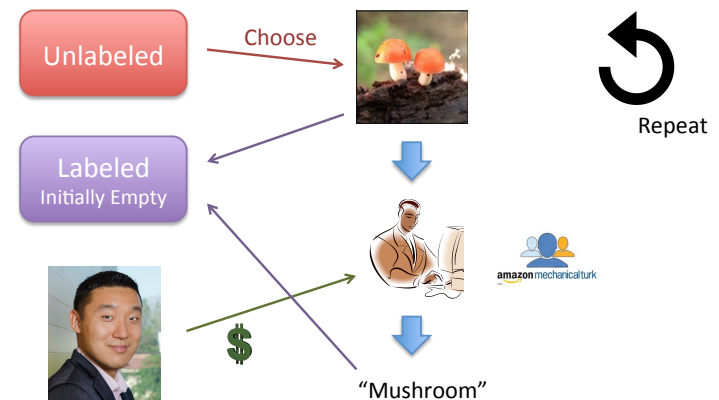


# Comparison with Passive Learning

- Conventional Supervised Learning is considered “Passive” Learning
- Unlabeled training set sampled according to test distribution
- So we label it at random
  - **Very Expensive!**

# Aside: Active Learning

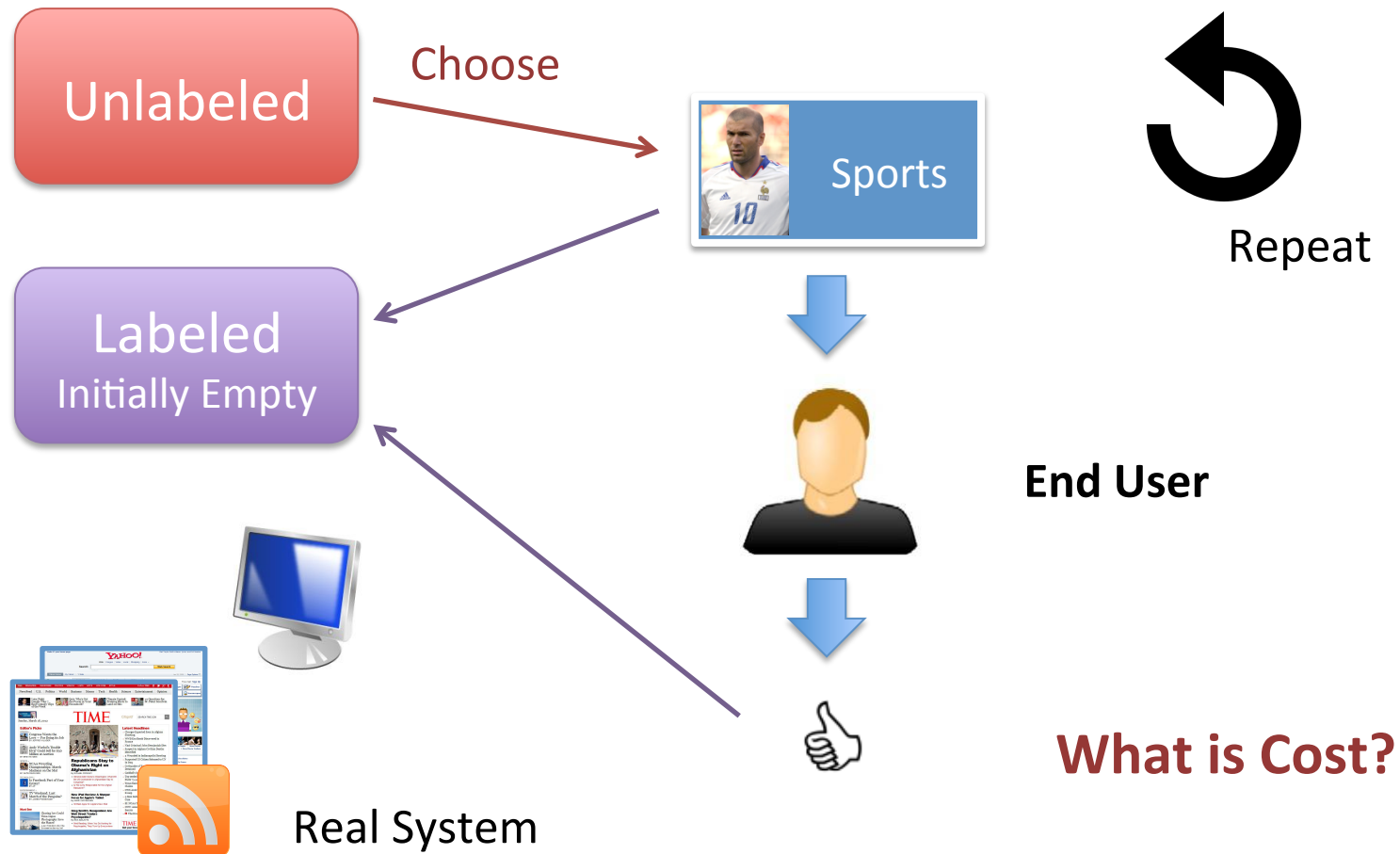
- Cost: uniform
  - E.g., each label costs \$0.10
- Goal: maximize accuracy of trained model
- Control distribution of labeled training data



# Problems with Crowdsourcing

- Assumes you can label by proxy
  - E.g., have someone else label objects in images
- But sometimes you can't!
  - Personalized recommender systems
    - Need to ask the user whether content is interesting
  - Personalized medicine
    - Need to try treatment on patient
  - **Requires actual target domain**

# Personalized Labels



# The Multi-Armed Bandit Problem

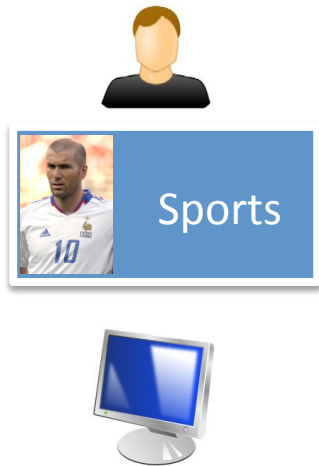


# Formal Definition

- K actions/classes
  - Each action has an average reward:  $\mu_k$ 
    - Unknown to us
    - Assume WLOG that  $u_1$  is largest
- Basic Setting  
K classes  
No features
- For  $t = 1 \dots T$ 
    - Algorithm chooses action  $a(t)$
    - Receives random reward  $y(t)$ 
      - Expectation  $\mu_{a(t)}$
- Algorithm Simultaneously  
Predicts & Receives Labels
- **Goal:** minimize  $Tu_1 - (\mu_{a(1)} + \mu_{a(2)} + \dots + \mu_{a(T)})$
- If we had perfect information to start
- Expected Reward of Algorithm

# Interactive Personalization

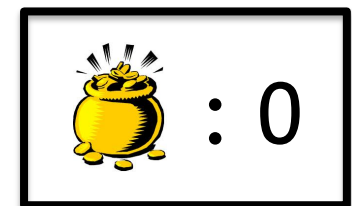
## (5 Classes, No features)



Average Likes

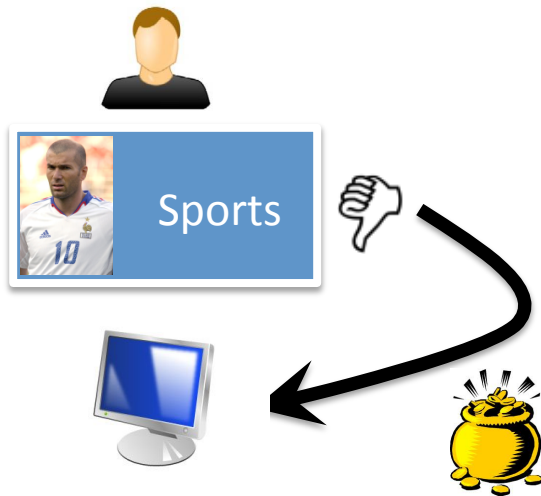
# Shown

					
Average Likes	--	--	--	--	--
# Shown	0	0	0	1	0



# Interactive Personalization

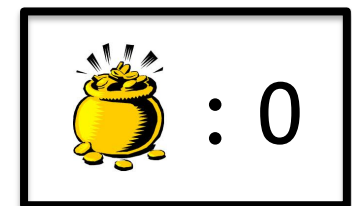
## (5 Classes, No features)



Average Likes

# Shown

					
Average Likes	--	--	--	0	--
# Shown	0	0	0	1	0



# Interactive Personalization

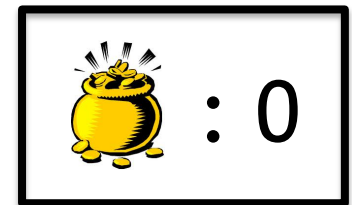
## (5 Classes, No features)



Average Likes

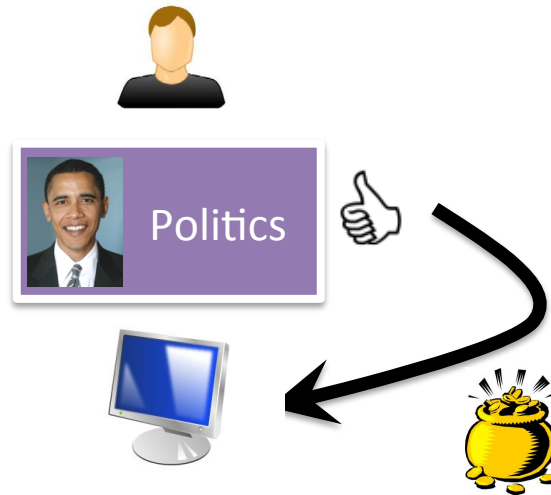
# Shown

					
Average Likes	--	--	--	0	--
# Shown	0	0	1	1	0



# Interactive Personalization

## (5 Classes, No features)



Average Likes

# Shown

					
Average Likes	--	--	1	0	--
# Shown	0	0	1	1	0



# Interactive Personalization

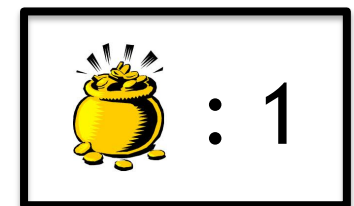
## (5 Classes, No features)



Average Likes

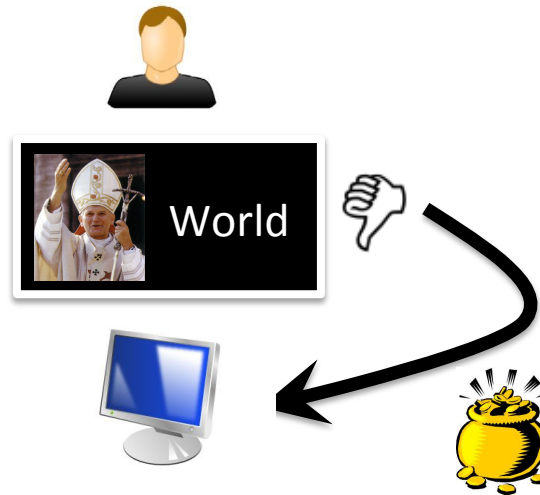
# Shown

					
Average Likes	--	--	1	0	--
# Shown	0	0	1	1	1



# Interactive Personalization

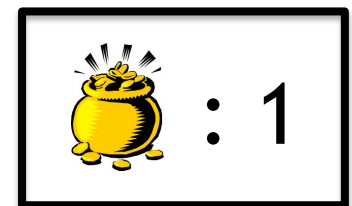
## (5 Classes, No features)



Average Likes

# Shown

					
Average Likes	--	--	1	0	0
# Shown	0	0	1	1	1



# Interactive Personalization

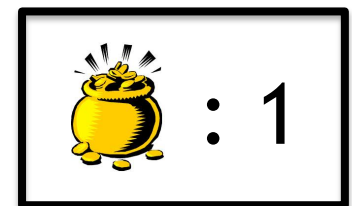
## (5 Classes, No features)



Average Likes

# Shown

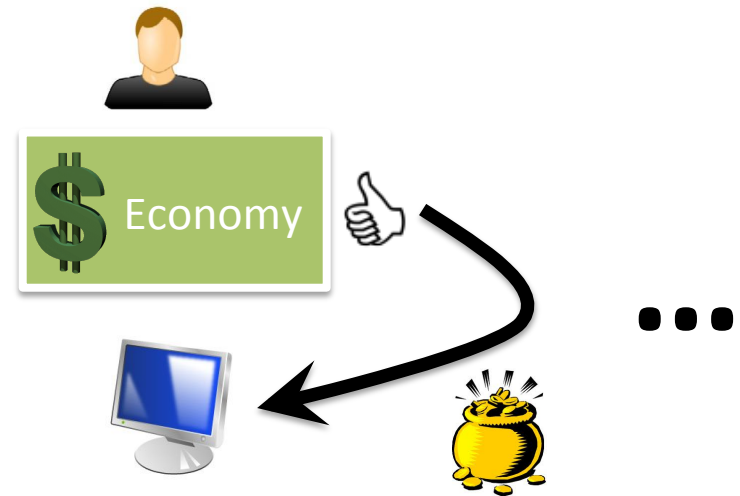
					
Average Likes	--	--	1	0	0
# Shown	0	1	1	1	1





# Interactive Personalization

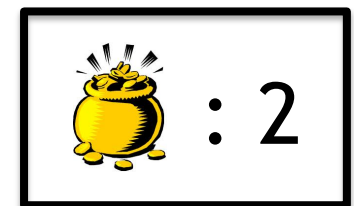
## (5 Classes, No features)



Average Likes

# Shown

					
Average Likes	--	1	1	0	0
# Shown	0	1	1	1	1

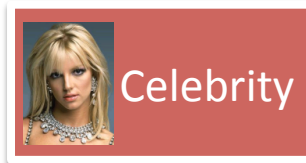


# What should Algorithm Recommend?

**Exploit:**



**Explore:**








**Best:**



**How to Optimally Balance Explore/Exploit Tradeoff?**  
Characterized by the Multi-Armed Bandit Problem

**Average Likes**

**# Shown**

					
Average Likes	--	0.44	0.4	0.33	0.2
# Shown	0	25	10	15	20



$$\text{💰}(OPT) = \text{💰}(\text{Obama}) + \text{💰}(\text{Obama}) + \text{💰}(\text{Obama}) \dots$$

$$\text{💰}(ALG) = \text{💰}(\text{Messi}) + \text{💰}(\text{Obama}) + \text{💰}(\text{Pope}) \dots$$

Time Horizon

**Regret:**  $R(T) = \text{💰}(OPT) - \text{💰}(ALG)$

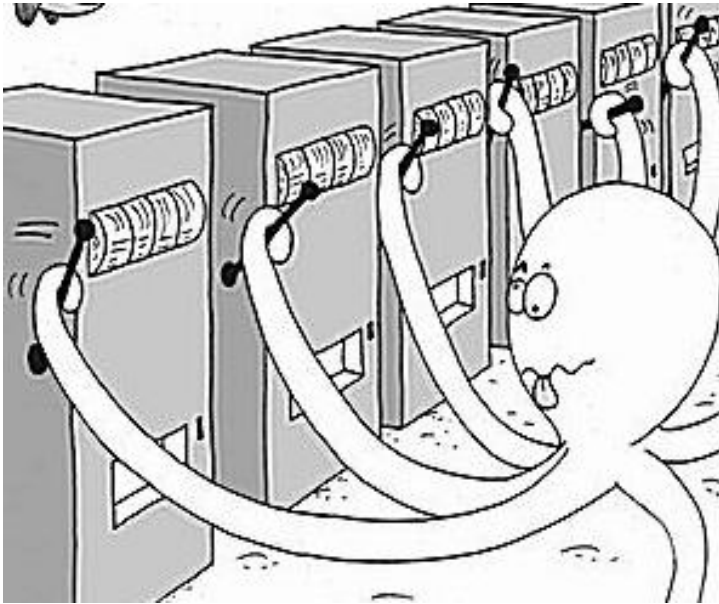
- Opportunity cost of not knowing preferences
- “no-regret” if  $R(T)/T \rightarrow 0$ 
  - Efficiency measured by convergence rate

# Recap: The Multi-Armed Bandit Problem

- K actions/classes
  - Each action has an average reward:  $\mu_k$ 
    - All unknown to us
    - Assume WLOG that  $u_1$  is largest
  - For  $t = 1 \dots T$ 
    - Algorithm chooses action  $a(t)$
    - Receives random reward  $y(t)$ 
      - Expectation  $\mu_{a(t)}$
  - Goal: minimize  $Tu_1 - (\mu_{a(1)} + \mu_{a(2)} + \dots + \mu_{a(T)})$
- Basic Setting  
K classes  
No features
- Algorithm Simultaneously  
Predicts & Receives Labels
- Regret

# The Motivating Problem

- Slot Machine = One-Armed Bandit



Each Arm Has  
Different Payoff

- **Goal:** Minimize regret From pulling suboptimal arms

[http://en.wikipedia.org/wiki/Multi-armed\\_bandit](http://en.wikipedia.org/wiki/Multi-armed_bandit)

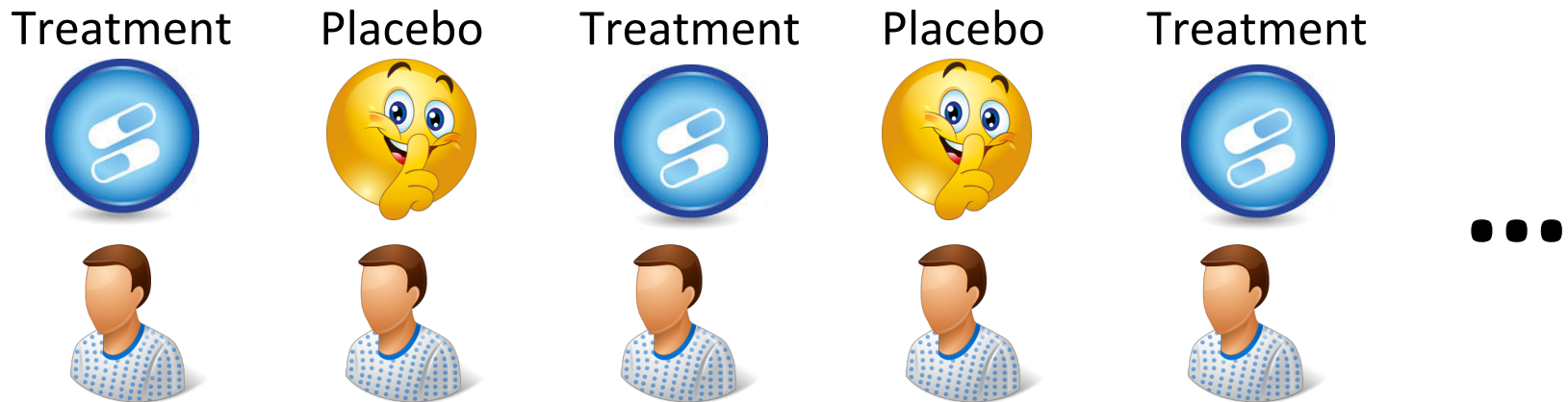
# Implications of Regret

**Regret:** 
$$R(T) = \text{💰}(OPT) - \text{💰}(ALG)$$

- If  $R(T)$  grows linearly w.r.t.  $T$ :
  - Then  $R(T)/T \rightarrow \text{constant} > 0$
  - I.e., we converge to predicting something suboptimal
- If  $R(T)$  is sub-linear w.r.t.  $T$ :
  - Then  $R(T)/T \rightarrow 0$
  - I.e., we converge to predicting the optimal action

# Experimental Design

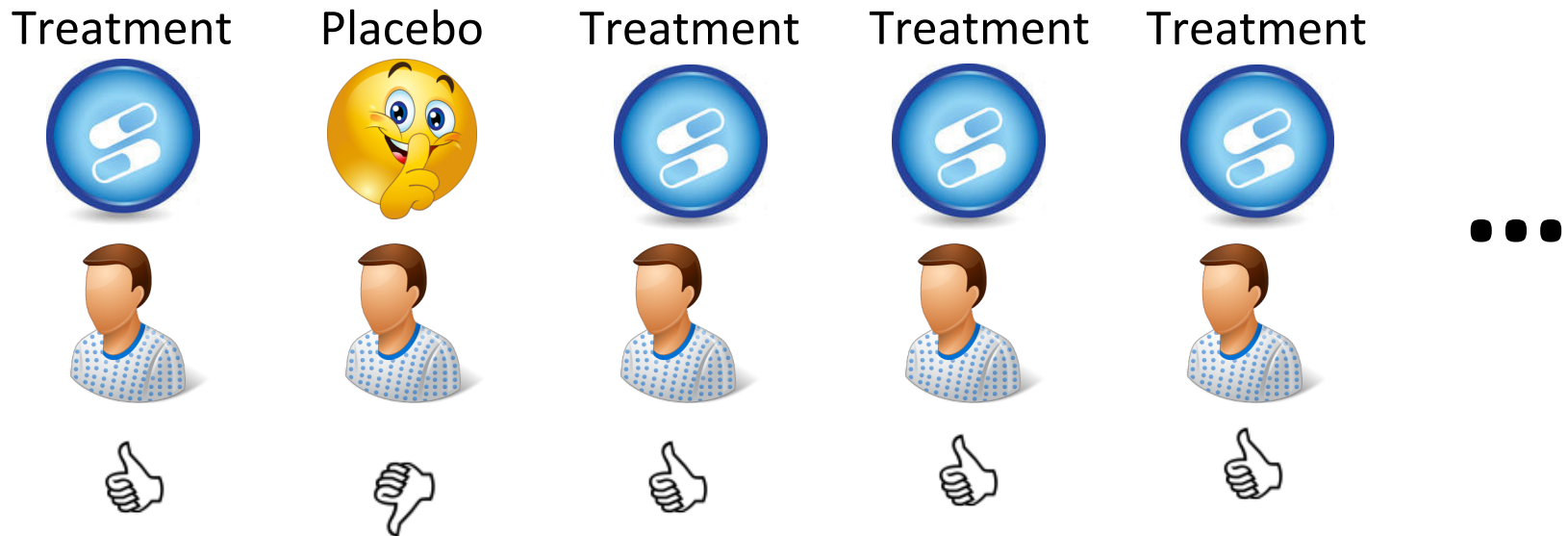
- How to split trials to collect information
- **Static Experimental Design**
  - Standard practice
  - (pre-planned)



[http://en.wikipedia.org/wiki/Design\\_of\\_experiments](http://en.wikipedia.org/wiki/Design_of_experiments)

# Sequential Experimental Design

- Adapt experiments based on outcomes





# Sequential Experimental Design Matters



Monica Almeida/The New York Times, left

**Two Cousins, Two Paths** Thomas McLaughlin, left, was given a promising experimental drug to treat his lethal skin cancer in a medical trial; Brandon Ryan had to go without it.

<http://www.nytimes.com/2010/09/19/health/research/19trial.html>

# Sequential Experimental Design

- MAB models <sup>↑  
basic</sup> sequential experimental design!
- Each treatment has hidden expected value
  - Need to run trials to gather information
  - “Exploration”
- In hindsight, should always have used treatment with highest expected value
- **Regret = opportunity cost of exploration**






# Online Advertising

macbook

Web Shopping News Images Videos More Search tools

About 97,000,000 results (0.39 seconds)

[Shop for macbook on Google](#) Sponsored ⓘ

 Apple MacBook Air... <b>\$899.00</b> Fry's Electroni... 📍 In store	 Apple® MacBook Pro... <b>\$719.00</b> Nomorack	 Refurbished Mac - MacBo... <b>\$249.00</b> Mac of All Tra...	 MacBook Pro with Retina di... <b>\$1,299.00</b> Apple Store	 Apple MacBook Pro... <b>\$550.05</b> GainSaver 👉 Special offer
--	---	---	--	--

**Official Apple Store®** ⓘ  
**Ad** [store.apple.com/MacBook](https://store.apple.com/MacBook) ▾  
4.4 ★★★★★ rating for store.apple.com  
**MacBook Pro and MacBook Air.** Free two-day shipping from Apple.  
Free iLife and iWork apps · 11, 13, or 15-inch  
📍 2126 Glendale Galleria, Glendale, CA - (818) 502-8310

<a href="#">Buy MacBook Pro</a>	<a href="#">Special Financing Offer</a>
<a href="#">Buy MacBook Air</a>	<a href="#">Free In-Store Pickup</a>

**Apple - MacBook Pro**  
<https://www.apple.com/macbook-pro/> ▾ Apple Inc. ▾  
With the latest-generation Intel processors, all-new graphics, and faster flash storage, **MacBook Pro** moves further ahead in power and performance.

<a href="#">Buy MacBook Pro with Retin...</a>	<a href="#">Compare Mac notebooks</a>
With top-of-the-line Intel processors, HD graphics, and ...	MacBook Air or iMac. No matter which Mac you choose, you're ...
<a href="#">More results from apple.com »</a>	

Largest Use-Case  
of Multi-Armed  
Bandit Problems

# The UCB1 Algorithm

<http://homes.di.unimi.it/~cesabian/Pubblicazioni/ml-02.pdf>

# Confidence Intervals

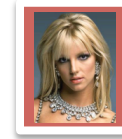
- Maintain Confidence Interval for Each Action
  - Often derived using Chernoff-Hoeffding bounds (\*\*)



= [0.1, 0.3]








= [0.25, 0.55]



Undefined

**Average Likes**

**# Shown**

					
Average Likes	--	0.44	0.4	0.33	0.2
# Shown	0	25	10	15	20

\*\* <http://www.cs.utah.edu/~jeffp/papers/Chern-Hoeff.pdf>  
[http://en.wikipedia.org/wiki/Hoeffding%27s\\_inequality](http://en.wikipedia.org/wiki/Hoeffding%27s_inequality)

# UCB1 Confidence Interval

Expected Reward  
Estimated from data




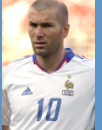

$$\bar{\mu}_k \pm \sqrt{\frac{2 \ln t}{t_k}}$$

Total Iterations so far  
(70 in example below)

#times action k was chosen

Average Likes

# Shown

					
Average Likes	--	0.44	0.4	0.33	0.2
# Shown	0	25	10	15	20






# The UCB1 Algorithm

- At each iteration
  - Play arm with highest **Upper Confidence Bound**:

$$\operatorname{argmax}_k \bar{\mu}_k + \sqrt{(2 \ln t) / t_k}$$

Average Likes

# Shown

				
--	0.44	0.4	0.33	0.2
0	25	10	15	20

# Balancing Explore/Exploit

## “Optimism in the Face of Uncertainty”

$$\operatorname{argmax}_k \bar{\mu}_k + \sqrt{(2 \ln t) / t_k}$$

Exploitation Term      Exploration Term

					
Average Likes	--	0.44	0.4	0.33	0.2
# Shown	0	25	10	15	20



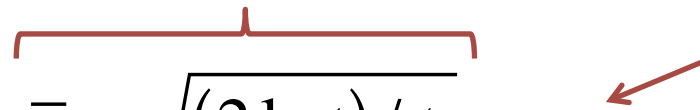
# Analysis (Intuition)

$$a(t+1) = \operatorname{argmax}_k \bar{\mu}_k + \sqrt{(2 \ln t) / t_k}$$

With high probability (\*\*):

Upper Confidence Bound of Best Arm

Value of Best Arm

$$\bar{\mu}_{a(t+1)} + \sqrt{(2 \ln t) / t_{a(t+1)}} \geq \bar{\mu}_1 + \sqrt{(2 \ln t) / t_1} \geq \mu_1$$


$$\mu_{a(t+1)} \geq \bar{\mu}_{a(t+1)} - \sqrt{(2 \ln t) / t_{a(t+1)}}$$

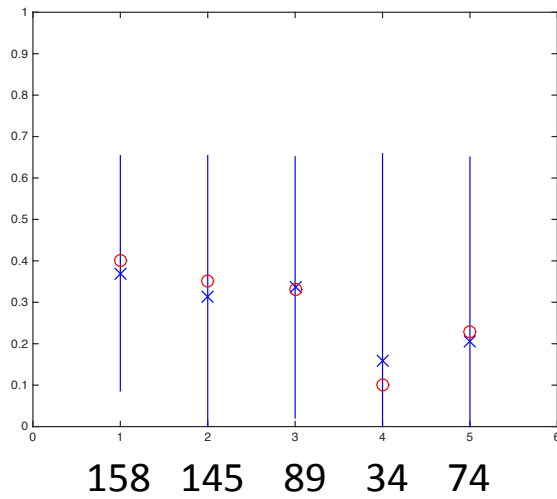
The true value is greater than the lower confidence bound.

$$\mu_1 - \mu_{a(t+1)} \leq 2\sqrt{(2 \ln t) / t_{a(t+1)}}$$

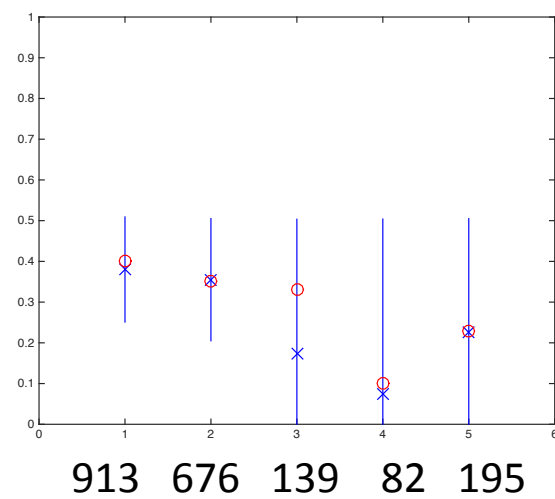
Bound on regret at time t+1

\*\* Proof of Theorem 1 in <http://homes.di.unimi.it/~cesabian/Pubblicazioni/ml-02.pdf>

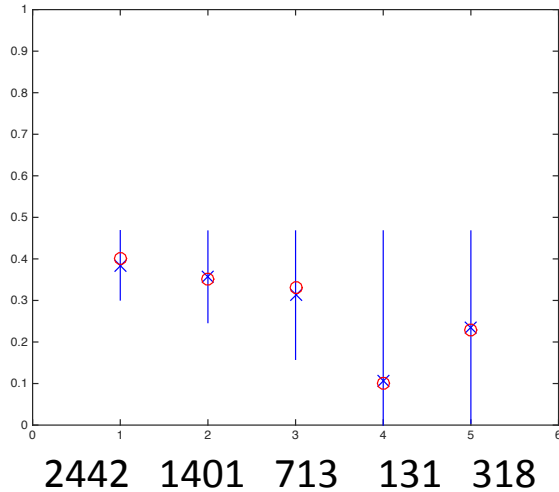
### 500 Iterations



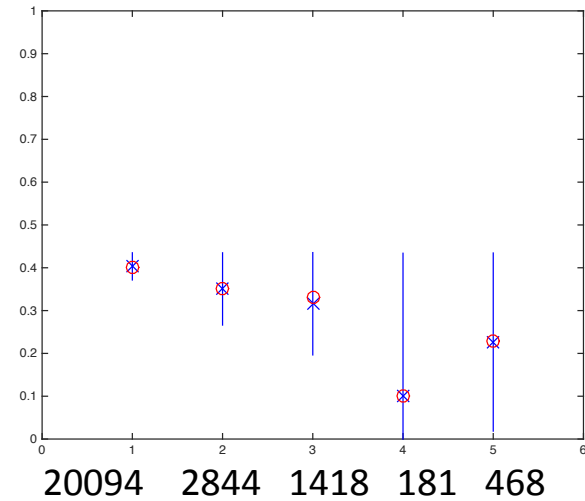
### 2000 Iterations



### 5000 Iterations



### 25000 Iterations



# How Often Sub-Optimal Arms Get Played


- An arm never gets selected if:

$$\mu_k + \sqrt{(2 \ln t) / t_k} \leq \mu_1$$

Bound grows  
slowly with time



Shrinks quickly  
with #trials



- The number of times selected:  $O\left(\frac{\ln t}{(\mu_1 - \mu_k)^2}\right)$ 
  - Prove using Hoeffding's Inequality

Theorem 1 in <http://homes.di.unimi.it/~cesabian/Pubblicazioni/ml-02.pdf>

# Regret Guarantee

- With high probability:
  - UCB1 accumulates regret at most:

$$R(T) = O\left(\frac{K}{\varepsilon} \ln T\right)$$

#Actions

Time Horizon

Gap between best & 2<sup>nd</sup> best  
 $\varepsilon = \mu_1 - \mu_2$

Theorem 1 in <http://homes.di.unimi.it/~cesabian/Pubblicazioni/ml-02.pdf>

# Recap: MAB & UCB1

- Interactive setting
  - Receives reward/label while making prediction
- Must balance explore/exploit
- Sub-linear regret is good
  - Average regret converges to 0

# Extensions

- Contextual Bandits
  - Features of environment
- Dependent-Arms Bandits
  - Features of actions/classes
- Dueling Bandits
- Combinatorial Bandits
- General Reinforcement Learning

# Contextual Bandits

- K actions/classes
  - **Rewards depends on context  $x$ :  $\mu(x)$**
- } K classes  
Best class depends on features
- For  $t = 1 \dots T$ 
    - **Algorithm receives context  $x_t$**
    - Algorithm chooses action  $a(t)$
    - Receives random reward  $y(t)$ 
      - Expectation  $\mu(x_t)$
- } Algorithm Simultaneously Predicts & Receives Labels  
Bandit multiclass prediction
- **Goal: Minimize Regret**

<http://arxiv.org/abs/1402.0555>

<http://www.research.rutgers.edu/~lihong/pub/Li10Contextual.pdf>

# Linear Bandits

- K actions/classes
  - Each action has features  $\mathbf{x}_k$
  - Reward function:  $\mu(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$

} K classes  
Linear dependence  
Between Arms

- For  $t = 1 \dots T$ 
  - Algorithm chooses action  $a(t)$
  - Receives random reward  $y(t)$ 
    - Expectation  $\mu_{a(t)}$

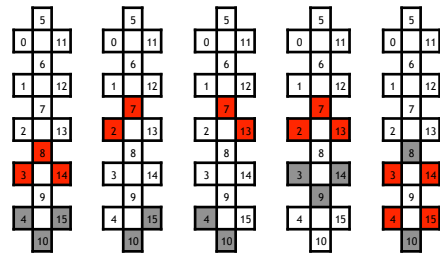
} Algorithm Simultaneously  
Predicts & Receives Labels  
Labels can share information  
to other actions

- **Goal:** regret scaling independent of K



# Example

- Treatment of spinal cord injury patients
  - Studied by Joel Burdick's group @Caltech



Want regret bound that scales independently of #arms

E.g., linearly in dimensionality of features x describing arms

- Multi-armed bandit problem:
  - Thousands of arms

UCB1 Regret Bound:

$$R(T) = O\left(\frac{K}{\epsilon} \ln T\right)$$

Images from Yanan Sui

# Dueling Bandits

- K actions/classes
    - **Preference model  $P(a_k > a_{k'})$**
  - For  $t = 1 \dots T$ 
    - **Algorithm chooses actions  $a(t)$  &  $b(t)$**
    - Receives random reward  $y(t)$ 
      - **Expectation  $P(a(t) > b(t))$**
  - **Goal:** low regret despite only pairwise feedback
- Annotations:*
- K classes*
  - Can only measure pairwise preferences*
  - Algorithm Simultaneously Predicts & Receives Labels*
  - Only pairwise rewards*

# Example in Sensory Testing

- (Hypothetical) taste experiment:
  - Natural usage context



- Experiment 1: **Absolute Metrics**

Very Thirsty!



3 cans



3 cans



2 cans



1 can



5 cans



3 cans

Total: 8 cans

Total: 9 cans

# Example in Sensory Testing

- (Hypothetical) taste experiment:
  - Natural usage context



- Experiment 1: **Relative Metrics**



2 - 1



3 - 0



2 - 0



1 - 0



4 - 1

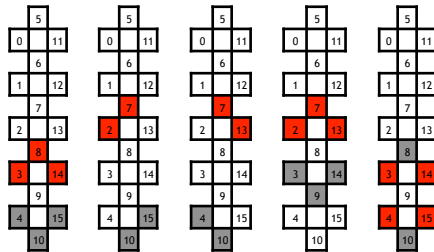


2 - 1

All 6 prefer Pepsi

# Example Revisited

- Treatment of spinal cord injury patients
  - Studied by Joel Burdick's group @Caltech



Patients cannot reliably  
rate individual treatments

Patients can reliably  
compare pairs of treatments

- Dueling Bandits Problem!

Images from Yanan Sui

<http://dl.acm.org/citation.cfm?id=2645773>

# Combinatorial Bandits

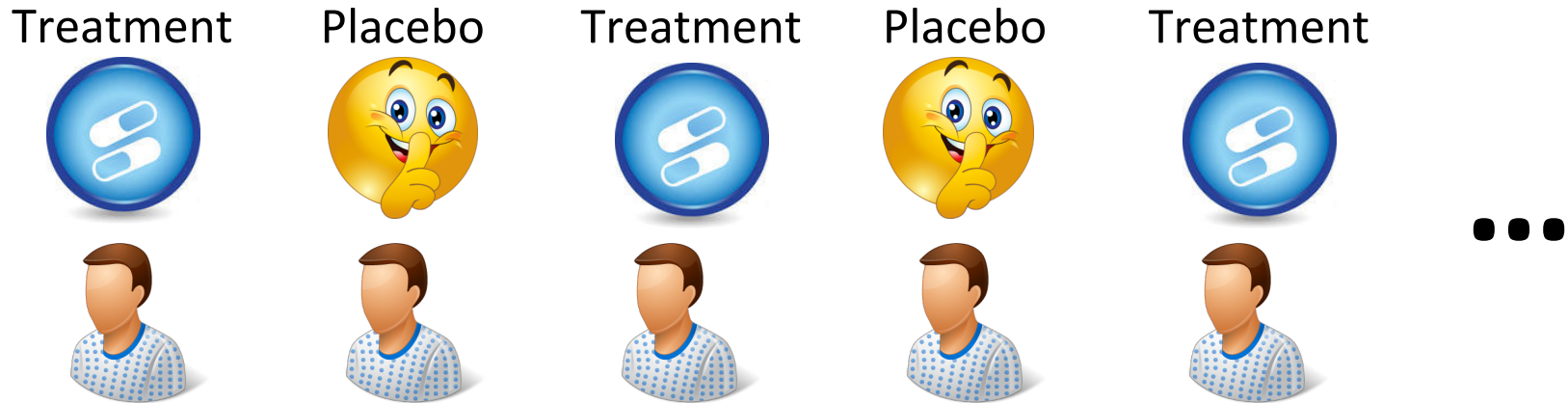
- Sometimes, actions must be selected from combinatorial action space:
  - E.g., shortest path problems with unknown costs on edges
    - aka: Routing under uncertainty
- If you knew all the parameters of model:
  - standard optimization problem

[http://www.yisongyue.com/publications/nips2011\\_submod\\_bandit.pdf](http://www.yisongyue.com/publications/nips2011_submod_bandit.pdf)

<http://www.cs.cornell.edu/~rdk/papers/OLSP.pdf>

<http://homes.di.unimi.it/cesa-bianchi/Pubblicazioni/comband.pdf>

# General Reinforcement Learning



- Bandit setting assumes actions do not affect the world
  - E.g., sequence of experiments does not affect the distribution of future trials

# Markov Decision Process

## Example: Personalized Tutoring



[Emma Brunskill et al.] (\*\*)

- $M$  states
- $K$  actions
- Reward:  $\mu(s,a)$ 
  - Depends on state
- For  $t = 1 \dots T$ 
  - Algorithm (approximately) observes current state  $s_t$ 
    - Depends on previous state & action taken
  - Algorithm chooses action  $a(t)$
  - Receives random reward  $y(t)$ 
    - Expectation  $\mu(s_t, a(t))$

\*\* [http://www.cs.cmu.edu/~ebrun/FasterTeachingPOMDP\\_planning.pdf](http://www.cs.cmu.edu/~ebrun/FasterTeachingPOMDP_planning.pdf)



# Summary

- Interactive Machine Learning
  - Multi-armed Bandit Problem
  - Basic result: UCB1
  - Surveyed Extensions
- Advanced Topics in ML course next year
- Next lecture: course review
  - Bring your questions!