

Machine Learning & Data Mining

CS/CNS/EE 155

Lecture 17:
Survey of Advanced Topics

What We Covered

Topic Overview

Supervised Learning

Linear Models

Overfitting

Loss Functions

Non-Linear Models

Learning Algorithms
& Optimization

Probabilistic Modeling

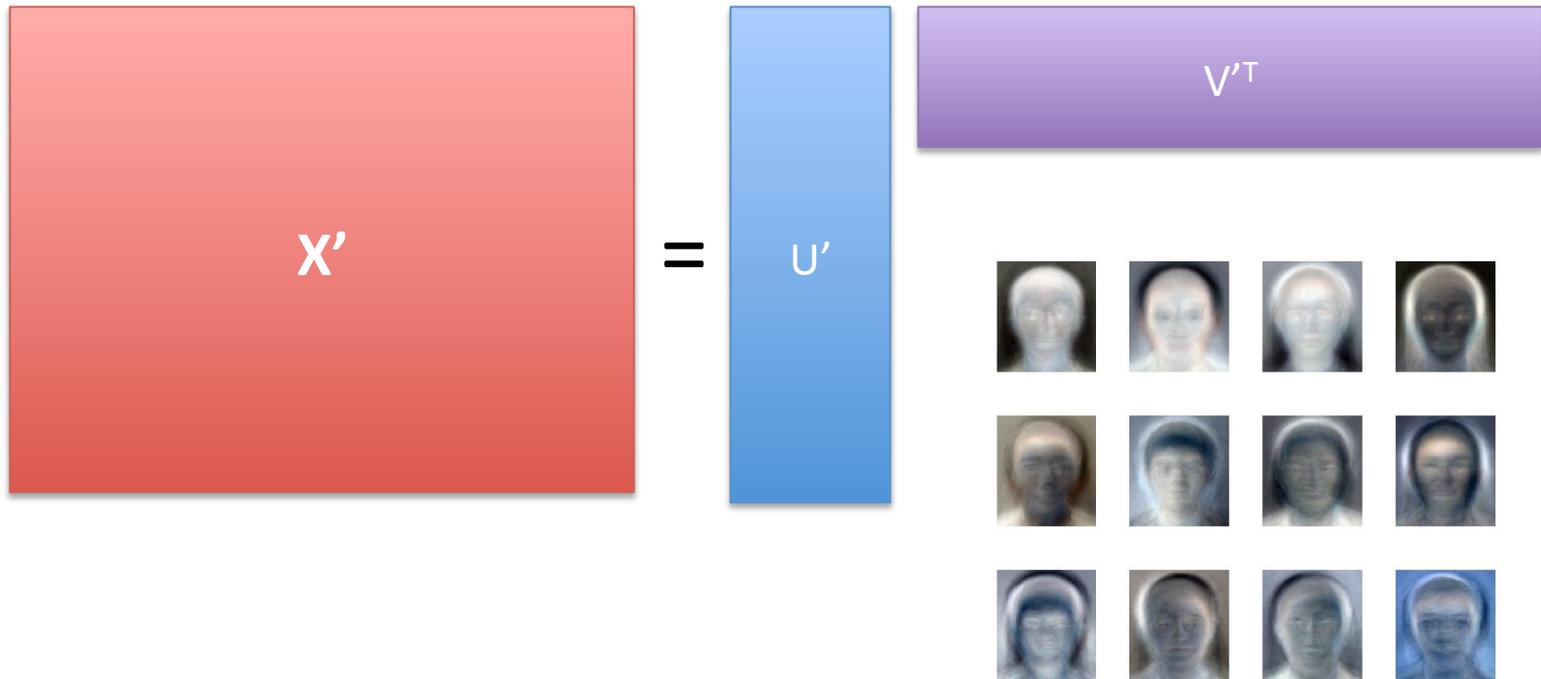
Unsupervised Learning

Basic Supervised Learning

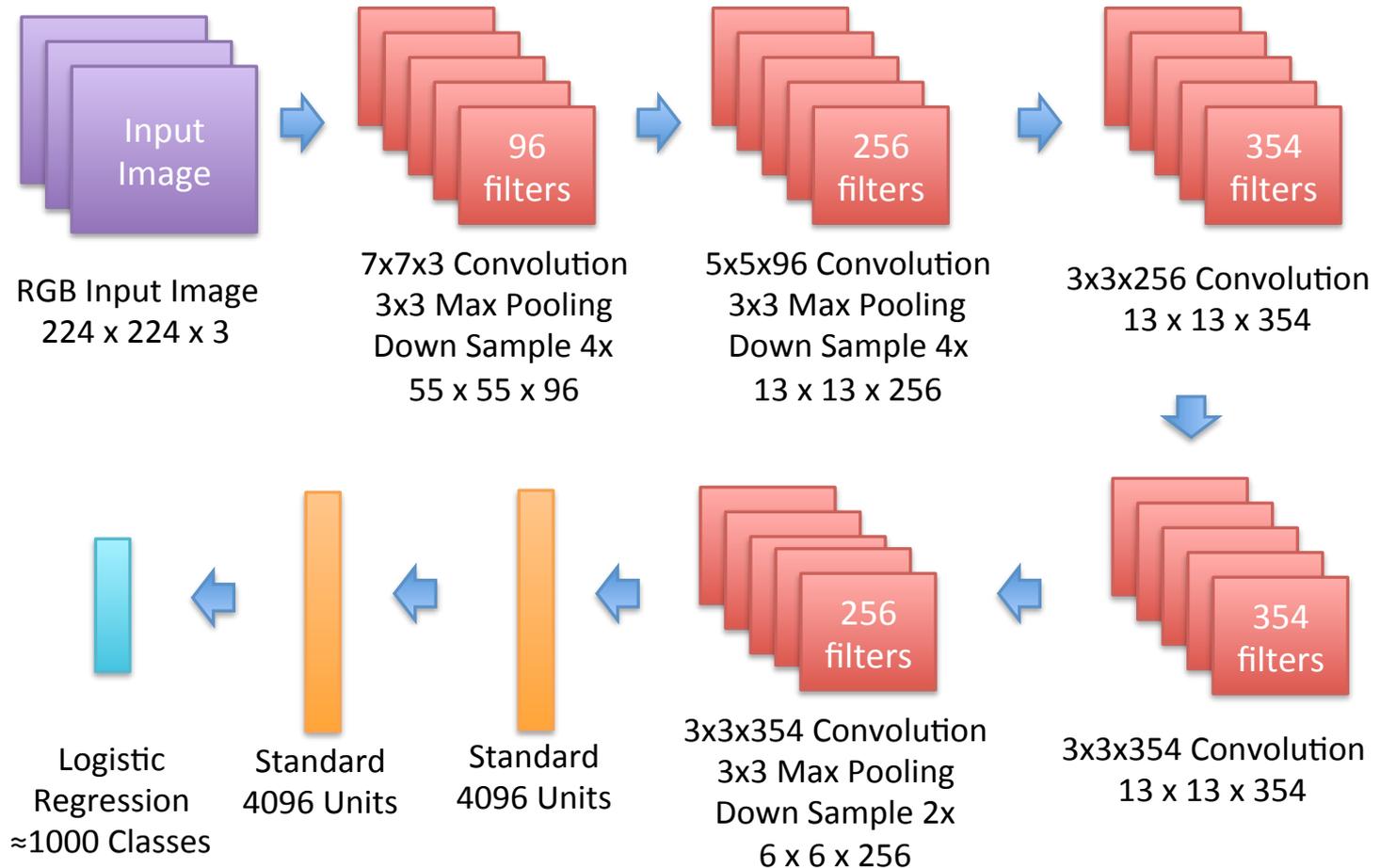
- Training Data: $S = \{(x_i, y_i)\}_{i=1}^N$ $x \in \mathbb{R}^D$
 $y \in \{-1, +1\}$
- Model Class: $f(x | w, b) = w^T x - b$ **Linear Models**
- Loss Function: $L(a, b) = (a - b)^2$ **Squared Loss**
- Learning Objective: $\operatorname{argmin}_{w, b} \sum_{i=1}^N L(y_i, f(x_i | w, b))$

Optimization Problem

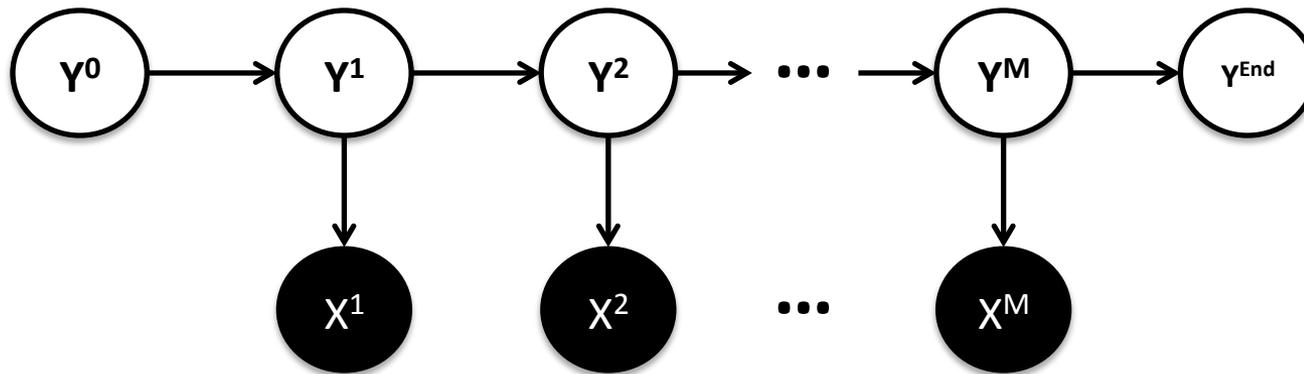
Basic Unsupervised Learning



Deep Learning



Sequence Prediction



Simple Optimization Algorithms

- Stochastic Gradient Descent
- EM algorithm (for HMMs)

Other Basic Concepts

- Cross Validation
- Overfitting
- Bias-Variance Tradeoff

Learning Theory

Generalization Bounds

- Formal characterization of overfitting
- Example result:

With Prob. $\geq 1 - \delta$:

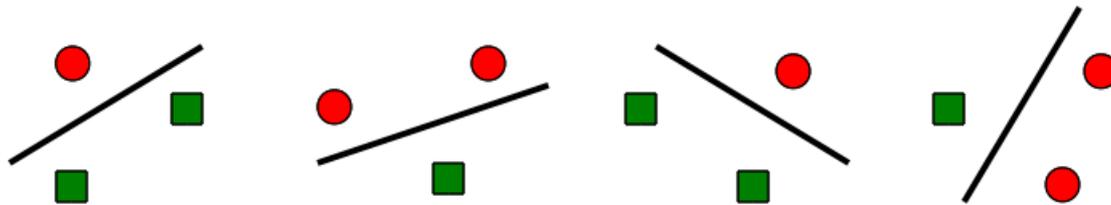
$$E_{out}(h) \leq E_{in}(h) + O\left(\frac{\log(1/\delta)}{\sqrt{N}}\right)$$

Diagram illustrating the generalization bound equation with annotations:

- Test Error**: Points to $E_{out}(h)$
- Training Error**: Points to $E_{in}(h)$
- Trained Model**: Points to the h in both $E_{out}(h)$ and $E_{in}(h)$
- Training Size**: Points to N in the denominator of the big-O term
- Make rigorous!**: A purple bracket above the big-O term indicates the need for a rigorous derivation of the constant in the big-O notation.

Shattering

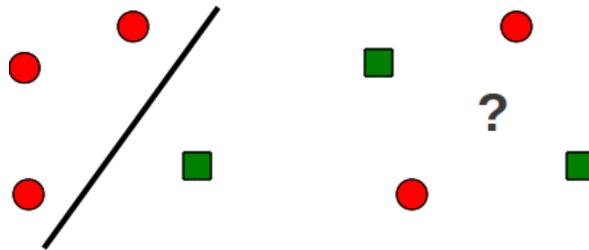
- **Definition:** A set of points is **shattered** by H if for all **possible binary labelings** of points, there exists some h that classifies perfectly.



In 2D, any 3 points can always be shattered by linear models!

Shattering

- **Definition:** A set of points is **shattered** by H if for all **possible binary labelings** of points, there exists some h that classifies perfectly.



In 2D, linear models cannot shatter 4 points!

VC Dimension

- $VC(H)$ = most # points that can be shattered
 - If H is linear models in 2D feature space:
 - $VC(H) = 3$

With Prob. $\geq 1-\delta$:

$$E_{out}(h) \leq E_{in}(h) + O \left(\sqrt{\frac{VC(H) \log \left(\frac{2N}{VC(H)} + 1 \right) + \log \left(\frac{1}{\delta} \right)}{N}} \right)$$

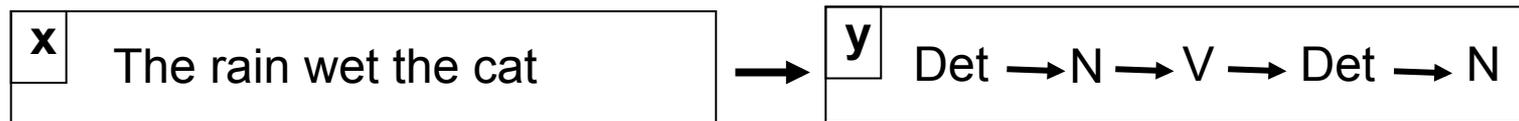
Structured Prediction

Topic of CS159

Examples of Complex Output Spaces

- Part-of-Speech Tagging

- Given a sequence of words x , predict sequence of tags y .
- Dependencies from tag-tag transitions in Markov model.



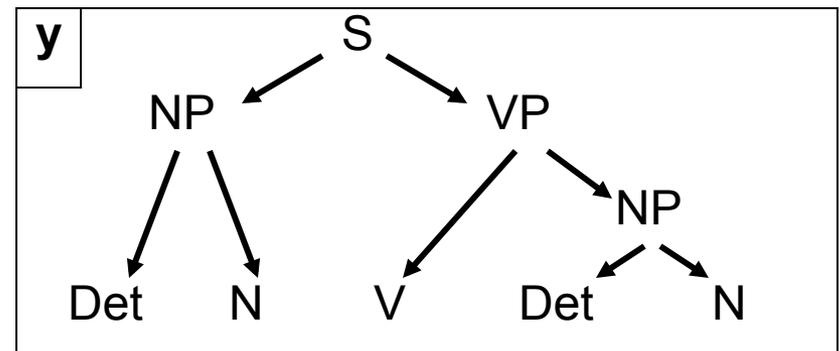
→ Similarly for other sequence labeling problems, e.g., RNA Intron/ Exon Tagging.

Examples of Complex Output Spaces

- Natural Language Parsing

- Given a sequence of words x , predict the parse tree y .
- Dependencies from structural constraints, since y has to be a tree.

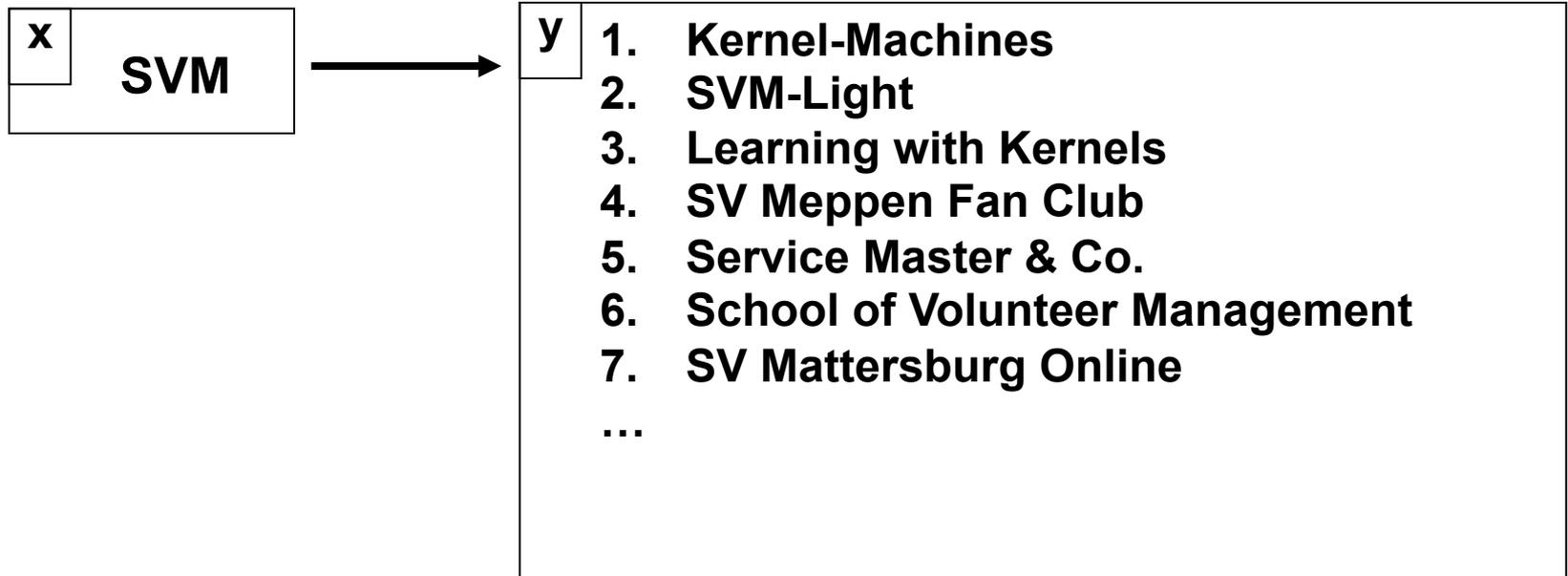
x The dog chased the cat



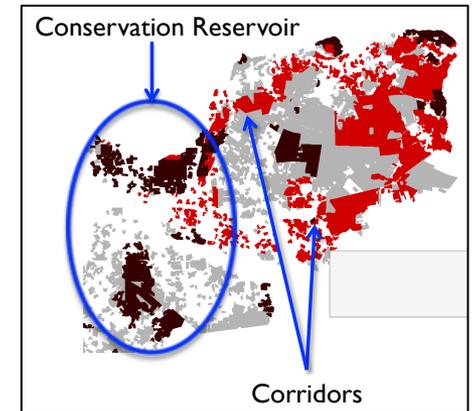
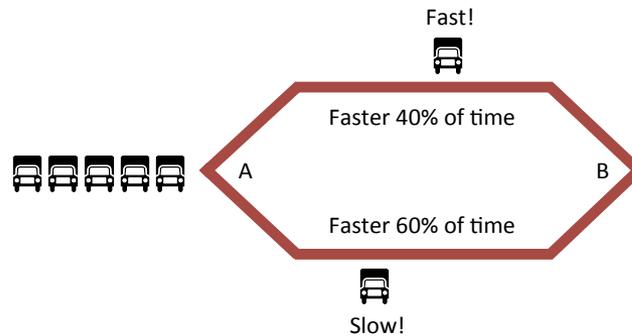
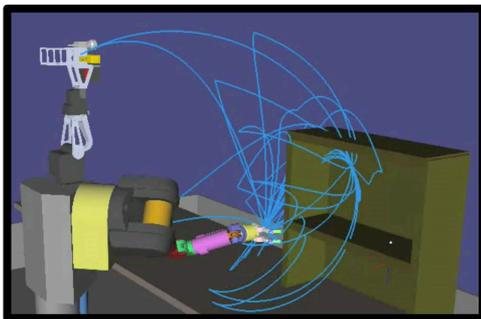
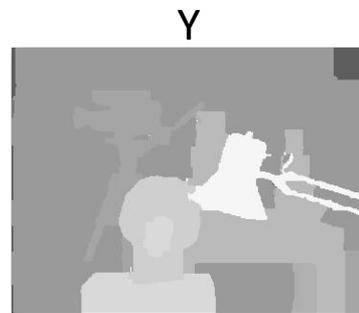
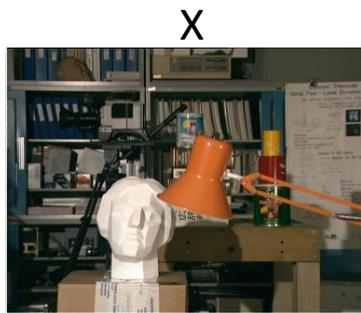
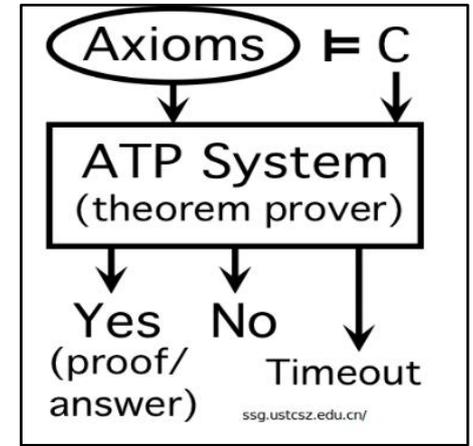
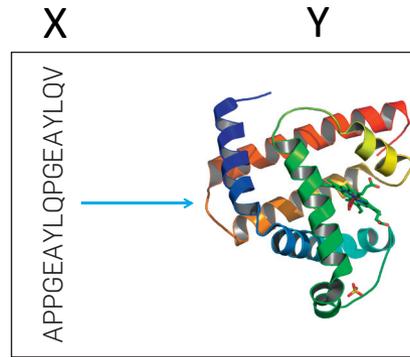
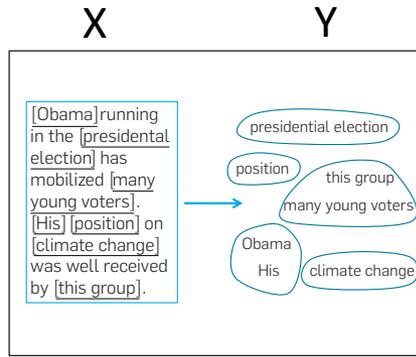
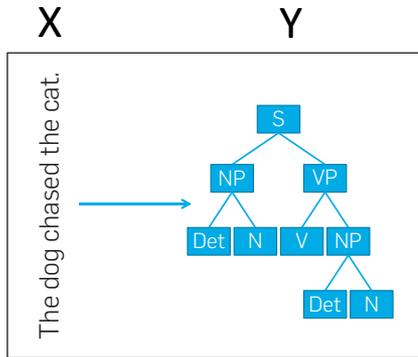
Examples of Complex Output Spaces

- Information Retrieval

- Given a query x , predict a ranking y .
- Dependencies between results (e.g. avoid redundant hits)
- Loss function over rankings (e.g. Average Precision)



Structured Prediction



General Formula (Linear Models)

- Assume scoring function F

$$h(\mathbf{x}; w) = \operatorname{argmax}_{y \in Y(\mathbf{x})} F(\mathbf{x}, y; w)$$

- Assume F is linear:

$$F(\mathbf{x}, y; w) = w^T \Psi(\mathbf{x}, y)$$

Example 1

$$h(\mathbf{x}; w) = \operatorname{argmax}_{y \in Y(\mathbf{x})} F(\mathbf{x}, y; w) \quad F(\mathbf{x}, y; w) = w^T \Psi(\mathbf{x}, y)$$

Binary Classification:

$$\Psi(\mathbf{x}, y) = y\mathbf{x}$$

$$Y(\mathbf{x}) = \{-1, +1\}$$

$$F(\mathbf{x}, y; w) = y(w^T \mathbf{x})$$

$$h(\mathbf{x}; w) = \operatorname{argmax}_{y \in \{-1, +1\}} y(w^T \mathbf{x})$$

Examples

$$h(\mathbf{x}; w) = \operatorname{argmax}_{\mathbf{y} \in Y(\mathbf{x})} F(\mathbf{x}, \mathbf{y}; w) \quad F(\mathbf{x}, \mathbf{y}; w) = w^T \Psi(\mathbf{x}, \mathbf{y})$$

1st Order Sequences:

$Y(\mathbf{x}) =$ all possible output sequences

$$\Psi(\mathbf{x}, \mathbf{y}) = \sum_j \phi(y^j, y^{j-1} \mid \mathbf{x})$$

$$F(\mathbf{x}, \mathbf{y}; w) = w^T \sum_j \phi(y^j, y^{j-1} \mid \mathbf{x})$$

Solve using Viterbi!

Examples

$$h(\mathbf{x}; w) = \operatorname{argmax}_{\mathbf{y} \in Y(\mathbf{x})} F(\mathbf{x}, \mathbf{y}; w)$$

$$F(\mathbf{x}, \mathbf{y}; w) = w^T \Psi(\mathbf{x}, \mathbf{y})$$

Integer Linear Program:

$Y(\mathbf{x}) =$ Feasible settings of \mathbf{y}

Each $y^j \in \{0, 1\}$

$$\Psi(\mathbf{x}, \mathbf{y}) = \sum_j y^j \phi^j(\mathbf{x})$$

$$F(\mathbf{x}, \mathbf{y}; w) = \mathbf{y}^T \mathbf{c} \quad \mathbf{c} = \begin{bmatrix} w^T \phi^1(\mathbf{x}) \\ w^T \phi^2(\mathbf{x}) \\ \vdots \end{bmatrix}$$

$$h(\mathbf{x}; w) = \operatorname{argmax}_{\mathbf{y} \in Y(\mathbf{x})} \mathbf{y}^T \mathbf{c}$$

Structured Prediction Learning Problem

- **Efficient Inference/Prediction**

$$h(\mathbf{x}; w) = \underset{y}{\operatorname{argmax}} w^T \Psi(y, \mathbf{x})$$

- Viterbi in sequence labeling
- CKY Parser for parse trees
- Sorting for ranking

- **Efficient Learning/Training**

- Learn parameters w from training data $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1..N}$
- Structural SVM: Hinge Loss Minimization
- Conditional Random Fields: Log Loss Minimization
- Structured Perceptron, etc...

Perceptron Learning Algorithm

- $w^1 = 0, b^1 = 0$
- For $t = 1 \dots$
 - Receive example (x, y)
 - If $h(x | w^t) = y$
 - $[w^{t+1}, b^{t+1}] = [w^t, b^t]$
 - Else
 - $w^{t+1} = w^t + yx$
 - $b^{t+1} = b^t + y$

$$h(x | w) = \text{sign}(w^T x - b)$$

Training Set:

$$S = \{(x_i, y_i)\}_{i=1}^N$$

$$y \in \{+1, -1\}$$

Go through training set
in arbitrary order
(e.g., randomly)



Structured Perceptron

- $w^1 = 0$

$$h(x | w) = \operatorname{argmax}_{y'} w^T \Psi(x, y')$$

- For $t = 1 \dots$

- Receive example (x, y)

- If $h(x | w^t) = y$

- $w^{t+1} = w^t$

- Else

- $w^{t+1} = w^t + \Psi(x, y)$

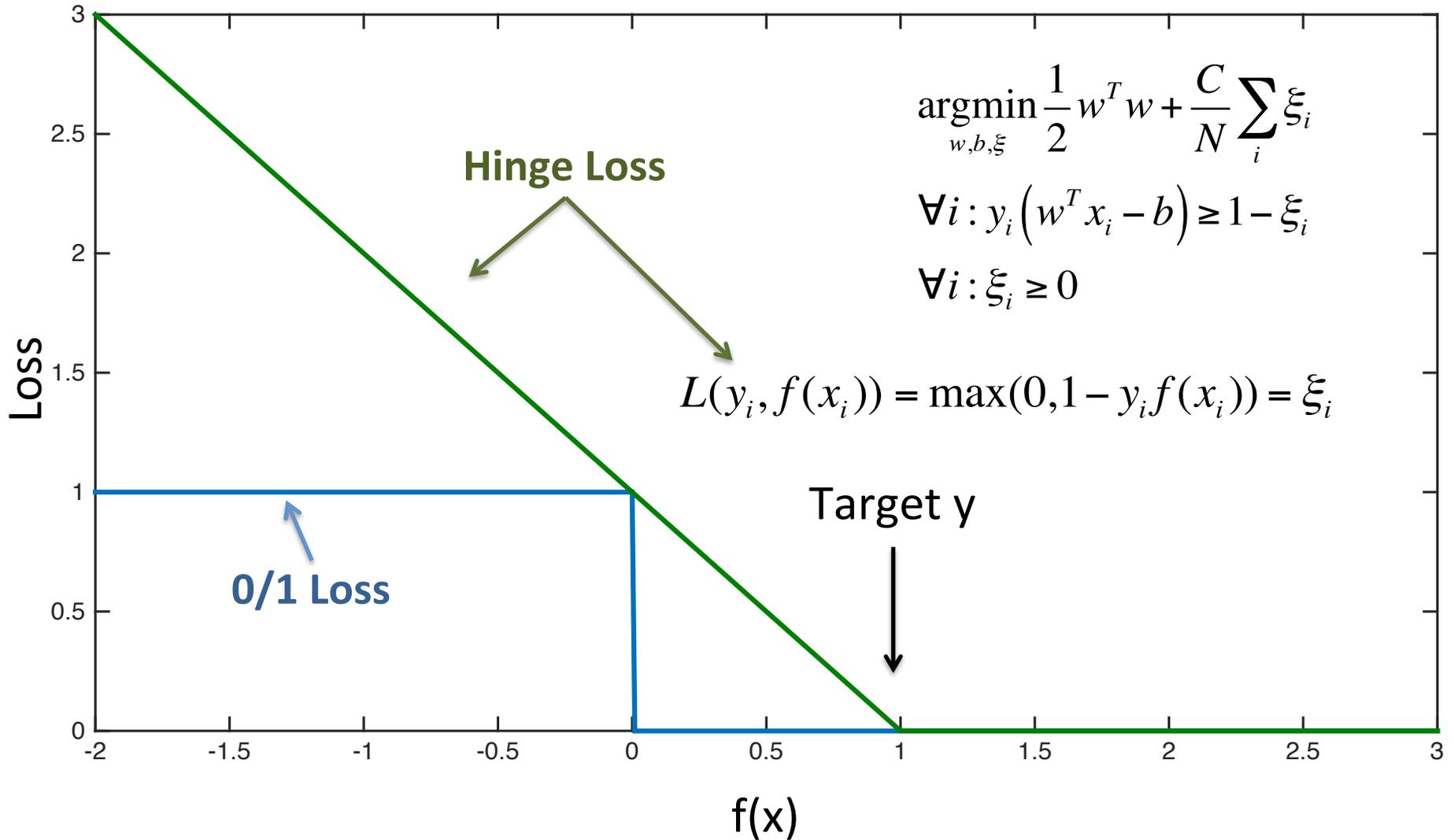
Training Set:

$$S = \{(x_i, y_i)\}_{i=1}^N$$

Go through training set
in arbitrary order
(e.g., randomly)



Conventional SVMs



Structural SVM

- Let \mathbf{x} denote a structured input (sentence)
- Let \mathbf{y} denote a structured output (POS tags)
- Standard objective function: $\frac{1}{2} w^2 + \frac{C}{N} \sum_i \xi_i$
- Constraints are defined for each incorrect labeling \mathbf{y}' over each \mathbf{x} .

$$\forall i, \forall \mathbf{y}' \neq \mathbf{y}^{(i)} : \underbrace{w^T \Psi(\mathbf{y}^{(i)}, \mathbf{x}^{(i)})}_{\text{Score}(\mathbf{y}^{(i)})} \geq \underbrace{w^T \Psi(\mathbf{y}', \mathbf{x}^{(i)})}_{\text{Score}(\mathbf{y}')} + \underbrace{\Delta_i(\mathbf{y}')}_{\text{Loss}(\mathbf{y}')} - \underbrace{\xi_i}_{\text{Slack}}$$

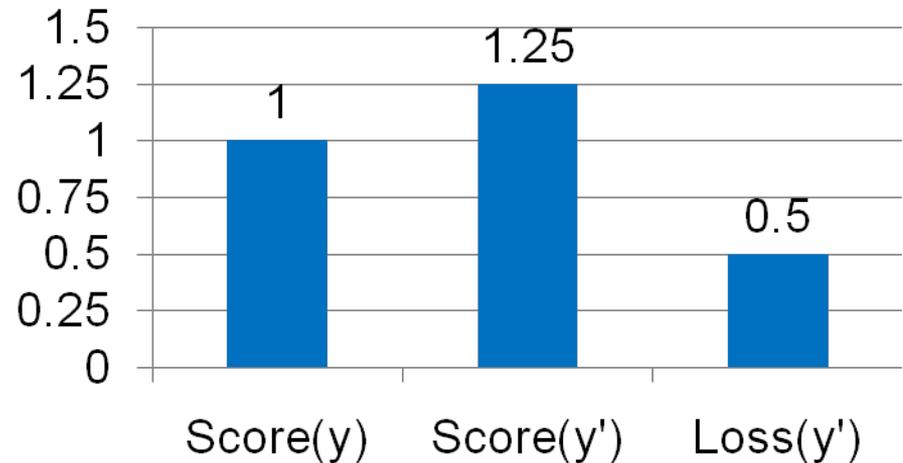
Interpreting Constraints

$$\frac{1}{2} w^2 + \frac{C}{N} \sum_i \xi_i$$

$$\forall i, \forall \mathbf{y}' \neq \mathbf{y}^{(i)} : \underbrace{w^T \Psi(\mathbf{y}^{(i)}, \mathbf{x}^{(i)})}_{\text{Score}(\mathbf{y}^{(i)})} \geq \underbrace{w^T \Psi(\mathbf{y}', \mathbf{x}^{(i)})}_{\text{Score}(\mathbf{y}')} + \underbrace{\Delta_i(\mathbf{y}')}_{\text{Loss}(\mathbf{y}')} - \underbrace{\xi_i}_{\text{Slack}}$$

Suppose for incorrect \mathbf{y}' :

Then: $\xi_i \geq 0.75 \geq \Delta(\mathbf{y}')$



Sample Research Questions

- Scale
 - Predicting over millions of variables
- Structured Representation Learning
 - Deep learning for structured outputs?
- Cost of labeling

Crowdsourcing

Acquiring Labels from Annotators

Keyword Tagging Attractions in Paris!

- Please inspect the attraction below.
- **SELECT ALL** keywords that are appropriate for this attraction.
- Selected keywords will turn **RED**.
- The right pane below displays additional information (e.g., wikipedia page) for your convenience.



Place de la Madeleine

- | | |
|---------------------|-----------------------|
| • Ancient Ruin | • Palace / Mansion |
| • Architecture | • Performance |
| • Art | • Plaza / Open Area |
| • Bridge | • Recreational |
| • Cabaret | • Relaxing / Leisure |
| • Cemetary | • Religious |
| • Comedy | • Scenic -- Nature |
| • Culture | • Scenic -- Urban |
| • Dining | • Scenic -- Water |
| • Fountain | • Shopping |
| • Garden / Park | • Sightseeing |
| • Historical | • Spa / Massage |
| • Large Building | • Sports |
| • Memorial | • Street |
| • Monument / Statue | • Theater / Opera |
| • Museum -- Art | • Tour |
| • Museum -- Other | • Transportation |
| • Nightlife | • Walking / Strolling |
| • Outdoors | • Zoo / Aquarium |

Submit

Search Wikipedia

La Madeleine, Paris

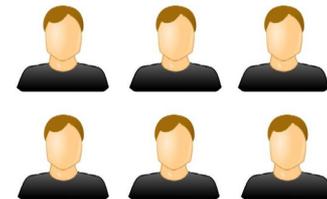


The Madeleine church

L'**église de la Madeleine** (French pronunciation: [ɛgliz də la madɛlɛn], *Madleine Church*; more formally, **L'église Sainte-Marie-Madeleine**; less formally, just **La Madeleine**) is a Roman Catholic church occupying a commanding position in the 8th arrondissement of Paris.

The Madeleine Church was designed in its present form as a temple to the glory of **Napoleon's army**. To its south lies the **Place de la Concorde**, to the east is the

amazon[®]
mechanical turk
beta



How Reliable are Annotators?

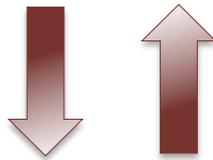
- If we knew what the labels were
 - Can judge workers on label quality
- If we knew who the good workers were
 - Can create labels from their annotations
- **Chicken and egg problem!**

Worker Reliability as Latent Variable

- Let z_m denote the reliability of worker m

Estimated label


$$y_i = \frac{1}{\sum_m z_m} \sum_m y_{im} z_m$$

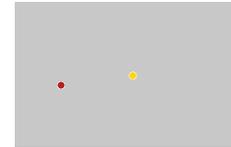


$$z_m = \frac{1}{N} \sum_i L(y_i, y_{im})$$

Differing Ambiguities Across Tasks

- Often collecting annotations for many tasks
- Some tasks are harder than others
- How many labels to collect for each task?

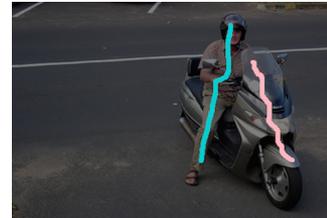
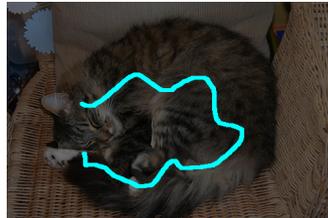
Structured Annotations



Full
supervision

Image-level
supervision

Point-level
supervision

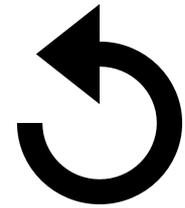


<http://arxiv.org/pdf/1506.02106v4.pdf>

Active Learning

Crowdsourcing

Unlabeled

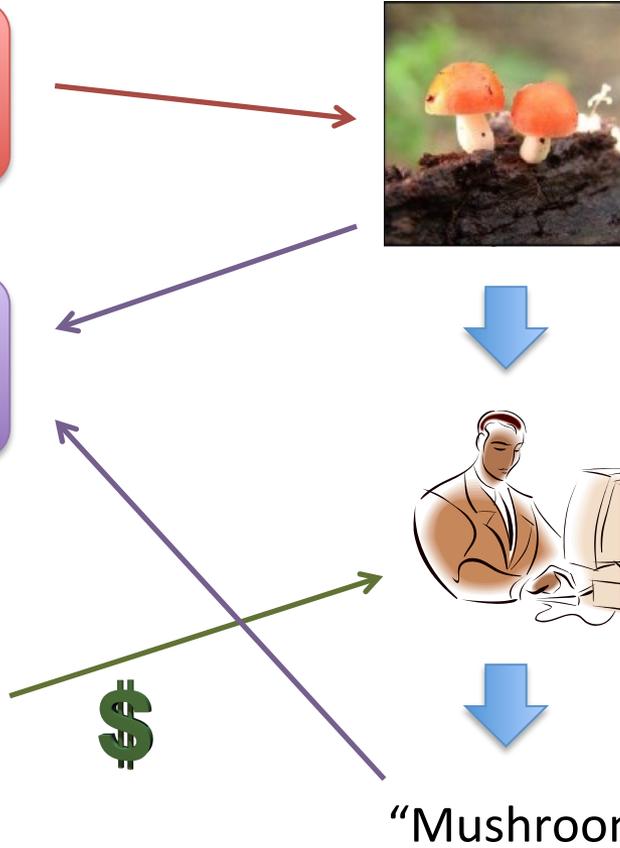


Repeat

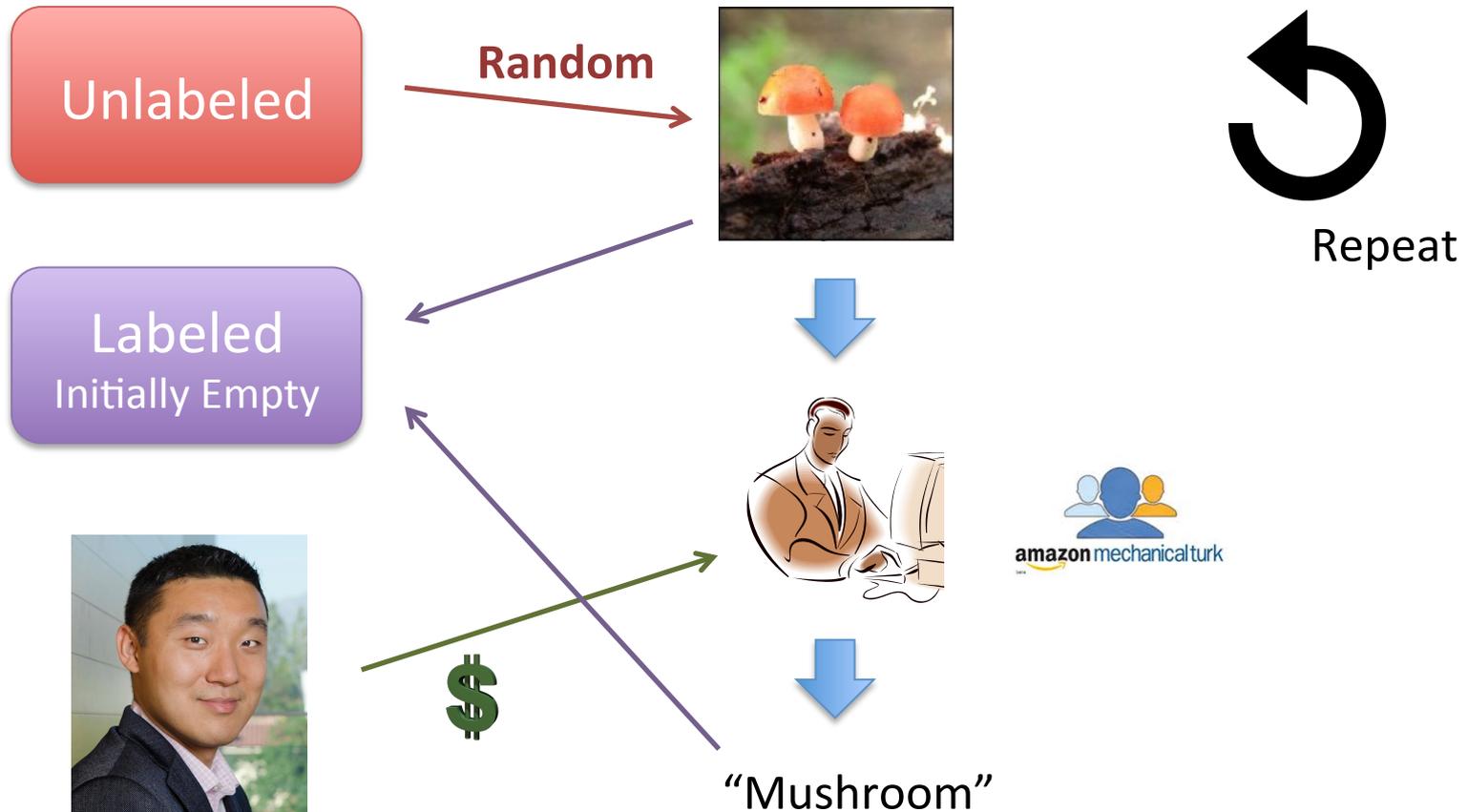
Labeled
Initially Empty



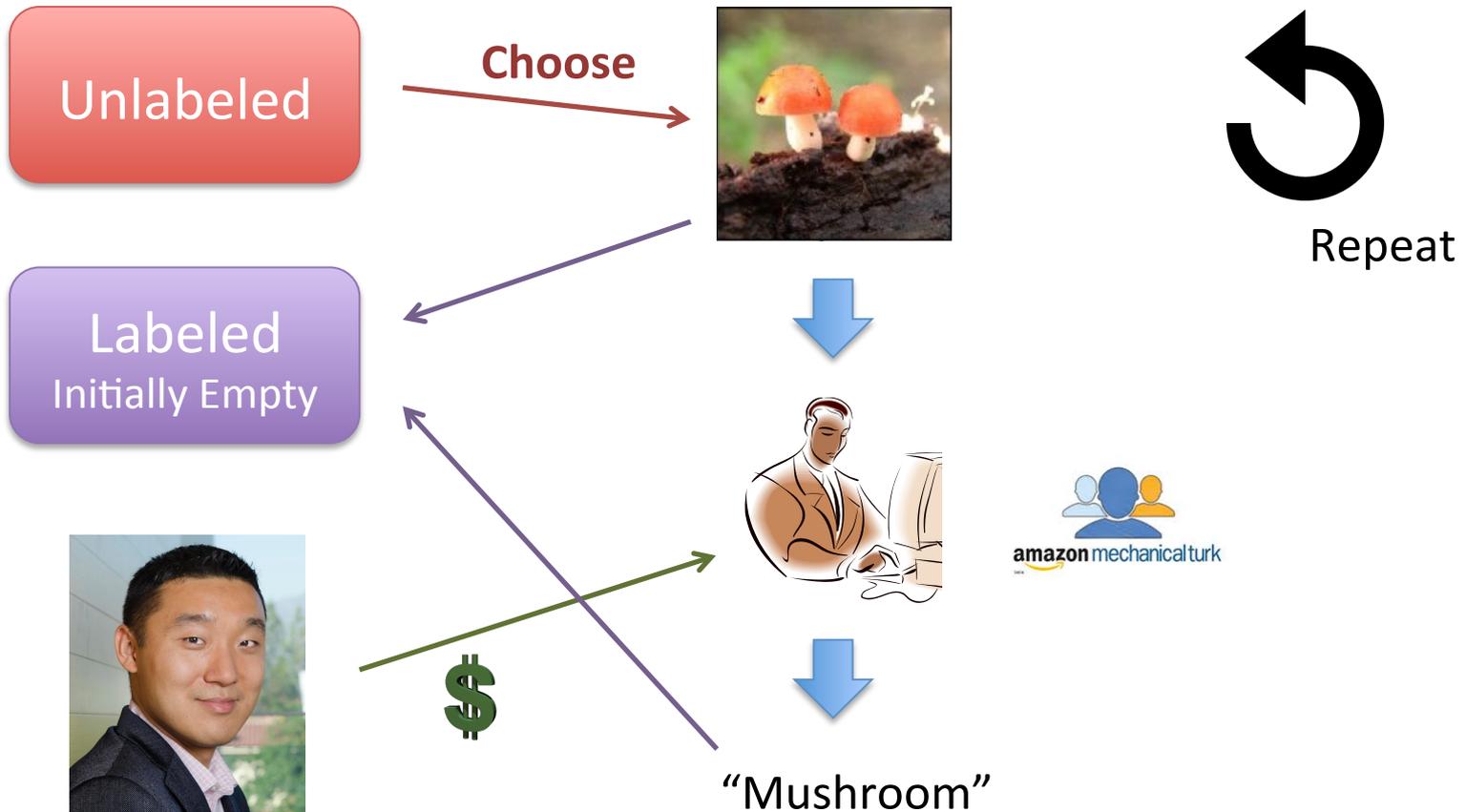
“Mushroom”



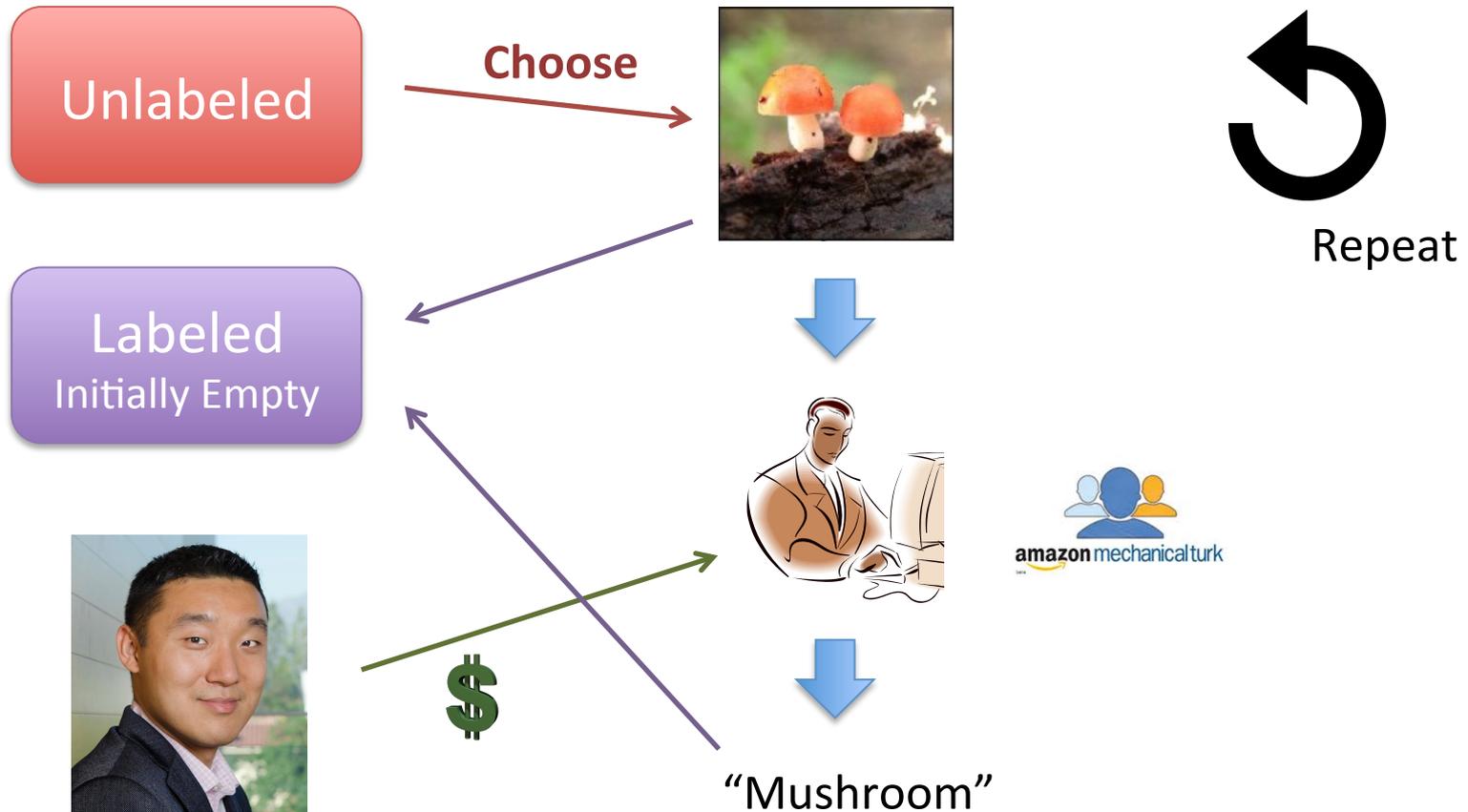
Passive Learning



Active Learning



Goal: Maximize Accuracy with Minimal Cost

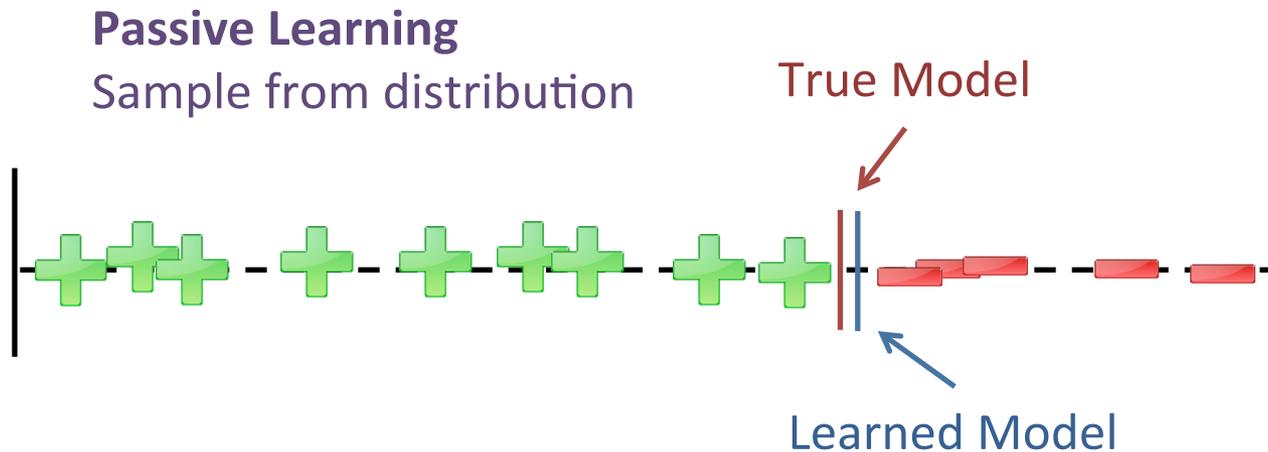


Comparison with Passive Learning

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

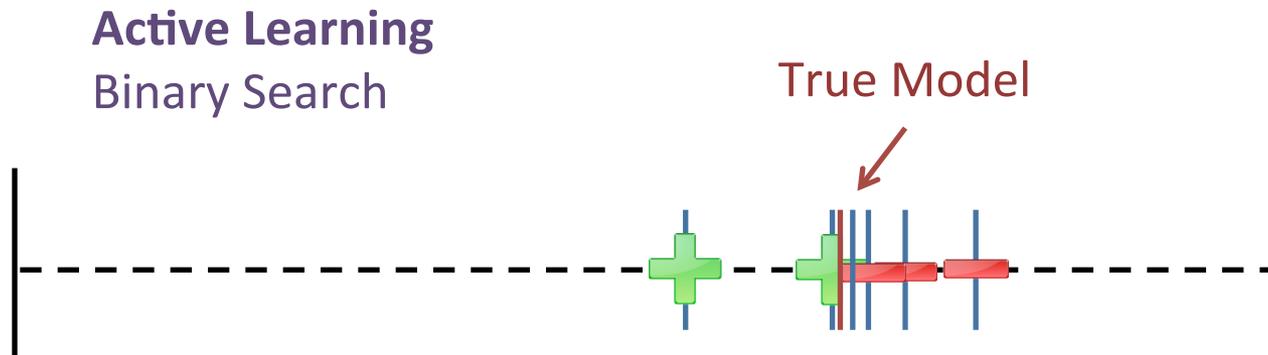
Simple Example

- 1 feature
- Learn threshold function



Simple Example

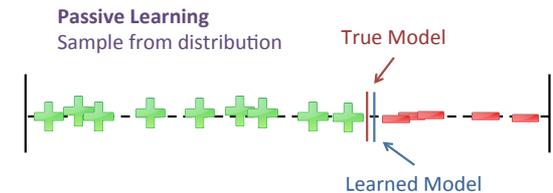
- 1 feature
- Learn threshold function



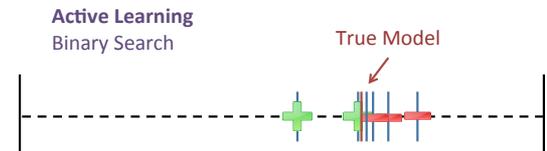
Comparison with Passive Learning

- # samples to be within ε of true model

- Passive Learning: $O\left(\frac{1}{\varepsilon}\right)$



- Active Learning: $O\left(\log \frac{1}{\varepsilon}\right)$

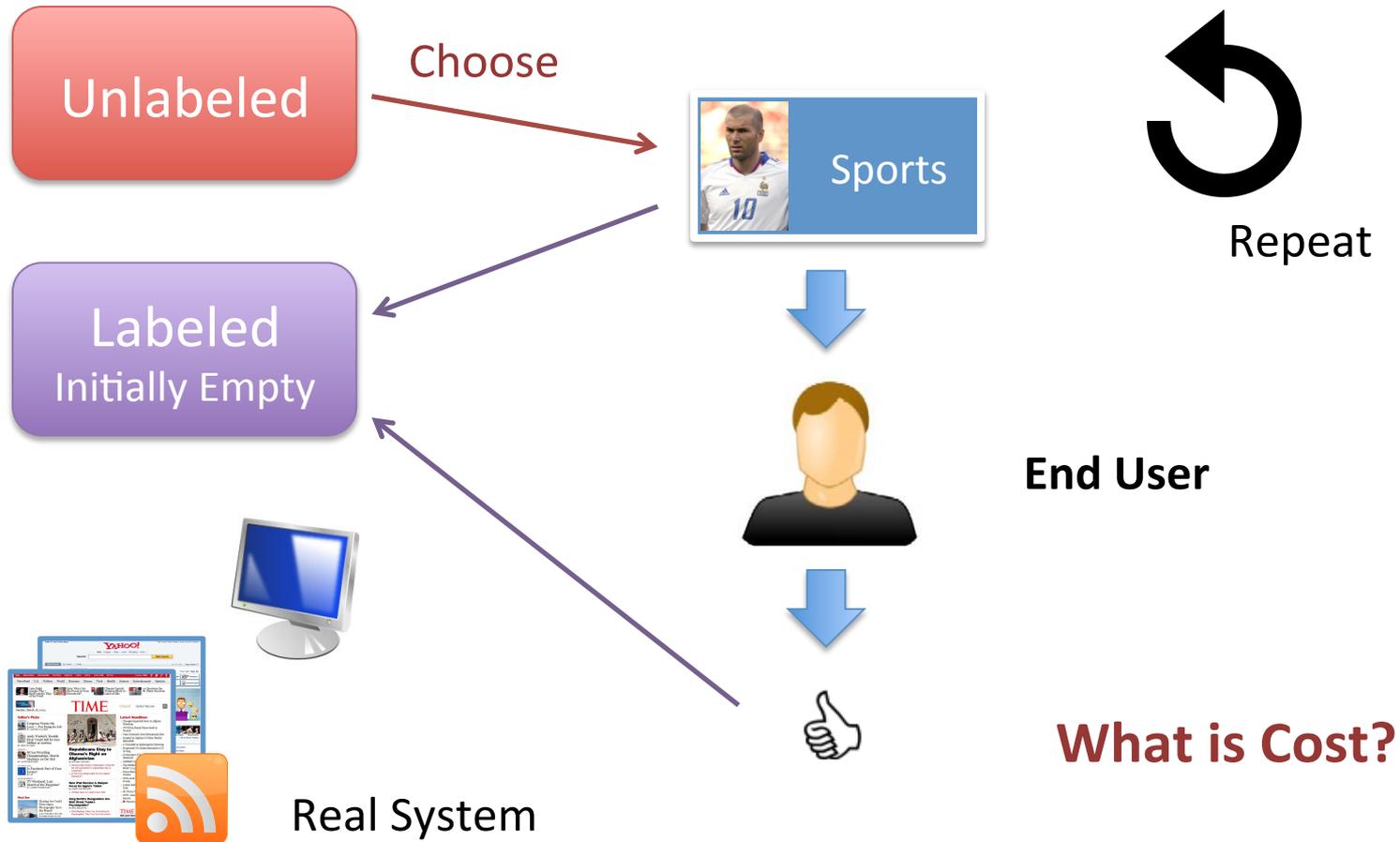


Multi-Armed Bandits

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



Formal Definition

- K actions/classes
- Each action has an average reward: μ_k
 - 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 $\mu_{a(t)}$

Basic Setting
K classes
No features

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	0	0	0	1	0



Average Likes

Shown

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

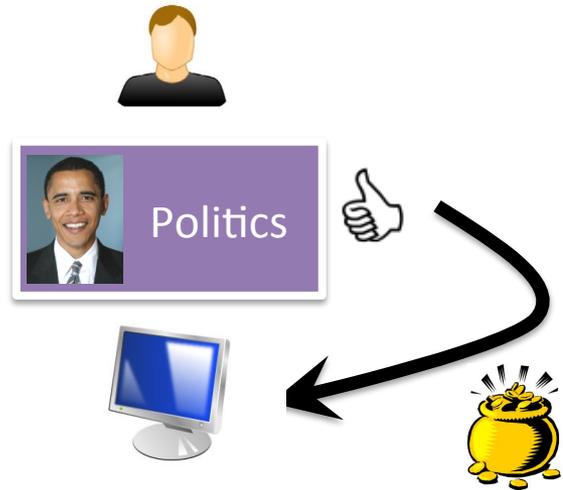
Shown

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



Interactive Personalization

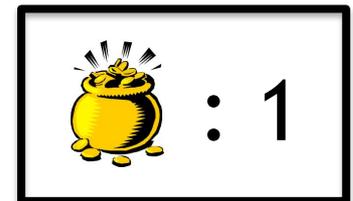
(5 Classes, No features)



Average Likes

Shown

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

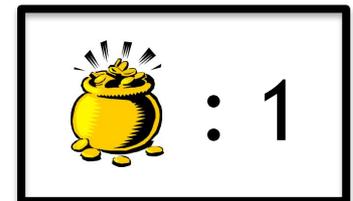


Interactive Personalization

(5 Classes, No features)

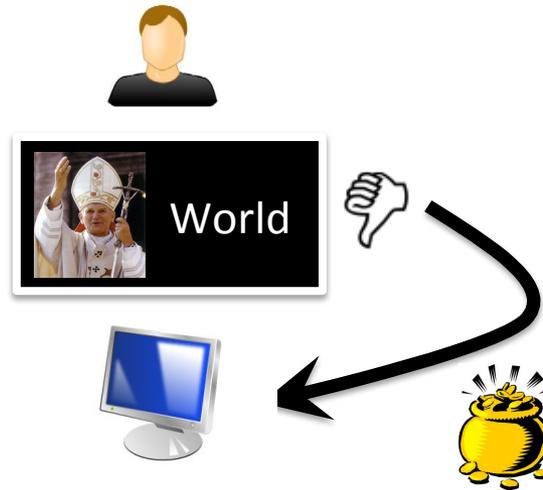


					
Average Likes	--	--	1	0	--
# Shown	0	0	1	1	1



Interactive Personalization

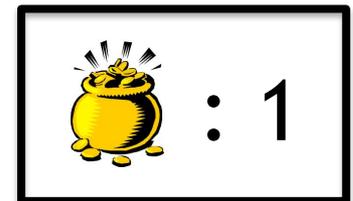
(5 Classes, No features)



Average Likes

Shown

				
--	--	1	0	0
0	0	1	1	1

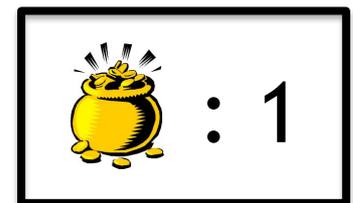


Interactive Personalization

(5 Classes, No features)



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

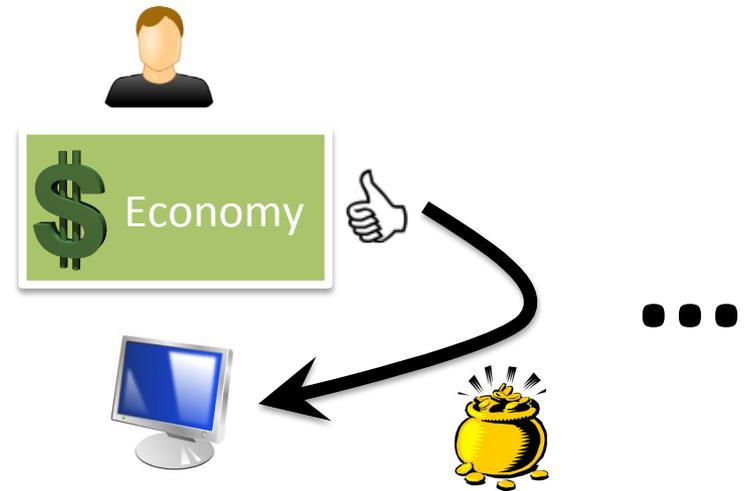


Average Likes

Shown

Interactive Personalization

(5 Classes, No features)



Average Likes

Shown

				
--	1	1	0	0
0	1	1	1	1

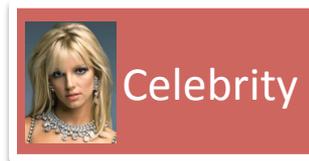


What should Algorithm Recommend?

Exploit:



Explore:



Best:



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

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



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

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

Time Horizon

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

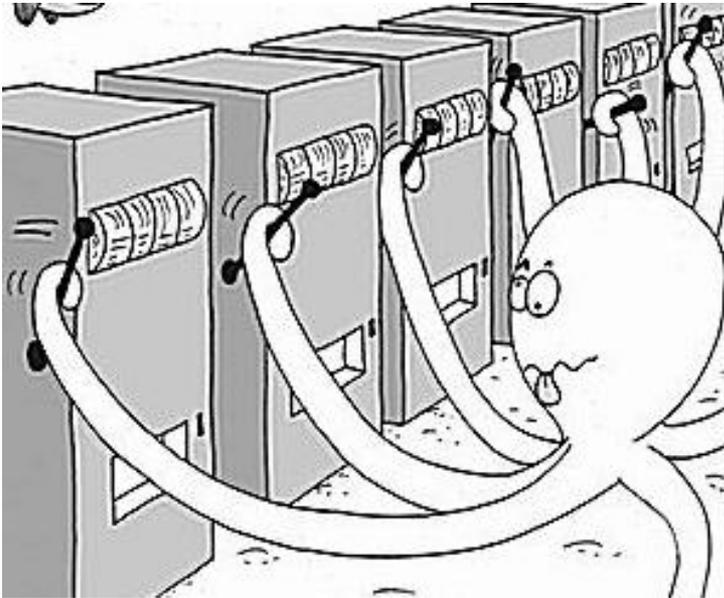
- Opportunity cost of not knowing preferences
- “no-regret” if $R(T)/T \rightarrow 0$
 - Efficiency measured by convergence rate

Recap: The Multi-Armed Bandit Problem

- K actions/classes
 - Each action has an average reward: μ_k
 - All unknown to us
 - Assume WLOG that μ_1 is largest
 - For $t = 1 \dots T$
 - Algorithm chooses action $a(t)$
 - Receives random reward $y(t)$
 - Expectation $\mu_{a(t)}$
 - Goal: minimize $T\mu_1 - (\mu_{a(1)} + \mu_{a(2)} + \dots + \mu_{a(T)})$
- 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

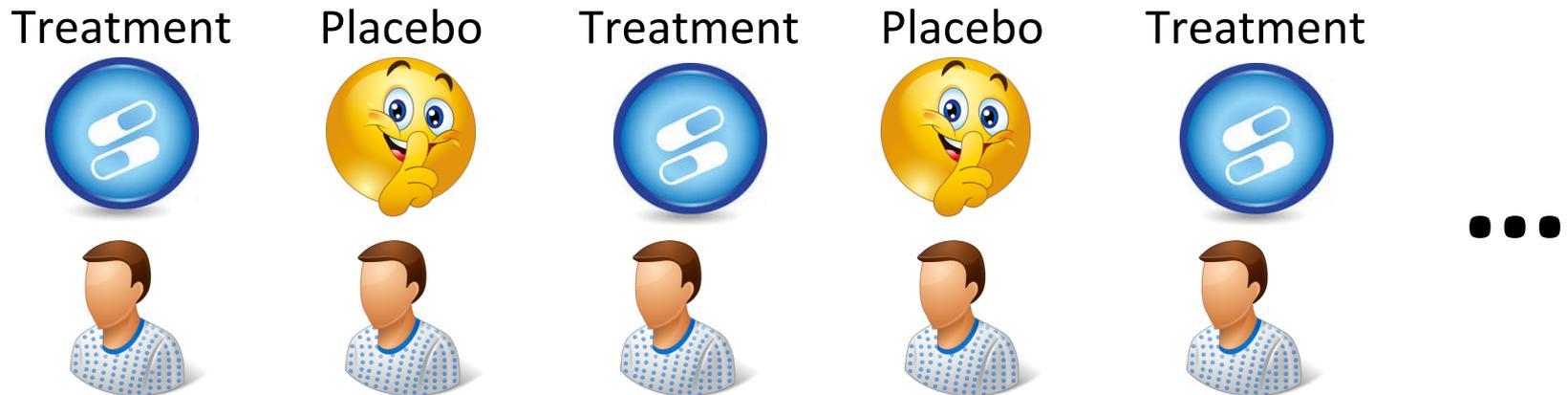
Implications of Regret

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

- 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

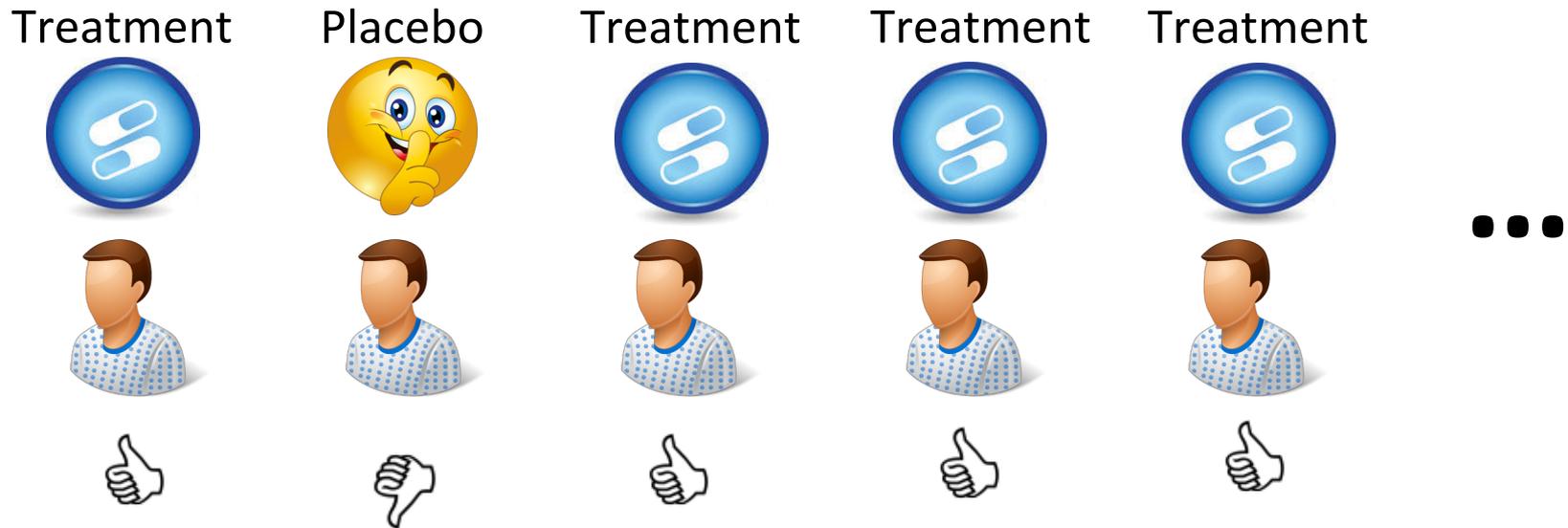
Experimental Design

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



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

Official Apple Store® ⓘ
Ad 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

Buy MacBook Pro	Special Financing Offer
Buy MacBook Air	Free In-Store Pickup

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.

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

Largest Use-Case
of Multi-Armed
Bandit Problems

Reinforcement Learning

Actions Impact State

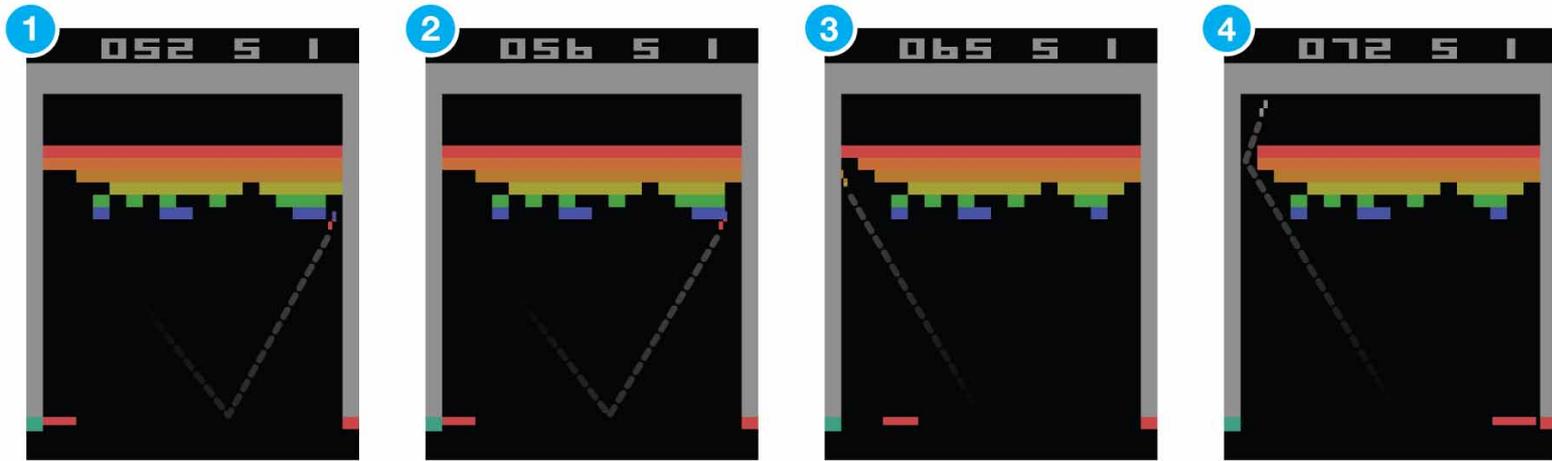
- In MAB:
 - Actions do not impact state
 - Constant reward function
- Reinforcement Learning
 - Actions effect state you're in
 - Reward function depends on state

Video Demo

(Deep Reinforcement Learning for Atari)

<https://www.youtube.com/watch?v=iqXKQf2BOSE>

What is State?



Reward of each action varies depending on state!

Action at current state impacts future states!

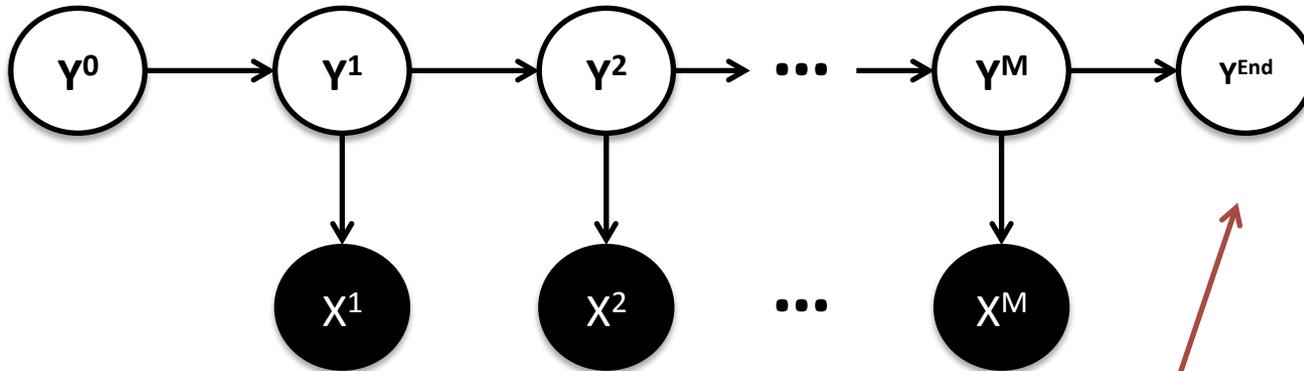
Much harder to do exploration!

Non-Convex Optimization



Anima
Anandkumar

Recall: Hidden Markov Models



Optional

$$P(x, y) = P(\text{End} | y^M) \prod_{i=1}^M P(y^i | y^{i-1}) \prod_{i=1}^M P(x^i | y^i)$$

Recall: EM Algorithm for HMMs

- If we had y 's \rightarrow max likelihood.
- If we had (A,O) \rightarrow predict y 's

Chicken vs Egg!

1. Initialize A and O arbitrarily

Expectation Step

2. Predict prob. of y 's for each training x

3. Use y 's to estimate new (A,O)

Maximization Step

4. Repeat back to Step 1 until convergence

Non-Convex Optimization Problem! Converges to local optimum.

- If we had y 's \rightarrow max likelihood.
- If we had (A,O) \rightarrow predict y 's

Chicken vs Egg!

1. Initialize A and O arbitrarily

Expectation Step

2

3

4

Can We Train HMMs Optimally?

Inspiration from Dimensionality Reduction

- Find best rank K approximation to Y:

$$\operatorname{argmin}_{U \in \mathbb{R}^{N \times K}, V \in \mathbb{R}^{M \times K}} \|Y - UV^T\|_2^2$$

- Non-convex optimization problem!
 - Due to non-convex feasible region
- **But optimally solved via SVD!**

Spectral Learning of HMMs

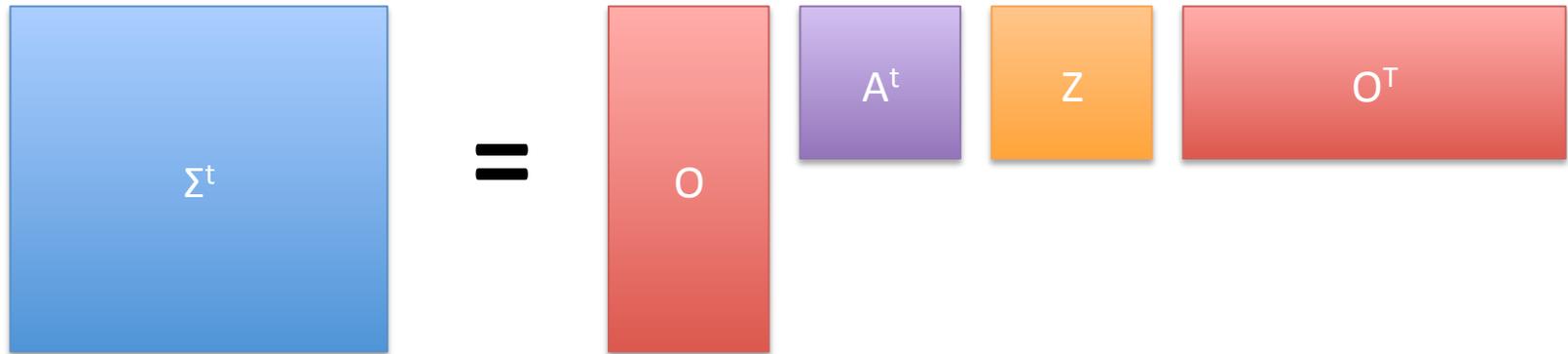
**Want to
Estimate:**

$$P(y^j | y^{j-1}) = A \quad P(x^j | y^j) = O$$

$$\begin{aligned} \Sigma^t &= E \left[x^{j+t} (x^j)^T \right] = E \left[E \left[x^{j+t} (x^j)^T \mid y^j \right] \right] \\ &= E \left[E \left[x^{j+t} \mid y^j \right] E \left[(x^j)^T \mid y^j \right] \right] \\ &= E \left[(OA^t k y^j) (O y^j)^T \right] \\ &= OA^t E \left[y^j (y^j)^T \right] O^T \\ &= OA^t Z O^T \end{aligned}$$

Treat each x^j and y^j
as indicator vector

Spectral Learning of HMMs



Rank-K SVD of Σ^1

Optimal Solution: $A = U^T \Sigma^2 \left(U^T \Sigma^1 \right)^{-1}$

(requires a lot of data)

...and many more topics!

- Probabilistic Models & Bayesian Reasoning
- Representation Learning
 - Deep learning is the most visible example
- Causal Reasoning
- ML + Game Theory
- ML + Systems
 - Large Scale Machine Learning
- Etc ...

CS 159

- Special Topics in Machine Learning
 - Taught Every Spring Term
 - Topics Rotate
- **Next Term:**
 - “Structured Prediction”
- Paper Reading & Presenting + Final Project
 - Graded on participation and final project



Taehwan Kim