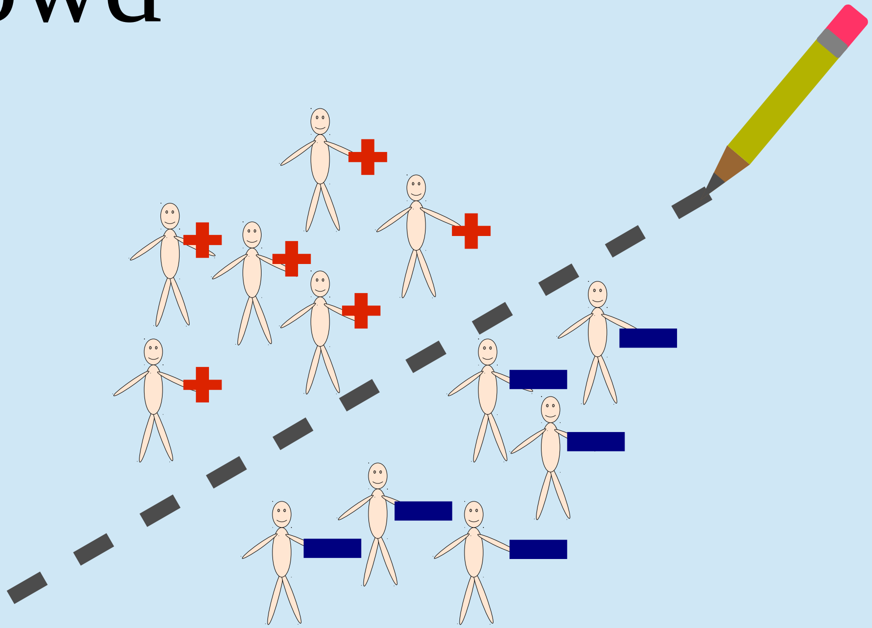
Toward Buying Labels From the Crowd

Jacob Abernethy Michigan
Yiling Chen Harvard
Chien-Ju Ho UCLA/Harvard
Bo Waggoner Harvard

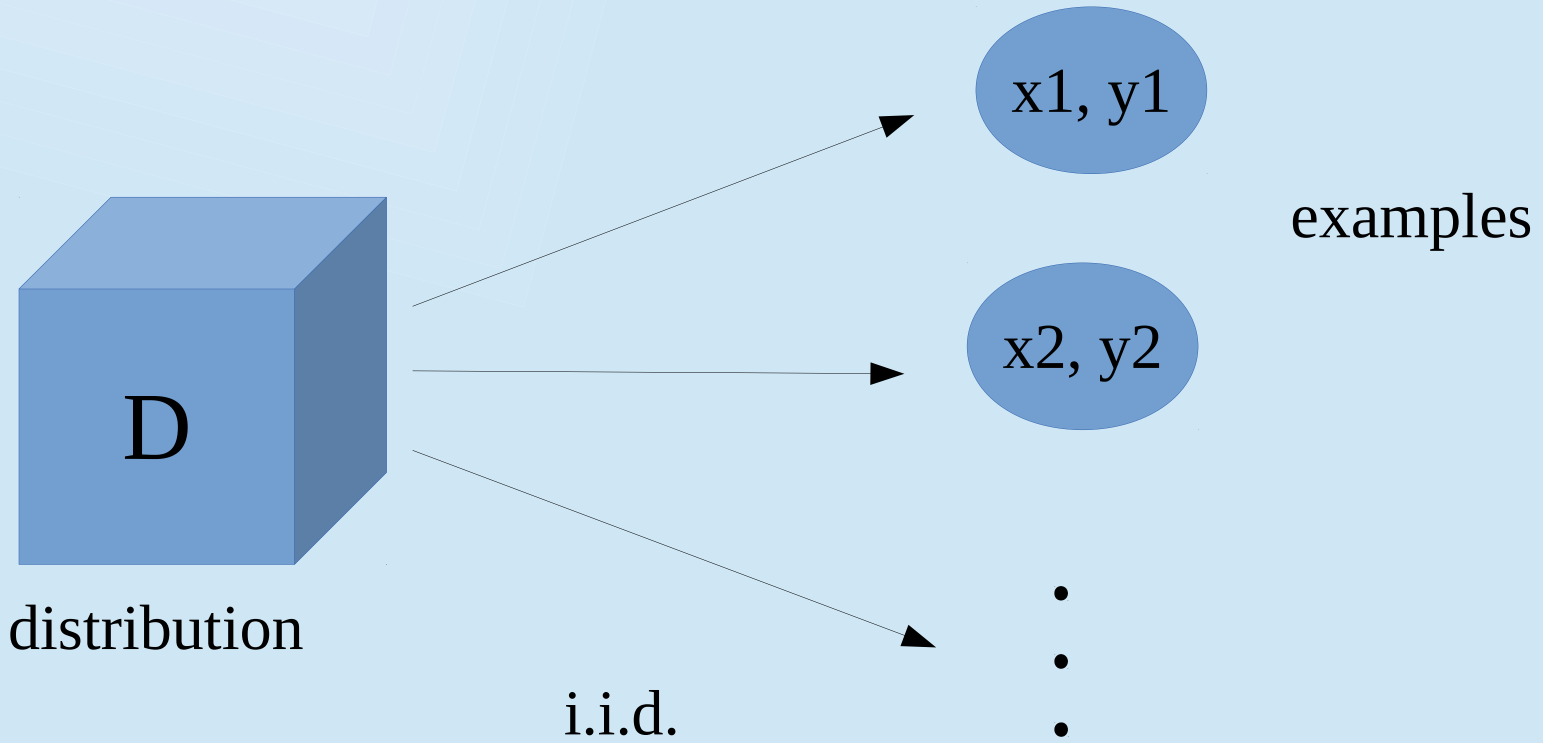


Indo-US Lectures Week in Machine Learning, Game Theory and Optimization
9th January 2014

Outline

- General setting
- Related work
- Our approach

Learning Setting



Learning Setting

$x1 \xrightarrow[\text{hypothesis}]{} h(x1)$

$h(x1), y1 \xrightarrow[\text{loss function}]{} \text{Loss}(h(x1), y1)$

examples

x1, y1

x2, y2

⋮

Learning Setting

x1 $\longmapsto$ h(x1)

hypothesis

h(x1), y1 $\longmapsto$ Loss(h(x1), y1)

loss function

x1, y1

x2, y2

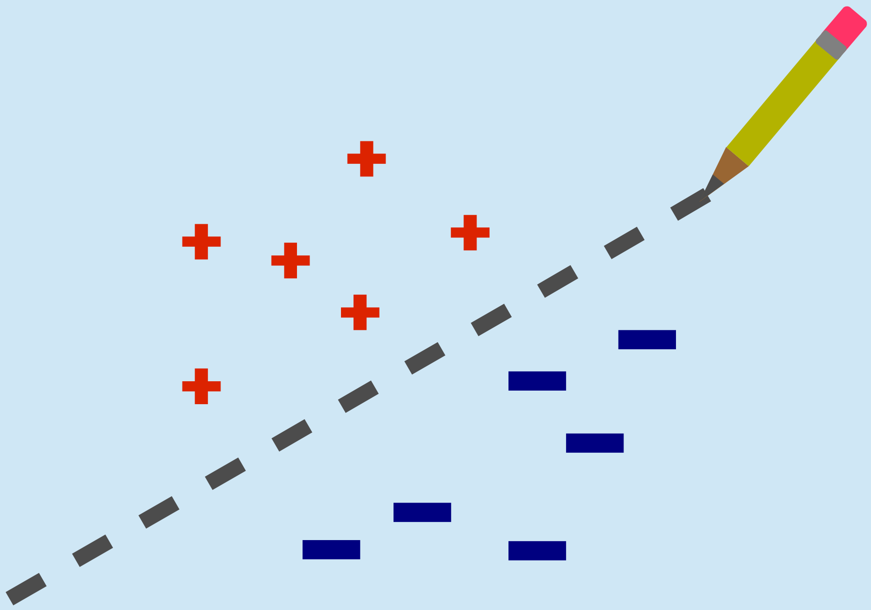
examples

•
•
•

Goal: from **few** examples,
pick a hypothesis with **small loss**
(in expectation, with high probability) w.r.t. D

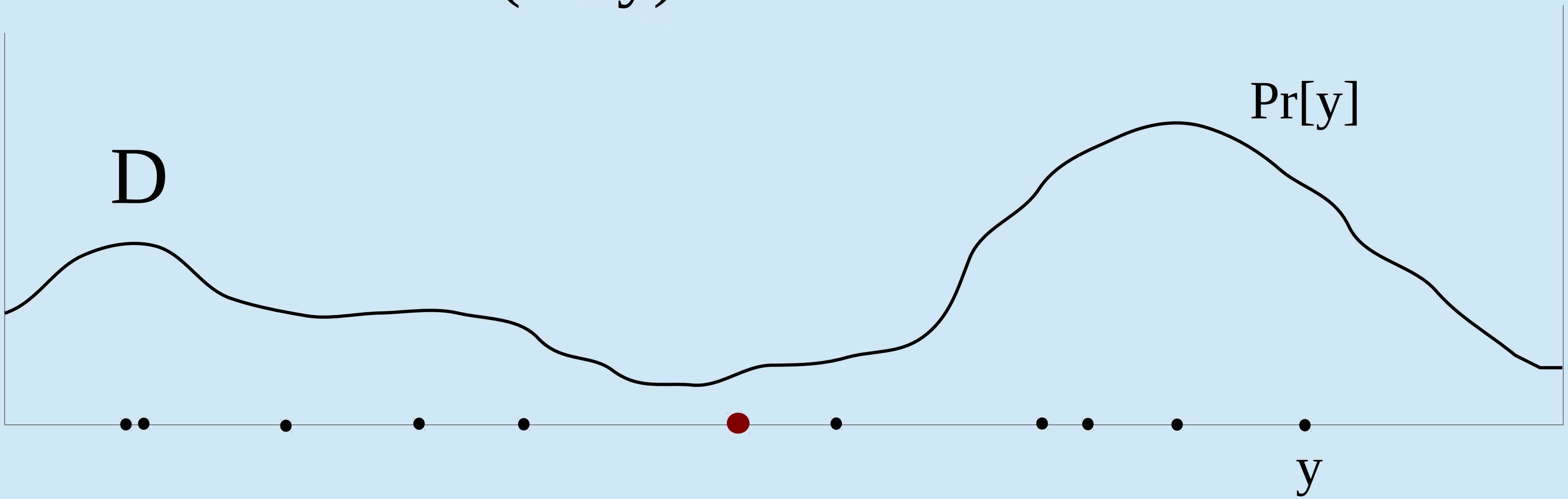
Example 1: Classification

x	= point in the plane
y	= "+" or "-"
hypothesis	= line
loss	= 0 if correct, 1 if incorrect or in [-1, 1] weighted by distance

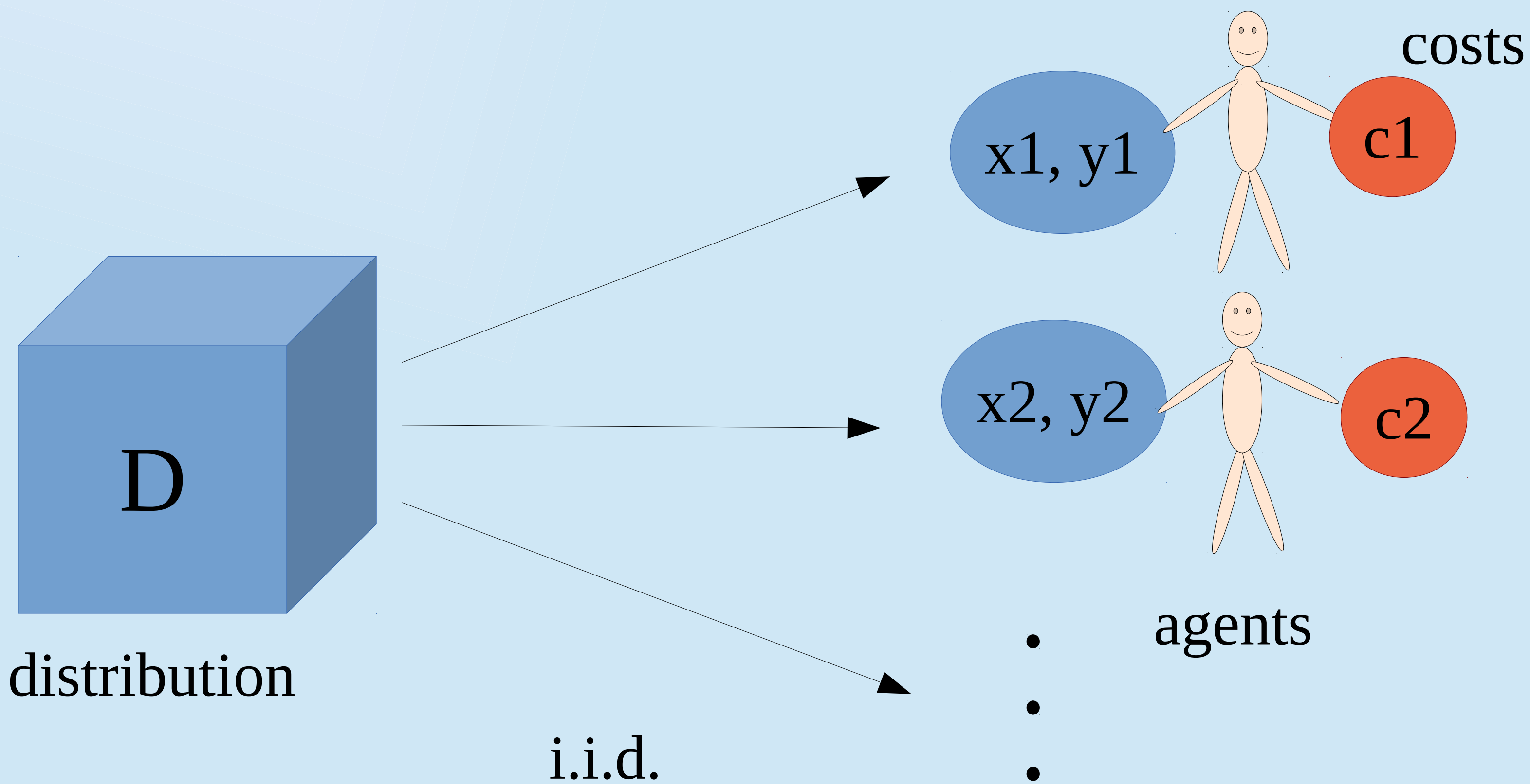


Example 2: Estimate the mean

x	= doesn't matter (e.g. always zero)
y	= real number in [0,1]
hypothesis	= real number in [0,1]
loss	= $(h - y)^2$

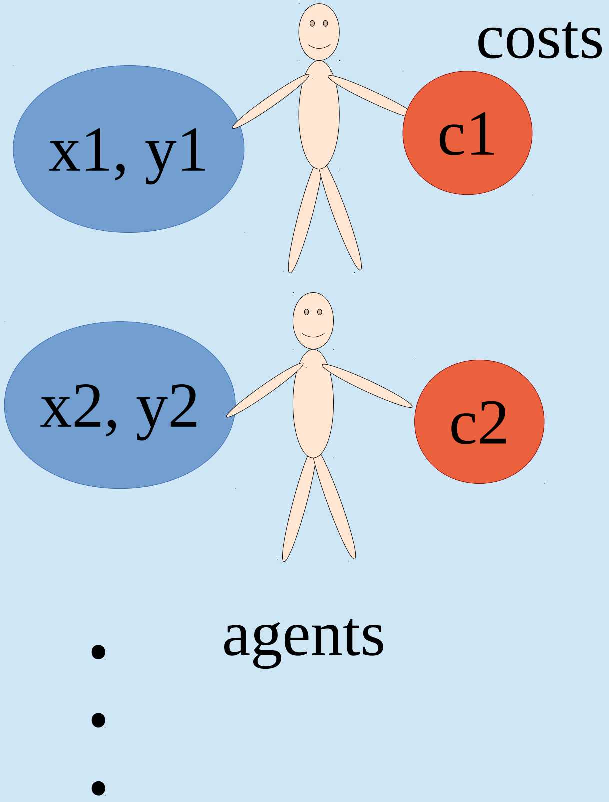


Adding incentives

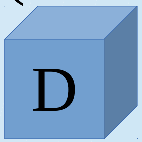


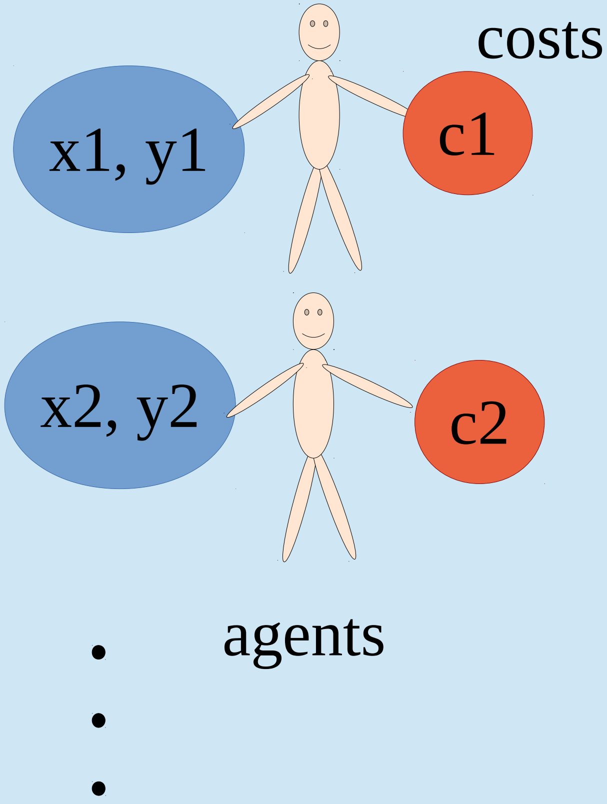
Incentives Setting

- (x1, y1, c1) drawn from D
- Must design **mechanism** and **learning algorithm** together
- Many possible assumptions:
  - costs in [0,1]
  - agents cannot misreport (x,y)
  - …



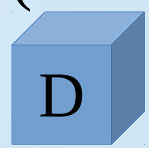
Goal

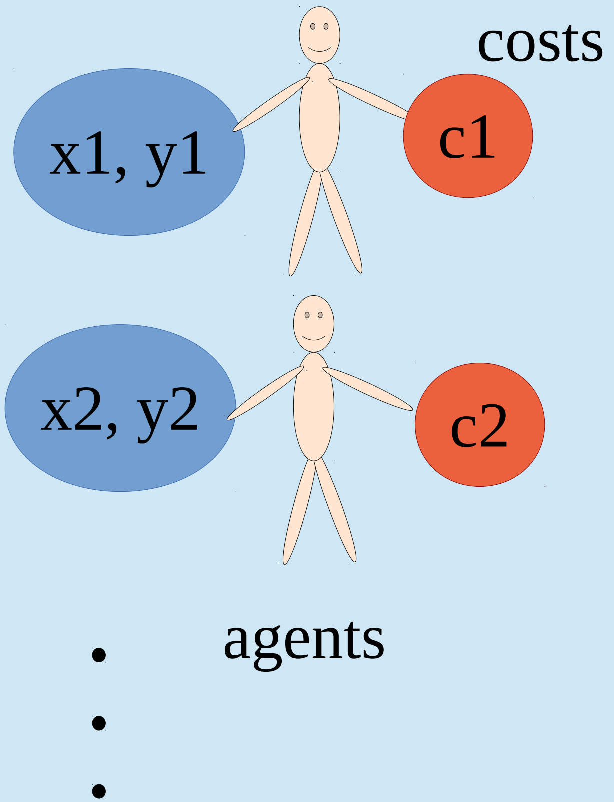
Goal: with **small budget**, purchase data and pick a hypothesis with **small loss** (in expectation, with high probability) w.r.t. 



Goal

Goal: with **small budget**, purchase data and pick a hypothesis with **small loss** (in expectation, with high probability) w.r.t.





Naive approach:
Offer B of the agents
a price of 1 (maximum).

→ Seems non-obvious how to improve on this!

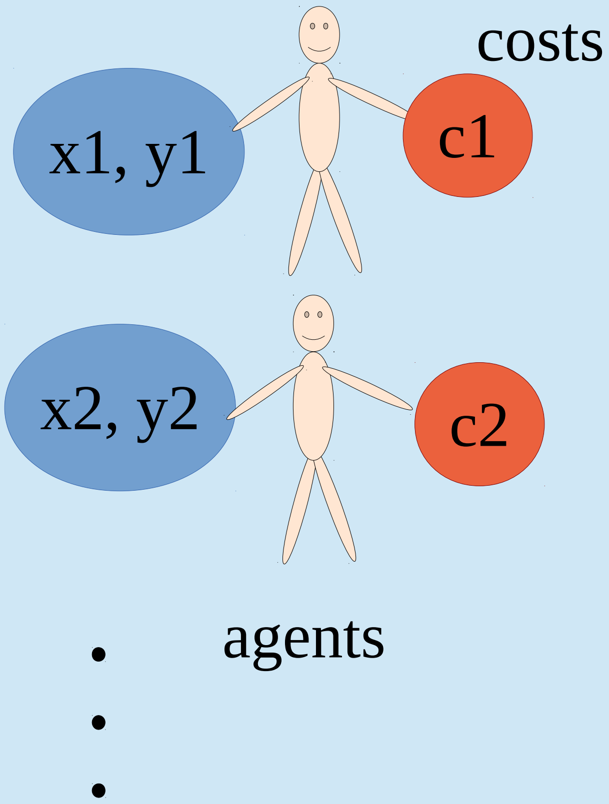
Three possible avenues

1. **Centralized/simultaneous:**
   auction of some sort.

2. **Decentralized/simultaneous:**
   survey offered to all agents.

→ Both miss interactions in the data!

3. **Iterative** (but perhaps myopic).



Digression: Importance Weighting

Goal: compute sum of y1, y2, …, yn.

Twist: each yi is observed independently with probability pi.

So: estimate sum $= \dfrac{y1}{p1} + \qquad \dfrac{y3}{p3} + \dfrac{y4}{p4} + \qquad \ldots.$

Digression: Importance Weighting

Goal: compute sum of y1, y2, …, yn.

Twist: each yi is observed independently with probability pi.

So: estimate sum = $\frac{y1}{p1} + \frac{y3}{p3} + \frac{y4}{p4} + \ldots$.

Can apply *Hoeffding*: Given independent Y1, …, Yn, with Yi in [0, bi]:
Let d = Pr[$|\Sigma_i$ Yi - expectation| > eps],
Then d < 2exp[-2 $eps^2 / \Sigma_i bi^2$].

Or, if I want probability 1-d, then I get error eps < $\sqrt{\frac{\ln(2/d)\ \Sigma_i bi^2}{2}}$

Outline

- General setting
- Related work
- Our approach

Conducting Truthful Surveys, Cheaply

Roth and Schoenebeck, EC 2011.

Problem: Estimate the mean.
Assumptions:
 - marginal on costs, F, is known.
 - **decentralized/simultaneous (*survey*) approach**.
Goal: unbiased estimator with minimum (or close to minimum) *worst-case expected variance*.

(*worst-case:* over all distributions D whose cost marginal is F.)
(*expected:* over the data points drawn from D.)
(*variance:* over the randomization of the mechanism.)

Conducting Truthful Surveys, Cheaply

Roth and Schoenebeck, EC 2011.

Results:

- WLOG to consider "Take-It-Or-Leave-It" posted price mechanisms.

→ **Reduces the problem to picking a single posted-price distribution.**

- Must assume agents then report true costs!

- Describes posted-price distribution giving unbiased estimator with close to minimum *worst-case expected variance*.

Conducting Truthful Surveys, Cheaply

Roth and Schoenebeck, EC 2011.

What we want to do differently:

- More complex learning problems.
- ***Iterative*** rather than their ***simultaneous/decentralized*** approach.
- Generalization-error type bounds.

Importance-Weighted Active Learning

Beygelzimer, Dasgupta, and Langford, ICML 2009.

Problem: Learn while buying a small *number of labels*.
Assumptions:
- All costs are 1.
- Algorithm **can observe x** before deciding.
- **Iterative approach!**

Goal: Buy few labels, compare to if we'd bought *all* labels.

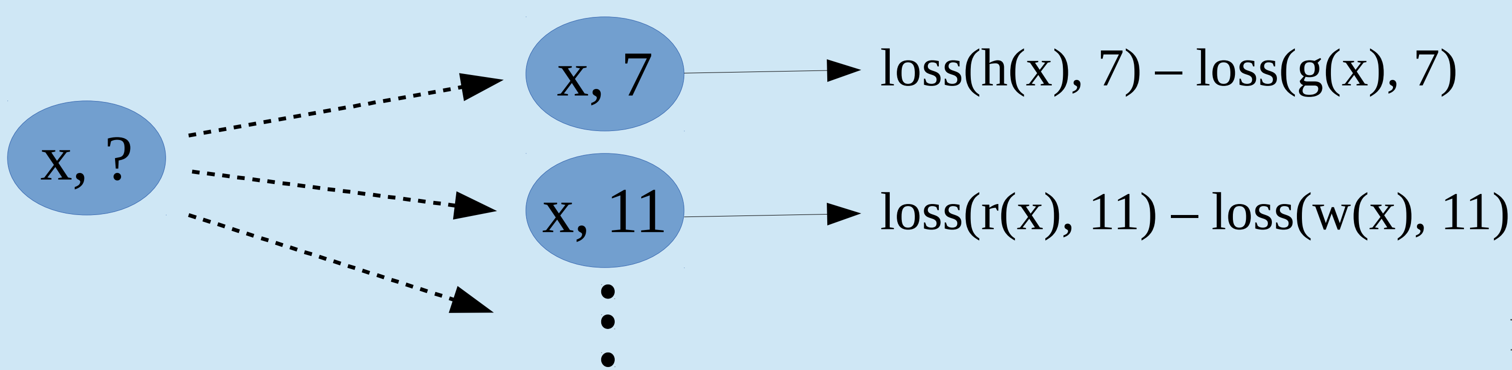
Importance-Weighted Active Learning

Beygelzimer, Dasgupta, and Langford, ICML 2009.

Results:

- **IWAL framework:** for each arriving point, set probability of sampling, then importance-weight losses to get unbiased estimators of expected loss.

- Instantiation: continuously narrow hypothesis set;
  sampling probability = possibility to distinguish within hypothesis set



Importance-Weighted Active Learning

Beygelzimer, Dasgupta, and Langford, ICML 2009.

What we'd like to do differently:

- Modify **existing** learning algorithms and (hopefully) leverage their guarantees.
  → We'll use no-regret algorithms.

- Agents have costs in [0,1].

- Not just worst-case guarantees, but understanding when we do well.

Outline

- General setting
- Related work
- Our approach

Our approach

Ideal world:

Here's my **learning** problem, and here's a good online learning algorithm for it!

Abra Kadabra Alakazam!
…
OK, here is a **mechanism** for you to use!

Our approach

Ideal world:

Here's my **learning** problem, and here's a good online learning algorithm for it!

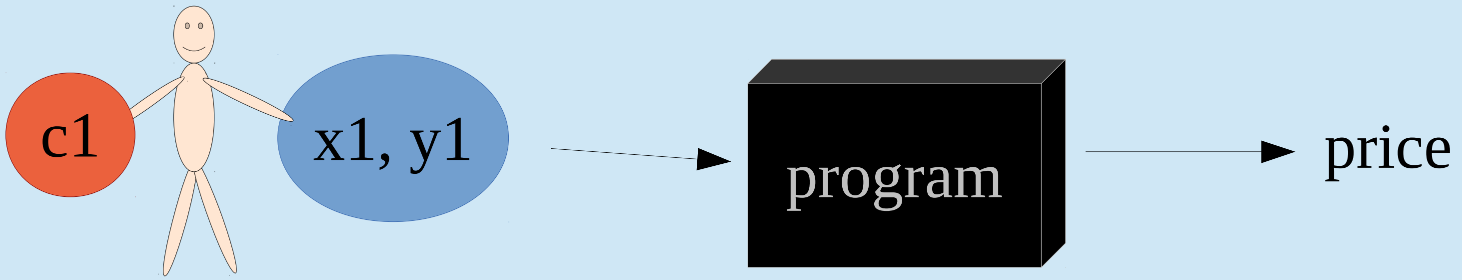
Abra Kadabra Alakazam!
…
OK, here is a **mechanism** for you to use!

By the way, here's a **regret bound** for that learning algorithm!

…
OK, here is a **generalization error and budget bound** for that mechanism!

Our approach

- Key assumption: mechanism can set price based on **both** x and y! (and agents cannot misreport x,y)
- Example: medical data (difficult to misreport).
- Implementation: give agents a price-calculating program.



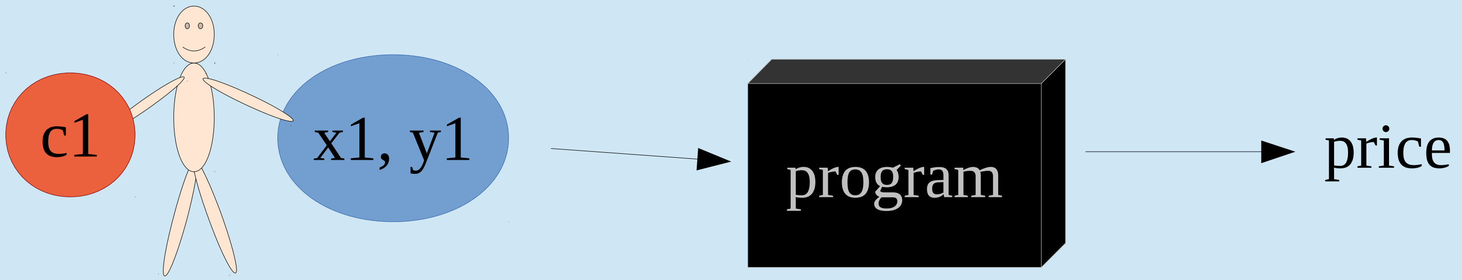
General Framework

Given a no-regret algorithm for the problem:
1. Decide the **"value"** of the next agent's data point.

2. (Randomly) set a **posted price** based on this value and the marginal cost distribution.

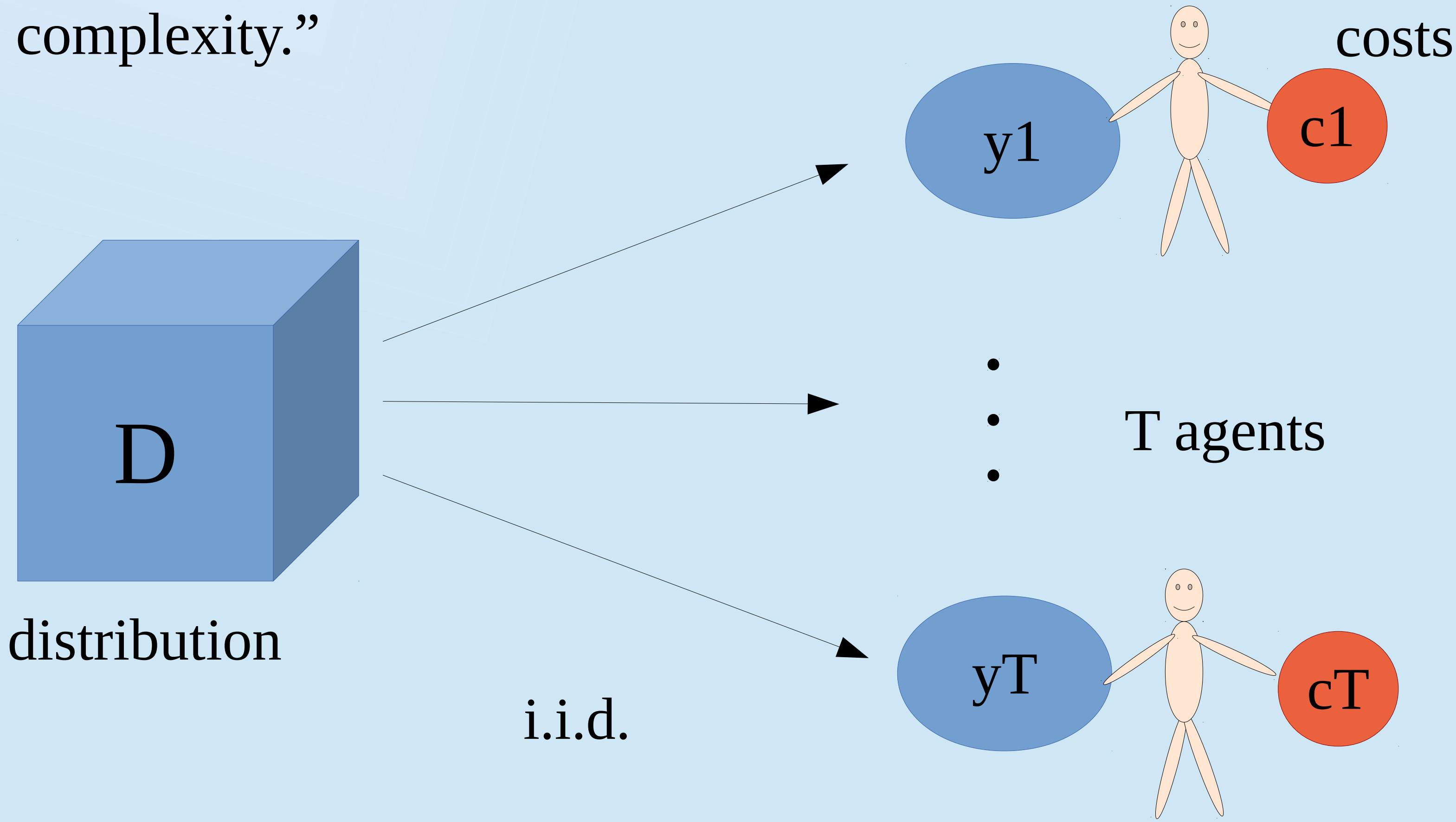
3. If taken, **importance-weight the loss** based on the probability the random price would've been accepted. Update the no-regret algorithm.

4. Repeat.



Simple example: estimate the mean

Assume all costs are 1.
→ “Label complexity.”



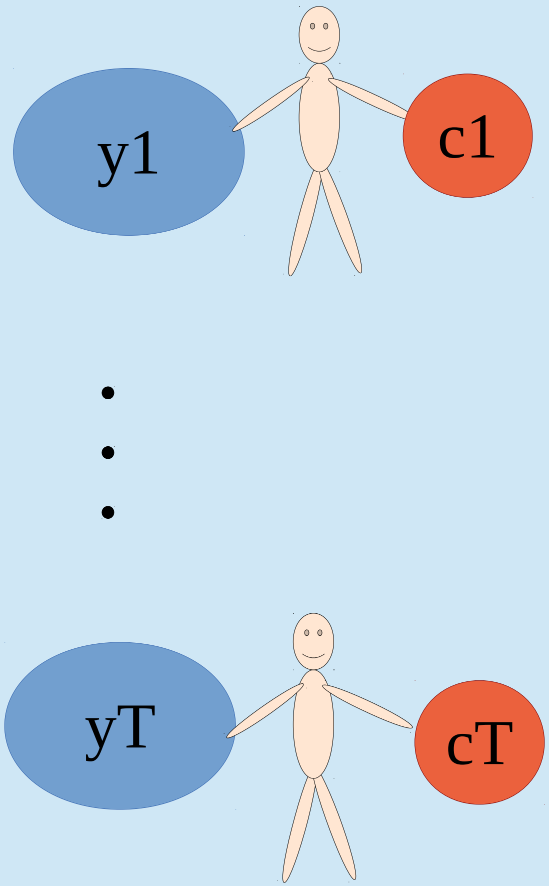
Simple example: estimate the mean

Assume all costs are 1.
→ "Label complexity."

No-regret algorithm: h = sample mean.

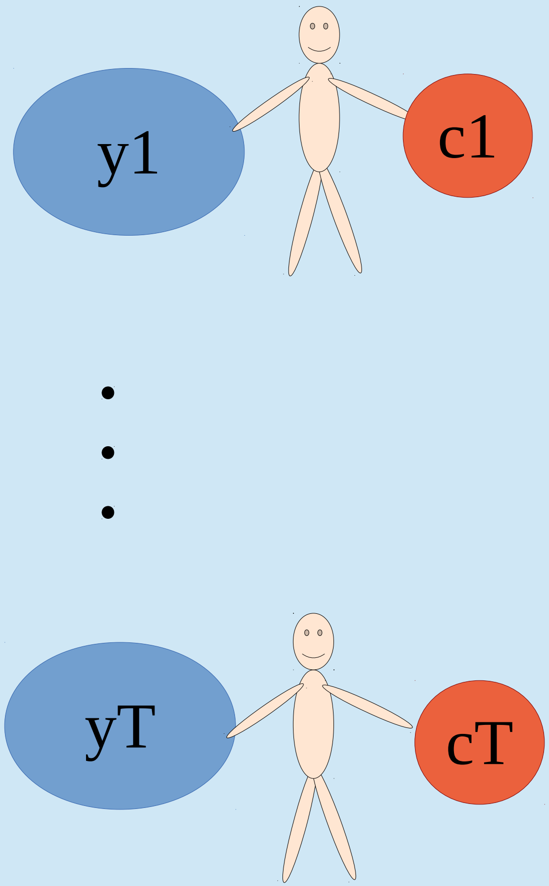
Benchmark: buy all T labels.
Let u = true mean.
→ with prob. 1-d, $|h - u| = O\left(\sqrt{\frac{\ln(2/d)}{T}}\right)$

Can we improve somehow??



Applying our framework

1. Decide the "value" of the next data point.
2. (Randomly) set a posted price.
3. If taken, importance-weight and update.
4. Repeat.



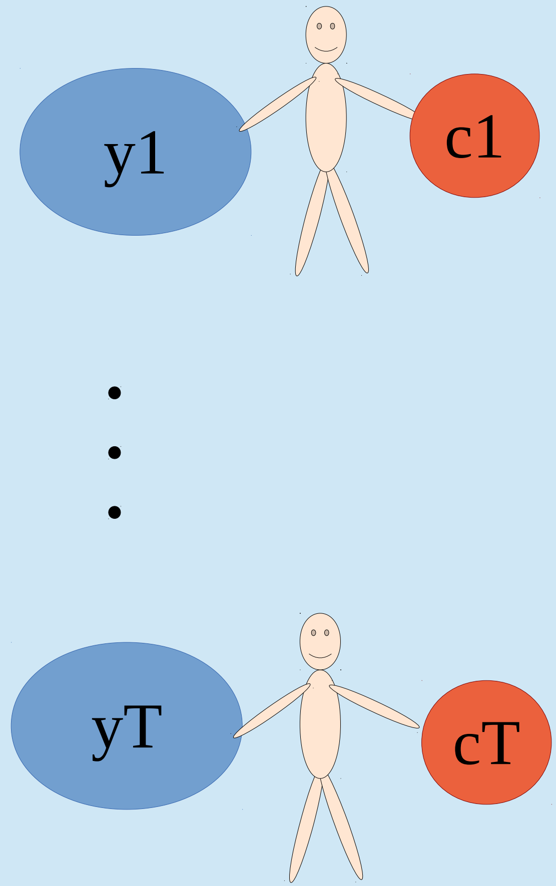
Scheme A:
Set value pt = yt.
Buy with probability pt.
→ Error within constant factor of benchmark!
→ Purchase ~ uT labels! (u = mean)

Applying our framework

1. Decide the "value" of the next data point.
2. (Randomly) set a posted price.
3. If taken, importance-weight and update.
4. Repeat.

Scheme B:
Set value pt = |ht – yt| + $\sqrt{\frac{\ln(T)}{t}}$

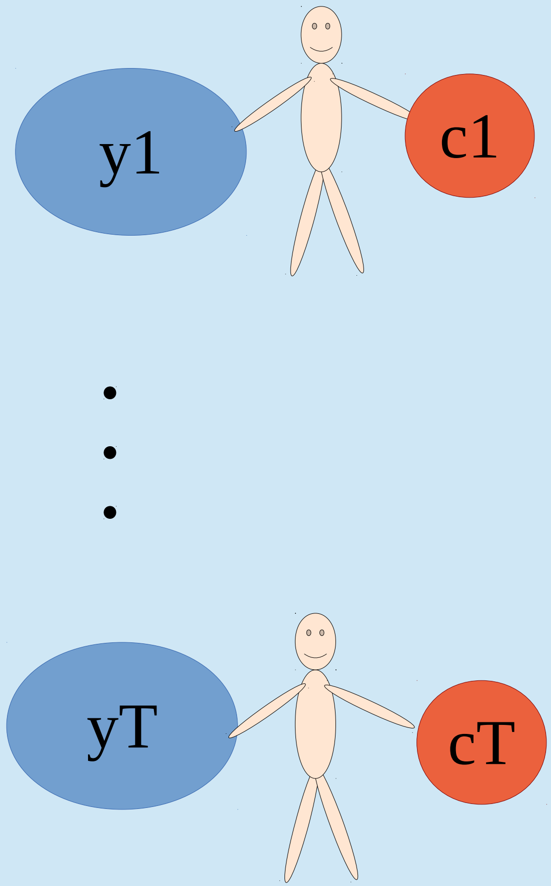
Buy with probability pt.
(I think) this should give:
→ Error "close" to benchmark
→ Purchase ~ oT labels (o = std deviation)



What about costs in [0,1]?

→ Could compose our mechanism with Roth-Schoenebeck.
→ Guarantees? (e.g. spend ~ uTc, where c = average cost?)

Seems hard to tell from their analysis, may want another approach.

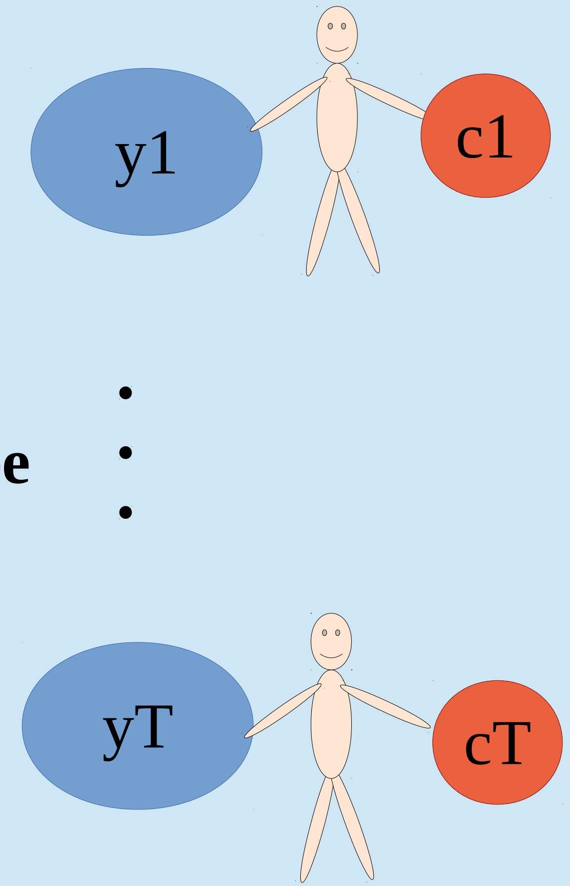


Why this might hopefully work in general

- No-regret algorithms guarantee average **regret** of 1/sqrt(T) or better.

- When drawing examples i.i.d., only want **generalization error** 1/sqrt(T).

- If problem has regret guarantee better than 1/sqrt(T), try to **convert to budget guarantee** while keeping acceptable g.e.



Wrapup of talk

- Problem: seemingly natural but tricky!
- Need to think carefully about **assumptions**.
- Our approach: tweak existing no-regret algorithms, use them to set prices and probabilities.
- When regret is smaller than needed for good generalization error, trade off **regret** and **budget** using **importance-weighting**.
- Todo: understand/prove this “generally”!