Announcements:

- By Tuesday midnight, start submitting your class reviews:
  - First paper: Human-powered sorts and joins
- Start thinking about projects!!

Question

- How much time should you spend reading the paper before a class?

A: Not too long! Min: 1-2 hours, Max: 3-4 hours (only if you're lacking necessary background or have a genuine interest)

An Introduction to Crowdsourced Data Management

+

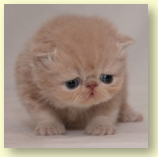
Crowdscreen: Algorithms for Filtering Data using Humans
(if time permits) Optimal Crowd-Powered Rating and Filtering Algorithms

Crowdsourcing: A Quick Primer

Asking the crowd for help to solve problems

Why? Many tasks done better by humans

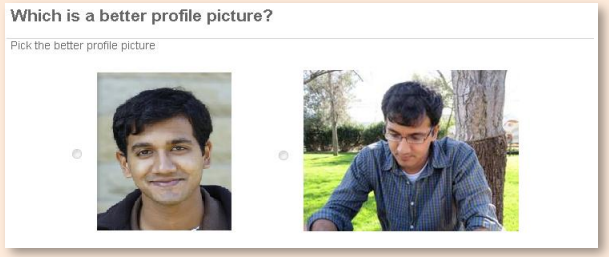
Pick the “cuter” cat




Is this a photo of a car?

How? We use an internet marketplace

Requester: Aditya Reward: 1$ Time: 1 day



At a high level…

- I (and other requesters) post my tasks to a marketplace
  - one such marketplace is Mechanical Turk, but there are 30+ marketplaces
- Workers pick tasks that appeal to them
- Work on them
- I pay them for their work

Pay anywhere from a few cents to dollars for each task; get the tasks done in any time from a few seconds to minutes

Why should I care?

- Most major companies spend millions of $$$ on crowdsourcing every year
  - This includes Google, MS, Facebook, Amazon

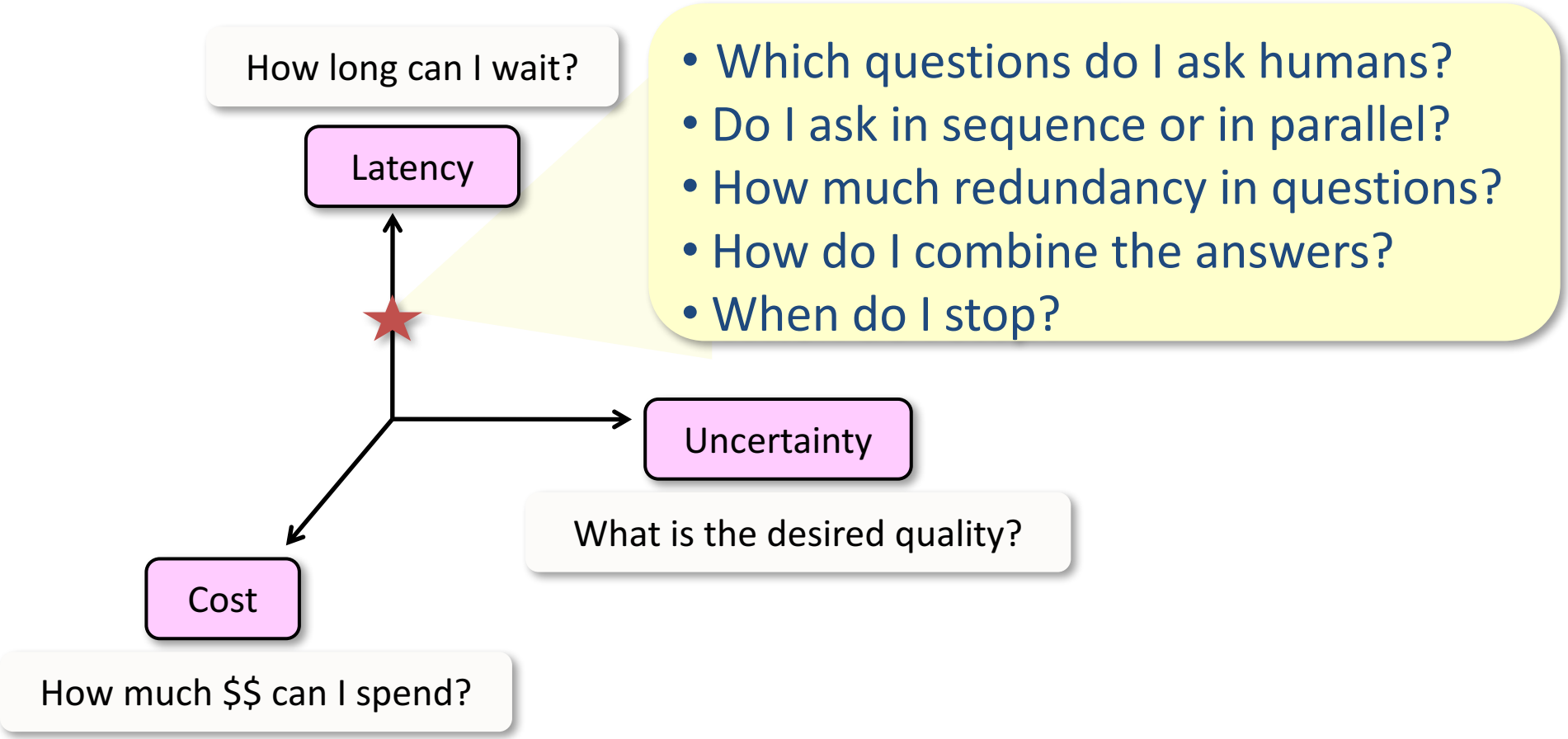
- Represents our only viable option for understanding unstructured data
- Represents our only viable option for generating training data @ scale

OK, so why is this hard?

- People need to be **paid**
- People take **time**
- People make **mistakes**

- And these three issues are correlated:
  - If you have more money, you can hire more workers, and thereby increase accuracy
  - If you have more time, you can pay less/hire more workers, and thereby increase accuracy/reduce costs
  - …

Fundamental Tradeoffs



Need to Revisit Basic Algorithms

- Given that humans are complicated unlike computer processors, we need to revisit even **basic data processing algorithms where humans are "processing data"**
  - Max: e.g., find the best image out of a set of 1000 images
  - Filter: e.g., find all images that are appropriate to all
  - Categorize: e.g., find the category for this image/product
  - Cluster: e.g., cluster these images
  - Search: e.g., find an image meeting certain criteria
  - Sort: e.g., sort these images in terms of desirability
- Using human unit operations:
  - Predicate Eval., Comparisons, Ranking, Rating

Goal: Design efficient crowd algorithms

Assumption: All humans are alike

- A common assumption to make in many papers of crowdsourcing is that humans are **identical, noisy, independent oracles**

- If the right answer (for a bool q.) is 1, humans will give 0 with equal probability p, and 1 with probability (1-p).
- Humans will also answer independent of each other.
- E.g., if I ask any human if a image contains a dog, they answer incorrectly with probability p.

Discussion Questions

Why is it problematic to assume that humans are *identical*, *independent*, *noisy* oracles with *fixed error* rates?

Discussion Questions

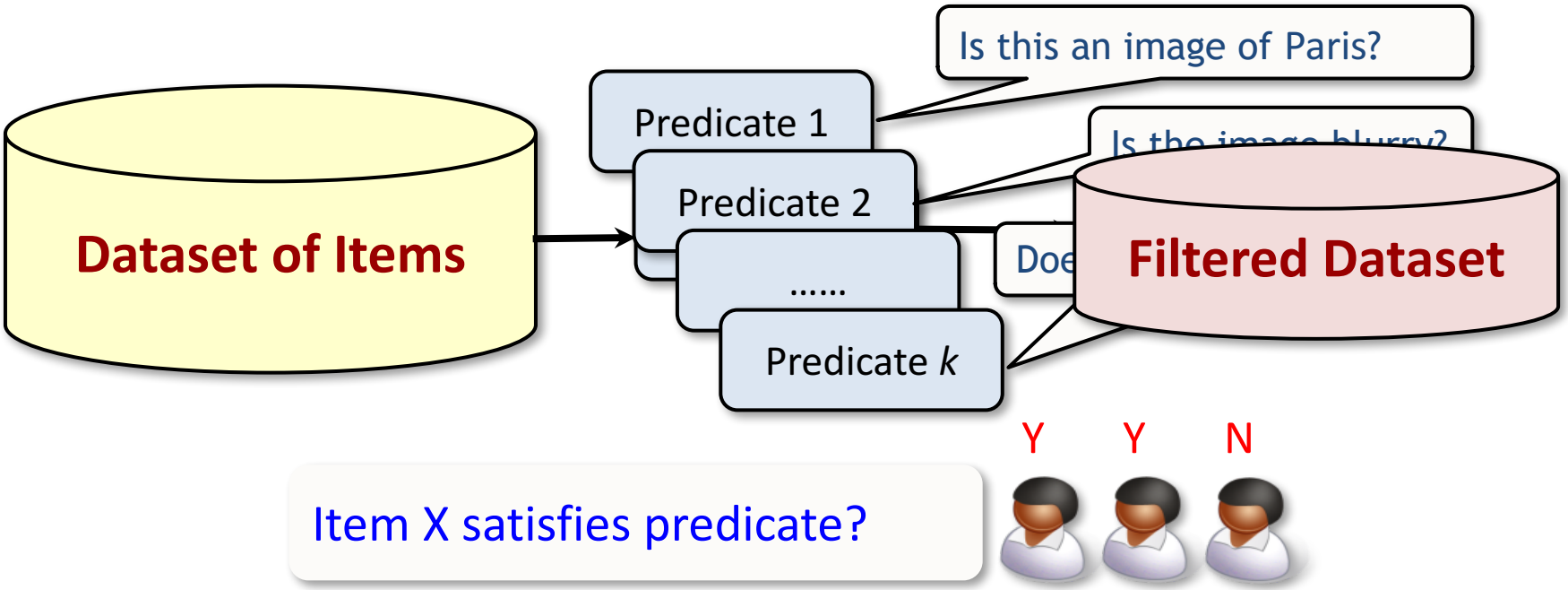
Why is it problematic to assume that humans are identical, independent, noisy oracles with fixed error rates?

- Humans may be different
- Error rates change over time, over questions
- Difficulty of question affects independence

Our Focus for Today

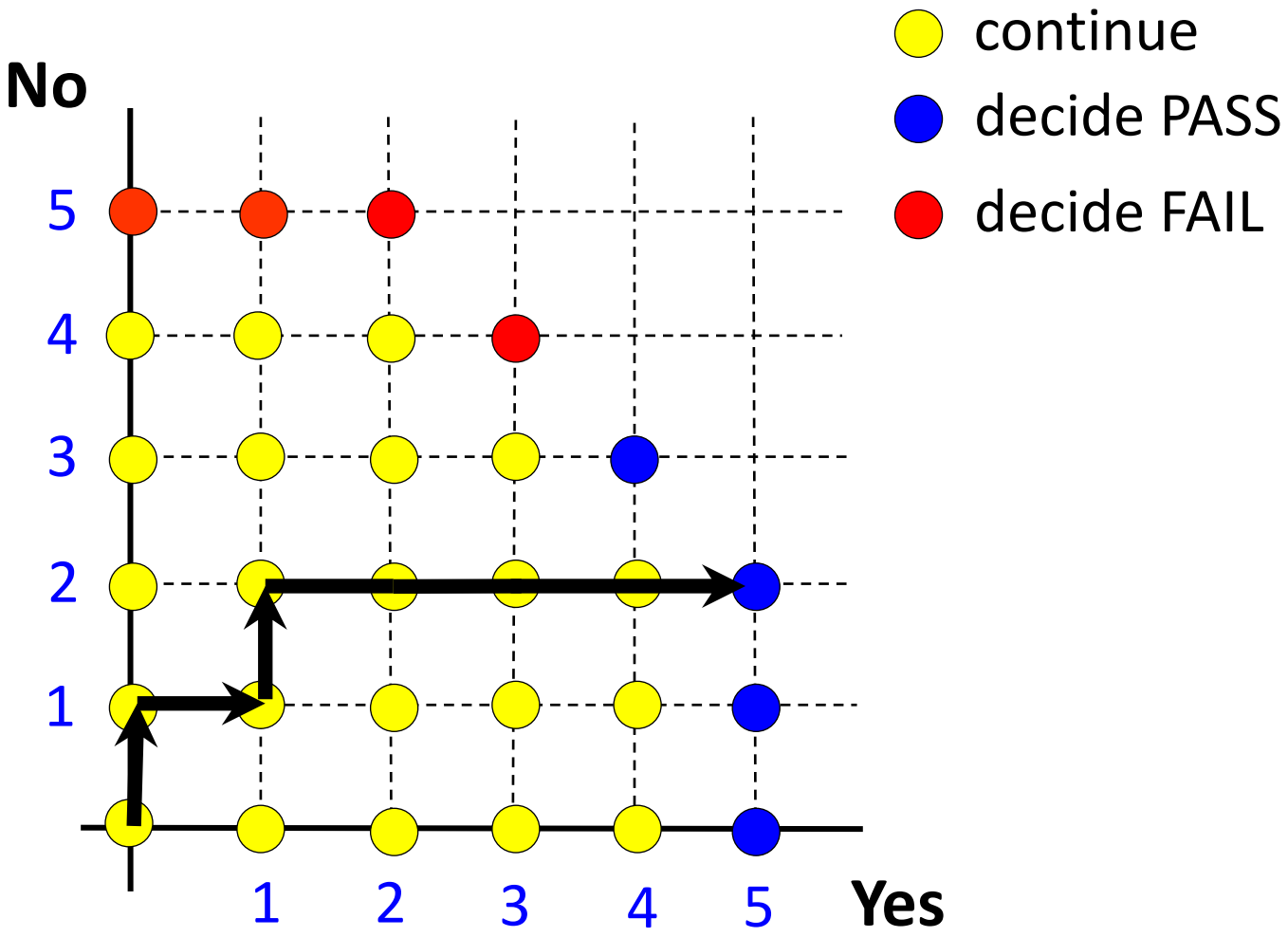
Filtering

Single Filter

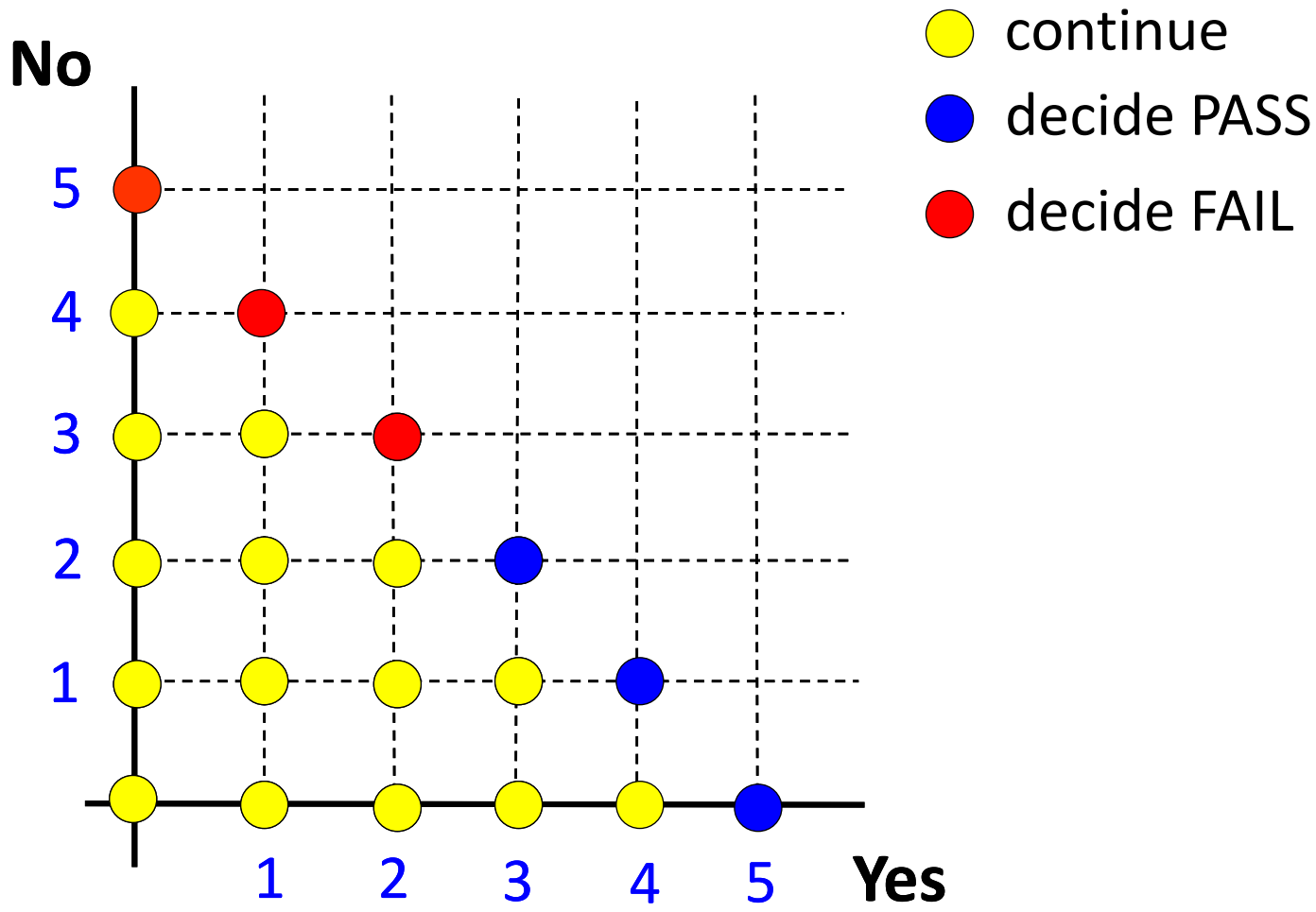


Applications: Content Moderation, Spam Identification, Determining Relevance, Image/Video Selection, Curation, and Management, …

Our Visualization of Strategies



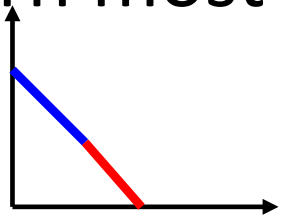
Strategy Examples



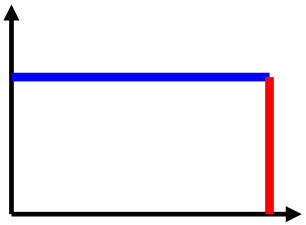
What's a short description for this strategy?

Common Strategies

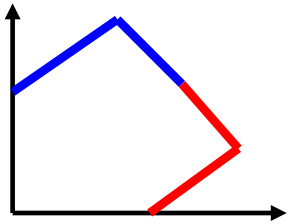
- Always ask X questions, return most likely answer
  - Triangular strategy



- If X YES return “Pass”, Y NO return “Fail”, else keep asking.
  - Rectangular strategy



- Ask until |#YES - #NO| > X, or at most Y questions
  - Chopped off triangle



Simplest Version

Probability of mistakes; all humans alike

Given:

- Human error probability (FP/FN)
  - Pr [Yes | 0]; Pr [No | 1]
- A-priori probability
  - Pr [0]; Pr[1]

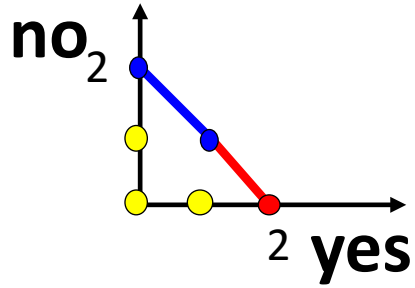
Selectivity

Via Sampling
Or Prior History

We will discuss if this is reasonable to assume later.

Illustration

If we have error prob = 0.1, selectivity 0.7, then:



Pr. (reaching (0,0)) = 1

Pr. (reaching (0,0) and item = 1) = 0.7

Pr. (reaching (1,0) and item = 1) =

Pr (reaching (0, 0) and item = 1 and human answers Yes) = 0.7 * 0.9

The ability to perform computations like this on the state space (x, y) means that this a Markov Decision Process (MDP)

Simplest Version

Given:

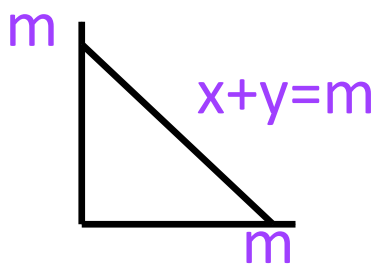
- Human error probability (FP/FN)
  - Pr [Yes | 0]; Pr [No | 1]
- A-priori probability
  - Pr [0]; Pr[1]

Via Sampling
Or Prior History

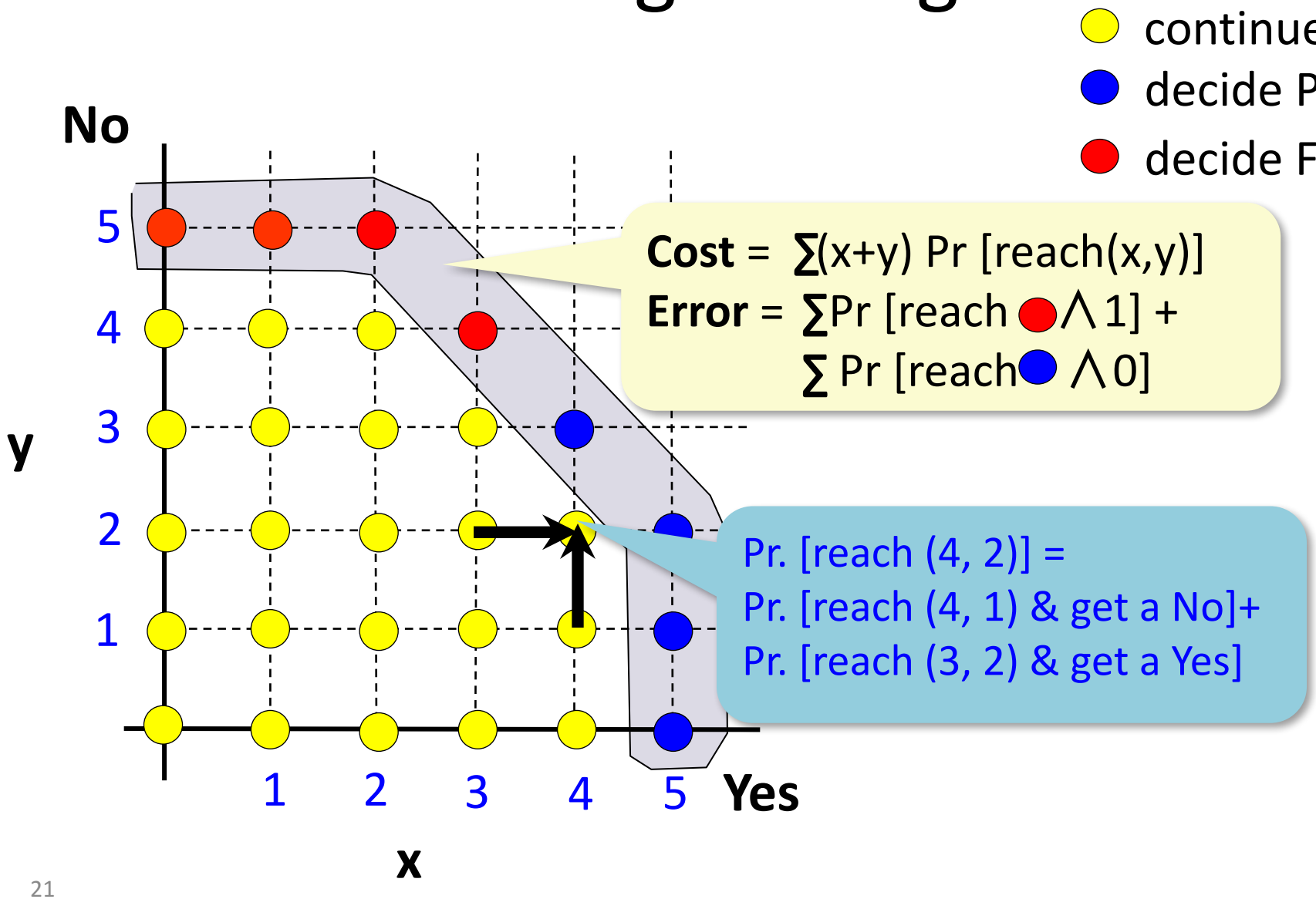
No Latency for now!

Find **strategy** with minimum expected cost (# of questions)

- Expected error < t (say, 5%)
- Cost per item < m (say, 20 questions)



Evaluating Strategies



Brute Force Approach

For each grid point

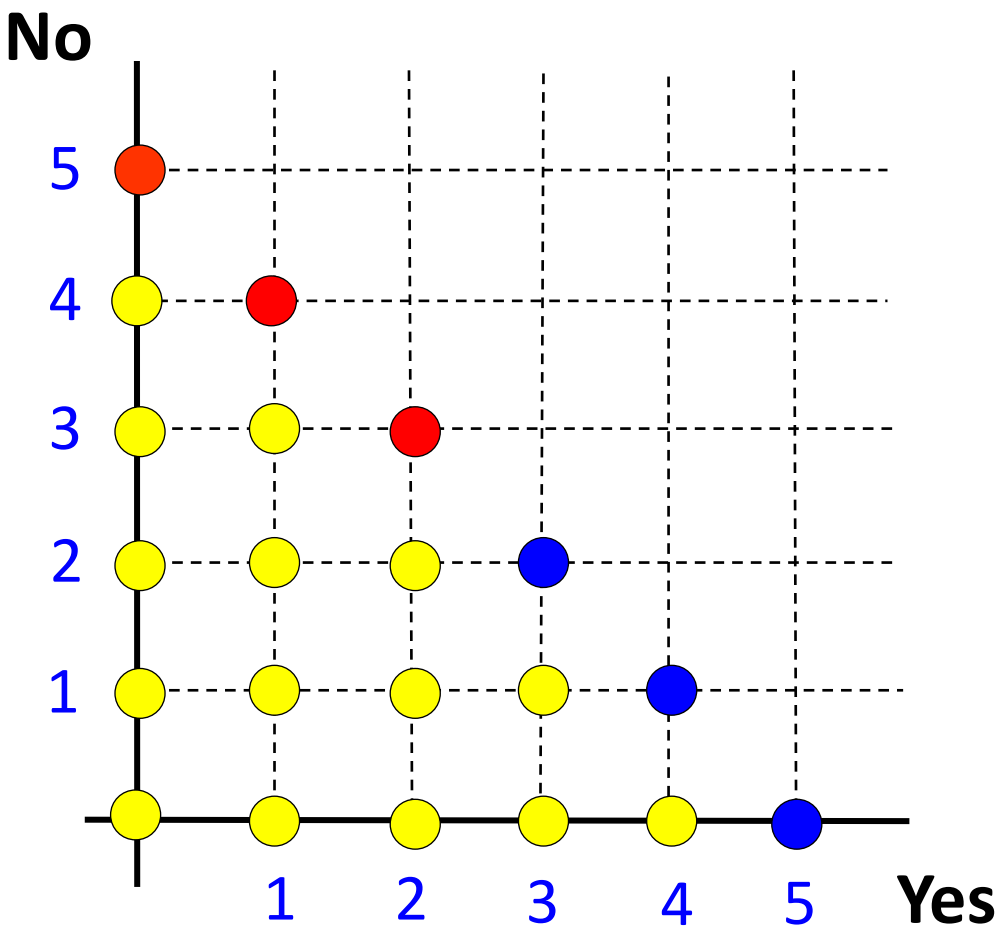
Assign ●, ●, or ●

For all strategies:

- Evaluate cost & error

Return the best

$O(3^g)$, $g = O(m^2)$



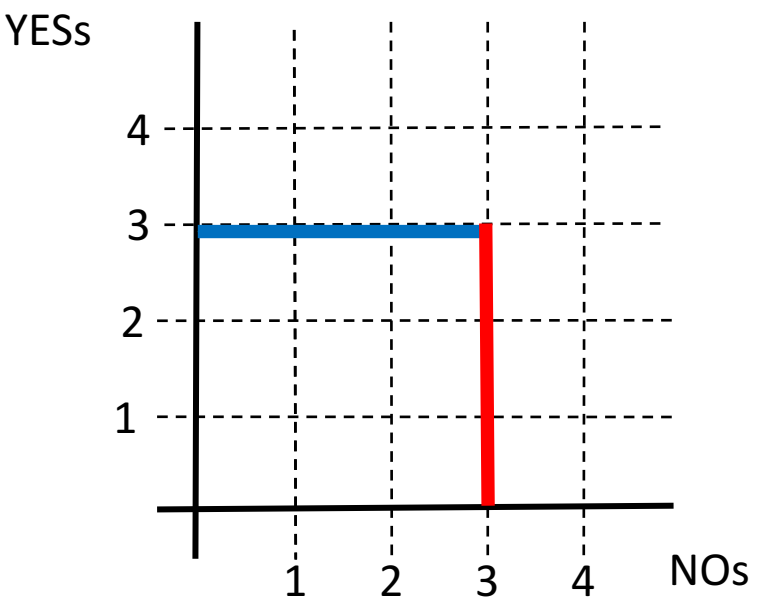
Brute Force Approach 2

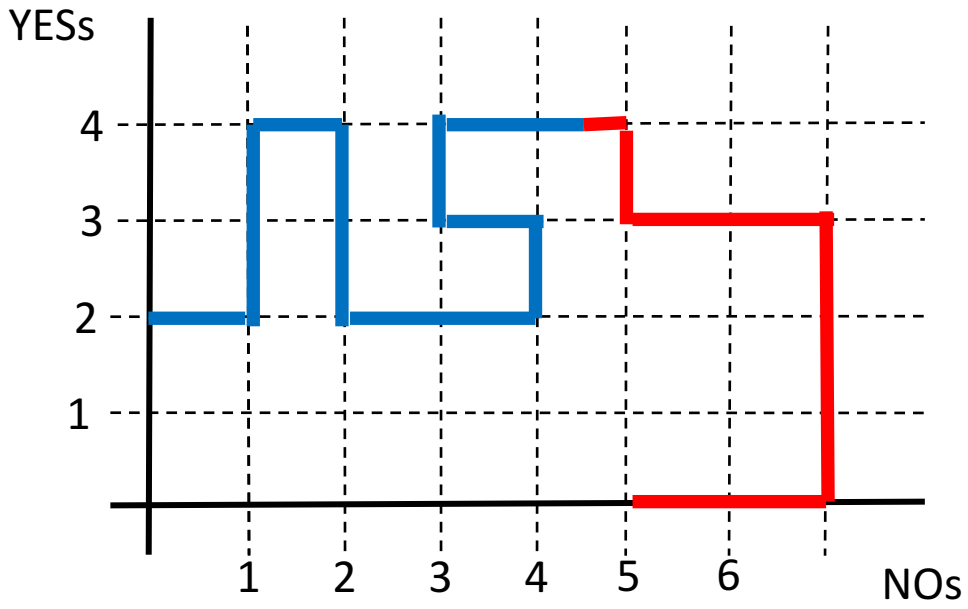
Try all "hollow" strategies

Too Long!

Sequence of blue + Sequence of red, connected

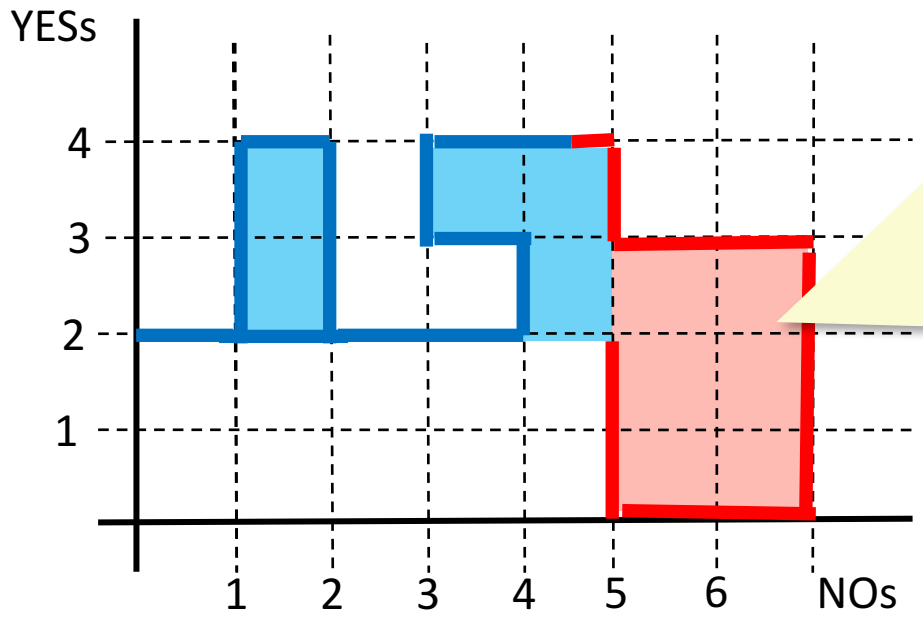
Why: Decisions to be made at boundary instead of internal



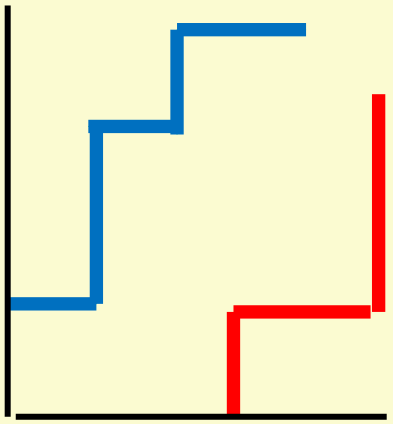


What's weird about this strategy?

Trick: Pruning Hollow Strategies



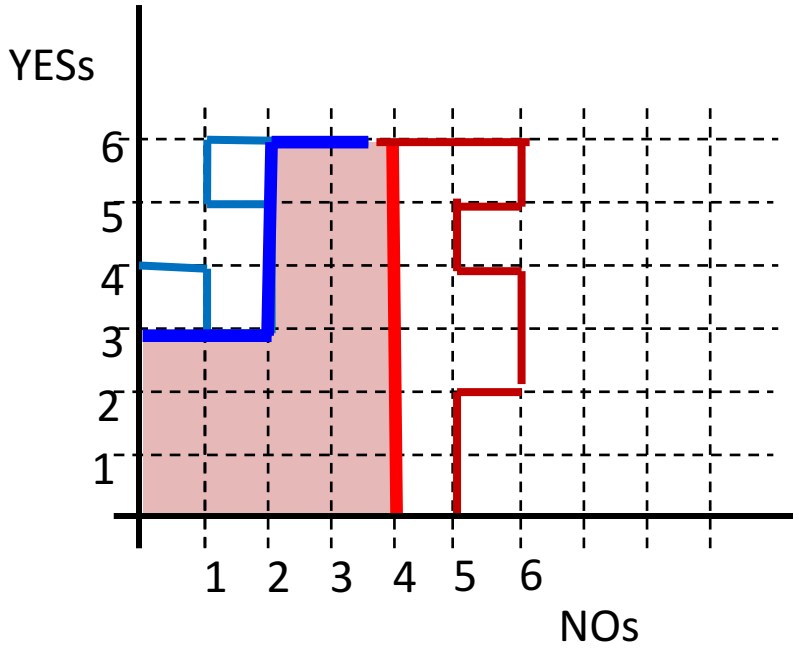
For every hollow strategy, there is a ladder strategy that is as good or better.

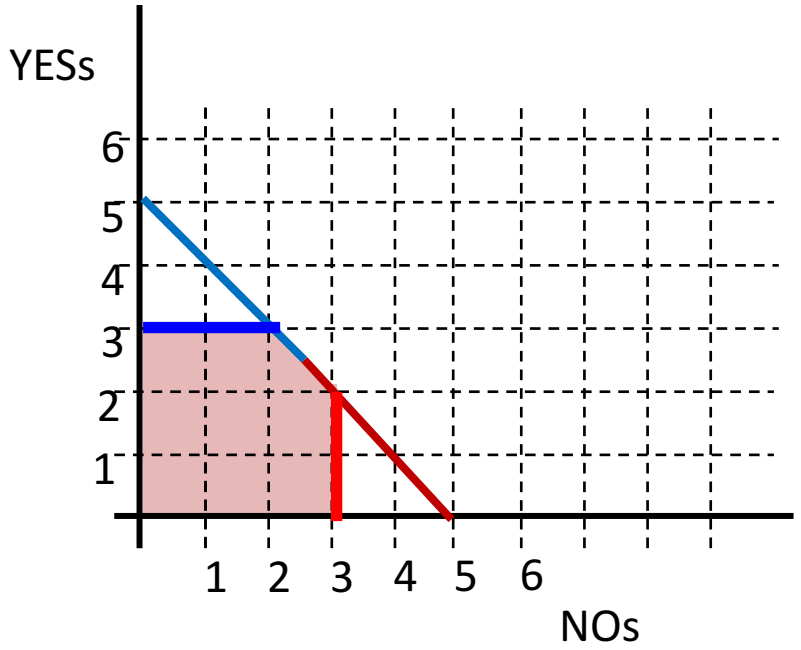


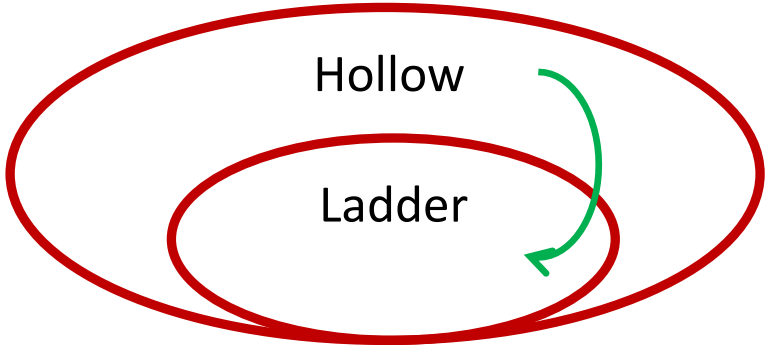
There are portions of the strategy that are unreachable and redundant.

If we prune them, we get a ladder strategy.

Other Pruning Examples





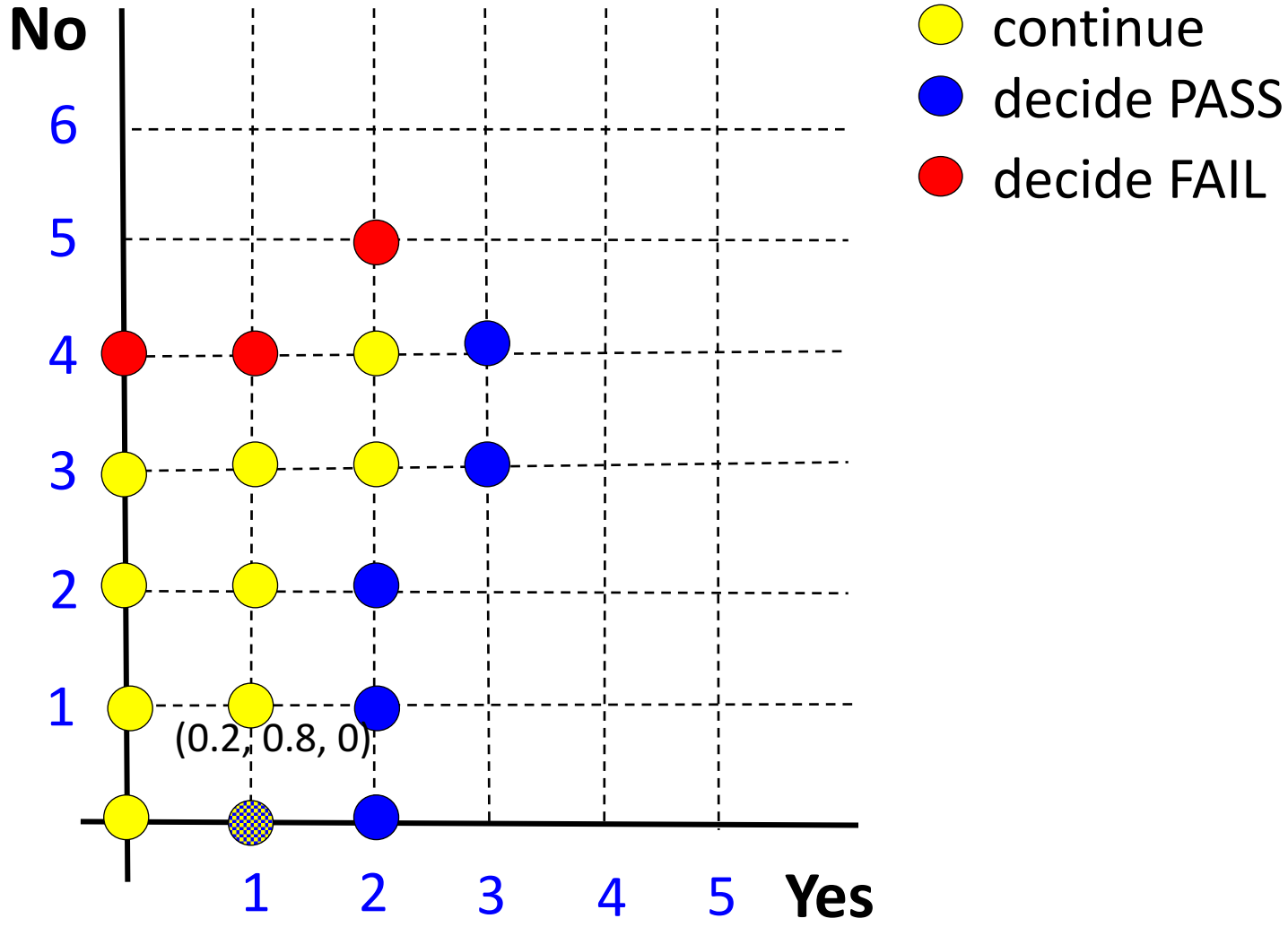


Comparison

	Computing Strategy	Money
Brute Force	Not feasible	$$
Ladder	Exponential; feasible	$$$

Now, let's consider a generalization of strategies

Probabilistic Strategy Example



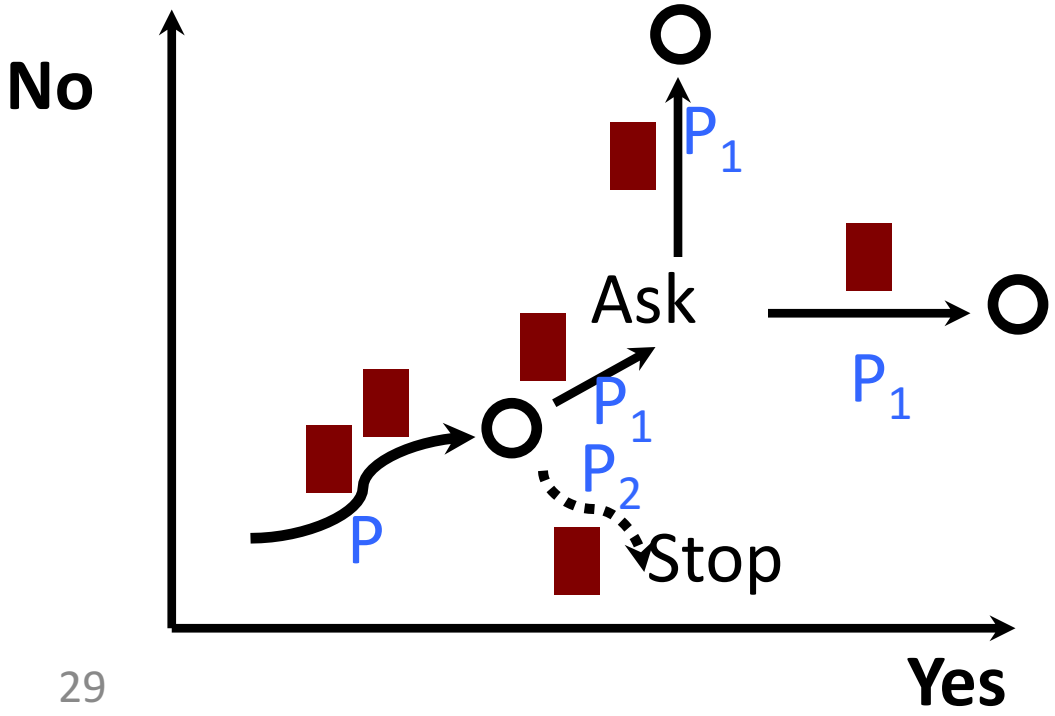
Comparison

	Computing Strategy	Money
Brute Force	Exponential; not feasible	$$
Ladder	Exponential; feasible	$$$
The best ~~probabilistic~~	Polynomial(m)	$

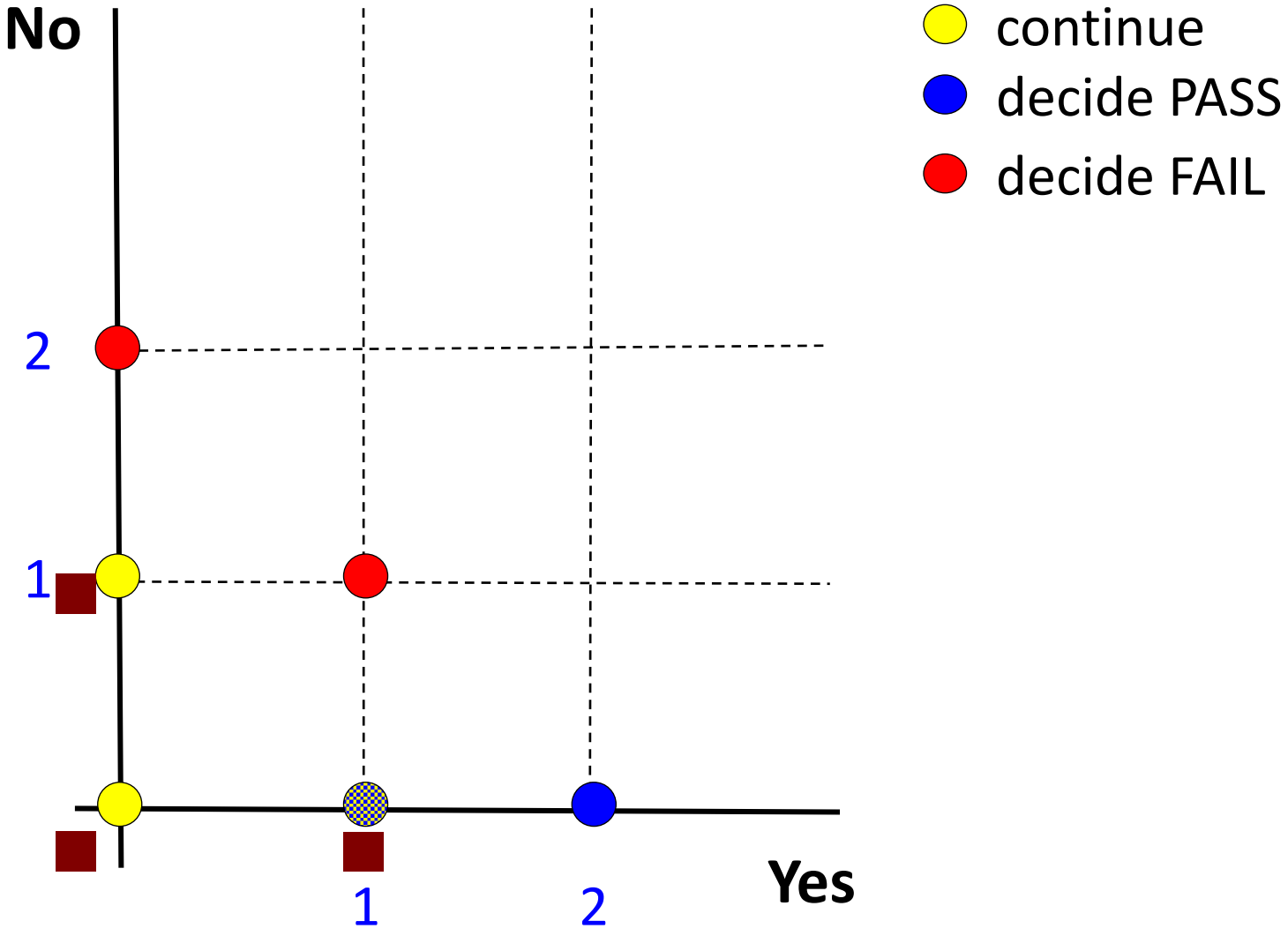
THE BEST

Key Property: Path Conservation

- P = # of (fractional) paths reaching a point
- Point splits paths: $P = P_1 + P_2$



Path Conservation in Strategies



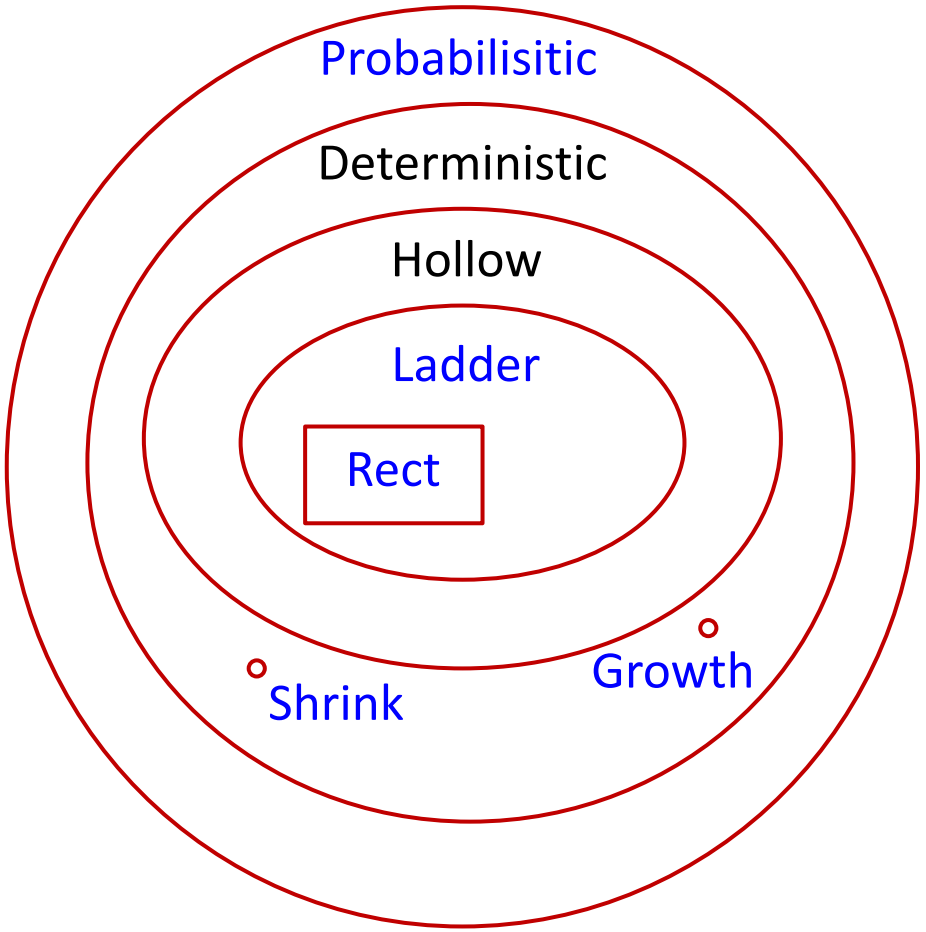
Finding the Optimal Strategy

Use Linear Programming:

$O(m^2)$ variables

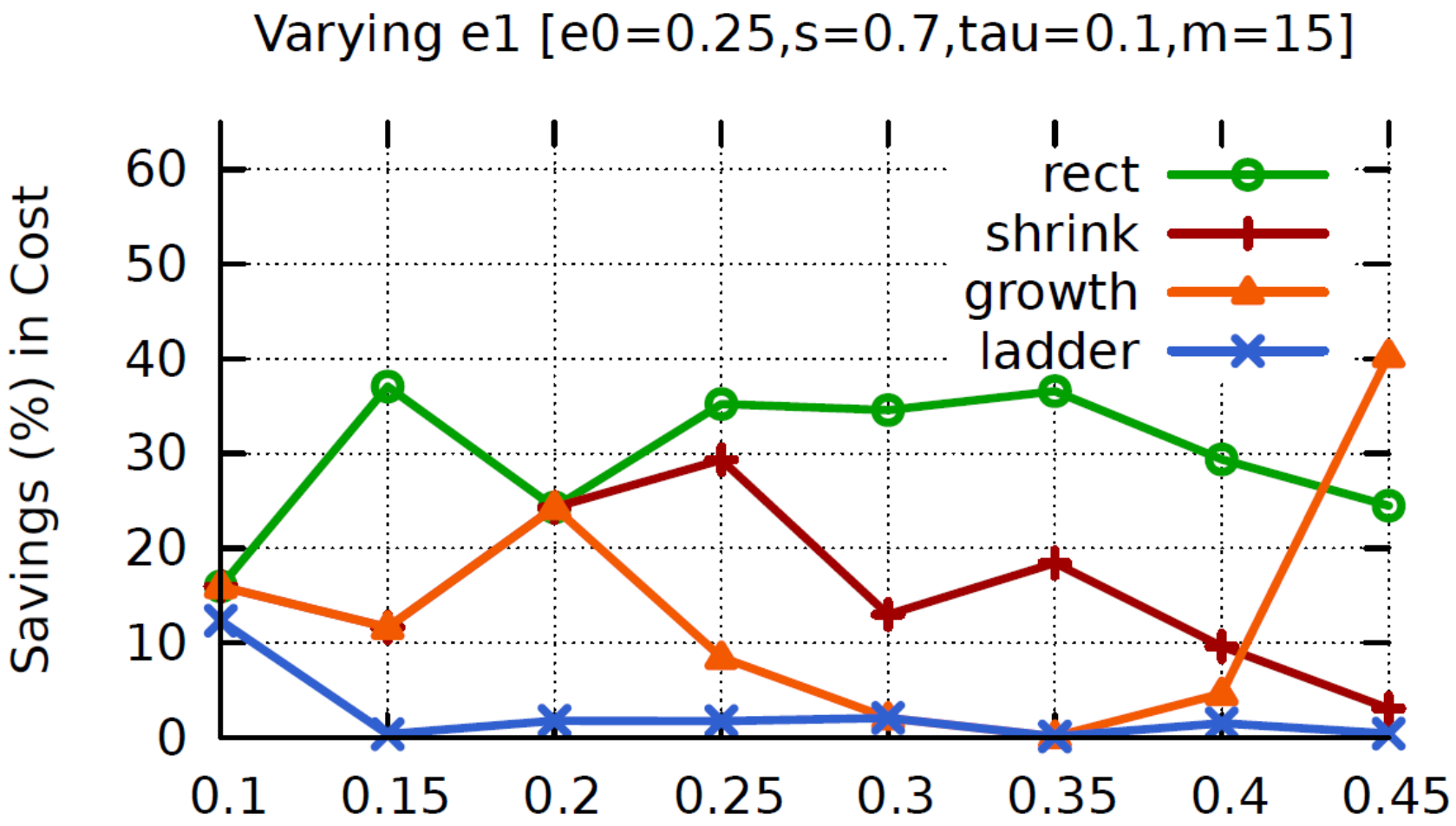
- Using Paths:
  - Paths P into each point X
  - Only decision at X is the split of P
- Also:
  - Pr [reach X] = c P
  - Pr [reach X $\wedge$ 1] = c' P
  - Saying PASS/FAIL at X does not depend on P

Experimental Setup

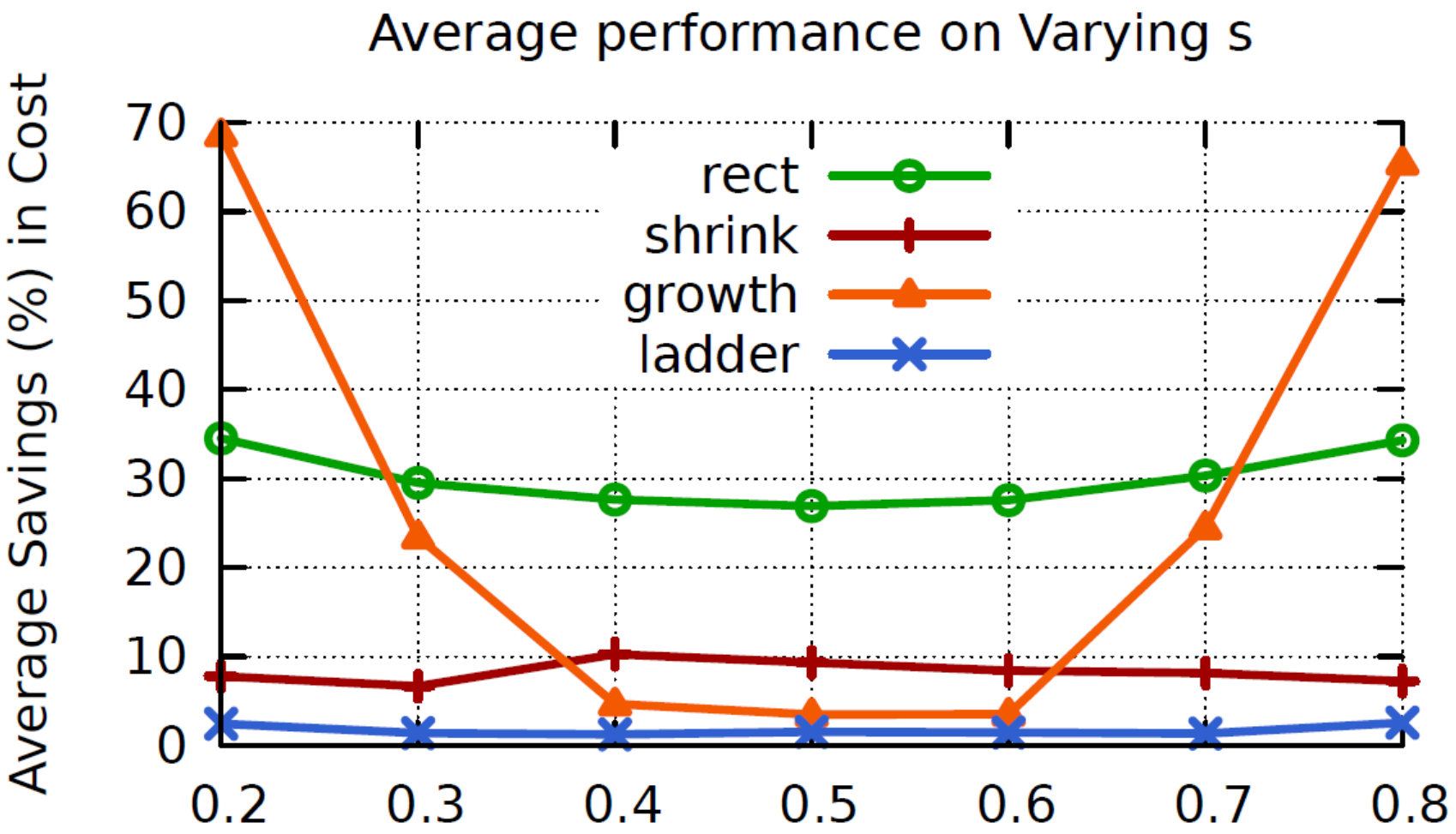


- Goal: Study cost savings of probabilistic relative to others
- Parameters → Generate Strategies → Compute Cost
- Two sample plots
  - Varying false positive error (other parameters fixed)
  - Varying selectivity (other parameters varying)

Varying false positive error

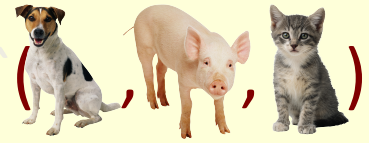


Varying selectivity



Two Easy Generalizations

Multiple Answers



(0, 1, 2, 3, 4, 5)

Multiple Filters

(Image of Paris) $\wedge$ (Not Blurry)

(F_1 Yes, F_1 No, …, F_K Yes, F_K No)
Ask F_1, …, Ask F_K, Stop

Time for discussion!

- Any questions, comments?

Discussion Questions

1: This paper assumes that error rates are known; why is that problematic?

Discussion Questions

1: This paper assumes that error rates are known; why is that problematic?

- How do you test? Testing costs money
- If error rates are imprecise, optimization is useless

Discussion Questions

2: How would you go about gauging human error rates on a batch of filtering tasks that you've never seen before?

Discussion Questions

2: How would you go about gauging human error rates on a batch of filtering tasks that you've never seen before?

- You could have the "requester" create "gold standard" questions, but hard: people learn, high cost, doesn't capture all issues
- You could try to use "majority" rule but what about difficulty, what about expertise?

Discussion Questions

3: Let’s say someone did all you asked: they created a gold standard, tested humans on it, generated an optimized strategy, but that is still too costly. How can you help them reduce costs?

Discussion Questions

3: Let's say someone did all you asked: they created a gold standard, tested humans on it, generated an optimized strategy, but that is still too costly. How can you help them reduce costs?

- Improve instructions & interfaces
- Training and elimination
- Only allow good workers to work on tasks
- Use machine learning

Discussion Questions

4: What other scenarios (beyond crowdsourcing) can these algorithms be useful?

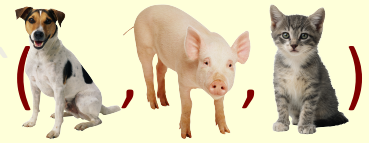
Discussion Questions

4: What other scenarios (beyond crowdsourcing) can these algorithms be useful?

- Anywhere you have noisy oracles!
  - Lab experiments
  - Automobile testing
  - Medical diagnosis

Discussion Questions

Multiple Answers

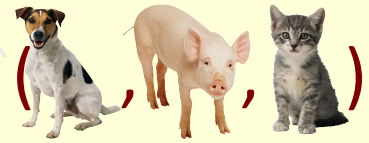


(0, 1, 2, 3, 4, 5)

5. Why is this generalization problematic? How would you fix it?

Discussion Questions

Multiple Answers



(0, 1, 2, 3, 4, 5)

5. Why is this generalization problematic? How would you fix it?

- You need O(n^k) data points

- Use generic error model, bucketize
- Priors

Discussion Questions

6. What are the different ways crowd algorithms like this can be used in conjunction with machine learning?

Discussion Questions

6. What are the different ways crowd algorithms like this can be used in conjunction with machine learning?

- Input/training
- Active learning
- ML feeds crowds

New Stuff

From “Optimal Crowd-Powered Rating and Filtering Schemes” VLDB 2014.

Generalization: Worker Abilities

	Item 1	Item 2	Item 3
Actual	0	1	0
W_1	0	1	0
W_2	1	1	1
W_3	1	0	1

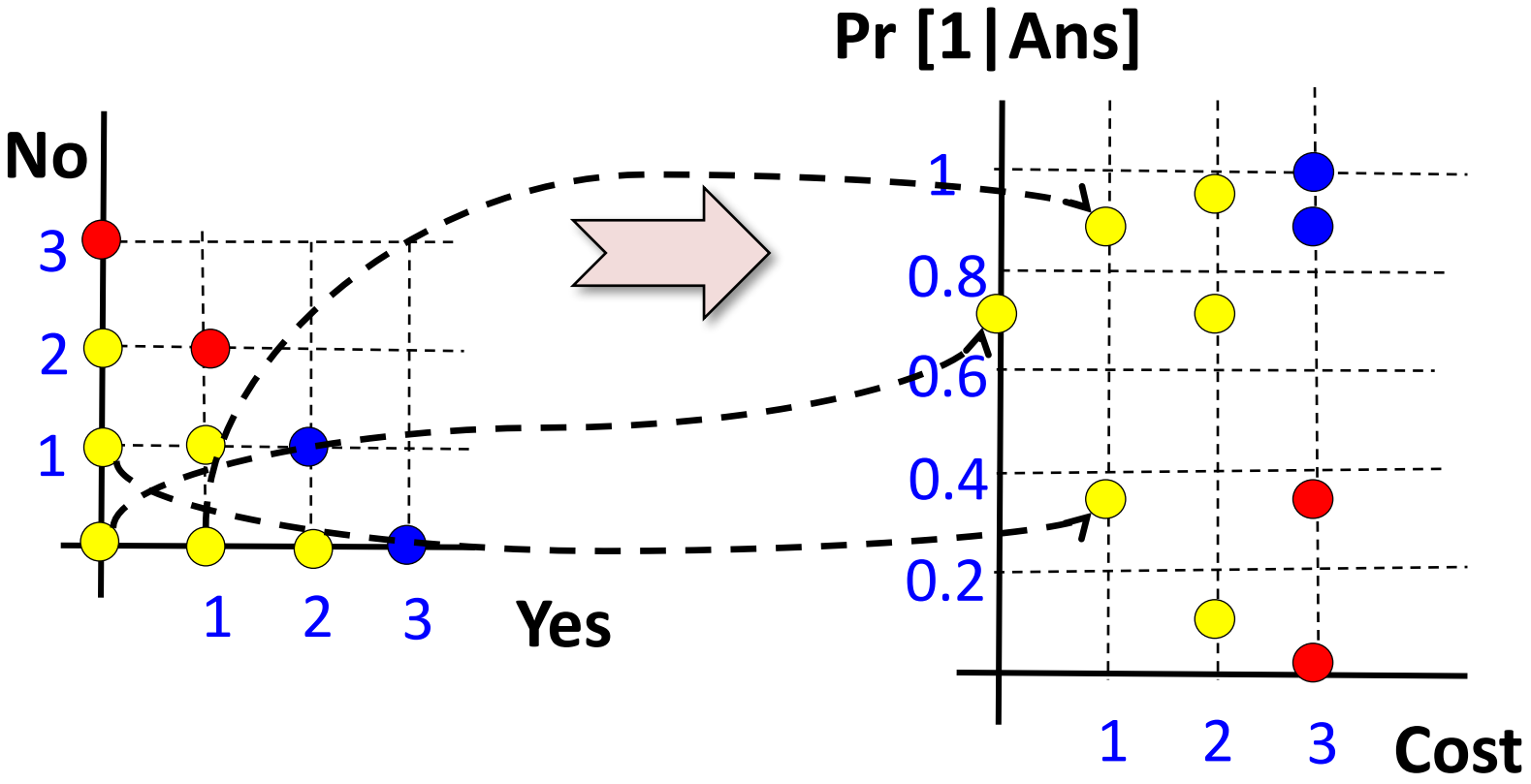
(W_1Yes, W_1 No, …, W_n Yes, W_n No)

$O(m^{2n})$ points

$n \approx 1000$

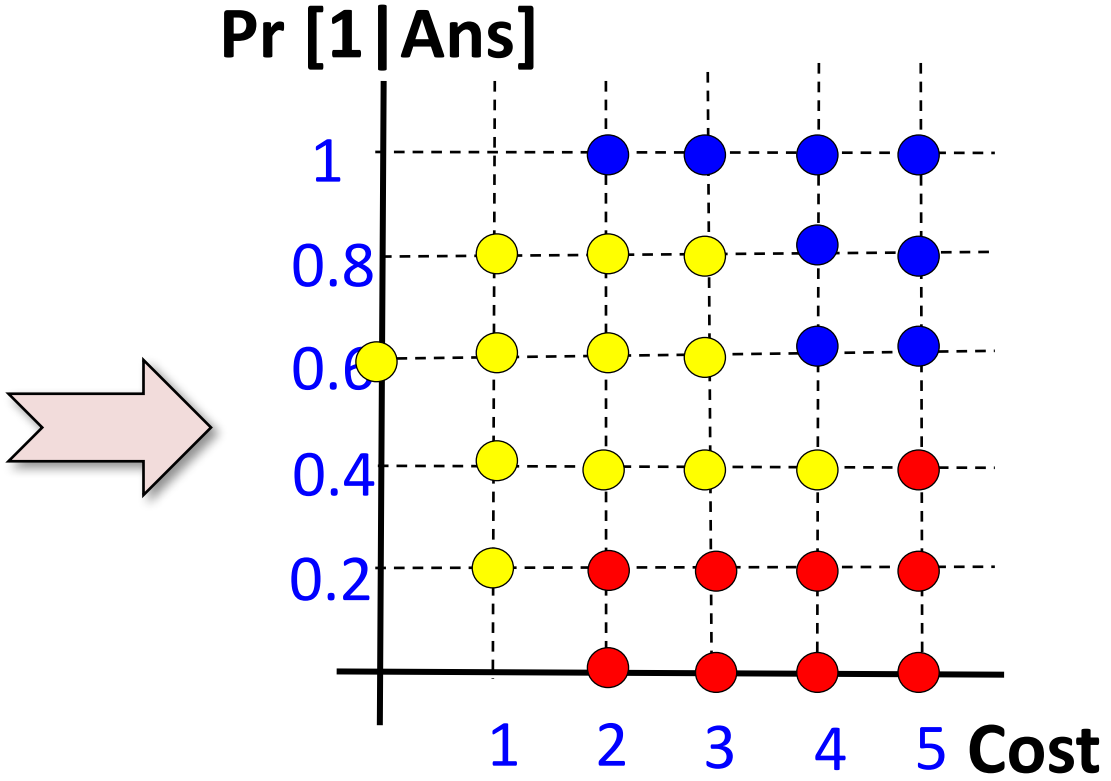
Explosion of state!

A Different Representation



Worker Abilities: Sufficiency

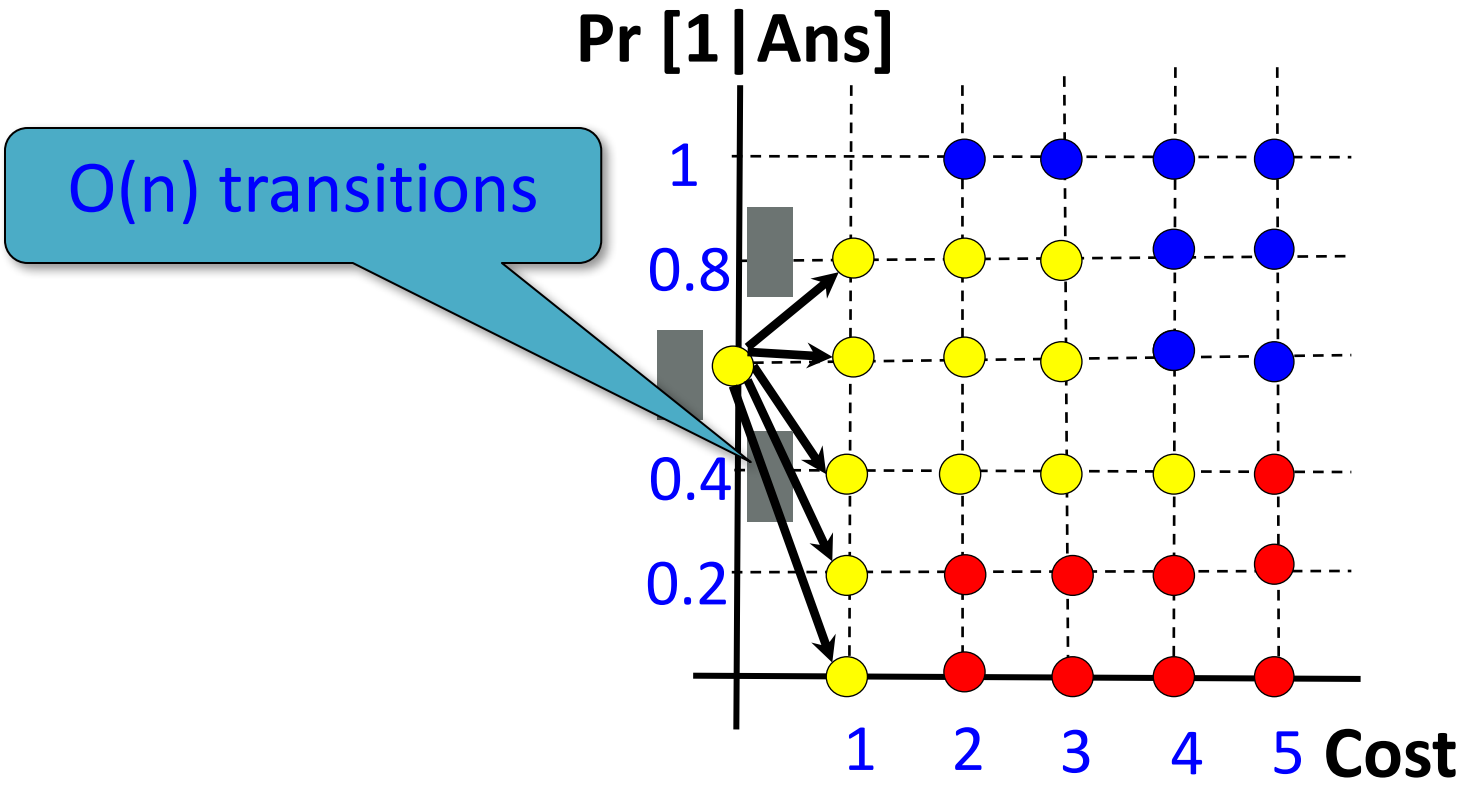
(W_1 Yes, W_1 No,
W_2 Yes, W_2 No,
…,
W_n Yes, W_n No)



Recording Pr[1 | Ans] is sufficient:

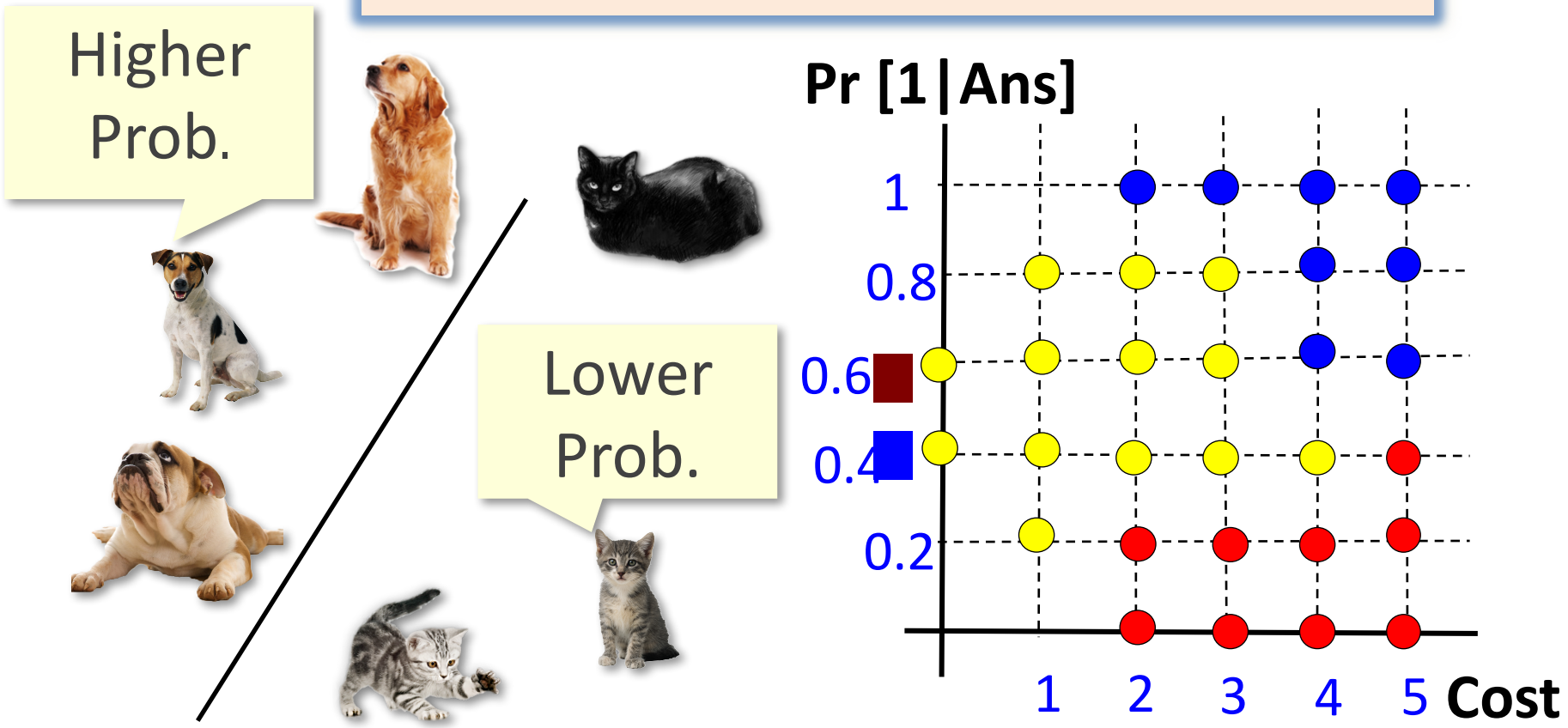
Strategy ➔ Optimal

Different Representation: Changes



Generalization: A-Priori Scores

ML algorithm providing, for each item, a probability estimate of passing filter



Generalizations

- Multiple answers (ratings, categories)
- Multiple independent filters
- Difficulty
- Different worker abilities
- Different worker probes
- A-priori scores
- Latency
- Different penalty functions

Hasn't this been done before?

- Solutions from elementary statistics guarantee the same error per item
  - Important in contexts like:
    - Automobile testing
    - Medical diagnosis
- We're worried about aggregate error over all items: a uniquely data-oriented problem
  - We don't care if every item is perfect as long as the overall error is met.
  - As we will see, results in $$$ savings