Micro-task Crowdsourcing Scalability

Lecture 6

Gianluca Demartini
University of Sheffield

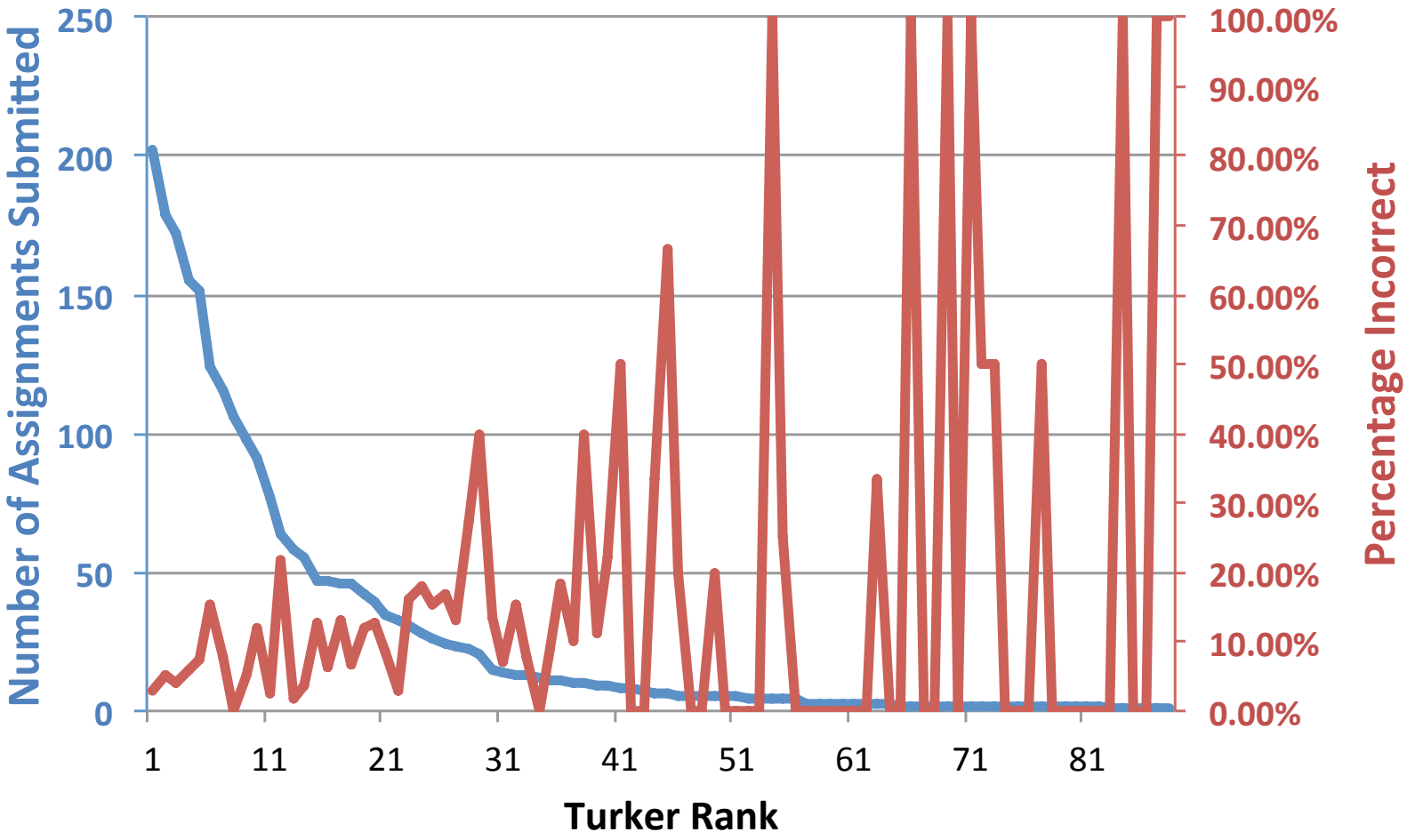
Crowdsourcing: Key Issues

- The role of machine (i.e., algorithm) and humans
  - use only humans? both? who's doing what?
- Recruiting models: Incentives
  - pay? volunteer?
- User interfaces
- Quality control
- Optimization: Volume and Speed

Crowdsourcing: Key Issues

- The role of machine (i.e., algorithm) and humans
  - use only humans? both? who's doing what?
- Recruiting models: Incentives
  - pay? volunteer?
- User interfaces
- Quality control
- **Optimization: Volume and Speed**

Turker Affinity and Errors



[Franklin, Kossmann, Kraska, Ramesh, Xin: CrowdDB: Answering Queries with Crowdsourcing. *SIGMOD*,2011]

Crowdsourcing Scalability

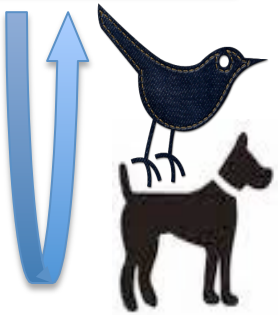
- **Machine Learning** algorithms take as input empirical data and figure out (*learn*) patterns in the mechanism that generated the data
- Crowd-sourcing is becoming an indispensible method of collecting labeled data, .e.g., Machine Learning
- BUT crowd-sourcing can be expensive, slow, and noisy
- All Human Intelligent Tasks (HIT) *are NOT equally difficult for the machine*
- To achieve scalability, we need to know when and how to use machines along with humans

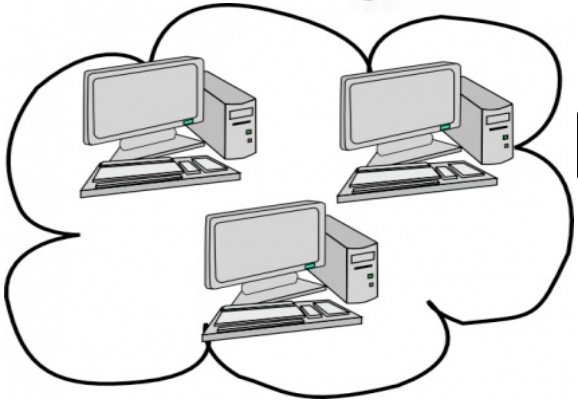
When to Ask Whom

Query

DB

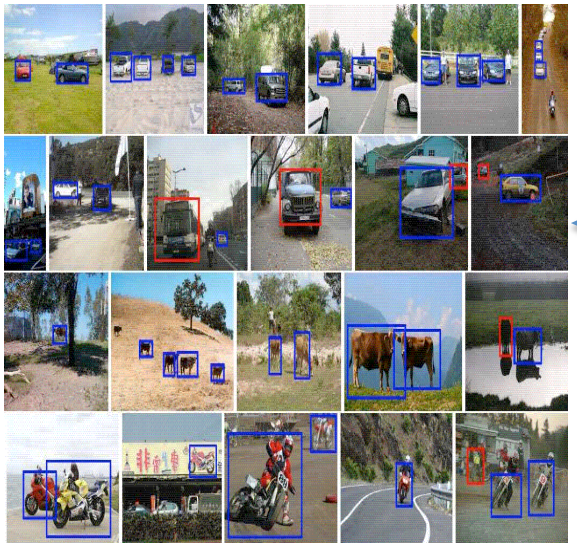
Crowd

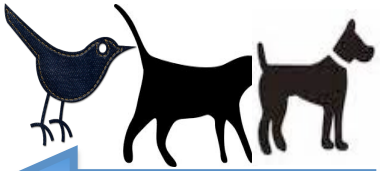




Machines

When to Ask Whom

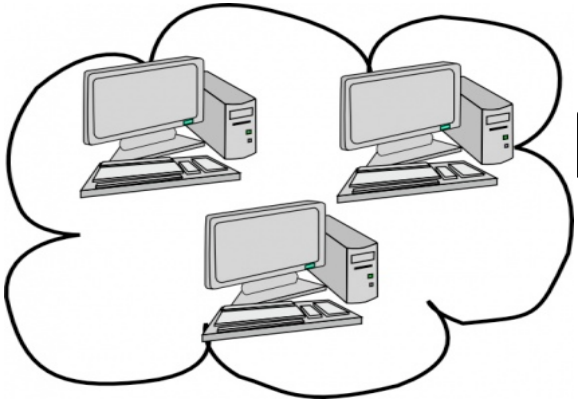




Answers

DB

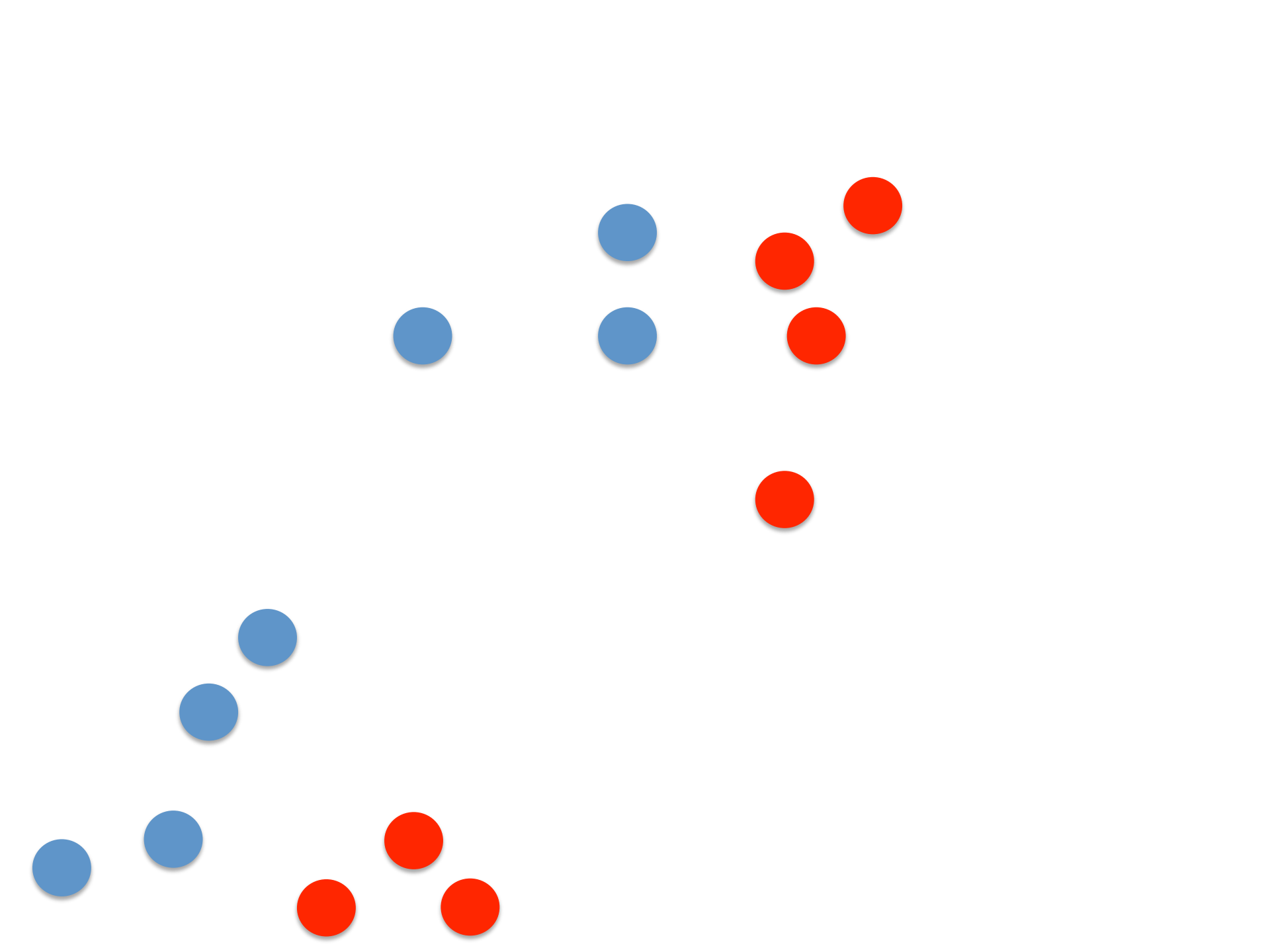
Crowd

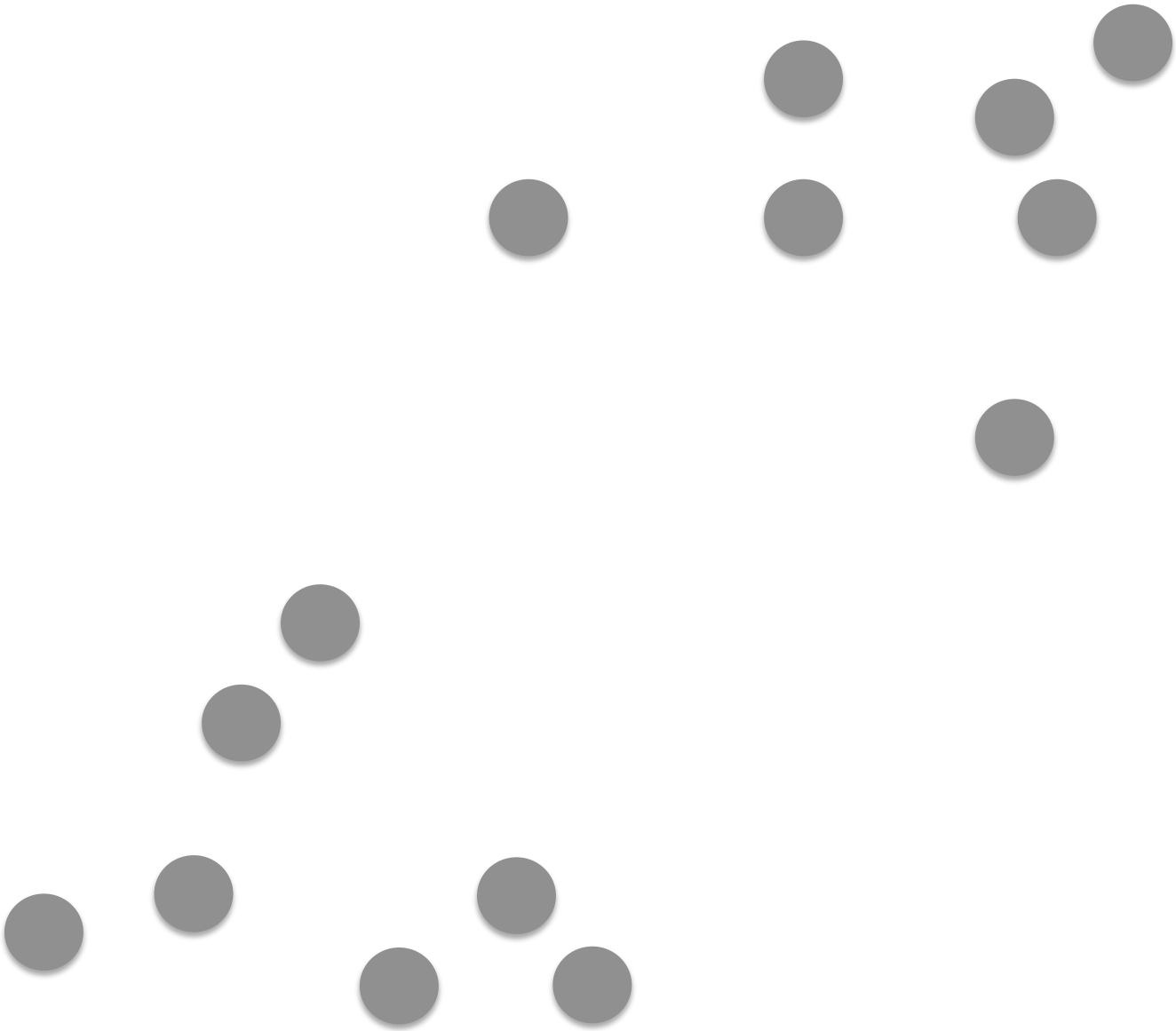


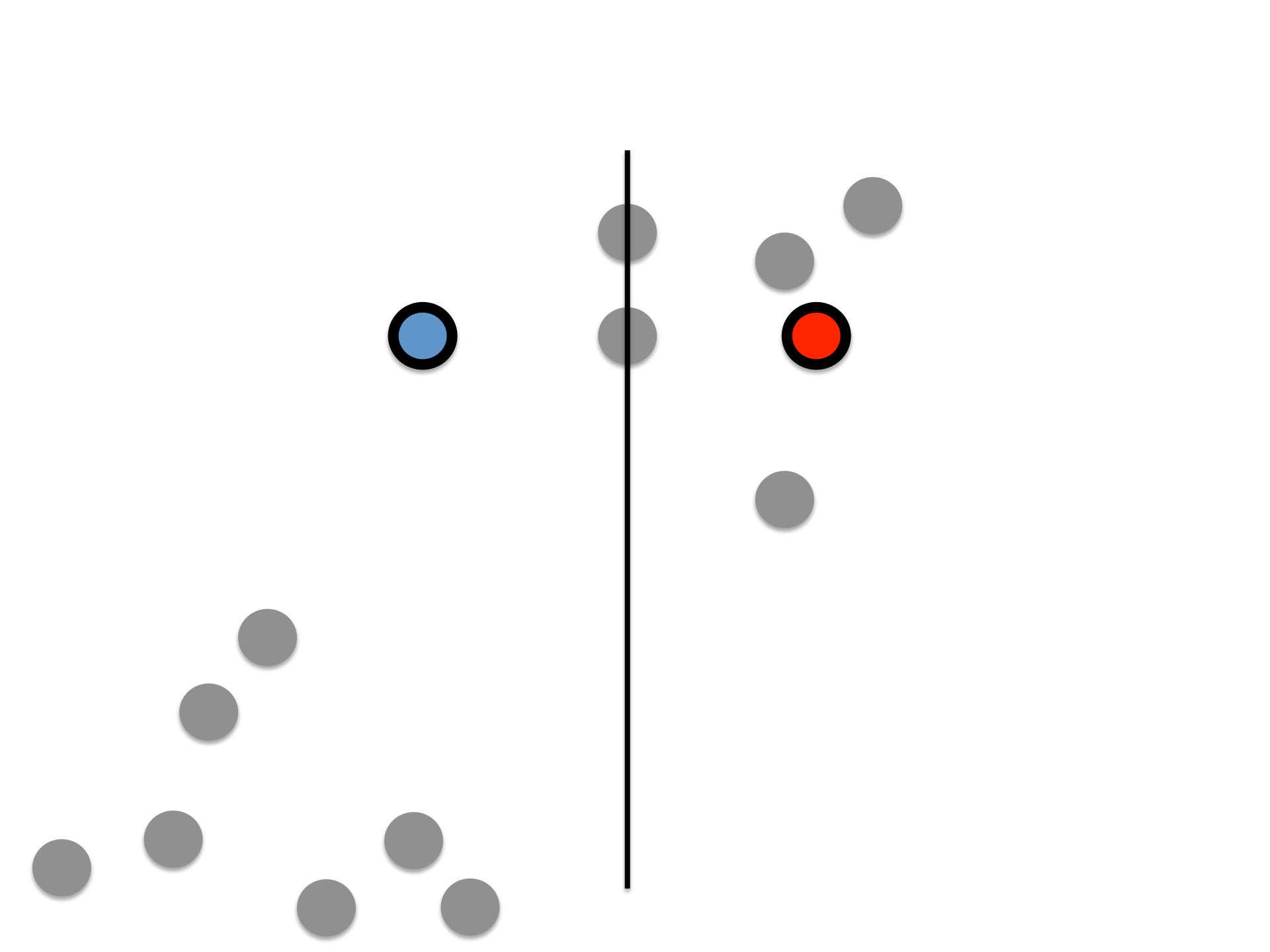
Machines

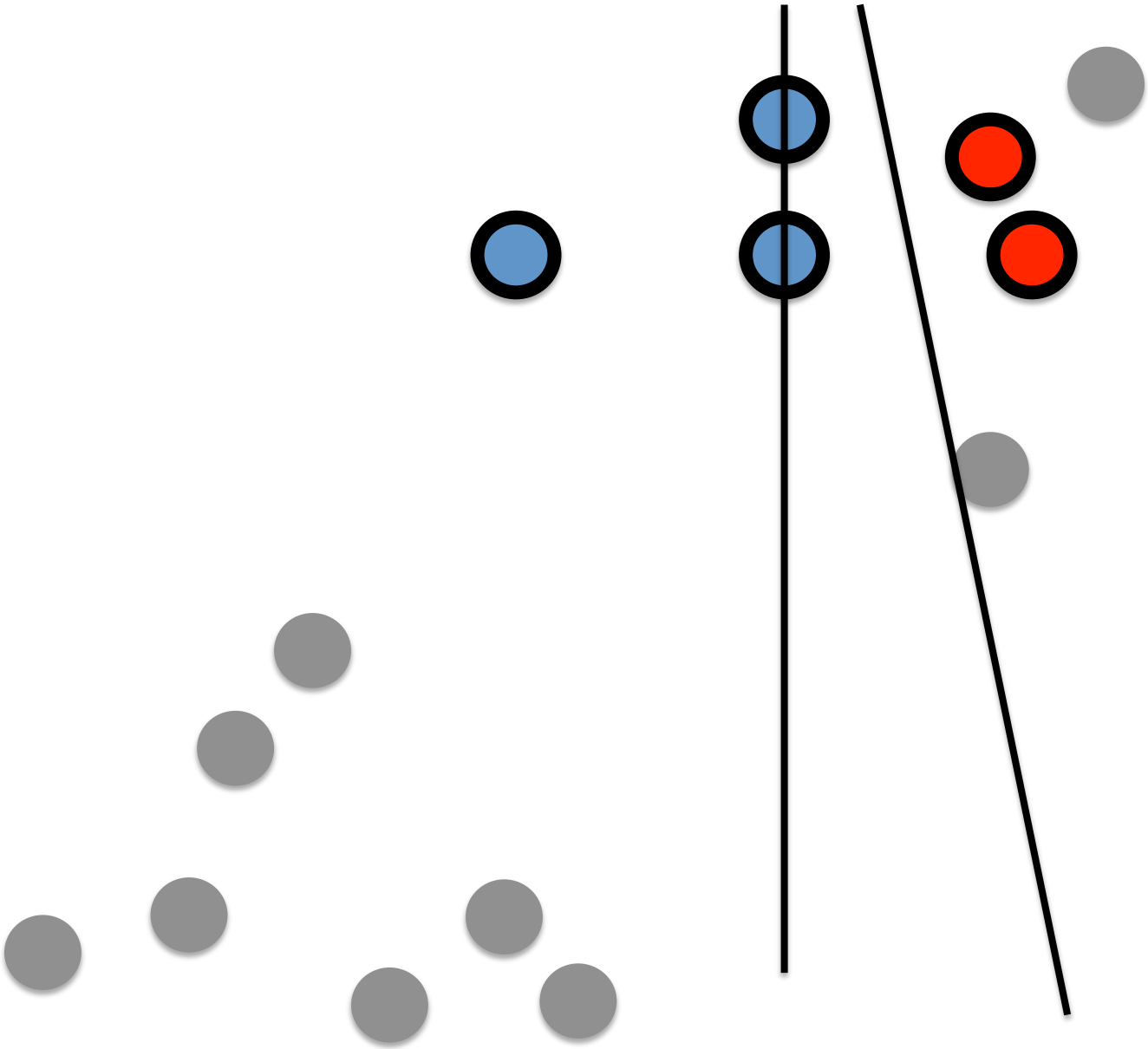
Active Learning

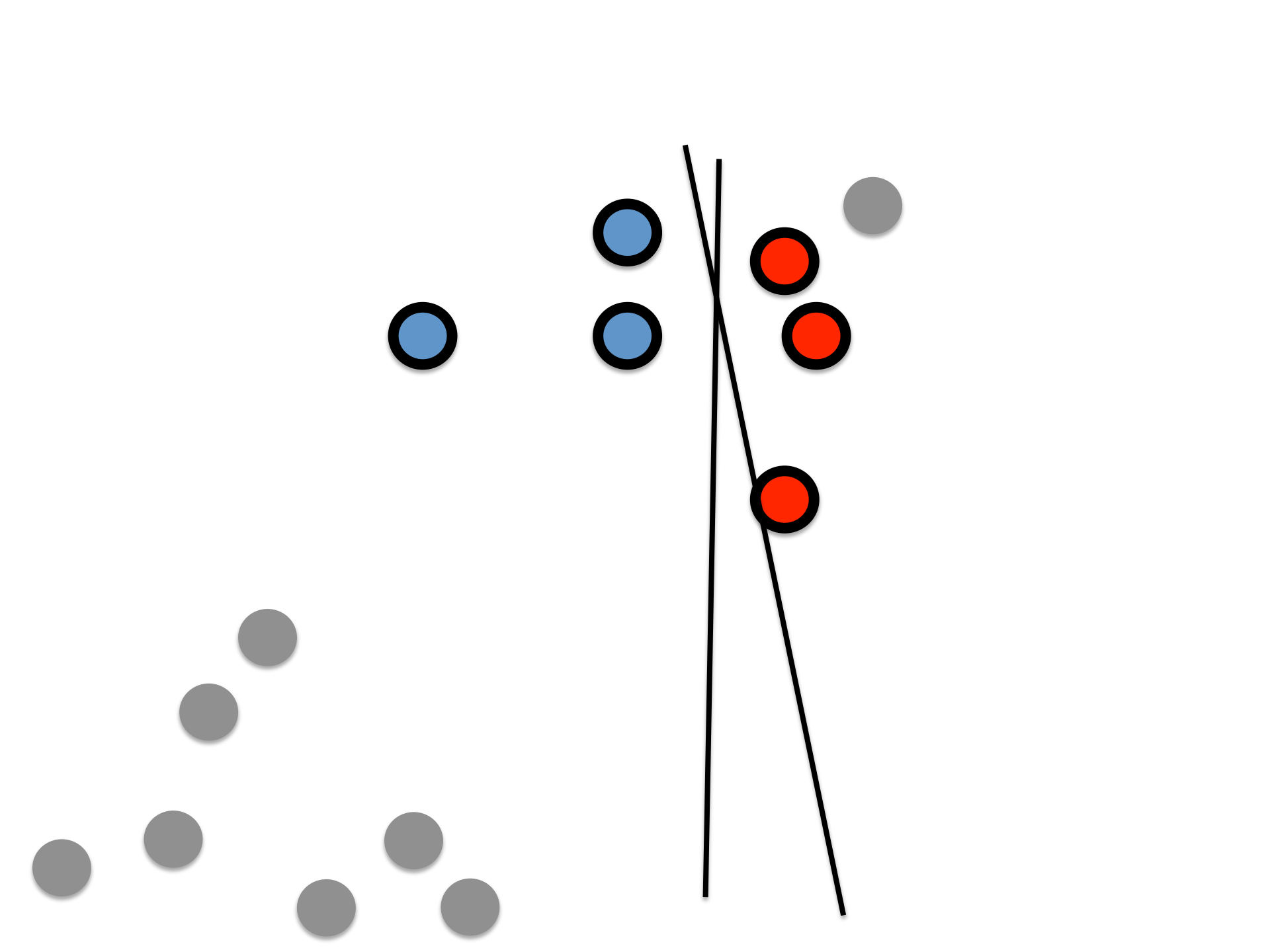
	Classical Active Learning	Active Learning with Crowd
Label Provider	Domain Experts	Crowd
Label's Accuracy	Perfect	With Error
Stopping Condition	Budget Exhaustion	Budget Exhaustion / Model's Accuracy Reaching Crowd's Accuracy
Goal	Maximizing Model's Accuracy	Maximizing Overall Accuracy

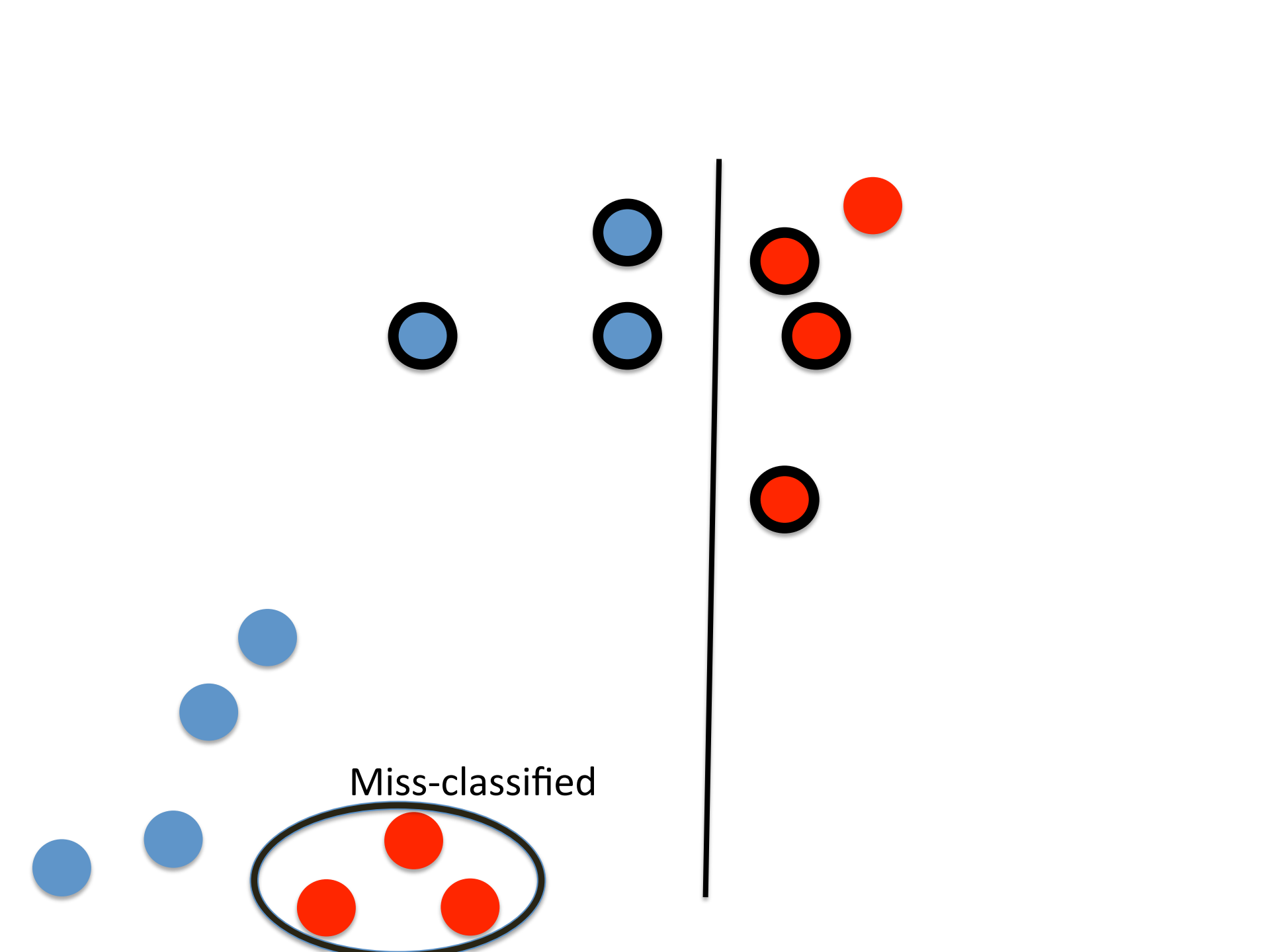


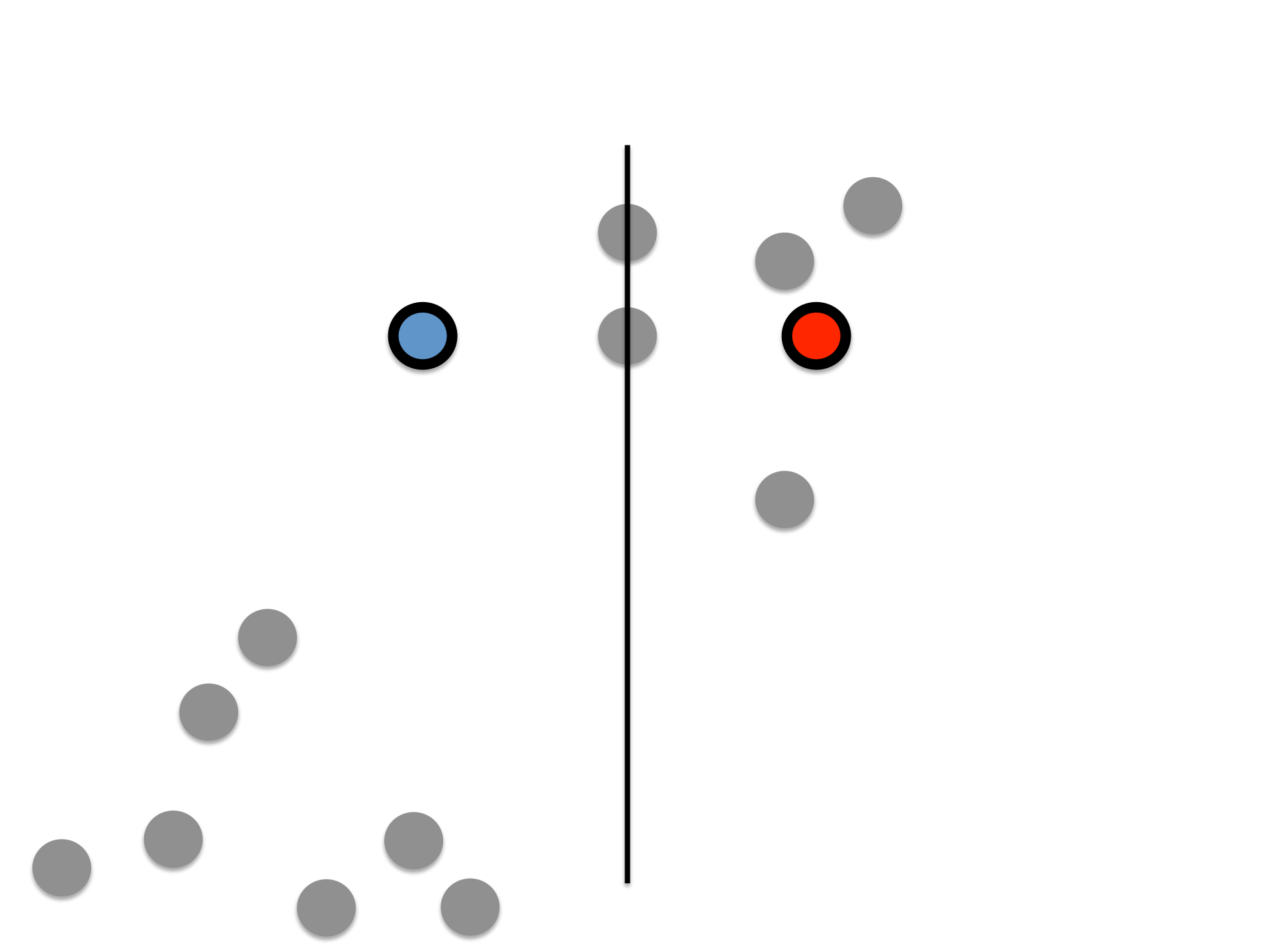


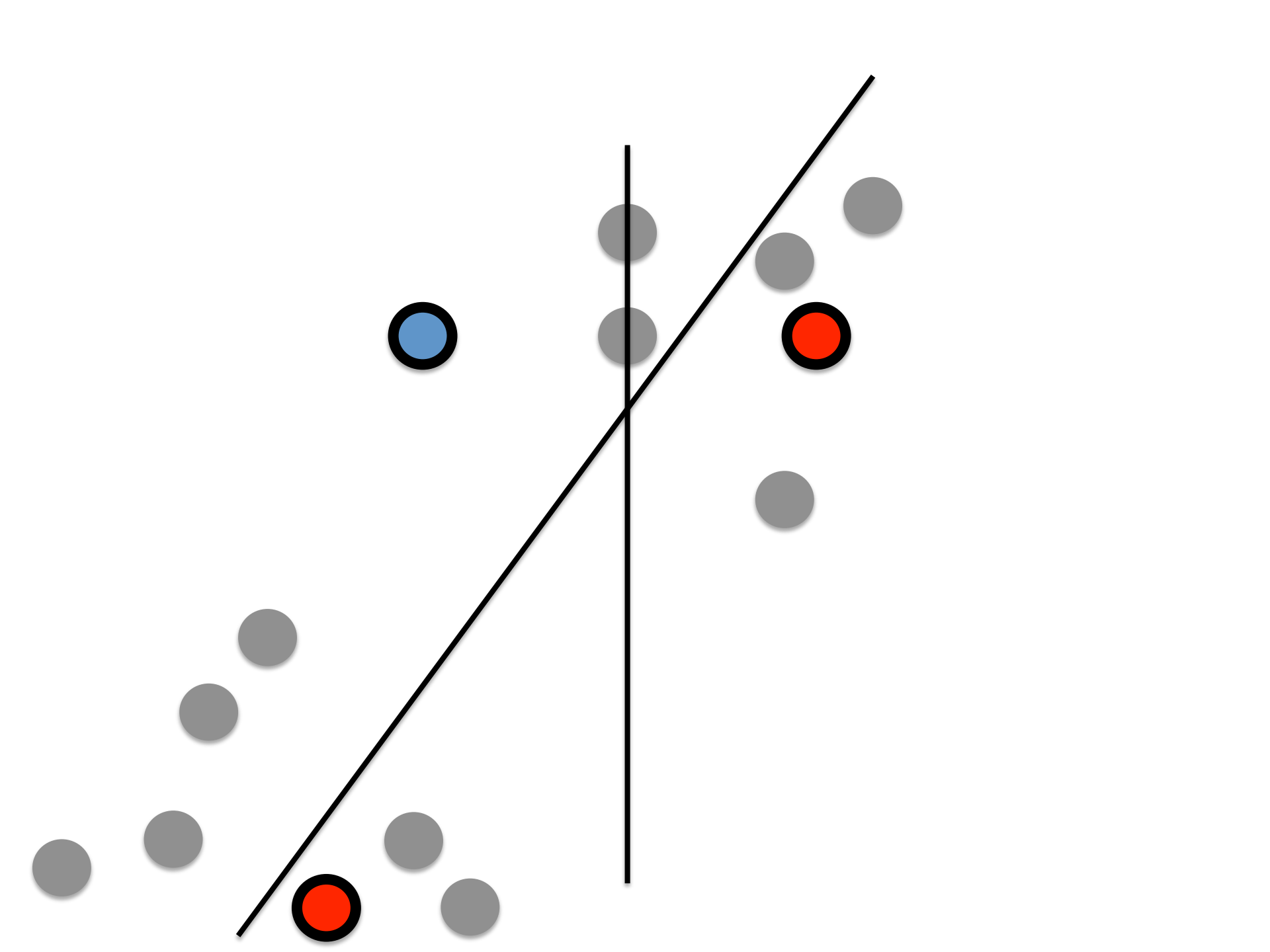


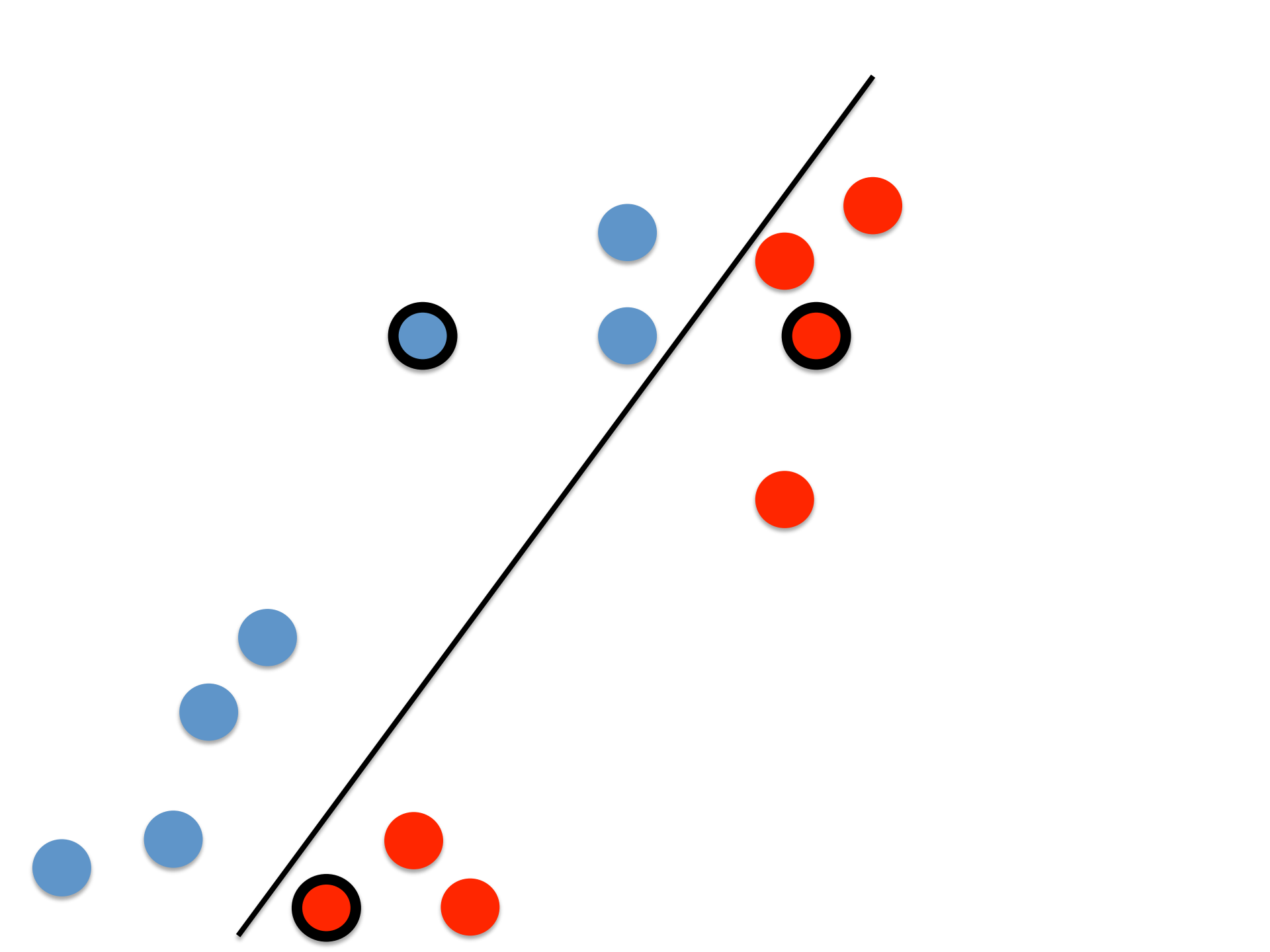





Miss-classified







Interaction Protocol

How often can we refer to the crowd?

1. **Upfront**: Ask all the B queries at once
2. **Iterative**: Ask K queries from the crowd and use them to improve the learner. Repeat this B/K times

Measures Used for Selection

- **Uncertainty:** Asking hardest (most ambiguous) questions
- **Explorer:** Ask questions with potential to have largest impact on the learner

Improving Crowdsourcing Latency

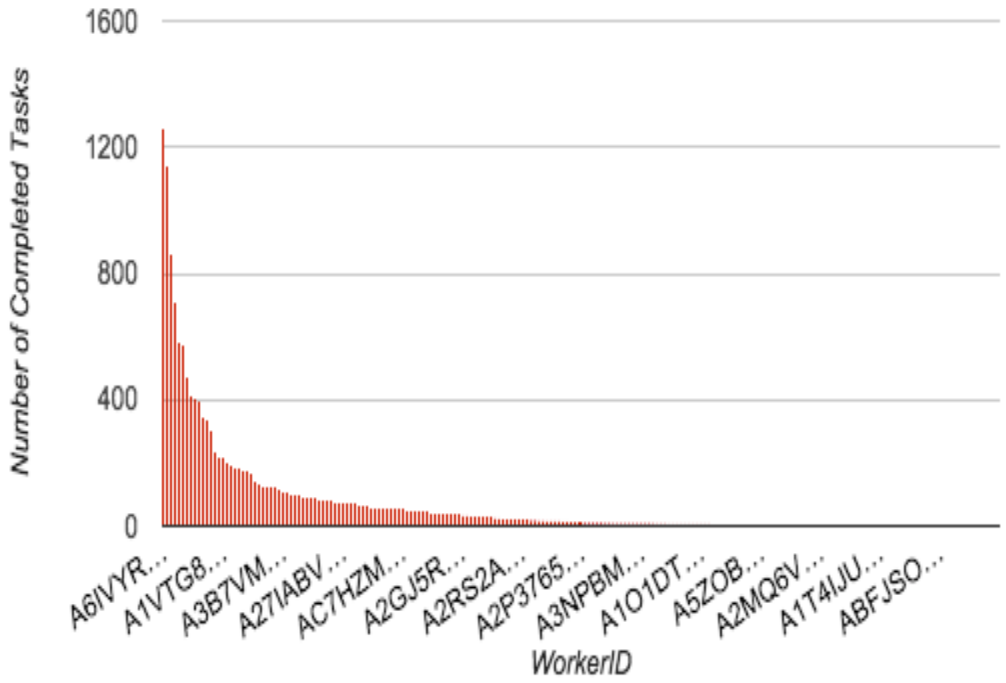
- Retaining workers on your batch
- Deciding which HITs should be done first

Pricing Schemes to Improve Worker Retention

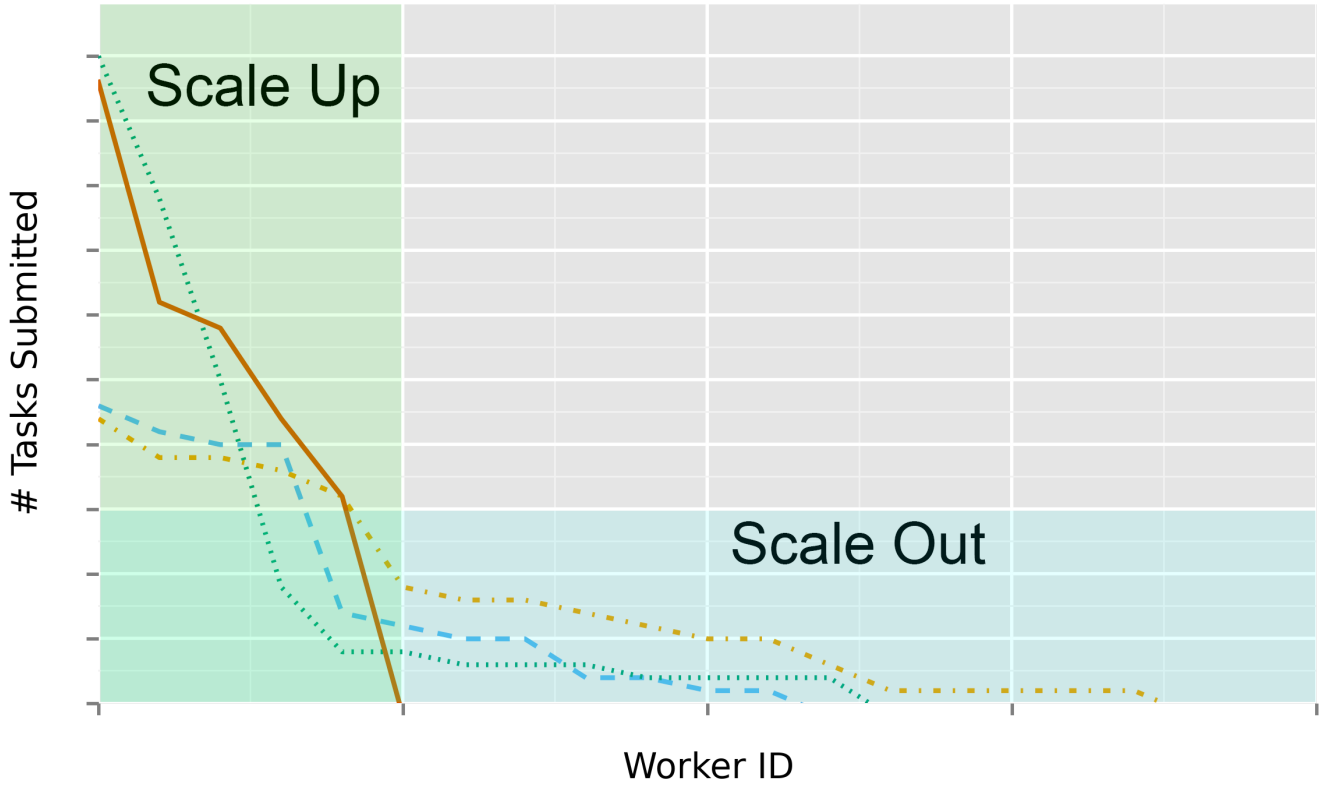
Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, and Philippe Cudré-Mauroux. **Scaling-up the Crowd: Micro-Task Pricing Schemes for Worker Retention and Latency Improvement**. In: Second AAAI Conference on Human Computation and Crowdsourcing (HCOMP-2014). Pittsburgh, USA, November 2014.

Workers VS Amount of Work

Assignments completed: 14964 (4988 tasks with 3 repetitions)
Time elapsed: 4:28:39 (h:mm:ss) / Average submit time: 20.0 seconds
228 Worker



Scale-up vs Scale-out



Scaling-up the Crowd (Definition)

Attaining higher **worker retention** rates such that they keep working longer on a given batch

This model potentially presents two advantages:

- It **minimizes the down times** incurred when waiting for new workers, and
- yields potentially **experienced workers** at handling a given task

Worker Retention - an HR practice

"Companies that do a better job of attracting, developing, exciting, and retaining their talent will gain more than their fair share of this critical and scarce resource and will boost their performance dramatically"

— Michaels et al., The War for Talent

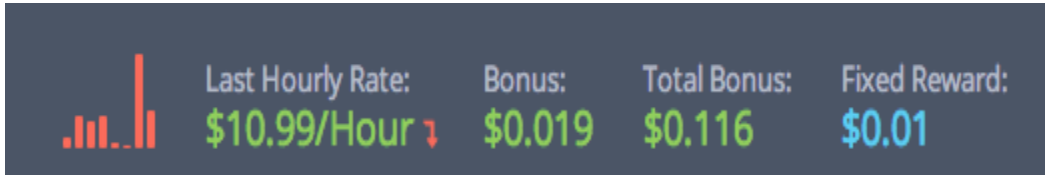
Micro Bonus Pricing Schemes

Given a fixed retention budget ***B*** allocated to pay workers $\{w_1,..w_n\}$ doing analogous tasks $\{h_1,..,h_K\}$

- Fixed Bonus
- Training (Descending) Bonus
- Increasing Bonus
- Random Bonus
- Milestone Bonus

The Bonus Bar

A visual cue for workers to track their own progress and the monetary impact of each pricing scheme



Milestone version

Experiments

Setup I

Tasks

- Item Matching
- Butterfly classification
- Customer Care Phone Number Search

Goal

- Observe the impact of our different pricing schemes on the total number of tasks completed by the workers in a batch (worker retention)

Implementation

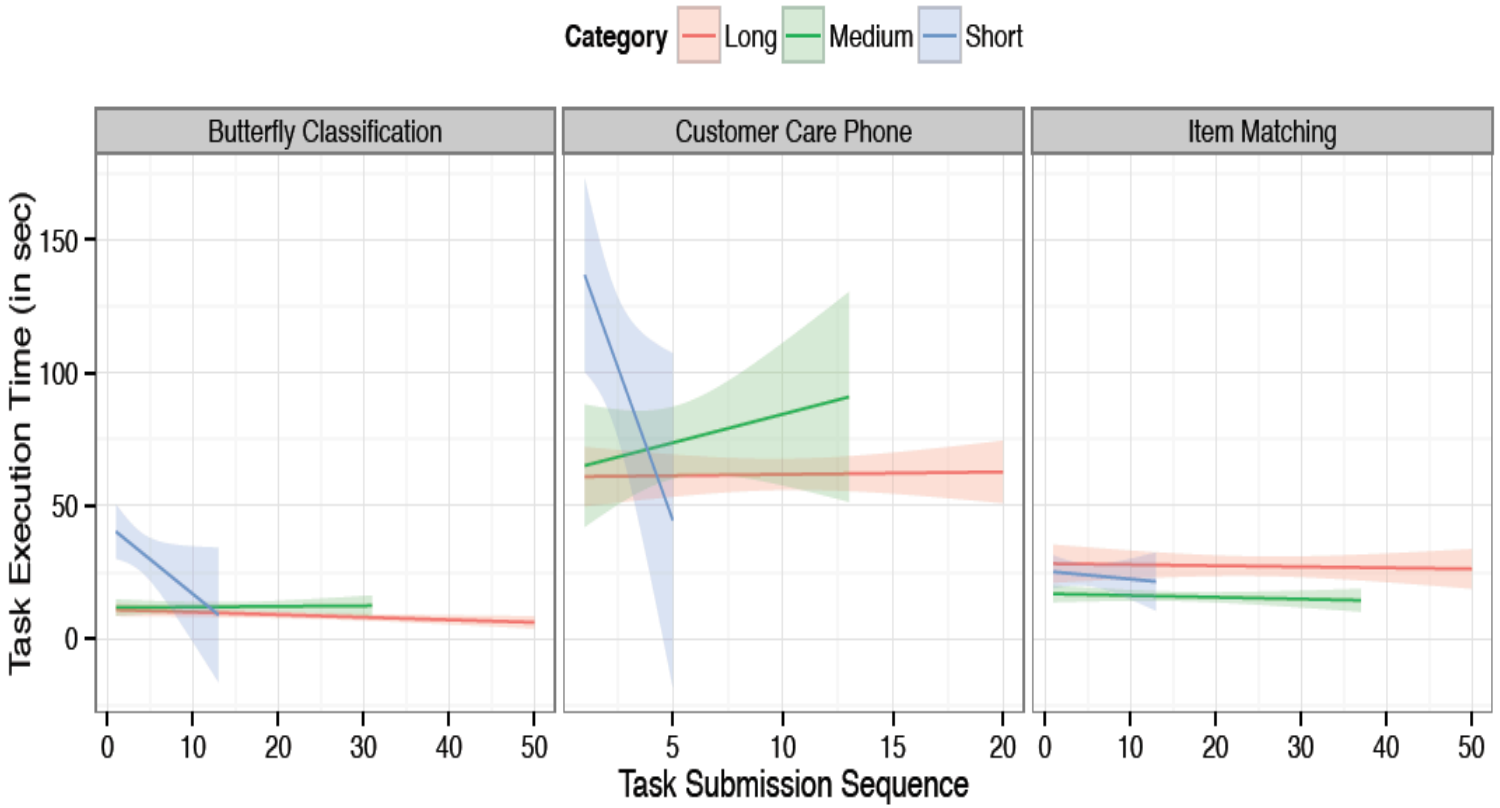
- Recruit exactly 50 distinct workers, and enabling 50 repetitions for all the tasks.

Worker Retention

Milestone Bonus is the winner across the 3 task types

*Workers are ordered by the number of completed HITs

Learning Curve

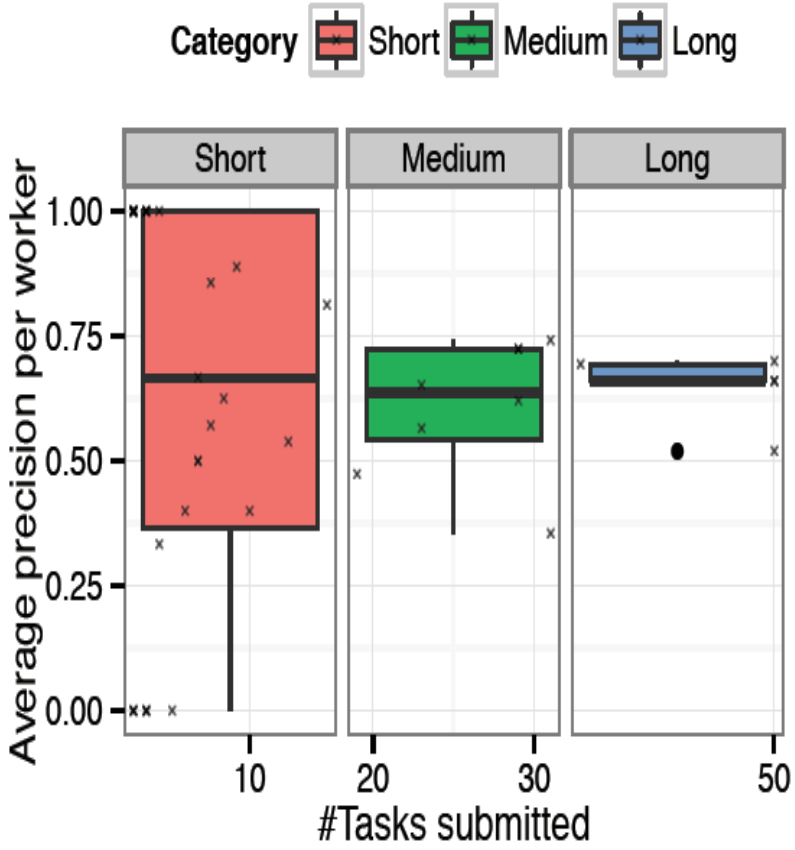


Retained workers seem to have slightly better execution time (also depending on the task)

Impact on Work Quality

Although retained workers did more tasks, their precision is similar to other classes of workers

*Butterfly Classification task (using Increasing Bonus)



Experiments

Setup II (non-controlled)

Goal

- Evaluate the impact of our best approach (Milestone Bonus) on the end-to-end execution of a batch of HITs.

Tasks

- Correcting 10 english essays from the ESOL with 3 repetitions each

Details

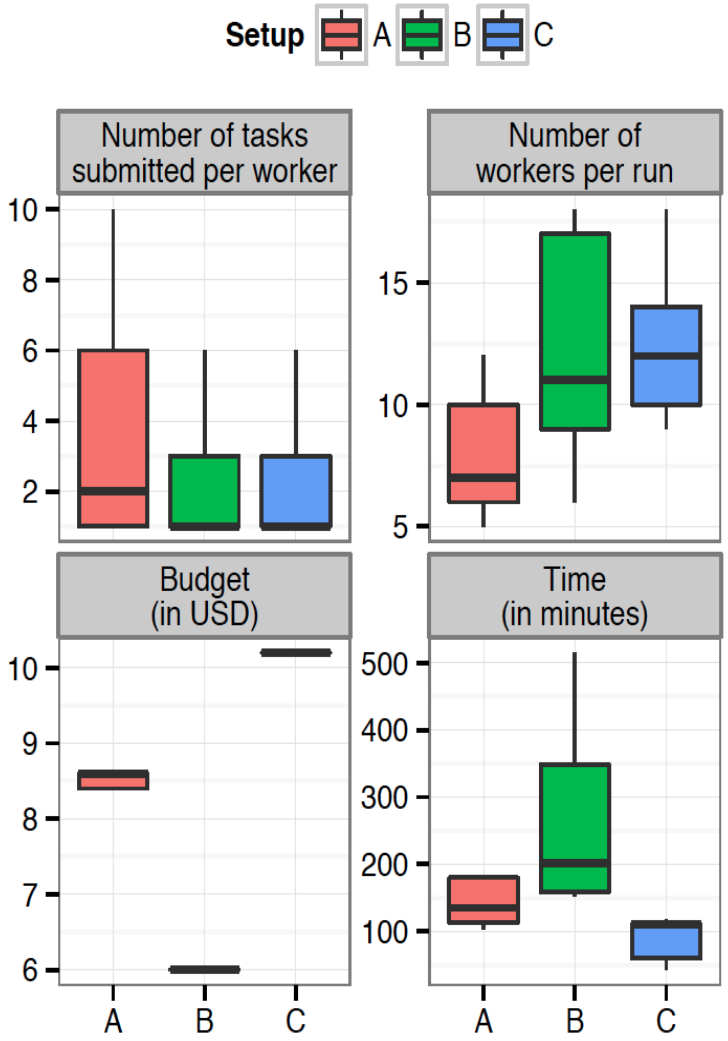
- Run 3 batches featuring Milestone Bonus, Uniform Bonus, No-bonus
- Reward, title and description are the same for the three batches and also “worker-exclusive” (A/B testing)
- Repeat the process 5 times while measuring beginning/end of each batch

Efficiency

A involves less workers who submit a greater number of HITs on average as compared to batches **B** and **C**

While **C** is faster overall, it uses the entirety of its budget, as compared to **A** that only uses $2.44 on average

A(Milestone), B(No Bonus), C(High Reward)



Take Away

- Avoid batch starvation in a competitive market.
- Giving workers a punctual bonus for reaching a predefined objective defined as a given number of tasks improves worker retention.
- While retention comes at a cost, it also improves latency. Which makes it a new dimension to explore when running critical jobs.

Summary

- To make crowdsourcing scalable over large amounts of data
  - Use machine-based algorithms: build hybrid systems
  - Select the right data sample to crowdsource
  - Keep workers on your tasks