

Sequential Crowdsourced Labeling

Vikas C. Raykar

IBM India Research Lab

Organization and References

✓ Crowdsourced labelling

[Learning From Crowds](#)

Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo H. Valadez, Charles Florin, Luca Bogoni, and Linda Moy

Journal of Machine Learning Research, Vol. 11, pp. 1297–1322, April 2010.

✓ Sequential crowdsourced labeling

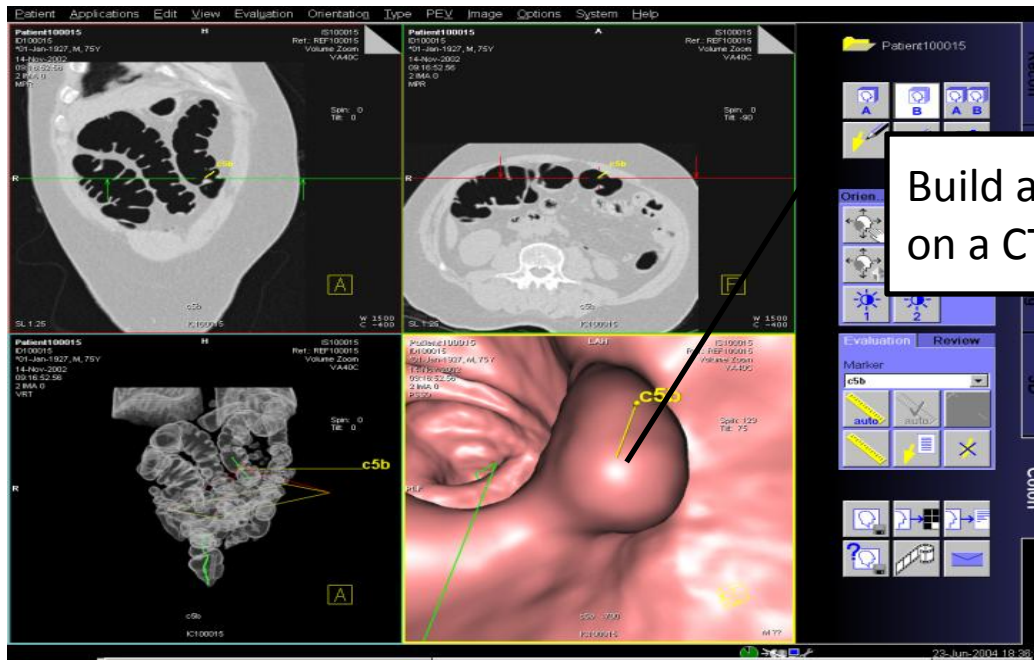
[Sequential crowdsourced labeling as an epsilon-greedy exploration in a Markov Decision Process](#) †

Vikas C. Raykar and Priyanka Agrawal

Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 832–840, Reykjavik, Iceland, 2014.

A motivating example

Computer aided diagnosis for colon cancer



Build a model to predict whether a region on a CT scan is cancer (1) or not (0)

Instance $x_i \in \mathbf{R}^d$	Label $y_i \in \mathcal{Y} = \{0, 1\}$
x_1	1
x_2	0
x_3	0
x_4	1
⋮	⋮
⋮	⋮
x_N	1

Formulate it as a supervised binary classification problem.

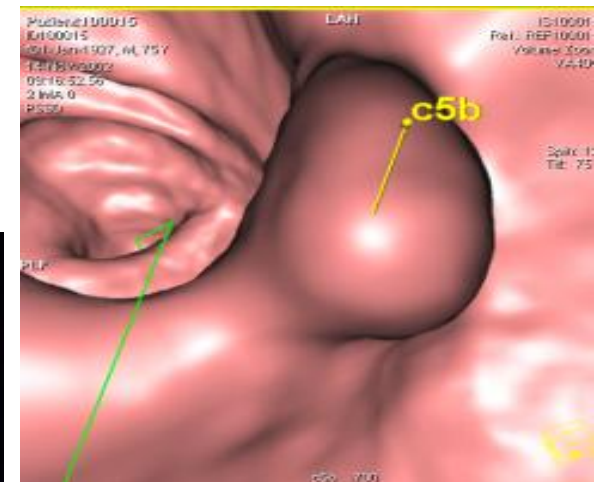
Collect a reasonably large labeled training set.

Learn a classifier $f : \mathbf{R}^d \rightarrow \mathcal{Y}$ which generalizes well on unseen data

Objective labels GOLD STANDARD

- How do we acquire the labels for training ?
 $y_i \in \mathcal{Y} = \{0, 1\}$
- However getting objective labels can be
 - Expensive
 - Tedious
 - Invasive
 - Sometimes potentially dangerous

Objective labels can be reliable obtained only by a biopsy of the tissue



Subjective labels APPROXIMATE GOLD STANDARD

- Acquiring objective annotations is hard.
- So we use opinion from an expert (radiologist)
- A radiologist visually examines the image and provides a subjective version of the truth.
- An expert provides his/her version of the truth and hence error prone.
- So we use multiple experts who label the same example.

Annotations from multiple experts

Each radiologist is asked to annotate whether a lesion is malignant (1) or not (0).

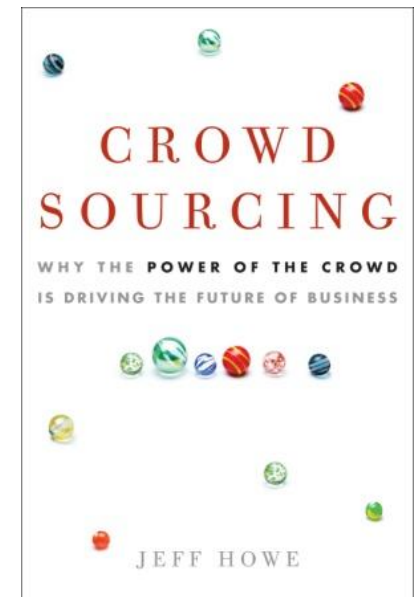
Lesion ID	Radiologist 1	Radiologist 2	Radiologist 3	Radiologist 4	Truth unknown
12	0	0	0	0	x
32	0	1	0	0	x
10	1	1	1	1	x
11	0	0	1	1	x
24	0	1	1	1	x
23	0	0	1	0	x
40	0	1	1	0	x

Each radiologist is not perfect.
In practice there is a substantial amount of disagreement.

How do we consolidate the multiple annotations ?
How do we evaluate the experts?

Crowdsourcing labeling tasks

- Accurate experts can be expensive and time-consuming.
- Why not use a large group of people who are not necessarily experts ?
- Can make the annotation process
 - Cheap
 - Fast
 - Reasonably accurate



- Crowd sourcing internet market place
- **HIT** Human Intelligence Task
 - **Requestor** (submits a labeling task)
 - **Worker** (receives a monetary payment)

amazonmechanical turk Artificial Intelligence

Your Account | HITs | Qualifications

Introduction | Dashboard | Status | Account Settings

Already have an account? Sign in as a Worker | Requester

Mechanical Turk is a marketplace for work.
 We give businesses and developers access to an on-demand, scalable workforce. Workers select from thousands of tasks and work whenever it's convenient.
54,637 HITs available. [View them now.](#)

Make Money by working on HITs

HITs - Human Intelligence Tasks - are individual tasks that you work on. [Find HITs now.](#)

As a Mechanical Turk Worker you:

- Can work from home
- Choose your own work hours
- Get paid for doing good work

Find an interesting task → Work → Earn money

[Find HITs Now](#)

[or learn more about being a Worker](#)

Get Results from Mechanical Turk Workers

Ask workers to complete HITs - Human Intelligence Tasks - and get results using Mechanical Turk. [Register Now](#)

As a Mechanical Turk Requester you:

- Have access to a global, on-demand, 24 x 7 workforce
- Get thousands of HITs completed in minutes
- Pay only when you're satisfied with the results

Fund your account → Load your tasks → Get results

[Get Started](#)

[or learn more about being a Requester](#)

Sample NLP annotation on AMT

https://workersandbox.mturk.com/mturk/preview?groupId=25FXP3TVOK5UQ6084WHERZGZQU125U

You are using the Mechanical Turk Developer Sandbox. This site is for test and development only. [Learn more >](#)

amazonmechanical turk Artificial Intelligence

Your Account HITS Qualifications 260,803 HITS available now

vikas c raykar | Account Settings | Sign Out | Help

All HITS | HITS Available To You | HITS Assigned To You

Find HITS containing that pay at least \$ 0.00 for which you are qualified require Master Qualification

Timer: 00:00:00 of 60 minutes

Want to work on this HIT? Want to see other HITS?

Total Earned: \$9.57
Total HITS Submitted: 212

2013-06-28_train_comparative_annotation.tsv
Requester: vikas c raykar
Qualifications Required: None
Reward: \$0.05 per HIT HITS Available: 25 Duration: 60 minutes

Is the sentence below comparative or non-comparative ?

A comparative sentence expresses a relation based on similarities or differences of more than one object. They generally use an indicator word: words ending with -er, -est, more, most, less, least, exceed, outperform, similar, etc. Your task is to label each of the sentences below as either comparative/non-comparative and mark the indicator word (select using your mouse and it will get copied to the textbox below once you release the mouse). For example *John is taller than Tom.* is a comparative sentence and the indicator word is *taller*. If a sentence is badly formed choose the option 'bad sentence'. Not all sentences with these indicator words are comparisons. Some comparative sentences do not use any indicator words.

The ad was the first in a campaign to dispute reports that smoking cigarettes could cause lung cancer and had other dangerous health effects [REF].

comparative non-comparative bad sentence indicator word/phrase

The case gained worldwide attention and is often said, inaccurately,

comparative non-comparative bad sentence indicator w

The cable, known as Reykjavik 13 was the first of the classified

comparative non-comparative bad sentence indicator w

The second of these maxims has become known as the social a

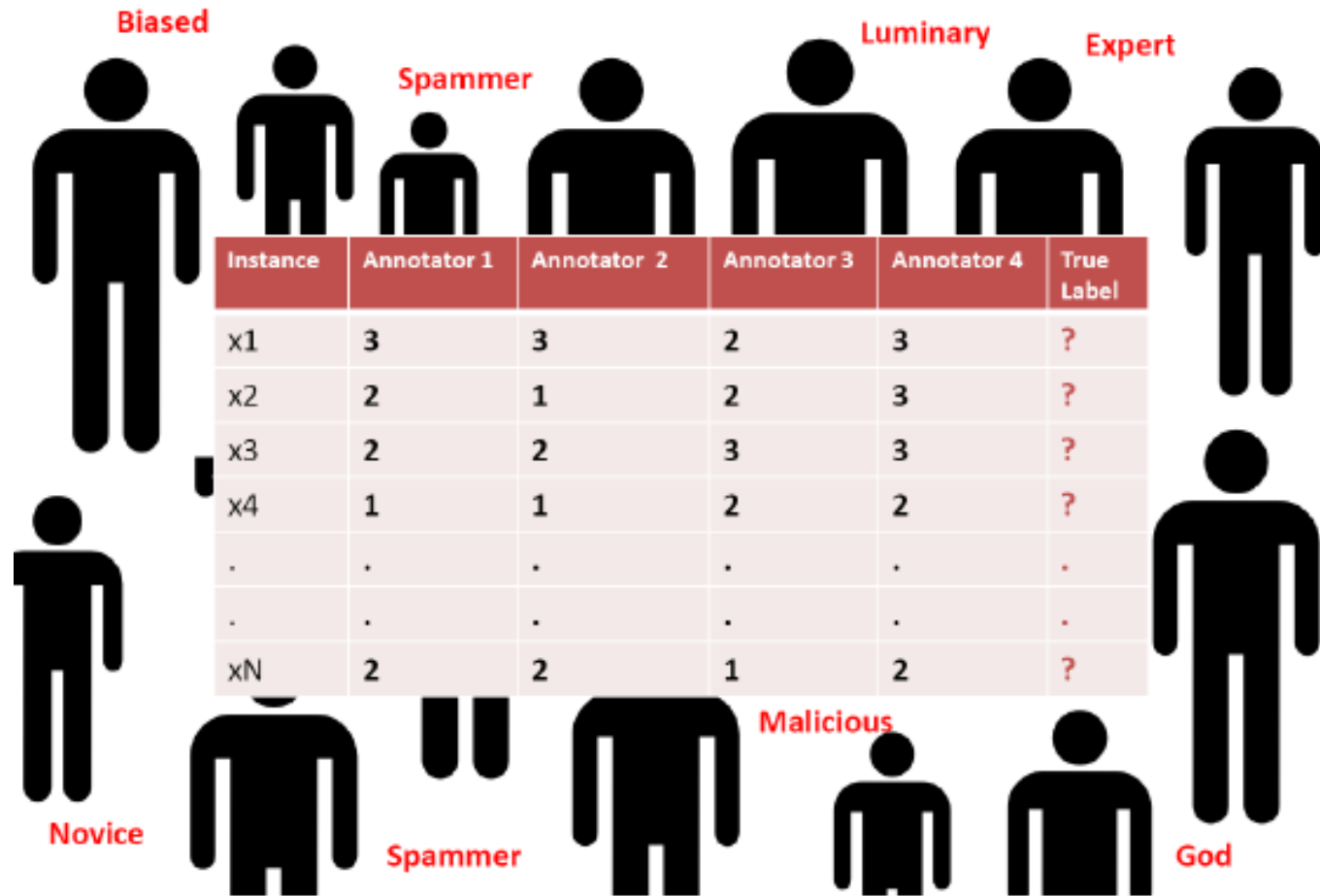
comparative non-comparative bad sentence indicator w

- Possibly thousands of annotators.
- Some are genuine experts.
- Most of novices.
- Some may be even malicious
- Without the gold standard how do we know?

Organization

- **Crowdsourced data annotation**
 - Multiple experts
 - Workers on crowd sourcing marketplaces
- **Consolidating multiple annotations**

Consolidating multiple annotations



How do we consolidate the multiple annotations ?
How do we evaluate the annotators?

Majority Voting

- Use the label on which most of them agree as an estimate of the truth.

Soft majority voting

ID	R1	R2	R3	R4	Truth	Majority	Pr [label=1]
12	0	0	0	0	x	0	0.00
32	0	1	0	0	x	0	0.25
10	1	1	1	1	x	1	1.00
11	0	0	1	1	x	?	0.50
24	0	1	1	1	x	1	0.75
23	0	0	1	0	x	0	0.25
40	0	1	1	0	x	?	0.50

What is wrong with majority voting?

- The problem is that it is just a majority.
- Assumes all experts are equally good.
- What if majority of them are bad and only one annotator is good?

Breast MR example	R1	R2	R3	Label from biopsy	Majority Voting
10	1	1	0	0	1
22	1	1	0	0	1

FIX : Give more importance to the expert you trust ? (weighted majority vote)

PROBLEM : How do we know which expert is good?

For that we need the actual ground truth ?

Chicken-and-egg problem

We need to model and correct for annotator biases

Annotator model

Sensitivity

$$\alpha^j := \Pr[y^j = 1 | y = 1]$$

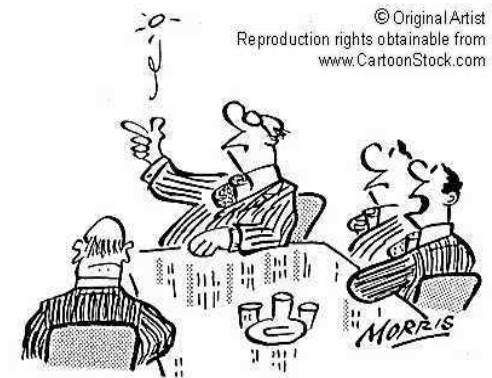
Label assigned by expert j

True Label

Specificity

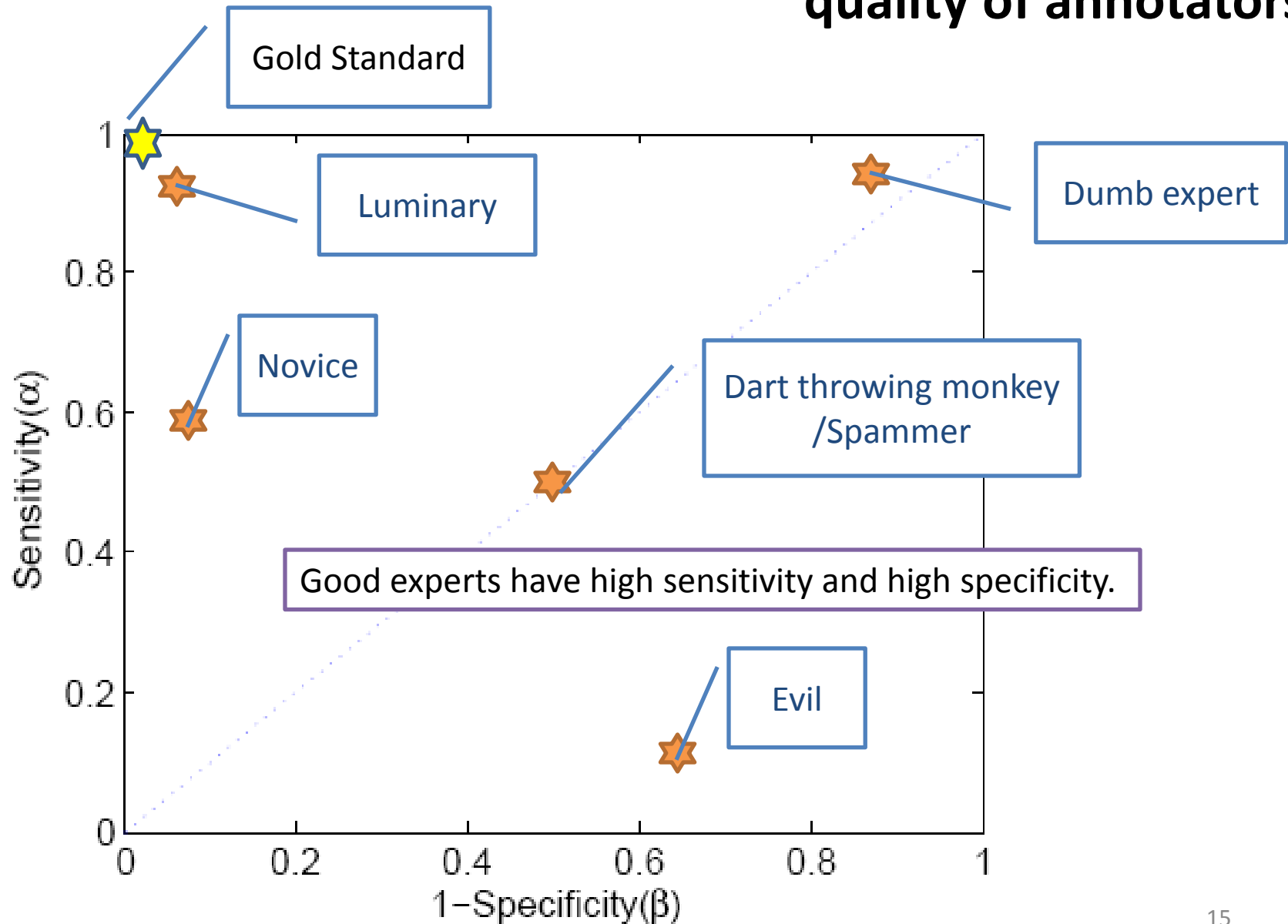
$$\beta^j := \Pr[y^j = 0 | y = 0]$$

An annotator with two coins



"I wish I could be as calm as JB when it comes to making decisions."

In crowdsourcing marketplaces we have no control over the quality of annotators



If we know the annotator parameters how do we estimate the true labels?

$$\mu_i = \Pr[y_i = 1 | \underbrace{y_i^1, \dots, y_i^R}_{\text{Observed labels From R annotators}}, \underbrace{\alpha, \beta}_{\text{Annotator Parameters}}]$$

Bayes Rule

$$\mu_i \propto \underbrace{\Pr[y_i^1, \dots, y_i^R | y_i = 1, \alpha, \beta]}_{\text{Likelihood}} \cdot \underbrace{\Pr[y_i = 1]}_{\text{Prevalence } p}$$

Conditional on the true label we assume the radiologists make their decisions independently.

$$\Pr[y_i^1, \dots, y_i^R | y_i = 1, \alpha] = \prod_{j=1}^R \Pr[y_i^j | y_i = 1] = \prod_{j=1}^R [\alpha^j]^{y_i^j} [1 - \alpha^j]^{1 - y_i^j}$$

So if someone provided me with the true sensitivity and specificity (and also the prevalence) for each annotator I could estimate the true label as

$$\mu_i = \frac{p \prod_{j=1}^R [\alpha^j]^{y_i^j} [1 - \alpha^j]^{1 - y_i^j}}{p \prod_{j=1}^R [\alpha^j]^{y_i^j} [1 - \alpha^j]^{1 - y_i^j} + (1 - p) \prod_{j=1}^R [\beta^j]^{1 - y_i^j} [1 - \beta^j]^{y_i^j}}$$

Why is this useful ? We really do not know the sensitivity, specificity, or the prevalence.

If we know the true labels how do we estimate the annotator parameters?

We can compute the sensitivity and specificity of each annotator

Sensitivity

$$\alpha^j := \Pr[y^j = 1 | y = 1]$$

Specificity


$$\beta^j := \Pr[y^j = 0 | y = 0]$$

$$\alpha^j = \frac{\sum_{i=1}^N y_i y_i^j}{\sum_{i=1}^N y_i} \quad \beta^j = \frac{\sum_{i=1}^N (1 - y_i)(1 - y_i^j)}{\sum_{i=1}^N (1 - y_i)}$$

Instead of a hard label (0 or 1)

If I had a soft label (probability that the label is 1)

Sensitivity and specificity with soft labels


$$\alpha^j = \frac{\sum_{i=1}^N \mu_i y_i^j}{\sum_{i=1}^N \mu_i} \quad \beta^j = \frac{\sum_{i=1}^N (1 - \mu_i)(1 - y_i^j)}{\sum_{i=1}^N (1 - \mu_i)}$$

The chicken and egg problem

If I knew the true label I can estimate how good each annotator is

M-step

$$\alpha^j = \frac{\sum_{i=1}^N \mu_i y_i^j}{\sum_{i=1}^N \mu_i} \quad \beta^j = \frac{\sum_{i=1}^N (1 - \mu_i)(1 - y_i^j)}{\sum_{i=1}^N (1 - \mu_i)} \quad p = \frac{1}{N} \sum_{i=1}^N \mu_i$$

The algorithm can be rigorously derived by writing the likelihood.

We can find the maximum-likelihood (ML) estimate for the parameters.

The log-likelihood can be maximized using an EM algorithm

The actual labels are the missing data for EM algorithm.

If I knew

Dawid and Skeene 1979, Raykar et al JMLR 2010

E-step

$$\mu_i = \frac{p \prod_{j=1}^R [\alpha^j]^{y_i^j} [1 - \alpha^j]^{1 - y_i^j}}{p \prod_{j=1}^R [\alpha^j]^{y_i^j} [1 - \alpha^j]^{1 - y_i^j} + (1 - p) \prod_{j=1}^R [\beta^j]^{1 - y_i^j} [1 - \beta^j]^{y_i^j}}$$

A few extensions

- **Bayesian approaches**
 - Raykar et al JMR 2010, Wauthier et al NIPS 2012
- **Variation Bayes approach**
 - Liu et al NIPS 2012
- **Modeling task complexity**
 - Welinder et al NIPS 2010, Whitehill et al NIPS 2010
- **Missing labels**
 - Raykar et al JMR 2010,
- **Categorical, ordinal, and continuous annotations**

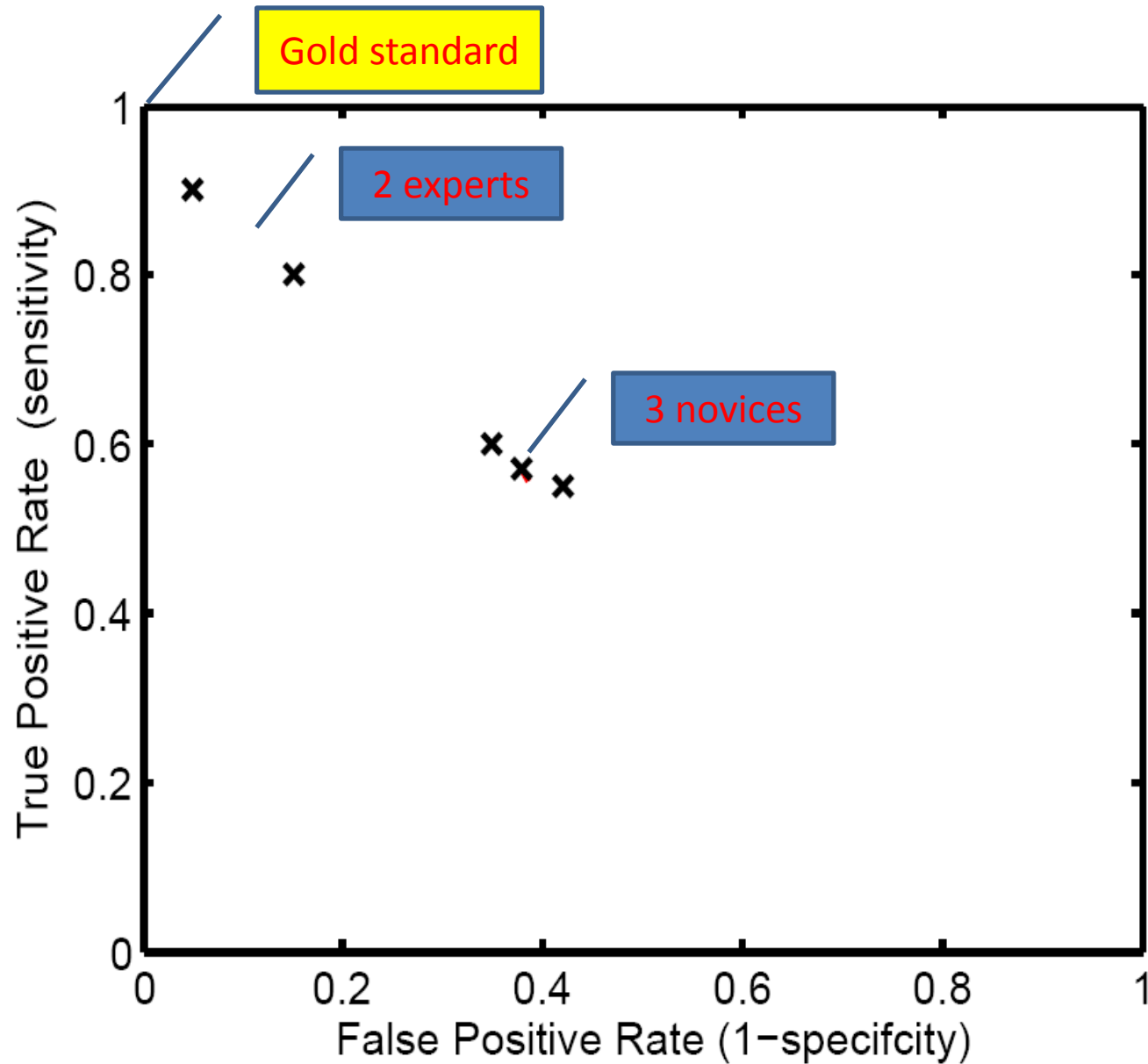
Experimental validation

Domain	Gold standard	Number of annotators	Number of positives	Number of negatives
Digital Mammography	Available (Biopsy)	5	497	1618

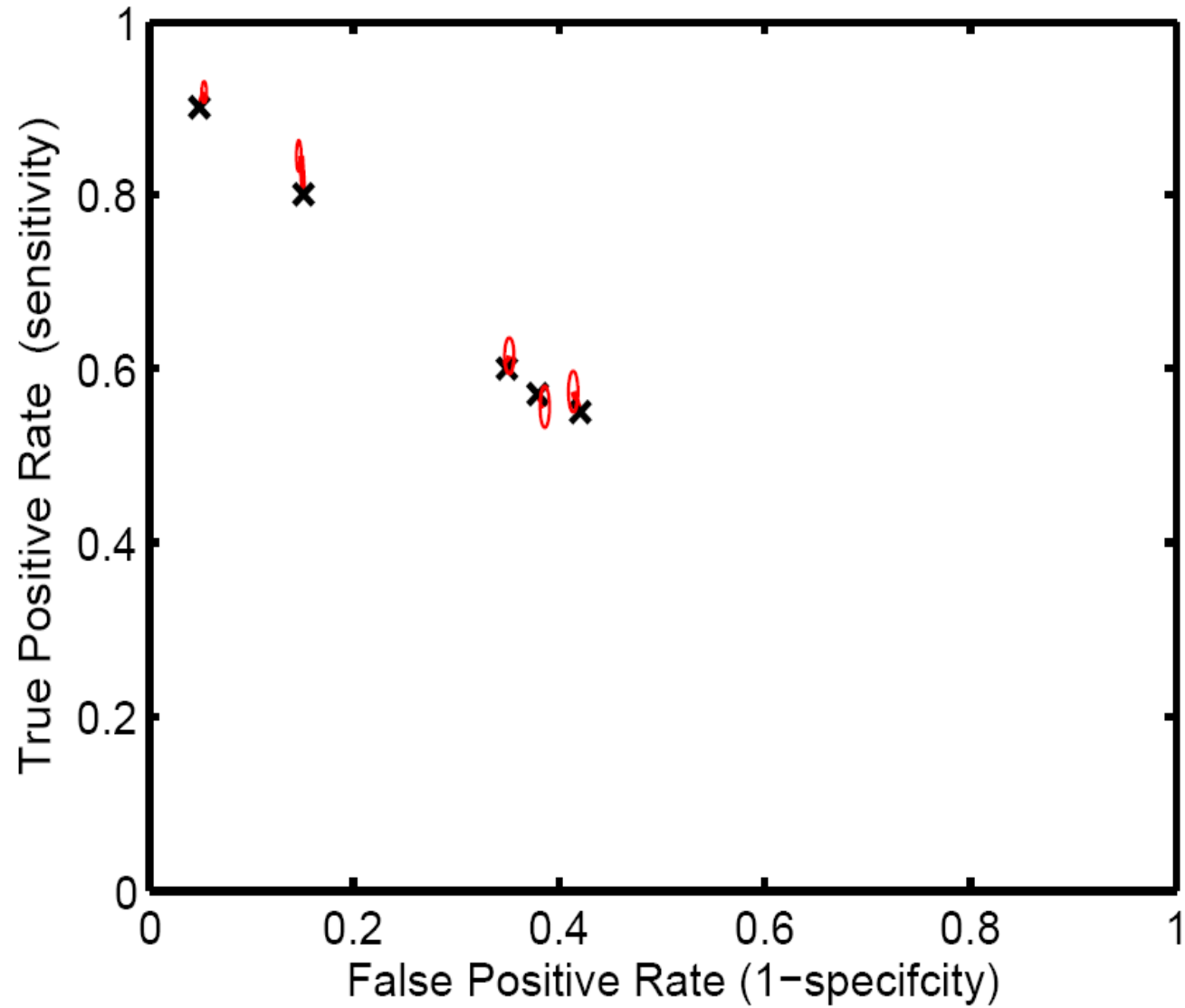
1. How well can you estimate the annotator performance?
2. How well can you estimate the actual ground truth ?

1. Proposed EM algorithm
2. Majority Voting

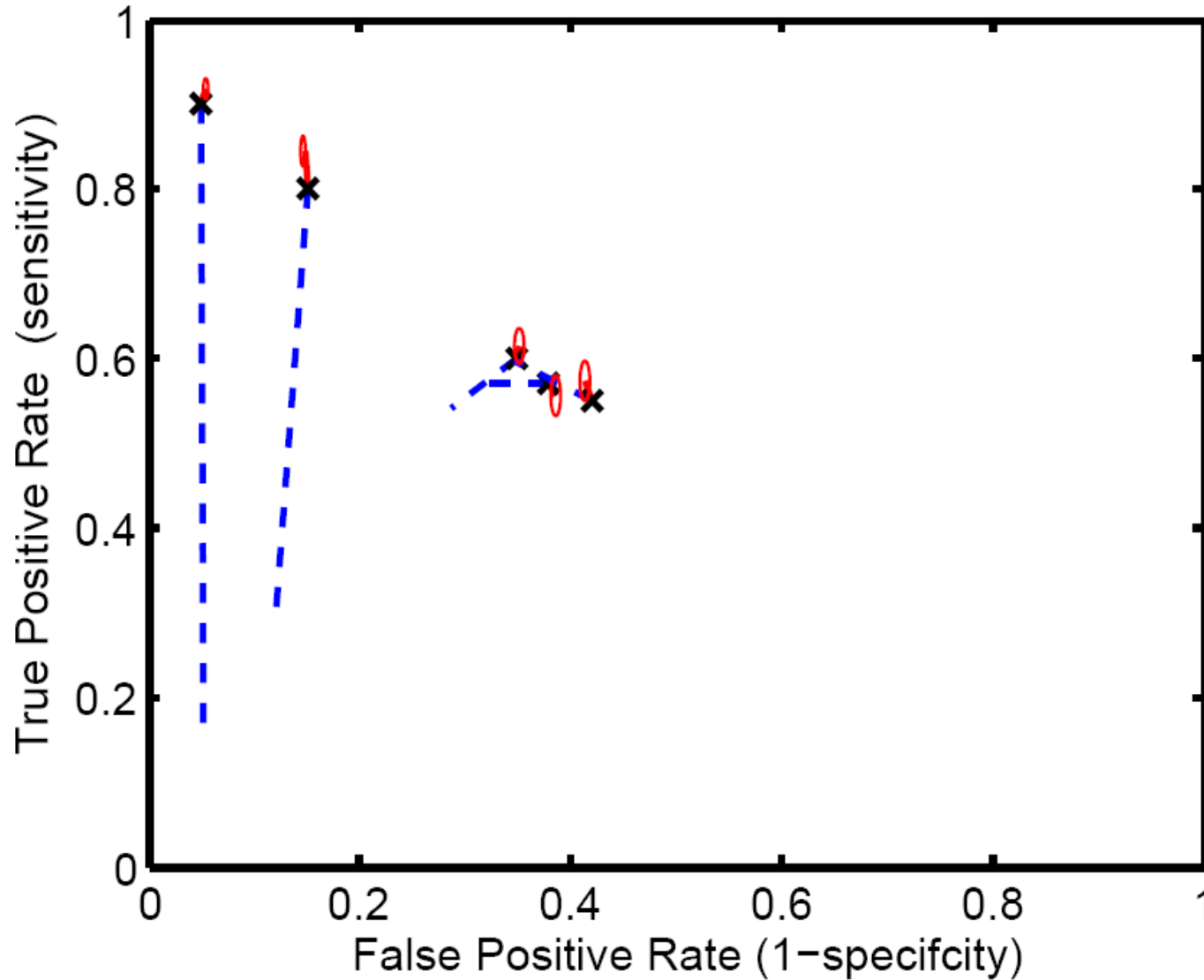
Mammography



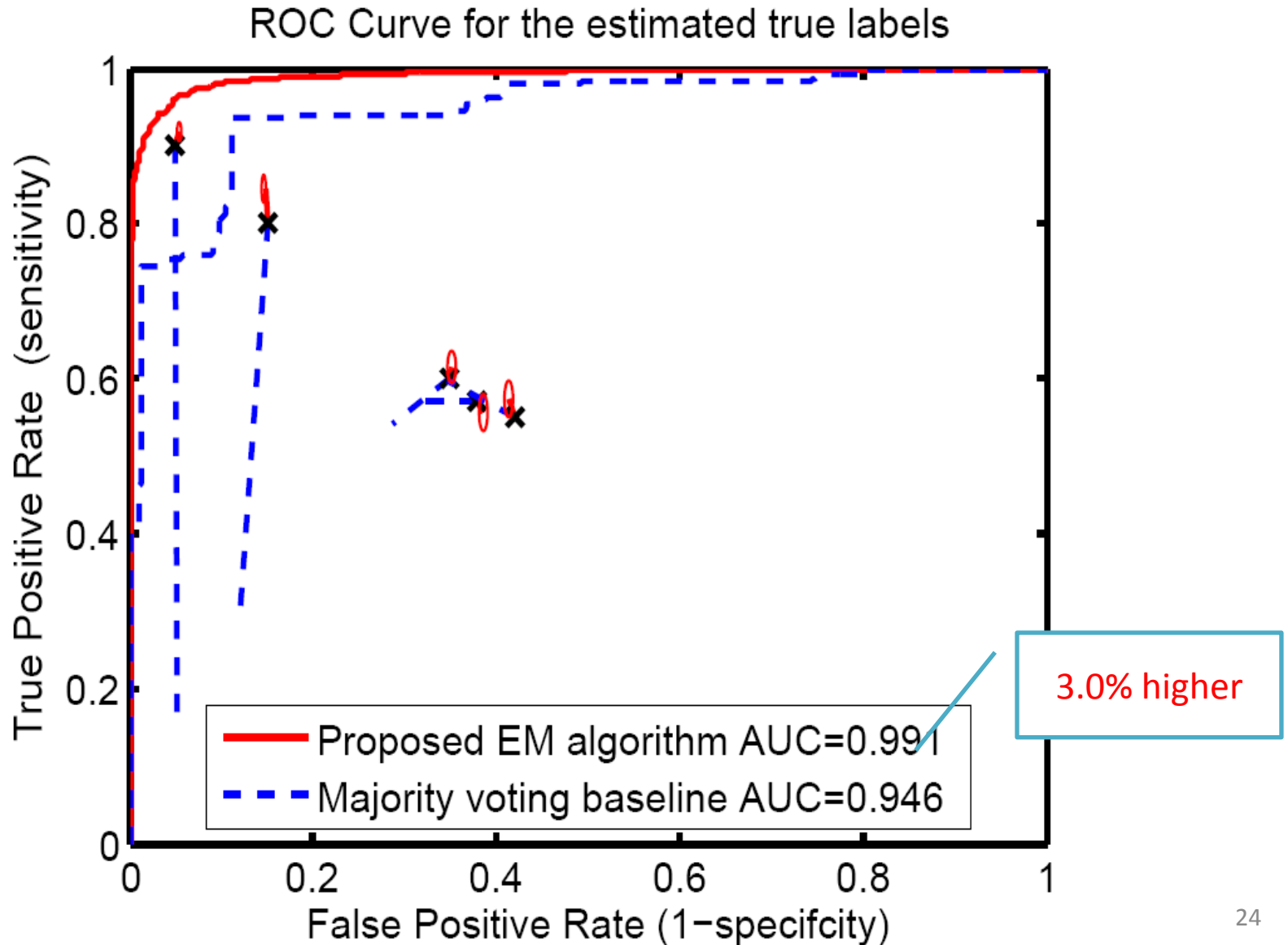
Estimated sensitivity and specificity Proposed algorithm



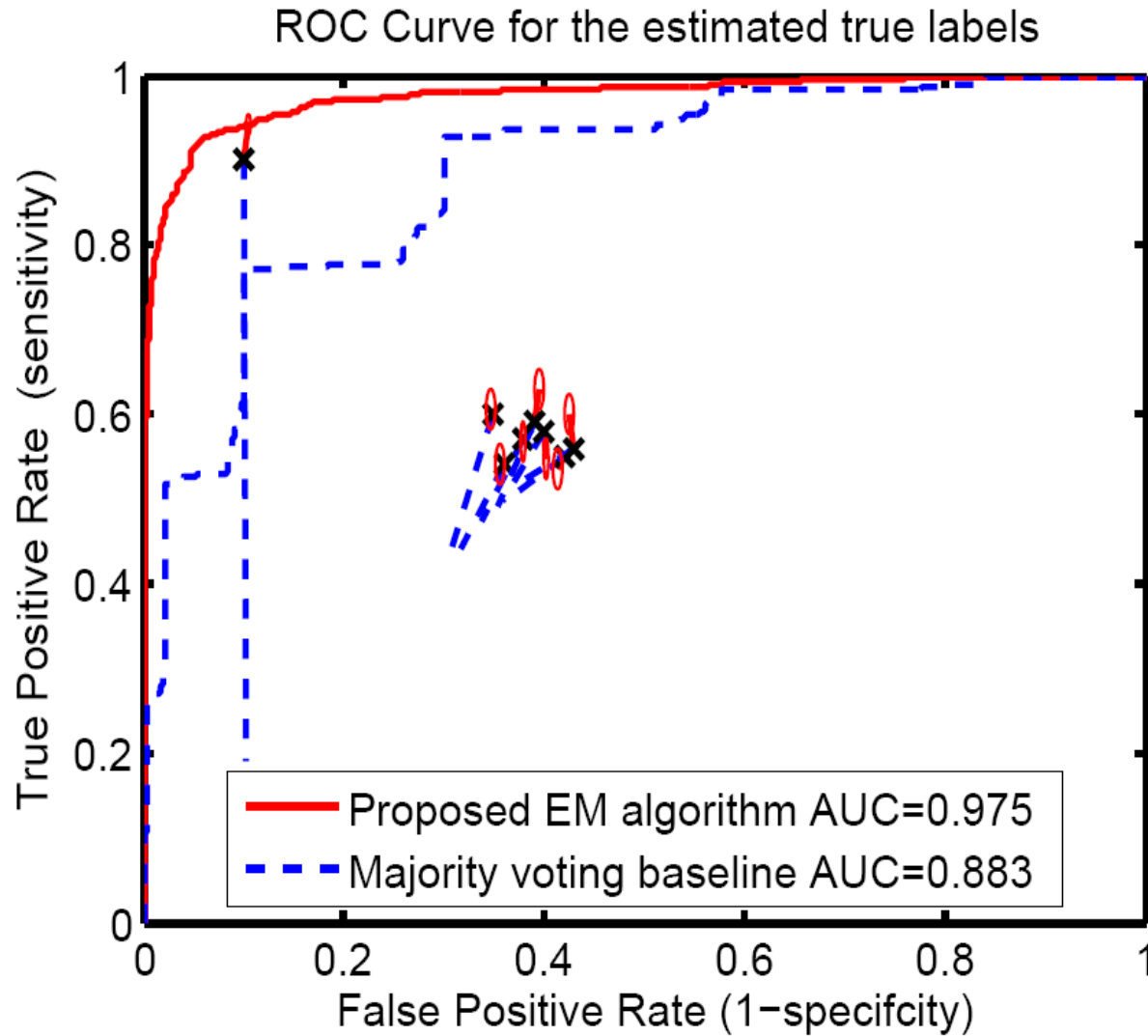
Estimated sensitivity and specificity Majority voting



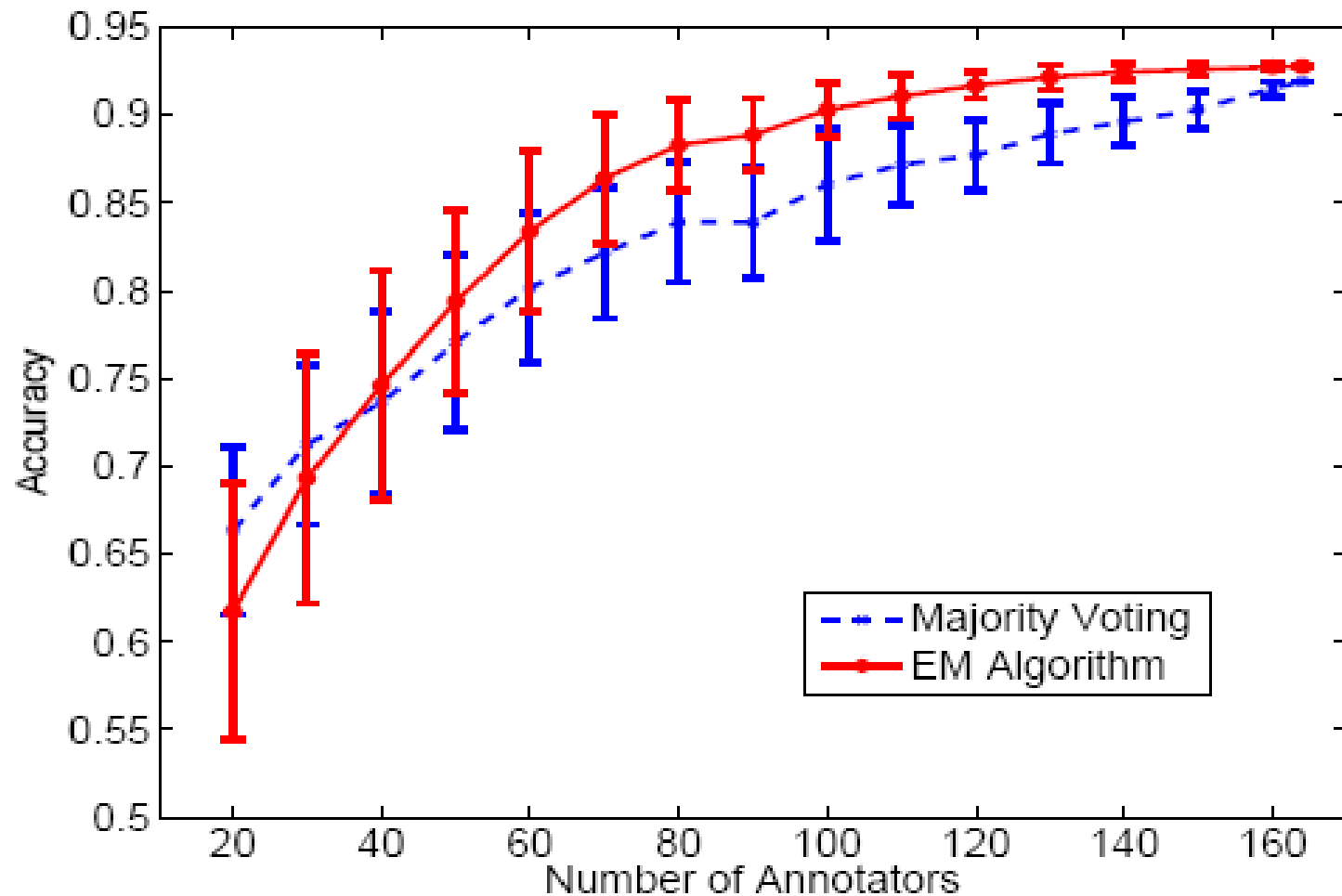
ROC for the estimated Ground Truth



We need just one good expert



Recognizing Textual Entailment



Organization

- **Crowdsourced data annotation**
 - Multiple experts
 - Workers on Amazon Mechanical Turk
- **Consolidating multiple annotations**
 - Majority Voting
 - EM algorithm via annotator models
- **Sequential crowdsourced labeling**

**Sequential Crowdsourced Labeling as
an Epsilon-greedy Exploration in a
Markov Decision Process**

Sequential Crowdsourced **Labeling** as
an Epsilon-greedy Exploration in a
Markov Decision Process

Binary Labeling with multiple annotators

Problem setup

unknown

$$\theta^j := \Pr[y_i^j = z_i]$$

annotator accuracies* 0.5 0.7 0.6 0.8 0.9

m annotators / workers

unknown

n instances / tasks		A1	A2	A3	A4	A5	true label
	1	1	1	0	1	1	1
	2	0	1	0	1	1	1
	3	0	0	0	1	1	1
	4	1	0	1	0	0	0
	:			y_i^j			:
	100	0	0	1	1	1	1

z_i

Goal

Estimate the true label and the annotator accuracies based on the observed labels.

Various approaches

Majority Voting
Weighted Majority Voting
EM algorithms

Dawid & Skene AS 1979

Bayesian approaches

Raykar et al. JMLR 2010, Liu et al. NIPS 2013

cost is nm (500) labels

Sequential **Crowdsourced Labeling** as
an Epsilon-greedy Exploration in a
Markov Decision Process

Crowdsourced Binary Labeling

Ask for k labels per instance

unknown

$$\theta^j := \Pr[y_i^j = z_i]$$

annotator accuracies*

0.5 0.7 0.6 0.8 0.9

large dynamic pool of m annotators / workers

n instances / tasks	k=3	A1	A2	A3	A4	A5	A6	...	A100
	1	1		0	1				
	2	0	1	0					
	3		0	0	1				
	4				1	0			1
	:				y_i^j				
	100	0	0	1					

cost is nk (300) labels

unknown

true label	z_i
1	
1	
1	
0	
:	
1	

Challenges

- Large pool of dynamic workforce
- No guarantees on the quality of workers
- Pull market place
- How to choose k?
- How much to pay?
- Long tail behavior

Three aspects of the problem



accuracy

cost

time

How accurate are the estimated binary labels ?

In crowdsourcing marketplaces annotators can come from a diverse pool including genuine experts, novices, biased annotators, malicious annotators, and spammers. Much of the recent work in the machine learning community has been in this area where the goal is to get an accurate estimate of the true labels based on the collected noisy labels from multiple annotators .

What is the cost of acquiring these labels?

If we collect k labels per instance the total cost is proportional to nk labels. There are no standard guidelines on how to choose the right k . Large k will result in large cost while small k results in the loss of accuracy. One solution is to perform small pilot studies with different values of k and choose the smallest k that results in a desired consensus or accuracy. In practice requesters typically try values of k in the range from 3 to 10, depending on the task and the budget.

How much time does it take?

One of the main advantages of going for crowdsourced labeling is that the task gets completed very quickly.

Can we do better than cost of nk labels?

This paper provides a (partial) solution to the cost aspect.

Sequential Crowdsourced Labeling as
an Epsilon-greedy Exploration in a
Markov Decision Process

Sequential Crowdsourcing

Ask for one label at a time

large dynamic pool of m workers

n instances / tasks	k=3	A1	A2	A3	A4	A5	A6	...	A100
	1	1		0	1	1			
	2				1	1			
	3		0	0	1	1	1		1
	4				0	0			1
	:								
	100	0	0	1		1			

Some instances need more labels to reach a consensus while a lot of instances need very few labels to reach a good consensus. This motivates the sequential crowdsourced labeling, instead of asking for labels in one shot we acquire labels from annotators sequentially.

Three questions ?

- When should you stop asking for more labels for a given instance?
- Which instance should we label next ?
- Which annotator should the requester ask for a label from?

Proposed solution has three components



Variational Bayes for approximating the posterior of the true label



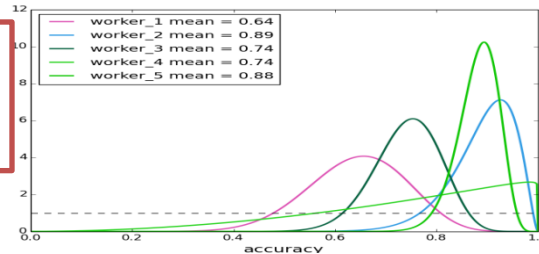
Decision theoretic reward function for the value of collecting a new label



Sequential label acquisition as an exploration in a Markov Decision Process

Variational Bayes

posterior of the annotator accuracies



annotator accuracies* 0.5 0.7 0.6 0.8 0.9

$$\theta^j := \Pr[y_i^j = z_i]$$

m annotators / workers

	A1	A2	A3	A4	A5
1		0	0		1
2					1
3	0	0			
4				1	1
:					
100	0		1		0

Given the observed labels \mathbf{y} the task is to estimate the posterior distribution $p(z, \theta | \mathbf{y})$ of the true labels z and the annotator parameters θ . We assume a beta prior $p(\theta^j | a^j, b^j) = \text{Beta}(\theta^j | a^j, b^j) \propto (\theta^j)^{a^j-1} (1 - \theta^j)^{b^j-1}$ for the annotator accuracies². We use Bernoulli prior $p(z_i | p_i) = \text{Bernoulli}(z_i | p_i) = p_i^{z_i} (1 - p_i)^{1-z_i}$ for z_i .

posterior

0.01	0.99
0.11	0.89
0.95	0.05
0.03	0.97
:	:
0.83	0.17

posterior of the true label after collecting k labels so far

$$\pi_i^{(k)}(z_i) := p(z_i | \mathbf{y}_i^{(k)})$$

We use Variational Bayes (VB) [Liu et al. NIPS 2013] to approximate the posterior.

Decision theoretic value

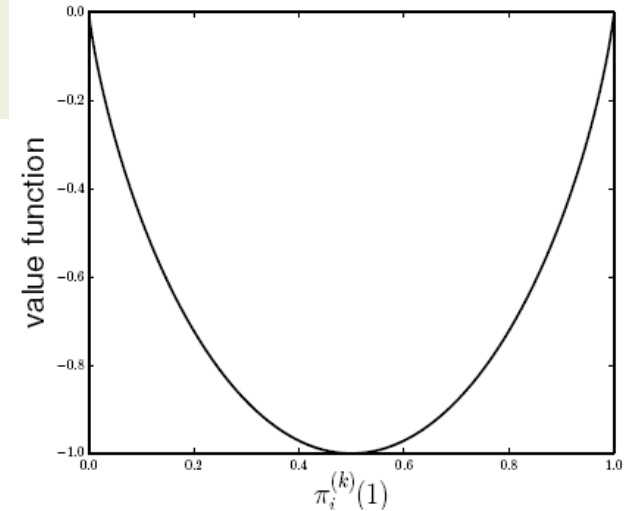
Value function (maximum expected utility)

Value to the task requestor of collecting k labels for instance i

$$V_i(\pi_i^{(k)}) = \sum_{z_i \in \{0,1\}} \pi_i^{(k)}(z_i) \log_2 \left(\pi_i^{(k)}(z_i) \right)$$

posterior

negative Shannon entropy



When should you stop asking for more labels for a given instance?

$V_i(\pi_i^{(k)}) \geq \delta$, for a user defined value δ close to zero.

Value of a new label

Expected value function If we now collect one more label we can recompute the posterior $\pi_i^{(k+1)}(z_i)$ and also the value function $V_i(\pi_i^{(k+1)})$. However we would like to know the value of collecting one more label, *prior to observing that label*. Let us say we ask for a la-

Which annotator and item should you ask for a label?

$$\mathcal{V}_i^j = \left[\sum_{y_i^j \in \{0,1\}} V_i(\pi_i^{(k+1)} | y_i^j) p(y_i^j) \right] - V_i(\pi_i^{(k)})$$

marginal

$$= E_{y_i^j} \left[E_{z_i | y_i^j} \left[\log_2 \left(\frac{\pi_i^{(k+1)}(z_i)}{\pi_i^{(k)}(z_i)} \right) \right] \right]$$

Lindley's information

expected value of the Kullback-Leibler divergence quantifies the expected increase in utility by asking for a label from annotator j

Markov Decision Process

Sequential crowdsourced labeling can now be formulated as an exploration/exploitation problem in an appropriately defined Markov Decision Process (MDP)

A MDP [Puterman, 1994] is a four tuple: $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}\}$,

1. \mathcal{S} is a finite set of *states*
 2. $\mathcal{A}(s)$ is a finite set of *actions* available in state s
 3. $\mathcal{P} : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$ denotes the *transition probabilities* between the states
 4. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto [0, R_{max}]$ is positive bounded reward function
1. At any time t the state s_t corresponds to the set of (instance,annotator) pairs which have been labeled so far.
 2. An action a_t corresponds to querying for the label of the i^{th} instance from the j^{th} annotator, and the acquired label is represented as y_i^j . At any given state s_t the set of actions available corresponds to the all the (instance,annotator) pairs which have not yet been queried for labels so far.
 3. The action a_t changes the state to s_{t+1} , the transition probabilities $\mathcal{P}(s_{t+1}|s_t, a_t)$ are given by the marginals $p(y_i^j)$.
 4. The expected immediate reward is given by utility function

If the agent is in state s and performs action a , then $\mathcal{P}(\cdot|s, a)$ is the distribution over next possible states and $\mathcal{R}(s, a)$ is the expected immediate reward received. The state transitions possess the *Markov property*, given s and a , the next state is conditionally independent of all previous states and actions.

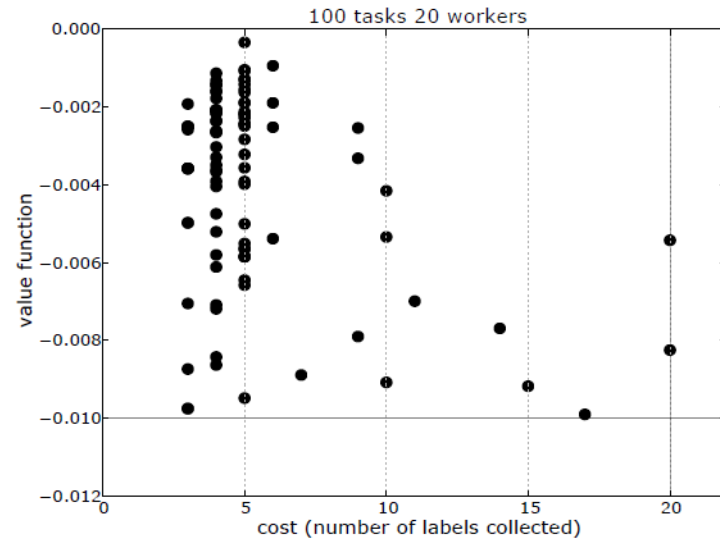
Sequential Crowdsourced Labeling as
an Epsilon-greedy Exploration **in a**
Markov Decision Process

Markov Decision Process

Greedy strategy to choose the action

If the dynamics of the MDP are known, a policy can be found mapping states to actions that maximizes the expected discounted reward by solving the recursively defined *Bellman optimality equations* [Bellman, 1957], $Q(s, a) = \mathcal{R}(s, a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s'|s, a) \max_{a' \in \mathcal{A}} Q(s', a')$, and selecting action $\pi^*(s) = \arg \max_{a \in \mathcal{A}} Q(s, a)$. The three standard methods [Russell et al., 1995] used are value iteration, policy iteration, and linear programming.

Since our defined MDP has a very large state space, direct computation with the above methods is infeasible. So we take a greedy strategy and ask for a label which maximizes the reward function. This essentially corresponds to the first step of the value iteration algorithm and can be viewed as a local multi armed bandit approximation [Duff and Barto, 1997] to the MDP.



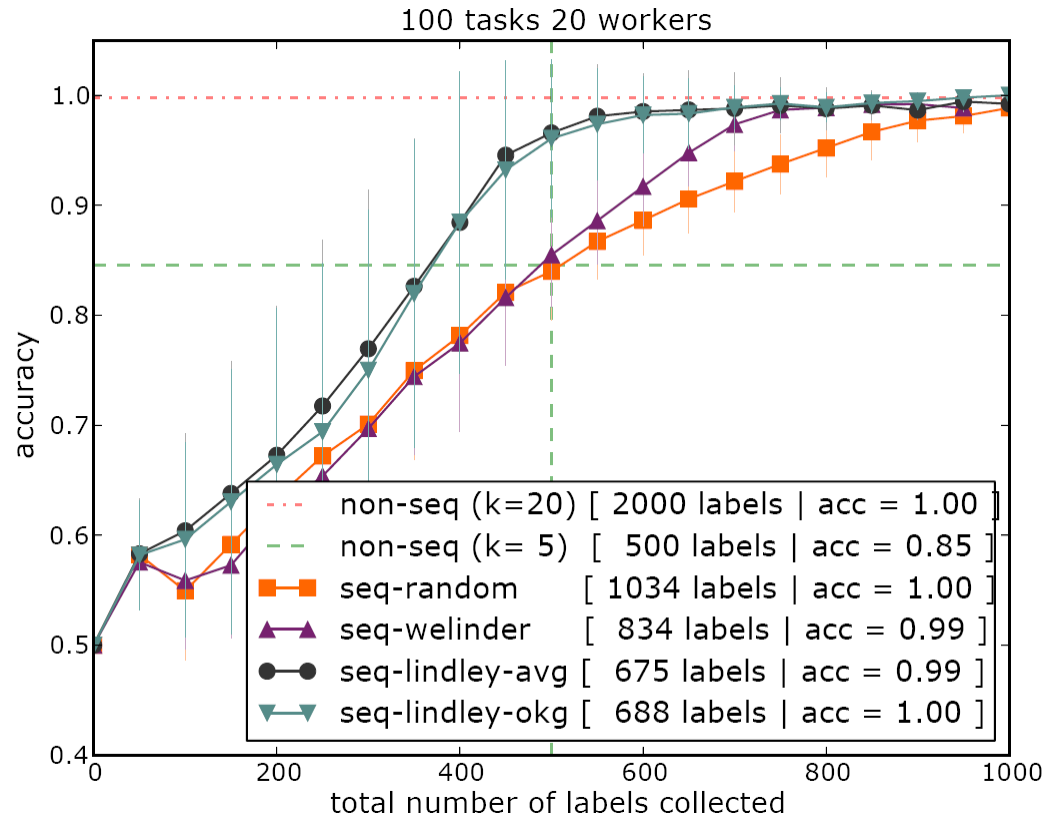
exploration/exploitation

If the model (transition probabilities) are known then we can get a near-optimal policy. In our case the parameters of the model are also re-estimated after each action. Hence one needs to also introduce an *exploration phase* to explore the state space. We use ϵ -greedy exploration [Watkins, 1989], where the agent chooses actions greedily with probability $1 - \epsilon$ and chooses actions randomly with probability ϵ .

**Sequential Crowdsourced Labeling as
an Epsilon-greedy Exploration in a
Markov Decision Process**

Experimental validation

We sample 100 instances with equal prevalence for positives and negatives. We simulate labels from a pool of 20 annotators with randomly chosen accuracies.



seq-lindley uses the least amount of labels

675 labels, a 35% reduction over seq-random and a 66% reduction over using all the annotators

methods compared

non-seq (k=20)

This corresponds to the non-sequential approach where we collect labels from all the 20 annotators. This essentially has the maximum accuracy (1.00) that can be achieved and costs a total of 2000 labels.

non-seq (k=5)

This is also a non-sequential approach where we collect 5 labels per task from randomly chosen annotators. This costs us 500 labels and achieves an accuracy of 0.85. This is the approach typically used on the AMT marketplace.

seq-random

This is the sequential labeling strategy where the next annotator is randomly chosen from the pool of annotators.

seq-welinder

This is the sequential labeling strategy proposed in [Welinder et al., CVPR 2010]. This is essentially same as seq-random except that at each round we eliminate spammers from the labeling process.

seq-lindley-avg

This is our proposed sequential labeling strategy which does an 0.1-greedy exploration in an MDP with the reward function based on the average value function.

seq-lindley-okg

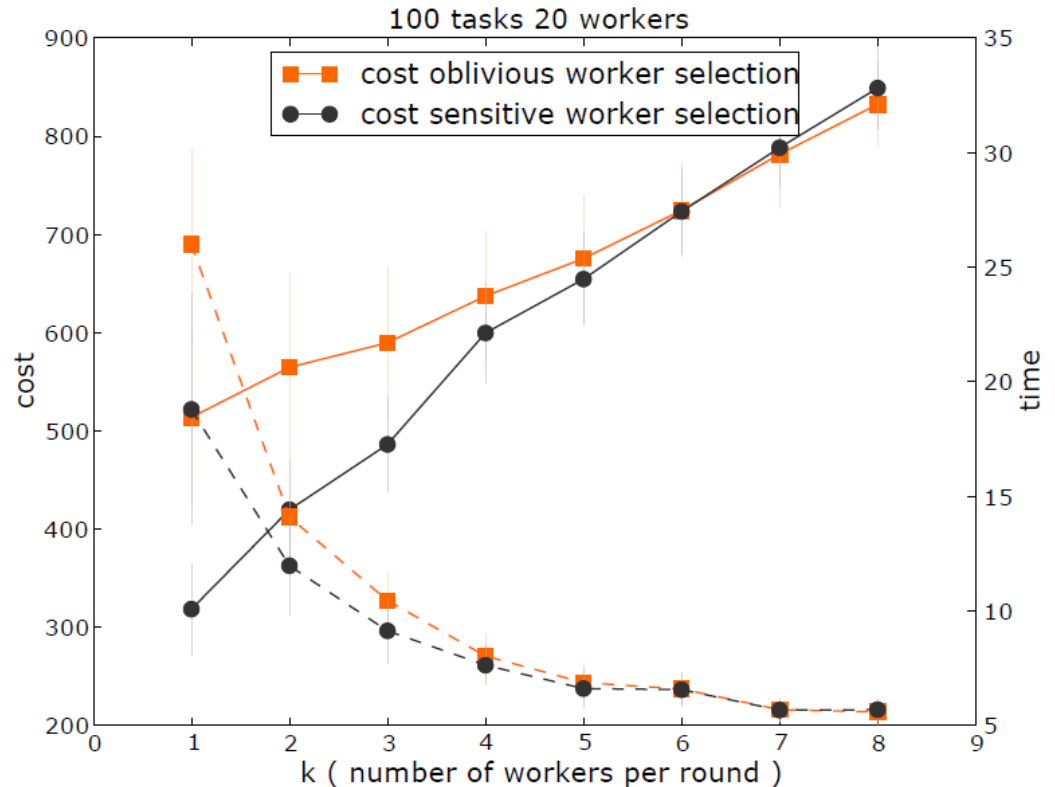
This is same as the earlier method except that instead of the average we use the maximum value of the value function as the reward as proposed in [Chen et al., ICML2013].

Incorporating labeling costs

$$\mathcal{V}_i^j = \underbrace{-c_i^j}_{\text{cost}} + \left[\sum_{y_i^j \in \{0,1\}} V_i(\pi_i^{(k+1)} | y_i^j) p(y_i^j) \right] - V_i(\pi_i^{(k)})$$

Annotators can specify the cost at which they are willing to provide the labels.

A highly accurate annotator may not necessarily contribute to the largest change in utility if the cost of providing the label is very high.



Push marketplace

In the push marketplace (for example annotators hired to perform specific annotation tasks) the requesters push the task to the workers. Once a task is allocated the workers are guaranteed to finish the task.

Pull marketplace

In contrast, in a pull market place (AMT being a prime example) the workers pull the tasks from the requesters. The requester posts tasks on the marketplace for a fixed price, the worker then goes through the list of tasks and takes up any task which he is interested in.

$$V_i^j = \frac{1}{p^j} \left[-c_i^j + \left[\sum_{y_i^j \in \{0,1\}} V_i(\pi_i^{(k+1)} | y_i^j) p(y_i^j) \right] - V_i(\pi_i^{(k)}) \right]$$

probability of task acceptance

Pull marketplaces

From the sequential labeling perspective, this implies that even if we assign a task to a particular worker, we are not guaranteed that the worker will provide the label.

A less accurate worker who always accepts the tasks may yield a higher utility than a highly accurate worker who seldom accepts the tasks.

AMT experimental results

100 instances 10 annotators	cost (# of labels)		% reduction in cost		accuracy		
	seq-random	seq-lindely	seq-random	seq-lindely	original	seq-random	seq-lindely
anger	462	385	53.8 %	61.5 %	0.96	0.97	0.96
disgust	463	409	53.7 %	59.1 %	1.00	0.99	1.00
fear	427	385	57.3 %	61.5 %	0.91	0.90	0.91
joy	419	349	58.1 %	65.1%	0.89	0.89	0.89
sadness	478	451	52.2 %	54.9%	0.94	0.93	0.95
surprise	386	343	61.4 %	65.7%	0.91	0.91	0.91

We perform experiments using the publicly available AMT dataset collected by [Snow et al.,EMNLP_2008]. We specifically use the six affective analysis datasets, wherein each annotator is presented with a list of short headlines, and is asked to give numeric judgments in the interval [0,100] rating the headline for six emotions: anger, disgust, fear, joy, sadness, and surprise. The dataset contains 100 tasks and 38 distinct annotators. Each task is labeled by a random set of 10 annotators.

Since each task is labeled by 10 annotators we have a total of 1000 labels. Using this dataset we can consolidate the labels using our proposed variational Bayes approach and evaluate the accuracy of the resulting consensus ground truth using the gold standard labels. The goal of this experiment is to analyze if using the proposed sequential crowdsourcing approach, the same accuracy could have been achieved at a reduced cost (that is, using fewer labels).

The sequential strategies can achieve the same accuracies as the original dataset at roughly half the cost (number of labels), resulting in a 50%-65% reduction of cost.

Some open problems

- How much should we pay the worker ?
- What about the time taken to complete the task ?
- Is the accuracy of the worker dependent on the pay (incentive) ?
- Can we design other incentives ?

Organization and References

✓ Crowdsourced labelling

[Learning From Crowds](#)

Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo H. Valadez, Charles Florin, Luca Bogoni, and Linda Moy

Journal of Machine Learning Research, Vol. 11, pp. 1297–1322, April 2010.

✓ Sequential crowdsourced labeling

[Sequential crowdsourced labeling as an epsilon-greedy exploration in a Markov Decision Process](#) †

Vikas C. Raykar and Priyanka Agrawal

Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 832–840, Reykjavik, Iceland, 2014.