Aalto University
School of Electrical
Engineering

Seyda Ertekin, Cynthia Rudin & Haym Hirsh

Approximating the crowd

In Data Mining and Knowledge Discovery
July 2014

Presented by Reima Karhila
November 12 2014

Motivation

- In tasks like image labelling for training of a recognition system, unreliable labellers can be ignored by using majority vote systems
- In tasks like
  - Question answering service,
  - Listening tests on speech processing techniques,
  - Developing a new, better running shoe or
  - Presidential elections,

  there is no single ground truth answer, but majority vote decides what is the “right” answer
- Economic motivation: How to determine the majority vote of a crowd for a task with minimal number of voters*?

*) Not recommended for presidential elections

Assumptions

- Traditional approach: Large enough sample should give us a reliable estimate, (& estimate of errors) if the crowd is uniform
- But the crowd consists of labelers/ voters with a range of capabilities, motives, knowledge, views, personalities, etc.
  - → Some people are more reliable/average than others
  - → Can we benefit from this?

Goals

- By biasing the more reliable/average votes we can improve the accuracy of the sampled opinion of the crowd
- And even better, let's do this on-line:
  - we can start by as few as three samples;
  - Bring in new samples one at a time:
    1. Update the reliability of workers
    2. Update the reliability of estimate
    3. Decide if another vote is needed

Mathematics

- We have M labellers
- For each task in t={1,...,T}, we have vector of votes V_t drawn from a binary distribution (ie. “Yes vs no” or “head vs tails”): $V_t \sim \mu(\{-1,1\}^M)$
- V_t represents the set of votes given by the labellers at time t
- Simple majority vote Y_t at time t:

$$Y_t = \begin{cases} 1 & \text{if } \sum_{i=1}^{M} V_{ti} > 0 \\ -1 & \text{otherwise.} \end{cases}$$

Mathematics cont.

- Assuming a fixed unit cost:
  How many votes should be “revealed” and in which order to “approximate the crowd” as cheap as possible?
- Cost of algorithm is cost(pi) and Accuracy of algorithm is defined as

$$\text{Reward}(\pi) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}_{\left[\hat{Y}_t = Y_t\right]}$$

- The optimisation for approximating the crowd is defined as (for hard/soft constraints)

$$\max_{\pi} \mathbb{E}_{\{V_t\}_t : V_t \sim \mu} \left[\text{Reward} \mid \pi\right] \quad \text{s.t.} \quad \text{Cost}(\pi) \leq C_{\text{hard}}$$

$$\max_{\pi} \mathbb{E}_{\{V_t\}_t : V_t \sim \mu} \left[\text{Reward} - C_{\text{soft}} \cdot \text{Cost}(\pi) \mid \pi\right]$$

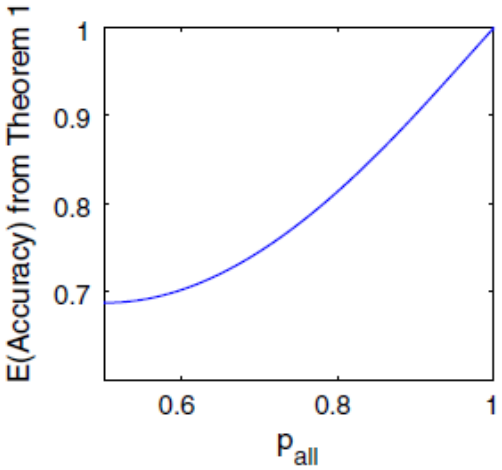
Mathematics still cont.

- These constraints define "***efficient frotier of solutions***"
- Examining a binary signal, where each labeler agrees with x_t^{signal} with a probability p_{all}
- Assume p_{all} is known and $p_{all} > 0.5$ (without loss of generality, due to symmetry)
- Allows analytic solving of
  - Expected accuracy for nontrivial policy that makes one vote per example, Cost= T*u, where u is unit cost of one vote from one labeller
  - Expected cost for optimal policy achieving perfect accuracy

Mathematics:
One vote per example

- Expected accuracy:

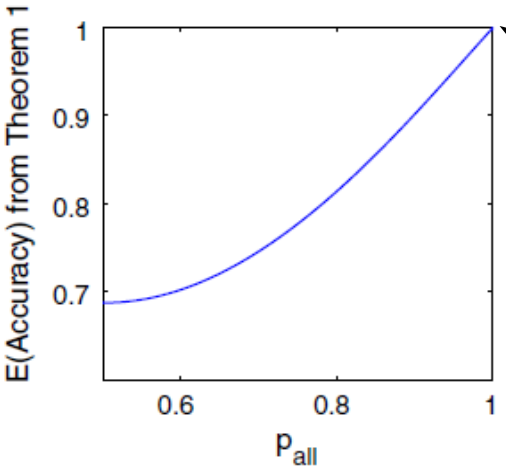
$$\mathbb{E}_{\{\mathbf{v}_1,\ldots,\mathbf{v}_T\sim\mu(\{-1,1\}^M)\}} = p_{all}\cdot\left[\sum_{x=\frac{M-1}{2}}^{M-1}\mathrm{Bin}(x, M-1, p_{all})\right] + (1-p_{all})\cdot\left[\sum_{x=\frac{M-1}{2}}^{M-1}\mathrm{Bin}(x, M-1, 1-p_{all})\right]$$



Mathematics:
One vote per example

- Expected accuracy:

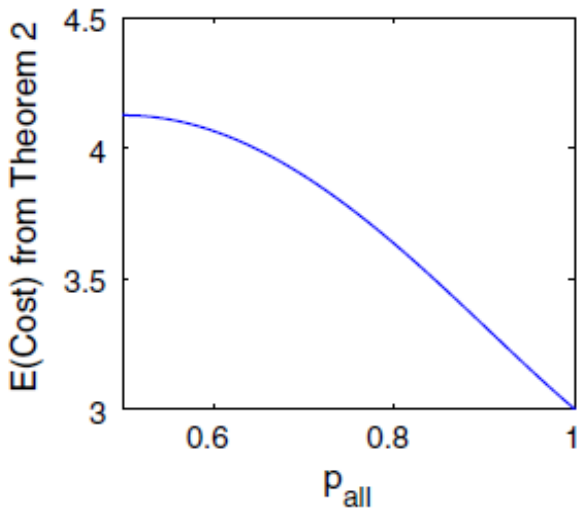
$$\mathbb{E}_{\{\mathbf{v}_1,\ldots,\mathbf{v}_T\sim\mu(\{-1,1\}^M)\}} = p_{all}\cdot\left[\sum_{x=\frac{M-1}{2}}^{M-1}\mathrm{Bin}(x, M-1, p_{all})\right] + (1-p_{all})\cdot\left[\sum_{x=\frac{M-1}{2}}^{M-1}\mathrm{Bin}(x, M-1, 1-p_{all})\right]$$



Mathematics: Perfect accuracy

- Expected cost of the optimal policy that achieves perfect accuracy in predicting the crowd's majority vote:

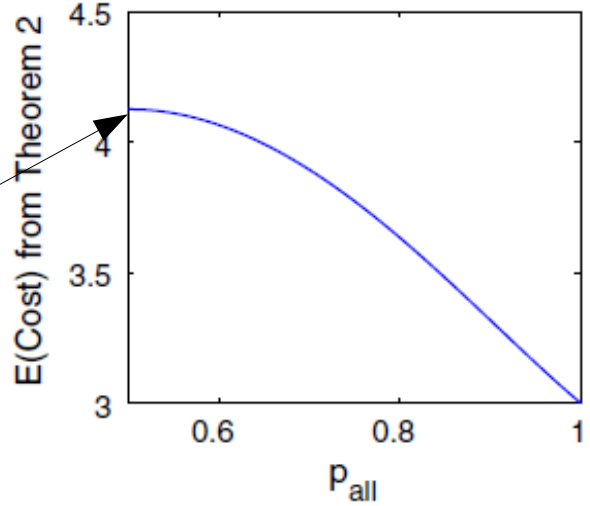
$$\Omega = T \sum_{j=\binom{M+1}{2}}^{M} \binom{j-1}{\frac{M-1}{2}} \left[p_{all}^{\frac{M+1}{2}} (1-p_{all})^{j-\frac{M+1}{2}} + (1-p_{all})^{\frac{M+1}{2}} p_{all}^{j-\frac{M+1}{2}} \right] \times j$$



Mathematics: Perfect accuracy

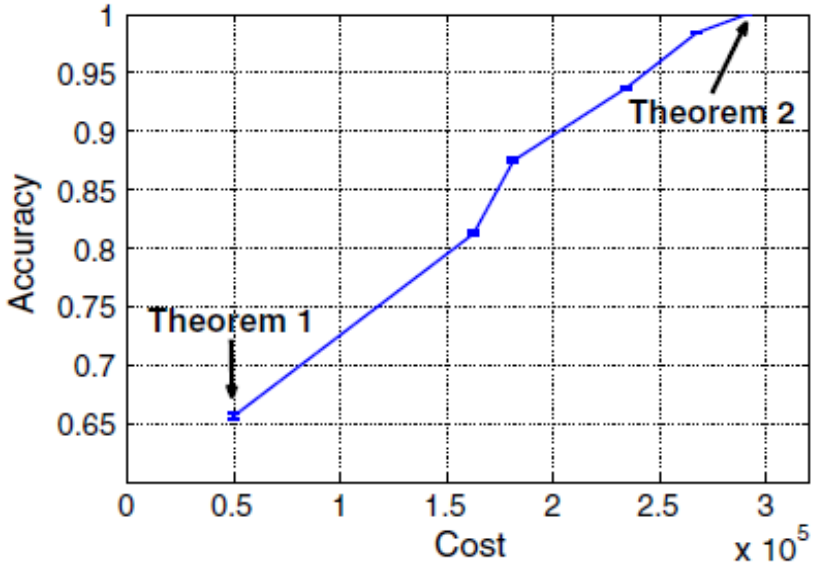
- Expected cost of the optimal policy that achieves perfect accuracy in predicting the crowd's majority vote:

$$\Omega = T \sum_{j=\binom{M+1}{2}}^{M} \binom{j-1}{\frac{M-1}{2}} \left[p_{all}^{\frac{M+1}{2}} (1-p_{all})^{j-\frac{M+1}{2}} + (1-p_{all})^{\frac{M+1}{2}} p_{all}^{j-\frac{M+1}{2}} \right] \times j$$



CrowdSense algorithm

- CrowdSense works between these limits:



CrowdSense algorithm: Worker weighting

- Model the labelers' quality estimates as a measure of their agreement with the crowd majority:
- Labelers $L=\{ l_1, l_2, \ldots l_M \}$, $X = \{ x_1, x_2, \ldots, x_t, \ldots, x_n \}$ sequence of examples, which can arrive one at a time
- $v_{it}=l_i(x_t)$ is l_i's vote on x_t and S_t is the set of labelers selected on label x_t.
- c_{it} is the number of times we've observed a label from l_i so far

$$c_{it} := \sum_{\tilde{t}=1}^{t} \mathbb{1}_{[i \in S_{\tilde{t}}]}$$

CrowdSense algorithm: Worker weighting cont.

- Define a_{it} as how many of those labels were consistent with other labelers

$$a_{it} := \sum_{\tilde{t}=1}^{t} \mathbb{1}_{\left[i \in S_{\tilde{t}}, V_{i\tilde{t}} = V_{S_{\tilde{t}}\tilde{t}}\right]}$$

- And finally get to the beef: Q_{it} is a smoothed estimate of the probability of labeler i agreeing with the crowd:

$$Q_{it} := \frac{a_{it} + K}{c_{it} + 2K}$$

where K is a smoothing parameter

CrowdSense algorithm: Confidence and online updates

- Start by selecting 2 labelers with highest Q_{it} and select one ***uniformly at random*** and put them into labeller set s_t
- Ask each of the 3 labelers in s_t to vote on an example
- Generate weighted majority vote of the labellers ***confidence score*** from votes v_{it} and labeler weights Q_{it}:

$$\text{Score}(S_t) = \sum_{i \in S_t} V_{it} Q_{it}$$

CrowdSense algorithm: Confidence and online updates cont.

- To check our confidence on the vote, we check if adding the labeler outside s_t with the highest Q_{it} would make us uncertain about the vote:

$$\frac{|\text{Score}(S_t)| - Q_{l_{\text{candidate}},t}}{|S_t| + 1} < \varepsilon \qquad , 0 < \varepsilon \leq 1$$

- Here epsilon represents acceptable level of uncertainty
- If equation is true, we keep adding new labelers until equation becomes untrue or we run out of labelers.

Tests and data sets

- Tested with 6 data sets
- All data sets transformed into {-1,1} tasks with thresholds
- Some noise added to some test sets
- Comparison with other algorithms:
  - IE Thresh
  - Labeling Quality Uncertainty (LU)
  - New label uncertainty(NLU)

MovieLens	ChemIR	Reuters	Adult	SpamBase	MTurk
Number of examples					
137	1,165	6,904	32,561	2,300	673
Labeler accuracies					
48.17(L1)	50.72(L1)	80.76(L1)	81.22(L1)	86.30(L1)	52.60(L1)
89.78(L2)	46.78(L2)	83.00(L2)	80.59(L2)	86.35(L2)	63.30(L2)
93.43(L3)	84.46(L3)	89.70(L3)	86.22(L3)	91.22(L3)	54.98(L3)
48.90(L4)	88.41(L4)	82.98(L4)	87.63(L4)	94.04(L4)	79.64(L4)
59.12(L5)	86.69(L5)	88.12(L5)	91.12(L5)	75.91(L5)	61.52(L5)
96.35(L6)	87.46(L6)	87.04(L6)	94.11(L6)	82.04(L6)	72.22(L6)
87.59(L7)	49.52(L7)	95.42(L7)	56.68(L7)	68.39(L7)	74.00(L7)
54.01(L8)	78.62(L8)	80.21(L8)	85.51(L8)	90.83(L8)	
47.44(L9)	82.06(L9)	78.68(L9)	81.32(L9)	94.35(L9)	
94.16(L10)	50.12(L10)	95.06(L10)	85.54(L10)		
95.62(L11)	50.98(L11)	82.88(L11)	79.74(L11)		
		71.57(L12)	84.86(L12)		
		87.54(L13)	96.71(L13)		

Tests

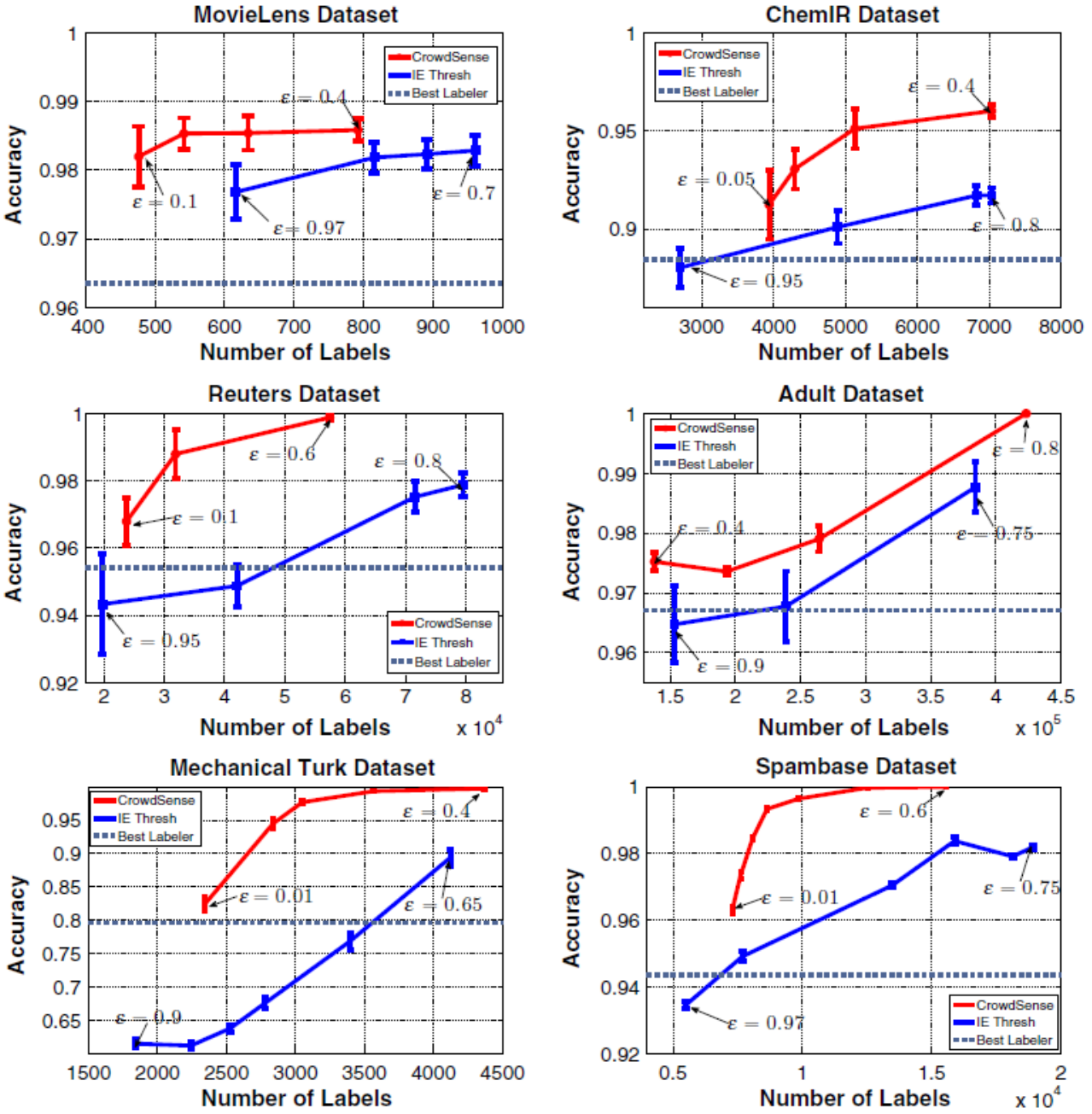


Fig. 3 Tradeoff curves for CrowdSense, baseline (b), and IEThresh, averaged over 100 runs. The x-axis is the total number of votes (the total cost) used by the algorithm to label the entire dataset. The y-axis indicates the accuracy on the full dataset

Tests cont.

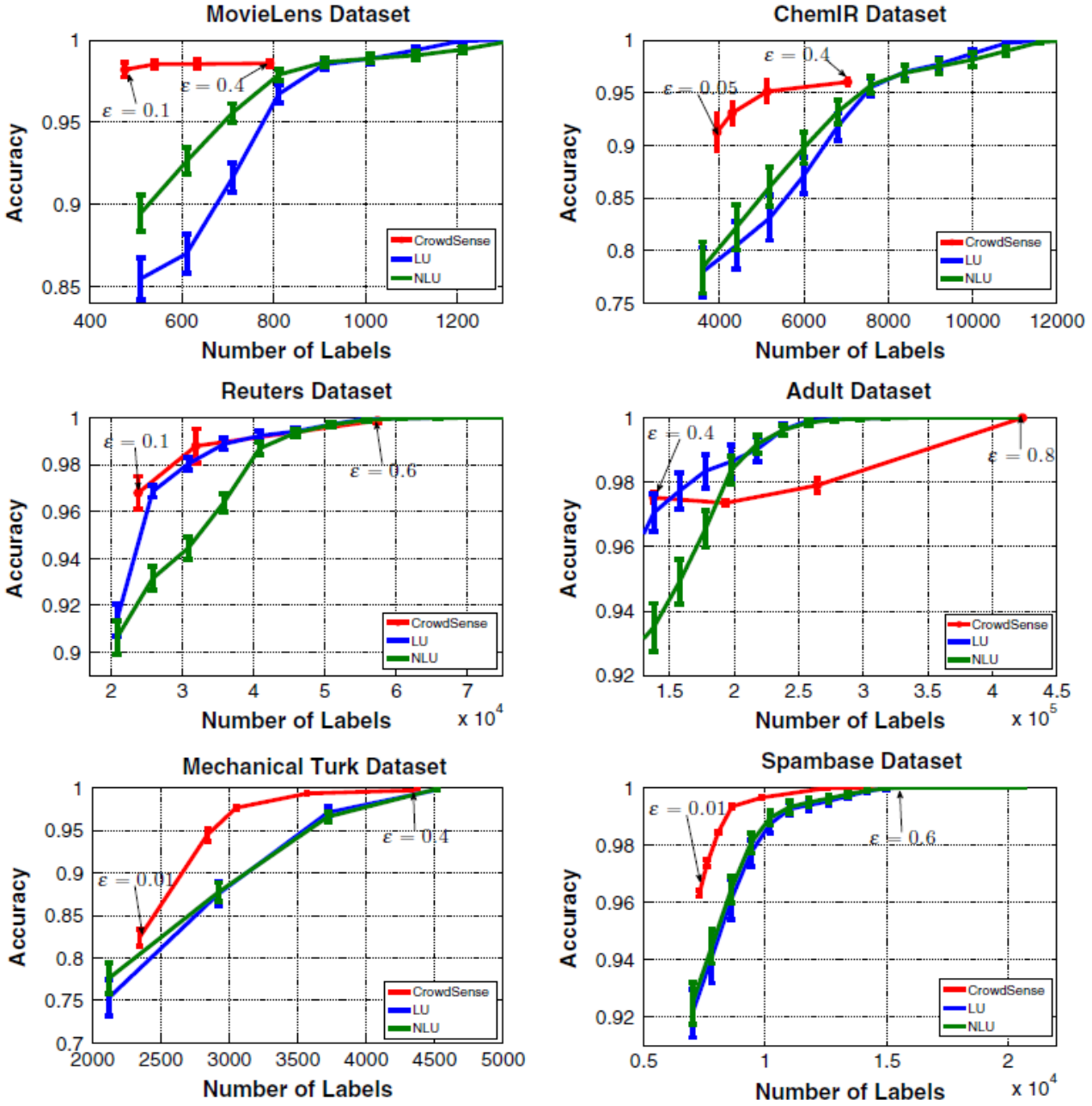


Fig. 4 Tradeoff curves for CrowdSense, LU and NLU, averaged over 100 runs. The x-axis is the total number of votes (the total cost) used by the algorithm to label the entire dataset. The y-axis indicates the accuracy on the full dataset

CrowdSense variations

- Variations of the algorithm are also presented, based on 2 different statistical assumptions:
  - CrowdSense.ind where votes are assumed to be independent (for large crowds)
  - CrowdSense.bin where the confidence score is sampled from the binomial distribution somehow (for small crowds)
- The mathematics looks serious, but these variations don't seem to improve the results on the test sets too dramatically

Thank you for your attention!