Comprehensive and Reliable Crowd Assessment Algorithms

MANAS JOGLEKAR
HECTOR GARCIA-MOLINA
ADITYA PARAMESWARAN

Background

- Crowdsourcing: Human workers perform tasks that are hard for computers, such as image tagging
- Workers are often *unreliable*
- Need to assess quality
- Need for Confidence intervals
  - 1/3 errors vs 10/30 errors

Problem Setting

- **m** tasks ($t_1 \ldots t_m$)
- Binary tasks OR **k** responses ($r_1 \ldots r_k$)
- **n** workers ($w_1 \ldots w_n$)
- Non-regular data
  - A worker may not respond to every task
- No gold standard
- Accuracy model
  - Worker $\mathbf{w_i}$ has error rate $\mathbf{p_i}$, or confusion matrix $\mathbf{P_i}$
  - Worker response **independent** of each other, given true answer

Warm-up: 3 workers, binary tasks

- Equal false positive and negative error rates
- To find: mean, variance of p_i for each I
- **Theorem**: $Y = f(X_1, X_2, X_3)$
  - X_i's **normal**, f **linear** ($\sim a_1X_1 + a_2X_2 + a_3X_3$)
  - $Var(Y) = a_1^2Var(X_1) + a_2^2Var(X_2) + a_3^2Var(X_3)$
- Works for approximately normal (binomial), locally linear (differentiable)
- Linear coefficients given by partial derivatives

Warm-up: 3 workers, binary tasks

- Compute agreement rates (Q_{ij} for worker w_i, w_j)

	t_1	t_2	t_3	t_4	t_5
w_1	Y	Y	-	N	N
w_2	N	-	Y	N	N
w_3	-	Y	Y	N	Y
True	N	Y	Y	N	N

$E[Q_{12}] = 2/3$

- $Q_{ij} \sim p_i p_j + (1-p_i)(1-p_j)$
- So $p_i = f_i(q_{ij}, q_{ik}, q_{jk})$

$$= ½ + ½ \sqrt{((q_{ij} - ½)(q_{ik} - ½)/(q_{jk} - ½))}$$

Warm-up: 3 workers, binary tasks

- Variance in p estimate depends on
  - Variance in q estimates
  - $\delta f_i/\delta q_{ij}$, $\delta f_i/\delta q_{jk}$
- $Var(p_i) = \Sigma_{q,q'}$ Cov (q,q') x $\delta f_i/\delta q$ x $\delta f_i/\delta q'$
- Estimate variances, derivatives
- Use $E[p_i]$, $Var(p_i)$ to get confidence interval

Generalizing to many workers

- Key Idea: Take multiple sets of 3 workers and combine estimates
- For each worker, form N/2 triples.
- E.g. For n=5, for 'w_1' we have groups (w_1, w_2, w_3) (w_1, w_4, w_5)
- Use each triple to compute estimate for p_1
- Compute variances, covariances of estimates
- Use weighted combination of estimates

Generalizing to many workers

- Optimum weights for combining
  - Covariance matrix A
  - Given by $A^{-1}L_{N/2}$ where L_k is a k-length vector with values 1/k
- Greedy way of group forming
  - Better to have two good workers in one group than one good worker in two groups, due to weighting
  - Greedily form groups

3 Workers Non-binary Tasks

- Confusion matrix P_i , Selectivity vector S, diagonal S^D

e.g.

0.8	0.1	0.3
0.1	0.7	0.1
0.1	0.2	0.6

,

0.4
0.2
0.4

,

0.4	0	0
0	0.2	0
0	0	0.4

- P_i's, S : Column-stochastic, unknown
- Observation probabilities given by $\mathbf{P_i S}$

0.46
0.22
0.32

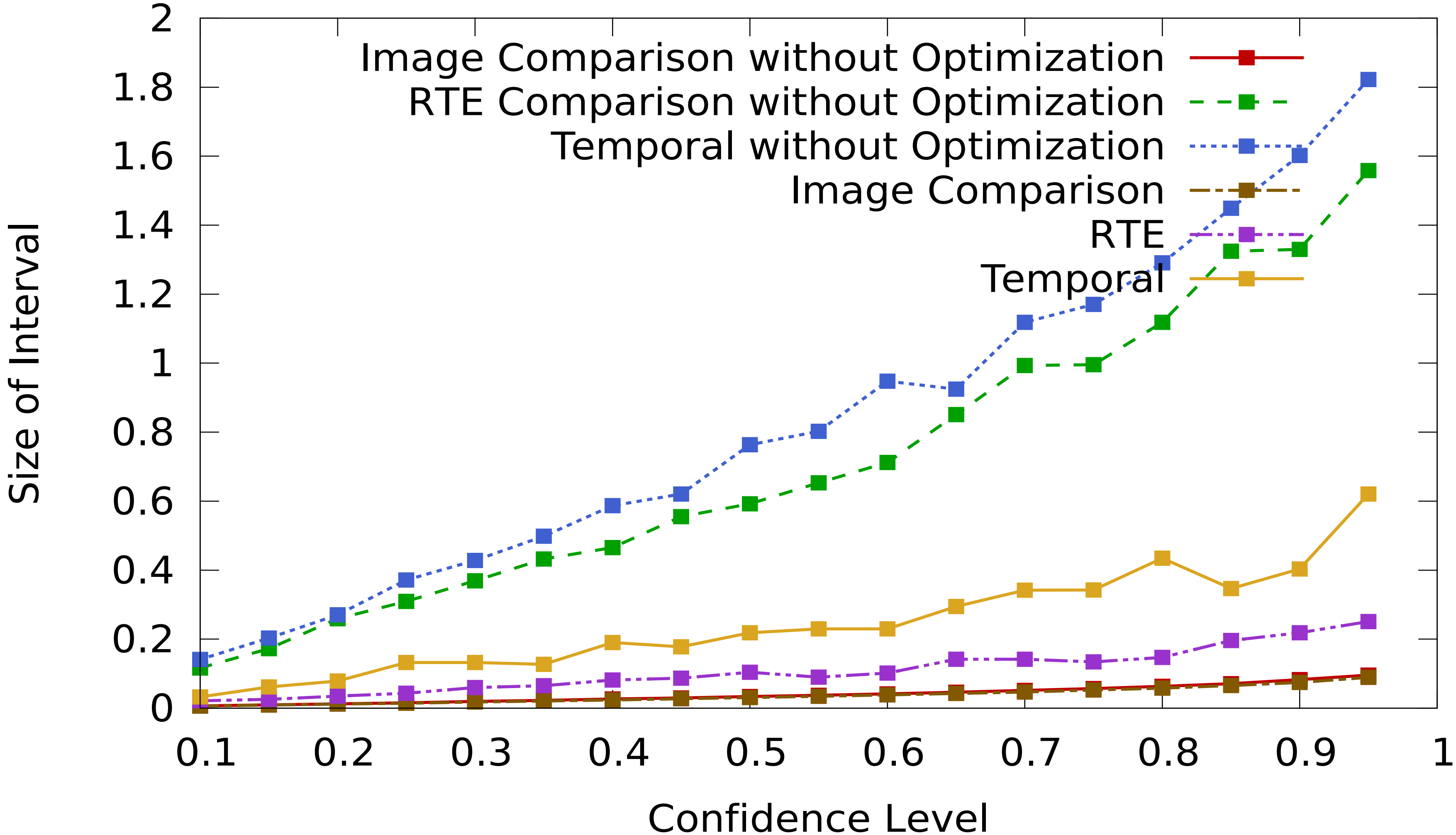
3 Workers Non-binary Tasks

- Compute comparison matrices (frequencies of response-pairs of workers w_i, w_j)
- $C_{ij} \sim P_i S^D P_j^T$
- Compute $D_i = C_{ij}(C_{jk}^T)^{-1} C_{ki} = P_i S^D P_i^T$
- Eigenvalue decomposition of D_i gives $V_i = U \sqrt{S^D} P_i$, for a unitary U.

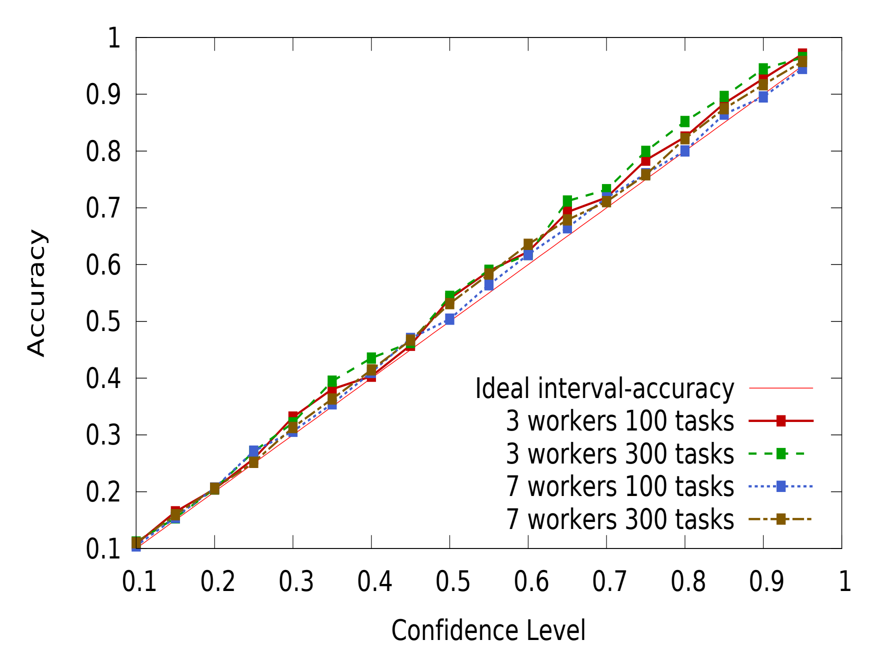
3 Workers Non-binary Tasks

- U can be recovered using 3-way comparisons
- S^D can be recovered using column stochasticity of P^i
- For confidence intervals, use variances/derivatives like in the binary tasks case
- Details in paper

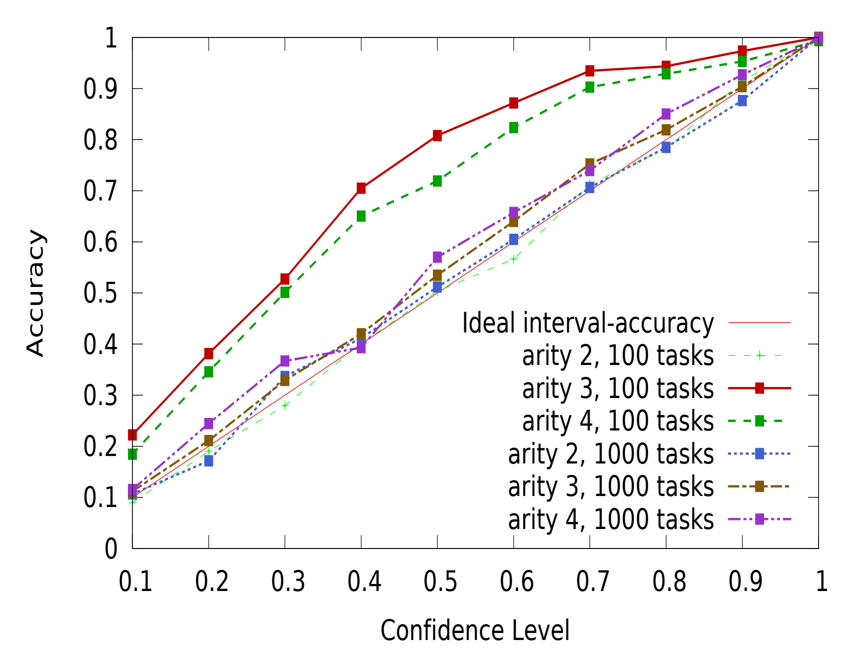
Results: Weight Optimization



Results: Calibration (Synthetic Data)

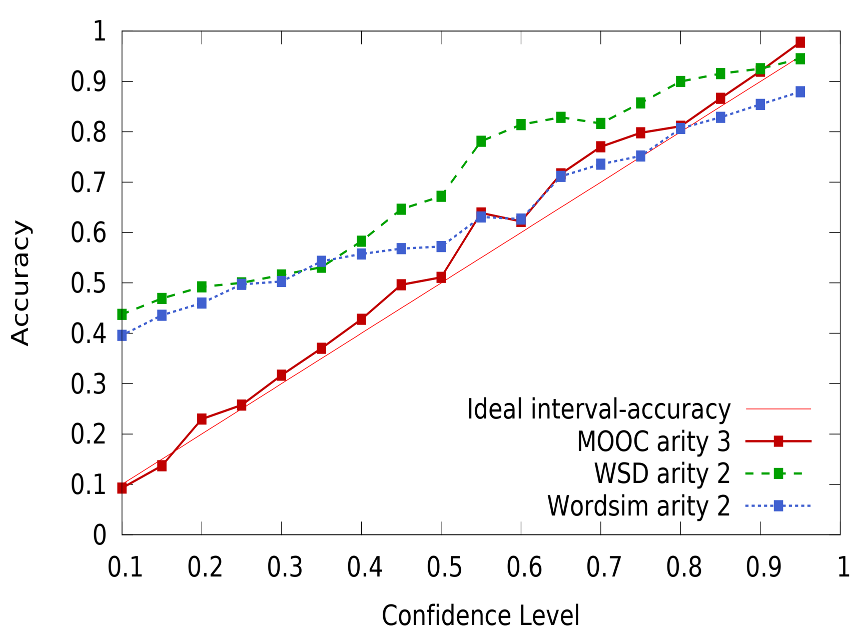


Binary Tasks

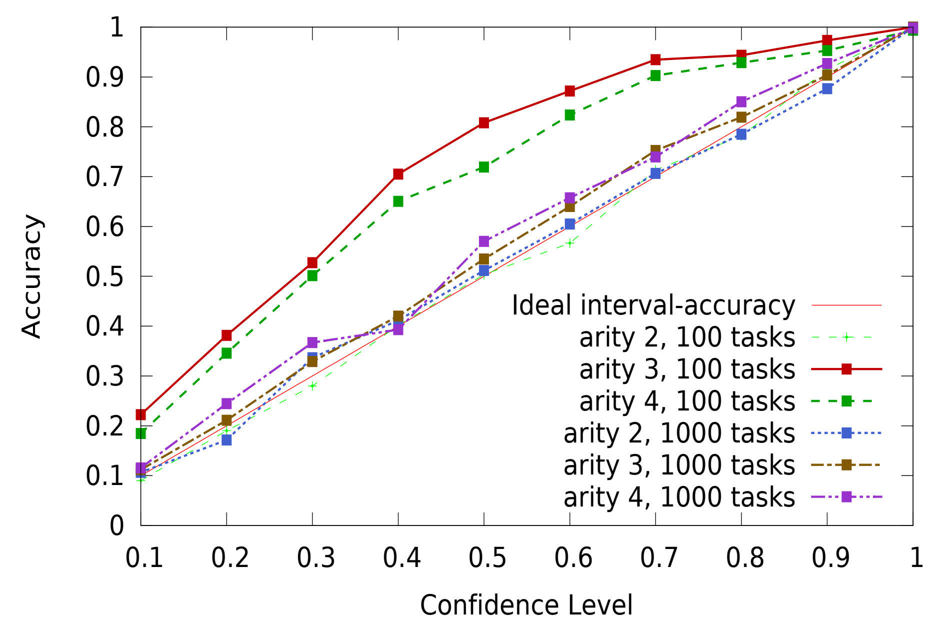


Non-Binary Tasks

Results: Calibration (Real Data)



Binary Tasks



Non-Binary Tasks

Thank You!
Questions?