Reshaping mobile crowd sensing using cross validation to improve data credibility

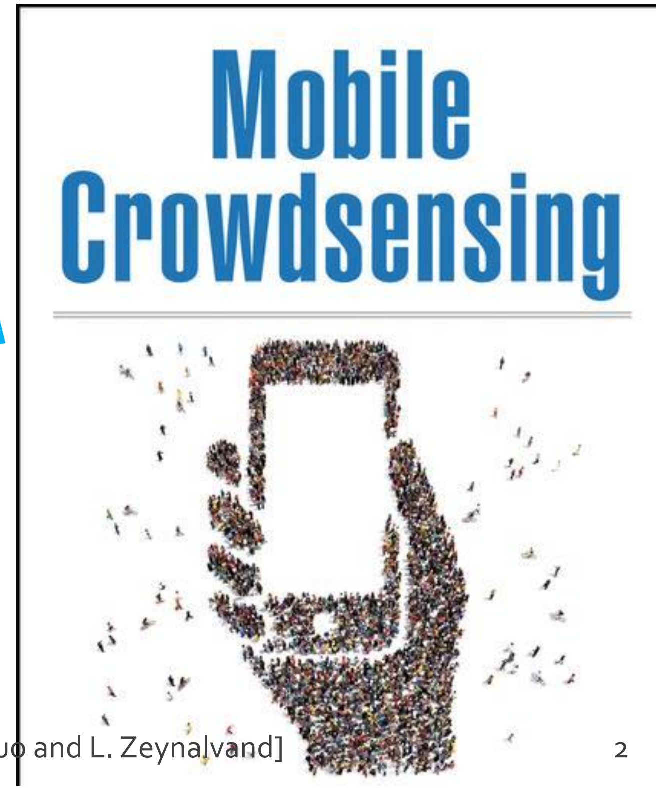
6 December 2017

Tony T. Luo*, Leonit Zeynalvand^

* Institute for Infocomm Research (I²R), A*STAR, Singapore
^ School of CSE, Nanyang Technological University, Singapore

IoT & mobile crowd sensing

- Ultimate goal of IoT: improve people's life



- People's role in IoT:

People are not merely **service consumers**;
They can also act as **data providers**

A critical issue in MCS

- **Data credibility**: Quality of crowdsensed data can be poor and inconsistent
- Existing solution approaches
  - Incentive mechanisms
  - Trustworthiness/quality of worker or data
  - Truth discovery
- "Power of crowds" not fully explored

New approach: Cross validation

- Introduce **Validating Crowd**

Contributing Crowd

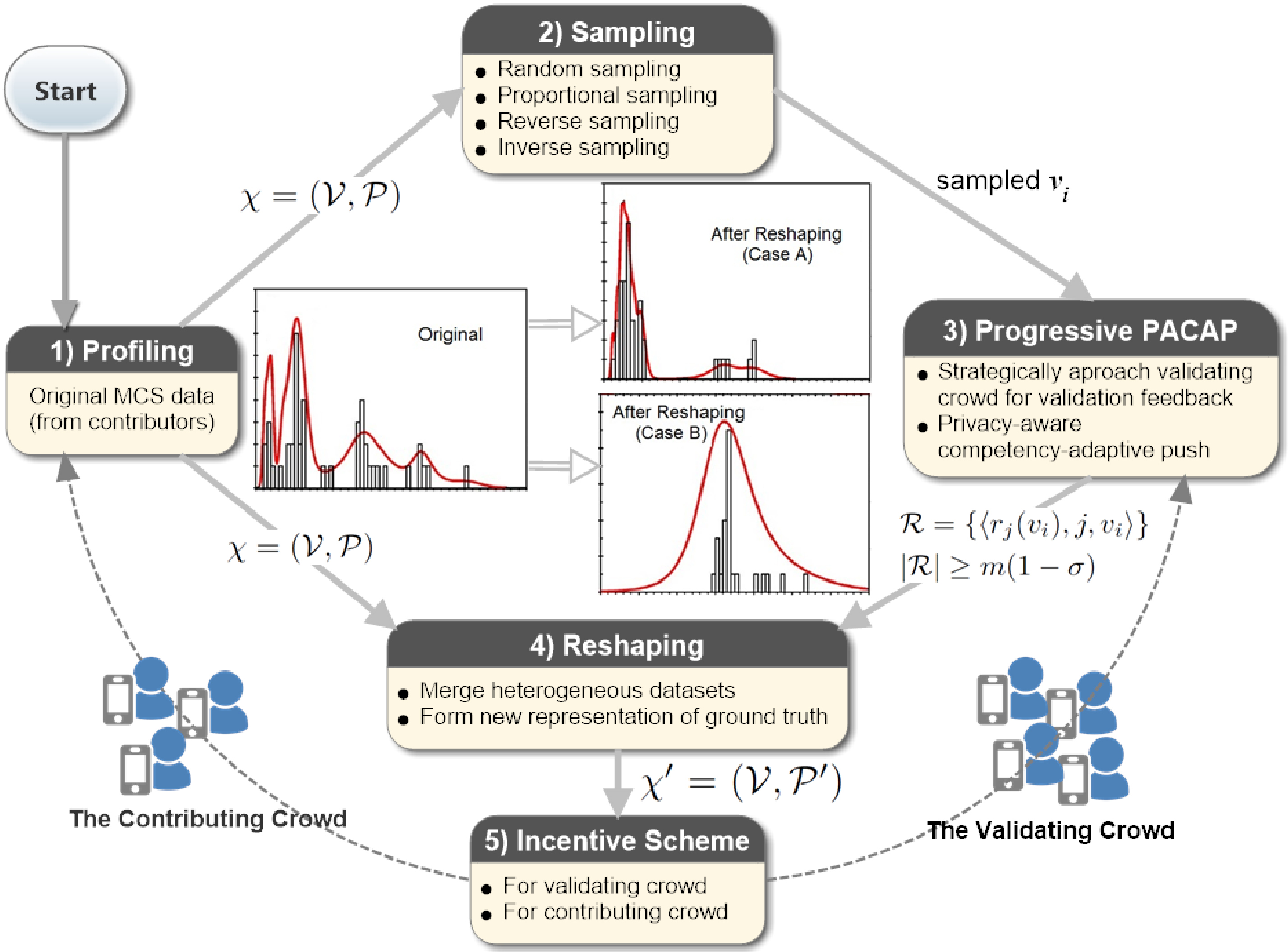
- To clarify:
  - Not expert-sourcing
  - Not the same concept as in statistics or machine learning

Reshaping mobile crowd sensing using cross validation to improve data credibility [T. Luo and L. Zeynalvand]

Challenges

1. Introduces another quality/credibility issue? (of validation)
2. How to present crowdsensed data to validators?
3. How to deal with validators' bias?
4. Privacy issue?
5. How to consolidate validation result with original crowdsensed data?
6. Need incentives for validators? And How?

Cross validation mechanism

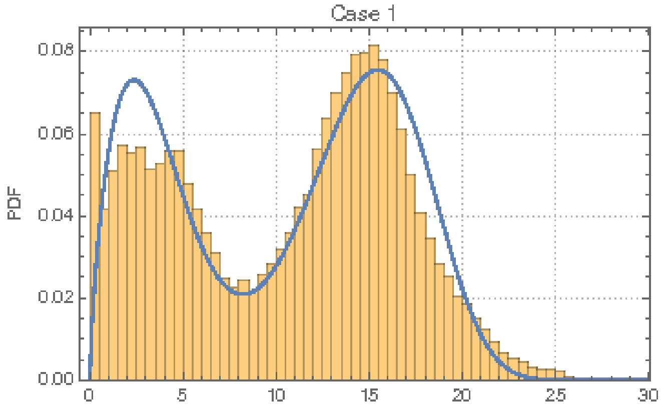


Reshaping mobile crowd sensing using cross validation to improve data credibility [T. Luo and L. Zeynalvand]

1) Profiling

- To obtain $\chi = (V, P)$
  - $V = \{v_i\}$: representative values of original data
  - $P = \{p_i\}$: probabilities of each $v_i \in V$

- Procedure
  - Create histogram
  - Select representative values
  - Normalize to probability measure



How to present data to validators?

- Candidate methods
  - Expose χ or V at a public venue (e.g., website)
    - Ex: Amazon, Quora, Stackoverflow, TripAdvisor
  - Expose χ or V to a selected group of workers
    - Ex: "Elite users" or forum admins
  - Present a subset of V to each selected worker
  - For each of the above, ask for a ranking or picking the best
    - Then perform *preference aggregation*, e.g., by using Borda count or Condorcet winner
  - All have issues: details see paper; more discussion in upcoming arXiv version (available in early 2018)
- Our method
  - **Single value, single rating**

Illustration

Rating Task

Is the following value representative of the avg. traffic speed of Broadway NY during morning peak hours today?

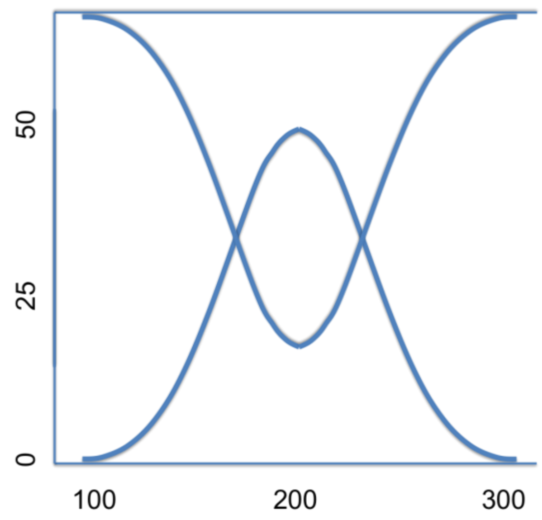
40 mph

How to select this “single value”?

2) Sampling

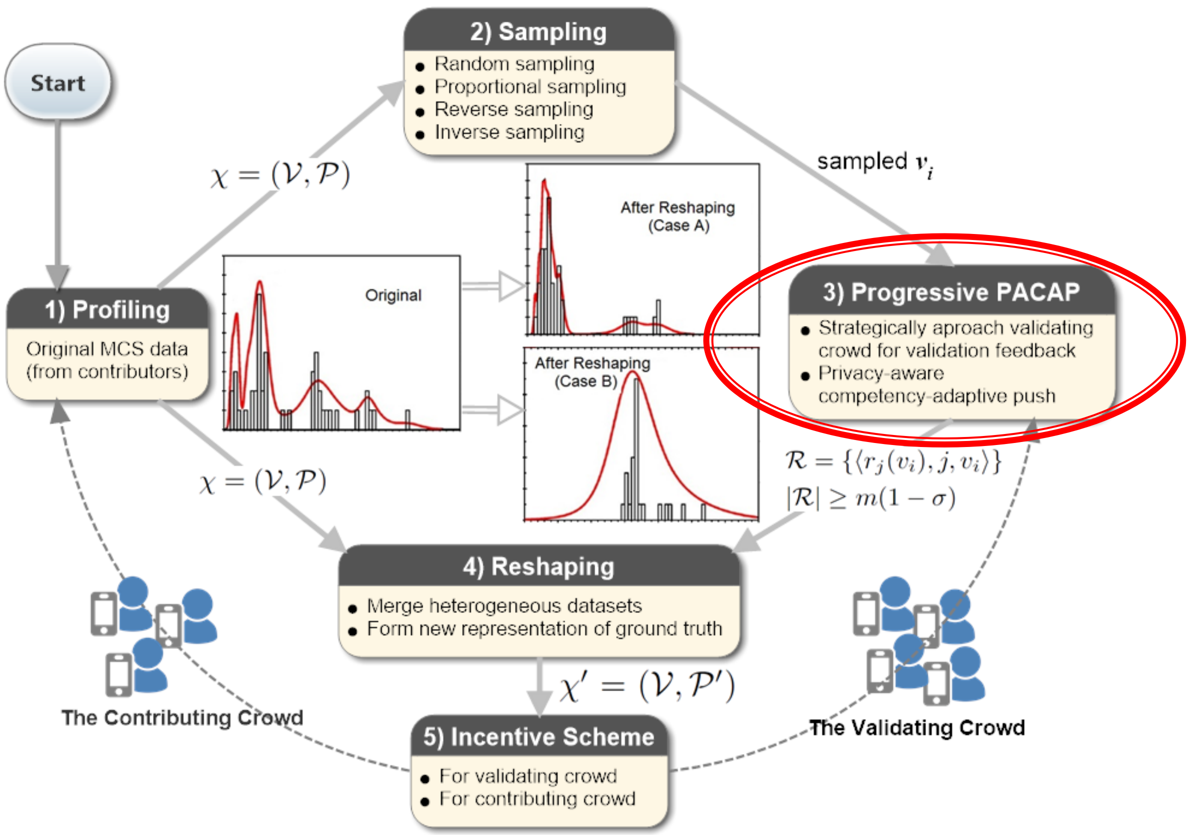
- How to select that "single value"?
  - **Sample** *V* with a certain probability distribution
  - Present each sampled value (not necessarily unique) to a (unique) validator

- Sampling methods:
  - **Random sampling**: $s_i = 1/n$, where $n = |V|$
  - **Proportional sampling**: $s_i = p_i, \forall p_i \in P$
- Other thoughts
  - Frequent values may need less validation
  - Catch "outliers": could they be uncommon truth?
- Additional sampling methods
  - **Reverse sampling**: $s_i \propto d - p_i$
    - We use: $s_i = \frac{d-p_i}{nd-1}$ where $d = p_{min} + p_{max}$
    - note: avoid *d*=1 (see paper)
  - **Inverse sampling**: $s_i \propto 1/p_i$
    - So by normalization, $s_i = \frac{1/p_i}{\sum_i 1/p_i}$



Given the sampled values…

- How to approach workers to seek ratings?



Reshaping mobile crowd sensing using cross validation to improve data credibility [T. Luo and L. Zeynalvand]

3) Privacy-aware competency-adaptive push (PACAP)

- **Proactive** approach: **Push** rating tasks to a set of strategically selected validators (raters)
- Issues with push:
  - **(Privacy) intrusive**
  - **Competency**: "are you pushing to the right people?"
- Other restrictions:
  - **Quantity** requirement: desire m ratings with a shortfall tolerance α, i.e., below $m(1-\alpha)$ unacceptable
  - **Time** constraint: collect all ratings within deadline T_o
- Solution: privacy-aware competency-adaptive push (PACAP)

Design considerations

- Anti-bias
- Competency control
- Privacy awareness

Select a rater *j* at time *t* with prob. $q_j(t)$:

$$q_j(t) = \frac{1 - e^{-\lambda_j (t - t_j^-)(R_j + \epsilon)}}{\sum_{j \in \Psi} \left[1 - e^{-\lambda_j (t - t_j^-)(R_j + \epsilon)} \right]}$$

R_j : Reputation of *j*; $R_j \geq 0$
λ_j: personalized elasticity parameter catering for *j*'s privacy preference; $\lambda_j \in [1, \lambda_{max}]$
t_j^- : the time when *j* receives the last offer
ϵ: ensure users with R_j=0 (e.g. new users) still have chance

Intuition:
1) higher reputation, higher chance
2) avoid too frequent pushes to the same rater while mitigating starvation
3) privacy customization via λ (details in paper)

Challenge

- **Rater behaviors are highly uncertain and dynamic** (decline offer, accept offer, delay, non-response)

Solution:

- Divide T_o into multiple cycles
- Perform **progressive push** over cycles
  - Each cycle to approach a different group of raters of a different group size with a different number of offers
    - Accumulate **statistics** for each cycle
    - Determine group size for next cycle by **predicting** an *effective response ratio* by learning from historical statistics
    - Select the group members using the selection probability $q_j(t)$

Algorithm

Algorithm 1: Progressive PACAP

Input: Crowdworkers $\mathcal{U}$, contributors $\mathcal{C}$, representatives $\mathcal{V}$, target m, tolerance α, deadline T_0

Output: $\mathcal{R} = \{\langle r_j(v_i), j, v_i \rangle | r_j(v_i) \neq 0, j \in \mathcal{U}, v_i \in \mathcal{V}\}$ with $|\mathcal{R}| \geq m \cdot (1-\alpha)$, or FAIL otherwise

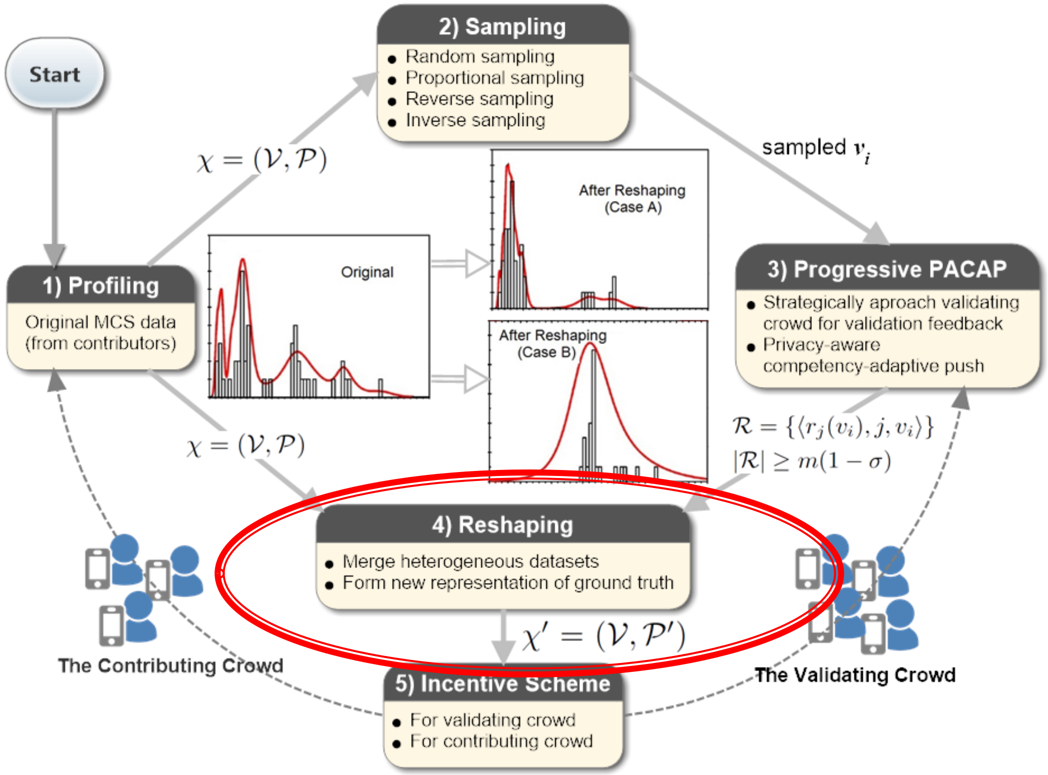
$\mathcal{R} \leftarrow \emptyset, \Psi \leftarrow \mathcal{U} \setminus \mathcal{C}$
$m(1) \leftarrow m$, $M_Y(0) \leftarrow 0$, $M_N(0) \leftarrow 0$
for $k \leftarrow 1$ **to** T_0/τ **do**
  select $m(k)$ raters, denoted by a set $\mathcal{M}(k)$, from Ψ according to Eq. (3)
  for each $j \in \mathcal{M}(k)$ **do**
    obtain one $v_i \in \mathcal{V}$ using a sampling method from Section III-B
    wrap v_i in a rating task and push it to rater j to seek rating $r_j(v_i)$
  end
  wait for τ while collecting ratings:
  ∘ $\mathcal{R}(k) \leftarrow \{\langle r_j(v_i), j, v_i \rangle | r_j(v_i) \neq 0\}$
  ∘ $m_N(k) \leftarrow \sum_{j} \quad {}_{r_j(v_i)=0}$
  $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}(k)$, $\quad m_Y(k) \leftarrow |\mathcal{R}(k)|$
  if $|\mathcal{R}| \geq m$ **then**
    return $\mathcal{R}$ // SUCCESS *
  end
  // Prepare for the next cycle:
  update $\Psi \leftarrow \Psi \setminus \mathcal{M}(k)$
  $M_Y(k) \leftarrow M_Y(k-1) + m_Y(k)$,
  $M_N(k) \leftarrow M_N(k-1) + m_N(k)$,
  determine the scale of next outreach:

$$m(k+1) \leftarrow [m - M_Y(k)] \left[1 + \frac{M_N(k)}{M_Y(k)}\right]$$

end
if $|\mathcal{R}| < m(1-\alpha)$ **then**
  return FAIL
else
  return $\mathcal{R}$ // SUCCESS
end

Next...

- Given the ratings, how to consolidate them with the original data?



Reshaping mobile crowd sensing using cross validation to improve data credibility [T. Luo and L. Zeynalvand]

4) Reshaping

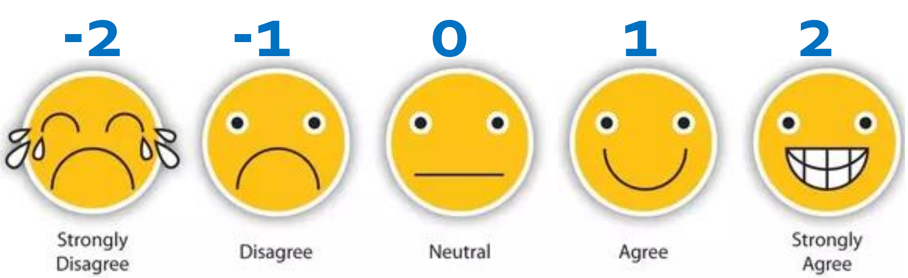
- Given: $\boldsymbol{\chi}$={V,P} and $\boldsymbol{R}$ (set of ratings)
- Output: $\boldsymbol{\chi}'$={V,P'} (reshaped profile)

$$\hat{p}_i = \frac{\kappa_i + \eta g_i \frac{\sum_{i=1}^{n} \kappa_i}{|\mathcal{R}|}}{\sum_{i=1}^{n} \kappa_i + \eta (g_i + b_i) \frac{\sum_{i=1}^{n} \kappa_i}{|\mathcal{R}|}} = \frac{p_i + \eta \frac{g_i}{|\mathcal{R}|}}{1 + \eta \frac{g_i + b_i}{|\mathcal{R}|}}.$$

$$g_i = \frac{1}{w_l} \sum_j r_j(v_i) \mathbb{1}_{r_j(v_i) > 0},$$

$$b_i = -\frac{1}{w_l} \sum_j r_j(v_i) \mathbb{1}_{r_j(v_i) < 0}$$

Intuition: each original p_i can be interpreted as the ratio of contributors who "**voted**" for v_i to be the truth; during CV, each v_i receives another set of votes from the raters to whom the same v_i was pushed.



5) Incentive scheme

- Need to cater for two crowds
- **Raters**: update reputation as

$$R'_j = [R_j + \Delta_j(v_i)]^+$$

where $[x]^+ = \max(0, x)$, and

$$\Delta_j(v_i) = \begin{cases} \frac{p'_i - p_i}{1 - p_i} \frac{r_j(v_i)}{w_l}, & \text{if } p'_i > p_i \\ \frac{p'_i - p_i}{p_i} \frac{r_j(v_i)}{w_l}, & \text{if } p'_i < p_i. \end{cases}$$

Intuition: reputation depends on
1) how consistent is her rating r_i with the final belief adjustment (p'_i - p_i)
2) how much her rating r_i has contributed to the belief adjustment

- **Contributors**: receive payments as

$$\pi'_c = \pi_c\left(u_c \frac{p'_i(c)}{p_i(c)}, \mathbf{u}'_{-c}\right),$$

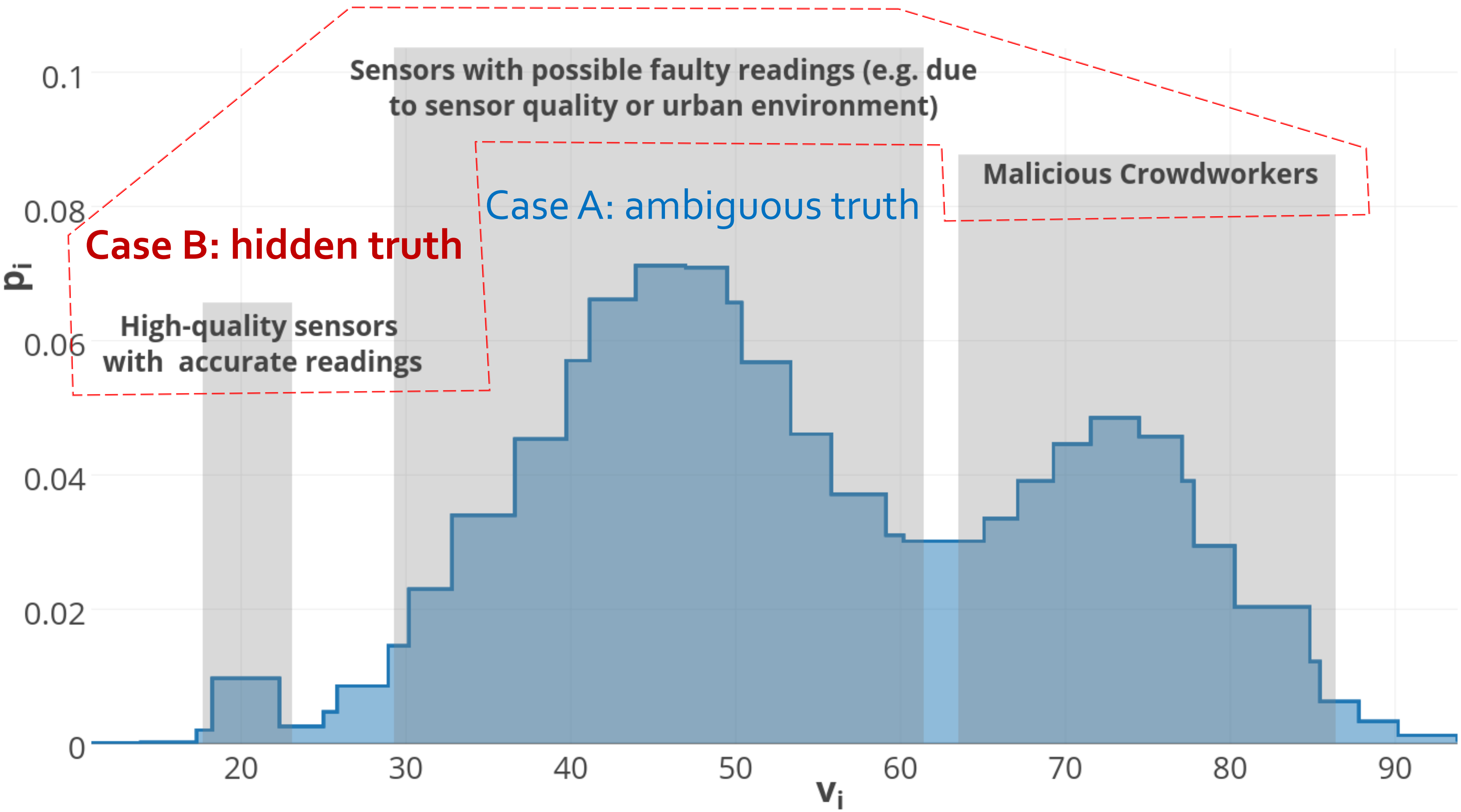
$$\mathbf{u}'_{-c} = \left\{u_{\tilde{c}} \frac{p'_{\tilde{i}}(\tilde{c})}{p_{\tilde{i}}(\tilde{c})} \middle| \tilde{c} \in \mathcal{C} \setminus \{c\}\right\}$$

Intuition: p'_i and p_i can be interpreted as the quality of contribution v_i (likelihood of v_i being the truth)

Performance evaluation

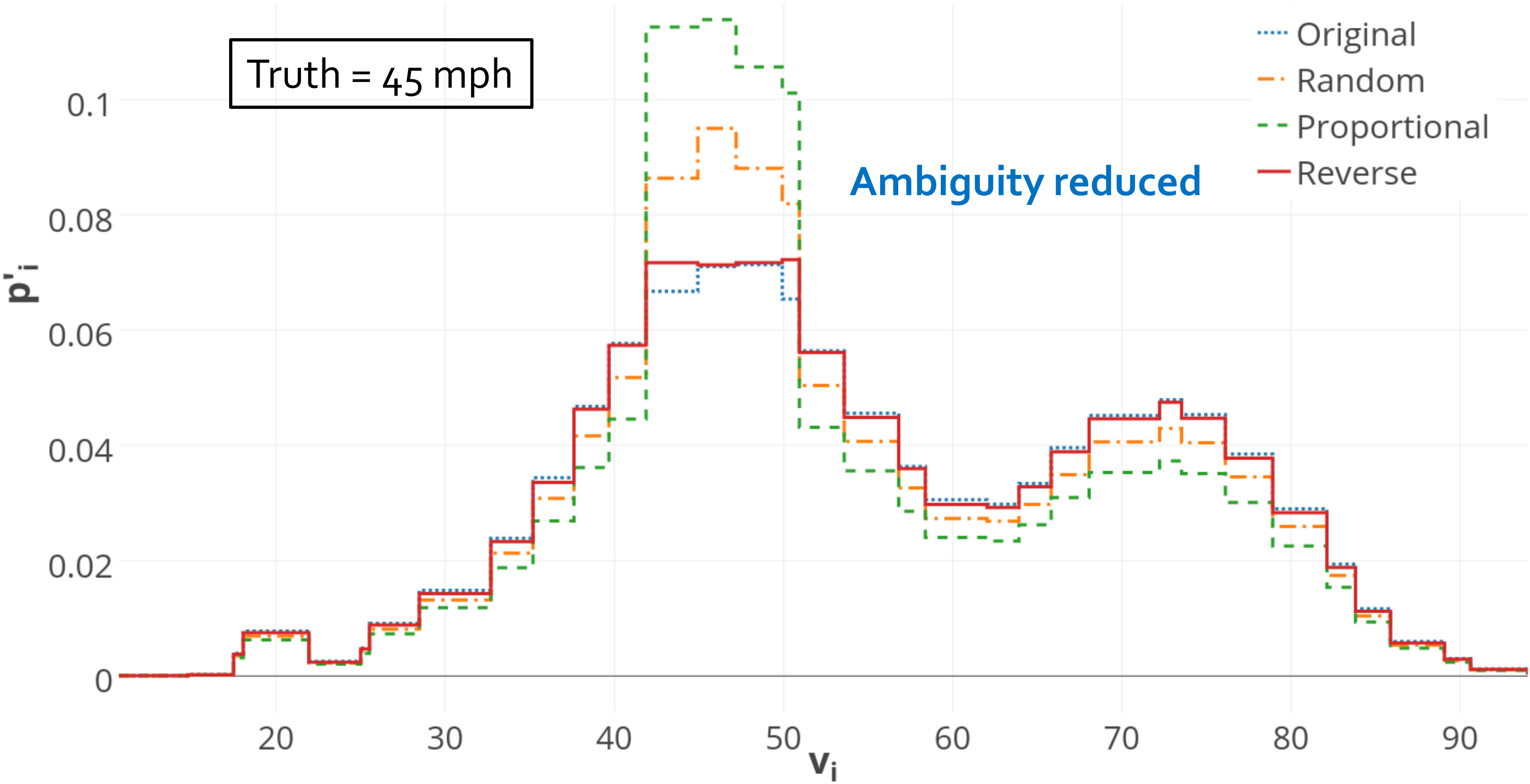
- Application: avg. traffic speed of a major road in CBD
- A platform like *mTurk* has 50,000 registered users
- 1,000 contributors
- Aim to collect m=1,000 ratings from the rest 49,000 users within T_o=1 hour, shortfall tolerance α=0.1
- Raters: commuters who work in the CBD and travelers who frequent the CBD
- Simulate rater behaviors: prob. of accepting /declining offers, distribution of individual beliefs of truth, how each rater rates, delay in response, etc. (details in paper)

Result of Profiling: $\chi=(V,P)$

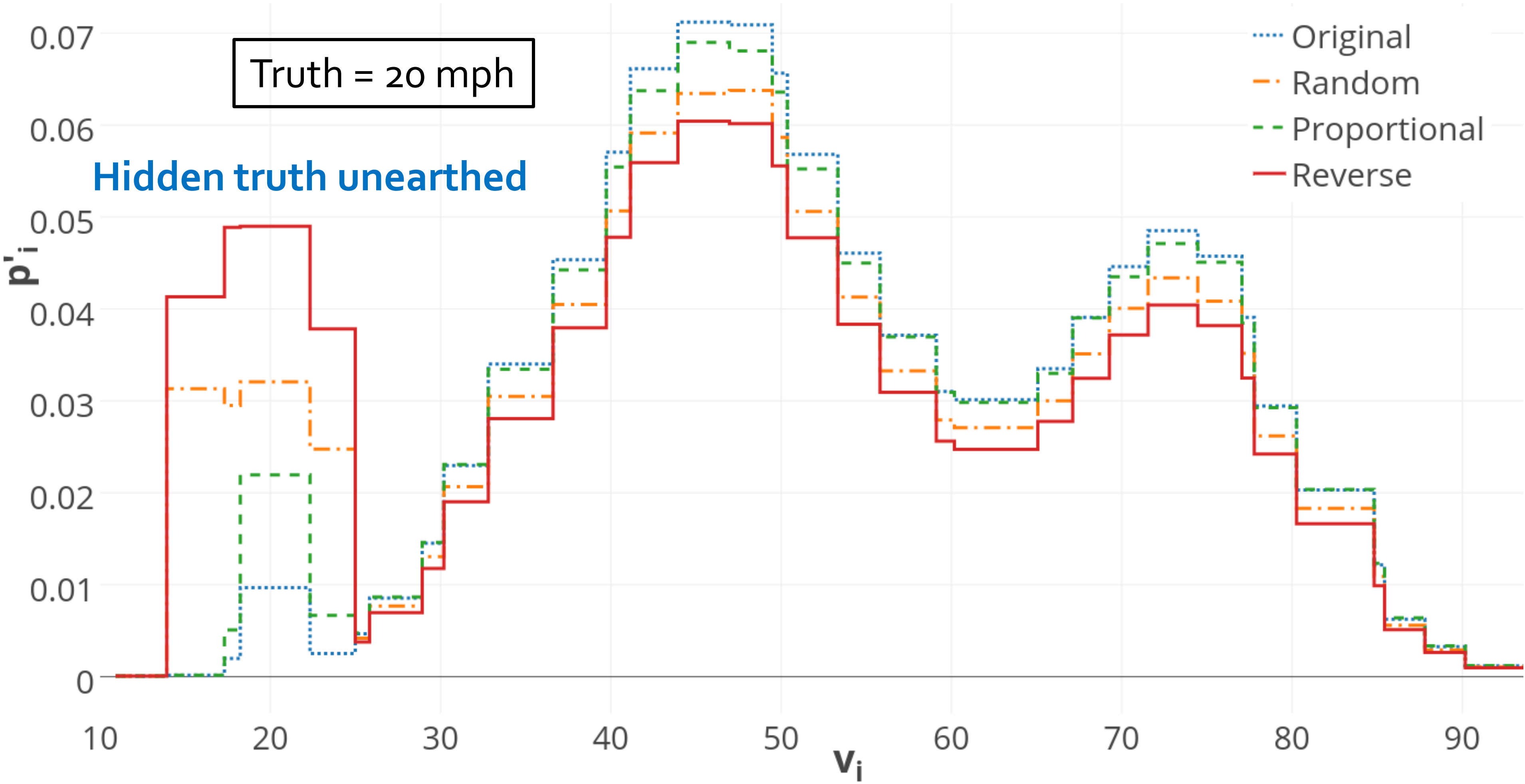


Reshaping mobile crowd sensing using cross validation to improve data credibility [T. Luo and L. Zeynalvand]

Case A: Truth reinforcement



Case B: Scavenging hidden truth



Conclusion

- Cross validation **approach** (general)
  - Further exploits power of crowds: crowd validates crowd
  - "Plug-in" (rather than redesign): co-crowdsourcing
- Cross validation **mechanism** (specific)
  - Profiling + Sampling + PACAP + Reshaping + Incentive
  - Suitable for time-sensitive and quality-critical applications
- **Practicality**:
  - No assumption on (game-theoretical) rationality
  - No assumption on underlying distribution (e.g., Gaussian) of the sensing phenomenon
  - No assumption on single or multiple truths
  - Minimal effort from validators
  - Simple to implement & operate

Thank You!

- Slides can be downloaded at: https://tonylt.github.io (including arXiv version once available)

early 2018