Crowdsourcing with Attributes

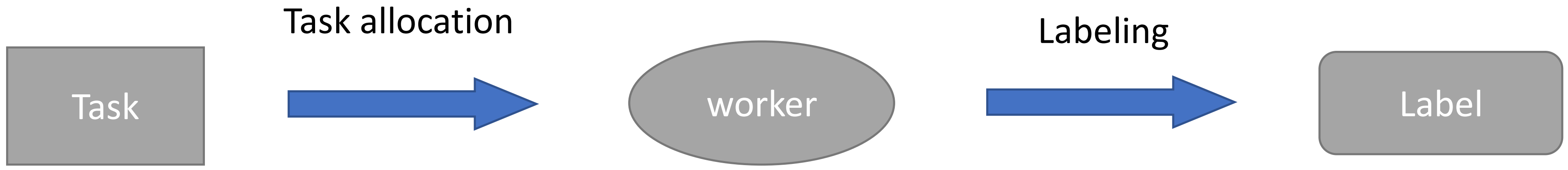
Sung-Wen Lan

2018 10/12

Outline

- Examples
- Multi-class classification with attributes
- Worker Model and Task Allocation Scheme
- KOS algorithm
- Future Work

Crowdsourcing Example (Classification)

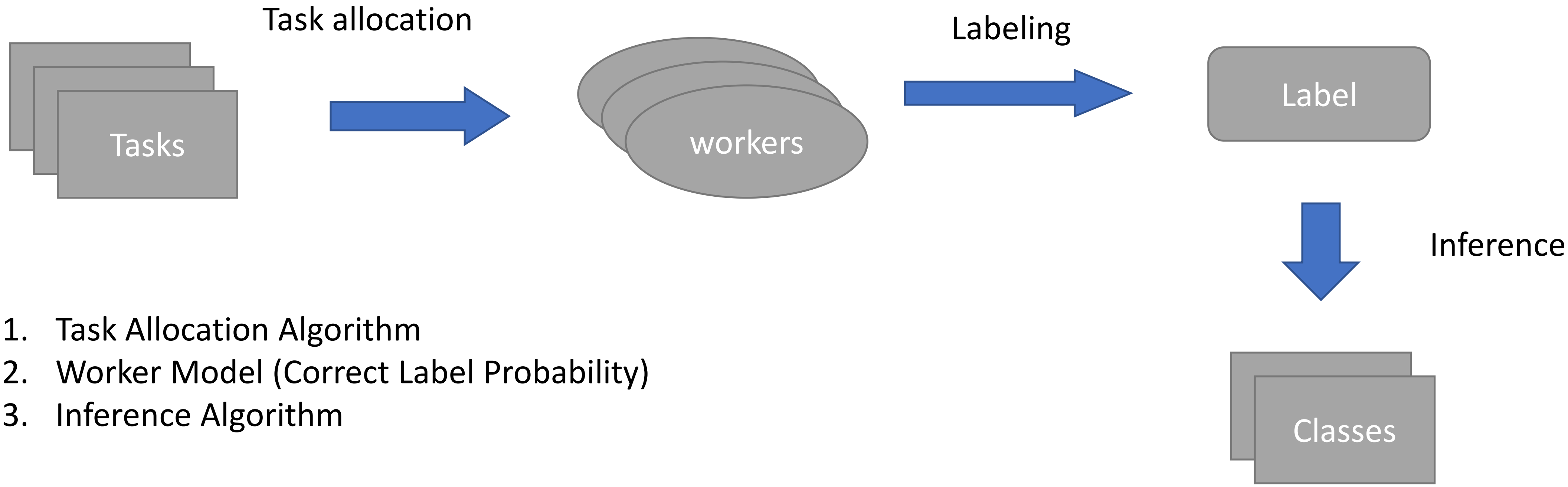


A dog (with dog breeds)

A person familiar with dogs

Shiba Inu?
Corgi?
…

Crowdsourcing Example (Classification)



1. Task Allocation Algorithm
2. Worker Model (Correct Label Probability)
3. Inference Algorithm

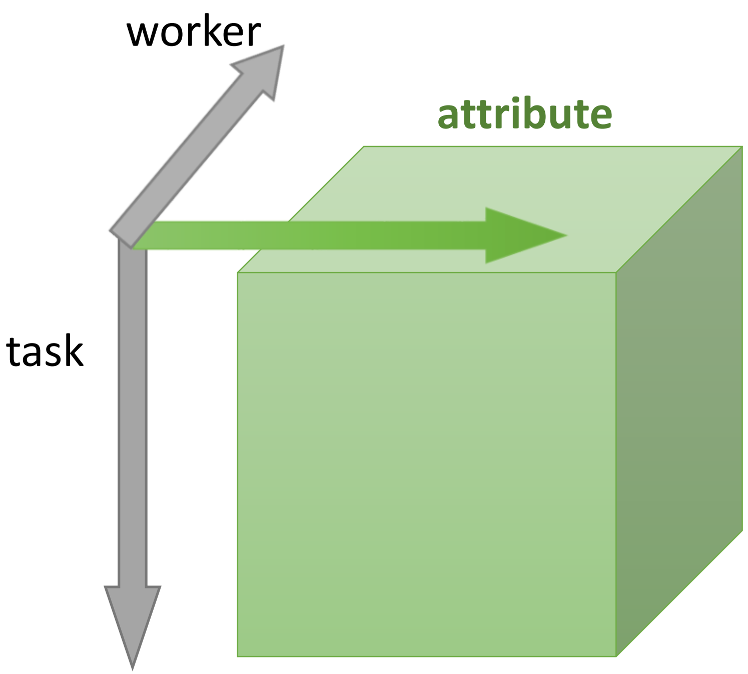
Attributes of Tasks

Attribute	Values		
Size	small	medium	large
Color	black	brown	white
Hair Length	short	medium	long

- If classification problems are too difficult to workers, we could let worker label attributes,
- and then run inference algorithm on attributes.

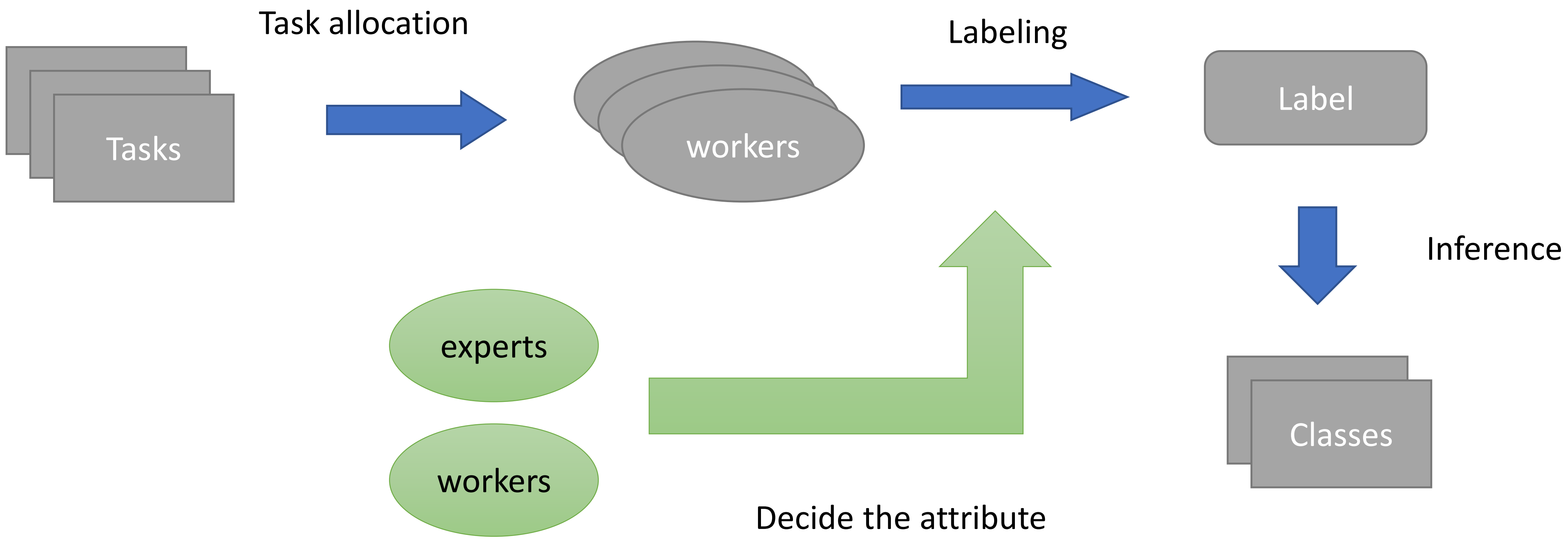
Label Tensor

$$L = \{L_{t,a,w} \in \{+1,0,-1\} \mid \forall t \in [n_t], \forall a \in [n_a], \forall w \in [n_w]\}$$



$L_{t,a,w} = 0$
means worker w does not label attribute a of task t

Crowdsourcing Example (Classification)



Classification Problem Formulation

- Task Allocation

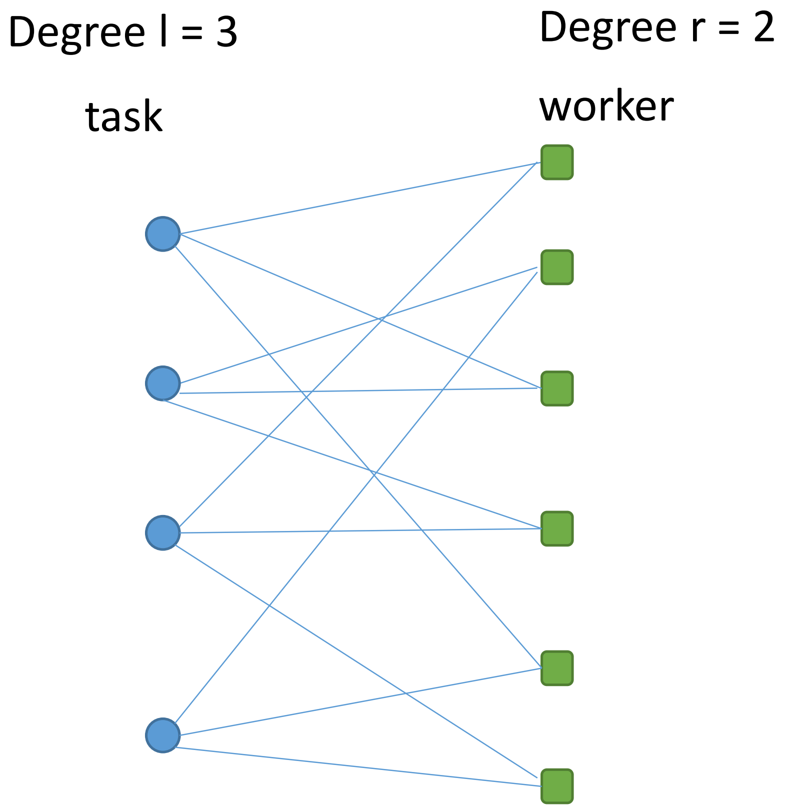
  Random Regular Bipartite Graph (Configuration Model)

- Worker Model

  Consider workers’ label probability are i.i.d sampled from an unknown distribution

- Inference Algorithm

Task Allocation (Configuration Model)



We allocate tasks to workers as a random (l,r)-regular bipartite graph G.

Worker Model

- Given a binary attribute $\boldsymbol{a}$ of a task $\boldsymbol{t}$:

  $$b_{t,a} \in \{+1, -1\}$$

- For worker $\boldsymbol{w}$, the probability of correct labels ($\boldsymbol{p_w}$):
  - p_w is sampled from as unknown distribution, where $\mu \equiv E[2p_w - 1] > 0$
  - The label on the attribute is denoted $L_{a,j}$
  - $\mathrm{P}\{L_{t,a,w} = b_a\} = p_w$

- Define $q \equiv E\left[(2p_w - 1)^2\right]$ as a reliability of all workers

Multi-class Classification Problem

- Given n_a number of attributes that can distinguish n_k classes.

- We collect labeled attributes

$$L = \left[L_{t,a,w} \in \{+1, 0, -1\}\right]_{0 \le t < n_t, 0 \le a < n_a, 0 \le w < n_w}$$

- Infer the class of task $\mathrm{Alg}(L) = \hat{C} = \left[\hat{c}_t \in [n_k]\right]_{0 \le t < n_t}$

- Objective : minimize $\left|\left|\hat{C} - C\right|\right|_1$

Inference Algorithm

- In multi-class classification problem, DS model are often assumed and the EM algorithms are applied.

- We introduce KOS algorithm[1] of binary classification problem

[1]: David R. Karger, Sewoong Oh, Devavrat Shah. Budget-Optimal Task Allocation for Reliable Crowdsourcing Systems, arXiv:1110.3564, 2011

Binary Classification Problem

- Given $n_a = 1$ number of attributes that can distinguish $n_k = 2$ classes.
- We collect labeled attributes

$$L = \left[L_{t,w} \in \{+1,0,-1\}\right]_{0 \le t < n_t, 0 \le w < n_w}$$

- Infer the class of task $\mathrm{Alg}(L) = \hat{C} = [\hat{c}_t \in \{-1,1\}]_{0 \le t < n_t}$
- Objective : minimize $\left|\left|\hat{C} - C\right|\right|_1$

Spectral Method

For binary task $c_t \in \{+1, -1\}$

The expectation of Label Matrix given p_j

$$
\mathrm{E}[L|p] = \frac{l}{n_w}\begin{bmatrix} c_1(2p_1 - 1) & \cdots & c_1(2p_{n_w} - 1) \\ \vdots & \ddots & \vdots \\ c_{n_t}(2p_1 - 1) & \cdots & c_{n_t}(2p_{n_w} - 1) \end{bmatrix}
$$

$$
= \frac{l}{n_w}\begin{bmatrix} t_1 \\ \cdots \\ t_{n_t} \end{bmatrix} \times \left[(2p_1 - 1) \ldots \left(2p_{n_w} - 1\right)\right]
$$

Tasks x Worker

Guess $\begin{bmatrix} c_1 \\ \cdots \\ c_{n_t} \end{bmatrix}$ is close to the eigenvector with the largest eigenvalue of LL^T

KOS Algorithm

- Iterative Algorithm $(n_k = 2, n_a = 1)$

Given Task Allocation E and Labels $\{L_{t,w}\}_{(t,w)\in E}$, n_{iter}

Initialize $y^{(0)}_{w\to t}$ with Gaussian distribution $N(1,1)$

For $\mathrm{n} = 1, \dots, \mathrm{n}_{\mathrm{iter}} - 1$ do

For all $(t, w) \in E\ do\ x^{(n)}_{t\to w} = \sum_{w'\in\partial t\backslash \mathrm{w}} L_{t,w'} y^{(n-1)}_{w'\to t}$;

For all $(t, w) \in E\ do\ y^{(n)}_{w\to t} = \sum_{w'\in\partial w\backslash \mathrm{t}} L_{t'w} x^{(n)}_{t\to w}$;

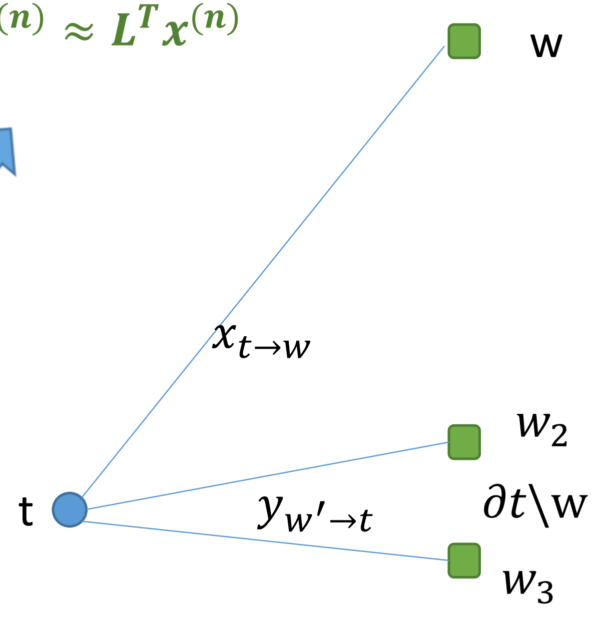
For all $t \in [n_t]$ do $x_t = \sum_{w\in\partial t} L_{t,w} y^{(n_{iter}-1)}_{w\to t}$;

Output $\widehat{c_t} = \mathrm{sign}(x_t)$

$\boldsymbol{x}^{(n)} \approx \boldsymbol{L}\boldsymbol{L}^T\boldsymbol{x}^{(n-1)}$

$\boldsymbol{x}^{(n)} \approx \boldsymbol{L}\boldsymbol{y}^{(n-1)}$

$\boldsymbol{y}^{(n)} \approx \boldsymbol{L}^T\boldsymbol{x}^{(n)}$



KOS Algorithm

We could simply apply KOS algorithm for each attribute, and solve the problem by finding the closest class that matches the inferred attributes.

Another way is to using expert guideline in KOS algorithm.

Future Work

- Extend the KOS algorithm to spectral method in tensor.
- If consider the varying difficulty and importance of attributes, then we could design attribute allocation to improve performance.

Reference

- David R. Karger, Sewoong Oh, Devavrat Shah. Budget-Optimal Task Allocation for Reliable Crowdsourcing Systems, arXiv:1110.3564, 2011

Error Bound of KOS Algorithm(1)

- By proving x_i is sub-Gaussian, we could use Chernoff bound
- Assume binary tasks are all $\{+1\}$

- Since branches of spanning tree are independent, iterative algorithm gives

$$E\left[e^{\lambda x_i^{(n)}}\right] = \left(E_p\left[pE\left[e^{\lambda y_j^{(n-1)}}\middle|p\right] + (1-p)E\left[e^{-\lambda y_j^{(n-1)}}|p\right]\right]\right)^{l-1}$$

$$E\left[e^{\lambda y_j^{(n)}}\right] = \left(pE\left[e^{\lambda x_i^{(n)}}\right] + (1-p)E[e^{-\lambda x_i^{(n)}}]\right).^{r-1}$$

Error Bound of KOS Algorithm(2)

$$P\left\{x_i^{(n)} \le 0\right\} \le E\left[e^{\lambda x_i^{(n)}}\right] \le e^{m_n\lambda+\frac{1}{2}(l-1)\sigma_n^2\lambda^2}$$

$$m_n = \mu(l-1)(q(l-1)(r-1))^{n-1}, \mu\ denotes\ E[2p-1]$$

$$\sigma_n^2 = 2\hat{l}\left(\hat{l}\hat{r}\right)^{n-1} + \mu^2\hat{l}^3\hat{r}(3q\hat{r})^{2n-4}\frac{1-\left(\frac{1}{\mathrm{q}^2\hat{l}\hat{r}}\right)^2}{1-\frac{1}{q^2\hat{l}\hat{r}}}$$

KOS Algorithm

For binary tasks.

$$x_{i \to j} = \sum_{\partial i \backslash \mathrm{j}} A_{i j'} y_{j' \to i}$$

$$y_{j \to i} = \sum_{\partial j \backslash \mathrm{i}} A_{i' j} x_{i' j}$$

KOS Algorithm

For binary tasks.

$$\begin{pmatrix} x^+ \\ x_- \end{pmatrix} = \sum_{\partial i} \begin{pmatrix} A_{ij} \\ -A_{ij} \end{pmatrix} y$$

$$y = \sum_{\partial j} \frac{1}{2} \begin{pmatrix} A_{ij} \\ -A_{ij} \end{pmatrix}^T \begin{pmatrix} x^+ \\ x_- \end{pmatrix}$$

Guidline: $k^+ = 1, k^- = -1$

$$\begin{pmatrix} A_{ij} \\ -A_{ij} \end{pmatrix} \; could\ be\ seen\ as \; \begin{pmatrix} 1 \\ -1 \end{pmatrix} A_{ij} = \begin{pmatrix} k^+(a = 1) \\ k^-(a = 1) \end{pmatrix} A_{ij}^T$$