TLLC: Transfer Learning-based Label Completion for Crowdsourcing

Wenjun Zhang, Liangxiao Jiang*, Chaoqun Li

China University of Geosciences, Wuhan 430074, PR China

The project is available at: https://github.com/jiangliangxiao/TLLC.

Outline

- **Background & Motivation**
- **Problem Description**
- **Method**
- **Experiments**

Crowdsourcing Learning

Crowdsourcing Learning

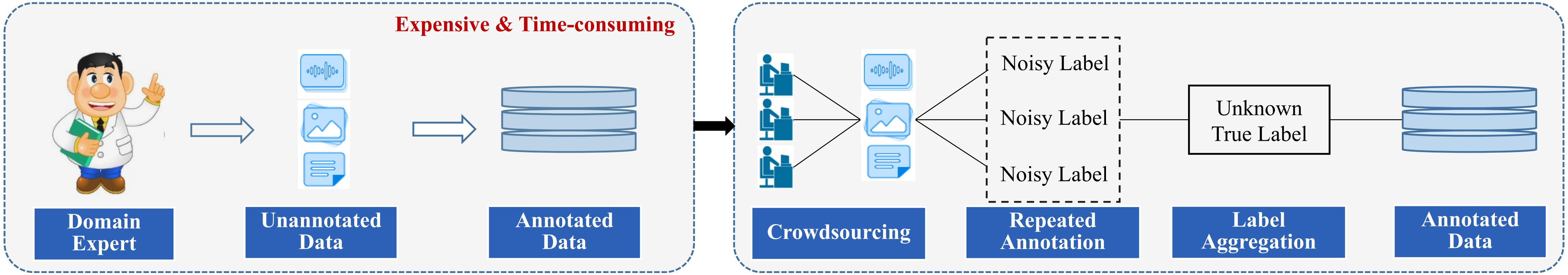
- Crowdsourcing offers a faster and more cost-effective data annotation approach. Although it reduces annotation costs, crowd workers with poor expertise also introduce noise.

Repeated Annotation

- To reduce the impact of noise, repeated annotation has been widely adopted, where each instance is annotated by multiple workers to obtain a multiple noisy label set.

Label Aggregation

- After repeated annotation, label aggregation is applied to infer the unknown true label of each instance based on its multiple noisy label set.



Label Completion

Sparse Label Matrix

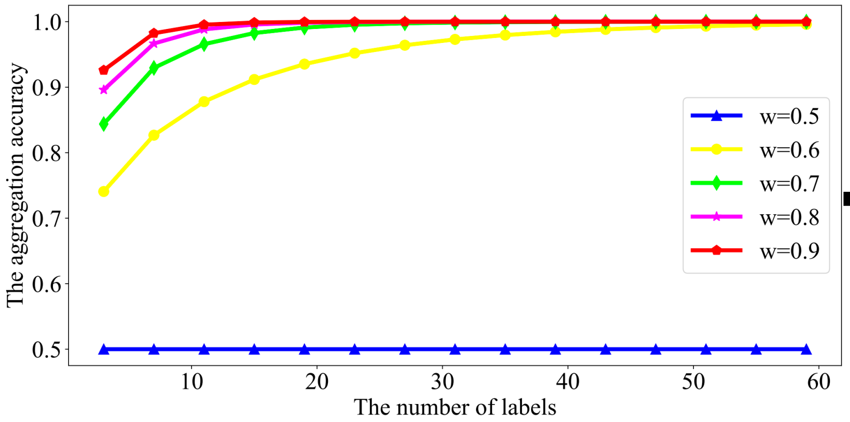
- In real-world scenarios, each worker typically annotates only a small number of instances, and few labels are typically collected per instance to reduce cost, resulting in a highly sparse crowdsourced label matrix.

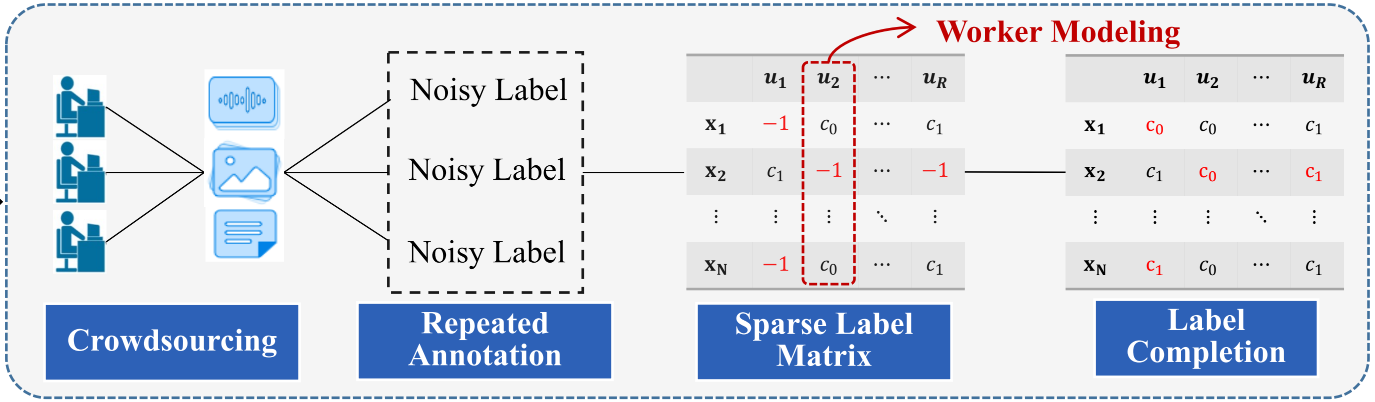
Label Completion

- Label aggregation failing to achieve the expected performance relying solely on the existing labels in the sparse label matrix. Therefore, label completion has been proposed to fill in missing labels in the sparse label matrix.

Worker Modeling

- Worker modeling is effective to improve the performance of label completion. However, insufficient annotated instances may lead to insufficient worker modeling.





Outline

- Background & Motivation
- **Problem Description**
- Method
- Experiments

Preliminary

Basic Notations

- Crowdsourcing data: $D = \{(\boldsymbol{x}_i, \boldsymbol{L}_i)\}_{i=1}^{N}$, where $\boldsymbol{x}_i$ is the i-th instance in D, $\boldsymbol{L}_i$ is the multiple noisy label set of $\boldsymbol{x}_i$, and N is the number of instances.
- Instance: $\boldsymbol{x}_i = \{x_{im}\}_{m=1}^{M}$, where M is the dimension of attributes, x_{im} is the attribute value of $\boldsymbol{x}_i$ on the m-th attribute A_m.
- Multiple noisy label set: $\boldsymbol{L}_i = \{l_{ir}\}_{r=1}^{R}$, where R is the number of workers, and l_{ir} is the label of $\boldsymbol{x}_i$ annotated by the r-th worker u_r.
- Label: l_{ir} takes a value from a fixed set $\{-1, c_1, \dots, c_q, \dots, c_Q\}$, where Q is the number of classes, c_q is the q-th class, and -1 means that u_r does not annotate $\boldsymbol{x}_i$.

	u_1	u_2	$\cdots$	u_R
$\mathbf{x_1}$	-1	c_0	$\cdots$	c_1
$\mathbf{x_2}$	c_1	-1	$\cdots$	-1
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{x_N}$	-1	c_0	$\cdots$	c_1

	u_1	u_2	$\cdots$	u_R
$\mathbf{x_1}$	-1	c_0	$\cdots$	c_1
$\mathbf{x_2}$	c_1	-1	$\cdots$	-1
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{x_N}$	-1	c_0	$\cdots$	c_1

→

	$\hat{y}$
$\mathbf{x_1}$	c_0
$\mathbf{x_2}$	c_1
$\vdots$	$\vdots$
$\mathbf{x_N}$	c_1

Label Aggregation

- Definition 1: Label aggregation infers the unknown true label y_i of each instance $\boldsymbol{x}_i$ based on $\{(\boldsymbol{x}_i, \boldsymbol{L}_i)\}_{i=1}^{N}$, minimizing the error between the aggregated label $\hat{y}_i$ and the unknown true label y_i.

Label Completion

- Definition 2: Label completion infers the missing label $l_{ir} = -1$ of each instance $\boldsymbol{x}_i$ based on $\{(\boldsymbol{x}_i, \boldsymbol{L}_i)\}_{i=1}^{N}$, ensuring that the completed label $\hat{l}_{ir}$ is the most likely label annotated to $\boldsymbol{x}_i$ by worker u_r.

	u_1	u_2	$\cdots$	u_R
$\mathbf{x_1}$	-1	c_0	$\cdots$	c_1
$\mathbf{x_2}$	c_1	-1	$\cdots$	-1
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{x_N}$	-1	c_0	$\cdots$	c_1

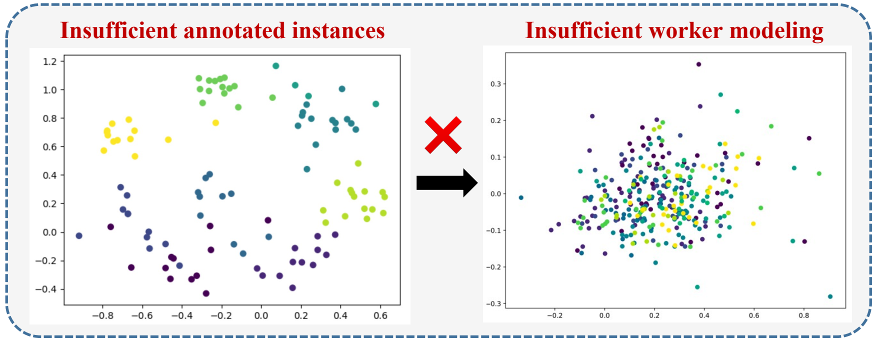
→

	u_1	u_2	$\cdots$	u_R
$\mathbf{x_1}$	c_0	c_0	$\cdots$	c_1
$\mathbf{x_2}$	c_1	c_0	$\cdots$	c_1
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\mathbf{x_N}$	c_1	c_0	$\cdots$	c_1

Problem Description

Limitations

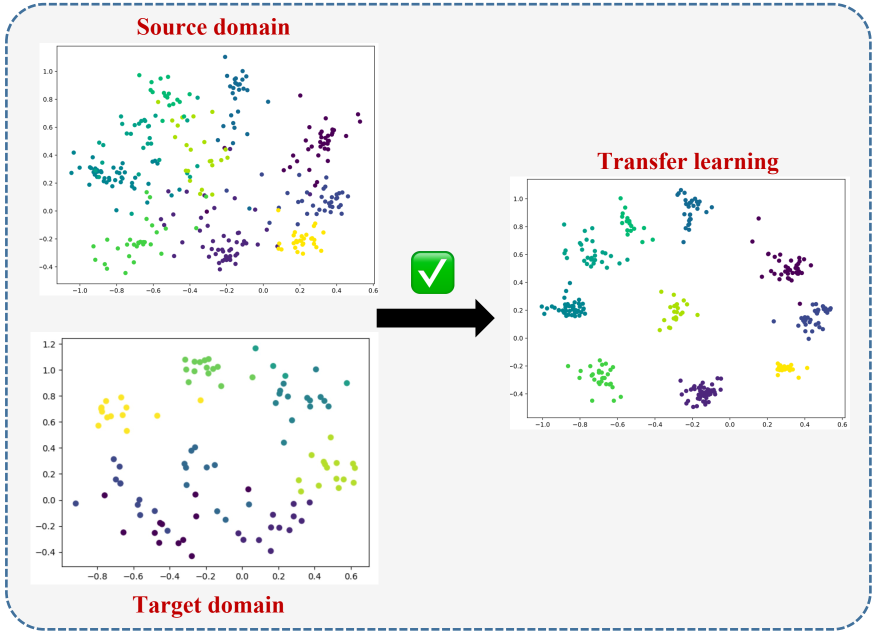
- ➢ Although worker modeling can improve the performance of label complement, it remains constrained by the limited number of instances annotated by each worker.
- ■ Insufficient annotated instances fail to accurately reflect the annotation ability of each worker, leading to insufficient worker modeling.
- ■ Insufficient worker modeling may misguide label completion, thereby limiting the improvement of label completion.



Problem to Be Solved

- ➢ It is reasonable to use transfer learning to address the issue of insufficient worker modeling. However, conducting transfer learning in crowdsourcing requires addressing:

- ■ How to construct the source and target domains from a given crowdsourced data?
- ■ How to perform worker modeling via transfer learning?
- ■ How to perform label completion?



Outline

- ☐ Background & Motivation
- ☐ Problem Description
- ☐ **Method**
- ☐ Experiments

Main Idea

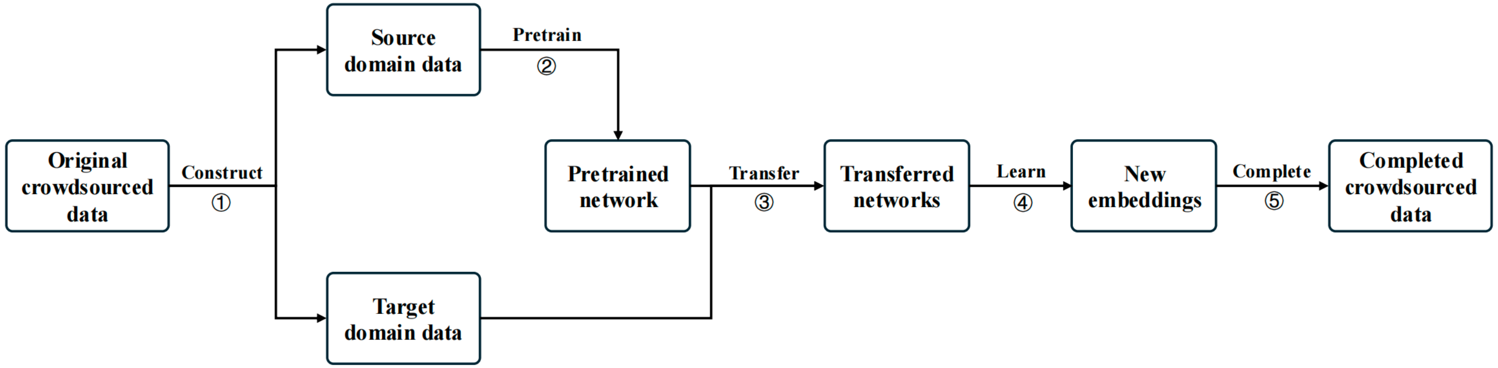
Algorithm 1 Source and Target Domains Construction

Require: crowdsourced data D.
Ensure: source and target domain data: $D_S, \{D_T^r\}_{r=1}^R$.
for $i = 1$ to N **do**
  Calculate $\hat{y}_i$ and $P(c_q|\boldsymbol{L}_i)$ by Equations (1) and (2);
end for
for $q = 1$ to Q **do**
  Calculate μ_{c_q} by Equation (3);
end for
Construct $\boldsymbol{X}_S$ by Equation (4);
Construct D_S by Equation (5);
for $r = 1$ to R **do**
  Construct D_T^r by Equation (6);
end for
return $D_S, \{D_T^r\}_{r=1}^R$.

Contribution 1 (Step ①):

1） We design a novel algorithm to construct the source and target domains from crowdsourced data, which makes it possible to introduce transfer learning into crowdsourcing.

2） Instance filtering based on confident learning ensures the quality of the source domain.



Contribution 3 (Steps ④⑤) :

1） We leverage the new embeddings learned by the transferred network to complete each worker's missing labels.

2） The proposed completion condition effectively avoids unreasonable completions.

Algorithm 3 Label Completion

Require: crowdsourced data D, networks $\{f_T^r\}_{r=1}^R$.
Ensure: completed crowdsourced data $\hat{D}$.
for $r = 1$ to R **do**
  Construct $\boldsymbol{X}^r$ and $\bar{\boldsymbol{X}}^r$ using D;
  for $i = 1$ to $|\boldsymbol{X}^r|$ **do**
    Learn $\boldsymbol{z}_i^r$ for X_i^r by Equation (9);
  end for
  for $q = 1$ to Q **do**
    Calculate $\bar{\boldsymbol{z}}_q^r$ for c_q by Equation (16);
    Calculate $\bar{d}_q^r$ for c_q by Equation (17);
  end for
  for $i = 1$ to $|\bar{\boldsymbol{X}}^r|$ **do**
    Learn $\boldsymbol{z}_i^r$ for $\bar{X}_i^r$ by Equation (9);
    for $q = 1$ to Q **do**
      if Equation (19) holds **then**
        Complete a label $\hat{l}_{ir} = c_q$ for $\bar{\boldsymbol{X}}^r$;
        break;
      end if
    end for
  end for
end for
Reconstruct $\hat{D}$ with $\{\boldsymbol{X}^r\}_{r=1}^R$ and $\{\bar{\boldsymbol{X}}^r\}_{r=1}^R$;
return $\hat{D}$.

Algorithm 2 Worker Modeling

Require: source and target domain data: $D_S, \{D_T^r\}_{r=1}^R$.
Ensure: transferred networks $\{f_T^r\}_{r=1}^R$.
Generate data D_S' using D_S by Equation (8);
Pretrain f_S using D_S' by Equation (11);
Share the parameters of f_S with f_T;
for $r = 1$ to R **do**
  Copy f_T as f_T^r;
  Generate data $D_T^{r'}$ using D_T^r by Equation (8);
  Fine-tune f_T^r using $D_T^{r'}$ by Equation (11);
end for
return $\{f_T^r\}_{r=1}^R$.

Contribution 2 (Steps ②③):

1） We train Siamese networks to model workers through transfer learning, which significantly mitigates the impact of insufficient worker modeling.

2） Siamese networks are suitable for small-sample scenarios with insufficient annotated instances.

Theoretical Analysis

➢ **Theorem 1:** Constructing source domain based on Equation (5) can reduce the generalization error in transfer learning.

$$\boldsymbol{X}_S = \{\boldsymbol{x}_i | P(\hat{y}_i | \boldsymbol{L}_i) \geq \mu_{\hat{y}_i}, \text{for } i = 1, 2, \ldots, N\} \quad (4)$$

$$D_S = \{(X_{Si}, l_{Si}) \mid \text{for } i = 1, 2, \ldots, |\boldsymbol{X}_S|\} \quad (5)$$

$$D_T^r = \{(X_i^r, L_i^r) \mid \text{for } i = 1, 2, \ldots, |\boldsymbol{X}^r|\} \quad (6)$$

$$\epsilon_T \leq \epsilon_S + L^1(\mathcal{D}_S, \mathcal{D}_T) + \lambda \quad (7)$$

➢ **Theorem 2:** Parameter-based transfer learning can reduce the generalization error in worker modeling.

➢ **Theorem 3:** When the noise in D' follows an independent and identically distributed (i.i.d.) Gaussian distribution, worker modeling is robust to noise.

$$y' = y_t' + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \quad (12)$$

$$\begin{aligned}\mathcal{L}_{mse} &= \mathbb{E}[(y' - d')^2] = \mathbb{E}[(y_t' + \epsilon - d')^2] \\ &= \mathbb{E}[(y_t' - d')^2] + 2\mathbb{E}[(y_t' - d')\epsilon] + \mathbb{E}[\epsilon^2]\end{aligned} \quad (13)$$

$$\mathcal{L}_{mse} = \mathbb{E}[(y_t' - d')^2] + \sigma^2 \quad (14)$$

Outline

- Background & Motivation
- Problem Description
- Method
- **Experiments**

Experimental Settings

Baseline

- The state-of-the-art WSLC (Wu et al., 2024) employs worker modeling and supports multi-class crowdsourcing scenarios, making it a key baseline for comparing with our proposed TLLC.

Real-world datasets

- Income: A binary classification dataset collected from Amazon Mechanical Turk (AMT), in which the proportion of missing labels is 0.85.
- Leaves: A multi-class classification dataset collected from Amazon Mechanical Turk (AMT), in which the proportion of missing labels is 0.88.
- Music_genre: A multi-class classification dataset collected from Amazon Mechanical Turk (AMT), in which the proportion of missing labels is 0.90.

Table 5. Descriptions of the three real-world datasets used in our experiments.

Dataset	*Income*	*Leaves*	*Music_genre*
#Instances	600	384	700
#Workers	67	83	44
#Labels	6000	3840	2946
#Attributes	10 (nominal)	64 (numeric)	31 (numeric)
#Classes	2	6	10
Averaged #Labels per instance	10	10	4.2
Proportion of missing labels	0.85	0.88	0.90

Label aggregation methods

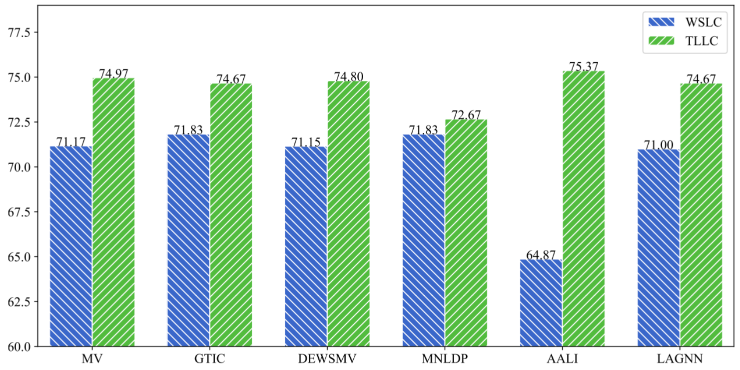
- MV: majority voting (Sheng et al., 2008)
- GTIC: ground truth inference using clustering (Zhang et al., 2016)
- DEWSMV: differential evolution-based weighted soft majority voting (Tao et al., 2021)
- MNLDP: multiple noisy label distribution propagation (Jiang et al., 2022)
- AALI: attribute augmentation-based label integration (Zhang et al., 2023)
- LAGNN: label aggregation with graph neural networks (Ying et al., 2024)

Metrics

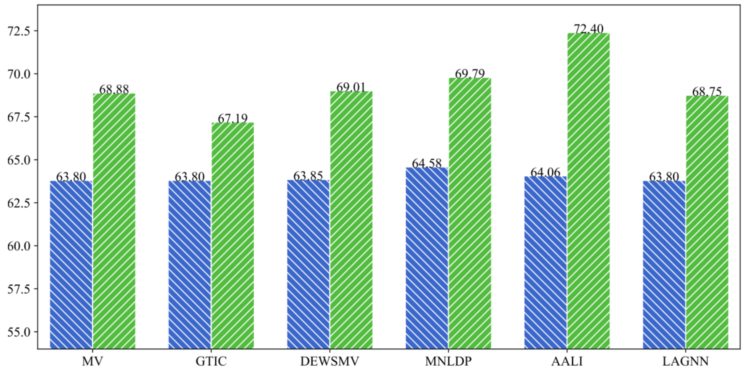
- We evaluate WSLC and TLLC by completing the same crowdsourced datasets and measuring the aggregation accuracy of label aggregation methods on their completed datasets.
- To reduce the impact of randomness on the experimental results, we independently repeat the experiments on each dataset ten times.

Effectiveness

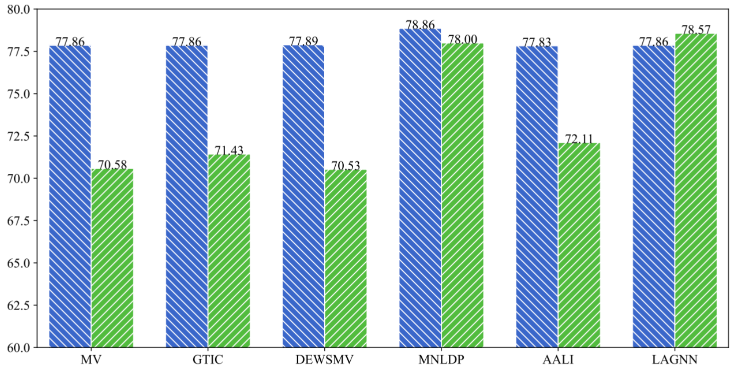
- **Setting:** For each dataset, we compare the average aggregation accuracy over ten experiments and conduct Friedman tests (with Nemenyi test as post-hoc tests) using the results of these ten experiments.



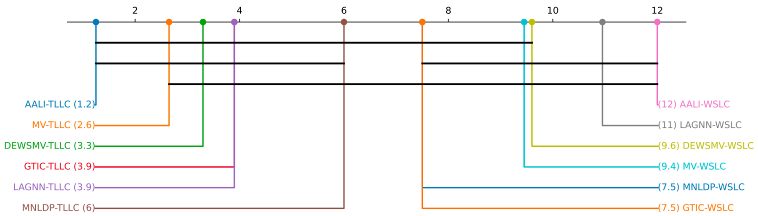
(a) *Income*



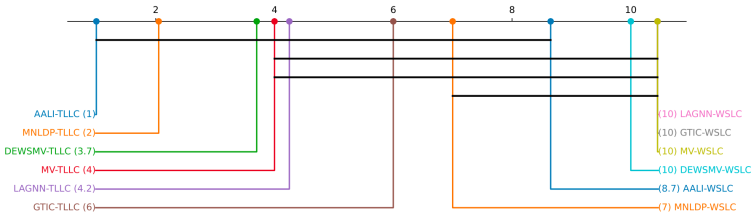
(b) *Leaves*



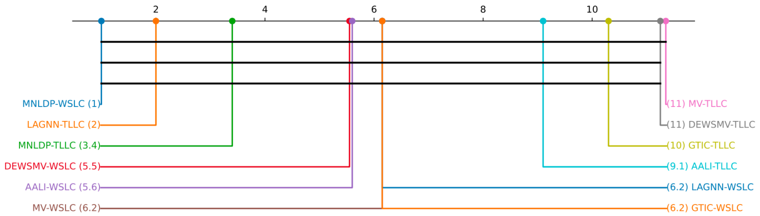
(c) *Music_genre*



(a) *Income*



(b) *Leaves*

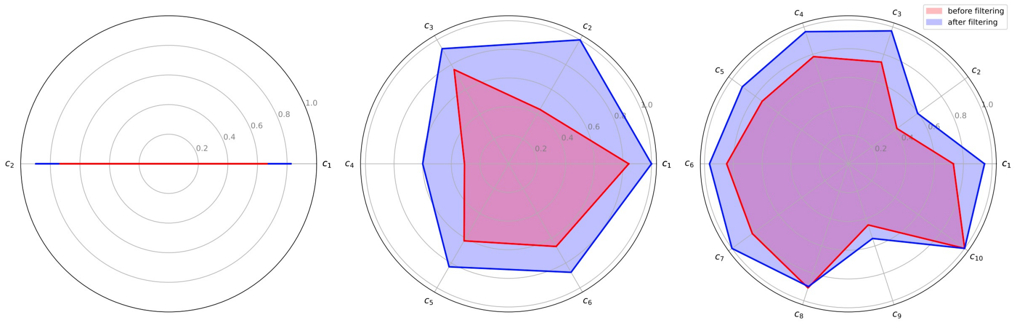


(c) *Music_genre*

Results: TLLC can achieve better aggregation accuracy with high significance.

Rationality

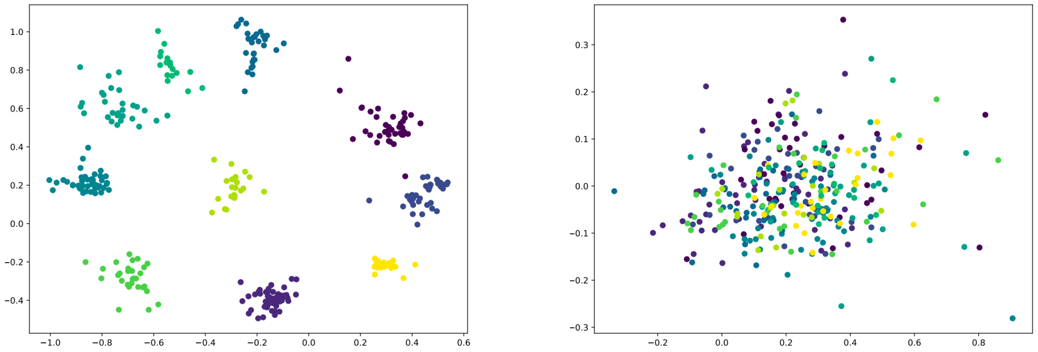
Setting: 1) We compare the aggregation accuracies in X (before filtering) and X_S (after filtering) for each class across three datasets.



(a) *Income* (b) *Leaves* (c) *Music_genre*

1) Results: After filtering, the aggregation accuracies for almost all classes across all datasets are significantly improved.

Setting: 2) We illustrate the new embeddings learned by the Siamese network corresponding to a worker.



(a) *With transfer learning* (b) *Without transfer learning*

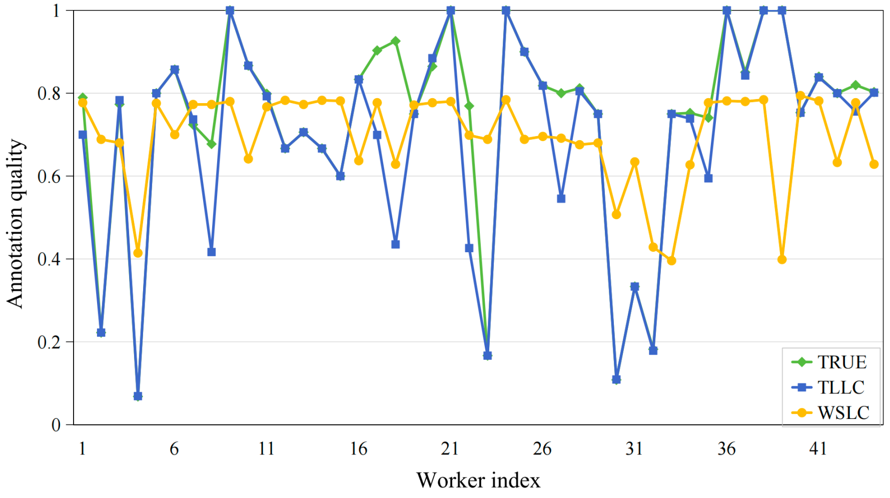
2) Results: The method with transfer learning better clusters instances with the same true labels.

Rationality

Setting: 3) We analyze the changes in annotation quality of workers before and after label completion.

(a) *Income*

(b) *Leaves*

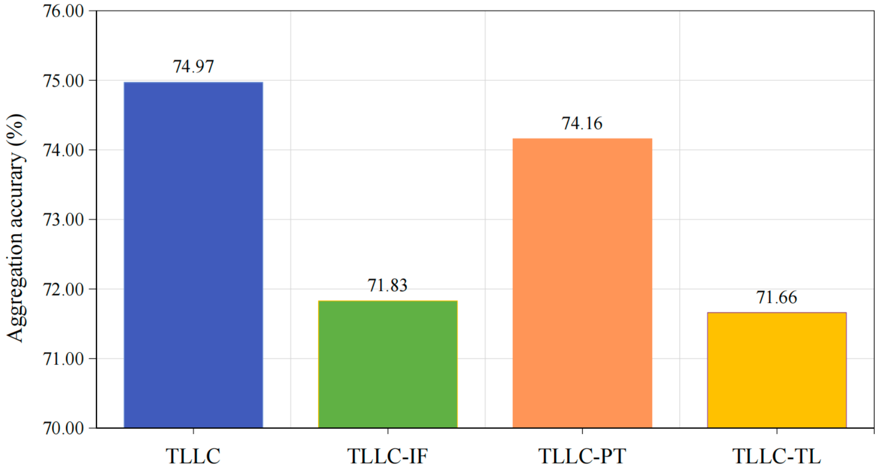


(c) *Music_genre*

3) Results: TLLC maintains smaller changes in workers' annotation quality, preserving their unique characteristics.

Other Analysis

Ablation experiment: We conduct an ablation experiment on dataset Income to investigate the impact of different strategies on TLLC's performance,.



Results: Instance filtering, pretraining, and transfer learning are all important in enhancing TLLC's performance.

Sensitivity analysis: We conduct sensitivity analysis experiments on dataset Income to observe the impact of these parameters on TLLC's performance.

	New embedding dimension				
Value	2	4	6	8	10
Accurary (%)	**74.94**	71.83	71.66	73.33	72.66
	Epochs				
Value	2	4	6	8	10
Accurary (%)	**74.94**	72.33	73.00	72.16	72.83
	Batch size				
Value	8	16	32	64	128
Accurary (%)	71.83	72.50	**74.94**	73.33	73.16

Results: The effectiveness of TLLC is not highly sensitive to parameter settings.

Conclusion

Contributions

- This paper is the first to reveal the limitations of insufficient worker modeling on label completion.
- We design a novel algorithm to construct the source and target domains from crowdsourced data, which makes it possible to introduce transfer learning into crowdsourcing.
- We train Siamese networks to model workers through transfer learning, which significantly mitigates the impact of insufficient worker modeling.
- Both the theoretical analysis and experimental results validate the effectiveness and rationality of the TLLC we proposed.

Limitations

- TLLC lacks robustness when dealing with adversarial workers who provide numerous labels.

ICML
International Conference
On Machine Learning

CUG-Miner

Thank you for your listening!

Paper

Code

Google Scholar

Feel free to contact us: wjzhang@cug.edu.cn; ljiang@cug.edu.cn