Intelligent Automation & Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2023.036794
Article

P&T-Inf: A Result Inference Method for Context-Sensitive Tasks in Crowdsourcing

Zhifang Liao[1], Hao Gu[1], Shichao Zhang[1], Ronghui Mo[1] and Yan Zhang[2,*]

[1]School of Computer Science and Engineering, Central South University, Changsha, 410075, China
[2]Glasgow Caledonian University, Glasgow, G4 0BA, UK
*Corresponding Author: Yan Zhang. Email: yan.zhang@gcu.ac.uk

Received: 12 October 2022; Accepted: 06 January 2023

Abstract: Context-Sensitive Task (CST) is a complex task type in crowdsourcing, such as handwriting recognition, route plan, and audio transcription. The current result inference algorithms can perform well in simple crowdsourcing tasks, but cannot obtain high-quality inference results for CSTs. The conventional method to solve CSTs is to divide a CST into multiple independent simple subtasks for crowdsourcing, but this method ignores the context correlation among subtasks and reduces the quality of result inference. To solve this problem, we propose a result inference algorithm based on the Partially ordered set and Tree augmented naive Bayes Infer (P&T-Inf) for CSTs. Firstly, we screen the candidate results of context-sensitive tasks based on the partially ordered set. If there are parallel candidate sets, the conditional mutual information among subtasks containing context information in external knowledge (such as Google n-gram corpus, American Contemporary English corpus, etc.) will be calculated. Combined with the tree augmented naive (TAN) Bayes model, the maximum weighted spanning tree is used to model the dependencies among subtasks in each CST. We collect two crowdsourcing datasets of handwriting recognition tasks and audio transcription tasks from the real crowdsourcing platform. The experimental results show that our approach improves the quality of result inference in CSTs and reduces the time cost compared with the latest methods.

Keywords: Crowdsourcing; result inference; tree augmented naive Bayes; context-sensitive

1 Introduction

Crowdsourcing is a process of publishing tasks that are difficult for machines but easy for humans to handle to the Internet, openly recruiting unknown crowds, and leveraging the collective wisdom of crowds to solve similar problems. At present, crowdsourcing has been successfully applied in different fields, such as privacy protection [1], social networks [2], data management [3], software testing [4], etc. The core problem of quality control in crowdsourcing is to infer high-quality results from noisy workers' answers. In crowdsourcing tasks, Context-Sensitive Task (CST) [5] is a complex task composed of a group of context-related simple subtasks. For example, in handwriting recognition tasks, each handwritten sentence recognition task is composed of many handwritten word recognition subtasks, and words in the sentence are context-related. The current result inference algorithms can be divided into two types according to task granularity: Task-Inf [6] and Subtask-Inf [7]. Task-Inf assigns each task to multiple workers and then infers results by machine learning or group voting [8]. Subtask-Inf divides complex tasks into several subtasks, crowdsources subtasks, infers the results of each subtask by task-level reasoning, and summarizes the optimal results of each subtask to generate the final answer to complex tasks [9].

For crowdsourcing quality control, both types of methods mentioned above are not suitable for CSTs [9]. Firstly, it is quite difficult to answer a complex CST completely and correctly. Therefore, the Task-Inf methods cannot obtain high-quality results by inferring workers' answer aggregation. Secondly, a CST cannot be directly split into multiple independent subtasks, because subtasks are interrelated in a specific context [5,9–11]. For example, the handwriting recognition task shown in Fig. 1 is to recognize the handwritten sentence "President Kennedy flew from London Airport last night to arrive in Washington this morning". Individual subtask t_9 is difficult to identify, but if workers consider the results of subtasks $t_3 \sim t_5$ are "from London Airport", it is easy to infer the results of subtasks $t_8 \sim t_9$ are "to arrive". Subtask-Inf methods ignore the correlation among subtasks, which will reduce the quality of result inference.

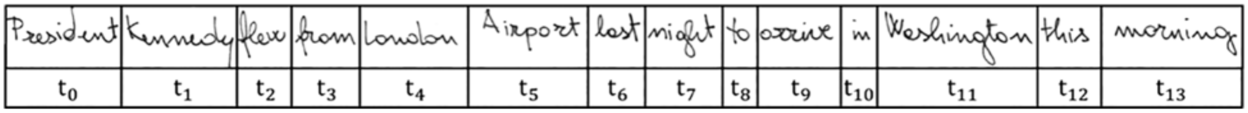


Figure 1: A handwriting recognition task

To solve the above problems in CSTs, we propose a novel algorithm based on the Partially ordered set and Tree augmented naive Bayes Infer (P&T-Inf).

The main contributions of the paper are as follows:

1. We construct a CST dependency tree combined with the context information in external knowledge to model the context correlation of CSTs.
2. We propose a probabilistic model based on tree augmented naive Bayes to describe the crowdsourcing process of CSTs.
3. We design a context-related result inference method P&T-Inf based on the partially ordered set and tree augmented naive (TAN) Bayes for CSTs.
4. We evaluate our method on the crowdsourcing platform Appen. The experimental results demonstrate that compared with the latest methods, our method can effectively improve the quality of CST results with a lower time complexity.

2 Related Work

In general, the quality of crowdsourcing mainly depends on the quality of results inferred from the noisy results provided by workers. The result inference algorithms of crowdsourcing tasks can be divided into two types according to task granularity: Task-inf and Subtask-Inf.

Task-Inf algorithms assign each task to multiple workers and then infer the results using machine learning or group voting [12]. For example, Zhang J analyzed the classical algorithms of DS, GLAD [13], RY [14], and ZenRow [15] based on expectation maximization (EM). They analyzed the performance differences of these four algorithms and proposed a non-expectation maximization (non-EM) adaptive weighted majority voting algorithm. It is quite difficult to answer a complex CST completely and correctly, so Task-inf algorithms cannot obtain high-quality results.

Subtask-Inf algorithms divide complex tasks into several subtasks, crowdsource subtasks independently, and then summarize the optimal results of subtasks to generate the final answer to complex tasks [16]. For example, Tran Thanh [17] proposed a Find-Fix-Verify (FFV) workflow for crowdsourcing tasks. The workflow is used to correct and shorten text in a three-step strategy. Different groups of workers are involved in three steps. The 'Find' step identified some mistakes in sentences and the 'Fix' step fixed these mistakes. Finally, the 'Verify' step verified these mistakes and aggregated the results. Because subtasks are interrelated in a specific context, a CST should not be directly divided into multiple independent subtasks, and Subtask-Inf algorithms are not suitable for the result inference in CSTs.

Sun [5] proposed the concept of CST, and the Context-Inf algorithm in the latest study [9] combined Hidden Markov Model (HMM) with Maximum Likelihood Estimation (MLE) algorithm and EM algorithm to infer results. Context-Inf assumes that the ground truth of a CST is a series of unobservable hidden state sequences in HMM, and the collected workers' answers are HMM observable sequences. The context correlation information among subtasks determines the transition probability of HMM hidden state. The workers' answer similarity is calculated for the output probability of HMM. HMM assumes that subtasks in CSTs are a series of sequential adjacent correlation, and the probability distribution of the ground truth of the next subtask in the crowdsourcing model can only be determined by the ground truth of the current subtask. However, subtasks in CSTs in reality do not necessarily have sequential adjacent correlation, but more complex correlation models.

3 Algorithm Flow

Aiming at the problem that subtasks in CSTs do not have necessarily sequential adjacent correlation, which leads to certain defects in the result inference, we propose a context correlation result inference algorithm P&T-Inf. Firstly, we construct worker output matrices and screen the candidate output results based on the partially ordered set. Then we combine the external knowledge and tree augmented naive Bayes to model the context correlation of CSTs. Finally, we infer the results combined with the probabilistic model, which improves the accuracy of result inference and reduces the time cost.

P&T-Inf algorithm flow is shown in Fig. 2, which is divided into four main steps.

Step 1: Publish tasks and collect answers. Independent subtasks lose context-sensitive information, so we publish the CST as a whole to the crowdsourcing platform. Each CST is answered by multiple workers. A CST can be divided into multiple context-related subtasks. For example, we can split the handwriting recognition task into context-related recognition subtasks, workers can selectively identify these subtasks (i.e., certain words in the sentence). The crowdsourcing platform is responsible for issuing CSTs, collecting worker answers, and then constructing worker output matrices according to the division of subtasks.

Step 2: Filter candidate results. The workers' answers for each subtask are extracted, arranged, and combined without repeated to obtain all possible candidate results, and then the candidate results are filtered based on the partially ordered set.

Step 3: Model context correlation. Based on TAN, each subtask is regarded as an attribute of the CST. Combined with the context information in external knowledge, the conditional mutual information between any two attributes of the CST is calculated. The CST dependency tree is constructed to model the CST context correlation in CSTs.

Step 4: Infer results. A probabilistic model is established to describe the crowdsourcing process of subtasks in CSTs, combined with the CST dependency tree into MLE and EM to infer the results.

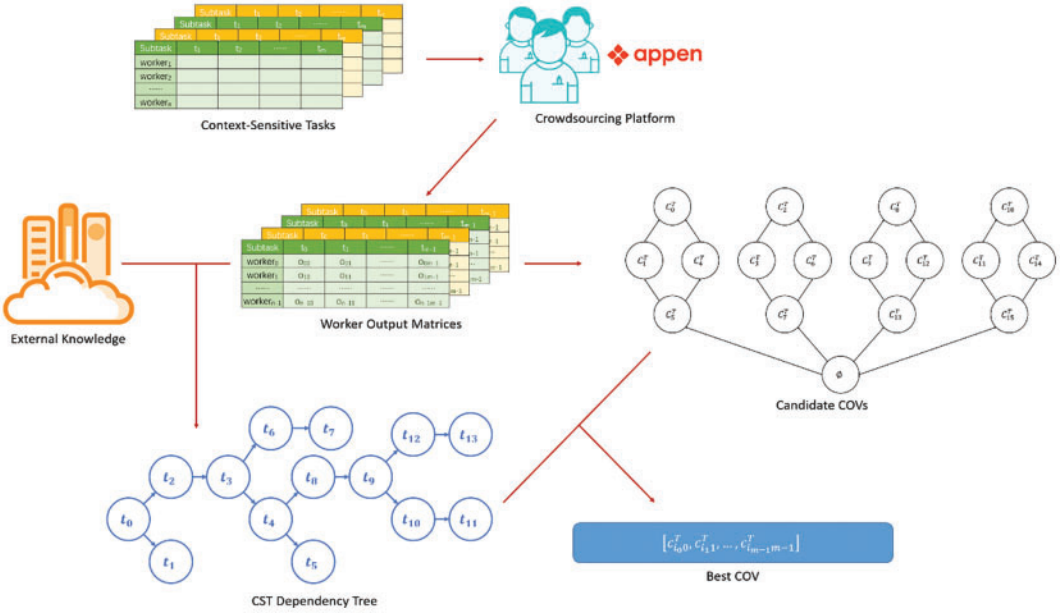

Context-Sensitive Tasks
Subtask
worker
Crowdsourcing Platform
appen
External Knowledge
Worker Output Matrices
Candidate COVs
CST Dependency Tree
t_0 t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t_{10} t_{11} t_{12} t_{13}
$[c^T_{i_00}, c^T_{i_11}, \ldots, c^T_{i_{m-1}m-1}]$
Best COV

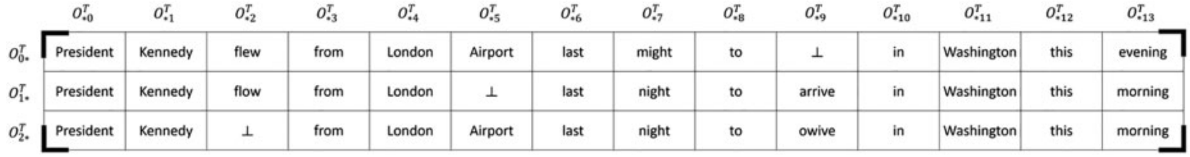
Figure 2: P&T-Inf algorithm flow

3.1 Publish Tasks and Collect Answers

A CST contains multiple subtasks, which are related to each other in a specific context. If these subtasks of a CST are split and crowdsourced separately, the context information will be lost, making it more difficult for workers to identify. We publish the CST as a whole to the crowdsourcing platform. Each CST is answered by multiple workers, and finally, collect the answers of workers. The formulaic definitions of the result inference problem in the P&T-Inf algorithm are as follows:

Definition 1: Worker Output Matrix. As shown in Fig. 3, suppose the CST $T = \{t_0, t_1, \ldots, t_{m-1}\}$ shown in Fig. 1 composed of m subtasks is assigned to n workers to answer. During the recognition, workers can choose to recognize part of words in the sentence, and then we get the recognition results of n workers. Finally, the workers' answers are expressed as an $n \times m$ matrix $O^T = \left\{o^T_{ij}\right\}_{n\times m}$. o^T_{ij}means the answer of worker i ($i \in \{0, 1, \ldots n-1\}$) to subtask t_j ($j \in \{0, 1, \ldots m-1\}$). If worker i does not provide the answer to subtask t_j, we set $o^T_{ij} = \perp$. For O^T, the output vector of worker i is $O^T_{i*} = \left[o^T_{i0}, o^T_{i1}, \ldots, o^T_{im-1}\right]$.

Definition 2: Candidate Output Vector. As shown in Fig. 3, for output vector $O^T_{*j} = \left[o^T_{0j}, o^T_{1j}, \ldots, o^T_{n-1j}\right]$ of the subtask, we remove duplicate outputs in O^T_{*j} to get candidate output set $C^T_{*j} = \left\{c^T_{i_jj} = o^T_{ij} | c^T_{i_jj} \in O^T_{*j}\right\}$ $(0 \le i_j \le n-1)$ of subtask t_j. Suppose $C^T = \left[c^T_{i_00}, c^T_{i_11}, \ldots, c^T_{i_{m-1}m-1}\right]$ is a Candidate Output Vector (COV) of T. Permuting and combining the output in C^T_{*j} of each subtask t_j of T, as shown in Table 1, we can get the set of all possible COVs of T. The problem of inferring the results of CST is transformed into a problem of identifying the optimal COV from all possible COVs of a CST.



$$\begin{array}{c|cccccccccccccc} & O^T_{\cdot 0} & O^T_{\cdot 1} & O^T_{\cdot 2} & O^T_{\cdot 3} & O^T_{\cdot 4} & O^T_{\cdot 5} & O^T_{\cdot 6} & O^T_{\cdot 7} & O^T_{\cdot 8} & O^T_{\cdot 9} & O^T_{\cdot 10} & O^T_{\cdot 11} & O^T_{\cdot 12} & O^T_{\cdot 13} \\ \hline O^T_{0\cdot} & \text{President} & \text{Kennedy} & \text{flew} & \text{from} & \text{London} & \text{Airport} & \text{last} & \text{might} & \text{to} & \perp & \text{in} & \text{Washington} & \text{this} & \text{evening} \\ O^T_{1\cdot} & \text{President} & \text{Kennedy} & \text{flow} & \text{from} & \text{London} & \perp & \text{last} & \text{night} & \text{to} & \text{arrive} & \text{in} & \text{Washington} & \text{this} & \text{morning} \\ O^T_{2\cdot} & \text{President} & \text{Kennedy} & \perp & \text{from} & \text{London} & \text{Airport} & \text{last} & \text{night} & \text{to} & \text{owive} & \text{in} & \text{Washington} & \text{this} & \text{morning} \end{array}$$

Figure 3: Worker output matrix

Table 1: The set of all possible COVs in a CST

Serial number	COV
C^T_0	President Kennedy flow from London Airport last night to owive in Washington this morning
C^T_1	President Kennedy flow from London Airport last night to owive in Washington this evening
C^T_2	President Kennedy flow from London Airport last night to arrive in Washington this morning
C^T_3	President Kennedy flow from London Airport last night to arrive in Washington this evening
C^T_4	President Kennedy flow from London Airport last might to owive in Washington this morning
C^T_5	President Kennedy flow from London Airport last might to owive in Washington this evening
C^T_6	President Kennedy flow from London Airport last might to arrive in Washington this morning
C^T_7	President Kennedy flow from London Airport last might to arrive in Washington this evening
C^T_8	President Kennedy flew from London Airport last night to owive in Washington this morning
C^T_9	President Kennedy flew from London Airport last night to owive in Washington this evening
C^T_{10}	President Kennedy flew from London Airport last night to arrive in Washington this morning
C^T_{11}	President Kennedy flew from London Airport last night to arrive in Washington this evening
C^T_{12}	President Kennedy flew from London Airport last might to owive in Washington this morning
C^T_{13}	President Kennedy flew from London Airport last might to owive in Washington this evening
C^T_{14}	President Kennedy flew from London Airport last might to arrive in Washington this morning
C^T_{15}	President Kennedy flew from London Airport last might to arrive in Washington this evening

3.2 Filter Candidate Results

The number of COVs of CST T is constant in the best case and is n^m in the worst case (but it is almost impossible to be the worst case). When a task is harder and workers' answers are more confusing, the number of COV of CST T will increase geometrically, which means that we must filter the COV set. Suppose COV $C^T = \left[c^T_{i_0 0}, c^T_{i_1 1}, \ldots, c^T_{i_{m-1} m-1}\right]$. If $c^T_{i_j j} \neq \perp$, we use $p^T_{i_j j} = \frac{\sum_{o^T_{ij} \in O^T_{*j}} \delta(c^T_{i_j j}, o^T_{ij})}{n}$ to measure the quality of subtask candidate output $c^T_{i_j j}$, which δ means Kronecker Delta Function. If $c^T_{i_j j} = \perp$, $p^T_{i_j j} = 0$. Then the Candidate Probability Vector (CPV) $P^T = \left[p^T_{i_0 0}, p^T_{i_1 1}, \ldots, p^T_{i_{m-1} m-1}\right]$ of the COV C^T is obtained. According to the principle of voting consistency, the greater the value of $p^T_{i_j j}$, the more likely the candidate output $c^T_{i_j j}$ is the correct answer to the subtask t_j. To rank the quality of COVs, based on the partially ordered set, we define the partially ordered relations of COVs which sort the COV partially ordered order by comparing the corresponding CPVs of two COVs. Suppose there are two COVs: C^T_a and C^T_b. The partially ordered relations are shown in Formulas (1) and (2):

$$C^T_a \leq C^T_b \Leftrightarrow \forall t_j \in T, c^T_{aj} \in C^T_a, c^T_{bj} \in C^T_b, p^T_{aj} \leq p^T_{bj} \tag{1}$$

$$C^T_a \geq C^T_b \Leftrightarrow \forall t_j \in T, c^T_{aj} \in C^T_a, c^T_{bj} \in C^T_b, p^T_{aj} \geq p^T_{bj} \tag{2}$$

COV^T represents the set of all COVs which conforms to the three major principles of partially ordered relations (reflexivity, antisymmetric, transitivity). We can infer that $(COV^T, \leq)$ is a partially ordered set. On this basis, a COV Hasse can be constructed where Ø is the minimum element, and the two connected COVs have a partially ordered relation. For example, for all COVs in Table 1, the COV Hasse is shown in Fig. 4:

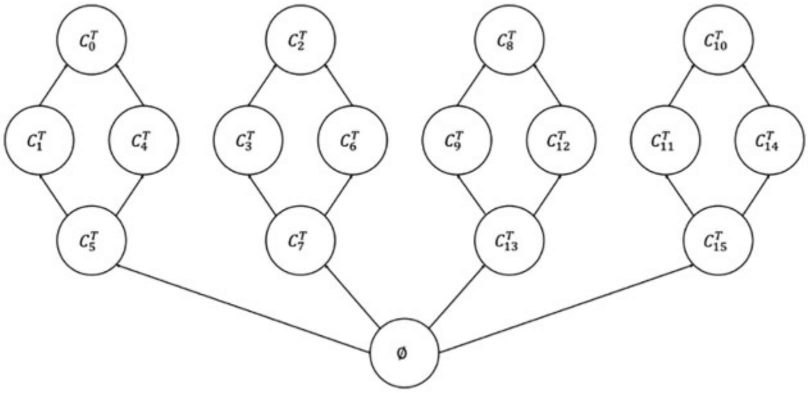


Figure 4: The COV Hasse

The COV partially ordered relationship is sorted from bottom to top and from small to large in Fig. 4. We screen out the four largest COVs, C^T_0, C^T_2, C^T_8, C^T_{10}, that is, the largest COVs set. We set it to $TCOVs^T$. When $(COV^T, \leq)$ has multiple maximum COVs, the best COV needs to be selected from them. To solve this problem, we need to consider workers' ability and the context of CST T.

3.3 Model Context Correlation

The TAN Bayes classifier is a semi-naive Bayes classifier of "One-Dependent Estimator" (ODE). TAN is suitable for extracting strong correlation information in the CST context. We regard each subtask as an attribute of the CST, and design a TAN-based context-correlation modeling algorithm for CSTs. The algorithm flow is as follows:

1. For CST T, the formula for calculating the conditional mutual information (CMI) between any two subtasks is as follows:

$$\mathrm{CMI}\left(t_{j\prime}|t_j\right) = \exp\left(j - j\prime\right) \sum_{c_{ij}^T \in TC_{*j}^T, c_{i\prime j\prime}^T \in TC_{*j\prime}^T} w\left(c_{ij}^T, c_{i\prime j\prime}^T\right) \log \frac{w\left(c_{ij}^T, c_{i\prime j\prime}^T\right)}{w\left(c_{ij}^T\right) w\left(c_{i\prime j\prime}^T\right)}, j < j\prime \tag{3}$$

where TC_{*j}^T indicates that subtask t_j appears in the candidate output set of $TCOVs^T$.

2. $w\left(c_{ij}^T\right) = \sum_{c_{k_1 j\prime}^T \in TC_{*j\prime}^T} w\left(c_{ij}^T, c_{k_1 j\prime}^T\right) \in [0, 1]$. $w\left(c_{ij}^T, c_{i\prime j\prime}^T\right)$ represents the degree of contextual relevance between the candidate result c_{ij}^T of subtask t_j and the candidate result $c_{i\prime j\prime}^T$ of subtask $t_{j\prime}$, which can be obtained from the external knowledge. For example, when quantifying the contextual relevance among handwriting recognition subtasks,$w\left(c_{ij}^T, c_{i\prime j\prime}^T\right)$can be expressed as the frequency of two handwritten word candidate results appearing simultaneously in the text corpus (such as Google n-gram corpus and American Contemporary English Corpus).
3. A complete graph is constructed with subtasks as nodes, and the weight of edges between any two nodes is set to $\mathrm{CMI}\left(t_{j\prime}|t_j\right)$.
4. For each edge $\langle t_j, t_{j\prime}\rangle$, t_j is the starting point, $t_{j\prime}$ is the endpoint. We set edges as directed and set t_0 as the root node, and the Prim algorithm is used to build the maximum weighted spanning tree as shown in Fig. 5, that is, the CST dependency tree.

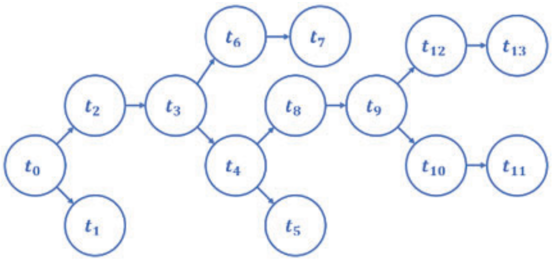


Figure 5: The CST dependency tree

It is easy to find that conditional mutual information $\mathrm{CMI}\left(t_{j\prime}|t_j\right)$ characterizes the attribute t_j and $t_{j\prime}$. The closer the distance between t_j and $t_{j\prime}$ in CST T, the higher the relevance of candidate answers, and the greater the CMI. Therefore, the maximum weighted spanning tree constructed by TAN through Prim algorithm retains only the dependencies among strongly related attributes.

3.4 Infer Result

3.4.1 A Crowdsourcing Probabilistic Model for Subtasks

Inspired by Sun et al. [9], we propose a crowdsourcing probabilistic model for subtasks in CSTs. As shown in Fig. 6, for CST T, the output o_{ij}^T of worker i submitting subtask t_j mainly depends on four elements:

1. z_j^T: the ground truth of t_j.
2. $z_{f(j)}^T$: the ground truth of $t_{f(j)}$ which is the father subtask node of t_j in a CST dependency tree. Workers can infer the truth of t_j based on the dependent subtasks of t_j.

3. $\frac{1}{a_i^T} \in [1, +\infty]$: the accuracy of worker i. $a_i^T \in [0, 1]$ is the reciprocal of the accuracy. To facilitate our use of a_i^T to describe the probabilistic model, the smaller the a_i^T, the higher the credibility of the result submitted by worker i.
4. $d_j^T \in [0, 1]$: the difficulty of subject t_j. The greater the d_j^T, the lower the credibility of the workers' answers.

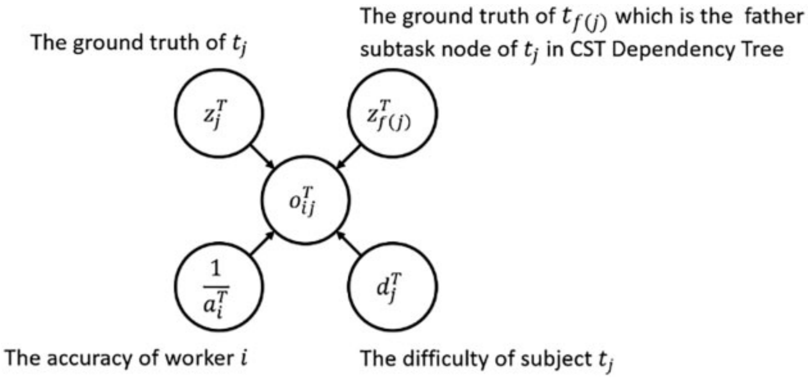

The ground truth of t_j
The ground truth of $t_{f(j)}$ which is the father subtask node of t_j in CST Dependency Tree
z_j^T
$z_{f(j)}^T$
o_{ij}^T
$\frac{1}{a_i^T}$
d_j^T
The accuracy of worker i
The difficulty of subject t_j

Figure 6: The crowdsourcing probabilistic model for subtask t_j

Assuming that $c^T_{i_{f(j)}f(j)} \in C^T_{*f(j)}$ is the correct answer to the t_j father subtask $t_{f(j)}$, then the probability of $c^T_{i_j j} \in C^T_{*j}$ is the correct answer of t_j and the output submitted by worker i $o^T_{ij} = c^T_{i_j j}$ is the correct answer is:

$$P\left(o^T_{ij} = c^T_{i_j j}, z^T_j = c^T_{i_j j} | z^T_{f(j)} = c^T_{i_{f(j)}f(j)}, a^T_i, d^T_j\right) = r\left(c^T_{i_j j} | c^T_{i_{f(j)}f(j)}\right)\left(s^T_{ij}\right)^{a^T_i d^T_j} \tag{4}$$

where

$$r\left(c^T_{i_j j} | c^T_{i_{f(j)}f(j)}\right) = \frac{w\left(c^T_{i_j j}, c^T_{i_{f(j)}f(j)}\right)}{\sum_{c^T_{kj} \in TC^T_{*j}} w\left(c^T_{kj}, c^T_{i_{f(j)}f(j)}\right)} \tag{5}$$

$$s^T_{ij} = \frac{\text{sim}\left(o^T_{ij}, o^T_{ij}\right)}{\sum_{c^T_{kj} \in TC^T_{*j}} \text{sim}\left(c^T_{kj}, o^T_{ij}\right)} \tag{6}$$

Formula (5) is the normalized correlation degree of $c^T_{i_j j}$ and $c^T_{i_{f(j)}f(j)}$. Especially the root subtask node t_0 has no father subtask nodes, namely $c^T_{i_{f(0)}f(0)} = \emptyset$, so we set $r\left(c^T_{i_0 0} | \emptyset\right) = E\left(c^T_{i_0 0}\right)$ as the expectation of $c^T_{i_0 0}$. The larger $r\left(c^T_{i_j j} | c^T_{i_{f(j)}f(j)}\right)$ is, the greater the probability of $z^T_j = c^T_{i_j j}$ is when $z^T_{f(j)} = c^T_{i_{f(j)}f(j)}$. $\text{sim}\left(c^T_{kj}, o^T_{ij}\right) \in [0, 1]$ is the similarity between c^T_{kj} and o^T_{ij}, especially $\text{sim}\left(o^T_{ij}, o^T_{ij}\right) = 1$. The larger $\text{sim}\left(c^T_{kj}, o^T_{ij}\right)$ is, the more similar the appearance of c^T_{kj} and o^T_{ij} is. For example, in handwriting recognition tasks, we can use the Levenshtein method to calculate the string similarity of handwritten words. So Formula (6) is the normalized similarity of o^T_{ij} itself. The larger s^T_{ij} is, the higher the probability that the output of the subtask t_j submitted by worker i $o^T_{ij} = c^T_{i_j j}$ is the correct answer.

Assuming that $c^T_{i'_j j} \in C^T_{*j}$ is a wrong answer to subtask t_j, then the probability that $c^T_{i_j j} \in C^T_{*j}$ is the correct answer to subtask t_j and the output submitted by worker i $o^T_{ij} = c^T_{i'_j j} \neq c^T_{i_j j}$ is the wrong answer is:

$$P\left(o^T_{ij} = c^T_{i'_j j}, z^T_j = c^T_{i_j j} | z^T_{f(j)} = c^T_{i_{f(j)} f(j)}, a^T_i, d^T_j\right)$$
$$= r\left(c^T_{i_j j} | c^T_{i_{f(j)} f(j)}\right)\left(1 - \left(s^T_{ij}\right)^{a^T_i d^T_j}\right)\beta\left(c^T_{i'_j j}, c^T_{i_j j}\right) \tag{7}$$

where

$$\beta\left(c^T_{i'_j j}, c^T_{i_j j}\right) = \frac{\mathrm{sim}\left(c^T_{i'_j j}, c^T_{i_j j}\right)}{\left[\sum_{c^T_{kj} \in TC^T_{*j}} \mathrm{sim}\left(c^T_{kj}, c^T_{i_j j}\right)\right] - \mathrm{sim}\left(c^T_{i_j j}, c^T_{i_j j}\right)} \tag{8}$$

Formula (8) represents the normalized error rate of the wrong answer $c^T_{i'_j j}$ compared to the correct answer $c^T_{i_j j}$. The larger β $\left(c^T_{i'_j j}, c^T_{i_j j}\right)$ is, the higher the probability that the output of the subtask t_j submitted by worker i $o^T_{ij} = c^T_{i'_j j} \neq c^T_{i_j j}$ is the wrong answer.

Define $\alpha\left(o^T_{ij}\right) = \left(s^T_{ij}\right)^{a^T_i d^T_j}$. Combining formula (3) and formula (7) can be simplified to get:

$$P\left(o^T_{ij}, z^T_j = c^T_{i_j j} | z^T_{f(j)} = c^T_{i_{f(j)} f(j)}, a^T_i, d^T_j\right)$$
$$= r\left(c^T_{i_j j} | c^T_{i_{f(j)} f(j)}\right)\alpha\left(o^T_{ij}\right)^{\delta\left(o^T_{ij}, c^T_{i_j j}\right)}\left[\left(1 - \alpha\left(o^T_{ij}\right)\right)\beta\left(o^T_{ij}, c^T_{i_j j}\right)\right]^{1-\delta\left(o^T_{ij}, c^T_{i_j j}\right)} \tag{9}$$

Formula (9) indicates the conditional probability when $c^T_{i_{f(j)} f(j)}$is the correct answer to the father subtask task of $t_j t_{f(j)}$, $c^T_{i_j j}$ is the correct answer to t_j, and the output o^T_{ij} submitted by worker i is observed, where δ represents the Kronecker function.

3.4.2 TAN-based Inference

We treat each COV in the largest COV set $TCOVs^T$as a label of the CST and treat each subtask as an attribute of the CST. The result inference problem of CSTs is transformed into a label classification problem that can be solved by a TAN classifier. $C^T = \left[c^T_{i_0 0}, c^T_{i_1 1}, \ldots, c^T_{i_{m-1} m-1}\right]$ is a COV of CST T. Assuming that the ground truth of T is $Z^T = C^T$, the reciprocal parameter set of the workers' accuracy for CST T $A^T = \left\{a^T_i | i \in \{0, 1, \ldots n - 1\}\right\}$and the parameter set of the difficulty of subtasks $D^T = \left\{d^T_j | j \in \{0, 1, \ldots m - 1\}\right\}$. According to the probabilistic model in Fig. 6 and TAN, we draw the following conclusions:

$$P\left(O^T, Z^T = C^T | A^T, D^T\right) = \prod_i P\left(O^T_{i*}, Z^T = C^T | a^T_i, D^T\right)$$
$$= \prod_{ij} P\left(o^T_{ij}, z^T_j = c^T_{i_j j} | z^T_{f(j)} = c^T_{i_{f(j)} f(j)}, a^T_i, d^T_j\right) \tag{10}$$

The posterior probability that the category of CST T is COV C^T is:

$$P\left(Z^T = C^T | O^T, A, D\right) = \frac{P\left(O^T, Z^T = C^T | A^T, D^T\right)}{P\left(O^T | A^T, D^T\right)} \propto P\left(O^T, Z^T = C^T | A^T, D^T\right) \tag{11}$$

The result inference problem of CSTs has been transformed into the problem of identifying the optimal COV from the set $TCOVs^T$:

$$C^T_{best} = \arg \max_{C^T \in TCOVs^T} P\left(Z^T = C^T | O^T, A^T, D^T\right) \tag{12}$$

3.4.3 Parameter Learning

The TAN inference process includes two parameter sets, namely the reciprocal parameter set of the accuracy of workers A^T and the parameter set of the difficulty of subtasks of CST D^T. During the crowdsourcing process, the crowdsourcing platform usually requires adding some golden standard tasks (GSTs) (i.e., tasks with known ground truth) to each group of crowdsourcing tasks to identify participating workers, then weed out fraudsters. The accuracy of workers is relatively stable and will not vary greatly due to different tasks. In summary, we use MLE for GSTs to learn the reciprocal parameter set of the accuracy of workers A^T. For general tasks, we use EM to learn the parameter set of the difficulty of subtasks of CST D^T.

1. Use the MLE algorithm to learn A^T. G is the set of all GSTs, A^G is the reciprocal parameter set of the accuracy of workers of all GSTs, and D^G is the parameter set of the difficulty of subtasks of all GSTs. GST $T_G \in G$, O^{T_G} is the output matrix of workers of T_G. $A^{T_G} = \left\{a_i^{T_G}\right\}$ is the reciprocal parameter set of the worker accuracy of T_G. $D^{T_G} = \left\{d_j^{T_G}\right\}$ is the parameter set of the difficulty of subtasks of T_G. $Z^{T_G} = C^{T_G}$ is the ground truth of T_G. The likelihood function of T_G is:

$$L_{T_G}\left(A^{T_G}, D^{T_G}\right) = P\left(O^{T_G}, Z^{T_G} = C^{T_G} | A^{T_G}, D^{T_G}\right) \tag{13}$$

According to Formula (13), the joint log-likelihood function of all GSTs is:

$$\begin{aligned} LL\left(A^G, D^G\right) &= \log \prod_{T_G \in G} L_{T_G}\left(A^{T_G}, D^{T_G}\right) \\ &= \sum_{T_G \in G} \log P\left(O^{T_G}, Z^{T_G} = C^{T_G} | A^{T_G}, D^{T_G}\right) \end{aligned} \tag{14}$$

To maximize $LL\left(A^G, D^G\right)$, we can use the gradient ascent method to differentiate $LL\left(A^G, D^G\right)$ to obtain the gradient:

$$\frac{\partial LL\left(A^G, D^G\right)}{\partial a_i^{T_G}} = \sum_{T_G \in G} \sum_j \frac{d_j^{T_G}\left[\delta\left(o_{ij}^{T_G}, c_{i_j j}^{T_G}\right) - \alpha\left(o_{ij}^{T_G}\right)\right] \log s_{ij}^{T_G}}{1 - \alpha\left(o_{ij}^{T_G}\right)} \tag{15}$$

$$\frac{\partial LL\left(A^G, D^G\right)}{\partial d_j^{T_G}} = \sum_i \frac{a_i^{T_G}\left[\delta\left(o_{ij}^{T_G}, c_{i_j j}^{T_G}\right) - \alpha\left(o_{ij}^{T_G}\right)\right] \log s_{ij}^{T_G}}{1 - \alpha\left(o_{ij}^{T_G}\right)} \tag{16}$$

Different from GSTs, the EM algorithm is difficult to learn the real accuracy of workers from general tasks, and the accuracy of workers is relatively stable. A^T is the reciprocal parameter set of the accuracy of workers of all GSTs. We can set $A^T = A^{T_G}$.

2. Use the EM algorithm to learn D^T. D^T is the parameter set of the difficulty of subtasks of CST T. We obtain the parameter set D^T through the EM algorithm. O^T is the observed variable set. $COVs^T$ is the unobserved hidden variable set. The optimal latent variable distribution $P\left(Z^T|O^T, A^T, D^T\right)$ and D^T can be inferred by iteratively executing E-step and M-step, starting with the initial value D_0^T. E-step (Exception). We infer the latent variable Z^T distribution $P_{C^T}^t = P\left(Z^T = C^T|O^T, A^T, D_t^T\right)$ based on the current parameter D_t^T and formula (11), and calculate the log-likelihood $LL\left(D^T|C^T\right)$ expectation of C^T:

$$\begin{aligned}
Q\left(D^T|D_t^T\right) &= \mathbb{E}_{C^T|O^T,D_t^T} LL\left(D^T|C^T\right) = \mathbb{E}_{C^T|O^T,D_t^T} \log P\left(O^T, Z^T = C^T|A^T, D^T\right)\\
&\propto \sum_{C^T \in TCOVs^T} P_{C^T}^t \cdot \log P\left(O^T, Z^T = C^T|A^T, D^T\right)\\
&\propto \sum_{C^T \in TCOVs^T} P_{C^T}^t \sum_j \log r\left(c_{i_j j}^T|c_{i_{f(j)} f(j)}^T\right)\\
&+ \sum_{C^T \in TCOVs^T} P_{C^T}^t \sum_{ij} \delta\left(o_{ij}^T, c_{i_j j}^T\right) \log \alpha\left(o_{ij}^T\right)\\
&+ \sum_{C^T \in TCOVs^T} P_{C^T}^t \sum_{ij} \left[1 - \delta\left(o_{ij}^T, c_{i_j j}^T\right)\right] \log\left[\left(1 - \alpha\left(o_{ij}^T\right)\right) \beta\left(o_{ij}^T, c_{i_j j}^T\right)\right]
\end{aligned} \tag{17}$$

Formula (17) can be derived from Formulas (9)–(11).

M-step (Maximization). We find the parameter to maximize the expected likelihood, namely:

$$D_{t+1}^T = \arg\max_{D^T} Q\left(D^T|D_t^T\right) \tag{18}$$

Like the MLE algorithm, we can also use the gradient ascent method to differentiate $Q\left(D^T|D_t^T\right)$ to obtain the gradient:

$$\frac{\partial Q}{\partial d_j^T} = \sum_{C^T \in TCOVs^T} P_{C^T}^t \sum_i \frac{a_i^T \left[\delta\left(o_{ij}^T, c_{i_j j}^T\right) - \alpha\left(o_{ij}^T\right)\right] \log s_{ij}^T}{1 - \alpha\left(o_{ij}^T\right)} \tag{19}$$

We do a two-step alternation calculation until it converges to a locally optimal solution by the EM algorithm.

Through the obtained parameter sets A^T and D^T, we can calculate the posterior probability that each COV in the set $TCOVs^T$ is the ground truth of CST T. The COV with the highest posterior probability is the best result.

4 Experiments

4.1 Datasets

The experiments contain two datasets, which are described as follows.

Handwriting recognition. The IAM Handwriting English Database [18] was established by the Computer Vision and Artificial Intelligence Research Group of the University of Bern, Switzerland. The handwritten text dataset contains about 1539 pages, 5685 isolated sentences, and 110,000 words samples by 657 authors. The dataset is publicly available for free. This paper randomly selects 150 sentence samples from the IAM handwritten English picture database. We make sure that each sentence contains 15–20 words and then re-cut and splice the new-line sentences in the original image so that each sentence is displayed on one line. We publish these images to be identified as independent crowdsourcing tasks on the crowdsourcing platform Appen. The cost of each task is one cent. There are ten tasks in a group, and each group has at least one GST published to the crowdsourcing workers. Each task is answered by at least 5 workers with level 3 quality, and finally, a crowdsourced dataset of handwriting recognition tasks is obtained.

Audio transcription. LibriSpeech English Speech Corpus [19] is a collection of audiobooks of the LibriVox project [20], suitable for training and evaluating speech recognition systems. LibriSpeech contains 1000 hours of speech sampled at 16 kHz. We cut and sort into audio files of about 10 s each, which have been marked with text. Similar to the handwriting recognition dataset, we randomly select 150 speech samples from the LibriSpeech English Speech Corpus. Each speech contains 15–25 words. Then, we post the transcriptions of these speeches as independent crowdsourcing tasks on Appen. The cost of each task is two cents. Five tasks are a group. Each group has at least one GST released to the crowdsourcing workers. Each task is answered by at least 5 workers with level 3 quality, and finally, the crowdsourced dataset of audio transcription tasks is obtained.

4.2 Evaluation Metrics

To better evaluate the P&T-Inf algorithm, we select Bayes-Inf and Context-Inf which are the latest result inference algorithms for CSTs [9] and MWK which is the latest result inference algorithm with external knowledge [21] for comparisons. The differences between them are that Bayes-Inf does not introduce context information in the inference process, Context-Inf introduces context information between adjacent subtasks in the inference process, and MWK introduces context information among all subtasks, but the task difficulty is not considered.

Experiments mainly verify the performance of algorithms from the accuracy and the time cost. The evaluation indexes are as follows:

(1) CST Accuracy rate (A):

$$A\left(T\right)=\frac{\left|Z^{T}\cap C_{best}^{T}\right|}{\left|Z^{T}\right|} \tag{20}$$

In Formula (20), $A\in[0,1]$, C_{best}^{T} represents the inference result of the result inference algorithm on CST T, Z^{T} represents the ground truth of CST T, C_{best}^{T} and Z^{T} are both considered a similar one-dimensional vector of the COV. For CST T, the closer the A evaluation index is to 1, the higher the accuracy of the inference algorithm. We apply the A evaluation index to each CST. This method can effectively evaluate the accuracy of each algorithm for a single CST.

(2) CST Improvement rate (I):

$$I\left(Inf_{1},Inf_{2}\right)=\frac{\text{number of CST Twhich } A_{1}\left(T\right)>A_{2}\left(T\right)}{\text{number of CST}} \tag{21}$$

In Formula (21), Inf_k represents the algorithm k, used to distinguish different algorithms. A_k represents the A index calculated by algorithm k. The I index represents the proportion of CSTs with improved quality of inference results of each algorithm.

(3) Accuracy Effective Improvement of CST (AEI):

$$AEI\left(Inf_{1},Inf_{2}\right)=\frac{\sum_{T\in CST}A_{1}\left(T\right)-A_{2}\left(T\right)}{\sum_{T\in CST}\delta\left(A_{1}\left(T\right),A_{2}\left(T\right)\right)} \tag{22}$$

In Formula (22), δ represents the Kronecker function. The AEI index excludes tasks which the accuracy rate does not change. Because for some CSTs, any inference algorithm can infer the optimal result. The combination of I index and AEI index can evaluate the comprehensive improvement of the accuracy of CSTs of different algorithms more accurately.

(4) CST Average Time (AT):

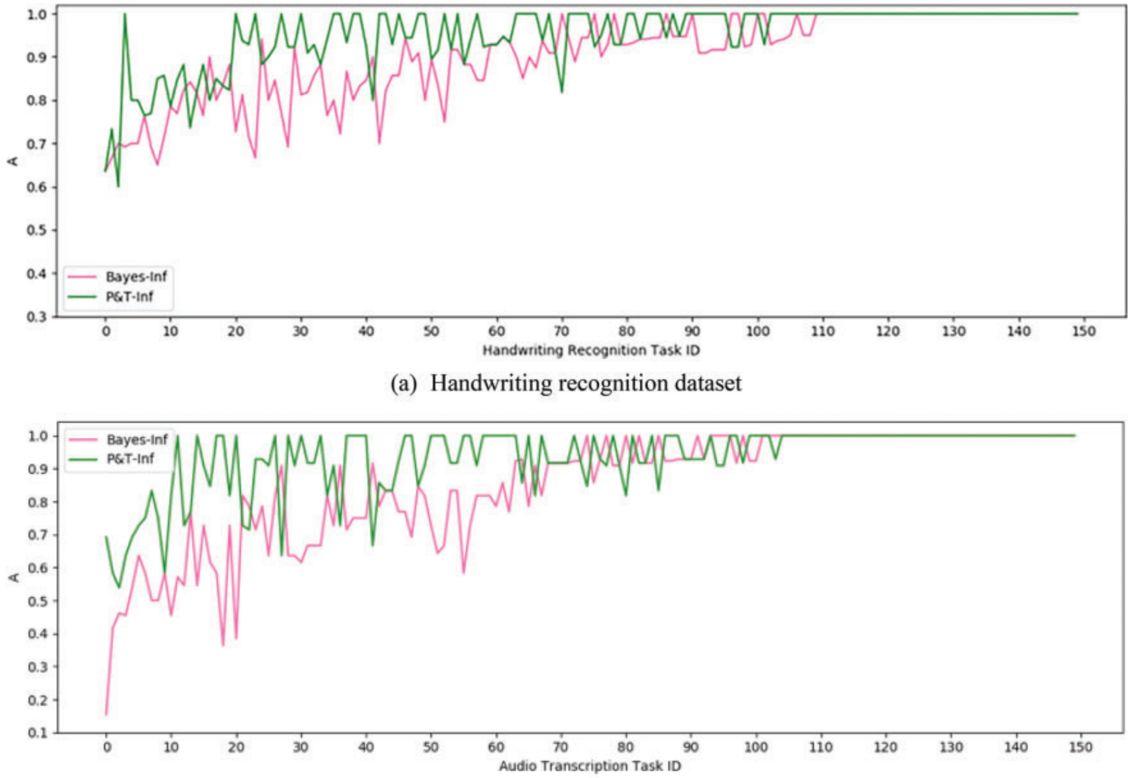
$$AT = \frac{\sum_{T \in CST} \text{Time (T)}}{\text{number of CST}} \tag{23}$$

In Formula (23), Time represents the function to calculate the running time for CSTs. The time cost of the same algorithm for different CSTs is roughly the same. AT can evaluate the average time cost of each algorithm for CSTs.

4.3 Experimental Results Analysis

4.3.1 Experiment 1: Comparison of the Accuracy of Algorithms that Introduce Context Information

Experiment 1 verified the effect of introducing context information into result inference algorithms for CSTs. The experiment selected two algorithms P&T-Inf and Bayes-Inf as a comparison. Both Bayes-Inf and P&T-Inf algorithms are EM iterative calculation algorithms, but the latter introduces contextual information in the inference process, while the former does not. Fig. 7 shows the A index curves of P&T-Inf and Bayes-Inf on both datasets respectively. The data curves have been sorted according to the average values of the A index.



(a) Handwriting recognition dataset

(b) Audio transcription dataset

Figure 7: A index curves of P&T-Inf and Bayes-Inf

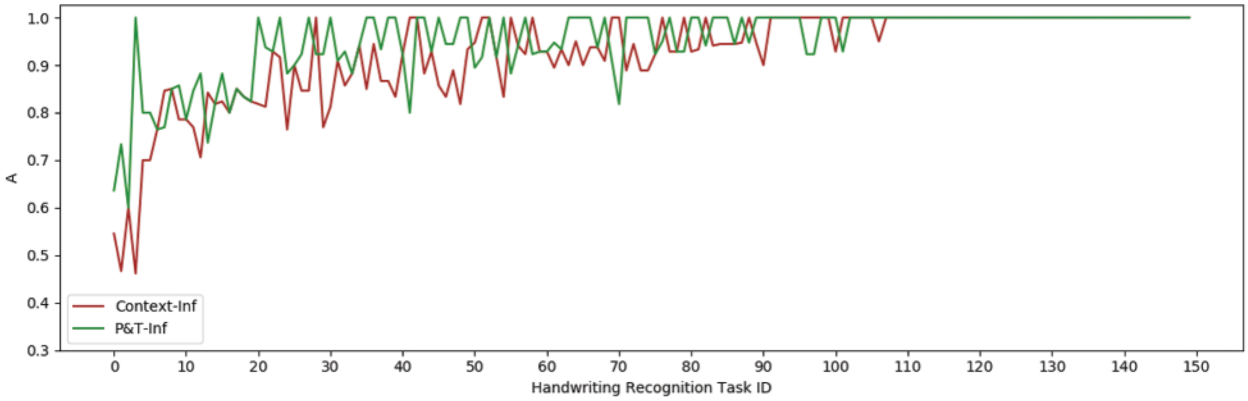
From the data curves, it can be concluded that the accuracy of the P&T-Inf algorithm in most CSTs is higher than that of Bayes-Inf. By calculating the I index, the improvement rate of the P&T-Inf algorithm relative to the Bayes-Inf on the two datasets is 48.7% and 44.0%. Compared with the Bayes-Inf algorithm, P&T-Inf introduces external knowledge to calculate the contextual relevance among the output of subtasks, and considers the worker accuracy and the task difficulty. By calculating the AEI index, it can be obtained that the accuracy effective improvement of the P&T-Inf algorithm on both datasets are 8.5% and 15.0% respectively compared with Bayes inf, as shown in Table 2:

Table 2: Comparison of P&T-Inf and Bayes-Inf accuracy index analysis

Dataset	Index I	Index AEI
Handwriting recognition	48.7%	8.5%
Audio transcription	44.0%	15.0%

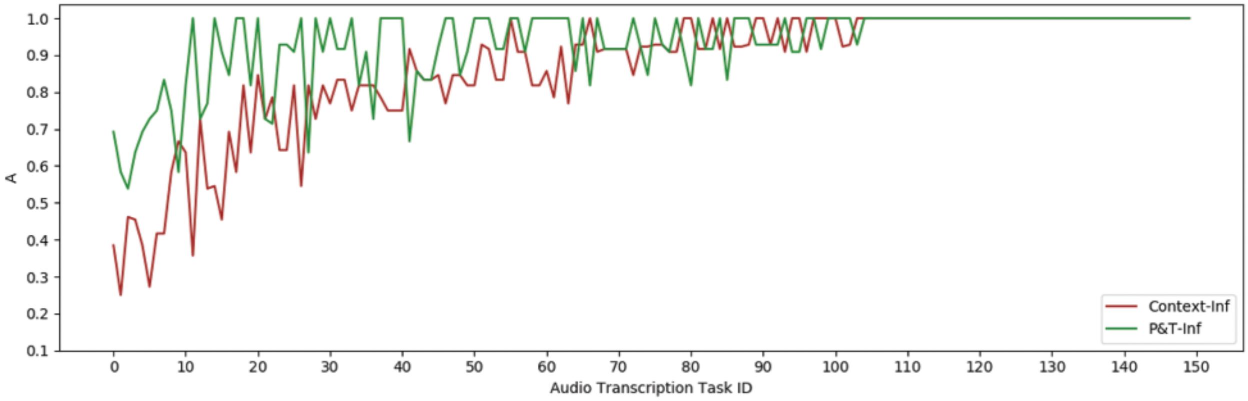
4.3.2 Experiment 2: Comparison of the Accuracy of Algorithms that Introduce Strong Correlation Information in the Context

Experiment 2 verified the effect of the CST result inference algorithm in crowdsourcing to introduce strong correlation information in the context. The experiment selected two algorithms Context-Inf and P&T-Inf as a comparison. Both Context-Inf and P&T-Inf algorithms are EM iterative calculation algorithms, but the former only uses the context information between adjacent subtasks in the CST result inference process, while the latter retains the more strongly related subtask dependency information through the constructed CST dependency tree. Fig. 8 shows the A index curves of P&T-Inf and Context-Inf on both datasets respectively. The data curves have been sorted according to the average values of the A index.



(a) Handwriting recognition dataset

Figure 8: (Continued)



(b) Audio transcription dataset

Figure 8: A index curves of P&T-Inf and Context-Inf

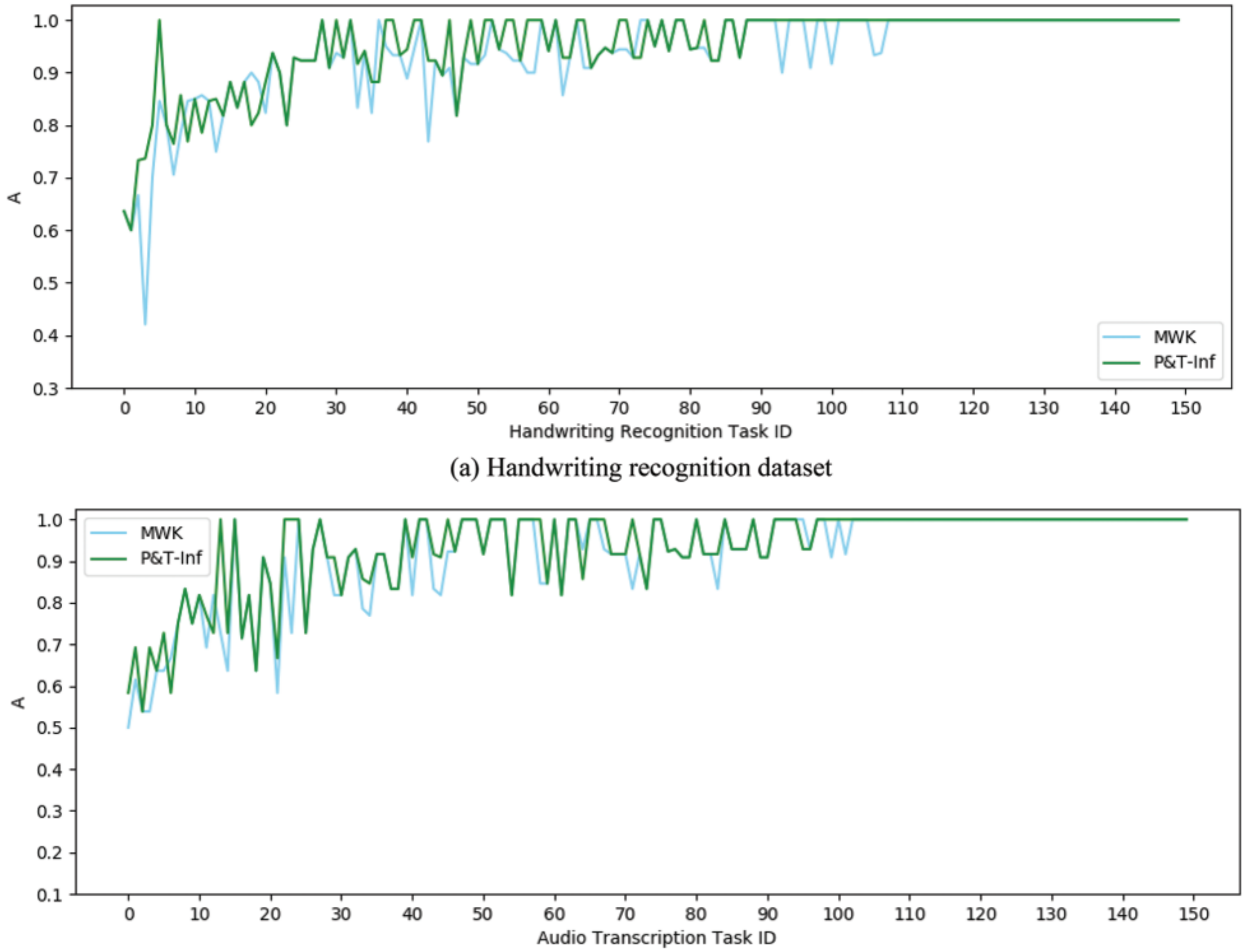
It can be concluded from the data curves that the accuracy of the P&T-Inf algorithm is higher than that of Context-Inf in most of CSTs, and the I index can be calculated to get the improvement rate of P&T-Inf relative to Context-Inf on both datasets. They were 34.7% and 42.7% respectively. Although the context information is introduced in the inference process of the Context-Inf algorithm with the worker accuracy and task difficulty, P&T-Inf considers the context information among adjacent and non-adjacent subtasks additionally. So the quality of the inferred results is improved. At the same time, we can find that the minimum and average accuracy of the Bayes-Inf and Context-Inf algorithms in the audio transcription dataset are lower than those in the handwriting recognition dataset. The results indicate that the more difficult the handwriting recognition task is, the greater the gap between the two algorithms and the P&T-Inf algorithm is. By calculating the AEI index, we can infer that the effective improvement of P&T-Inf relative to Context-Inf is 6.0% and 12.2% on both datasets, as shown in Table 3:

Table 3: Comparison of P&T-Inf and Context-Inf accuracy index analysis

Dataset	Index I	Index AEI
Handwriting recognition	34.7%	6.0%
Audio transcription	42.7%	12.2%

4.3.3 Experiment 3: Comparison of the Accuracy of Algorithms that Introduce Task Difficulty

Experiment 3 verified the effect of the CST result inference algorithm in crowdsourcing to introduce task difficulty. The experiment selected two algorithms MWK and P&T-Inf as a comparison. Both MWK and P&T-Inf algorithms introduce contextual information among subtasks and worker accuracy, but the latter introduces task difficulty during inference, while the former does not. Fig. 9 shows the A index curves of P&T-Inf and MWK on both datasets respectively. The data curves have been sorted according to the average values of the A index.



(a) Handwriting recognition dataset

(b) Audio transcription dataset

Figure 9: A index curves of P&T-Inf and MWK

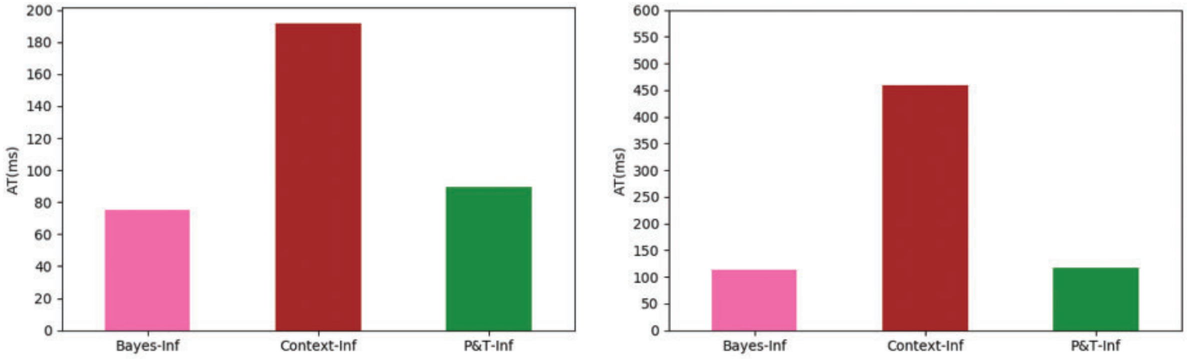
From the data curves, it can be concluded that the accuracy of the P&T-Inf algorithm in most CSTs is higher than that of MWK. By calculating the I index, the improvement rate of the P&T-Inf algorithm relative to the MWK on the two datasets is 22.0% and 15.3%. Compared with the MWK algorithm, P&T-Inf not only introduces external knowledge to calculate the contextual relevance among outputs of subtasks and the worker accuracy, but also considers the task difficulty. By calculating the AEI index, it can be obtained that the accuracy effective improvement of the P&T-Inf algorithm on both datasets are 6.0% and 8.2% respectively compared with MWK, as shown in Table 4:

Table 4: Comparison of P&T-Inf and MWK accuracy index analysis

Dataset	Index I	Index AEI
Handwriting recognition	22.0%	6.0%
Audio transcription	15.3%	8.2%

4.3.4 Experiment 4: Comprehensive Analysis of the Time Complexity of Algorithms

Experiment 4 verified the time cost of P&T-Inf compared to Bayes-Inf and Context-Inf. Since MWK introduces context information among all subtasks with an extremely high time complexity, it is not compared in this experiment. Fig. 10 and Table 5 show the AT index histograms and values of the three algorithms on both datasets respectively.



(a) Handwriting recognition dataset （b）Audio transcription dataset

Figure 10: AT index histograms

Table 5: AT index analysis

Algorithm	Handwriting recognition dataset AT (ms)	Audio transcription dataset AT(ms)
Bayes-Inf	75.23	113.25
Context-Inf	191.84	459.44
P&T-Inf	89.69	117.46

It can be concluded from the figure that the time cost of the Context-Inf algorithm is much higher than that of the P&T-Inf. This is because P&T-Inf reduces the candidate result of CST T $\left|TCOV^{T}\right|$ to a constant level. The time complexity of each iteration is lower. The time of inferring parameters by the EM algorithm is greatly reduced. There is little difference in time cost between P&T-Inf and Bayes-Inf. Although the time of each iteration of the P&T-Inf is lower, it is also costly to build the CST dependency tree to model the dependencies among subtasks of CSTs.

From the Table 5, it can be calculated that the average time of P&T-Inf is 46.8% and 25.6% of Context-Inf in both datasets, which is shorter, and the time complexity is lower.

4.3.5 Case Study

In the P&T-Inf algorithm flow, in order to preserve context-sensitive information, we publish the CST as a whole to the crowdsourcing platform, and workers can choose to recognize some words in the sentence. As an example of CSTs, a handwriting recognition task is shown in Fig. 11. If a CST is split into multiple subtasks for independent crowdsourcing, the context information in the sentence will be lost, which will cause some difficulty for identification, as shown in Fig. 12. When recognizing the words “an”, “Englishman”, “named”, and “Lawrence” in Fig. 11 (i.e., context-dependent subtasks), it is easy to infer the results based on the context. But when recognizing the independent crowdsourcing subtasks shown in Fig. 12, the contextual information is lost, making it difficult to recognize these few words.

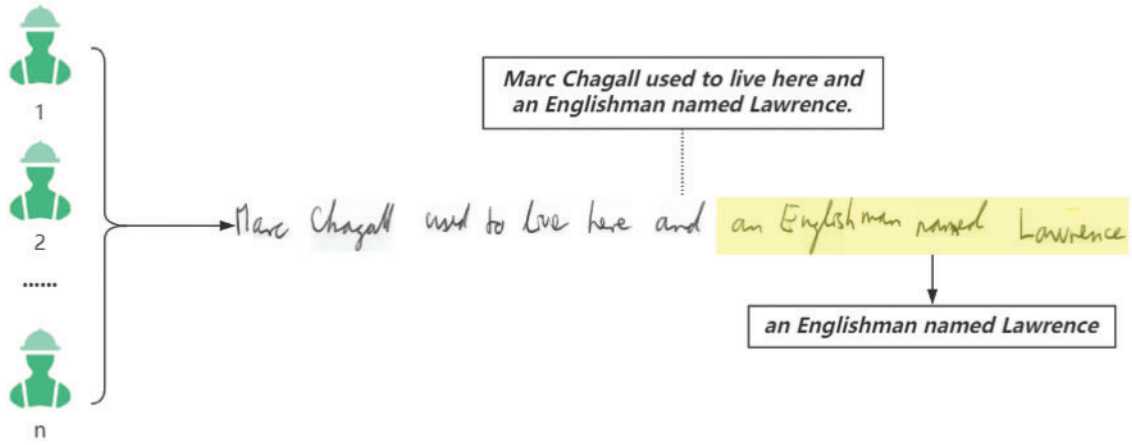

1
2
......
n
Marc Chagall used to live here and an Englishman named Lawrence.
Marc Chagall used to live here and an Englishman named Lawrence
an Englishman named Lawrence

Figure 11: The process for workers to identify subtasks in a CST

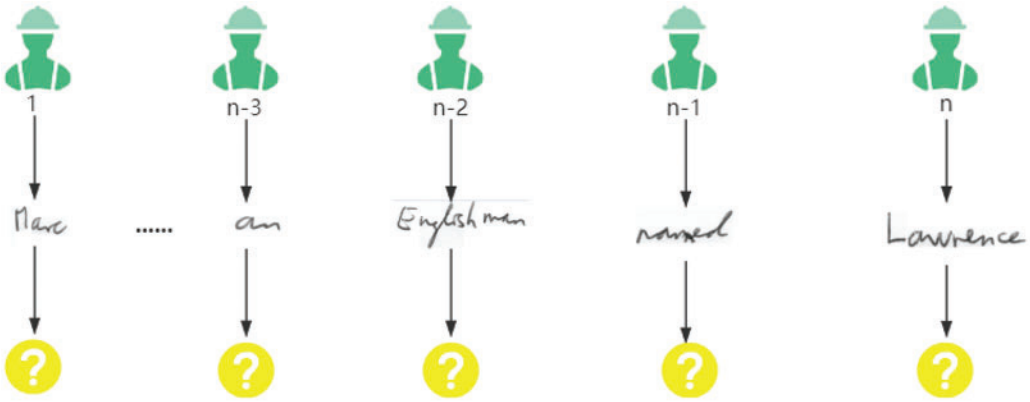


Figure 12: The process for workers to identify independent subtasks

5 Discussion

In the paper, we consider objective facts without typos as the ground truth of tasks. We conduct experiments on two crowdsourced datasets of handwriting recognition and audio transcription. The results demonstrate that our algorithm outperforms existing algorithms. Additionally, our algorithm flow can also handle other types of CSTs, such as sentiment analysis [22] and object localization [23].

In sentiment analysis, it is essential to infer the speaker's emotion based on the context information in the dialogue. Firstly, we take a context-related dialogue as a task and each sentence of a dialogue as a subtask. Workers can selectively mark sentences in a dialogue considering the context in the whole dialogue. Then, we integrate the emotional information labeled by workers to obtain the worker output matrix. After that, we use TAN and the third-party sentiment analysis library to get the frequency of any two emotional information appearing in a dialogue scene at the same time and the correlation between subtasks. Finally, the optimal emotion recognition results are obtained.

According to the similar process, in object localization, we can take an entire picture as a task, and each object to be identified in the picture as a subtask. Workers can recognize objects in the image and mark the objects' positions selectively. Then, we collect the answers to get the worker output matrix. Similarly, we use TAN and the third-party image analysis library to obtain the frequency of any two objects in the picture appearing in the same scene and the correlation between subtasks. Finally, the optimal object location recognition results are obtained.

6 Conclusion and Future Work

Context-Sensitive Task is a common type of complex tasks in crowdsourcing, but there are few studies on CSTs at present. The result inference algorithms in previous studies have certain defects. To solve this challenge, we propose a novel result inference algorithm P&T-Inf. P&T-Inf screens out candidate results for CSTs based on the partially ordered set and introduces contextual information from external knowledge combined with TAN Bayes to model the dependencies among subtasks of CSTs. It improves the accuracy of result inference effectively and reduces the time cost at the same time.

Introducing external knowledge to capture the context of CSTs is an efficient way, but the P&T-Inf algorithm has a very small part of low-precision outliers. This indicates that the introduction of external knowledge also introduces some errors, which are led to the exception in some inference results. How to avoid this situation is one of the future research directions.

Acknowledgement: This work was supported by the National Social Science Fund of China (Grant No. 22BTQ033).

Funding Statement: This work was supported by the National Social Science Fund of China (Grant No. 22BTQ033).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. Shu, X. Liu, X. Jia, K. Yang and R. H. Deng, "Anonymous privacy-preserving task matching in crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3068–3078, 2018.

[2] D. Prasetya and M. Z. C. Candra, "Microtask crowdsourcing marketplace for social network," in *Proc. 2018 5th Int. Conf. on Data and Software Engineering (ICoDSE)*, Lombok, Indonesia, pp. 1–6, 2018.

[3] C. Chai, J. Fan, G. Li, J. Wang and Y. Zheng, "Crowdsourcing database systems: Overview and challenges," in *Proc. 2019 IEEE 35th Int. Conf. on Data Engineering (ICDE)*, Macau SAR, China, pp. 2052–2055, 2019.

[4] S. Guo, R. Chen, H. Li and Y. Liu, "Capability matching and heuristic search for job assignment in crowdsourced web application testing," in *Proc. 2018 IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*, Miyazaki, Japan, China, pp. 4387–4392, 2018.

[5] Y. Fang, H. Sun, G. Li, R. Zhang and J. Huai, "Effective result inference for context-sensitive tasks in crowdsourcing," in *Proc. Int. Conf. on Database Systems for Advanced Applications*, Dallas, Texas, USA, pp. 33–48, 2016.

[6] J. Feng, G. Li, H. Wang and J. Feng, "Incremental quality inference in crowdsourcing," in *Proc. Int. Conf. on Database Systems for Advanced Applications*, Bali, Indonesia, pp. 453–467, 2014.

[7] M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman *et al.,* "Soylent: A word processor with a crowd inside," in *Proc. of the 23nd Annual ACM Symp. on User Interface Software and Technology. UIST'10*, New York, NY, USA, pp. 313–322, 2010.

[8] J. Zhang, V. S. Sheng, Q. Li, J. Wu and X. Wu, "Consensus algorithms for biased labeling in crowdsourcing," *Information Sciences*, vol. 382–383, no. 1, pp. 254–273, 2017.

[9] Y. Fang, H. Sun, G. Li, R. Zhang and J. Huai, "Context-aware result inference in crowdsourcing," *Information Sciences*, vol. 460–461, no. 4, pp. 346–363, 2018.

[10] Y. Fang, H. Sun, R. Zhang, J. Huai and Y. Mao, "A model for aggregating contributions of synergistic crowdsourcing workflows," in *Proc. Twenty-Eighth AAAI Conf. on Artificial Intelligence*, Quebec, Canada, pp. 3102–3103, 2014.

[11] Y. Jiang, Y. Sun, J. Yang, X. Lin and L. He, "Enabling uneven task difficulty in micro-task crowdsourcing," in *Proc. of the 2018 ACM Conf. on Supporting Groupwork*, Sanibel Island Florida USA, pp. 12–21, 2018.

[12] M. Salek, Y. Bachrach and P. Key, "Hotspotting—A probabilistic graphical model for image object localization through crowdsourcing," in *Proc. Twenty-Seventh AAAI Conf. on Artificial Intelligence*, Bellevue, Washington, USA, 2013.

[13] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma and J. Movellan, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *Proc. of the 22nd Int. Conf. on Neural Information Processing Systems. NIPS'09*, Vancouver, B.C., Canada, pp. 2035–2043, 2009.

[14] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin *et al.,* "Learning from crowds," *The Journal of Machine Learning Research*, vol. 11, pp. 1297–1322, 2010.

[15] G. Demartini, D. E. Difallah, P. Cudr′e-Mauroux and Zencrowd, "Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking," in *Proc. of the 21st Int. Conf. on World Wide Web. WWW'12*, Lyon, France, pp. 469–478, 2012.

[16] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, M. Blum *et al.,* "Human-based character recognition via web security measures," *Science*, vol. 321, no. 5895, pp. 1465–1468, 2008.

[17] L. Tran-Thanh, T. D. Huynh, A. Rosenfeld, S. D. Ramchurn and N. R. Jennings, "C rowdsourcing complex workflows under budget constraints," in *Proc. Twenty-Ninth AAAI Conf. on Artificial Intelligence*, Austin, Texas, USA, 2015.

[18] U. V. Marti and H. Bunke, "The IAM-database: An english sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39–46, 2002.

[19] V. Panayotov, G. Chen, D. Povey and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. 2015 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, Queensland, Australia, pp. 5206–5210, 2015.

[20] J. Kearns, "LibriVox: Free public domain audio-books," *Reference Reviews*, vol. 28, no. 1, pp. 7–8, 2014.

[21] T. Han, H. Sun, Y. Song, Y. Fang and X. Liu, "Find truth in the hands of the few: Acquiring specific knowledge with crowdsourcing," *Frontiers of Computer Science*, vol. 15, no. 4, pp. 1–12, 2021.

[22] S. Chen, C. Hsu, C. Kuo, T. Huang and L. Ku, "EmotionLines: An emotion corpus of multi-party conversations," in *Proc. of the Eleventh Int. Conf. on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, pp. 1597–1601, 2018.

[23] C. Galleguillos, B. McFee, S. Belongie and G. Lanckriet, "Multi-class object localization by combining local contextual interactions," in *Proc. 2010 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, pp. 113–120, 2010.