Crowdsourcing with Multiple-Source Knowledge Transfer

Guangyang Han[1] , **Jinzheng Tu**[1] , **Guoxian Yu**[1,*] , **Jun Wang**[1] and **Carlotta Domeniconi**[2]
[1]College of Computer and Information Sciences, Southwest University, Chongqing, China
[2]Department of Computer Science, George Mason University, VA, USA
{gyhan, tujinzheng, gxyu,kingjun}@swu.edu.cn, carlotta@cs.gmu.edu,

Abstract

Crowdsourcing is a new computing paradigm that harnesses human effort to solve computer-hard problems. Budget and quality are two fundamental factors in crowdsourcing, but they are antagonistic and their balance is crucially important. Induction and inference are principled ways for humans to acquire knowledge. Transfer learning can also enable induction and inference processes. When a new task comes, we may not know how to go about approaching it. On the other hand, we may have easy access to relevant knowledge that can help us with the new task. As such, via appropriate knowledge transfer, for example, an improved annotation can be achieved for the task at a small cost. To make this idea concrete, we introduce the Crowdsourcing with Multiple-source Knowledge Transfer (CrowdMKT) approach to transfer knowledge from multiple, similar, but different domains for a new task, and to reduce the negative impact of irrelevant sources. CrowdMKT first learns a set of concentrated high-level feature vectors of tasks using knowledge transfer from multiple sources, and then introduces a probabilistic graphical model to jointly model the tasks with high-level features, workers, and their annotations. Finally, it adopts an EM algorithm to estimate the workers' strengths and consensus. Experimental results on real-world image and text datasets prove the effectiveness of CrowdMKT in improving the quality and reducing the budget.

1 Introduction

Many data management and analytic tasks, such as protein structure prediction [Cooper *et al.*, 2010] and sentiment analysis [Liu *et al.*, 2012], cannot be solved effectively by existing computer-only algorithms. Crowdsourcing has emerged as an effective way to address such tasks by utilizing hundreds of thousands of Internet workers. Thanks to public crowdsourcing platforms, e.g., Amazon Mechanical Turk (AMT)[1] and CrowdFlower[2], we have easy access to the crowd.

Two key problems in crowdsourcing are quality control and budget saving. Workers engage into crowdsourcing platforms for different reasons [Li *et al.*, 2017]; some want to make some money while others participate just for fun. As such, many workers may answer questions randomly to maximize their income, while reliable (expert) workers may do their best to gain a sense of achievement and reputation. In addition, workers with different backgrounds have different specialties. A southeast Asian farmer with little education may not be as good at math as a college graduate, but the former has more experience in farming. If we have a task related to rice cultivation, then the farmer with little education may do better than the college student. To better accomplish tasks within limited budget, we need to jointly model the workers and the tasks, so that we can allocate appropriate tasks to suitable workers [Daniel *et al.*, 2018; Tu *et al.*, 2020a]. Some attempts have been made to jointly model workers and tasks to improve the quality and to save the budget [Kurve *et al.*, 2015; Tu *et al.*, 2020b; Yu *et al.*, 2020]. They model the difference between workers' skills and task difficulty. If a worker's skill exceeds the difficulty of a task, the worker is assigned to the task.

However, when we face a new domain, for which limited knowledge is available, it is hard to discover capable workers for the tasks. A natural thought is to borrow knowledge from other related domains (*i.e.,* transfer learning). For example, if we want to separate a pear from a pomelo, but we are not familiar with either, how could we learn the characteristics of the two quickly? We can resort to the known discriminative patterns between apples and oranges (as illustrated in Figure 1), and use them to differentiate a pear from a pomelo. In other words, we can use the intrinsic induction and inference ability of transfer learning to acquire new knowledge [Pan and Yang, 2009; Weiss *et al.*, 2016]. Motivated by this observation, we advocate the integration of transfer learning into crowdsourcing to handle tasks from a new domain by referring to tasks in other related but different domains. By extracting and transferring knowledge from the source domain to the target domain, we can draw a sketch of the task, achieve better task assignments and annotation quality, and also remedy the sparsity and the cold-start problem (too few data to build an initial stable mod-

[*] Corresponding author, guoxian85@gmail.com. This work is supported by NSFC (61872300 and 61873214).

[1] https://www.mturk.com/

[2] https://www.figure-eight.com/

(a) apple (b) orange (c) pear (d) pomelo

Figure 1: The differentiation between *pear* and *pomelo* can be made by knowledge transfer from the similar scenario of apple vs. orange (discriminative patterns: color, shapes, sarcocarp, etc.).

el) in crowdsourcing. However, ponderously using transfer learning is not always safe, since the transferred knowledge may not always be helpful for the target task [Rosenstein *et al.*, 2005].

In this paper, we introduce a two stage Crowdsourcing approach using Multiple-source Knowledge Transfer (CrowdMKT) to reduce the chance of negative transfer from a single source domain. CrowdMKT firstly extracts a set of concentrated high-level pattern vectors of the target task and other related tasks by sparse coding [Lee *et al.*, 2007]. We advocate that these high-level pattern vectors contain key features and important information for the target tasks. CrowdMKT then introduces a probabilistic graphical model based on the new patterns to jointly model the workers' ability, the collected annotations and the crowdsourcing workflow. Next, it infers the truth, workers' abilities and other parameters used in the model with an EM (expectation–maximization) algorithm. Via task transfer and truth inference, CrowdMKT can achieve a better understanding of the target tasks, and can assign tasks to workers more effectively. Our main contributions are summarized as follows:

- CrowdMKT combines crowdsourcing with multi-source transfer learning to make use of the external abundant and free knowledge to alleviate the lack of information in an unfamiliar task domain, and it enables the modeling of tasks and workers in a natural and reliable manner.
- CrowdMKT is the *first* approach to learn from multiple domains in crowdsourcing. As such, it maximizes the chance of useful and reliable knowledge transfer, and reduces the chance of negative transfer.
- By using free and abundant external knowledge, our CrowdMKT alleviates the cold start problem, and obtains higher quality annotations than state-of-the-art methods [Fang *et al.*, 2014; Zhang *et al.*, 2017; Demartini *et al.*, 2012; Koller and Friedman, 2009] and its variants.

2 Related Work

Our work is closely related to two branches of research, transfer learning and task assignment in crowdsourcing. Transfer learning aims to improve the performance of a learner in target domains by transferring knowledge embedded in different but related source domains [Pan and Yang, 2009; Weiss *et al.*, 2016]. As such, the dependence on a large number of target domain data can be alleviated for the target learner. Typical transfer learning focuses on knowledge transfer from samples, features, parameters, relations and the combination of them. To reduce the risk of negative knowledge transfer from a single source domain to the target domain, several multi-source transfer learning approaches have been invented and proved that transferring knowledge from multiple related but different source domains can boost the performance of the target domain [Yao and Doretto, 2010; Ding *et al.*, 2018], while they require the source domains having sufficient label information to evaluate the relatedness between source and target domains. Our CrowdMKT is built on multiple sources and feature-based transfer learning in an unsupervised manner to extract the high-level patterns of tasks and to reduce the chance of negative transfer.

An effective task assignment strategy helps to achieve high-quality annotation with a limited budget. Diverse strategies have been proposed [Li *et al.*, 2017], some focus on the individual worker's reliability and intention [Demartini *et al.*, 2012; Tu *et al.*, 2020a], task difficulty and information [Hu and Zhang, 2018], the cost of tasks [Gao *et al.*, 2013], both worker and task [Mavridis *et al.*, 2016; Yu *et al.*, 2020]. However, most of these efforts can not generalize to tasks in a new domain, since they all require sufficient accomplished tasks of individual workers for task assignment.

Some attempts have been made to plug transfer learning into crowdsourcing for better task assignment. Transfer learning for crowdsourcing [Mo *et al.*, 2013] borrows knowledge from auxiliary historical tasks to improve the data veracity for a target task and to relieve the sparsity and cold-start problem. However, it only focuses on the worker and requires workers with a lot of experience and job logging. Crowd-selection on Twitter [Zhao *et al.*, 2013] uses transfer learning to classify tasks and workers, and then treats the user's followers and followings on Twitter as potential workers to assign task. Domain vector estimation [Zheng *et al.*, 2016] analyzes the domains of a task with respect to the knowledge base (i.e., Wikipedia and Freebase) using transfer learning and utilizes the domain-sensitive worker model to accurately infer the true answer of a task. Active learning for crowdsourcing with knowledge transfer [Fang *et al.*, 2014] combines task domain and source domain data using sparse coding to get a more concise high-level expression and then uses active learning to select the right worker for the right task. However, it only transfers from one source domain and is vulnerable to negative transfer, where the source domain has negative impact on the task domain. The main reason is that these two domains are not so 'alike'; another possible reason is that the two domains contain similar noisy data.

To effectively handle tasks in new domains, we introduce CrowdMKT to transfer knowledge from multiple source domains and to increase the chance of positive transfer. CrowdMKT not only can jointly capture the intrinsic patterns of tasks, but also the specialty of workers, and thus contribute to better annotations and save the budget.

3 Methodology

3.1 Problem Definition

Let us consider a set of crowdsourcing tasks, and let $\mathcal{C}$ be the set of possible answers. W workers are invited to annotate the tasks using $|\mathcal{C}|$ distinct labels. The unknown ground truth of a task is denoted as $y_i \in \mathcal{C}$. Let $\mathcal{X}_t = \{\mathbf{x}_1^t, \mathbf{x}_2^t, \cdots, \mathbf{x}_{N_t}^t\}$ denote N_t tasks in the target domain. Each $\mathbf{x}_i^t \in \mathbb{R}^d$ denotes a target task. Similarly $\mathcal{X}_s = \{\mathbf{x}_1^s, \mathbf{x}_2^s, \cdots, \mathbf{x}_{N_s}^s\}(s = 1, \cdots, S)$ denotes a source domain. The collected annotations on N_t tasks from W workers form an annotation data matrix $\mathbf{A} \in \mathbb{R}^{N_t \times W}$, where each element $a_{iw} = c \in \{0\} \cup \mathcal{C}$ represents the fact that the w-th worker annotated the i-th task with label c. Specifically, $a_{iw} = 0$ states that the w-th worker does not provide any annotation for this task.

Given N_t tasks and a fixed budget, we want to use external free information to better model the tasks and the workers with the aim of completing the tasks with high quality answers and a reduced budget. To achieve this goal, we represent the raw feature vectors of target tasks using transfer learning on the feature vectors of tasks in other domains. After performing the transfer learning process (mainly via sparse coding with under-complete dictionary [Lee *et al.*, 2007]), the mutual features shared by the source and target domains are amplified and strengthened, and the noisy features are eliminated. By doing so, the positive knowledge transfer from multiple source domains to the target domain can be augmented and the raw vectors are concentrated. Each base in the dictionary potentially expresses some high-level discriminative patterns that help us to model the workers and the tasks. The following subsections describe the process in detail.

3.2 Transfer Learning Process

The transfer learning process aims at learning high-level patterns of tasks to better model tasks and hence workers by sparse coding (SC) [Lee *et al.*, 2007]. The basic idea of SC is to represent input vectors with a number of basis vectors (typically called dictionary). Suppose the number of basis vectors is k. Based on whether k is larger or smaller than the original task feature dimensiona d, we have an over-complete ($k > d$), a complete ($k = d$), or an under-complete ($k < d$) dictionary. The loss function of sparse coding is:

$$\min_{\mathbf{b}_j, \mathbf{z}_i} \sum\nolimits_i \|\mathbf{x}_i - \sum\nolimits_{j=1}^k z_{ij}\mathbf{b}_j\|^2 + \gamma\|\mathbf{z}_i\|_1, \quad s.t.\ \|\mathbf{b}_j\| \le c,\ \forall j \in 1, \cdots, k \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ is the raw feature vector, $\{\mathbf{b}_1, \cdots, \mathbf{b}_k\}$ is the dictionary of k distinct words, and each $\mathbf{b}_j \in \mathbb{R}^d$ is a basis vector. By linearly combining the bases, we get the new representation of $\mathbf{x}_i$ as $\mathbf{z}_i = [z_{i1}, \cdots, z_{ik}]$. The last term is an l_1-norm regularization enforcing the sparsity of $\mathbf{z}_i$.

Our problem has a different setting than the standard SC problem, where we want to find the shared features between source and target domains, and the new representation in the target domain is sparse. Eq. (1) should be applied to the two domains simultaneously with l_1-norm regularization. With an under-complete dictionary ($k < d$), we can compress input vectors to k-dimensional vectors without losing too much information; in this way the dimension inflation problem can be avoided. Eq. (1) is extended as follows:

$$\begin{aligned} \min_{\mathbf{b}_j, \mathbf{z}_i} & \sum\nolimits_{s=1}^S \sum\nolimits_{i=1}^{N_s} \|\mathbf{x}_i^s - \sum\nolimits_{j=1}^k z_{ij}^s \mathbf{b}_j\|^2 \\ & + \sum\nolimits_{i=1}^{N_t} \|\mathbf{x}_i^t - \sum\nolimits_{j=1}^k z_{ij}^t \mathbf{b}_j\|^2 + \gamma\|\mathbf{z}_i^t\|_1, \\ & s.t.\ \|\mathbf{b}_j\| \le 1,\ \forall j \in 1, \cdots, k \end{aligned} \tag{2}$$

The first and second parts are transfer learning processes taking place between the source and target domains. Since we only need the target domain feature vector, the l_1-norm regularization is only applied to the target domain, and γ controls the weight of this regularization. The norm constraint for bases ($\|\mathbf{b}_j\| \le 1,\ \forall j \in 1, \cdots, k$) is needed to prevent the following undesirable situation: there always exists a linear transformation of $\mathbf{b}_j$'s and z_{ij}^t's which keeps $\sum_{j=1}^k \mathbf{b}_j z_{ij}^t$ unchanged, while making z_i^j close to zero [Lee *et al.*, 2007].

The above equation can be rewritten in matrix form as follows:

$$\begin{aligned} \min_{\mathbf{B}, \mathbf{Z}^t} & \sum\nolimits_{s=1}^S \|\mathbf{X}^s - \mathbf{B}\mathbf{Z}^s\|_F^2 + \|\mathbf{X}^t - \mathbf{B}\mathbf{Z}^t\|_F^2 + \gamma\|\mathbf{Z}^t\|_1 \\ & s.t.\ \sum\nolimits_i |b_{i,j}| \le 1,\ \forall j \in 1,\ 2, \cdots, k \end{aligned} \tag{3}$$

where $\mathbf{X}^s$ and $\mathbf{X}^t$ are the original feature data matrices for the tasks, $\mathbf{Z}^s$ and $\mathbf{Z}^t$ are the representation matrices for the tasks in multiple source domains and the target domain. The columns of these matrices are the new feature vectors of the corresponding tasks. $\mathbf{B}$ contains the shared basis vectors (or dictionary) which we want to extract from all the tasks.

To optimize Eq. (3), we use the alternative optimization strategy, which optimizes one variable while fixing the other variables in an iterative manner. With $\mathbf{B}$ and $\mathbf{Z}^t$ fixed, the problem becomes an easy convex optimization problem with respect to $\mathbf{Z}^s$, with no sparse constraints on each source domain $\mathbf{X}^s$, and can be solved using a gradient method, such as steepest descent. Once $\mathbf{B}$ and all the $\mathbf{Z}^s$ are fixed, the problem becomes a classic l_1-norm regulation optimization problem, and we can use standard solvers (i.e., Least Angle Regression and Orthogonal Matching Pursuit [Wright *et al.*, 2010]). When $\mathbf{Z}^t$ and all the $\mathbf{Z}^s$ are fixed, the objective becomes a dictionary learning problem, and we can update each column of $\mathbf{B}$ using singular value decomposition or the least squares method. Our empirical study shows that the above optimization converges within 300 iterations.

3.3 Crowdsourcing Process

In this section, we propose a novel probabilistic generative model, illustrated in Fig. 2. This model describes the process that generates noisy annotations $\{a_{iw}\}_{i=1,w=1}^{N_t,W}$ of multiple workers with different expertise e_i^w based on the transferred task features $\{\mathbf{z}_i^t\}_{i=1}^{N_t}$. Our goal is to study the variational relationship between the workers' ability, the transferred high-level features of tasks, and the true labels.

Generation of true labels. We advocate that the transferred high-level task features $\mathbf{z}_i^t$ for $\mathbf{x}_i^t$ carry information concerning its ground truth. In other words, the ground truth

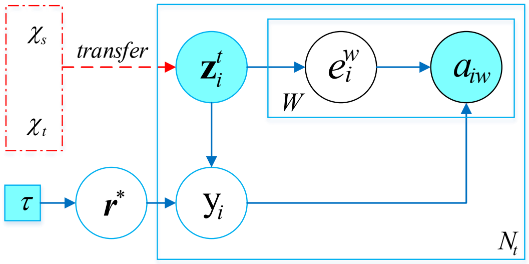


Figure 2: Probabilistic generative model of CrowdMKT. Circular nodes are random variables and square nodes are factor nodes. The shaded nodes represent observed values (worker annotations a_{iw} and transferred task features $\mathbf{z}_i^t$). The model describes the process of generating an answer a_{iw} for $\mathbf{z}_i^t$ with unknown ground truth y_i by worker w with expertise e_i^w. $\boldsymbol{r}^*$ is the vector that maps $\mathbf{z}_i^t$ to y_i and is determined by the hyper parameter τ.

of a task mainly depends on its features. To quantify the relationship between task features and the true label, a logistic regression is used as follows:

$$p(y_i \mid \mathbf{z}_i^t, \boldsymbol{r}^*) = (1+\exp(-\boldsymbol{r}^{*T}\cdot \mathbf{z}_i^t))^{-1} \tag{4}$$

where $\boldsymbol{r}^*$ is the parameter of the logistics regression. Of course, other approaches can also be adopted to quantify the relationship. Eq. (4) may be contaminated with additive noises, which can result in over-fitting. Thus, we assume a Normal prior on $\boldsymbol{r}^*$:

$$(\boldsymbol{r}^* \mid \tau) \sim \mathcal{N}(\mathbf{0}, \frac{1}{\tau}\mathbf{I}) \tag{5}$$

where $\tau > 0$ is a constant and tuned using a validation set. Other priors can also be readily adopted. For example, if $\boldsymbol{r}^*$ is expected to be sparse, the Laplacian prior can be used.

Generation of crowdsourced labels. In practice, workers with different expertise may annotate the same tasks differently, and the expertise of a worker is associated with the specific task features. Thus, we define $\mathbf{e}^w \in \mathbb{R}^k$ as the whole expertise and estimate the specific label-quality e_i^w of the w-th worker on the i-th task using a linear combination as follows:

$$e_i^w = (\mathbf{z}_i^t)^T \cdot \mathbf{e}^w \tag{6}$$

For simplicity, the conditional probability of the workers' expertise with respect to the task features is defined as follows:

$$p(e_i^w \mid \mathbf{z}_i^t) = (1+\exp(-e_i^w))^{-1} \tag{7}$$

We give an example to further reformulate Eqs. (6) and (7). Suppose that all the workers share the same four main skills ('politics', 'sports', 'science', and 'entertainment'), and two workers w_1 and w_2 have expertise $\mathbf{e}^1 = [0.8, 0, 0.8, 0]$ and $\mathbf{e}^2 = [0, 0.8, 0, 0.8]$, respectively. If the task '*Will IJCAI2020 be held in Yokohama, Japan?*' with high-level feature vector $\mathbf{z}_1^t = [0, 0, 1, 0]$ is assigned to w_1 and w_2, the quality of w_1 on $\mathbf{z}_1^t$ is $e_1^1 = \mathbf{z}_1^{t\,T} \cdot \mathbf{e}^1 = 0.8$, while the quality of w_2 on this task is $e_1^2 = \mathbf{z}_1^{t\,T} \cdot \mathbf{e}^2 = 0$. Using Eq. (7), we can estimate the conditional probabilities $p(e_1^1 \mid \mathbf{z}_1^t) = 0.690$ and $p(e_1^2 \mid \mathbf{z}_1^t) = 0.500$.

Eq. (7) quantifies the quality of the annotation provided by worker w to task i. A larger e_i^w value leads to a higher probability that a_{iw} will be consistent with the ground truth y_i. On the other hand, a smaller e_i^w results in a higher probability that worker w will make mistakes. Thus, the conditional probability $p\left(a_{iw} | y_i, e_i^w\right)$ can be defined based on the binomial distribution as follows:

$$p(a_{iw} \mid y_i, e_i^w) = p(e_i^w)\mathbb{I}(y_i = a_{iw}) + \frac{1-p(e_i^w)}{|\mathcal{C}|-1}\mathbb{I}(y_i \neq a_{iw}) \tag{8}$$

$|\mathcal{C}|$ is the number of alternative answers and $\mathbb{I}(\cdot)$ is the indicator function: $\mathbb{I}(x)$ returns 1, if x is true; and 0 otherwise. In Eq. (8), $p(e_i^w)$ represents the probability of obtaining a correct annotation, consistent with the ground truth y_i; $(1 - p(e_i^w))$ is the probability of receiving one of the $(|\mathcal{C}| - 1)$ incorrect annotations. Alternative estimators of the conditional probability are also suitable here. Our choice is driven by simplicity and intuitiveness.

Given a group of tasks $\mathcal{X}_t$, each worker w (with expertise $\mathbf{e}^w$) independently completes a subset of tasks of $\mathcal{X}_t$, and provides the answers a_{iw}. To approximate the above process, we define the conditional joint probability of our probabilistic model as follows:

$$p(\mathbf{A}|\boldsymbol{r}^*, \mathbf{e}^w, \mathbf{Z}^t) = \prod_i^{N_t} p(y_i \mid \mathbf{z}_i^t, \boldsymbol{r}^*) \prod_w^{W} p(e_i^w \mid \mathbf{z}_i^t) p(a_{iw} \mid y_i, e_i^w) \tag{9}$$

Note that the index i is used to differentiate different tasks, and not to indicate the order according to which the workers complete the tasks.

3.4 Parameter Inference Process

Given a set of extracted feature vectors $\{\mathbf{z}_i^t\}_{i=1}^{N_t}$ of tasks and collected annotations $\{a_{iw}\}_{i=1,w=1}^{N_t,W}$ of these tasks, we need to infer the corresponding ground truth $\{y_i\}_{i=1}^{N_t}$ and the workers' special ability measurement $\{\mathbf{e}^w\}_{w=1}^{W}$ based on the probabilistic graphical model. In other words, we have to infer two groups of variables in Ψ, $\Psi = \{\{y_i\}_{i=1}^{N_t}, \{\mathbf{e}^w\}_{w=1}^{W}\}$.

E-step: Based on Bayesian theorem, y_i corresponds to the largest posterior probability on the estimated ground truth $\widehat{p}(y_i)$, which can be transformed to the following equation:

$$\begin{aligned}\widehat{p}(y_i) &= p(y_i \mid \mathbf{e}_i, \boldsymbol{a}_i, \mathbf{z}_i^t, \boldsymbol{r}^*) \propto p(y_i, \mathbf{e}_i, \boldsymbol{a}_i \mid \mathbf{z}_i^t, \boldsymbol{r}^*) \\ &= p(y_i \mid \mathbf{z}_i^t, \boldsymbol{r}^*) \prod_w^{W} p(a_{iw} \mid y_i, e_i^w) p(e_i^w \mid \mathbf{z}_i^t)\end{aligned} \tag{10}$$

All the variables in Eq. (10) are taken from Ψ obtained in the last M-step. After computing all possible values of y_i and the corresponding $\widehat{p}(y_i)$, we choose the y_i with largest $\widehat{p}(y_i)$ as ground truth.

M-step: In the M-step, we need to find the two groups of variables in Ψ, which maximize the logarithm of the posterior expectation on the ground truth y_i obtained in the last E-step:

$$\begin{aligned}\Psi &= \max_{\Psi} \mathbb{E}_{\mathbf{y}}[\log(p(\mathbf{z}_i^t, \mathbf{e}_i, \boldsymbol{a}_i \mid y_i, \boldsymbol{r}^*))] \\ &= \max_{\Psi} \sum\nolimits_i^{N_t} \sum\nolimits_w^{W} \mathbb{E}_{y_i}[\log p(e_i^w \mid \mathbf{z}_i^t) \\ &+ \log p(y_i \mid \mathbf{z}_i^t, \boldsymbol{r}^*) + \log p(a_{iw} \mid y_i, e_i^w)]\end{aligned} \tag{11}$$

This is an unconstrained maximization problem and we solve it using the L-BFGS algorithm [Fletcher, 2013]. We iterative perform the above two steps until a maximum number of iterations is reached, or until the change in parameter values is small enough.

4 Experiments

4.1 Experimental Setup

We study the effectiveness of our CrowdMKT through a series of experiments on two real-world datasets (20-newsgroups and CUB-200-2011 [Wah *et al.*, 2011]) with multiple source and target domains. The statistical information of the two datasets is listed in Table 1. For the Image dataset, we use VGG-19 [Simonyan and Zisserman, 2014] to obtain its raw feature vector; for the Text dataset, we use TF-IDF to obtain the raw feature vector. We fix the feature dimension of all domains to $d = 1000$, and set the dictionary size to $k = 20$. Note, CrowdMKT can work on domains with diverse numbers of samples.

The following seven methods are used for experimental comparison:
(i) **MV** directly uses majority vote to integrate annotations, without transfer learning or modeling tasks and workers.
(ii) **AWMV** [Zhang *et al.*, 2017] utilizes the frequency of positive labels in the multiple noisy label sets of each task to estimate a bias rate, and then assigns weights derived from the bias rate to negative and positive labels.
(iii) **ZC** [Demartini *et al.*, 2012] uses one parameter to model the reliability of each worker and infers the true labels of tasks using an EM algorithm.
(iv) **PGM** is a probabilistic generative model (as shown in Fig. 2) [Koller and Friedman, 2009] built on the raw task features without transfer learning.
(v) **STL+ALM** [Fang *et al.*, 2014] takes advantage of single-source transfer learning, active learning, and a PGM alike ours. We exclude the active learning component, and use the same training data as ours to train this model.
(vi) **SKT** is a degenerated version of CrowdMKT; it performs single-source knowledge transfer for crowdsourcing and then adopts PGM on the new feature vector $\mathbf{z}^t$.
(vii) **CrowdMKT** employs transfer learning with multiple source domains to extract the high-level features, and then adopts PGM to infer the truth labels and workers' capability.

The first four methods build consensus models without using transfer learning, and the other three use transfer learning. For CrowdMKT and its variants, we simply set $\gamma = 0.1$ and $\tau = 1$. We empirical found that $\gamma \in [0.05, 0.5]$ gives a good and stable result, while a too small value can not ensure a sparse feature vector, and a too large value causes too sparse regularization. For the other methods, we set the parameters (if any) following the suggestions of the authors. Following the canonical setting [Kazai *et al.*, 2011], we simulate four types of workers (spammer, random, normal, and expert), with different capacity ranges and proportions as shown in Table 3. The weighted average capacity is 0.6. We generate 50 workers and ask each worker to give 24 annotations to the image tasks, each of which has to have at least one annotation. As a result, each task on average has five annotations.

Worker type	Lower range	Upper range	Proportion
spammer	0.1	0.25	10%
random	0.25	0.5	10%
normal	0.5	0.8	70%
expert	0.8	1.0	10%

Table 3: Experimental setup of worker compositions and capacity ranges.

4.2 Consensus Results

We independently run each method ten times to compute the consensus annotation and report the average accuracy in Table 2. We have several important observations.
(i) **Multi** vs. **single**-source transfer: CrowdMKT uses three source domains and gains a performance superior to its degenerate version MKT, which uses any two source domains. MKT in turns outperforms SKT, which uses any single domain. This confirms the fact that our multiple-source knowledge transfer indeed reduces the chance of negative transfer, and the source domains are complementary and related with the target domain. Although STL+ALM uses the first (the best) source domain for the target domain and has a similar transfer idea as SKT, its accuracy is still lower than SKT. This is because SKT (and also CrowdMKT) can better model workers' specialty, latent ground-truth, and collected annotations, by a binomial distribution (via Eq. (8)), as to STL+ALM, it models these factors using a Gaussian probability-density function.
(ii) **Transfer** vs. **non-transfer**: SKT works better than PGM, which using original task features instead of the high-level features extracted by transfer learning. This contrast supports the advantage of knowledge transfer strategy for approaching tasks in the target domain. PGM has an accuracy comparable to ZC and to AWMV, and outperforms MV by a large margin. MV has an accuracy of 0.610, closing to the weighted average capacity 0.6 (as shown in Table 3). This supports the effectiveness of the consensus algorithms and the contribution of knowledge transfer. The accuracy of ZC and AWMV are lower than that of SKT, which considers a numeric vector to model the capability of a worker, while ZC and AWMV do not. ZC only uses the target domain and adopts a binary value to model the workers' quality, thus ignoring the fact that a worker's ability may vary across different tasks. Like MV, AWMV does not model workers and tasks, but it considers the bias of tasks' labels, so its accuracy is higher than MV.

SKT(1) and SKT(2) have a higher accuracy than SKT(3) on the Image dataset, which implies that the third source domain is less related to the target domain than the other two. In fact, looking at Table 1, we find that the first category ('Albatross') of the target domain is also within the first and second source domains, but not in the third source domain. But combining the third source domain with others can still boost the target domain. This fact again confirms that multi-source transfer can reduce the chance of negative transfer.

Fig. 3 provides another example that demonstrates the effect of multi-source knowledge transfer. 'Black_footed_albatross' comes from the target domain. We can discriminate the black-footed albatross from other birds by extracting and transferring knowledge from three birds of three different source domains. Collectively, the latter three birds provide the different key patterns (black plumage, markings around beak and under eye,

Image dataset	
Target domain (N_t = 240)	Source domains (N_s = 240, 240, 240)
Black_Footed_Albatross, Eared_Grebe Pileated_Woodpecker, Acadian_Flycatcher	Laysan_Albatross, Horned_Grebe, Red_Bellied_Woodpecker, Least_Flycatcher
	Sooty_Albatross, Pied_Billed_Grebe, Red_Headed_Woodpecker, Yellow_Bellied_Flycatcher
	Brandt_Cormorant, Western_Grebe, Pileated_Woodpecker, Great_Crested_Flycatcher
Text dataset	
Target domain (N_t = 4000)	Source domains (N_s = 4000, 4000, 4000)
comp.os.ms-windows.misc, sci.crypt rec.motorcycles, talk.politics.guns	comp.sys.ibm.pc.hardware, rec.autos, sci.electronics, talk.politics.mideast
	comp.sys.mac.hardware, sci.med, rec.sport.baseball, talk.politics.misc
	comp.windows.x, sci.space, rec.sport.hockey, talk.religion.misc

Table 1: Datasets used in the experiments. Each dataset has one target domain and up to three source domains.

	MV	AWMV	ZC	PGM	STL+ALM	SKT(1)	SKT(2)	SKT(3)	MKT(1+2)	MKT(1+3)	MKT(2+3)	CrowdMKT
Image	.6100	.6832	.6521	.6638	.6813	.7019	.7106	.6831	.7356	.7281	.7300	.7801
Text	.6688	.7189	.6854	.7263	.6743	.7548	.7440	.7435	.7855	.7773	.7760	.8120

Table 2: Experiment results (Accuracy) of compared methods. SKT(s) only uses the source domain s, and MKT($s+u$) uses two source domains, which are ordered in Table 1. The standard deviations are always < 0.001 and thus excluded.

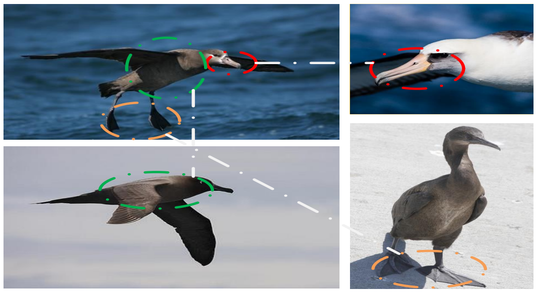


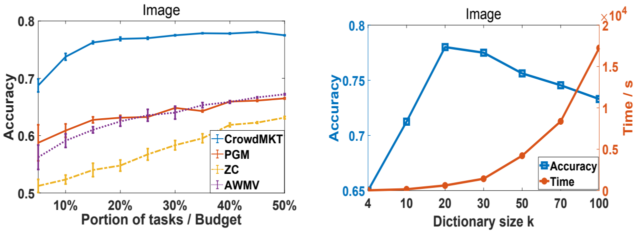
Figure 3: An example of multi-source knowledge transfer for characterizing a bird in the *target* domain of the Image dataset. The top left bird (Black_footed_albatross) belongs to target domain, and has three discriminative patterns: black plumage, markings around beak and under eye, and black foot (from *Wikipedia*), which can be collectively learned and transferred from birds in three source domains.

and black foot) that characterize the target bird.

4.3 Further Analysis

By transfer learning, we gain useful knowledge to initialize the model, and thus we need fewer instances to achieve a stable model. We evaluate the relief from the cold start problem by comparing the needed number of training instances to obtain a stable model on the Image dataset. We gradually add 5% training data per round and then repeat PGM, ZC, AWMV, and CrowdMKT ten times for each round. The results can be seen in Fig. 4(a). As expected, the accuracy of all methods increases as the number of completed tasks grows. With 25% tasks, our CrowdMKT reaches a stable (variance < 0.001) and superior performance, while $\geq$45% is needed for PGM. Both ZC and AWMV need a larger number of accomplished tasks (budget) to reach a stable model (more than 50% data). This study confirms that multi-source knowledge transfer can alleviate the cold start problem, and thus can reduce the budget needed to set up a stable model by requiring fewer annotated tasks, once the model being set up, fewer workers are needed to work on one task, the overall budget can be saved.

We conduct additional experiments to study the impact of the dictionary size (k) and show the results in Fig. 4(b). The best accuracy is achieved when $k \in [20, 30]$. A larger k not only reduces the accuracy, but also increases the runtime, where runtime is recorded on a PC with WinOS 10, AMD Ryzen 7 2700x and 16GB RAM. This is not surprising, since the dictionary size k is associated with the categorization of workers' specialities; a too small k cannot capture discriminative high-level features (specialities), while a too large k tends to over-categorize the specialities.



(a) Accuracy vs. labeled tasks. (b) Accuracy & runtime vs. k.

Figure 4: (a). Consensus accuracy vs. annotated tasks. To achieve a stable and superior performance, CrowdMKT needs fewer annotated tasks than methods without knowledge transfer; (b) Consensus accuracy under different dictionary sizes k for knowledge transfer.

We also tested the four types of workers with other capacity ranges and proportions. The results lead to similar conclusions. In addition, we observed that CrowdMKT is more robust to spammers than other compared methods.

5 Conclusions

In this paper, we leverage the inductive and inference ability of transfer learning to transfer knowledge from multiple source domains to tackle new tasks in unfamiliar domains. We introduce a multi-source knowledge transfer and probabilistic generative model-based solution (CrowdMKT) that learns high-level patterns of tasks to facilitate the crowdsourcing process. Experimental results show that CrowdMKT achieves annotations of a higher quality than other competitive methods at a reduced cost. Our future work will investigate techniques to establish whether target and source domains are similar, and if yes, how. The first question answers whether we should transfer, and the second answers what we should transfer.

References

[Cooper *et al.*, 2010] Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, et al. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756, 2010.

[Daniel *et al.*, 2018] Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Computing Surveys*, 51(1):7, 2018.

[Demartini *et al.*, 2012] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *WWW*, pages 469–478, 2012.

[Ding *et al.*, 2018] Zhengming Ding, Ming Shao, and Yun Fu. Incomplete multisource transfer learning. *TNNLS*, 29(2):310–323, 2018.

[Fang *et al.*, 2014] Meng Fang, Jie Yin, and Dacheng Tao. Active learning for crowdsourcing using knowledge transfer. In *AAAI*, pages 1809–1815, 2014.

[Fletcher, 2013] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

[Gao *et al.*, 2013] Jinyang Gao, Xuan Liu, Beng Chin Ooi, Haixun Wang, and Gang Chen. An online cost sensitive decision-making method in crowdsourcing systems. In *SIGMOD*, pages 217–228, 2013.

[Hu and Zhang, 2018] Zehong Hu and Jie Zhang. A novel strategy for active task assignment in crowd labeling. In *IJCAI*, pages 1538–1545, 2018.

[Kazai *et al.*, 2011] Gabriella Kazai, Jaap Kamps, and Natasa Milic-Frayling. Worker types and personality traits in crowdsourcing relevance labels. In *CIKM*, pages 1941–1944, 2011.

[Koller and Friedman, 2009] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[Kurve *et al.*, 2015] Aditya Kurve, David J. Miller, and George Kesidis. Multicategory crowdsourcing accounting for variable task difficulty, worker skill, and worker intention. *TKDE*, 27(3):794–809, 2015.

[Lee *et al.*, 2007] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *NeurIPS*, pages 801–808, 2007.

[Li *et al.*, 2017] Guoliang Li, Yudian Zheng, Ju Fan, Jiannan Wang, and Reynold Cheng. Crowdsourced data management: Overview and challenges. In *SIGMOD*, pages 1711–1716, 2017.

[Liu *et al.*, 2012] Xuan Liu, Meiyu Lu, Beng Chin Ooi, Yanyan Shen, Sai Wu, and Meihui Zhang. Cdas: a crowdsourcing data analytics system. *VLDB*, 5(10):1040–1051, 2012.

[Mavridis *et al.*, 2016] Panagiotis Mavridis, David Gross-Amblard, and Zoltán Miklós. Using hierarchical skills for optimized task assignment in knowledge-intensive crowdsourcing. In *WWW*, pages 843–853, 2016.

[Mo *et al.*, 2013] Kaixiang Mo, Erheng Zhong, and Qiang Yang. Cross-task crowdsourcing. In *KDD*, pages 677–685, 2013.

[Pan and Yang, 2009] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *TKDE*, 22(10):1345–1359, 2009.

[Rosenstein *et al.*, 2005] Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. To transfer or not to transfer. In *NuerIPS workshop on Transfer Learning*, pages 1–4, 2005.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Tu *et al.*, 2020a] Jingzheng Tu, Guoxian Yu, Jun Wang, Carlotta Domeniconi, and Xiangliang Zhang. Attention-aware answers of the crowd. In *SDM*, pages 451–459, 2020.

[Tu *et al.*, 2020b] Jinzheng Tu, Guoxian Yu, Carlotta Domeniconi, Jun Wang, Guoqiang Xiao, and Maozu Guo. Multi-label crowd consensus via joint matrix factorization. *KAIS*, 62(4):1341–1369, 2020.

[Wah *et al.*, 2011] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.

[Weiss *et al.*, 2016] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–9, 2016.

[Wright *et al.*, 2010] John Wright, Yi Ma, Julien Mairal, Guillermo Sapiro, Thomas S Huang, and Shuicheng Yan. Sparse representation for computer vision and pattern recognition. *Proc. of IEEE*, 98(6):1031–1044, 2010.

[Yao and Doretto, 2010] Yi Yao and Gianfranco Doretto. Boosting for transfer learning with multiple sources. In *CVPR*, pages 1855–1862, 2010.

[Yu *et al.*, 2020] Guoxian Yu, Jinzheng Tu, Jun Wang, Carlotta Domeniconi, and Xiangliang Zhang. Active multi-label crowd consensus. *TNNLS*, 99(1):1–12, 2020.

[Zhang *et al.*, 2017] Jing Zhang, Victor S Sheng, Qianmu Li, Jian Wu, and Xindong Wu. Consensus algorithms for biased labeling in crowdsourcing. *Information Sciences*, 382:254–273, 2017.

[Zhao *et al.*, 2013] Zhou Zhao, Da Yan, Wilfred Ng, and Shi Gao. A transfer learning based framework of crowd-selection on twitter. In *KDD*, pages 1514–1517, 2013.

[Zheng *et al.*, 2016] Yudian Zheng, Guoliang Li, and Reynold Cheng. Docs: a domain-aware crowdsourcing system using knowledge bases. *PVLDB*, 10(4):361–372, 2016.