International Journal of Crowd Science

Quality Assessment in Crowdsourced Classification Tasks

Received (22 June 2019)
Revised (16 August 2019)

Abstract

Purpose: Ensuring quality is one of the most significant challenges in microtask crowdsourcing. Aggregation of the collected data from the crowd is one of the important steps to infer the correct answer but the existing study seems to be limited to the single-step task. This study looks at multiple-step classification tasks and understands aggregation in such cases, hence is useful for assessing the classification quality.

Design/methodology/approach: We present a model to capture the information of the workflow, questions, and answers for both single-question and multiple-question classification tasks. We propose an adapted approach on top of the classic approach so that our model can handle tasks with several multiple-choice questions in general instead of a specific domain or any specific hierarchical classifications. We evaluate our approach with three representative tasks from existing citizen science projects in which we have the gold standard created by experts.

Findings: The results show our approach can provide significant improvements to the overall classification accuracy. Our analysis also demonstrates that all algorithms can achieve higher accuracy for the volunteer- versus paid-generated datasets for the same task. Furthermore, we observed interesting patterns in the relationship between the performance of different algorithms and workflow specific factors including the number of steps, and the number of available options in each step.

Originality/value: Due to the nature of crowdsourcing, aggregating the collected data is an important process to understand the quality of crowdsourcing results. Different inference algorithms have been studied for simple microtasks consisting of single questions with two or more answers. However, as classification tasks typically contain many questions, our proposed method is able to to be applied to a wide range of tasks including both single-question and multiple-question classification tasks.

Keywords: crowdsourcing; human computation; quality assessment; classification; aggregation.

Paper type: **Research paper**

1. Introduction

Microtask crowdsourcing has attracted interest from researchers, businesses and government as a means to leverage human computation into their activities in a fast, accurate and affordable way. In the last ten years, we have seen it applied to anything from spotting sarcasm on social media to discovering new galaxies and helping digitise large cultural heritage collections. The underlying model is relatively straightforward: a problem is decomposed into smaller chunks that can be

tackled independently by several people. Their individual outputs are then compared and consolidated into a final solution [Shahaf and Horvitz (2010)]. However, none of these steps is actually easy: some problems are less amenable to microtasking and need to be turned into bespoke microtask workflows [Bernstein *et al.* (2010); Kulkarni *et al.* (2011); Kittur *et al.* (2011)]; the performance of the crowd varies across tasks [Mao *et al.* (2013); Redi and Povoa (2014)]; and determining which answers are the most useful ones can be both complex and computationally expensive [Kittur *et al.* (2008); Snow *et al.* (2008); Vickrey *et al.* (2008); Demartini *et al.* (2012); Wiggins *et al.* (2011)]. It is on this last aspect, determining the correct answers, that we focus on in this paper. The aggregation method proposed in this paper is able to infer the correct answer for a range of tasks involving either single-step or multiple-step classifications when gold answers are not available. It also serves as a proxy to help task requesters to assess the quality of the crowdsourced results when they already have some gold answers, such as piloting specific multiple-step task design before putting it online for a larger scale.

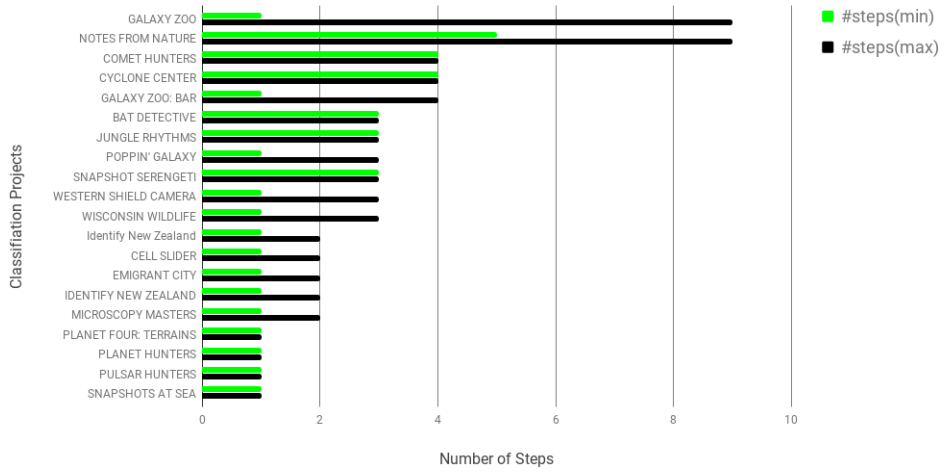


Fig. 1: Classification tasks from Zooniverse

Quality Assessment in microtask crowdsourcing refers to the evaluation of quality of the workers' work. First, quality can be assessed based on different criteria, as it has many dimensions [Kahn *et al.* (2002); Batini *et al.* (2009)]. Under the crowdsourcing context, it depends on the type of the data, which is decided by the task type [Malone *et al.* (2010); Gadiraju *et al.* (2014); Gadiraju *et al.* (2015)]. The most common quality metric we have seen is to calculate the accuracy [Bernstein *et al.* (2010); Gelas *et al.* (2011); Hung *et al.* (2013); Zhang *et al.* (2017)] with available gold standards. However, in lots of the cases

the gold standard is not available. This is where different inference algorithms come into picture which help to infer or predict the correct (gold) answer. Second, quality assessment can be done either on the fly [Ipeirotis *et al.* (2014)] during the task running that can be used to optimize task assignment hence reduce cost, or in the post aggregation [Whitehill *et al.* (2009); Ipeirotis *et al.* (2010); Bachrach *et al.* (2012); Difallah *et al.* (2015)] to assess the overall quality of the classification. This work focus on aggregating the result after the crowdsourcing task has been completed, so that accuracy can be calculated based on the gold standards we have.

There are many different types of tasks where microtask crowdsourcing are applied [Eickhoff and de Vries (2011); Difallah *et al.* (2015); Yang *et al.* (2016); Zheng *et al.* (2017)]. We focus on inferring the correct answer for a classification task which is one of the most popular type of crowdsourcing tasks. We are by no means the first to do so; previous research has proposed a range of methods to infer and predict the quality of crowd answers [Bachrach *et al.* (2012); Dawid and Skene (1979); Difallah *et al.* (2015); Hare *et al.* (2013); Ipeirotis *et al.* (2010); Karger *et al.* (2011); Loni *et al.* (2014); Paulheim and Bizer (2014); Hung *et al.* (2013); Rosenthal and Dey (2010); Simpson *et al.* (2013); Whitehill *et al.* (2009)]. Whilst all methods have their benefits, they work on relatively simple task models that consist of single questions with one or more answers [Sheshadri and Lease (2013); Hung *et al.* (2013); Zhang *et al.* (2017); Zheng *et al.* (2017)]. The scenario we are targeting is different. We take a close look at existing classification tasks from Zooniverse, and notice a large percentage of these tasks are multiple-step tasks, as shown in Figure 1. In fact, in a random sampling of 20 tasks, only 20% has a single question. Consider the example in Figure 2, which is taken from a labelled citizen science project in which pictures taken in the Serengeti national park in Tanzania are analysed online by thousands of volunteers.[a] The crowd is asked to answer a series of related, independent questions about what they see in the image, including the types and number of animals.

Our work is motivated by a range of online crowd science classification projects. Each of them uses a slightly different type of task to classify an object, for example, an image, according to a number of criteria. For a relatively complex task, it is split into several steps, typically in the form of multiple-choice answers. Sometimes there are dependencies between steps as the answer chosen for one questions prompts other questions to be displayed. For instance, in the Cities at Night project, which uses microtask crowdsourcing to analyse night-time photographs taken by astronauts onboard the ISS[b], seven different *Options* are provided for the first question to identify what the given image contains, a city, stars, aurora, astronaut, black image, no photo or none of these, and only when "city" is identified, two more independent questions will be asked to classify cloudiness (three *Options*: cloudy, someclouds, clear) and sharpness (two *Options*: sharp, blurry) respectively. In the

[a] `https://www.snapshotserengeti.org/`
[b] `http://citiesatnight.org/`

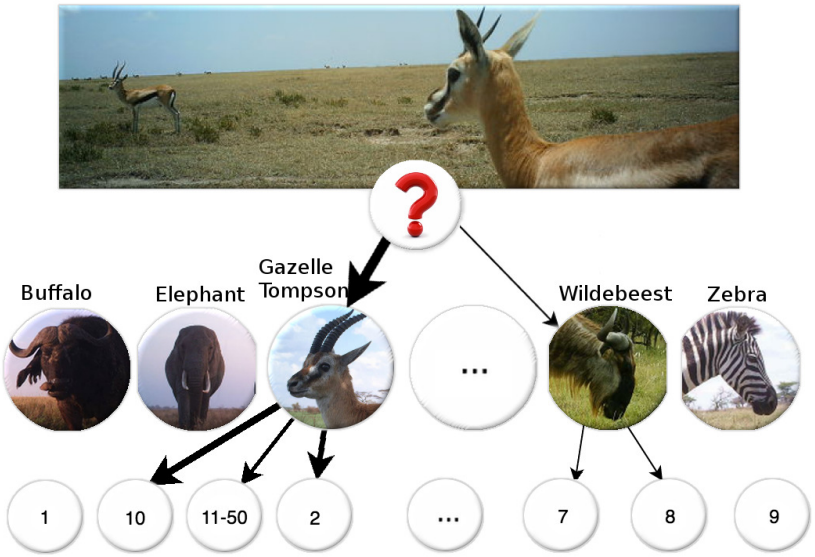


Fig. 2: Example classification paths collected from 20 workers for a given photo. The crowd is asked to choose the animal type and estimate how many animals are in the picture. Wider arrows indicate options that are popular with the crowd.

GalaxyZoo[c] project, several different questions were asked in sequence depending on the answers to previous questions, and questions and answers are arranged in a decision tree. It has a more complex workflow in which more questions are involved, and questions vary based on what has been chosen in previous classification step. For instance, the first question is "Is the galaxy simply smooth and rounded, with no sign of a disk?" and three options are provided: "Smooth", "Features or disk", and "Star or artifact". When choosing "Smooth", a new question will be asked "How rounded is it?" and available options are "Completely round", "In between" and "Cigar shaped". If "Features or disk" is chosen as the answer to the first question, a different set of subsequent questions will be asked. Other times, workflows are rather sequences of independent, though related questions, such as what we see in Snapshot[a] (in Figure 2). Determining the correct answer for such complex classification task can be tricky and has not been fully studied yet. Existing research also does not investigate how inference methods could affect the classification accuracy when using different crowd types for complex classification tasks. As a result, there is the need to understand whether different algorithms and aggregation strategies are required for different crowd contexts.

In order to tackle the issue of determining the correct answer from crowd pro-

[c] https://www.galaxyzoo.org/

duced annotations for the classification task with multiple questions, we model the problem of complex classification tasks that span over multiple, related questions as a graph. To the best of our knowledge, we are the first to propose using the structure of a microtask crowdsourcing workflow as an additional feature to support inference algorithms in making decisions about correct labels, using output data produced by the crowd. We look at three inference algorithms (majority voting [Paulheim and Bizer (2014); Hung *et al.* (2013)], message passing [Karger *et al.* (2011)] and expectation maximisation [Dawid and Skene (1979); Whitehill *et al.* (2009)]), which have been commonly used in answer inference in microtask crowdsourcing previously. We adapt these algorithms to work on the graph modeled from crowdsourcing tasks with multiple steps. We perform a large-scale evaluation of the performance of these algorithms on six datasets across two crowd contexts from three image classification tasks: Darkskies[b], GalaxyZoo[c] and Snapshot Serrengeti[a]. The rationale behind choosing datasets from both volunteer and paid crowd context is that algorithms may perform differently in these contexts. The experiments show that our aggregation strategy achieves significantly better performance than the current approach of naively applying individual algorithms on each node level. The result also indicates that majority voting, despite its simplicity, compares well with more sophisticated approaches that consider additional factors such as user performance and hence need more computation time. Sophisticated algorithms such as expectation maximization, however, can complement majority voting for relatively complex tasks. We also prove that each algorithm obtains better inference accuracy in the volunteer context compared to paid crowdsourcing context.

This rest of this paper is structured as follows: Section 2 provides the foundations of existing algorithms which we have adapted to handle answer inference in classification tasks with multiple questions, and illustrate how these aggregation fits in the quality assessment process. In section 3, we explain our graph model and notations used in the graph, formalize the classification problem, and elaborate our aggregation approach. In section 4, we perform large-scale evaluation and demonstrate the performance of different algorithms. Section 5 discusses our findings. Section 6 reviews existing work which has inspired our research and section 7 summarizes our result and future work.

2. Foundations

A classification task generally has one single question and a few options to choose from, such as the one shown in Figure 3. It looks like a simple tree structure where the classification starts with a root node which refers to the object to be classified and has a few branches which represent the available options. In this section, we present three existing algorithms, MV, MP and EM, that have been used in inferring the true label for a single-step multiple-choice classification task. These are the foundations to understand our proposed adapted approach. Notations used in

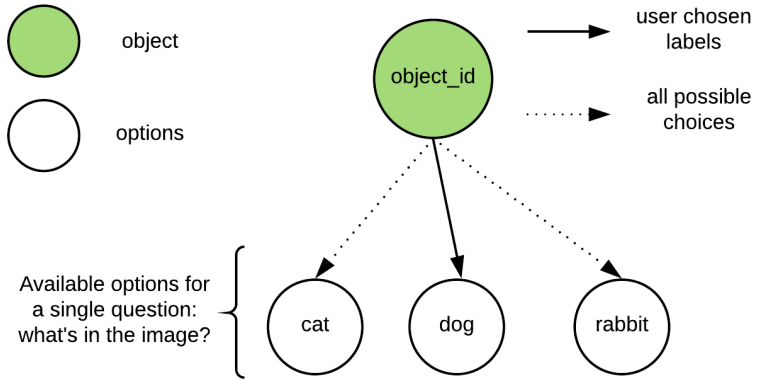


Fig. 3: Representation of a task with a single question

elaborating these algorithms are defined in table 1. For the sake of explaining the individual algorithms and our method, we use following notations throughout this paper.

Table 1: Notations

Notation	Definition
o	the current object being classified
O	the set of all objects in a dataset
A	all available options
u	user u
U	the set of all users who contributed to the current dataset
U_o	all users who have classified object o
L	all labels received from the crowd, and $L \subseteq A$
L_o	the set of all labels from the crowd for object o
L^u	the set of all labels from user u
l_o^u	the label for object o from user u
$\hat{l}_o$	the inferred label for object o

2.1. *Majority Voting (MV)*

Due to its simplicity, Majority Voting has been used in many microtask projects [Hung *et al.* (2015); Liu *et al.* (2012)] and is the standard aggregation method in some existing crowdsourcing platforms[d]. Given the list of options for a labelling task and an object, the MV algorithm chooses those options with the highest number of

[d] `https://success.crowdflower.com/hc/en-us/articles/203527635-CML-Attribute-Aggregation`

votes from the crowd. Formally, it takes as input an object o and the crowd labels L_o and outputs the resulting candidate label $\tilde{l_o}$ that received the most votes from the users.

Algorithm 1 Majority voting (MV)

procedure FINDUNIQUELABEL(L_o)
  $L_{unique} \leftarrow \{l_o^u\}$, where $L_{unique} \subseteq A$ and $l_o^u \in$ A and $u \leq U_o$;
$\tilde{l_o} =$ "";
$num_{max} = 0$;
for $i \in |L_{unique}|$ **do**
  if $count(l_{unique(i)}) \geq num_{max}$ **then**
    $num_{max} \leftarrow count(l_{unique(i)})$;
    $\tilde{l_o} \leftarrow l_{unique(i)}$;
return $\tilde{l_o}$;

2.2. *Expectation Maximisation (EM)*

Expectation Maximisation (EM) is another algorithm that has been widely used and involves two steps to infer the true label for a given object. In the first step, the true label for the current object is estimated using simple majority voting, where the input of all users is considered equally. Then, in the next step, the error rate of each user is estimated based on this result and used in turn to calculate the new estimation for the first step. The steps are alternating iteratively until the algorithm converges and a maximum is found. It takes as input an object o and all labels L. It starts by estimating the true label for each object and each user's error rate by comparing their answers (using an indicator function $I()$ to check whether the user classifies object to a certain category/class) for all objects they have looked at. The error rate is used subsequently to update the confusion matrix for each user. The output is candidate labels for o with the *probability* (indicated by p) of the corresponding candidate label to be correct.

2.3. *Message Passing (MP)*

Message Passing (MP) is an algorithm that takes into account both the labels and the performance of the users. MP constructs object and user-specific messages to represent the reliability of the particular user, and iteratively updates the object and the user messages. More specifically, at each object update, it adds up more weight to labels that come from more trustworthy parts of the crowd; and at each user update, it adds more trust (a confidence value) to the user if the labels they give for other objects are in line with the current estimates of object labels. The iterative updates continue until the algorithm converges or a specified threshold

Algorithm 2 Expectation maximisation (EM)

procedure INITIALISE(p_l)
  $p_l \leftarrow count(l) \div |L_o|$ ▷ probability of l being the true label for object o (l $\in$ A);
while not converged **do**
  Estimate error rate for user u:
  $\theta^u_{ll^-} \leftarrow \lambda^u_{ll^-} + \sum_{o \in L_o} p_l \times I(l^u_o = l^-)$
  Estimate confusion matrix:
  $e^u_{ll^-} \leftarrow \theta^u_{ll^-} \div \sum_q \theta^u_{lq}$ ▷ q is the accuracy of user u
  Estimate class priors:
  $pr_l \leftarrow \sum_o p^o_l \div |O|$
  Calculate class probability for object o:
  $p_l \leftarrow pr_l \prod_{u \in U_o} \prod_m (e^j_{am}) I(l^u = m) \div \sum_q pr_q \prod_m (e^u_{qm}) I(l^u = m)$
$\tilde{l_o} =$ "";
$p_{max} = 0$;
for $l \in A$ **do**
  if $p_l \geq p_{max}$ **then**
    $p_{max} \leftarrow p_l$;
    $\tilde{l_o} \leftarrow l$;
return $\tilde{l_o}$;

is hit. The threshold for the stopping condition is a parameter that has to be empirically determined. It takes as input an object o, a label $a \in A$, all labels received from the crowd L and a threshold k_{max}. MP computes the object message by firstly iterating all previous labels from the users who have been assigned the object o and then looking at whether each label is the same as the given one. In a next step, it uses the object message $x_{o->u}$ ($\in L$) to update the user message $y_{u->o}$ ($\in L$), which is computed by iterating over the labels they have submitted. Until convergence, the object message for object o is aggregated by weighing the user messages (confidence) for that object and the computed sign is stored in E_{ou}. MP outputs the candidate label l for o and the *sign* of whether the label applies or not. A detailed description of the algorithm can be found in [Karger *et al.* (2011)]. Whilst providing accurate estimations, MP is also known for its high computational costs as the number of labels and users increase.

2.4. *Quality assessment*

In the microtask crowdsourcing context, achieving a good quality result is one of the major goals and when we talk about quality it generally means the quality of the data collected from the crowd. For the classification microtasks, existing work in quality assessment mostly use the accuracy metric [Khattak and Salleb-Aouissi (2011); Hung *et al.* (2013); Zhang *et al.* (2017)]. Some research also uses

Algorithm 3 Message passing (MP)

procedure INITIALISATION($y_{u->u}$)
  for $(o, u) \in L$ **do**
    Initialise $y_{u->o}$ ($\sim \mathcal{N}(-1, 1)$) ;
procedure ITERATION(k_{max})
  for $k \in \{1, ..., k_{max}\}$ **do**
    for $(o, u) \in L$ **do**
      $x^k_{o->u} \leftarrow \sum_{u^- \in U} E_{ou^-} \times y^{k-1}_{u->o}$ ($u^- \neq$u)
    for $(o, u) \in L$ **do**
      $y^k_{u->o} \leftarrow \sum_{o^- \in O} E_{o^-u} \times x^k_{o->u}$ ($o^- \neq$o)
$x_o \leftarrow \sum_{u \in U} E_{ou} \times y^{k_{max}-1}_{u->o}$
if sign(x_o) == 1 **then**
  $\tilde{l_o} = x_o$
  return $\tilde{l_o}$

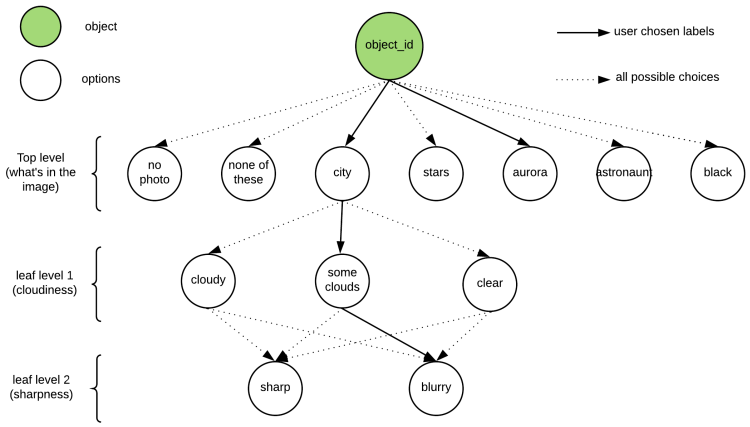


Fig. 4: Representation of Dark Skies workflow from Cities at Night

precision/recall [Hung *et al.* (2015); Zhang *et al.* (2017)] or F1 score [Zheng *et al.* (2017)], while other work use ROC [Zheng *et al.* (2017)] or RMSE [Bachrach *et al.* (2012)]. For classification, the quality of the result refers to how good the overall collected classifications are, which is a data-value centric dimension to reflect how accurate the classifications are. In this work, if not specially specified, when referring to quality of the input/answer/data/result, it means Accuracy – "The degree to which data values correctly represent the real-world facts" [Zaveri *et al.* (2013)]; definition in science [JCGM (2008)] as "closeness of agreement between a measured quantity value and a true quantity value of a measurand". We can look at individual crowd worker's work to evaluate whether its work is of good quality, or we can look

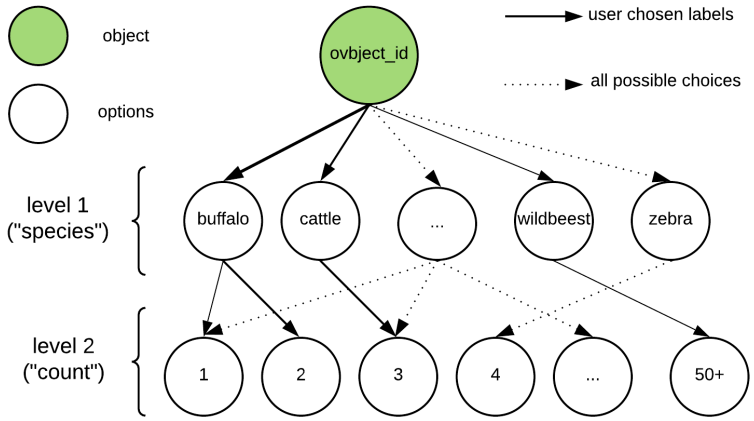


Fig. 5: Representation of Snapshot Sergenti workflow from Zooniverse

at the overall result from all the workers to see how accurate they classify the given objects. The later one which involves aggregating the input from different crowd workers in a multiple-step classification task is the focus of this paper.

In the crowdsourcing context, the ground truth is not usually available. In order to assess the quality of the result, we need to understand what algorithms or mechanisms can be used to infer or predict the correct answer based on all the input from the crowd workers. Correspondingly each existing different algorithm has been studied by researchers and evaluated its performance in various contexts (section 6.2). This work mainly takes a look at three popular existing algorithms elaborated above, and investigates how the adaptation of these algorithms can be used for aggregating the crowdsourced data and hence help to assess the quality of the classification result. The whole process, in a nutshell, includes three major phases, data collection (microtask design and task execution) from the crowd which is available to this study, aggregation to infer the correct answer/label, and evaluation of the quality (in this work is the Accuracy metric) by comparing the inferred result to the gold standards we have. This research focuses on the aggregation and evaluates the accuracy accordingly.

3. Our Approach

In this section, we first illustrate the range of classification tasks we address via a set of examples: classification tasks with a single question and multiple-questions. We then introduce a set of notations and formalize the classification problem as a path searching problem in a graph. Following that, we present our aggregation method by illustrating how existing established algorithms can be adapted to handle more complex cases.

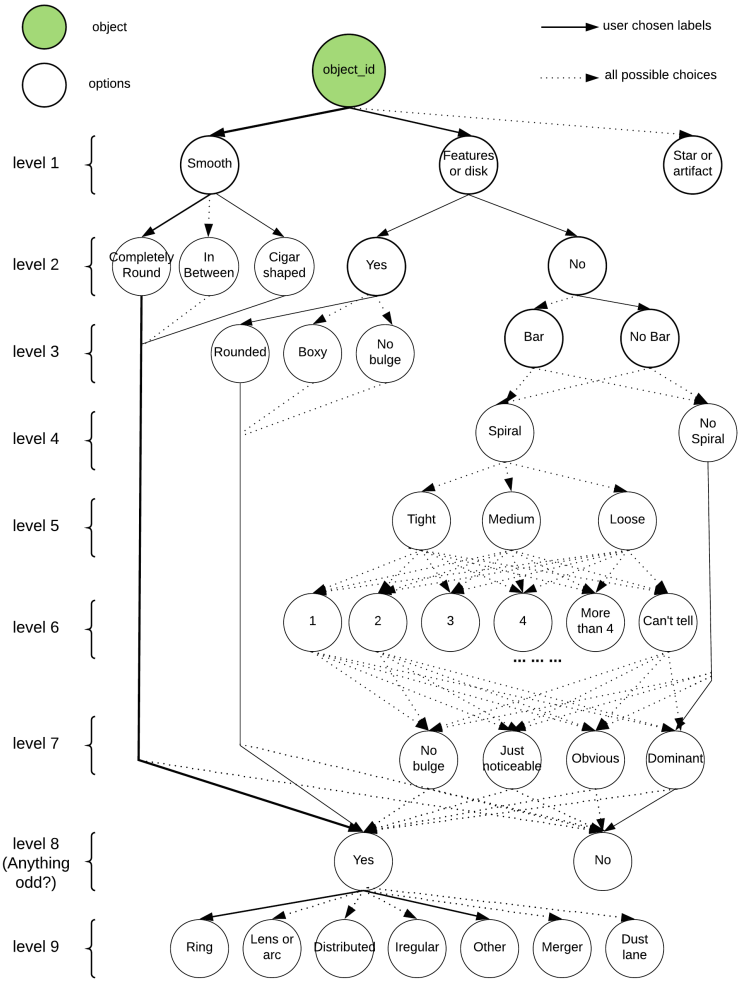


Fig. 6: Representation of GalaxyZoo workflow from Zooniverse

3.1. *Multi-level workflow model and problem formalisation*

A classification task, as shown in Figure 3, is generally considered as a simple task as it contains only one question. A relatively complex task normally involves more than one question and hence more options. It will be more like a tree with branches which has further branches and leaves.

If we draw such a 'tree' for the three tasks we are exploring in this paper, we can see each of them uses a different type of workflow consisting of several independent/interdependent steps. Each step in the workflow is associated with a *Question* to classify an object according to a criterion. To answer the question the crowd needs to choose among a set of *Options*. Figure 4 involves minimum one step

and maximum three steps for the classification task. Figure 5 has a fixed two steps to complete a classification task and each step has more than ten options. For the GalaxyZoo[e] task, it can involve minimum one step and a maximum of nine steps to complete a classification, as shown in Figure 6. It is notable though these different tasks have various number of questions, and various number of available options, there are indeed nodes which have more than one parent node.

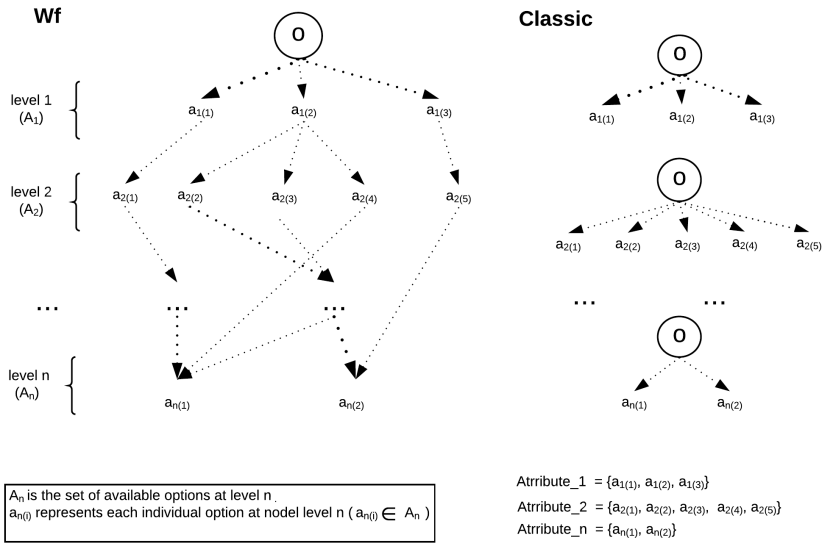


Fig. 7: Graph representation of an example classification workflow W_f vs. the corresponding classic way of looking at the classification with multiple questions

As a result, the workflow can be modelled as a directed acyclic graph (DAG), where the root node is the object under consideration and all other nodes are classification *options*. Each node can be reached via multiple paths from the root, which prompts the first question of the workflow.[f] For a given object o, the crowd is asked to carry out a *labelling task*, which implies answering a series of (independent or dependent) classification questions with a set of *labels* which identify the outstanding features of the object being classified. We define this task as a path search problem in a workflow W_f modelled as directed acyclic graph (DAG) with a *root* entry point and *levels* (similar to tree levels, representing the number of questions in the task), each corresponding to a set of *options* as depicted in Figure 7. Each *node* in such a graph represents a particular labelling *option*. The labelling finishes when a leaf in

[e] `https://data.galaxyzoo.org/gz_trees/gz_trees.html`

[f] In a lot of cases, the workflows are tree-shaped, but some cases are not a tree such as the three tasks presented above.

the graph is reached, that is a label that does not lead to any further questions. In our definition, the *level* corresponds to classification question(s) and the *level* of a *node* is serialized and counted at the lowest level. We use *level* exchangeably with *depth* of a node which is indicated by the number of edges from the node to the root node. A directed *edge* represents a label chosen for the corresponding question related with that node level. Table 2 has a summary of the definitions we use.

On top of the notations we defined in section 2, we also define the notations which are specific to our workflow graph model in table 3. The problem we are solving in the paper can be defined as follows:

Table 2: Definitions

Term	**Definition**
Task	a general term referring to an action or a series of action need to be executed.
Classification Task	task classifying objects into given categories, it could be a simple task (one question) or a relatively complex task (more than one question).
Microtask	a task is decomposed into smaller unit making it easier for the crowd. One microtask is equivalent to one question in classification task.
Workflow	microtasks are arranged/chained in a way to automatically complete the task
Question	classification task asked of the user to elicit/assign a label to an attribute of the object to be classified
Option	the set of possible labels
Chosen Option	an option user chooses per question
Correct Label	the correct label for a question
Chosen Path	a user chooses a set of labels for entire workflow
Correct Path	the correct set of labels for entire workflow
Workflow Graph	the workflow can be modelled as a directed acyclic graph (DAG), in which the root node represents the object under consideration and all other nodes are classification options
Node	a representation of an option in our model
Node Level	the sequence that the question is presented to the user within a workflow

Definition 3.1. *The Correct Labelling Problem:*
Given a particular object o, a workflow-based graph W_f, a set of labels L_o for object o , and (optionally) a set of previous labels from all users on all objects L, our aim is to infer the correct label path $\tilde{L_o}$ in W_f for object o.

3.2. *Adapted aggregation*

In the classic approaches, it does not look at the dependency between node levels hence naively putting inferred result from each node level together does not guarantee a valid result. It is obvious that producing a valid path with possible choices should improve the accuracy of the users. As such, a basic adaptation of the classic

Table 3: Notations Specific to Our Model

Notation	Definition
W_f	represents the graph based on the workflow of classifying object o, it has node levels to indicate the questions to classify the corresponding attributes of the given object, and nodes to represent the options available for each attribute
$A_{(n)}$	represents the available options at node level n
$a_{n(j)}$	represents the individual option at node level n, where j $\in \{1, ..., \lvert A_{(n)} \rvert\}$
$l_{o}{}^{u}_{(n)}$	represents the label chosen by user u at node level n for object o. Thus, the labelling result ($l_{o}{}^{1}_{(1)}$,$l_{o}{}^{1}_{(2)}$,...,$l_{o}{}^{1}_{(n)}$) will represent the ordered list of nodes (the traversal path) visited by user 1 when classifying o , which is called as a label *path*
$L_{o}{}^{u}$	the label path chosen by user u for object o
$L_{o(n)}$	all labels for object o at node level n
$L_{o(n)(unique)}$	unique labels for object o at node level n, $L_{o(n)(unique)} \subseteq A_{(n)}$
$\tilde{L_o}$	represents the inferred label path for object o. It is a set of inferred labels for each node level described as ($\tilde{l_{o_{(1)}}}, ..., \tilde{l_{o_{(n)}}}$)
L_{gold_o}	true label path for object o

algorithms should show some improvement over multiple level workflows. We show such a basic adaptation in Algorithm 4.

Algorithm 4 Our Adapted Approach

procedure PREDICT_BY_NODELEVEL(L_o)
  $num\_levels = n$;
  for $level \in range(n)$ **do**
    if $method == mv$ **then**
      procedure FINDUNIQUELABEL(L_o)
        $L_{unique} \leftarrow \{l_o^u\}$, where $L_{unique} \subseteq A$ and $l_o^u \in$ A and $u \leq U_o$;
      for $l \in L_{unique}$ **do**
        $p_l \leftarrow count(l) \div |L_i|$ ▷ percentage of l being voted as the label for object o;
      return $LC_n \leftarrow \{(l, p_l)\}$ ▷ list of candidate labels and their percentage for o;
    if $method == em$ **then**
      procedure INITIALISE(p_l)
        $p_l \leftarrow count(l) \div |L_o|$ ▷ percentage of l being the true label for object o (l $\in$ A);
      while not converged **do**
        Estimate error rate for user u:
        $\theta^u_{ll^-} \leftarrow \lambda^u_{ll^-} + \sum_{o \in L_o} p_l \times I(l^u_o = l^-)$
        Estimate confusion matrix:
        $e^u_{ll^-} \leftarrow \theta^u_{ll^-} \div \sum_q \theta^u_{lq}$ ▷ q is the accuracy of user u
        Estimate class priors:

$pr_l \leftarrow \sum_o p_l^o \div |O|$
Calculate class probability for object o:
$p_l \leftarrow pr_l \prod_{u \in U_o} \prod_m (e_{am}^j) I(l^u = m) \div \sum_q pr_q \prod_m (e_{qm}^u) I(l^u = m)$
return $LC_n \leftarrow \{(l, p_l)\}$ ▷ list of label candidates and corresponding probability for o;
if *method* == *mp* **then**
procedure INITIALISATION($y_{u->u}$)
for $(o, u) \in L$ **do**
Initialise $y_{u->o}$ ($\sim \mathcal{N}(-1, 1)$) ;
procedure ITERATION(k_{max})
for $k \in \{1, ..., k_{max}\}$ **do**
for $(o, u) \in L$ **do**
$x_{o->u}^k \leftarrow \sum_{u^- \in U} E_{ou^-} \times y_{u->o}^{k-1}$ ($u^- \neq$u);
for $(o, u) \in L$ **do**
$y_{u->o}^k \leftarrow \sum_{o^- \in O} E_{o^-u} \times x_{o->u}^k$ ($o^- \neq$o);
$x_o \leftarrow \sum_{u \in U} E_{ou} \times y_{u->o}^{k_{max}-1}$
if sign(x_o) == 1 **then**
$LC_n.append((x_o, 1.0))$
procedure ASSEMBLE_MOSTPOSSIBLEPATH(L_o)
$num\_levels = n$;
$LC = \{\}$;
for $z_1 \in LC_1$ **do**
for $z_2 \in LC_2$ **do**
...
for $z_n \in LC_n$ **do**
$LC.append((z_1, z_2, ..., z_n), (p_{z_1} \times p_{z_2} ... \times p_{z_n}))$;
$\tilde{L_o} =$;
$p_{max} = 0$;
for $Z \in LC$ **do**
if $p_Z \geq p_{max}$ **then**
$p_{max} \leftarrow p_Z$;
$\tilde{L_o} \leftarrow Z$;
return $\tilde{L_o}$;

Our *adapted* approach assumes that labels at different levels in the workflow are independent, then assemble the label path from each node level based on the workflow graph. In the *adapted* approach, not only we reward partially correct answers from the crowd by applying each of the algorithms at each node level in the graph and compute scores for each individual labels, but also we consider the valid path when inferring the correct path. We also specially choose two algorithms that take into account the performance of the crowd in their computations, *EM*

and MP. The EM algorithm sums up all node probabilities along each path to determine the ranking score. The MP algorithm returns *true* if that particular label at the node level is relevant or $false$ otherwise. This means that we assign the score for the candidate paths correspondingly either as 1.0 or 0.0. By studying it, we want to allow MP and EM to be able to better identify those users who, while not doing so well overall, are very skilled at a particular sub-task (question) in the workflow.

4. Evaluation

To evaluate the three algorithms and our adapted approach, we compare the classic approach where algorithms are applied on each node level and simply put together (we call it "naive-approach" here) with our "adapted-approach" which utilises classic approach while strives to infer a valid correct path by considering the workflow graph. Thus we have six different approaches: *mv_adapted*, *mv_naive*, *mp_adapted*, *mp_naive*, *em_adapted*, *em_naive*. Each inference algorithm was applied to six datasets with different microtask crowdsourcing workflows. We start with the evaluation setup of the data in section 4.1 and the evaluation metrics in section 4.2. Then we present the evaluation of inferred result in section 4.3.

4.1. *Data*

First, we used three existing datasets. The first one is from the Snapshot Serengeti[a] project and consists of all crowd classifications within the time span from December 10th 2012 until July 17th, 2013. It contains $7,800,896$ labels from $890,280$ volunteers for a total of $66,892$ objects. For our evaluation, we used a gold standard with curated labels for $4,149$ objects, which was created by professional scientists working on the Snapshot Serengeti project. To evaluate our approach we took all labels received from the crowd for the $4,149$ objects which contains $112,027$ labels submitted by $8,304$ volunteers. The second dataset is from the Dark Skies app within the Cities at Night[b] project. It consists of $1,275,354$ classifications by $19,818$ volunteers submitted in a time span from April 27th, 2014 until December 5th, 2016. The gold standard consisted of 200 objects whose labels were manually validated by the science team in Cities at Night. These 200 objects received $1,341$ labels from 692 users from CrowdCrafting[g]. The third one is from the GalaxyZoo[c] project where we randomly choose 500 objects consisting classifications from February 16th, 2009 to May 21st, 2009. The workflows for the three datasets are depicted in Figures 4,5 and Figure 6, respectively. In order to explore the effects of volunteers/paid context on the results, the tasks are also setup on paid crowdsourcing platform to mimic the tasks done by volunteers.

[g] `https://crowdcrafting.org/`

4.2. *Metric*

To measure the performance of our aggregation approach, we employ the *Accuracy* metric which has been commonly used in classification evaluation in previous work [Khattak and Salleb-Aouissi (2011); Kamar *et al.* (2012); Sheshadri and Lease (2013); Hung *et al.* (2013); Zhang *et al.* (2017); Zheng *et al.* (2017)]. Accuracy is a measure allowing us to understand the percentage of correct answers (inferred by algorithms). The accuracy is defined as the percentage of objects that have been correctly inferred. Higher accuracy indicates better performance.

$$Accuracy = \frac{\sum_{o}^{|O|} Bernoulli(L_{gold_o} == \tilde{L}_o)}{|O|}$$

The above equation is by default for calculating the accuracy for the inferred label path. $Bernoulli(L_{gold_o} == \tilde{L}_o)$ indicates the outcome (either 0 or 1) of comparing gold category with the category predicted by different predictor. As we use the *adapted* node-level based implementation, it makes sense to also evaluate how accurate the inferred label is on each node level. In such context, $L_{gold_o}[n]$ represents the ground truth for object o at node level n and $\tilde{L}_o[n]$ represents the inferred true label at node level n. Hence the accuracy at node level n for the top answer can be calculated by:

$$Accuracy_{\_level_n} = \frac{\sum_{o}^{|O|} Bernoulli(L_{gold_o}[n] == \tilde{L}_o[n])}{|O|}$$

To understand whether our adapted approach is significantly better, we will also run significant testing for all algorithms chosen. We will use standard 5% significance level. For each dataset, we will randomly select 100 objects and select 50 times. The accuracy for each selection is calculated for MV,MP and EM for both *naive* and *adapted* approach. We will use the function scipy.stats.ttest_ind from Python[h] to perform the two-sided test for *naive* and *adapted* samples in all six cases (three workflows, each has two contexts: volunteer and paid).

4.3. *Results*

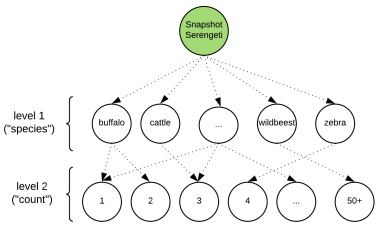
Table 4 shows the accuracy of each algorithm on each dataset for the inferred answer. Considering the overall classification accuracy (by path), our adapted methods have better performance than the naive approach in both volunteer and paid crowd context; at the same time, each algorithm generally has higher accuracy for volunteer context compared to the paid crowd. Note that the best accuracy achieved increases as the depth of the workflow increases for the paid crowd context, where Serengeti with two questions achieves 45.9%, darkskies with three questions achieves 53.0% and galaxyzoo with maximium of nine questions achieves 57.9%. Similar pattern

[h] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html`

is not observed for the volunteer context. If looking at the accuracy breakdown by node level (Figure 8,9 and 10), it is notable that for mutiple-questions task with more steps, *adapted* method of MP and EM generally shows better accuracy at most of the node levels. For the datasets from a task with fewer steps in its workflow (less number of levels in the graph), such as the Serengeti task in Figure 8, MV performs better.

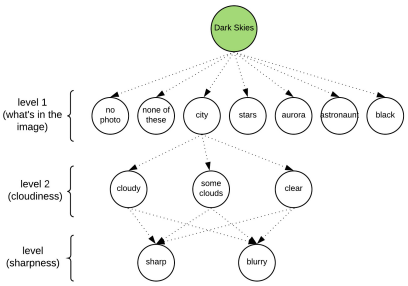
Table 4: Accuracy (by path) of each algorithm

dataset	graph depth/size	crowd type	algorithm	accuracy
serengeti	54-11	volunteer	mv_naive	0.590
			mv_adapted	0.776
			em_naive	0.572
			em_adapted	0.655
			mp_naive	0.755
			mp_adapted	0.755
		paid	mv_naive	0.299
			mv_adapted	0.459
			em_naive	0.244
			em_adapted	0.337
			mp_naive	0.083
			mp_adapted	0.207
darkskies	8-3-2	volunteer	mv_naive	0.690
			mv_adapted	0.785
			em_naive	0.040
			em_adapted	0.450
			mp_naive	0.340
			mp_adapted	0.495
		paid	mv_naive	0.405
			mv_adapted	0.530
			em_naive	0.020
			em_adapted	0.385
			mp_naive	0.335
			mp_adapted	0.305
galaxyzoo	3-3-2-3-2-2-3-6-4-2-7	volunteer	mv_naive	0.554
			mv_adapted	0.631
			em_naive	0.470
			em_adapted	0.564
			mp_naive	0.002
			mp_adapted	0.562
		paid	mv_naive	0.371
			mv_adapted	0.579
			em_naive	0.000
			em_adapted	0.331
			mp_naive	0.002
			mp_adapted	0.367



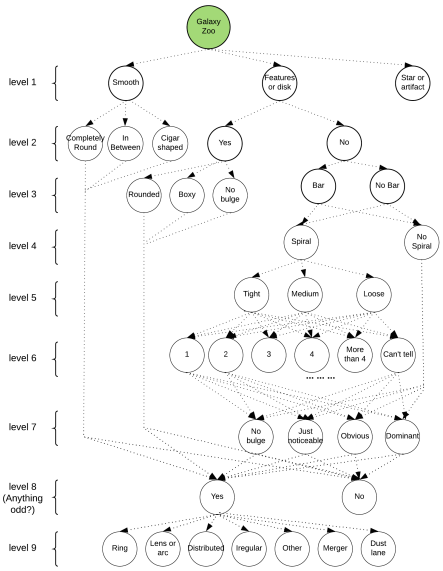
Accuracy												
Volunteer						Paid					
mv_naive	mv_adapted	mp_naive	mp_adapted	em_naive	em_adapted	mv_naive	mv_adapted	mp_naive	mp_adapted	em_naive	em_adapted
0.851	0.977	0.979	0.824	0.834	0.919	0.549	0.740	0.112	0.240	0.476	0.599
0.697	0.789	0.767	0.781	0.690	0.720	0.483	0.549	0.661	0.411	0.489	0.544

Fig. 8: Accuracy by node level (Serengeti)



Accuracy											
Volunteer						Paid					
mv_naive	mv_adapted	mp_naive	mp_adapted	em_naive	em_adapted	mv_naive	mv_adapted	mp_naive	mp_adapted	em_naive	em_adapted
0.770	0.844	0.599	0.655	0.050	0.665	0.530	0.675	0.635	0.520	0.035	0.584
0.854	0.915	0.520	0.635	0.609	0.805	0.760	0.824	0.694	0.429	0.800	0.765
0.910	0.959	0.814	0.739	0.670	0.770	0.775	0.864	0.699	0.555	0.699	0.734

Fig. 9: Accuracy by node level (Darkskies)



Accuracy											
Volunteer						Paid					
mv_naive	mv_adapted	mp_naive	mp_adapted	em_naive	em_adapted	mv_naive	mv_adapted	mp_naive	mp_adapted	em_naive	em_adapted
0.859	0.845	0.877	0.839	0.817	0.849	0.625	0.781	0.873	0.675	0.212	0.609
0.666	0.714	0.036	0.734	0.574	0.67	0.464	0.649	0.036	0.569	0.278	0.426
0.863	0.855	0.841	0.807	0.837	0.857	0.751	0.851	0.841	0.569	0.793	0.627
0.875	0.861	0.851	0.837	0.877	0.865	0.775	0.867	0.851	0.605	0.883	0.845
0.907	0.905	0.899	0.841	0.911	0.907	0.903	0.909	0.899	0.589	0.917	0.915
0.915	0.909	0.899	0.829	0.899	0.905	0.903	0.915	0.899	0.543	0.921	0.919
0.865	0.853	0.851	0.789	0.857	0.859	0.781	0.865	0.851	0.537	0.863	0.837
0.841	0.869	0.036	0.815	0.869	0.863	0.669	0.827	0.036	0.721	0.805	0.779
0.851	0.887	0.873	0.767	0.855	0.875	0.813	0.877	0.873	0.527	0.727	0.821

Fig. 10: Accuracy by node level (Galaxyzoo)

Meanwhile, from the table 4 we can see MV shows an acceptable accuracy for most of the volunteered datasets (mostly over 75%, except for GalaxyZoo dataset), but has poor accuracy (less than 60%) in the paid crowd context though it performs better than other individual algorithms we tested, which suggests it need to be complemented by other methods which might be good at specific objects where MV cannot perform well. Looking at the accuracy by level results, it does not seem to suggest that as the depth of the task (number of levels) increases, accuracy has a tendency to consistently increase or decrease. The accuracy of each level is more relevant to its intrinsic character (eg, number of options in that level, and ambiguity or subjectivity of the corresponding object). For instance, the darkskies task asks the user to evaluate the sharpness and cloudiness of the image, which can be subjective to some degree. This is also why the result by node level seems to show an interesting picture that on different node level for different workflow, sometimes *em* has the best result (such as level 4 and 5 of GalaxyZoo), sometimes *mp* has the best result (such as level 1 of Serengeti in volunteer case), other times *mv* has the best result (level 1,2,3 of Darkskies in both volunteer and paid context).

Notice that MP for the darkskies paid crowd context, it is the only case we observe that the *naive* approach has higher overall accuracy (by path) than *adapated*, which is due to the fact that both the level 2 and 3 (determining cloudiness and sharpness of the image) of darkskies workflow are in essence independent questions of the first node level (whether it is a city, or stars or anything else) though the task workflow made it a subsequent question only when "city" is chosen as the label for first node level. Similarly, the accuracy by level result from *mp_adapated* is lower than *mp_naive* on a few other occasions at different node level, but in those occasions, there is always one node level *mp_naive* has considerably poor accuracy, such as in Galaxyzoo node level 2, which subsequently leads to the very low overall accuracy considering the whole path. The reason that the *mp_adapted* approach could have lower accuracy at certain level is that *mp* approach actually only returns 1.0 or 0.0 to indicate whether that is the predicted label, but our adapted approach tried to assemble/infer a most probable valid label path (as shown in algorithm 4) based on the candidate of predicted labels from individual node level. So for the *mp* case, the randomness of ranking the combinations might not do well for the corresponding node level, however, the overall accuracy has shown to be better than the *naive* approach which completely neglects the validity of a label path.

Notice that though our *adapted* approaches achieve higher accuracy for the first node level in most case, *mv_adapted* has slightly lower accuracy comparing to *mv_naive* for GalaxyZoo workflow under volunteer context, which is because the way we assemble the result is based on the overall possibility (percentage of voting at each node level multiplied) of a path instead of assuming the top voted label at node level 1 is correct (and then traversing subsequent node based on that assumption). Our main purpose is to obtain the most possible valid label path, which has been shown effective in Table 4. We have run the significant testing for all algorithms

chosen. The result is statistically significant for all our *adapted* approach as the p-value is smaller than the pre-defined significant level (5%) in all cases.

5. Discussion

In this section, we expand on the key findings of the evaluation results introduced earlier.

Crowd context matters We have deliberately chosen three representative tasks each presenting two datasets produced by volunteers and paid crowd respectively. Based on our results, there is a distinctive difference in performance for the same algorithm applied in these two different contexts. For all algorithms, the accuracy it can achieve under the volunteer context is evidently higher than the paid crowd, without any exception. For the same workflow, the overall accuracy (by path) it can achieve in volunteer context is normally around 30% higher than the paid crowd context for workflows with 2-3 questions. However, this does not seem to be the case when workflow involves more questions, such as in the galaxyzoo case where the best accuracy all the algorithms can achieve is only around 5% higher in volunteer context compared to paid crowd context.

Workflow counts From the representative tasks we have shown so far, there are two main factors that need to be taken into account when designing a classification crowdsourcing workflow especially when classification steps are interdependent: the number of questions (determining the depth of the graph) and how many answer options per each question (width of the corresponding node level, affecting cognitive efforts required for passing that node level with correct chosen options). In our evaluation, we found evidence that both depth and width impact on overall performance of the inference algorithms. One visible pattern is for the paid crowd datasets. In this setting, overall accuracy (by path) increases as the depth of the graph increases (for both *mv_adapted* and *mp_adapted*), which suggests that it might be a good idea to have more classification questions each with fewer options rather than having fewer questions and giving many options to choose from, particularly for the case where the crowd's skill level is uncertain. The other notable aspect is for volunteer context, the *mp* algorithm has a comparative performance with *mv* in Serengeti workflow, but not in the other two workflows with more levels.

Heuristics-based aggregation as an addition On observing the result in section 4.3, it seems to be a promising way if we consider combining output from these algorithms using a heuristic strategy to perform better inference. We want to use results from *mv_adapted*, *em_adapted* and *mp_adapted* in combination in order to exploit their strengths and weaknesses for complex classification tasks. To do so, we could have an aggregator which is based on following intuitions: 1. the number of **unique** classifications of an object (defined by u) shows the degree that the crowd workers agree/disagree on the classification where the higher number indicates higher degree of disagreement and normally imply the object is either a bit difficult or ambiguous to be classified. 2. the **ratio** (defined by r) between the

unique number of classifications/answers collected from the crowd and the total number of classifications/judgments also demonstrates how diverse the answers are for the corresponding object and hence similarly. 3. As three-sigma rule[Pukelsheim (1994)] in the empirical sciences suggests that almost all values should lie within three standard deviations of the mean in a normal distribution, and theoretically mean plus one, two or three standard deviation(s) covers 68%, 95% and 99.7% of the data. In the case where majority voting might potentially fail (where workers tend to disagree), the number of unique classification or the ratio of the number of unique to the total number of classification for an object falls within the higher range of the distribution. Thus, a heuristic aggregation strategy we could consider: Look at the intrinsic characteristics of collected classifications for each object, such as the number of **unique** classifications and the **ratio** of that against the total number of classifications. Then, based on the third intuition above, we can utilize the **skewness** (defined by s below) of the distribution for number of unique ($U \sim N(u_\mu, u_\sigma)$) and ratio (defined by $R \sim N(r_\mu, r_\sigma)$) respectively to heuristically chosen bound where MV can be potentially complemented by other approaches. However, choosing an optimal threshold is not straightforward and need to be explored in future work.

6. Related Work

Our approach is informed by existing work on microtask crowdsourcing and quality assurance in crowdsourcing, which we review in section.

6.1. *Microtask crowdsourcing and workflows*

In crowdsourcing, a problem needs to be sometimes decomposed into smaller, fine-granular microtasks and then arranged in a workflow for more effective processing. In general, a workflow consists of a set of microtasks; the microtasks are sometimes of different types and can be dependent or independent of each other. For instance, the find-fix-verify workflow proposed by [Bernstein *et al.* (2010)] uses microtask crowdsourcing to proof-read and shorten text in three steps: finding areas of improvement in the text; fixing or improving them; and verifying the quality of the changes. In each step, the crowd is asked to carry out the same type of microtask, sometimes iteratively. In [Kittur *et al.* (2008); Kittur *et al.* (2013); Acosta *et al.* (2013)], researchers have proposed to group the same or similar microtasks into batches as a means to facilitate learning effects. Previous studies have also shown that task performance can be improved as a function of several factors, including the design of tasks and workflows, motivation and incentives, and training [Bernstein *et al.* (2010); Demartini *et al.* (2012); Kittur *et al.* (2008); Wiggins *et al.* (2011)].

In the citizen science platform such as Zooniverse[i], most of the classifica-

[i] `https://www.zooniverse.org/`

tion projects are not simple tasks with one-question, instead is multiple-questions chained together. Zooniverse[i] uses workflow to "group a collection of tasks into a logic unit"[j] which is, in essence, referring to the relatively multiple-questions task which need to be finsihed in several steps. In Snapshot Serengeti,[a] classifying an image means answering a set of independent questions, sometimes several times when more than one animal is present in the image. In Cities at Night[b] and Galaxy Zoo[c], questions are inter-related and the answers given in one step determine the questions in the subsequent steps. In the context of such classification task, a workflow is used to refer to the logical organization of each classification questions and corresponding options.

Most previous studies around crowdsourcing workflows have focused on the design of the workflows and have shown that a particular type of workflow can be crowdsourced effectively (in terms of the accuracy of outputs, budget, time etc.) [Little *et al.* (2009); Bernstein *et al.* (2010); Tran-Thanh *et al.* (2015)]. In some cases, researchers have proposed bespoke quality assurance methods for their workflows [Lintott *et al.* (2011); Willett *et al.* (2013)]. Our work proposes a strategy which can be applied to determine the correct label path for a whole range of classification tasks, spanning over several steps with independent or dependent multiple-choice questions, which is different than existing research that mainly focus on the result for the final step (no matter how many other previous steps exist in its workflow).

6.2. *Inference algorithms*

Researchers have proposed *inference algorithms*, mathematical models that can automatically infer the correct solution to a given problem from a solution space defined by the crowd. For example, Ipeirotis et al. presented an algorithm that assesses the performance of crowd workers and exploits this information to estimate the quality of answers on Mechanical Turk [Ipeirotis *et al.* (2010)]. Karger et al. proposed to use message passing to infer correct answers from worker's answers [Karger *et al.* (2011)]. [Bachrach *et al.* (2012)] used a Bayesian graphical model to grade test answers in scenarios where the ground truth cannot be made available. [Whitehill *et al.* (2009)] followed an expectation maximization approach to identify correct classifications, depending on the expertise of the workers and the level of difficulty of the task. In the citizen science project Galaxy Zoo Supernovae, crowd answers were analysed using a Bayesian generalisation of the same expectation maximization idea [Simpson *et al.* (2011)]. More recently, [Difallah *et al.* (2015)] compiled a set of features that can be used to predict answer quality, based on an analysis of Mechanical Turk logs. Several studies have shown that it is possible to combine automatic prediction methods (such as Bayesian or generative probabilistic models) with additional input from the crowd to further improve the accuracy of the predictions [dos Reis *et al.* (2015); Hare *et al.* (2013); Ipeirotis *et al.* (2010); Loni *et al.* (2014); Simpson *et al.* (2013)].

[j]`https://blog.zooniverse.org/2013/06/20/how-the-zooniverse-works-the-domain-model/`

Other studies have analysed and compared different algorithms [Zheng *et al.* (2017); *et al.* (2015); Sheshadri and Lease (2013)], emphasizing the need for more research to understand the interplay among different sets of design parameters on the overall performance.

All these existing methods have considerably advanced the state of the art. However, they cannot be applied to every type of microtask crowdsourcing workflow without restrictions. Moreover, most of the research carried out so far in this space has looked at rather simple binary or multiple-choice classification tasks with the aim to identify a single, correct answer. This class of microtasks, albeit important and widely used, is not always the norm. As we have seen in the examples from the previous section, there are cases where a problem cannot be easily decomposed into independent microtasks, or where different, related microtasks should be grouped into more complex workflows for efficiency reasons. Although there are a few recent works looking into the relatively complex multiple-step classification tasks, each of them has a domain-specific or problem-specific focus [Parameswaran *et al.* (2011); Kim *et al.* (2002); Wu *et al.* (2012); Bragg *et al.* (2013); Kamar and Horvitz (2015); Otani *et al.* (2016)]. [Bragg *et al.* (2013)] and [Otani *et al.* (2016)] both research the entity classification that normally involve categorising the given entity into parent-child classes in different steps but have very different perspectives. [Bragg *et al.* (2013)] focus on improving the workflow for generating taxonomy, as well as inference methods to induce the parent-child relationship, while [Otani *et al.* (2016)] focus on the task where a parent-child relationship exists between two adjacent classification steps, and propose label aggregation methods that adapt from existing GLAD method ([Whitehill *et al.* (2009)]) by considering the hierarchical class-subclass structure. In addition, [Wu *et al.* (2012)] investigate the sequential data labelling scenario and present Sembler to ensemble crowd sequential labellings by leveraging the statistical correlation and dependency among multiple instances/sentences which is domain specific and not applicable to other multiple-step classification where no such statistics can be exploited. [Parameswaran *et al.* (2011)] and [Kamar and Horvitz (2015)] particularly look at the multiple-step image classification tasks while both took the approaches that are not easy to be generalised to suit for other multiple-step classification. [Parameswaran *et al.* (2011)] explicitly formulate the classification task as human-assisted graph search problem, presenting the dimensions characterising the different type of classification and developing algorithms to optimize the questions to be asked (at the different node) which is evaluated with simulation. On the other hand, [Kamar and Horvitz (2015)] focus on optimizing worker allocation in the hierarchical classification task (HCT) and develop answer models and evidence models for HCT consensus while both models are constructed with supervised learning, assisting with the Sloan Digital Sky Survey (SDSS) features identified by machine visions available for GalaxyZoo[c] dataset. There is also a few research particularly dedicating to automatic hierarchical classification where an taxonomy is given and a parent-child relationship among

classes exists, but all are bound to a certain domain. For instance, [Dumais (2000)] investigate automatic hierarchical classification using Support Vector Machine with existing web pages whose category are known as training data. [Su *et al.* (2006)] present an automatic method to classify structured web databases by leveraging probing queries, the returned count of query result and the SVM classifier. Such automatic hierarchical classification not only needs existing labelled data as training data but also focus on the classification where answers to further classification step down the line (child classes) are always a sufficient condition to confirm the answer to the previous classification step (parent classes).

Our approach differs from existing work mainly in the fact it is not restricted to a specific type of multiple-step classification and does not need additional information such as the machine identified features of the image or frequency/correlation among word usage, neither does it rely on the parent-child relationships between classification steps. Our method is general and intuitively easy to be applied in any multi-step classifications. We discussed the three main individual algorithms in section 2 and noted that whilst all three algorithms can be used to infer the correct answer for a multiple-choice question, they differ in terms of the inputs and outputs. In our approach, we devised a new strategy to utilize existing algorithms to achieve higher classification accuracy.

7. Conclusion

Ensuring quality is one of the grand challenges of microtask crowdsourcing. While previous research has looked at inferring correct answers for microtasks consisting of single binary or multiple-choice questions, our research proposes a model that can be applied to both single-question and multiple-question scenarios, filling the gap for understanding how to aggregate in the multiple-question scenarios. We propose a graph model and an "adapted" aggregation method that can improve the accuracy in inferring true label path in complex workflows with several interdependent questions. Though a few previous works tried to address similar multiple-step classification, they are either limiting it to the hierarchical classification scenarios where a parent-child relationship exists between classification steps or restricting the method by having to involve additional information. We propose using the graph to model a microtask crowdsourcing workflow and to support inference algorithms in making decisions about correct labels for classification tasks with multiple-questions, where the answer to one question does not have to be the sufficient condition to or imply the answer to the previous question is correct. We believe this is the first work that investigates aggregation in a multiple-step classification task with interdependent questions to infer the correct label path and assess the classification accuracy accordingly.

To this end, we explored three inference algorithms: majority voting, message passing, and expectation maximisation, each with proven benefits in quality assurance in crowdsourcing. We compared the performance of our *adapted* approach and

the existing *naive* approach, using six representative datasets. We evaluate the performance of individual algorithms for overall accuracy where a full labelling path is considered as an atomic, correct answer; and a more refined measure which looks at accuracy in individual node level of the workflow graph. The results have shown that our *adapted* approach has significantly improved the accuracy compared with the *naive* approach. The result also demonstrates that while majority voting does well in overall accuracy, a deeper analysis of the accuracy in each node level revealed a more interesting picture. Hence a heuristic-based aggregation approach might be a potentially better solution by combining results from multiple algorithms leveraging the strength of each other. This suggests the need for more dynamic inference approaches that can adapt to the complexity of the crowdsourcing workflow.

In future work, we plan to devise inference methods that take other, more workflow-specific factors into account. Our current method assumes independence between labels from different levels when inferring the answer for each level. It can be potentially improved to consider the possible correlation between labels in different node levels. For instance, it can consider giving different weight to labels based on the inferred result from the previous level. Such method requires a top-down traversal process which might bring side-effects since it counts heavily on the inferred result from the previous level, and carries on the effect (weight) to subsequent levels even the choice in the previous levels may be incorrect. As the correlation between labels in different node level is complicated, the feasibility of incorporating such correlation information into the aggregation process needs further investigation. Meanwhile, the number of options and the length of possible paths in a workflow deserves more in-depth experiments. One promising direction will be to employ other machine learning approaches for truth inference. For instance, using the workflow properties along with the crowdsourcing generated data to learn and explore features automatically [Huynh *et al.* (2013)], and produce decision tree to help choose the proper inference algorithm. Alternatively, certain properties from crowd-collected data could be further exploited to train machine learning algorithm(s) with selective labels in order to directly infer true label path.

References

D. Shahaf and E. Horvitz, "Generalized Task Markets for Human and Machine Computation.," in *AAAI*, 2010.

M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich, "Soylent: a word processor with a crowd inside," in *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pp. 313–322, ACM, 2010.

A. P. Kulkarni, M. Can, and B. Hartmann, "Turkomatic," *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems - CHI EA '11*, p. 2053, 2011.

A. Kittur, B. Smus, S. Khamkar, and R. E. Kraut, "CrowdForge: Crowdsourcing Complex Work," *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*, pp. 43–52, 2011.

A. Mao, E. Kamar, Y. Chen, E. Horvitz, M. E. Schwamb, C. J. Lintott, and A. M. Smith, "Volunteering Versus Work for Pay: Incentives and Tradeoffs in Crowdsourcing," *First AAAI Conference on Human Computation and Crowdsourcing*, pp. 94–102, 2013.

J. Redi and I. Povoa, "Crowdsourcing for Rating Image Aesthetic Appeal: Better a Paid of a Volunteer Crowd?," *Proceedings of the 2014 International ACM Workshop on Crowdsourcing for Multimedia - CrowdMM '14*, no. NOVEMBER 2014, pp. 25–30, 2014.

A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with mechanical turk," in *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 453–456, ACM, 2008.

R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast–but is it good? ee-valuating non-expert annotations for natural language tasks," in *Proceedings of the conference on empirical methods in natural language processing*, pp. 254–263, Association for Computational Linguistics, 2008.

D. Vickrey, A. Bronzan, W. Choi, A. Kumar, J. Turner-Maier, A. Wang, and D. Koller, "Online word games for semantic data collection," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 533–542, Association for Computational Linguistics, 2008.

G. Demartini, D. E. Difallah, and P. Cudré-Mauroux, "Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking," in *Proceedings of the 21st international conference on World Wide Web*, pp. 469–478, ACM, 2012.

A. Wiggins, G. Newman, R. D. Stevenson, and K. Crowston, "Mechanisms for data quality and validation in citizen science," in *e-Science Workshops (eScienceW), 20111 IEEE Seventh International Conference on*, pp. 14–19, IEEE, 2011.

C. Eickhoff and A. de Vries, "How crowdsourcable is your task," 2011.

D. E. Difallah, M. Catasta, G. Demartini, P. G. Ipeirotis, and P. Cudré-Mauroux, "The Dynamics of Micro-Task Crowdsourcing: The Case of Amazon MTurk," pp. 238–247, may 2015.

J. Yang, J. Redi, G. Demartini, and A. Bozzon, "Modeling Task Complexity in Crowdsourcing," 2016.

Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: Is the problem solved?," *Proceedings of the VLDB Endowment*, vol. 10, no. 5, 2017.

Y. Bachrach, T. Minka, and J. Guiver, "How To Grade a Test Without Knowing the Answers — A Bayesian Graphical Model for Adaptive Crowdsourcing and Aptitude Testing," 2012.

A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," *Applied statistics*, pp. 20–28, 1979.

J. S. Hare, M. Acosta, A. Weston, E. Simperl, S. Samangooei, D. Dupplaw, and P. H. Lewis, "An Investigation of Techniques that Aim to Improve the Quality of Labels provided by the Crowd," in *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop, Barcelona, Spain, October 18-19, 2013.*, vol. 1043 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2013.

P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on Amazon Mechanical Turk," *Proceedings of the ACM SIGKDD Workshop on Human Computation - HCOMP '10*, p. 64, 2010.

D. R. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *Advances in neural information processing systems*, pp. 1953–1961, 2011.

B. Loni, J. Hare, M. Georgescu, M. Riegler, X. Zhu, M. Morchid, R. Dufour, and M. Larson, "Getting by with a little help from the crowd: Practical approaches to social image labeling," *Proceedings of the 2014 International ACM Workshop on Crowdsourcing for*

Multimedia, pp. 69–74, 2014.

H. Paulheim and C. Bizer, "Improving the quality of linked data using statistical distributions," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 10, no. 2, pp. 63–86, 2014.

N. Quoc Viet Hung, N. T. Tam, L. N. Tran, and K. Aberer, "An evaluation of aggregation techniques in crowdsourcing," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8181 LNCS, no. PART 2, pp. 1–15, 2013.

S. L. Rosenthal and A. K. Dey, "Towards maximizing the accuracy of human-labeled sensor data," in *Proceedings of the 15th international conference on Intelligent user interfaces - IUI '10*, (New York, New York, USA), p. 259, ACM Press, feb 2010.

E. Simpson, S. Roberts, I. Psorakis, and A. Smith, "Dynamic bayesian combination of multiple imperfect classifiers," *Studies in Computational Intelligence*, vol. 474, pp. 1–35, 2013.

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," *Advances in Neural Information Processing Systems*, vol. 22, no. 1, pp. 1–9, 2009.

A. Sheshadri and M. Lease, "SQUARE: A Benchmark for Research on Computing Crowd Consensus," *First AAAI Conference on Human Computation and ...*, pp. 156–164, 2013.

J. Zhang, V. S. Sheng, Q. Li, J. Wu, and X. Wu, "Consensus algorithms for biased labeling in crowdsourcing," *Information Sciences*, vol. 382-383, pp. 254–273, 2017.

N. Q. V. Hung, D. C. Thang, M. Weidlich, and K. Aberer, "Minimizing efforts in validating crowd answers," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, vol. 2015-May, pp. 999–1014, 2015.

X. Liu, M. Lu, C. Ooi, Y. Shen, S. Wu, and M. Zhang, "CDAS : A Crowdsourcing Data Analytics System," *Vldb*, vol. 5, no. 10, pp. 1040–1051, 2012.

F. K. Khattak and A. Salleb-Aouissi, "Quality Control of Crowd Labeling through Expert Evaluation," *Second Workshop on Computational Social Science and the Wisdom of Crowds (NIPS 2011)*, pp. 1–5, 2011.

E. Kamar, S. Hacker, and E. Horvitz, "Combining human and machine intelligence in large-scale crowdsourcing," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pp. 467–474, International Foundation for Autonomous Agents and Multiagent Systems, 2012.

F. Pukelsheim, "The three sigma rule," *The American Statistician*, vol. 48, no. 2, pp. 88–91, 1994.

A. Kittur, J. V. Nickerson, M. Bernstein, E. Gerber, A. Shaw, J. Zimmerman, M. Lease, and J. Horton, "The future of crowd work," in *Proceedings of the 2013 conference on Computer supported cooperative work - CSCW '13*, (New York, New York, USA), p. 1301, ACM Press, feb 2013.

M. Acosta, A. Zaveri, E. Simperl, D. Kontokostas, S. Auer, and J. Lehmann, "Crowdsourcing linked data quality assessment," *The Semantic Web–ISWC 2013*, pp. 260–276, 2013.

G. Little, L. B. Chilton, M. Goldman, and R. C. Miller, "Turkit: tools for iterative tasks on mechanical turk," in *Proceedings of the ACM SIGKDD workshop on human computation*, pp. 29–30, ACM, 2009.

S. R. L Tran-Thanh, TD Huynh, A Rosenfeld, "Crowdsourcing Complex Workflows under Budget Constraints," *Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*, pp. 1298–1304, 2015.

C. Lintott, K. Schawinski, S. Bamford, A. Slosar, K. Land, D. Thomas, E. Edmondson,

K. Masters, R. C. Nichol, and M. J. Raddick, "Galaxy Zoo 1: data release of morphological classifications for nearly 900 000 galaxies," *Monthly Notices of the Royal Astronomical Society*, vol. 410, no. 1, pp. 166–178, 2011.

K. W. Willett, C. J. Lintott, S. P. Bamford, K. L. Masters, B. D. Simmons, K. R. V. Casteels, E. M. Edmondson, L. F. Fortson, S. Kaviraj, W. C. Keel, T. Melvin, R. C. Nichol, M. Jordan Raddick, K. Schawinski, R. J. Simpson, R. A. Skibba, A. M. Smith, and D. Thomas, "Galaxy zoo 2: Detailed morphological classifications for 304 122 galaxies from the sloan digital sky survey," *Monthly Notices of the Royal Astronomical Society*, vol. 435, no. 4, pp. 2835–2860, 2013.

E. Simpson, S. J. Roberts, A. Smith, and C. Lintott, "Bayesian combination of multiple, imperfect classifiers," 2011.

F. J. C. dos Reis, S. Lynn, H. R. Ali, D. Eccles, A. Hanby, E. Provenzano, C. Caldas, W. J. Howat, L.-A. McDuffus, and B. Liu, "Crowdsourcing the general public for large scale molecular pathology studies in cancer," *EBioMedicine*, vol. 2, no. 7, pp. 679–687, 2015.

J. Wang, P. G. Ipeirotis, and F. Provost, "Cost-Effective Quality Assurance in Crowd Labeling," 2015.

A. Parameswaran, A. D. Sarma, H. Garcia-Molina, N. Polyzotis, and J. Widom, "Human-Assisted Graph Search : It ' s Okay to Ask Questions," *Proceedings of the VLDB Endowment*, vol. 4, no. 5, pp. 267–278, 2011.

J.-H. Kim, I.-H. Kang, and K.-S. Choi, "Unsupervised named entity classification models and their ensembles," *Proceedings of the 19th international conference on Computational linguistics- Volume 1*, pp. 1–7, 2002.

X. Wu, W. Fan, and Y. Yu, "Sembler: Ensembling crowd sequential labeling for improved quality," *Proceedings of the National Conference on Artificial Intelligence*, vol. 2, pp. 1713–1719, 2012.

J. Bragg, D. S. Weld, *et al.*, "Crowdsourcing multi-label classification for taxonomy creation," in *First AAAI conference on human computation and crowdsourcing*, 2013.

E. Kamar and E. Horvitz, "Planning for Crowdsourcing Hierarchical Tasks," *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, p. 2030, 2015.

N. Otani, Y. Baba, and H. Kashima, "Quality control for crowdsourced hierarchical classification," *Proceedings - IEEE International Conference on Data Mining, ICDM*, vol. 2016-Janua, pp. 937–942, 2016.

S. Dumais, "Hierarchical Classification of Web Content," pp. 256–263, 2000.

W. Su, J. Wang, and F. Lochovsky, "Automatic Hierarchical Classification of Structured Deep Web Databases BT - Web Information Systems WISE 2006," (Berlin, Heidelberg), pp. 210–221, Springer Berlin Heidelberg, 2006.

T. D. Huynh, M. Ebden, M. Venanzi, S. Ramchurn, S. Roberts, and L. Moreau, "Interpretation of Crowdsourced Activities Using Provenance Network Analysis," *The First AAAI Conference on Human Computation and Crowdsourcing*, pp. 78–85, 2013.

Thomas W Malone, Robert Laubacher, and Chrysanthos Dellarocas, "The collective intelligence genome," *IEEE Engineering Management Review*, 38:38, 2010.

Ujwal Gadiraju, Ricardo Kawase, and Stefan Dietze., "A taxonomy of microtasks on the web," *Proceedings of the 25th ACM conference on Hypertext and social media*, pages 218–223. ACM, 2014.

Ujwal Gadiraju, Gianluca Demartini, Ricardo Kawase, and Stefan Dietze, "Human Beyond the Machine: Challenges and Opportunities of Microtask Crowdsourcing," *IEEE Intelligent Systems*, 30(4):81–85, 2015.

Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman,

David R Karger, David Crowell, and Katrina Panovich, "Soylent: a word processor with a crowd inside," *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 313–322. ACM, 2010.

Hadrien Gelas, Solomon Teferra Abate, Laurent Besacier, Laboratoire Dynamique, Du Langage, Cnrs Universit, De Lyon, Laboratoire Informatique De Grenoble, Cnrs Universit, and Fourier Grenoble., "Quality assessment of crowdsourcing transcriptions for African languages,"(August):3065–3068, 2011.

Panagiotis G Ipeirotis, Foster Provost, Victor S Sheng, and Jing Wang, "Repeated labeling using multiple noisy labelers,"*Data Mining and Knowledge Discovery*, 28(2):402–441, 2014.

Jacob Whitehill, Paul Ruvolo, Tingfan Wu, Jacob Bergsma, and Javier Movellan, "Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise," *Advances in Neural Information Processing Systems*, 22(1):1–9, 2009.

Panagiotis G. Ipeirotis, Foster Provost, and Jing Wang, "Quality management on Amazon Mechanical Turk," *Proceedings of the ACM SIGKDD Workshop on Human Computation - HCOMP '10*, page 64, 2010.

Yoram Bachrach, Tom Minka, and John Guiver, "How To Grade a Test Without Knowing the Answers — A Bayesian Graphical Model for Adaptive Crowdsourcing and Aptitude Testing," 2012.

Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, Panagiotis G. Ipeirotis, and Philippe Cudré-Mauroux, "The Dynamics of Micro-Task Crowdsourcing: The Case of Amazon MTurk," pages 238–247, may 2015.

Beverly K Kahn, Diane M Strong, and Richard Y Wang, "Information quality benchmarks: product and service performance,"*Communications of the ACM*, 45(4):184–192, 2002.

CarloMethodologies Batini, Cinzia Cappiello, Chiara Francalanci, and Andrea Maurino, "Methodologies for data quality assessment and improvement,"*ACM Computing Surveys*, 41(3):1–52, 2009.

Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, Sören Auer, and Pascal Hitzler, "Quality assessment methodologies for linked open data," *Submitted to Semantic Web Journal*, 2013.

JCGM. JCGM 200 : 2008 International vocabulary of metrology ??? Basic and general concepts and associated terms (VIM) Vocabulaire international de métrologie ??? Concepts fondamentaux et généraux et termes associés (VIM),"*International Organization for Standardization Geneva ISBN*, 3(Vim):104, 2008.