Transition from Observation To Knowledge To Intelligence (TOKI)

Editors

Dr. Victor ODUMUYIWA, Dr. Olufade ONIFADE,
Prof. Amos DAVID & Prof. Charles UWADIA

Victor ODUMUYIWA
Department of Computer Sciences,
University of Lagos
Nigeria

ISBN: 978-978-976-000-8

The responsibility for opinions expressed in articles, studies and other contributions in this proceeding rests solely with their authors.

Transition from Observation to Knowledge to Intelligence

3rd Biennial International Conference on Transition from Observation to Knowledge to Intelligence (TOKI)
15-16 August 2019
University of Lagos, Nigeria

Editors

Dr. Victor ODUMUYIWA
Dr. Olufade ONIFADE
Prof. Amos DAVID
Prof. Charles UWADIA

A Comparative Analysis of Four Label Extraction Algorithms for Crowdsourced Data

ADEOGUN Yetunde
Department of Computer Sciences
University of Lagos, Nigeria

ODUMUYIWA Victor
Department of Computer Sciences
University of Lagos, Nigeria

Abstract: The need for data has increased over the years with the advent of data science. Data scientists all over the world have required more and more data to be able to train or model various systems in machine learning and artificial intelligence. This need for data has made crowdsourcing a major requirement and has brought about websites like Amazon Turk, ClickWorker and others to the limelight to help in getting the required data. Crowdsourcing platforms are important because they bring together the requesters and the workers and provide data at a cheaper cost than what it takes to pay experts to get the same data. Overtime, it's been discovered that the data collected from crowdsourcing needs to be properly processed and analyzed to get value from it. This is because the data is provided by both experts and novice. This has made researchers delve more into various classification models, which are usually statistical in nature. Some of these approaches include majority voting, Dawid-Skene, bilayer clustering, etc. This research work provides a general understanding of crowdsourcing. It explains some of the classification approaches used to remove noise from crowdsourced data and then goes ahead to implement two of them (Majority voting and Multi-class ground truth inference with clustering). A comparison is done amongst these implementations and two other existing implemented approaches (Dawid-Skene and Fast-Dawid-Skene) to determine which behaves better on four datasets. The comparison done is based on the accuracy of the ground truth generated, the running time and the number of iterations before convergence.

Keywords: Crowdsourcing, classification model, data science

1. Introduction

Data has become a major commodity required all over the world to make decision in various spheres of life. In recent times, crowdsourcing has become one of the ways of acquiring data from various sources. Crowdsourcing has been defined by various people over time. One of the definitions that is most frequently cited in literature is that of Jeff Howe, who is considered as the person who coined the term crowdsourcing.

Howe described crowdsourcing as "the process of bringing in many people to achieve great feats from tasks that used to be handled by only a specialized few" (Howe, 2006). In 2009, Howe gave a more technical definition of crowdsourcing as "the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call" (Howe cited in Schenk & Guittard, 2011). Generally, crowdsourcing is about getting the best of ideas from various people in the world through the Internet and this aid in providing innovations for organizations, countries, e.t.c. This approach can take various forms which include bringing in people together to participate in public competitions (usually with a prize given to the best participant), undertake work collaboratively through peer production (Haythornthwaite, 2009) and in more recent times, using mobile devices and sensor networks to get a great amount of information (Burke et al., 2006).

Crowdsourced data can be used for things like sentiment analysis (determining if a statement is negative, positive or neutral), search relevance (ensures only relevant results are returned on the first search), data categorization, image labeling, document translation and many others. Getting this data from the crowd is the first major step in crowdsourcing while bringing out the true label from the answers received to the various questions is the next step, which requires the use of various algorithms.

In recent times, various applications have been developed to help organizations and individuals to acquire data from the crowd. Examples of these applications are: Clickworker, Amazon Mechanical Turk and CrowdFlower. These applications have made crowdsourcing easier by bringing together the relevant stakeholders, mainly the requesters and the workers. The requester is the person or institution requesting for help from the public while the workers are the people, individuals or teams who take up the tasks after they are advertised on the crowdsourcing platform. A crowdsourcing activity can only be declared successful if the requester is able to get the value required from the responses provided by the crowd and have enough participation while also considering various constraints like time, budget, and quality (Simperl, 2015). Most of the online crowdsourcing platforms do a level of assessment to determine which task should be assigned to which worker.

Responses to crowdsourced data can either be pre-determined by the requester (the requester provides the labels the workers choose an answer(s) from) or the requester allows the workers to type in the answers to the questions. If the requester chooses to allow the workers to type in the answers, they need a system to sieve through the various labels given and categorize them before analyzing the response to pick the ground truth. But in a case where the requester provides labels, focus will be on getting the ground truth from the responses rather than preprocessing the data to get all the labels provided by the workers and then try to get the ground truth (Muhammadi, Rabiee, & Hosseini, 2013 p.3).

This research work uses the approach where the labels are provided by the requester. The requester would usually depend on a few experts to come up with the predefined labels that the workers would choose answers from.

2. Related Works

A crowdsourcing project is not completed until the requester is able to harvest the information he/she wants from the crowd. Hence, validating the results collected from the crowd and assessing them is an important and compulsory aspect of crowdsourcing. Over the years, various methods have been used for classifying crowdsourced data.

The most common approach used to remove noise in crowdsourced data is Majority Voting. This was the main approach used when crowdsourcing first started in its various forms. It takes the label with the highest vote without any consideration for hidden things like if the workers make random guesses or mistakes or if they are spammers (who generate answers at random). It doesn't consider biases, expertise and difficulty of the classification task. This is the simplest approach used in getting ground truths from crowdsourced data. (Tao, Cheng, Yu, Yue, & Wang, 2019) improved on this approach by assigning higher scores to responses provided by experts in the field. They assessed the domain expertise of each worker in assigning task and value to their responses.

Other researchers have used Expectation Maximization in getting the ground truth. Some other authors considered the capability of each worker based on responses provided by the worker and models this using a confusion matrix (Dawid and Skene, 1979; Sinha, Rao and Balasubramanian, 2018). Whitehill, Wu, Bergsma, Movellan, and Ruvolo (2009) improved upon this by including a check for the bias of the worker and the difficulty of the task alongside the expertise of the worker. Some other work used clustering as a means of removing noise from crowdsourced data (Zhang, Sheng, & Wu, 2019; Zhang, Sheng, Wu, & Wu, 2016). A few other approaches allow people to choose more than one option from the list of labels provided (Be˜naran-Mu˜noz, Jer´onimo, & P´erez, 2018) while others require the workers to fill in the responses themselves (Demartini, Difallah, & Cudré-mauroux, 2012). All these approaches do comparisons that imply that they give better accuracy than other existing algorithms.

This research work however compares majority voting alongside two expectation maximization approach and one clustering approach.

3. Research Methodology

The methodology used for this research includes the review of various literature on crowdsourcing and existing classification approaches to remove noise. Gathering of existing dataset like the Adult2 data (Ipeirotis, Provost, & Wang cited in Zhang et al., 2016) and manipulations of some other datasets to reflect crowdsourced data. It involved the implementation of two of the algorithms (GTIC and majority voting) and then the comparison of the various algorithms (both the two implemented and two existing implemented ones) against the various dataset with regards to accuracy, time taken to run and the number of iterations before convergence. To carry out the task of removing noise from crowdsourced labels, this work focuses on four main algorithms: Majority voting, Dawid-Skene Approach, Fast Dawid-Skene Approach and Multi-Class Ground Truth Inference with clustering.

All four algorithms were implemented in python. The python version used is Python 3.7.4. The configuration of the workstation used for the implementation is an Intel® Core™ i5-8250U CPU with processor speed of 1.60GHz and RAM size of 12.0GB on a 64-bit windows operating system.

The datasets used in this research work are:

1) **Adult2:** This dataset was gotten from various workers on Amazon MTurk by (Ipeirotis, Provost, & Wang cited in Zhang et al., 2016). They requested that labelers should review the ratings of various websites under the category of G (General), PG (parental Guidance), R (Restricted for anyone under 17) and X (Adults only with explicit scenes).
2) **Valence5:** (Snow, O’Connor, Jurafsky, & Ng, 2010) selected a 100-headline sample from the SemEval-2007 Task 14 and collected labels for the dataset valence, where each example was labelled by 10 unique labelers. Using the valence data which is numeric, the valence value was divided into 5 classes (Strong, Negative, Neutral, Positive, Strong Positive)
3) **EmployeeReviewData:** This data shows how employees rated various organizations. The data was gathered from Kaggle website (“Employee Review Dataset,” 2018). An extract of 500

unique glassdoor links for google employee review having 10 participants per link was used for this process. The rating was numeric within the range of (0 and 5) and this was divided into 5 classes (BAD (0,1); OKAY (2); AVERAGE (3); GOOD(4); BEST(5)).

4) **NigerianHeadlineEmoticon:** This dataset was generated through the crowdsourcing process using Google forms. Micro tasks which consisted of 102 headlines from Nigeria newspapers online was divided into 20 questions each with two of them having 21 questions each. The workers were all in different locations and were meant to pick the first emotion they felt when they saw the headline. The emotions identified were - Joy/Gladness, Anger, Surprise, Disgust/Indignation, Sadness/Sorrow/Grief, Romance/Love, Fear/Scared, Kindness/Benevolence, Trust/Admire and Anticipation. The emotion, Indifference was deliberately left out because an initial pre-test showed that responders would be quick to pick that response for most questions and that would generate inaccurate data. Each set of questions was sent to different group of people via WhatsApp using the google form link.

The dataset breakdown is shown in table.

Table 1: Dataset used in the experiment

Name of Dataset	**Number of classes**	**Number of participants**	**Number of questions**	**Average number of participants per question**	**Number of collected data**
Adult2	4	269	309	9	3,260
Valence5	5	10	100	10	1000
EmployeeReviewdata	5	5000	500	10	5000
Nigerian Headline Emoticon	10	131	103	23	2713

4. Implementation Details

The implementation was done in python and is outlined below.

Algorithm: Majority Voting

Input: Crowdsourced choices of *Q* questions by *A* participants (annotators) from *C* choices

Output: Proposed true choices – T_{qc}

1) Estimate T_{qc} using highest frequency. That is, iterate through choices *c* picked by the annotators to pick the choice with the highest count. In an eventuality that two responses have equal count and are the highest, it picks the first choice as the choice for that question.

Algorithm: Dawid-Skene

Input: Crowdsourced choices of *Q* questions by *A* participants (annotators) from *C* choices

Assumption: The developer of the implementation specified a minimum number of participants for each question to make the implementation easier (Sinha et al., 2018). Each question must have been answered by this minimum number of participants.

Output: Proposed true choices – T_{qc}

1) Take initial estimates of the T_s
2) **Repeat**
3) *M-Step:* The system gets the count of responses per question and uses this in calculating the confusion matrix of each annotator *a* and other parameters.
4) *E-step:* Calculate new estimates of T_s using the confusion matrix and the parameters obtained above.
5) **Until Convergence**. Convergence is said to occur when the difference in the error rate is less than or equal to a predetermined answer or when the maximum number of iterations reaches a predetermined number.

Algorithm: Fast-Dawid-Skene

Input: Crowdsourced choices of *Q* questions by *A* participants (annotators) from *C* choices

Assumption: The developer of the implementation specified a minimum number of participants for each question to make the implementation easier (Sinha et al., 2018). Each question must have been answered by this minimum number of participants.

Output: Proposed true choices – T_{qc}

1) Estimate T_s using majority voting.
2) **Repeat**
3) *M-Step:* The system gets the count of responses per question and uses this in calculating the confusion matrix of each annotator *a* and other parameters.
4) *E-step:* Estimate T_s using the confusion matrix and the parameters obtained above.
5) *C-Step*: Assign T_s using the values obtained in the E-Step by getting the highest argument.
6) **Until Convergence**. Convergence is said to occur when the difference in the error rate is less than or equal to a predetermined answer or when the maximum number of iterations reaches a predetermined number.

Algorithm: Multi-class ground truth Inference with Clustering

Input: Crowdsourced choices of *Q* questions by *A* participants (annotators) from predetermined K choices (Classes)

Output: Proposed true choices for the questions – T_{qc}

1) For each *q* in *Q* question, generate the *K* and the *K +1* features of the classes.
2) Select a *K-centroid* set based on the features of the examples
3) Run K-Means clustering algorithm with Euclidean distance by setting the initial centroids as defined above (Chen, 2014; Zhang et al., 2016).
4) For each cluster returned from K-Means, create a new vector by adding all the *K* features in each cluster

5) For each cluster, based on the vector created, assign this cluster with the class that has highest value under the constraint that a cluster is mapped to one and only one class.
6) Assign each question *q* an inferred label according to the label of each cluster and return T_s.

5. Results

All four datasets above were run through the four different algorithms and comparisons was made based on the following:

1) Accuracy of the response returned.
2) Running time of the algorithm against the responses provided.
3) Number of iterations before convergence (applies to all others but not to majority voting).

5.1. Accuracy of the responses provided

The accuracy of the algorithm is achieved by comparing the returned responses with the gold data for the various datasets. Each algorithm was run against all four datasets and the result is displayed in Figure 1.

Figure 1 shows that Fast-Dawid-Skene and GTIC come very close in accuracy ratio except in the Adult2 dataset where Fast-Dawid-Skene has an accuracy of 75.08% compared to that of GTIC. But in EmployeeReviewData, GTIC is actually closer in accuracy to majority voting than Fast-Dawid-Skene. Dawid-Skene is observed to have the lowest accuracy ratio of all the datasets except for its high accuracy in adult2 dataset where its accuracy exceeds that of even majority voting algorithm by 3.3%. Though above 70%, GTIC has the least accuracy in the Adult2 data.

Generally, majority voting has the best accuracy index in the graph though this could be due to the way the gold data was chosen for the test. As explained earlier, majority voting does not consider important biases like the expertise of the workers, the way the task is distributed and the likes. This makes it not the best type of algorithm to use in real life scenario (DAWID & SKENE, 1979; Sinha et al., 2018).

5.2. Running time of the algorithm against the responses provided

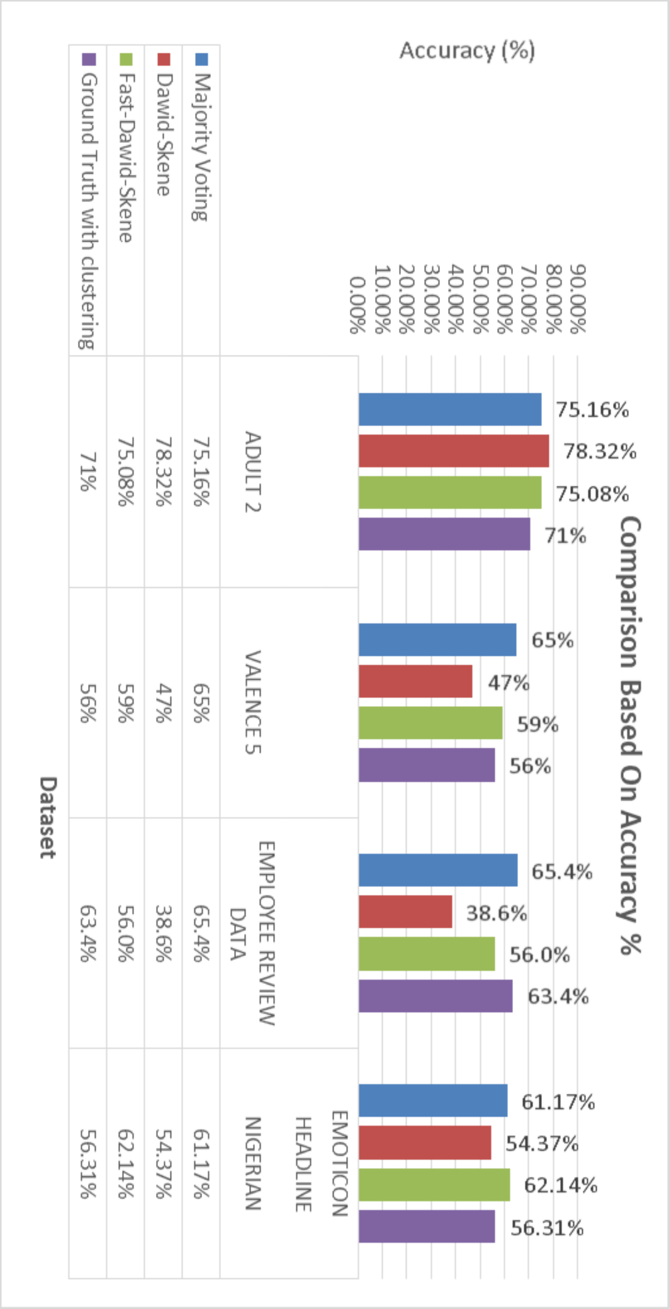
The running time is the total time it took each algorithm to run against the various dataset. Overall, Majority voting had the least processing time range as expected because it does not involve any kind of continuous iterations to get to an answer. The graph for this is shown in figure 2.

The running time parameter in figure 2 show that the Majority voting algorithm and fast-Dawid screen algorithm give the least running time amongst the four algorithms. If the parameter to determine which algorithm to use is running time, then majority voting and Fast-Dawid_Skene will be the best algorithms to use.

Running time for GTIC is close to that of Dawid-Skene. GTIC has the highest running time in 2 datasets (NigerianEmoticon and Valence5) while Dawid-Skene has the highest running time in the other 2 datasets (Adult2 and EmployeeReviewData).

5.3 Number of iterations before convergence

The number of iterations before convergence shows how many times the algorithm runs before either the error rate is achieved or the system has reached the highest number of iterations provided in the code.The writer of Fast_Dawid-Skene emphasized in his paper with example (Sinha et al., 2018), that the Fast-Dawid-Skene would always be faster than actual Dawid Skene because of the approach of first using majority voting to get the initial answer and then using the confusion matrix for each participant for each question and picking the maximum value of it to get the correct answer. When compared with the other algorithms, Fast-Dawid Skene had the least number of iterations before convergence except in the Valence5 dataset where GTIC has the least number of iterations as shown in figure 3.



	ADULT 2	VALENCE 5	EMPLOYEE REVIEW DATA	NIGERIAN HEADLINE EMOTICON
🟦 Majority Voting	75.16%	65%	65.4%	61.17%
🟥 Dawid-Skene	78.32%	47%	38.6%	54.37%
🟩 Fast-Dawid-Skene	75.08%	59%	56.0%	62.14%
🟪 Ground Truth with clustering	71%	56%	63.4%	56.31%

Figure 1: Accuracy level (%) achieved by each algorithm when run against the 4 dataset

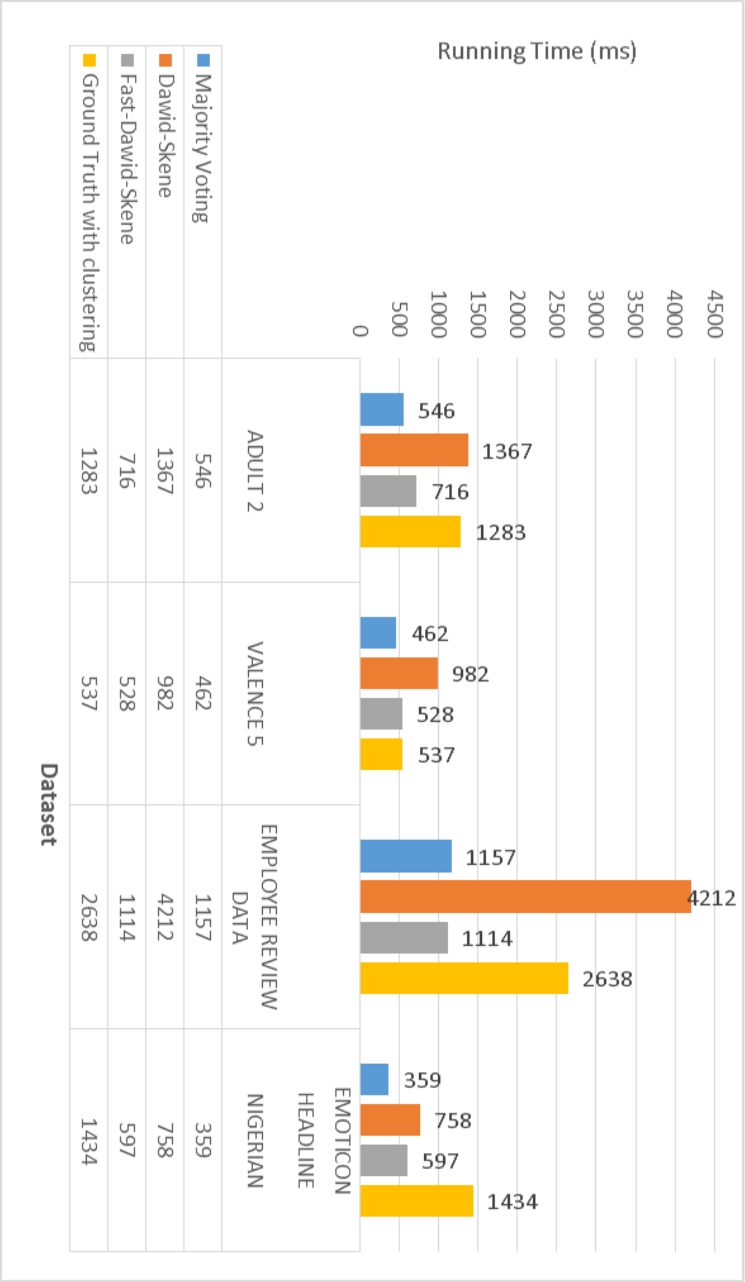


Figure 2: Running time (milliseconds) achieved by each algorithm when run against the 4 dataset

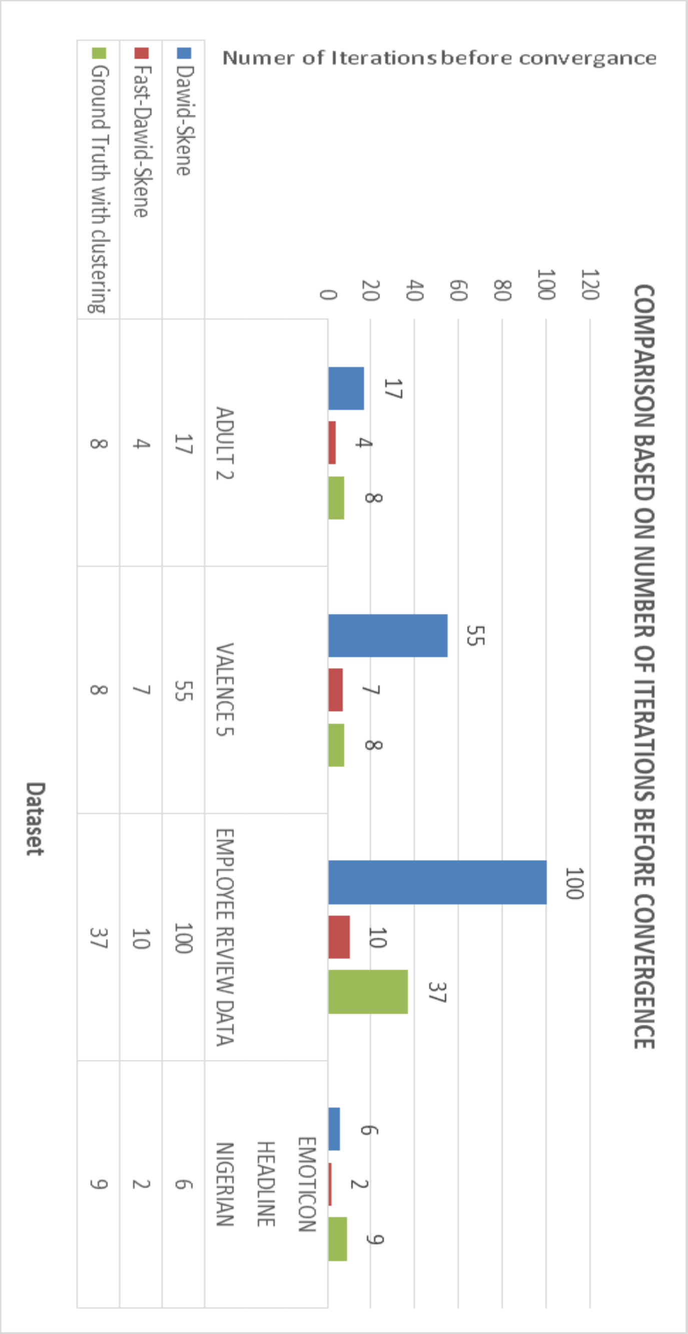


Figure 3: Number of iterations before convergence achieved by each algorithm when run against the 4 dataset

6. Conclusion

Our experiments show that Majority voting behaved better with regards to time and accuracy for virtually all the datasets. But the accuracy level was low showing the impact of biases and other factors. Fast-Dawid-Skene performed best amongst the three statistical approach having the least number of iterations before convergence and also a higher accuracy level than the other two. More work is however on-going to ensure a better accuracy level than majority voting can offer when things like the user biases, age and other factors are considered (Hope, 2018; Servajean, Joly, Shasha, Champ, & Pacitti, 2017; Tao et al., 2019).

List of References

Be˜naran-Mu˜noz, I., Jer´onimo, H.-G. alez, & P´erez, A. (2018). Weak labeling for crowd learning. *ArXiv:1804.10023v1 [Stat.ML]*, 1–6.

Chen, K. (2014). Cloud Computing Labs Code. Retrieved April 15, 2019, from https://github.com/NRNB-GSoC2017-SBML2SBGNML-Converters/SBML2SBGNML/blob/master/code/src/org/sbfc/converter/sbgnml2sbml/SBMLPointsKMeans.java

DAWID, A. P., & SKENE, A. M. (1979). Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society Series C Applied Statistics*, *28*(1), 20–28. https://doi.org/10.2307/2346806

Demartini, G., Difallah, D. E., & Cudré-mauroux, P. (2012). ZenCrowd : Leveraging Probabilistic Reasoning and Crowdsourcing Techniques for Large-Scale Entity Linking. *WWW'12 - Proceedings of the 21st Annual Conference on World Wide Web*, 469–478. https://doi.org/10.1145/2187836.2187900

Employee Review Dataset. (2018). Retrieved December 13, 2018, from https://www.kaggle.com/mitchelfruin/tech-company-employee-reviews/report

Hope, T. (2018). Ballpark Crowdsourcing : The Wisdom of Rough Group Comparisons. *WSDM '18 Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 234–242. https://doi.org/https://doi.org/10.1145/3159652.3159670

Ipeirotis, P. G., Provost, F., & Wang, J. (2010). Quality Management on Amazon Mechanical Turk. *KDD-HCOMP'10, July 25, 2010, Washington DC, USA*, 0–3.

Servajean, M., Joly, A., Shasha, D., Champ, J., & Pacitti, E. (2017). Crowdsourcing Thousands of Specialized Labels: A Bayesian Active Training Approach. *IEEE Transactions on Multimedia*, *19*(6), 1376–1391. https://doi.org/10.1109/TMM.2017.2653763

Sinha, V. B., Rao, S., & Balasubramanian, V. N. (2018). Fast Dawid-Skene: A Fast Vote Aggregation Scheme for Sentiment Classification. Retrieved from http://arxiv.org/abs/1803.02781

Snow, R., O'Connor, B., Jurafsky, D., & Ng, A. Y. (2010). Cheap and fast---but is it good?, 254. https://doi.org/10.3115/1613715.1613751

Tao, D., Cheng, J., Yu, Z., Yue, K., & Wang, L. (2019). Domain-Weighted Majority Voting for Crowdsourcing. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(1), 163–174. https://doi.org/10.1109/TNNLS.2018.2836969

Whitehill, J., Wu, T., Bergsma, J., Movellan, J. R., & Ruvolo, P. L. (2009). Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, & A. Culotta (Eds.), *Advances in Neural Information Processing Systems 22* (pp. 2035–2043). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/3644-whose-vote-should-count-more-optimal-integration-of-labels-from-labelers-of-unknown-expertise.pdf

Zhang, J., Sheng, V. S., & Wu, J. (2019). Crowdsourced Label Aggregation Using Bilayer Collaborative Clustering. *IEEE Transactions on Neural Networks and Learning Systems*, *PP*, 1–14. https://doi.org/10.1109/TNNLS.2018.2890148

Zhang, J., Sheng, V. S., Wu, J., & Wu, X. (2016). Multi-Class Ground Truth Inference in Crowdsourcing with Clustering. *IEEE Transactions on Knowledge and Data Engineering*, *28*(4), 1080–1085. https://doi.org/10.1109/TKDE.2015.2504974