CROWNN: HUMAN-IN-THE-LOOP NETWORK WITH CROWD-GENERATED INPUTS

Yusuke Sakata[⋆]*, Yukino Baba*[†§]*, Hisashi Kashima*[⋆§]

Kyoto University[⋆], University of Tsukuba[†], RIKEN Center for AIP[§]

ABSTRACT

Input features are indispensable for almost all machine learning methods; however, their definitions themselves are sometimes too abstract to extract automatically. Human-in-the-loop machine learning is a promising solution to such cases where humans extract the feature values for machine learning models. We use crowdsourcing for feature value extraction and consider a problem to aggregate the feature values to improve machine learning classifiers. We propose a novel neural network model called CROWNN, a neural network with crowd-generated inputs with the *worker convolution layer*, that learns both the capabilities of human feature extractors and the weights of a neural network classifier by applying the idea of the convolution neural network to feature aggregation. Our experiments using four datasets show the proposed method outperforms the baseline method using unsupervised aggregation methods in some datasets. We also show the robustness of the proposed model against the existence of spam workers, especially when they are malicious workers who intentionally flip the feature values.

***Index Terms*—** Crowdsourcing, human computation, human-in-the-loop

1. INTRODUCTION

The recent significant advances of machine learning have created many promising applications in various fields in science and industry. When we apply machine learning to a particular problem, the first thing to do is to represent the data as machine-readable feature values. However, the definitions of the features are sometimes so abstract that automatic extraction is difficult for machines. On the other hand, it is sometimes relatively easier for humans to extract them. In such case, the idea of *human-in-the-loop* machine learning is a promising option where humans extracts the feature values and machines learn using them. Traditionally, the time and financial cost of recruiting human labors had been the major bottleneck to execute such human-in-the-loop process; however, the recent rise of crowdsourcing platforms has made it easier to include human labors into a machine-learning process because they allow us to access a large amount of human labors at cheap costs in an on-demand manner [1, 2]. Examples of problems in which human-machine collaboration is effective include identification of painters of paintings, detecting fake laughs in a video; they are relatively difficult problems for both machines and humans without expertise. On the other hand, it is rather easier for humans to give useful abstract features, for example, whether or not a painting has a positive atmosphere, or whether or not a person laughs shaking by sobs. These features are expected to contribute to the prediction targets; in such cases, the human-in-the-loop process has an advantage over machine-only processes.

One of the major concerns when using crowdsourcing in the human-in-the-loop processes is the variable quality crowdsourcing results. The ability and motivation of crowdsourcing workers are not even, which can result in uneven quality of crowdsourcing results [3]. A common solution to this problem is introduce redundancy; we ask a same task to multiple workers and aggregate their answers to obtain a more reliable result. The simplest way is to take majority voting; however, it does not consider the variety of worker ability. In addition, sometimes we have uncooperative workers such as spam workers and malicious workers. Spam workers try to earn easy money, for example, by answering a constant answer or random answers to all questions, while malicious workers try to deceive the requester by answering wrong answers. Existence of such workers degrades the quality of the aggregated answers; therefore, more sophisticated statistical methods considering worker ability [4] and task difficulty [5] have been studied. Most of such label aggregation methods are unsupervised, i.e. the ground truth labels are not given. On the other hand in our situation, although the ground truth labels for features are not given, the correct class labels are available. The existence of such ground truth class labels are expected to indirectly contribute the quality control of the feature labels, and therefore we expect simultaneous estimation of worker ability and a classification model performs better than separate estimation of them.

In this paper, we propose CROWNN, a neural network classifier based on the feature labels extracted with the help of crowd workers. CROWNN has a special layer called the *worker convolution* layer to take the ability of the workers into account. CROWNN learns the worker ability (i.e., the worker convolution filters) and the classifier simultaneously. Our experiments investigate the applicability and effectiveness of the proposed method using four tasks: identification of painters, detection of spontaneous smiles, finding fake hotel

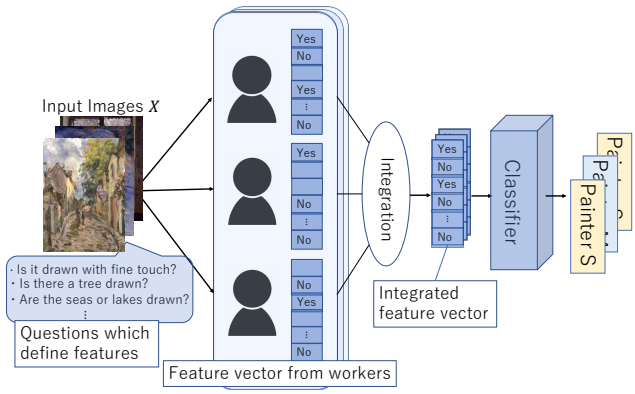


Fig. 1: An example of a painter recognition task.

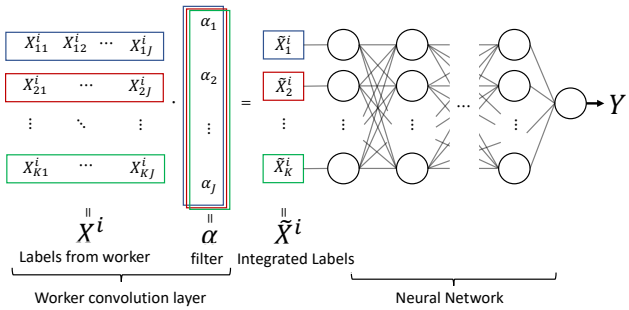


Fig. 2: The CROWNN architecture.

reviews, and estimating news publicity. Our results show the proposed method outperforms the baseline method using an existing unsupervised aggregation method in some datasets. We also show the robustness of the proposed model against the existence of spam workers, especially when they are malicious workers who intentionally flip the feature values.

2. RELATED WORK

Human-in-the-loop machine learning builds in human intelligence for helping machines to solve AI-hard problems. Crowdsourcing is often employed as its platform because of the ease of recruiting human labor at low costs in an on-demand manner. There are at least least two viewpoints when we consider human-in-the-loop machine learning, i.e. which part of the analysis pipeline we include human intelligence, and the quality control problem of human-integrated systems.

The most popular use of crowdsourcing in machine learning processes is data collection. When we apply (semi-)supervised learning, we always require a certain amount of correct output labels as the training dataset, and crowdsourcing is a promising way to collect them on a large scale [6]. When the target task requires expert knowledge, it is sometimes difficult even for ordinary people to directly give the ground truth outputs. An alternative approach is to ask them questions which are relatively easier but related to the target output. For example, it is sometimes hard for humans to directly tell bird species from images, but instead they can easily tell the color of their bellies or the shape of the beaks [7]. Instead of the original target labels, we use the collected answers as input features of a machine learning classifier. We study this type of approach in this paper.

Design of input features have a large impact on the resultant classifier and is a nontrivial problem. Some of the existing work attempt to create the feature definitions using crowdsourcing [8, 9]. Although crowd-feature generation is indeed an important task, we assume the feature definitions are available and focus only on feature value extraction.

Crowdsourcing allows us easy accesses to a large amount of human labors in an on-line and on-demand manner; however, we often suffer from the variation in the crowdsourcing results due to the variations in the ability or diligence of crowd workers and the task difficulty. One of the typical solutions is to ask a same task to multiple workers and aggregate their answers to obtain a reliable answer [10, 6]. Majority voting [11] is an easiest way, but more sophisticated statistical models have been proposed. One of the well-accepted models is the Dawid-Skene model [4] that has the worker ability parameters. GLAD is another model that also considers the task difficulty [5]. Welinder et al. [12] also considered the affinity between workers and tasks in their model. Raykar et al. [13] proposed an approach that directly estimates the prediction model as well as the class labels. Some work addresses spam worker detection by focusing on the randomness and bias in their answers [14]. Inspired by the recent rise of deep learning, some approaches extended the above idea to neural network models [15, 16]. We also employ a neural network in this work; the main difference from the previous work is that we focus on crowd-generated *feature* values, while the previous work focuses on crowd-generated outputs. They require different model architectures.

3. PROBLEM SETTING

Our goal is to obtain a binary classifier $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is the input domain and $\mathcal{Y} = \{-1, +1\}$ is the output domain, given the training dataset $\{(x^{(i)}, y^{(i)}) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^{N}$. The obtained classifier is expected to correctly predict the output labels for the test dataset $\{x^{(i)} \in \mathcal{X}\}_{i=N+1}^{N+M}$.

In the standard setting, the input domain is readily associated with a D-dimensional vector space each of whose dimensions corresponds to a feature characterizing the input data domain. However, in our setting, the definitions of the features are rather abstract and hard to extract automatically, and therefore we resort to use crowdsourcing to extract the feature values for each data instance. For example, if we want to recognize the artist who drew a painting, possible questions to the crowdworkers include "does the sky occupy a large

area?" or "are the objects vaguely outlined?" We assume the features are given in the form of binary questions (e.g., 'yes' or 'no'). Figure 1 shows an example of the pipeline of the painter recognition task using crowd-extracted features.

We denote by $x_{kj}^{(i)}$ the feature value that the j-th crowdworker gave for the k-th feature of the i-th data instance. Note that each worker does not necessarily give feature values for all of the training and test instances.

4. PROPOSED METHOD: CROWNN

We propose CROWNN, a neural network model based on crowd-extracted feature values (Fig. 2). The key idea is to use the notion of convolution neural network for simultaneous learning of worker abilities and a classification model. We introduce *worker convolution layer* parameterized by $\alpha = (\alpha_1, \ldots, \alpha_J)$ where α_j is the ability (or weight) of the j-th crowdworker. The worker convolution layer obtains the feature values $\left(x_{k1}^{(i)}, \cdots, x_{kJ}^{(i)}\right)$ from J workers for instance i and feature k, and outputs an aggregated feature value $\tilde{x}_k^i$ as

$$\tilde{x}_k^{(i)} = \sum_{j=1}^{J} x_{kj}^{(i)} \alpha_j. \tag{1}$$

The aggregated feature values $\tilde{x}^{(i)} = \left(\tilde{x}_1^{(i)}, \cdots, \tilde{x}_K^{(i)}\right)$ are used as a feature vector of instance i. $\tilde{x}^{(i)}$ is input to a neural network $f_\theta(x)$ with parameters θ. Given the crowd-generated feature values $x_{kj}^{(i)}$ for each $i \in \{1, \cdots, N\}$, $j \in \{1, \cdots, J\}$ and $k \in \{1, \cdots, K\}$, and the training labels $\{y^{(i)}\}_{i=1}^N$, CROWNN performs backpropagation to learn the weights of the worker convolution layer α, and the neural network parameters θ. When CROWNN assigns a high weight to a worker, the feature values given by the worker are emphasized in creating the aggregated feature values. For workers who (intentionally or carelessly) provide wrong feature values, CrowNN assigns negative weights so that their answers are flipped when input into the classifier.

The main difference between our method and the existing crowd aggregation methods is that CROWNN learns the worker abilities and the model parameters simultaneously, while the existing methods estimate them separately. When applying the existing methods to the crowd-extracted features, one would first aggregate the worker answers in an unsupervised manner and then learn a classifier in a supervised manner; this approach does not count how much the feature values given by each worker contribute to achieve an accurate classification. Additionally, the existing methods require multiple workers answer to a same extraction task for aggregating the answers; in contrast, CROWNN can work even when only a single worker answers to an extraction task.

The worker answers in the feature extraction tasks are given in 'yes' or 'no'. We set $x_{kj}^{(i)} = 1$ if the answer is 'yes' and $x_{kj}^{(i)} = 0$ if it is 'no'. In our problem setting, crowdworkers are not assumed to extract all the features for all the instances; that is, some $x_{kj}^{(i)}$ can be missing. We set $x_{kj}^{(i)} = 0$ for missing cases. Both 'no' and a missing case are represented by zero. We take this approach because we consider the answer of 'yes' is more important than 'no' in the feature extraction tasks.

5. EXPERIMENTS

We conducted experiments to investigate the effectiveness of the proposed method. We additionally designed experiments to examine the robustness of the proposed method against spam crowdworkers who randomly chose an answer, and malicious workers who intentionally returned a wrong answer.

We designed four binary classification tasks [9]: *Paintings* task identifies which of Claude Monet or Alfred Sisley is the author of a given painting. *Smiles* task distinguishes a spontaneous smile and a posed smiles of a person in a video. *Reviews* task tells whether a given hotel review is fake or not. *Articles* task tells if a given news should be highlighted on the top page of a news media. Each dataset contains 200 positive instances and 200 negative instances. We defined 100 abstract features for each dataset. Ten workers were recruited on LANCERS crowdsourcing platform to give feature values for each instance. We sorted the workers according the number of their submitted results, and used the feature values given by the first three workers for each feature and each instance. We generated synthetic spam workers whose number is chosen from $m \in \{1, 2, 3, 6\}$, and their answers were randomly chosen from 'yes' and 'no'. We also simulated malicious workers whose answers were flipped from their original answers.

We compare the proposed method (CROWNN) with the following two baseline methods for aggregating worker answers: MEAN simply calculates the average of feature values given by workers. The DAWID&SKENE (D&S) [4] model is a standard statistical model for aggregating worker answers, which estimates the reliability of each worker as well as the true answers. We only use worker answers in training data for estimating worker reliability that is then utilized for estimating the true answers for test data. The aggregated answers are used as the input of a neural network which has the same structure as CROWNN except the worker convolution layer.

Each method is evaluated with 10-fold cross validation. The average accuracy is used as the evaluation metric. Hyper parameters are tuned by using 10-fold cross validation as well. The candidate parameters are given as follow: the number of units within each hidden layer: $\{100, 150, \cdots, 300\}$; the number of layers: $\{2, 3, \cdots, 7\}$; the activation function of each hidden layer: $\{\text{reul}, \text{leaky\_relu}\}$; the number of epochs: $\{20, 30, \cdots, 100\}$; the mini batch size: $\{20, 40, \cdots, 200\}$.

Figure 3 shows classification accuracies of each method with the four datasets with three worker per feature and instance. CROWNN outperforms the baselines in the articles

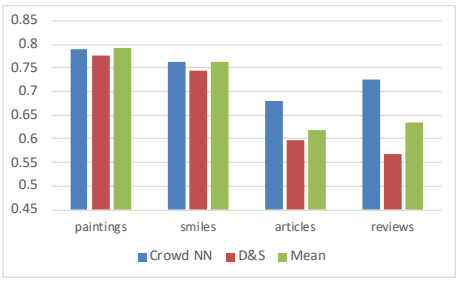


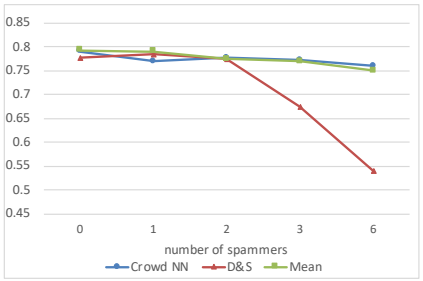
Fig. 3: Comparison of classification accuracy.

and reviews datasets, and shows comparable performance to the mean method in the smiles dataset, but is slightly inferior in the paintings dataset. This can be partially explained by the characteristics of the datasets. The answers by crowd workers have higher agreement ratios in the smiles and painting datasets than in the articles and reviews datasets. This means that feature extraction is easier in the former datasets than the latter ones; and therefore even the simple aggregation method (i.e., the mean method) can produce accurate aggregation results in the first two datasets. On the other hand, our end-to-end method demonstrates the superior performance in the other relatively difficult datasets.

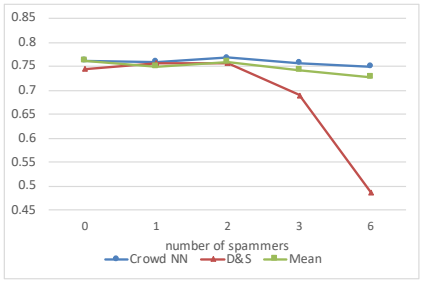
Figure 4 evaluates robustness against random spam workers. The accuracy of the baselines method decline along with the number of the spam workers, while CROWNN shows the higher robustness. Because the D&S model usually uses the majority answers as the initial estimates of worker abilities, it fails to estimate accurate abilities. The worker weights in CROWNN depends on the contribution to the classification; because random answers from spam workers make no contributions, CROWNN is able to leave out their answers. It is worth noticing that CROWNN shows consistent performance even when $2/3$ of the whole workers are spam workers. Figure 5 evaluates robustness against malicious workers dominating 30% of the whole workers. The D&S model and the mean approach suffer from the existence of the malicious workers; in contrast, CROWNN successfully assigns negative weights to malicious workers and leverages their answers.

6. CONCLUSION

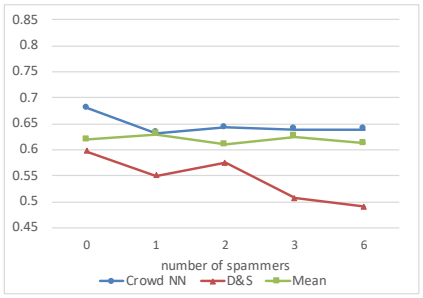
We proposed CROWNN, a human-in-the-loop neural network using crowd-generated feature values which can consider the worker ability using the worker convolution layer. The experiments showed its advantage over the unsupervised aggregation methods as well as the robustness against spam workers. One of the limitations of the proposed model is that we assume the same set of workers participate in both training and test phases, which sometimes not quite a realistic assumption. A possible scenario is on-line learning, where training and test phases are not separated. Extension to such on-line scenarios is an interesting future work.



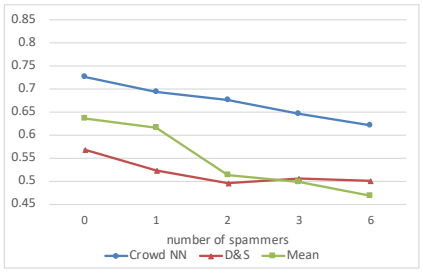
(a) Paintings



(b) Smiles



(c) Articles



(d) Reviews

Fig. 4: Robustness against spam workers.

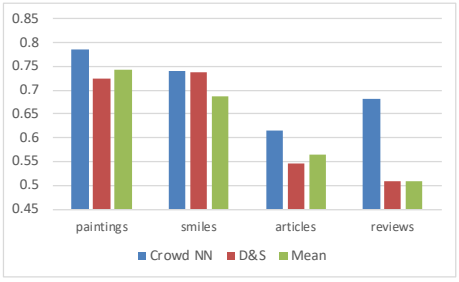


Fig. 5: Robustness against malicious workers.

7. REFERENCES

[1] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng, "Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2008, pp. 254–263.

[2] Michael D. Buhrmester, Tracy Kwang, and Samuel D. Gosling, "Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data?," *Perspectives on Psychological Science*, vol. 6, no. 1, pp. 3–5, 2011.

[3] Pinar Donmez and Jaime G. Carbonell, "Proactive learning: Cost-sensitive active learning with multiple imperfect oracles," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM)*. 2008, pp. 619–628, ACM.

[4] Alexander P. Dawid and Allan M. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," *Applied Statistics*, , no. 1, pp. 20–28, 1979.

[5] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *Advances in Neural Information Processing Systems 22*, pp. 2035–2043. 2009.

[6] Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis, "Get another label? improving data quality and data mining using multiple, noisy labelers," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2008, pp. 614–622.

[7] Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge Belongie, "Visual recognition with humans in the loop," in *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, 2010, pp. 438–451.

[8] Justin Cheng and Michael S. Bernstein, "Flock: Hybrid crowd-machine learning classifiers," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW)*, 2015, pp. 600–611.

[9] Ryusuke Takahama, Yukino Baba, Nobuyuki Shimizu, and Hisashi Kashima Sumio Fujita, "Adaflock: Adaptive feature discovery for human-in-the-loop predictive modeling," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[10] Panagiotis G. Ipeirotis, "Analyzing the amazon mechanical turk marketplace," *XRDS*, vol. 17, no. 2, pp. 16–21.

[11] Lionel S. Penrose, "The elementary statistics of majority voting," *Journal of the Royal Statistical Society*, vol. 109, no. 1, pp. 53–57, 1946.

[12] Peter Welinder, Steve Branson, Pietro Perona, and Serge J. Belongie, "The multidimensional wisdom of crowds," in *Advances in Neural Information Processing Systems 23*, pp. 2424–2432. 2010.

[13] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy, "Learning from crowds," *Journal of Machine Learning Research*, vol. 11, pp. 1297–1322, 2010.

[14] Vikas C. Raykar and Shipeng Yu, "Ranking annotators for crowdsourced labeling tasks," in *Advances in Neural Information Processing Systems 24*, pp. 1809–1817. 2011.

[15] Jingjing Li, Victor S. Sheng, Zhenyu Shu, Yanxia Cheng, Yuqin Jin, and Yuan feng Yan, "Learning from the crowd with neural network," in *Proceedings of the 14th International Conference on Machine Learning and Applications (ICMLA)*, 2015, pp. 693–698.

[16] Filipe Rodrigues and Francisco C. Pereira, "Deep learning from crowds," *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, 2018.