Sentiment Analysis via Deep Hybrid Textual-Crowd Learning Model

Kamran Ghasedi Dizaji, Heng Huang*
Electrical and Computer Engineering, University of Pittsburgh, USA
kag221@pitt.edu, heng.huang@pitt.edu

Abstract

Crowdsourcing technique provides an efficient platform to employ human skills in sentiment analysis, which is a difficult task for automatic language models due to the large variations in context, writing style, view point and so on. However, the standard crowdsourcing aggregation models are incompetent when the number of crowd labels per worker is not sufficient to train parameters, or when it is not feasible to collect labels for each sample in a large dataset. In this paper, we propose a novel hybrid model to exploit both crowd and text data for sentiment analysis, consisting of a generative crowdsourcing aggregation model and a deep sentimental autoencoder. Combination of these two sub-models is obtained based on a probabilistic framework rather than a heuristic way. We introduce a unified objective function to incorporate the objectives of both sub-models, and derive an efficient optimization algorithm to jointly solve the corresponding problem. Experimental results indicate that our model achieves superior results in comparison with the state-of-the-art models, especially when the crowd labels are scarce.

Introduction

Recently rapidly growing use of social media has provided a huge source of public opinions about different topics. Efficient mining of these opinions is very valuable for various industries and businesses. For instance, hotels, airlines, lenders, banks and even politicians utilize these data to find new costumers, target new products, analyze the personality of clients and make better decisions. However, exploring the sentiment of public opinions is a very challenging task for automatic language models due to different variations in the texts, such as diverse contexts, genders of authors, writing styles and varied viewpoints.

Crowdsourcing platforms like Amazon Mechanical Turk[1] provide an efficient tool to solve this type of the problems by using the knowledge of crowd workers in different tasks at low cost and time. Hence, the human skills in language understanding can be used to interpret the sentiments of texts with different variations. However, the collected labels via crowdsourcing are often noisy and inaccurate, because crowd workers are usually inexpert in the assigned task. In order to address this issue, it is common to collect multiple crowd labels for each sample to increase the credibility of the estimated true labels. Several studies have proposed different models to aggregate the crowd labels and estimate the potential true labels, which are also called truths (Dawid and Skene 1979; Chen, Lin, and Zhou 2013; Whitehill et al. 2009; Zhou et al. 2014; Ghasedi Dizaji, Yang, and Huang 2017). *But these crowdsourcing aggregation models become drastically incompetent, when the number of crowd labels per worker is not enough to train the reliability parameters of workers, or a document dataset is extremely large that collecting crowd labels for all samples is not practically feasible. In addition, crowdsourcing aggregation models do not utilize text data, and only use crowd labels as the source of information (i.e. input data).*

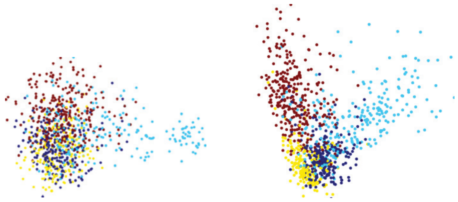

(a) MV-DeepAE (b) CrowdDeepAE

Figure 1: 2D visualization of *CrowdDeepAE* (ours) and *MV-DeepAE* features on *CrowdFlower* dataset using PCA, when only 20% of the crowd data is available.

In this paper, we propose a new hybrid model for sentiment analysis, which utilizes both crowd labels and text data. In particular, our proposed model, called *CrowdDeepAE*, consists of a generative aggregation model for crowd labels and a deep autoencoder for text data. These two sub-models are coupled in a probabilistic framework rather than a heuristic approach. Using this probabilistic framework, we introduce a unified objective function that incorporates the interests of both sub-models. We further derive an efficient optimization algorithm to solve the corresponding problem via an alternating approach, in which the parameters are updated while the truths are assumed to be known, and the truths are estimated when the parameters are fixed.

Therefore, CrowdDeepAE exploits the intelligence of crowd workers and the underlying informations of text data

*To whom all correspondence should be addressed.

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

to categorize sentiments more accurately. To do so, it employs a non-linear generative aggregation model to flexibly aggregate noisy crowd labels, and leverages a deep denoising autoencoder to learn a discriminative embedding for text data. In particular, the deep autoencoder uses text data to find similar patterns between text samples and prevent the crowd aggregation model from overfitting, and the crowd aggregation model utilizes human language skills to assist the autoencoder in differentiating the samples with large (semantic) variations.

Experimental results indicate that our model achieves superior or competitive results compared to the state-of-the-art models on two large text-crowd datasets. Specifically, CrowdDeepAE outperforms the alternative models with significant margins when the crowd labels are scarce. Figure 1 visualizes the discriminative ability of our model (*CrowdDeepAE*) compared to an alternative hybrid model (*MV-DeepAE*), when only 20% of the crowd labels are available on *CrowdFlower* dataset. *MV-DeepAE* contains majority voting aggregation method (*MV*) and our autoencoder sub-model (*DeepAE*). The outcome demonstrates more discriminative features using our model, indicating the importance of our joint learning framework. The contribution of this paper can be summarized as follows:

- Proposing a hybrid crowd-text model for sentiment analysis, consisting of a generative crowd aggregation model and a deep sentimental autoencoder, which are combined based on a probabilistic framework;
- Defining a unified objective function for the hybrid model, and deriving an efficient optimization algorithm to solve the problem;
- Achieving superior or competitive results compared to alternative models in our experiments, especially when the crowd labels are scarce.

Related Works

There are several datasets in different applications, which are labeled using crowdsourcing platforms like Amazon Mechanical Turk (Bachrach et al. 2012; Willett et al. 2013; Sadoughi, Liu, and Busso 2014; Sadoughi and Busso 2017). However, crowd labels are often noisy and unreliable, since crowd workers mostly lack expertise in the assigned tasks. To tackle this issue, each sample is usually labeled by multiple crowd workers, then these redundant crowd labels are used to estimate the potential true labels (*i.e.* truths). There are several studies, which proposed discriminative and generative models to efficiently aggregate crowd labels (Sheshadri and Lease 2013; Zheng et al. 2017). The discriminative aggregation models directly estimate the truths regardless of the crowd data distribution. Majority voting (*MV*) is the simplest discriminative aggregation model, which considers equal reliability for crowd workers and simply averages their votes. An intuitive and fast extension of majority voting, called iterative weighted majority voting (*IWMV*), is introduced in (Li and Yu 2014), which improves *MV* by considering a reliability parameter for each worker. Tian and Zhu also enhanced *MV* model by adopting the notion of max-margin from support vector machines, and introduced max-margin mjority voting (M^3V) as a new discriminative aggregation models (Tian and Zhu 2015).

In contrast to the discriminative aggregation models, the generative models employ a probabilistic model to represent the distribution of noisy observations (crowd labels) given the unknown variables (true labels) and model parameters. Dawid and Skene introduced a well known model (*DS*), which considers a confusion matrix as a reliability parameter for each worker in (Dawid and Skene 1979). Furthermore, several studies extended *DS* by assuming a prior distribution for parameters, and used Bayesian approach to compute their posterior distributions (Raykar et al. 2010; Chen, Lin, and Zhou 2013). Another generative model, called *GLAD*, considers a scalar parameter for the reliability of each worker and the difficulty of each task, and calculates the probability of truths using the logistic function of the parameters (Whitehill et al. 2009). Moreover, *GLAD* is extended in (Welinder et al. 2010) such that a vector instead of a scalar is considered as the parameter of each worker and sample. In addition to a confusion matrix as the reliability parameter of each worker, Zhou *et al.* assigned a confusion matrix as a difficulty parameter for each sample, and proposed an aggregation model based on minimax conditional entropy of crowd labels (Zhou et al. 2014; 2015). Later, Tian and Zhu regularized a variant of *DS* with the discriminative M^3V model, and jointly learned the parameters of both sub-models (Tian and Zhu 2015). In order to tackle the aggregation problem when crowd labels per worker are scarce, Venanzi *et al.* proposed CommunityBCC, which groups crowd workers into a few types (communities) and learns similar reliability parameters for each community (Venanzi et al. 2014).

The aforementioned aggregation models only use crowd labels to estimate the truths, but do not benefit from text data. There are a few studies on sentiment analysis using both crowd and text data (Brew, Greene, and Cunningham 2010; Musat Thisone, Ghasemi, and Faltings 2012; Simpson et al. 2015). In a recent work (Simpson et al. 2015), a Bayesian model is employed to combine the two modalities by considering a confusion matrix for each word and worker. Our proposed model also utilizes both crowd labels and text data; however, it leverages the power of deep models to provide a more discriminative language model, despite the shallow *BCCwords* model in (Simpson et al. 2015). Our model is also unique in the way that it combines a generative crowd aggregation model with a deep sentimental autoencoder using a probabilistic framework. Moreover, experimental results show the superiority of our model compared to *BCCwords*, especially when crowd labels are scarce.

Hybrid Sentiment Analysis Model

In this section, we first introduce our hybrid model by showing its architecture and explaining the intuition behind it. We then formulate its unified objective function based on a probabilistic framework, and derive an optimization algorithm for updating parameters and estimating truths.

CrowdDeepAE Architecture

The proposed hybrid model, denoted by *CrowdDeepAE*, consists of two main parts, a deep denoising autoencoder for text data and an aggregation model for crowd labels. Figure 2 demonstrates the architecture of *CrowdDeepAE*, in which the deep denoising autoencoder has two tasks, reconstructing the corrupted text data by noise and estimating the truths from text data. The crowdsourcing aggregation model is also supposed to estimate the truths from the noisy crowd labels. Hence, the truths are obtained by contributions of both crowd and text data.

Coupling the deep autoencoder and crowd aggregation model in *CrowdDeepAE* has several advantages: 1) *CrowdDeepAE* exploits two sources of information to estimate the truths more accurately, text data via the encoder pathway of the autoencoder and crowd data through the aggregation model; 2) The multi-layer autoencoder provides powerful discriminative features for the text samples, which have more capabilities than shallow models in learning the non-linear embedding space of real-world text data; 3) The reconstruction loss function in the denoising autoencoder plays a role of data-dependent regularization term, indirectly preventing the crowdsourcing aggregation model from overfitting; 4) *CrowdDeepAE* is able to annotate the entire dataset, even the samples without any crowd labels, since the autoencoder can be efficiently trained using the supervision of limited number of crowd labels and the unsupervised reconstruction task; 5) The aggregation model assists training the autoencoder using the semantic knowledge of crowd workers, which is very beneficial due to the large variations on text data; 6) The joint learning framework used for *CrowdDeepAE* leads to more optimal results compared to a naive non-joint learning approach, where the textual and crowd sub-models are trained separately.

CrowdDeepAE Objective Function

Lets consider the crowdsourcing task includes N questions, each with K possible options. The crowd and text data are represented by $\mathbf{X} = \{\mathbf{X}^{Cr}, \mathbf{X}^{Te}\}$, respectively, and $\mathbf{Y}$ indicates the unknown true labels. We provide a probabilistic framework to combine our autoencoder and aggregation sub-models, and consequently define a unified objective function for our hybrid model. The general likelihood function of *CrowdDeepAE* parameters ($\boldsymbol{\psi}$) given the observations ($\mathbf{X}^{Cr}, \mathbf{X}^{Te}$) is:

$$P(\mathbf{X}^{Cr},\mathbf{X}^{Te}|\boldsymbol{\psi}) = \prod_{i=1}^{N} P(\mathbf{X}_i^{Cr}, \mathbf{X}_i^{Te}|\boldsymbol{\psi}) \tag{1}$$

$$= \prod_{i=1}^{N}\sum_{c=1}^{K} P(\mathbf{X}_i^{Cr}, \mathbf{X}_i^{Te}, Y_i = c|\boldsymbol{\psi})$$

$$= \prod_{i=1}^{N}\sum_{c=1}^{K} P(\mathbf{X}_i^{Cr}, \mathbf{X}_i^{Te}|Y_i = c, \boldsymbol{\psi})P(Y_i = c|\boldsymbol{\psi})$$

$$= \prod_{i=1}^{N}\sum_{c=1}^{K} \underbrace{P(\mathbf{X}_i^{Cr}|Y_i = c, \boldsymbol{\theta})}_{\text{Crowd Aggregation Model}} \underbrace{P(Y_i = c|\mathbf{X}_i^{Te}, \mathbf{W})P(\mathbf{X}_i^{Te}|\mathbf{W})}_{\text{Deep Autoencoder}},$$

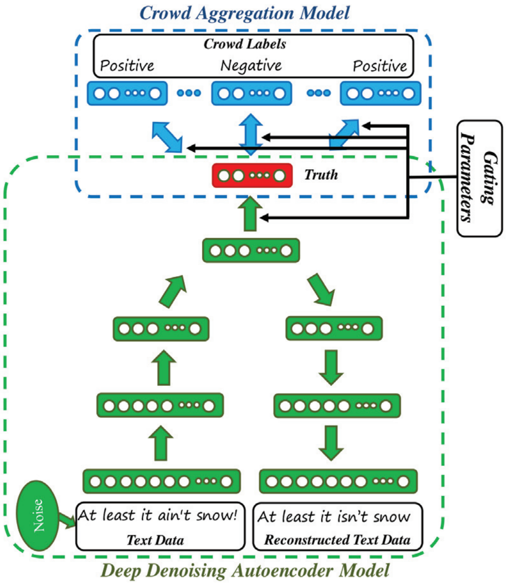


Figure 2: *CrowdDeepAE* architecture, consisting of a deep denoising autoencoder and a crowd aggregation model.

where i and c are the indices of questions and options, and $\mathbf{W}$ and $\boldsymbol{\theta}$ represent the parameters of autoencoder and crowd aggregation sub-models, respectively. Note that the samples are assumed independent and identically distributed (i.i.d), and $\mathbf{X}_i^{Cr}$ and $\mathbf{X}_i^{Te}$ are supposed to be conditionally independent given the true labels.

We are now able to decompose the likelihood function in Eq. (1) into the crowd aggregation and deep autoencoder objectives. Considering that M crowd workers are hired in the crowdsourcing task, our generative crowd aggregation model has the following form.

$$P(\mathbf{X}_i^{Cr}|Y_i = c, \boldsymbol{\theta}) = \prod_{j=1}^{M}\prod_{k=1}^{K} \Big[\frac{exp(\theta_{jck})}{\sum_{k'} exp(\theta_{jck'})}\Big]^{\mathbf{1}(x_{ij}^{Cr}=k)}$$

$$= \prod_{j=1}^{M}\prod_{k=1}^{K} [p_{ijck}]^{\mathbf{1}(x_{ij}^{Cr}=k)}\,, \tag{2}$$

where $\mathbf{X}_i^{Cr} = \{\mathbf{x}_{i1}^{Cr}, ..., \mathbf{x}_{iM}^{Cr}\}$ is the set of crowd labels for the i-th question. Also p_{ijck} shows the probability of a crowd label such that the j-th worker selects the k-th option for the i-th question, when c is the true label. Therefore, the joint probability of crowd data for each question is based on the probability of each conditionally independent crowd label. The aggregation model considers a confusion matrix $\boldsymbol{\theta}_j$ as the reliability parameter of each worker, in which higher diagonal elements θ_{jkk} indicate more reliability for the worker. Moreover, the exponential non-linearity increases the flexibility of our crowdsourcing aggregation model in dealing with the noisy crowd labels.

The direct optimization of log-likelihood function $\mathscr{L}(\boldsymbol{\psi}|\mathbf{X}) = \log P(\mathbf{X}^{Cr}, \mathbf{X}^{Te}|\boldsymbol{\psi})$ is difficult, hence we use Expectation-Maximization (EM) learning approach to solve this problem. Following, we present Proposition 1 to alleviate the optimization problem, and provide its proof in Appendix A. For the sake of simpler notations, hereafter we denote $P(Y_i = c|\mathbf{X}_i^{Te}, \mathbf{W}) = e_{ic}$ and $P(\mathbf{X}_i^{Te}|\mathbf{W}) = d_i$ as the probability of $\underline{e}$ncoder and $\underline{d}$ecoder pathways, respectively, and $\mathbf{1}(x_{ij}^{Cr} = k) = \mathbf{1}_{ijk}$.

Proposition 1: Iteratively improving the following auxiliary function $\mathbb{Q}$ is sufficient to maximize the log-likelihood function $\mathscr{L}(\boldsymbol{\psi}|\mathbf{X})$.

$$\mathbb{Q}(\boldsymbol{\psi}|\boldsymbol{\psi}^{(t)}) = \sum_{ijck} q_{ic}^{(t)} \log \Big(d_i e_{ic} [p_{ijck}]^{\mathbf{1}_{ijk}} \Big) \tag{3}$$

$$where \;\; q_{ic}^{(t)} = \frac{\prod_{jk} e_{ic} [p_{ijck}]^{\mathbf{1}_{ijk}}}{\sum_{c'} \prod_{jk} e_{ic'} [p_{ijc'k}]^{\mathbf{1}_{ijk}}}$$

where $q_{ic}^{(t)}$ shows the probability distribution of a truth. Technically, it is the expectation of an unknown true label with respect to the current parameters. Thus, we can iteratively improve the auxiliary function $\mathbb{Q}$ instead of the log-likelihood function $\mathscr{L}$.

In the $\mathbb{Q}$ function, the autoencoder and crowd workers have similar effects in the objective function and calculating the truths. However, it is expected that the deep autoencoder has more accurate predictions than the inexpert crowd workers due to the learned knowledge from all of questions. Hence, we control the influence of each factor in our objective function using adjustable weights. Following, we present the updated objective function, which can be derived similar to proposition 1.

$$\max_{\boldsymbol{\theta}, \mathbf{W}, \mathbf{1}^T \boldsymbol{\alpha} = M+1, \boldsymbol{\alpha} \geq 0} \; \sum_{ijck} q_{ic}^{(t)} \log \Big([d_i]^{\lambda_d} [e_{ic}]^{\alpha_0} [p_{ijck}]^{\alpha_j \mathbf{1}_{ijk}} \Big)$$
$$where \;\; q_{ic}^{(t)} \propto \prod_{jk} (e_{ic})^{\alpha_0} (p_{ijck})^{\alpha_j \mathbf{1}_{ijk}} \tag{4}$$

where $\boldsymbol{\alpha}$ and λ_d are the adjustable weights and the hyperparameter for the reconstruction loss of autoencoder, respectively. Note that $\boldsymbol{\alpha}$ can be seen as the gating parameters (see Figure 2), which adjust the contribution of each worker and also the autoencoder in estimating the truths. In other words, $\boldsymbol{\alpha}$ gives one more degree of freedom to our hybrid model about the credibility of crowd workers and autoencoder. For example, when there are several (non-expert) crowd workers labeling a question with (very noisy) crowd labels, a high weight for (discriminative) autoencoder can help estimating the truth accurately. Note that we define the weight for probability of decoder pathway by λ_d, since d_i does not affect the truths, and only regulates the autoencoder objective function. Furthermore, we add two more regularization penalty terms for the parameters to avoid overfitting.

$$\min_{\boldsymbol{\theta}, \mathbf{W}, \mathbf{1}^T \boldsymbol{\alpha} = M+1, \boldsymbol{\alpha} \geq 0} \; - \sum_{ijck} q_{ic}^{(t)} \log \Big([d_i]^{\lambda_d} [e_{ic}]^{\alpha_0} [p_{ijck}]^{\alpha_j \mathbf{1}_{ijk}} \Big)$$
$$+ \lambda_\theta \sum_j \|\boldsymbol{\theta}_j\|_F + \lambda_\alpha \|\boldsymbol{\alpha}\|_2 \,, \tag{5}$$

where λ_θ and λ_α are the hyperparameters of regularization terms. Also adding two constraints for $\boldsymbol{\alpha}$ (under min operation) is beneficial in our objective function for having competitive learning and avoiding the trivial solution $\boldsymbol{\alpha} = \mathbf{0}$.

CrowdDeepAE Optimization Algorithm

In order to efficiently solve problem (5), we employ an alternating learning strategy to update the parameters and estimate the truths. In particular, each one of the parameters $\boldsymbol{\psi} = \{\boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{W}\}$ is updated while the other parameters and truths are fixed, and the probability of truths $\mathbf{Q} = \{\mathbf{q}_1, ..., \mathbf{q}_N\}$ are estimated when the parameters are assumed to be known.

Update $\boldsymbol{\theta}$: The problem for updating the parameters of crowd aggregation model is reduced to:

$$\min_{\boldsymbol{\theta}} \; - \sum_{ijck} q_{ic}^{(t)} \log \Big([p_{ijck}]^{\alpha_j \mathbf{1}_{ijk}} \Big) + \lambda_\theta \sum_j \|\boldsymbol{\theta}_j\|_F \tag{6}$$

There are several first-order optimization algorithms that can be used to solve this problem. Using the following gradient of the objective function *wrt* the parameter $\boldsymbol{\theta}$, we employ L-BFGS algorithm to iteratively update the parameters.

$$\frac{\partial \mathbb{Q}}{\partial \theta_{jck}} = \sum_i q_{ic}^{(t)} \alpha_j [\mathbf{1}_{ijk} - p_{ijck}] \tag{7}$$

Update $\boldsymbol{\alpha}$: The problem to update the gating parameters boils down to:

$$\min_{\mathbf{1}^T \boldsymbol{\alpha} = M+1, \boldsymbol{\alpha} \geq 0} \; \lambda_\alpha \boldsymbol{\alpha}^T \boldsymbol{\alpha} - \boldsymbol{\alpha}^T \boldsymbol{\beta} \tag{8}$$

where $\beta_0 = \sum_{ic} q_{ic}^{(t)} \log e_{ic}$, $\beta_j = \sum_{ick} q_{ic}^{(t)} \mathbf{1}_{ijk} \log p_{ijck}$. We efficiently solve this problem using the Lagrangian multiplier method as shown in Appendix B.

Update $\mathbf{W}$: The problem to update the parameters of deep denoising autoencoder has the following form.

$$\min_W \; - \sum_{ic} q_{ic}^{(t)} \log P_W(Y_i = c|\mathbf{X}_i^{Te}) - \frac{\lambda_d}{\alpha_0} \log P_W(\mathbf{X}_i^{Te}) \,,$$

where the first term is the standard cross entropy loss function for classification problems. But for the second probability term, we use a theorem in (Bengio et al. 2013) in order to change the term to reconstruction loss function in the standard denoising autoencoder.

The general idea is that if the observation variable X is corrupted into $\tilde{X}$ by a noise with conditional distribution $\mathscr{C}(\tilde{X}|X)$, training a denoising autoencoder actually estimates the reverse conditional distribution $P(X|\tilde{X})$. It has been shown that a consistent estimator of $P(X)$ can be estimated using a Markov chain that alternates between sampling from $P(X|\tilde{X})$ and sampling from $\mathscr{C}(\tilde{X}|X)$ as follows.

$$X_t \sim P_W(X|\tilde{X}_{t-1}) \qquad \tilde{X}_t \sim \mathscr{C}(\tilde{X}|X_t)$$

The theorem proves that $P_W(X|\tilde{X})$ of conventional denoising autoencoder (Vincent et al. 2008; Bengio et al. 2013; Ghasedi Dizaji et al. 2017) is a consistent estimator of the true conditional distribution. Also as the number of samples $N \to \infty$, the asymptotic distribution of the generated

```
Algorithm 1: CrowdDeepAE Algorithm
Initialize $\mathbf{q}_i$ by majority voting $\forall i \in \{1, ..., N\}$
while not converged do
    Solve problem (6) to update $\boldsymbol{\theta}$
    Solve problem (8) to update $\boldsymbol{\alpha}$
    Solve problem (9) to update $\mathbf{W}$
    $q_{ic} \propto \prod_{jk} (e_{ic})^{\alpha_0} (p_{ijck})^{\alpha_j \mathbf{1}_{ijk}}$
end
```

samples by the denoising autoencoder converges to original data-generating distribution. Hence, we reformulate the objective function of the text model as follows:

$$\min_{W} \; -\sum_{ic} q_{ic}^{(t)} \log P_W(Y_i = c|\mathbf{X}_i^{Te}) - \frac{\lambda_d}{\alpha_0} \log P_W(\mathbf{X}_i^{Te}|\tilde{\mathbf{X}}_i^{D}) \, , \tag{9}$$

where $\tilde{\mathbf{X}}_i^D$ is a sample corrupted by a random noise. Now it is clear how we can use the denoising autoencoder as our text-based sub-model.

Interestingly, our learning approach does not have memory exhaustion problems when handling very large datasets. In order to learn the reliability parameters $\boldsymbol{\theta}$ for large number of crowd workers, we can split the crowd data into several mini-batches, each one only including the crowd labels of a few workers. Dealing with a large set of text samples, we are able to distribute the text samples into a set of mini-batches and train the autoencoder parameters with stochastic optimization algorithms. Therefore, the computation and space complexities can be managed using stochastic and parallel learning approaches.

Algorithm 1 shows the *CrowdDeepAE* algorithm, in which the truths are first initialized by majority voting. It then alternates between updating the model parameters and estimating the truths until convergence. It is worth mentioning that we compute the truths using the clean text samples in E-step. But the classification loss function with noisy text inputs in Eq. (9) has the regularization effect in training the parameters $\mathbf{W}$, and results in to the more robust and generalized autoencoder model.

Experiments and Discussions

In this section, we first evaluate the performance of our hybrid model in the crowd aggregation task, and then examine the quality of the learned language models. In order to compare the proposed model with the state-of-the-art aggregation models, we use two large-scale crowdsourcing datasets, which have text data along with crowd labels for sentiment analysis.

Datasets: *CrowdFlower* (*CF*) dataset was a part of the 2013 Crowdsourcing at Scale shared task challenge, collected by CrowdFlower[2] as a rich source for the sentiment analysis of tweets about the weather. The dataset includes 569,375 crowd labels for 98,980 tweets. But the gold-standard (true) labels are only provided for 300 tweets, which correspond to 1720 crowd labels collected from 461 workers. In the crowd task, workers are requested to label the sentiment of tweets related to weather using the following options, negative (0), neutral (1), positive (2) and not related to weather (4). The crowd workers are also able to skip the questions by the can not tell (5) option.

Sentiment Polarity (*SP*) dataset includes the sentiment analysis of crowd workers about the movie reviews across two categories, "fresh" (positive) and "rotten" (negative). The dataset consists of 5,000 sentences from the movie reviews in RottenTomatoes website[3], which is extracted by (Pang and Lee 2004). A task requester hired 203 crowd workers to label the dataset, resulting in 27,747 crowd labels totally. The gold-standard labels for all the questions are available in *SP* dataset.

Implementation details: For both *CF* and *SP* datasets, we first use the stemming approach to parse the texts (Porter 1980), then remove the common English stop words and finally extract the top 1000 words according to the term frequency-inverse document frequency (tf-idf) score (Baeza-Yates, Ribeiro-Neto, and others 1999).

For the deep autoencoder, we consider three fully connected layers for both encoder and decoder pathways with 512, 256, and 128 neurons as the feature maps, and then add a softmax layer on top of the encoder pathway. The leaky rectified activation (leaky RELU) is used as the activation function for the autoencoder layers, except the reconstruction layer at the end of decoder pathway, which has rectified activation (RELU) to reconstruct text samples. Moreover, we set the learning rate to 10^{-4} and adopt Adam (Kinga and Adam 2015) as our optimization method. The weights of all layers are also initialized by the Xavier or GlorotUniform initialization approach (Glorot and Bengio 2010).

Since the crowdsourcing task is an unsupervised problem, we did not use any true labels for setting the hyper-parameters $\{\lambda_\theta, \lambda_\alpha, \lambda_d\}$ and dropout noise value. We use a trick in (Tian and Zhu 2015), that employs the non-related likelihood for selecting the hyper-parameters. In particular, we utilize the likelihood function $p(\mathbf{X}^{Cr}|\mathbf{Y}, \boldsymbol{\theta})$ to choose λ_α, λ_d and dropout from $\lambda_\alpha^{set} = \{0.01, 0.1, 1\}$, $\lambda_d^{set} = \{0.01, 0.1, 1\}$ and dropout$^{set} = \{0.1, 0.2, 0.3\}$, and adopt $p(\mathbf{Y}|\mathbf{X}^{Te}, \mathbf{W})$ as a criterion to choose λ_θ from $\lambda_\theta^{set} = \{0.01, 0.1, 1\}$. Thus using this approach, we make sure to select the hyper-parameters without any knowledge from the true labels.

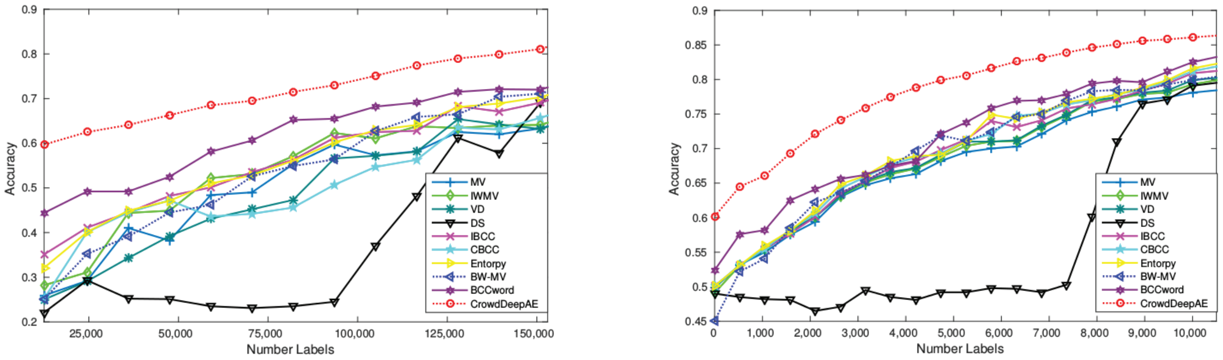
Evaluation of Aggregation Models

To evaluate the performance of our model, we run several experiments using *CF* and *SP* datasets to estimate the truths using crowd labels and text data. For the sake of comparison, we use the following alternative models and comparison metrics.

Alternative models: We compare our model, *CrowdDeepAE*, with several baseline methods, including majority voting (*MV*), iterative weighted majority voting (*IWMV*) (Li and Yu 2014), vote distribution (*VD*), Dawid and Skene model (*DS*) (Dawid and Skene 1979), Independent Bayesian

[2] www.crowdflower.com

[3] www.rottentomatoes.com



(a) *CrowdFlower* (*CF*)

(b) *SentimentPolarity* (*CF*)

Figure 3: Accuracy of crowdsourcing aggregation models on *CrowdFlower* (*CF*) and *SentimentPolarity* (SP) datasets, when increasing the number of crowd labels.

		CF (20% labels)				SP (20% labels)			
	Model	Accuracy	Ave. recall	NLPD	AUC	Accuracy	Ave. recall	NLPD	AUC
Crowd	*MV*	0.625	0.550	1.392	0.725	0.710	0.710	1.192	0.704
	IWMV	0.630	0.562	1.368	0.735	0.710	0.710	1.167	0.715
	VD	0.650	0.585	1.252	0.745	0.710	0.710	1.112	0.728
	DS	0.610	0.488	1.285	0.681	0.500	0.500	0.695	0.500
	IBCC	0.688	0.545	0.972	0.822	0.740	0.740	0.516	0.835
	CBCC	0.635	0.532	1.052	0.800	0.726	0.726	0.540	0.818
	Entropy	0.688	0.545	1.014	0.818	0.745	0.745	0.508	0.842
Crowd-Text	*MV-BW*	0.665	0.602	2.133	0.749	0.722	0.722	0.648	0.784
	MV-DeepAE	0.682	0.611	1.372	0.792	0.738	0.738	0.615	0.800
	BCCwords	0.715	0.578	0.918	0.830	0.750	0.750	0.516	0.840
	CrowdDeepAE	**0.790**	**0.642**	**0.889**	**0.876**	**0.816**	**0.816**	**0.500**	**0.875**

Table 1: Comparison of crowdsourcing aggregation models on *CrowdFlower* (*CF*) and *SentimentPolarity* (*SP*) datasets, When 20% of crowd labels are available. The comparison metrics are accuracy, ave. recall, AUC (the higher the better), and NLPD (the lower the better).

Classifier Combination model (*IBCC*) (Simpson et al. 2013), Community-Based Bayesian Classifier Combination model (*CBCC*) (Venanzi et al. 2014), multi-class minimax entropy model (*Entropy*) (Zhou et al. 2014), combination of majority voting aggregation model and bag-of-words text classifier (*MV-BW*), combination of majority voting aggregation model and a deep sentimental autoencoder similar to our autoencoder (*MV-DeepAE*), and Bayesian classifier combination with words model (*BCCwords*) (Simpson et al. 2015).

It should be noted that *VD* can be considered as a probabilistic version of *MV*, since it computes the probability of each option, while assuming equal reliability for all the workers. Moreover, the *MV-BW* model trains a classical bag-of-words classifier for text data using the target label induced by majority voting aggregation model. Similarly, *MV-DeepAE* uses the predicted labels of majority voting aggregation model to train the deep autoencoder model for text data. The results of alternative models are reported from reference papers, except *MV-DeepAE* that is implemented by us with the similar autoencoder network to *CrowdDeepAE*.

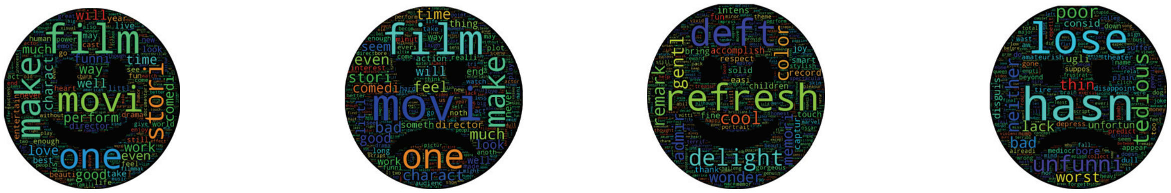
Comparison metrics: Following (Simpson et al. 2015), we measure the performance of models using *accuracy*, *average recall*, *negative log-probability density (NLPD)* (Venanzi et al. 2014), and *area under curve (AUC)* (Simpson et al. 2013). For *CF* dataset, we use mean AUC over pair of classes as shown in (Hand and Till 2001).

Performance comparison: In order to examine the effectiveness of the aforementioned aggregation models, we run several experiments with different subsets (number of crowd labels) of CF and *SP* datasets. Following (Simpson et al. 2013), we estimate the truths using the aggregation models when only 2% randomly-chosen crowd labels are available. Then, we increase the number of crowd labels by adding an extra 2% randomly-chosen crowd labels, and rerun all the models. This process is repeated until all of the crowd labels are used for training.

Figure 3 shows the accuracy of aggregation models on both *CF* and *SP* datasets. As it is shown, *CrowdDeepAE* consistently outperforms the other models with significant margins, especially when a small number of crowd labels are available. Interestingly in *CF* dataset, our model only requires 16% of the crowd labels to have a better accuracy than the all of other models, which are using 30% of the crowd labels. *CrowdDeepAE* also achieves a higher accuracy with 8% of the crowd labels in *CF* dataset versus *MV* model with 30% of the crowd labels. Furthermore, *CrowdDeepAE* consistently improves the performance of *MV-DeepAE*, and consequently confirms the importance of our joint learning

	Model	*CF* (all labels) Accuracy	 Ave. recall	 NLPD	 AUC	*SP* (all labels) Accuracy	 Ave. recall	 NLPD	 AUC
Crowd	*MV*	0.840	0.764	0.921	0.852	0.852	0.852	0.797	0.885
	IWMV	0.860	0.764	0.912	0.041	0.885	0.885	0.752	0.891
	VD	0.883	0.779	0.458	0.942	0.887	0.887	**0.338**	0.947
	DS	0.830	0.745	0.459	0.897	0.914	0.914	0.340	**0.957**
	IBCC	0.860	0.763	**0.437**	0.935	**0.915**	**0.915**	0.374	**0.957**
	CBCC	0.886	0.746	0.526	0.942	**0.915**	**0.915**	0.383	**0.957**
	Entropy	0.886	0.746	0.551	0.938	0.914	0.914	0.391	**0.957**
Crowd-Text	*MV-BW*	0.867	0.764	0.921	0.859	0.885	0.885	0.797	0.891
	MV-DeepAE	0.880	0.768	0.571	0.922	0.885	0.885	0.752	0.891
	BCCwords	0.890	0.807	0.591	0.877	**0.915**	**0.915**	0.389	**0.957**
	CrowdDeepAE	**0.912**	**0.825**	0.479	**0.948**	**0.915**	**0.915**	0.389	**0.957**

Table 2: Comparison of crowdsourcing aggregation models on *CrowdFlower* (*CF*) and *SentimentPolarity* (*SP*) datasets, When all crowd labels are available. The comparison metrics are accuracy, ave. recall, AUC (the higher the better), and NLPD (the lower the better).



(a) Pos-docStatistic (b) Neg-docStatistic (c) Pos-CrowdDeepAE (d) Neg-CrowdDeepAE

Figure 4: Word clouds of the positive (Pos) and negative (Neg) sentiments in SP dataset. The extracted word clouds using the statistics of documents (docStatistic) and our language model (*CrowdDeepAE*) are shown in the left and right, respectively. The colors are only for legibility.

framework and our crowd aggregation sub-model. Note that we only show a limited portion of the results (approximately 150,000 and 10,000 crowd labels in *CF* and *SP* datasets) in Figure 3 for the sake of a clear visualization.

Furthermore, Table 1 and 2 report the mentioned comparison metrics for the aggregation models on *CF* and *SP* datasets, when 20% and 100% of crowd labels are available, respectively. We divide the models in the tables into two groups of single and hybrid models, where the first ones only employ crowd labels to estimate the truths, and the second ones utilize both crowd labels and text data for the prediction task. Using only 20% of crowd labels, approximately 70% of the text samples have at least one crowd label. In this case, using text data is more crucial, since enough crowd labels are not available for training the crowd parameters. The hybrid crowd-text models have relatively better performances than the crowd models, because the hybrid models are able to employ language model to classify the samples with no crowd labels. But the crowd models suffer from insufficient crowd labels for training, and assign a default category for the unlabeled samples based on their prior distribution. Our proposed model, *CrowdDeepAE*, benefits from the deep autoencoder trained by a small subset of crowd data, and is able to efficiently label the samples with no crowd labels. When only 20% of crowd labels are available, our model outperforms the alternative models on both *SP* and *CF* datasets according to all metrics. In addition, *CrowdDeepAE* still achieves superior or competitive results in comparison with the state-of-the-art models on both datasets using all crowd labels. It indicates that *CrowdDeepAE* leverages the powerful deep language model along with the efficient crowd aggregation model to provide accurate predictions using crowd and text data.

Evaluation of Language Models

In order to visualize the learned language model in *CrowdDeepAE*, we show the word clouds for both *CF* and *SP* datasets. In particular, the word cloud represents the importance (probability) of each word in a document with its font size. Using this visual representation, a viewer can quickly identify the dominant words in a document using their relative sizes. For each word in the datasets, we generate an auxiliary variable by setting the corresponding element in $\mathbf{X}_i^{Te}$ equal to 1 and the remaining ones to zero, and then compute the probability of the word for every class. Figure 4 demonstrates the word clouds of *CrowdDeepAE* in *CF* dataset for the positive and negative classes. We also show the word cloud of *CF* dataset using the probability (frequency) of each word in every sentiment class. The world clouds extracted from the documents statistic (docStatistic) mostly assign greater importance to the highly repeated words like “movie”, “film”, and “stori”, which do not differentiate the two classes. However, the word clouds of *CrowdDeepAE* discriminantly represent the positive sentiments using the words with roots like “refresh”, “deft”, “delight” and “gentl”; and the negative class with the words like “lose”, “hasn”, “tedious”, and “unfunni”. The word clouds for *CF* dataset are shown in Appendix C.

Conclusion

In this paper, we proposed a new crowdsourcing aggregation model that is augmented by a deep sentimental autoencoder. The crowd aggregation and autoencoder sub-models are combined in a probabilistic framework rather than a heuristic way. We introduced a unified objective function, and then derived an efficient optimization algorithm to alternatingly solve the corresponding problem. Experimental results showed that our model outperforms the alternative models, especially when the crowd labels are scarce. Although the proposed model was applied only in sentiment analysis, it can be used as the general hybrid model for different applications in future works.

Acknowledgement

This work was partially supported by the following grants: NSF-IIS 1302675, NSF-IIS 1344152, NSF-DBI 1356628, NSF-IIS 1619308, NSF-IIS 1633753, NIH R01 AG049371.

References

Bachrach, Y.; Graepel, T.; Minka, T.; and Guiver, J. 2012. How to grade a test without knowing the answers, a bayesian graphical model for adaptive crowdsourcing and aptitude testing. In *arXiv preprint arXiv:1206.6386*.

Baeza-Yates, R.; Ribeiro-Neto, B.; et al. 1999. *Modern information retrieval*, volume 463. ACM press New York.

Bengio, Y.; Yao, L.; Alain, G.; and Vincent, P. 2013. Generalized denoising auto-encoders as generative models. In *NIPS*, 899–907.

Brew, A.; Greene, D.; and Cunningham, P. 2010. Using crowdsourcing and active learning to track sentiment in online media. In *ECAI*, 145–150.

Chen, X.; Lin, Q.; and Zhou, D. 2013. Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In *ICML*, 64–72.

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

Ghasedi Dizaji, K.; Herandi, A.; Deng, C.; Cai, W.; and Huang, H. 2017. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *ICCV*, 5736–5745.

Ghasedi Dizaji, K.; Yang, Y.; and Huang, H. 2017. Joint generative-discriminative aggregation model for multi-option crowd labels. In *WSDM*.

Glorot, X., and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 249–256.

Hand, D. J., and Till, R. J. 2001. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine learning* 45(2):171–186.

Kinga, D., and Adam, J. B. 2015. A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Li, H., and Yu, B. 2014. Error rate bounds and iterative weighted majority voting for crowdsourcing. *arXiv preprint arXiv:1411.4086*.

Musat Thisone, C.-C.; Ghasemi, A.; and Faltings, B. 2012. Sentiment analysis using a novel human computation game. In *Proceedings of the 3rd Workshop on the People's Web Meets NLP: Collaboratively Constructed Semantic Resources and their Applications to NLP*, 1–9. Association for Computational Linguistics.

Pang, B., and Lee, L. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, 271. Association for Computational Linguistics.

Porter, M. F. 1980. An algorithm for suffix stripping. *Program* 14(3):130–137.

Raykar, V. C.; Yu, S.; Zhao, L. H.; Valadez, G. H.; Florin, C.; Bogoni, L.; and Moy, L. 2010. Learning from crowds. *The Journal of Machine Learning Research* 11:1297–1322.

Sadoughi, N., and Busso, C. 2017. Joint learning of speech-driven facial motion with bidirectional long-short term memory. In *International Conference on Intelligent Virtual Agents*, 389–402. Springer.

Sadoughi, N.; Liu, Y.; and Busso, C. 2014. Speech-driven animation constrained by appropriate discourse functions. In *Proceedings of the 16th International Conference on Multimodal Interaction*, 148–155. ACM.

Sheshadri, A., and Lease, M. 2013. Square: A benchmark for research on computing crowd consensus. In *First AAAI Conference on Human Computation and Crowdsourcing*.

Simpson, E.; Roberts, S. J.; Psorakis, I.; and Smith, A. 2013. Dynamic bayesian combination of multiple imperfect classifiers. *Decision making and imperfection* 474:1–35.

Simpson, E. D.; Venanzi, M.; Reece, S.; Kohli, P.; Guiver, J.; Roberts, S. J.; and Jennings, N. R. 2015. Language understanding in the wild: Combining crowdsourcing and machine learning. In *WWW*, 992–1002.

Tian, T., and Zhu, J. 2015. Max-margin majority voting for learning from crowds. In *NIPS*, 1612–1620.

Venanzi, M.; Guiver, J.; Kazai, G.; Kohli, P.; and Shokouhi, M. 2014. Community-based bayesian aggregation models for crowdsourcing. In *WWW*, 155–164.

Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P.-A. 2008. Extracting and composing robust features with denoising autoencoders. In *ICML*, 1096–1103. ACM.

Welinder, P.; Branson, S.; Belongie, S. J.; and Perona, P. 2010. The multidimensional wisdom of crowds. In *NIPS*, volume 23, 2424–2432.

Whitehill, J.; Wu, T.-f.; Bergsma, J.; Movellan, J. R.; and Ruvolo, P. L. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, 2035–2043.

Willett, K. W.; Lintott, C. J.; Bamford, S. P.; Masters, K. L.; Simmons, B. D.; Casteels, K. R.; Edmondson, E. M.; Fortson, L. F.; Kaviraj, S.; Keel, W. C.; et al. 2013. Galaxy zoo 2: detailed morphological classifications for 304 122 galaxies from the sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society* 435(4):2835–2860.

Zheng, Y.; Li, G.; Li, Y.; Shan, C.; and Cheng, R. 2017. Truth inference in crowdsourcing: is the problem solved? *Proceedings of the VLDB Endowment* 10(5):541–552.

Zhou, D.; Liu, Q.; Platt, J.; and Meek, C. 2014. Aggregating ordinal labels from crowds by minimax conditional entropy. In *ICML*, 262–270.

Zhou, D.; Liu, Q.; Platt, J. C.; Meek, C.; and Shah, N. B. 2015. Regularized minimax conditional entropy for crowdsourcing. *arXiv preprint arXiv:1503.07240*.