Aggregating Crowd Wisdoms with Label-aware Autoencoders

Li'ang Yin, Jianhua Han, Weinan Zhang, Yong Yu
Shanghai Jiao Tong University
No.800 Dongchuan Road
Shanghai, 200240, China
{yinla,hanjianhua44,wnzhang,yyu}@apex.sjtu.edu.cn

Abstract

Aggregating crowd wisdoms takes multiple labels from various sources and infers true labels for objects. Recent research work makes progress by learning source credibility from data and roughly form three kinds of modeling frameworks: weighted majority voting, trust propagation, and generative models. In this paper, we propose a novel framework named **L**abel-**A**ware **A**utoencoders (LAA) to aggregate crowd wisdoms. LAA integrates a classifier and a reconstructor into a unified model to infer labels in an unsupervised manner. Analogizing classical autoencoders, we can regard the classifier as an encoder, the reconstructor as a decoder, and inferred labels as latent features. To the best of our knowledge, it is the first trial to combine label aggregation with autoencoders. We adopt networks to implement the classifier and the reconstructor which have the potential to automatically learn underlying patterns of source credibility. To further improve inference accuracy, we introduce object ambiguity and latent aspects into LAA. Experiments on three real-world datasets show that proposed models achieve impressive inference accuracy improvement over state-of-the-art models.

1 Introduction

Aggregating crowd wisdoms is also known as label aggregation for crowdsourcing or truth discovery [Li *et al.*, 2016]. It is an increasingly important topic in machine learning. Many tasks of machine learning require large labeling datasets. Traditional label collection from domain experts is usually expensive and time-consuming, which may not match the increasing requirement for labels. Labeling by the crowd has become popular with the blooming of online crowdsourcing platforms such as Amazon Mechanical Turk [Ipeirotis, 2010] and CrowdFlower [De Winter *et al.*, 2015]. Such a platform divides the whole labeling task into small parts and distributes them to ordinary web users (sources). Despite of low cost, crowdsourced labeling commonly suffers from (much) lower accuracy than that from experts. Therefore in many labeling tasks, for each object we need to aggregate multiple labels from different users to reduce the labeling noise [Tian and Zhu, 2015a].

Label aggregation takes multiple labels from various sources as input and infers true labels for objects. This is a typical unsupervised learning task as there is no ground truth provided for inferring labels. The most simple and widely used method is majority voting [Aydin *et al.*, 2014]. It treats sources equally and picks the most voted label as the true label. Recent research work mainly models source credibility (or capability). The underlying assumption is that sources with high credibility assign labels more accurately than those with low credibility [Yin *et al.*, 2008; Li *et al.*, 2014]. There are roughly three kinds of modeling frameworks: weighted majority voting, trust propagation, and generative models. Weighted majority voting is the direct extension from traditional majority voting [Aydin *et al.*, 2014; Li *et al.*, 2014]. Trust propagation models both credibility of sources and reliability of provided labels [Yin *et al.*, 2008; Pasternack and Roth, 2010; Galland *et al.*, 2010]. More recent work can be categorized into the generative framework [Whitehill *et al.*, 2009; Welinder *et al.*, 2010; Bachrach *et al.*, 2012; Qi *et al.*, 2013; Simpson *et al.*, 2013; Tian and Zhu, 2015a]. These methods generate source labels from underlying (unknown) true labels by probabilistic models and infer true labels by MAP or Bayesian estimation. Though these methods have superior inference performance to majority voting, they need to model sophisticated relationships between source labels and inferred labels (by experts). There are two weak points of such models. One is that they are usually designed for data with typical characteristics but may not generalize to the data with some other characteristics. The other one is that even experts may improperly model relationships between source labels and inferred labels which limits the inference performance (e.g. missing effective factors or adding too many constraints).

In this paper, we propose a novel framework named **L**abel-**A**ware **A**utoencoders (LAA) to aggregate crowd wisdoms. By vectorizing source labels, label aggregation is simplified as a classification problem to predict true labels from source labels. Since label aggregation is unsupervised and there is no ground truth for training a classifier, we combine a classifier and a reconstructor into a unified framework. The idea is motivated by classical autoencoders which encode input into latent features in the hidden layer and reconstruct the input

from latent features in the output layer [Vincent *et al.*, 2008]. We can regard the classifier in LAA as an encoder and inferred labels as latent features of the input. To the best of our knowledge, it is the first trial to combine label aggregation with autoencoders.

The framework is flexible for various implementations. In this paper we adopt networks to implement the classifier and the reconstructor. Instead of manually modeling sophisticated relationships between source labels and inferred labels, networks have the potential to automatically learn those underlying patterns. That property makes proposed model domain-free and easy to implement for different data. To further improve inference accuracy, we introduce object ambiguity and latent aspects into the classifier and the reconstructor.

Experiments on three real-world datasets show that even the basic version of LAA has competitive inference performance with the state-of-the-art. Modeling object ambiguity and latent aspects further improves the inference accuracy significantly over the state-of-the-art. We also examine learned patterns in networks to support the effectiveness of proposed models.

2 Related Work

The research of label aggregation can be traced back to 1979. Dawid and Skene proposed a probabilistic model to aggregate observations for patients [Dawid and Skene, 1979]. Recent research about this topic rises with the concept of truth discovery [Yin *et al.*, 2008].

Models of label aggregation can be roughly categorized into three frameworks: weighted majority voting, trust propagation, and generative models. Weighted majority voting is the direct extension from traditional majority voting [Aydin *et al.*, 2014; Li *et al.*, 2014]. The key of these methods is estimating source weights or credibility. Despite of mediocre inference accuracy, weighted majority voting is intuitive and easy to implement. Trust propagation models [Yin *et al.*, 2008; Pasternack and Roth, 2010; Galland *et al.*, 2010] assume labels provided by trustworthy (high credible) sources are more reliable and sources providing reliable labels are more trustworthy. Without prior structures or parameters, these models need a sufficient number of labels and may suffer from sparse data. More recent work usually utilizes the generative framework [Whitehill *et al.*, 2009; Welinder *et al.*, 2010; Bachrach *et al.*, 2012; Qi *et al.*, 2013; Simpson *et al.*, 2013; Tian and Zhu, 2015a]. These methods generate source labels from the underlying (unknown) true labels by probabilistic models and are trained via maximizing a posteriori (MAP) or Bayesian estimation. Besides modeling source credibility, various factors are introduced with the flexibility of probabilistic models, such as object difficulty [Bachrach *et al.*, 2012] and confusion matrix [Simpson *et al.*, 2013]. Other interesting work about label aggregation includes truth existence modeling [Zhi *et al.*, 2015], minimax conditional entropy [Zhou *et al.*, 2012], rank aggregating [Metrikov *et al.*, 2015], crowd clustering [Gomes *et al.*, 2011], etc.

With the prevalence of deep learning and learning representations, autoencoders have become a widely adopted unsupervised model during the last five years [Vincent *et al.*, 2010]. Autoencoders perform unsupervised learning in a supervised learning fashion: trying to recover the input through a network with a small-sized hidden layer [Vincent *et al.*, 2008]. Recently, variational autoencoders [Kingma and Welling, 2013], a marriage between Bayesian inference and autoencoders, attract much attention in building deep generative models to learn data distributions.

Despite of the wide usage of autoencoders, to the best of our knowledge, there is no previous work on label aggregation that leverages autoencoders to learn the latent data patterns amongst the source labels and infer true labels.

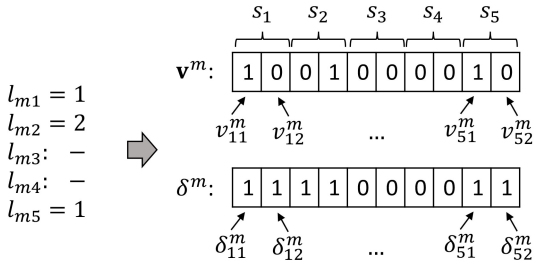


Figure 1: An example of constructing a source label vector. Here 5 sources label several objects with binary labels. For object o_m, source 1, 2, and 5 give their labels respectively while source 3 and 4 do not. A source label vector $\mathbf{v}^m$ is constructed by one-hot encoding for each source block. An accompanying mask vector δ^m indicates whether a source gives a label for object o_m.

3 Methods

3.1 Problem Definition

Suppose there are M objects, N sources, and a set of labels the sources give to objects. We denote l_{mn} the label object o_m received from source s_n, $m \in \{1, ..., M\}$ and $n \in \{1, ..., N\}$. A categorical label $l_{mn} \in \{1, ..., K\}$ where K is the number of categories. The goal of label aggregation is to infer a true label $\tilde{y}_m$ for each object o_m.

3.2 Label Vectorization

We represent an object by a source label vector. For object o_m, vector $\mathbf{v}^m$ of $N \times K$ dimensionality is constructed to contain all labeling information of the object. The vector can be divided into N blocks where each block contains K consecutive elements. The n-th block corresponds to the label given by source s_n. We use source-wise one-hot encoding to make the vector discriminative between categories. Figure 1 illustrates the construction of a source label vector. Let v^m_{nk} denote the k-th element of block n for object o_m. If source s_n labels o_m as category k (i.e. $l_{mn} = k$), v^m_{nk} is set to 1 while the other elements of the same block are 0. If source s_n does not assign any label to o_m, then all K elements of block n are set to 0. An accompanying mask vector δ^m is constructed to conveniently indicate whether source s_n gives the label or not. δ^m has the same dimensionality as $\mathbf{v}^m$. If l_{mn} exists, then all corresponding K elements of δ^m_{nk} are set to 1 ($k \in \{1, ..., K\}$), 0 otherwise.

It is obvious that all the label vectors of different objects have the same dimensionality. This property makes following learning methods feasible. A source label vector is also called an input vector when used in the label-aware autoencoder.

3.3 Label-aware Autoencoders

Since a source label vector contains all given information of one object, we can exploit a model or a classifier to take a source label vector $\mathbf{v}^m$ as input and output the true label $\tilde{y}_m$. However, training a classifier in traditional supervised machine learning problems needs partial ground-truth labels, but label aggregation is totally unsupervised [Li *et al.*, 2016]. Therefore, we propose a novel framework named **L**abel-**A**ware **A**utoencoders (LAA) to infer true labels in such an unsupervised scenario. LAA integrates a classifier and a reconstructor into a unified model. The classifier infers true labels from input and the reconstructor reconstructs input from inferred labels in an unsupervised manner. Analogizing classical autoencoders, we can regard the classifier as an encoder, the reconstructor as a decoder, and inferred labels as latent features of input [Vincent *et al.*, 2008].

Formally, we describe the mechanism of LAA from the view of maximizing the log-likelihood of input. For a given input vector v (the superscript m is omitted for simplicity), denote the classifier as $q_\theta(y|v)$ and the reconstructor as $p_\phi(v|y)$, where θ and ϕ are model parameters, y is the inferred label. LAA maximizes the lower bound of log-likelihood $\log p(v)$, which is an analogy to variational autoencoders [Kingma and Welling, 2013].

$$\begin{aligned}\log p(v) &= \sum_{y=1}^{K} q_\theta(y|v) \log \frac{p(y,v)}{q_\theta(y|v)} + D_{KL}(q_\theta(y|v)||p(y|v)) \\ &\geq \sum_{y=1}^{K} q_\theta(y|v) \log \frac{p_\phi(v|y)p(y)}{q_\theta(y|v)} \\ &= \mathbb{E}_{q_\theta(y|v)} \log p_\phi(v|y) - D_{KL}(q_\theta(y|v)||p(y)).\end{aligned} \tag{1}$$

On the right hand side in formula (1), the first term measures the expectation of reconstruction quality. It encourages the probability $p_\phi(v|y)$ to be 1 to achieve good reconstruction. The second term is the negative KL divergence between the distribution of inferred label $q_\theta(y|v)$ and the prior distribution $p(y)$, which acts as the regularization term to constrain the inferred label distribution to the prior one.

3.4 Network Implementation

In this paper we adopt networks to implement the classifier and the reconstructor in LAA. We first construct a basic version LAA-B. It does not need extra knowledge about sources or objects. Figure 2 illustrates the architecture. The classifier $q_\theta(y|v)$ is modeled by a network where we obtain a label vector $\bar{\mathbf{y}}^m$ from input $\mathbf{v}^m$.

$$\bar{\mathbf{y}}^m = \sigma(\mathbf{v}^m \mathbf{w}_q) \tag{2}$$

where weight matrix $\mathbf{w}_q$ corresponds to classifier parameter θ (the bias term is omitted for simplicity). $\sigma(\cdot)$ is the softmax operator to make $\bar{\mathbf{y}}^m$ a distribution. $\bar{\mathbf{y}}^m$ is a K-dimensional

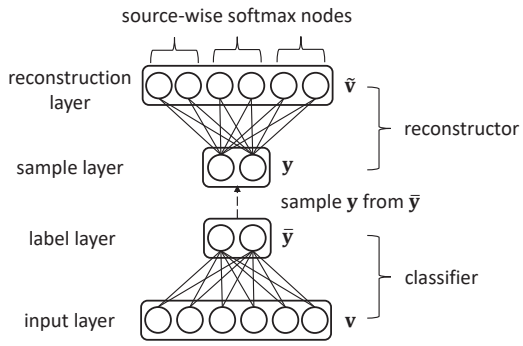


Figure 2: The architecture of LAA-B. Here the number of categories $K = 2$ for demonstration.

vector where K is the number of categories. We then sample $\mathbf{y}^m$ from the distribution $\bar{\mathbf{y}}^m$. For convenience in the network, $\mathbf{y}^m$ is one-hot encoded. The reconstructor is also modeled by a network which takes $\mathbf{y}^m$ as input and reconstructs $\tilde{\mathbf{v}}^m$ as

$$\tilde{\mathbf{v}}^m = \tilde{\sigma}(\mathbf{y}^m \mathbf{w}_p), \tag{3}$$

where weight matrix $\mathbf{w}_p$ corresponds to reconstructor parameter ϕ. $\tilde{\sigma}(\cdot)$ is the source-wise softmax operator. The operator applies a softmax operator only on nodes in same source block. By the treatment, reconstructed $\tilde{\mathbf{v}}^m$ has the same structure as input $\mathbf{v}^m$. Then the reconstruction term $\log p_\phi(v|y)$ in formula (1) is equivalent to the negative cross entropy between input $\mathbf{v}^m$ and reconstructed $\tilde{\mathbf{v}}^m$.

$$\begin{aligned}\log p_\phi(v|y) &= \log p_\phi(\mathbf{v}^m|\mathbf{y}^m) \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} \delta_{nk}^m v_{nk}^m \log \tilde{v}_{nk}^m,\end{aligned} \tag{4}$$

where element $\tilde{v}_{nk}^m$ of reconstructed vector $\tilde{\mathbf{v}}^m$ corresponds to v_{nk}^m of input vector $\mathbf{v}^m$, and δ_{nk}^m is the corresponding element of mask vector δ^m. The mask vector makes the calculation focus on observed labels only.

A potential problem of inferring labels by networks is that nodes are exchangeable. For a label vector $\bar{\mathbf{y}}^m$, its first element can either represent category 1 or category 2 if without any constraint. Therefore, we introduce a proper prior distribution for the KL divergence in formula (1) to constrain the representation of label vector $\bar{\mathbf{y}}^m$ (i.e. to make the first element always represent category 1 while the second element always represent category 2). A simple and reasonable choice is to use voting results

$$D_{KL}(q_\theta(y|v)||p(y)) = D_{KL}(\bar{\mathbf{y}}^m||\mathbf{r}^m), \tag{5}$$

where $\mathbf{r}^m$ is a vector of voting distribution from $\mathbf{v}^m$ where its k-th element $r_k^m = \frac{\sum_{n=1}^{N} v_{nk}^m}{\sum_{k=1}^{K}\sum_{n=1}^{N} v_{nk}^m}$. $\mathbf{r}^m$ has fixed positions for categories that constrain the representation of label vectors and solves the problem of node exchangeability.

Some real-world datasets are sparse, where one object receives labels from only a few sources or one source only labels a few objects. In such cases, we introduce l_1-norm for weight matrices in the classifier and the reconstructor

$$L_s = ||\mathbf{w}_p||_1 + ||\mathbf{w}_q||_1. \tag{6}$$

Indistinctive elements in weight matrices are pushed to zero to reduce noise from sources which label only a few objects.

Taking formulas (4), (5), and (6) into (1) and summing over all objects, LAA-B has the loss function

$$L_B(\{\mathbf{v}^m\}) = -\sum_{m=1}^{M}\Big(\mathbb{E}_{q_\theta(\mathbf{y}^m|\mathbf{v}^m)}\log p_\phi(\mathbf{v}^m|\mathbf{y}^m) - \lambda_{kl}D_{KL}(\bar{\mathbf{y}}^m||\mathbf{r}^m)\Big) + \lambda_s L_s, \quad (7)$$

where $\{\mathbf{v}^m\}$ denotes the set of all input vectors. λ_{kl} and λ_s are constraint strength. Note that we regard the KL divergence term as a regularizer, and giving it a small λ_{kl} achieves good performance in practice. Model parameters are learned by minimizing the loss. When the model is well trained, true label $\tilde{y}_m$ can be simply predicted from $\bar{\mathbf{y}}^m$ by choosing the category with maximum probability: $\tilde{y}_m = \arg\max_k \bar{y}_k^m$.

3.5 Relationship with Weighted Majority Voting

We demonstrate the intuition of inferring labels of LAA-B from the view of extended weighted majority voting. That also explains what is learned in the weight matrix of the classifier. Here we change the notation of weight matrix $\mathbf{w}_q$ a little bit. Let w_{ij}^n denote the weight from the i-th element in source block n of an input vector to the j-th element of a label vector, $i, j \in \{1, ..., K\}$ and $n \in \{1, ..., N\}$. Weights corresponding to source block n constitute a weight block.

We write out the expression of label vector $\bar{\mathbf{y}}^m$ with the interaction between n-th blocks in the input vector and the weight matrix (the number of categories is set as 2)

$$[\bar{y}_1^m, \bar{y}_2^m] = \sigma([\cdots, v_{n1}^m, v_{n2}^m, \cdots]\begin{bmatrix}\vdots & \vdots\\ w_{11}^n & w_{12}^n\\ w_{21}^n & w_{22}^n\\ \vdots & \vdots\end{bmatrix}). \quad (8)$$

The expression extends weighted majority voting [Li *et al.*, 2014] which only assigns one weight for each source. A weight block for the corresponding source has K^2 weights. w_{ij}^n is the weight from source labeled category i to inferred category j. A positively larger weight assigns more contribution from the corresponding source label to the inferred category. Therefore a weight block describes labeling credibility of the corresponding source. Note that a weight block with large diagonal weights represents a credible source which usually gives correct labels.

3.6 Object Ambiguity

Based on LAA-B, we can introduce more factors to further improve the inference performance. Here we introduce object ambiguity. An ambiguous object usually contains conflicting or little labeling information, that may produce large noise for learning. By contrast, an unambiguous object has clean and sufficient labeling information. One may have an easy understanding of object ambiguity by referring to object difficulty [Bachrach *et al.*, 2012].

A model is expected to put more efforts to correctly label and reconstruct an unambiguous object than an ambiguous one. To achieve this goal, we introduce a scalar z^m for each object o_m and combine it into the classifier and the reconstructor respectively. A large z^m indicates an object is unambiguous while a small z^m indicates an object is ambiguous.

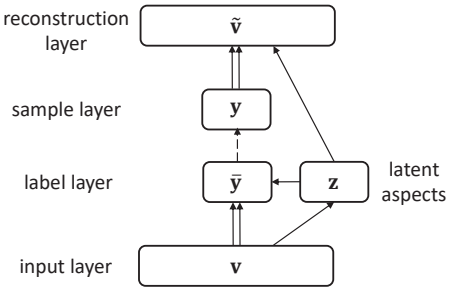


Figure 3: The architecture of LAA-L. (Single) arrows represent a function relationship in the network (usually with corresponding weight matrices). The dashed arrow (from $\bar{\mathbf{y}}$ to $\mathbf{y}$) indicates sampling. Solid paired arrows represent weight matrices corresponding with latent aspects.

$$\bar{\mathbf{y}}^m = \sigma(z^m\mathbf{v}^m\mathbf{w}_q), \quad (9)$$

$$\tilde{\mathbf{v}}^m = \tilde{\sigma}(z^m\mathbf{y}^m\mathbf{w}_p). \quad (10)$$

We can see a larger z^m results in more heterogeneous distribution after the softmax operator. Heterogeneous distribution leads to large loss if the object is not well reconstructed, that forces the model to improve the reconstruction quality. Since the input vector $\mathbf{v}^m$ contains all object information, z^m can be modeled based on it:

$$z^m = \tau(\mathbf{v}^m\mathbf{w}_o), \quad (11)$$

where $\mathbf{w}_o$ is the network weight which is learned in the training process. τ is the softplus activation function to ensure that z^m is positive. This model is called LAA-O (LAA with **O**bject ambiguity).

3.7 Latent Aspects

Further extension for LAA-O is to introduce latent aspects. One object may have more than one latent aspects, such as colors and shapes of flowers. One source may be good at classifying flowers by shapes, but not that good by colors. The performance of a model can be improved by distinguishing source credibilities under different aspects.

Suppose one object o_m has I latent aspects which is denoted by an I-dimensional latent aspect vector $\mathbf{z}^m$. Its element z_i^m indicates the weight of the i-th latent aspect. For the i-th latent aspect, there are corresponding weight matrices $\mathbf{w}_q^i$ and $\mathbf{w}_p^i$ for the classifier and the reconstructor respectively to represent source credibility under that aspect. Label vector $\bar{\mathbf{y}}^m$ and reconstructed vector $\tilde{\mathbf{v}}^m$ are obtained by summing over all aspects.

$$\bar{\mathbf{y}}^m = \sigma(\sum_{i=1}^{I} z_i^m(\mathbf{v}^m\mathbf{w}_q^i)), \quad (12)$$

$$\tilde{\mathbf{v}}^m = \tilde{\sigma}(\sum_{i}^{I} z_i^m(\mathbf{y}^m\mathbf{w}_p^i)). \quad (13)$$

Table 1: Accuracy Comparison on Real-world Datasets

Algorithm	Bluebirds	Flowers	Web Search
MV	0.7593	0.8000	0.7310
TruthFinder	0.7593	0.8050	0.7867
CATD	0.7685	0.8400	0.7806
DARE	0.7778	0.8100	0.8240
DS	0.8981	0.8700	0.8308
BCC	0.8981	0.8700	0.8562
CrowdSVM	0.8981	0.8650	0.9058
LAA-B	0.8889	0.8700	0.8971
LAA-O	0.9074	0.8800	**0.9118**
LAA-L	**0.9259**	**0.9000**	0.9107

The latent aspect vector $\mathbf{z}^m$ is produced from $\mathbf{v}^m$.

$$\mathbf{z}^m = \tau(\mathbf{v}^m \mathbf{w}_l). \tag{14}$$

The model is named LAA-L (LAA with **L**atent aspects). Figure 3 gives its architecture. We can see LAA-O is a special case of LAA-L where the number of latent aspects is 1.

4 Experiments

Three real-world datasets are used in experiments. **Bluebirds** [Welinder *et al.*, 2010] consists of 108 bluebird pictures. There are 2 breeds among all the images, and each image is labeled by all 39 sources. **Flowers** [Tian and Zhu, 2015b] contains 200 flower pictures. Each source is asked whether the flower is a peach flower. 36 sources participate in the labeling task and contribute 2,366 binary labels in total. **Web Search** [Zhou *et al.*, 2012] contains 2,665 query-URL pairs. 177 sources are asked to rate each pair by 5 relativity levels. In total 15,567 labels are collected.

Inference accuracy is used as the measurement

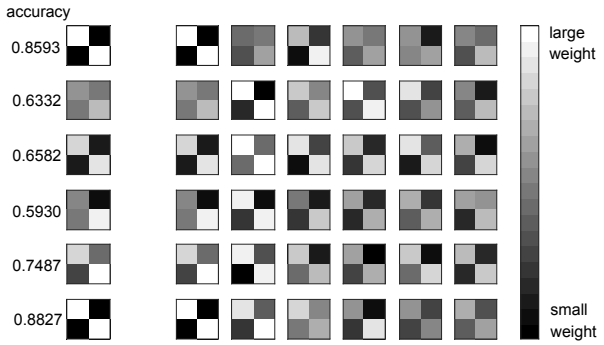
$$\text{accuracy} = \frac{\text{number of correctly inferred objects}}{\text{number of all objects}}. \tag{15}$$

We implement proposed models by TensorFlow[1] which offers GPU acceleration . Gradient descent is exploited to minimize the loss. A dataset is split into training set and validation set. Training process stops when the loss on the validation set begins to increase. We grid-search proper hyperparameters by choosing the combination which achieves the lowest loss on the validation set. Hyperparameters include learning rate $\eta \in [0.001, 0.1]$, constraint strength $\lambda_{kl} \in [0.0001, 0.1]$, and $\lambda_s \in [0.0001, 0.1]$. After determining the optimal hyperparameters, we train the model by using all data with chosen hyperparameters. For LAA-L, we set the number of latent aspects as 2 (further discussion is in Section 4.4). In this paper, we implement networks with one layer for the classifier and the reconstructor respectively. Though deep networks can be easily exploited, we find they do not further improve inference accuracy on the datasets due to data size.

4.1 Accuracy Comparison

Representative label aggregation methods are used as baselines. They are MV (majority voting), CATD (a weighted majority voting model which estimates the confidence interval of source credibility [Li *et al.*, 2014]), TruthFinder (the first trust propagation model [Yin *et al.*, 2008]), DARE (a generative model which models source credibility and object difficulty [Bachrach *et al.*, 2012]), DS (the first label aggregation model [Dawid and Skene, 1979]), BCC (a generative model using confusion matrix [Kim and Ghahramani, 2012]), CrowdSVM (a recent proposed method combining max margin majority voting and DS Model [Tian and Zhu, 2015a]). Proposed label-aware autoencoders LAA-B, LAA-O, and LAA-L are compared as well. The results of inference accuracy are illustrated in Table 1. We can see even the basic model LAA-B is competitive with the state-of-the-art methods. Note that LAA-B does not use knowledge about sources or objects. By introducing object ambiguity, LAA-O improves the inference accuracy significantly. LAA-L further improves the inference accuracy by exploiting latent aspects. The results show that proposed LAA has advantages on label aggregation compared with other methods.

[1] www.tensorflow.org



(a) Accuracy (b) Weight Matrix

Figure 4: Illustration of the weight matrix in LAA-B learned on the Flowers dataset. Each block corresponds to a source and has 2×2 weights. 20 gray levels are used to indicate weight values.

4.2 Source Credibility in Weight Matrix

Weight matrix of the classifier represents source credibility. To illustrate that, we take weight matrix $\mathbf{w}_p$ after training LAA-B on the Flowers dataset. There are 36 blocks corresponding to 36 sources and each block has 2×2 weights. We use 20 gray levels to color weights according to their values. White indicates a large weight while black indicates a small weight. Figure 4b illustrates the weight blocks. For each block, its diagonal weights are relatively large. That means the inference accuracy of most sources are better than random guessing. LAA-B distinguishes sources with high credibility from others by giving large diagonal weights. To see that, we illustrate in Figure 4a labeling accuracy for sources which correspond to the first block column. Labeling accuracy of a source is: the ratio of correctly labeled object number to the total labeled object number by the source. We can observe that sources with high labeling accuracy correspond to blocks with large diagonal weights. The observation shows networks in LAA have the capability to capture source credibility by learning the weight matrix.

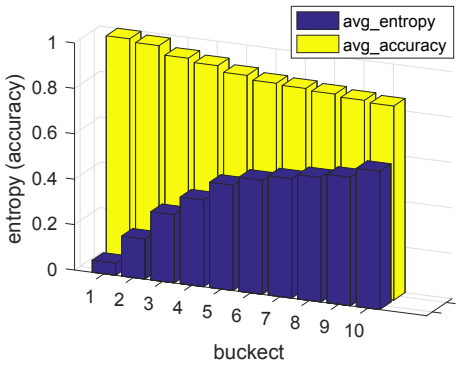


Figure 5: Illustration of the effect of object ambiguity. Objects are divided into 10 buckets according to their ambiguity. For objects in each bucket, a blue bar indicates average entropy of their inferred label vectors, while a yellow bar indicates average inference accuracy.

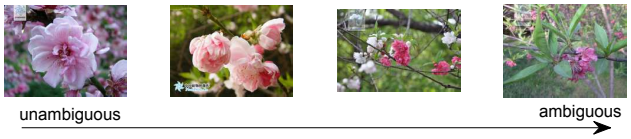


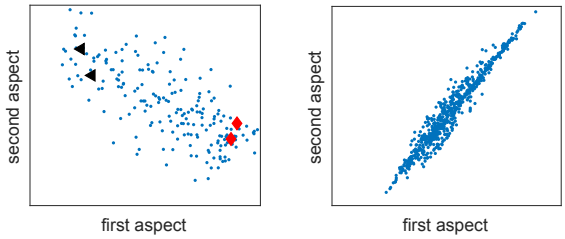
Figure 6: Four images of peach flowers from the Flowers dataset. They are arranged from unambiguous to ambiguous.

4.3 Effect of Object Ambiguity

In this subsection we show the effect of object ambiguity. After training LAA-O on the Web Search dataset, we sort objects by their ambiguity in ascending order and divide them into 10 buckets. The first bucket contains objects with the least ambiguity while the last bucket contains objects with the most ambiguity. For objects in each bucket, we calculate: 1. Average entropy of their inferred label vectors; 2. Average inference accuracy. The results are illustrated in Figure 5. Objects are given heterogeneous distribution of inferred labels (with small entropy) if the model treats them as unambiguous (blue bars). Those unambiguous objects usually lead to high inference accuracy (yellow bars). On the other hand, low accuracy is caused by objects with conflicting labeling information. LAA-O treats them as ambiguous and gives relatively balanced distribution of inferred labels (with large entropy) to reduce their effect in the learning process. That decreases the noise from those objects and improves the overall accuracy. To visualize object ambiguity, we show four images of peach flowers from the Flowers dataset and arrange them from unambiguous to ambiguous in Figure 6. The unambiguous peach flower is easy to recognize while the ambiguous one is not.

4.4 Effect of Latent Aspects

LAA-L is the extension for LAA-O by introducing more than one latent aspects. From Table 1, we observe that accuracy is significantly improved by LAA-L on the Bluebirds and the Flowers dataset, but not improved on the Web Search dataset. In Figure 7, we compare latent aspect vectors learned on Flowers and Web Search datasets to explain the reason. A latent aspect vector with 2 dimensionality can be illustrated as a point in a 2-D figure. On the Web Search dataset, two latent aspects show strong positive correlation (Figure 7b). Therefore they can be merged into one aspect without decreasing the inference accuracy. On the Flowers dataset, however, latent aspects have slight negative correlation (Figure 7a) which means different objects have different dominant aspects. That supports the necessity of introducing latent aspects. We show two typical flower images for each aspect in Figure 8. Images dominated by the first aspect are pink flowers, while images dominated by the second aspect are white flowers with special petal shape. Figure 7 also shows a method to determine a proper number of latent aspects. We can try to add one latent aspect at a time, and train LAA-L to see whether there is positive correlation between aspects. If two aspects do not have positive correlation, then the added aspect is effective. If two aspects have strong positive correlation, then adding the extra aspect is not necessary. Through experiments, we find that the best numbers of latent aspects for Bluebirds, Flowers, and Web Search datasets are 2, 2, and 1 respectively.



(a) Flowers (b) Web Search

Figure 7: Illustration of latent aspect vectors on Flowers and Web Search datasets. The X-axis indicates the weight of first aspect and the Y-axis indicates the weight of second aspect.

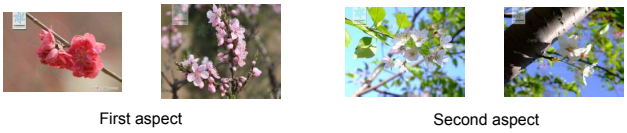


Figure 8: Illustration of two typical flower images for each aspect. Images dominated by the first aspect (corresponding to red diamonds in 7a) are pink flowers, while images dominated by the second aspect (corresponding to black triangles in 7a) are white flowers with special petal shape.

5 Conclusion

In this paper, we propose **L**abel-**A**ware **A**utoencoders (LAA) for aggregating crowd wisdoms. We exploit networks to implement the classifier and the reconstructor in LAA which have the potential to automatically learn underlying source labeling patterns. Object ambiguity and latent aspects are introduced to the basic model to further improve inference accuracy. Experiments on three real-world datasets show the advantages of proposed framework, where the basic model LAA-B is competitive with the state-of-the-art, LAA-O and LAA-L further improve inference accuracy significantly.

References

[Aydin *et al.*, 2014] Bahadir Ismail Aydin, Yavuz Selim Yilmaz, Yaliang Li, Qi Li, Jing Gao, and Murat Demirbas. Crowdsourcing for multiple-choice question answering. In *AAAI*, pages 2946–2953, 2014.

[Bachrach *et al.*, 2012] Yoram Bachrach, Thore Graepel, Tom Minka, and John Guiver. How to grade a test without knowing the answers—a bayesian graphical model for adaptive crowdsourcing and aptitude testing. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1183–1190, 2012.

[Dawid and Skene, 1979] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.

[De Winter *et al.*, 2015] JCF De Winter, Miltos Kyriakidis, Dimitra Dodou, and Riender Happee. Using crowdflower to study the relationship between self-reported violations and traffic accidents. *Procedia Manufacturing*, 3:2518–2525, 2015.

[Galland *et al.*, 2010] Alban Galland, Serge Abiteboul, Amélie Marian, and Pierre Senellart. Corroborating information from disagreeing views. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 131–140. ACM, 2010.

[Gomes *et al.*, 2011] Ryan G Gomes, Peter Welinder, Andreas Krause, and Pietro Perona. Crowdclustering. In *Advances in Neural Information Processing Systems*, pages 558–566, 2011.

[Ipeirotis, 2010] Panagiotis G Ipeirotis. Analyzing the amazon mechanical turk marketplace. *XRDS: Crossroads, The ACM Magazine for Students*, 17(2):16–21, 2010.

[Kim and Ghahramani, 2012] Hyun-Chul Kim and Zoubin Ghahramani. Bayesian classifier combination. In *AISTATS*, pages 619–627, 2012.

[Kingma and Welling, 2013] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[Li *et al.*, 2014] Qi Li, Yaliang Li, Jing Gao, Lu Su, Bo Zhao, Murat Demirbas, Wei Fan, and Jiawei Han. A confidence-aware approach for truth discovery on long-tail data. *Proceedings of the VLDB Endowment*, 8(4):425–436, 2014.

[Li *et al.*, 2016] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. A survey on truth discovery. *ACM SIGKDD Explorations Newsletter*, 17(2):1–16, 2016.

[Metrikov *et al.*, 2015] Pavel Metrikov, Virgil Pavlu, and Javed A Aslam. Aggregation of crowdsourced ordinal assessments and integration with learning to rank: A latent trait model. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1391–1400. ACM, 2015.

[Pasternack and Roth, 2010] Jeff Pasternack and Dan Roth. Knowing what to believe (when you already know something). In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 877–885. Association for Computational Linguistics, 2010.

[Qi *et al.*, 2013] Guo-Jun Qi, Charu C Aggarwal, Jiawei Han, and Thomas Huang. Mining collective intelligence in diverse groups. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1041–1052. International World Wide Web Conferences Steering Committee, 2013.

[Simpson *et al.*, 2013] Edwin Simpson, Stephen Roberts, Ioannis Psorakis, and Arfon Smith. Dynamic bayesian combination of multiple imperfect classifiers. In *Decision Making and Imperfection*, pages 1–35. Springer, 2013.

[Tian and Zhu, 2015a] Tian Tian and Jun Zhu. Max-margin majority voting for learning from crowds. In *Advances in Neural Information Processing Systems*, pages 1621–1629, 2015.

[Tian and Zhu, 2015b] Tian Tian and Jun Zhu. Uncovering the latent structures of crowd labeling. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 392–404. Springer, 2015.

[Vincent *et al.*, 2008] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[Vincent *et al.*, 2010] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[Welinder *et al.*, 2010] Peter Welinder, Steve Branson, Pietro Perona, and Serge J Belongie. The multidimensional wisdom of crowds. In *Advances in neural information processing systems*, pages 2424–2432, 2010.

[Whitehill *et al.*, 2009] 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*, pages 2035–2043, 2009.

[Yin *et al.*, 2008] Xiaoxin Yin, Jiawei Han, and Philip S Yu. Truth discovery with multiple conflicting information providers on the web. *Knowledge and Data Engineering, IEEE Transactions on*, 20(6):796–808, 2008.

[Zhi *et al.*, 2015] Shi Zhi, Bo Zhao, Wenzhu Tong, Jing Gao, Dian Yu, Heng Ji, and Jiawei Han. Modeling truth existence in truth discovery. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1543–1552. ACM, 2015.

[Zhou *et al.*, 2012] Dengyong Zhou, Sumit Basu, Yi Mao, and John C Platt. Learning from the wisdom of crowds by minimax entropy. In *Advances in Neural Information Processing Systems*, pages 2195–2203, 2012.