Learning Attributes from the Crowdsourced Relative Labels

Tian Tian[†], **Ning Chen**[‡], **Jun Zhu**[†]
[†]Dept. of Comp. Sci. & Tech., CBICR Center, State Key Lab for Intell. Tech. & Systems
[‡]MOE Key lab of Bioinformatics, Bioinformatics Division and Center for Synthetic & Systems Biology
TNList, Tsinghua University, Beijing, China
{tiant16@mails, ningchen@, dcszj@}tsinghua.edu.cn

Abstract

Finding semantic attributes to describe related concepts is typically a hard problem. The commonly used attributes in most fields are designed by domain experts, which is expensive and time-consuming. In this paper we propose an efficient method to learn human comprehensible attributes with crowdsourcing. We first design an analogical interface to collect relative labels from the crowds. Then we propose a hierarchical Bayesian model, as well as an efficient initialization strategy, to aggregate labels and extract concise attributes. Our experimental results demonstrate promise on discovering diverse and convincing attributes, which significantly improve the performance of the challenging zero-shot learning tasks.

Introduction

Extracting concise attributes and then drawing a conclusion is a usual pattern when humans make decisions (Hwang and Yoon 1981). For example, when we judge whether a research paper is good or not, instead of making a decision directly, we usually ask whether this paper is novel, possesses good technique quality or has a potential impact to the literature, and then reach a decision based on the answers of these sub-questions. Here *novelty*, *technique quality* and *potential impact* are three *attributes* extracted from the raw textual data to judge the quality of a paper. They can help us to solve the comprehensive judgment. In the machine learning literature, compared with raw features such as pixel representation of images, attributes are usually more concise and easily interpretable (Farhadi et al. 2009). Moreover, due to the extensive knowledge background of humans, attributes naturally contain cross domain information, some of which cannot be deduced directly from the data themselves. For example, humans can extract behavior attributes of an animal only from static pictures of it. Thus attributes are suitable for transferring knowledge between tasks and classifying objects. The benefits have been shown by works in the zero-shot learning literature (Jayaraman and Grauman 2014; Lampert, Nickisch, and Harmeling 2009; Norouzi et al. 2014; Palatucci et al. 2009; Parikh and Grauman 2011b; Romera-Paredes and Torr 2015).

In this paper, we ask a new question: *can we learn high quality attributes from the crowds?* This question arises from practice since finding attributes to describe related concepts is typically a very hard task. Although humans use attributes to make decisions in every minute, it is usually a subconscious process and hard to be characterized. Moreover, due to human perception variations, different people may extract different attributes from a same task, so the responses are inherently diverse. As a result of these difficulties, the commonly used attributes in most fields are designed by experts, which is an expensive and time-consuming process.

Fortunately, the rise of crowdsourcing provides a new way to collect information from a group of people fast and cheaply. Systems like Amazon Mechanical Turk and CrowdFlower have been widely used by machine learning researchers to label large-scale datasets (Deng et al. 2009; Welinder et al. 2010; Kovashka et al. 2016), cluster items (Gomes et al. 2011) and even build classifiers (Cheng and Bernstein 2015). To learn attributes by crowdsourcing, we first design an analogical interface to collect human opinions, which is efficient, and can lead people to give expected results. Then the human responses are represented by a collection of relative labels. Next we build a hierarchical Bayesian aggregating model to ensemble the results collected from different people and tasks, which is robust to the crowdsourcing noise. After the above process, we test the efficacy of our method, and the results demonstrate promise on discovering diverse and convincing attributes, whose qualities are further proved by significantly improving zero-shot learning performances compared with the expert-designed attributes.

Analogical Encoding of Attributes

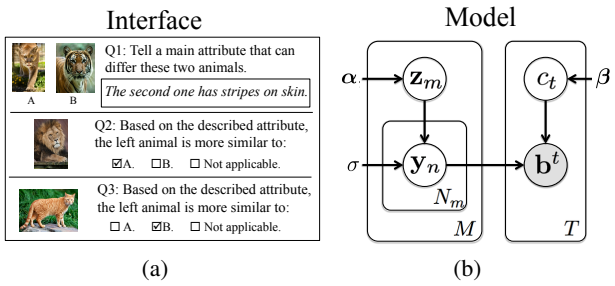
The first step is to collect information from humans. Suppose we want to collect main visual attributes[1] of animals, a naive approach is to directly show pictures of each animal, then ask humans to answer some straightforward questions, such as:

"what is the main visual characteristic of this animal?"

However, in practice we found this approach often results in a poor response due to several reasons: (1) the question is not well defined, thus it can be interpreted in many different ways and has too many possible answers. Moreover, usually people tend to give apparent attributes. For example, when we show annotators flower pictures, 94% responses focus on the

Copyright © 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1]In this work, we define the attributes at the category level, which means items in a same category should share same attribute values.



(a) (b)

Figure 1: (a) Demo of our labeling interface with 4 categories and 3 questions. (b) Graphical model for aggregating.

color differences among species while ignoring other types of attributes; (2) the textual descriptions in the answers are hard to aggregate, since people may describe a same thing with completely different words; (3) after asking this question, an additional labeling step is required to collect the values of this attribute for other categories.

To tackle these difficulties, we propose to use an *analogical interface* on this problem. Analogy is argued to be the most important way to learn new things for humans (Gentner, Loewenstein, and Thompson 2003). People usually focus on surface-level features when considering single examples, but discover deeper structural characteristics when comparing examples. In our case, a labeling task is divided into two parts. First, we show people a pair of categories. For example, we let people see pictures of a tiger and a lion at the same time, then ask the first question:

"what is the main visual attribute that differentiates between the tiger and the lion?"

This analogical question could be better than the naive approach since it has fewer possible answers. During labeling, the response for this question can ensure the annotator to focus on one same attribute through this task.

After this descriptive question, we then show a list of animal pictures, each from a different animal category except the two displayed before. Along with them we ask questions:

"based on the described attribute, is this animal more similar to tiger or lion?"

Finally, we collect the answers for these questions. This approach is more appropriate than previous work (Parikh and Grauman 2011a; Law et al. 2011) since it can give answers to questions from two different aspects: (1) know what this attribute is; (2) know the values of this attribute for each different animal category. An extra benefit is that by viewing multiple categories in a single task annotators can have a better global understanding of the dataset. For example, if all the animals here are terrestrial, people will hopefully avoid trivial attributes such as whether this animal lives on land. Fig. 1(a) shows a demo for this analogical interface.

Now we encode the results of each task. Suppose the dataset contains N items and M categories, and we design T tasks to collect labels. Each task t starts with a descriptive question about the two items j and k from the analogical categories. Then we show annotators items from other categories and solicit similarity labels based on the reference attribute. $b^t_{i,j,k}$ denotes the response for item i. Without loss of generality, we provide three options for the answer of each question: positive value $b^t_{i,j,k} = 1$ means that item i is more similar to the item j; negative value $b^t_{i,j,k} = -1$ means that it is more similar to the item k; and a neutral value $b^t_{i,j,k} = 0$ means that this item does not possess the reference attribute.

If we concatenate the response values from a same task, and set the values of the two analogical examples as 1 and -1, then the answers of this task can be encoded by a vector $\boldsymbol{h} \in \{1, 0, -1\}^M$. This vector can concisely deliver the characteristic of the attribute that the annotator used for this task. We call it an *attribute signature*, and we use this, instead of the textual description, to determine a crowdsourced attribute.

Aggregation via a Bayesian Model

Due to the problem property and crowdsourcing nature, the human responses could be noisy and diverse, and the attribute signatures directly obtained from the annotators could be duplicated. So our next step is to aggregate the answers collected from different people and different tasks. To accomplish this goal, we propose a hierarchical Bayesian model to describe the labeling procedure. This model takes the observations $\boldsymbol{b}$ as the input, and its outputs are K independent crowdsourced attributes. The model parameters and the posterior distribution are learned by a variational EM algorithm, and we also present an efficient nonparametric initialization strategy.

Generative Model

The graphical model is illustrated in Fig. 1(b) in a plate notation. According to our model, each category has an inherent signature $\boldsymbol{z}$ to determine its attribute values, and each item has its own item signature $\boldsymbol{y}$ due to the variation among instances. For each labeling task, the annotator first selects an attribute $\boldsymbol{c}$, then gives the relative labels $\boldsymbol{b}$ according to the corresponding item signatures. Below we show the details.

Category Level. We assume that there exists K independent attributes to describe the relationship among different categories. In a dataset of M animal species, the attributes may include black or white, big or small, living on land or in water, etc. The values of these attributes can show the characteristics of each category. Similar to the attribute signature, we denote the vector that consists of attribute values for a same category as a *category signature*. For category m, its category signature is $\boldsymbol{z}_m \in \{-1, 0, 1\}^K$. Each element $z_{m,k} = 1$ or -1 means that category m has positive or negative value on this attribute, respectively, and $z_{m,k} = 0$ means that category m does not possess this attribute. From a global perspective, the attribute signatures and the category signatures correspond, respectively, to the rows and the columns of the attribute-category value matrix. They are actually two orthogonal views of a same concept. Since the signature entries are discrete random variables, we put multinomial priors on them as

$$p(z_{m,k}) \quad = \quad \text{Mult}(z_{m,k}|\boldsymbol{\alpha}), \tag{1}$$

where $\boldsymbol{\alpha}$ is a parameter to control the priors.

Item Level. Let $l_n \in [M]$ ($n \in [N]$) denotes the category that item n belongs in, where $[M]$ means the set $\{1, 2, \cdots, M\}$. Ideally all the items in a same category should share the same attribute values. However, due to the variation among instances, some values may show differences. To model this variability, we denote the attribute values of each item as its *item signature*, and for item n it is denoted by $\boldsymbol{y}_n \in \mathbb{R}^K$. These values are defined in the real number domain so that the variations are represented by their distances to 0, 1 or -1. Since item signatures are originated from the attribute signatures, we naturally assume that each item signature $\boldsymbol{y}_n$ is generated from a normal distribution,

$$p(\boldsymbol{y}_n|\boldsymbol{\mu}_n) \quad = \quad \mathcal{N}(\boldsymbol{y}_n|\boldsymbol{\mu}_n, \sigma^2\boldsymbol{I}), \tag{2}$$

where $\boldsymbol{\mu}_n = \boldsymbol{z}_{l_n}$ is its corresponding category's signature. The parameter σ controls the variability.

Task Level. For each labeling task t, $c_t \in [K]$ is a random variable to denote the attribute that is selected by the annotator. This attribute is then used as the judging reference for answering further questions. The membership variable $\boldsymbol{c}$ also has a multinomial prior as

$$p(c_t) \quad = \quad \text{Mult}(c_t|\boldsymbol{\beta}), \tag{3}$$

where $\boldsymbol{\beta}$ is a control parameter.

Likelihood. To give the likelihood $p(\boldsymbol{b}|\boldsymbol{y}, \boldsymbol{c})$, we assume that the similarity with the reference attribute c_t between item i and item j is only related to the c_t-th elements of their item signatures $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$. Then given the distribution of $\boldsymbol{y}$, we can define the likelihood for an observation $b^t_{i,j,k} = 1$ by a softmax function as

$$p(b^t_{i,j,k}=1|\boldsymbol{y}_i, \boldsymbol{y}_j, \boldsymbol{y}_k, c_t) = \frac{e^{-D^t_{i,j}}}{e^{-D^t_{i,j}}+e^{-D^t_{i,k}}+e^{-D^t_{i,0}}}, \tag{4}$$

where $D^t_{i,j}$ is defined as the Jensen-Shannon divergence between the distributions of y_{i,c_t} and y_{j,c_t}. [2] For $b^t_{i,j,k} = -1$, we can switch j and k to fit above definition. For $b^t_{i,j,k} = 0$, we introduce $D^t_{i,0}$ [3] and then define the likelihood as

$$p(b^t_{i,j,k}=0|\boldsymbol{y}_i, \boldsymbol{y}_j, \boldsymbol{y}_k, c_t) = \frac{e^{-D^t_{i,0}}}{e^{-D^t_{i,j}}+e^{-D^t_{i,k}}+e^{-D^t_{i,0}}}. \tag{5}$$

So now with the above definitions, the generative process according to this hierarchical model is

1. for each m and k, sample $z_{m,k}$ from $z_{m,k} \sim \text{Mult}(\boldsymbol{\alpha})$.
2. for each n, sample its signature from $\boldsymbol{y}_n \sim \mathcal{N}(\boldsymbol{\mu}_n, \sigma^2\boldsymbol{I})$.
3. for each t, sample its membership from $c_t \sim \text{Mult}(\boldsymbol{\beta})$.
4. for each relative question (i, j, k) in task t, sample $b^t_{i,j,k}$ from likelihood $b^t_{i,j,k} \sim p(b \mid \boldsymbol{y}_i, \boldsymbol{y}_j, \boldsymbol{y}_k, c_t)$.

We do not explicitly consider the annotators' abilities in this model, since including more factors may raise the risk of over-fitting when training with a limited amount of labels.

[2] We denote $P = p(y_{i,c_t})$ and $Q = p(y_{j,c_t})$, then $D^t_{i,j} = \text{JS}(P\|Q) = [\text{KL}(P\|H)+\text{KL}(Q\|H)]/2$, where $H = (P+Q)/2$.

[3] We denote $U = \mathcal{N}(0, \sigma^2)$, then $D^t_{i,0} = \text{JS}(P\|U)$.

Posterior Inference and Parameter Estimation

Now we briefly discuss how to perform the posterior inference and parameter estimation. When the parameters $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma\}$ are known, given the priors and likelihood above, the exact posterior can be computed by

$$p(\boldsymbol{z}, \boldsymbol{y}, \boldsymbol{c}|\boldsymbol{b}, \boldsymbol{\theta}) = \prod_m p(\boldsymbol{z}_m) \prod_n p(\boldsymbol{y}_n|\boldsymbol{\mu}_n, \sigma)$$
$$\prod_t \left[p(c_t|\boldsymbol{\beta}) \prod_{(i,j,k)} p(b^t_{i,j,k}|\boldsymbol{y}_i, \boldsymbol{y}_j, \boldsymbol{y}_k, c_t) \right] / p(\boldsymbol{b}|\boldsymbol{\theta}). \tag{6}$$

Since it is hard to compute the evidence, exact posterior inference is intractable. To tackle this problem, we introduce a variational distribution to approximate the optimal posterior. Under the mean-field assumption (Wainwright and Jordan 2008), the variational distribution can be factorized as

$$q(\boldsymbol{c}, \boldsymbol{z}, \boldsymbol{y}) = \prod_t q(c_t|\gamma_t) \prod_{m,k} q(z_{m,k}|\phi_{m,k}) \prod_n q(\boldsymbol{y}_n|\boldsymbol{\psi}_n, \tau_n),$$

where the parametric distributions are in the following form:

$$\begin{aligned} q(c_t|\gamma_t) &= \text{Mult}(c_t; \gamma_t), \\ q(z_{m,k}|\phi_{m,k}) &= \text{Mult}(z_{m,k}; \phi_{m,k}), \\ q(\boldsymbol{y}_n|\boldsymbol{\psi}_n, \tau_n) &= \mathcal{N}(\boldsymbol{y}_n; \boldsymbol{\psi}_n, \tau_n^2\boldsymbol{I}). \end{aligned} \tag{7}$$

Then we introduce a tractable evidence lower bound (ELBO). With the above variational distribution $q(\boldsymbol{c}, \boldsymbol{z}, \boldsymbol{y})$, the marginal log-likelihood can be bounded by

$$\begin{aligned} \log p(\boldsymbol{b}|\boldsymbol{\theta}) &= \mathcal{L}(\boldsymbol{\gamma}, \boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\tau}) + \text{KL}\left(q(\boldsymbol{c}, \boldsymbol{y}, \boldsymbol{z}) \| p(\boldsymbol{c}, \boldsymbol{y}, \boldsymbol{z}|\boldsymbol{b}, \boldsymbol{\theta})\right) \\ &\geq \mathcal{L}(\boldsymbol{\gamma}, \boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\tau}) \\ &= \mathbb{E}_q[\log p(\boldsymbol{c}, \boldsymbol{z}, \boldsymbol{y}, \boldsymbol{b}|\boldsymbol{\theta})] - \mathbb{E}_q[\log q(\boldsymbol{c}, \boldsymbol{z}, \boldsymbol{y})]. \end{aligned} \tag{8}$$

We can optimize the variational parameters to maximize this tractable ELBO by a coordinate ascent algorithm, where we update $\boldsymbol{\gamma}$, $\boldsymbol{\phi}$, $\boldsymbol{\psi}$ and $\boldsymbol{\tau}$ iteratively. Our likelihood $p(\boldsymbol{b}|\boldsymbol{y}, \boldsymbol{c})$ is conditioned on the distribution of $\boldsymbol{y}$. So when calculating the expected likelihood, we can directly plug in the variational distribution $q(\boldsymbol{y})$ for the ease of calculation (Karaletsos et al. 2016).

To estimate parameters, we can also maximize the ELBO rather than the marginal log-likelihood with respect to the parameters $\boldsymbol{\theta}$. So the complete updating is actually a variational EM (Neal and Hinton 1999) procedure. We perform variational inference in the E-step while perform parameter estimation in the M-step. These two steps alternate in an iteration. For brevity, we put the derivations in Appendix A[4].

Nonparametric Initialization

Since the learning process involves the non-convex optimization, the results can be sensitive to the initial values. The amount of attributes K can also influence the results. To deal with these issues, we propose a strategy to initialize the attribute signatures and find a reasonable K.

[4] Please find appendix at:
http://ml.cs.tsinghua.edu.cn/%7Etian/p/CrowdAttributesSupp.pdf

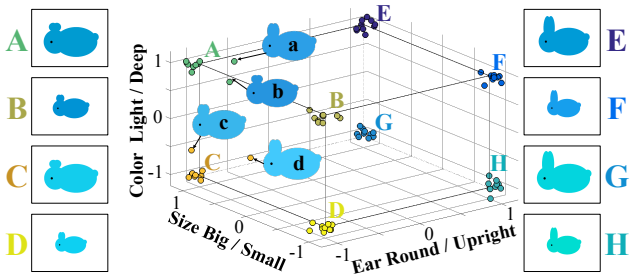


Figure 2: Pictures A-H show examples of the 8 rabbit categories. The middle part shows the signatures learned from the rabbits dataset. Each axis denotes one crowdsourced attribute. Each point denotes one rabbit picture. Different colors denote different rabbit categories. (Best viewed in color).

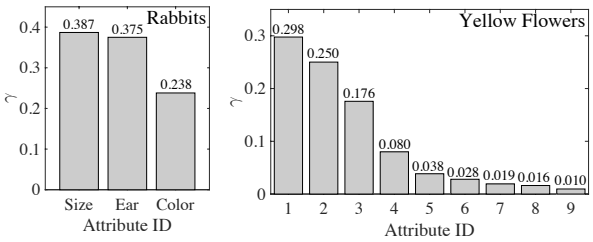


Figure 3: Empirical distribution of the membership variable c for (a) the rabbits and (b) the yellow flowers datasets.

The main idea of this strategy is to aggregate similar attributes which are directly obtained from the annotators. To this end, we define the similarities between the attribute signatures. Specifically, for signatures $\boldsymbol{h}_s$ and $\boldsymbol{h}_t$ of tasks s and t, the similarity between them is defined as

$$f(\boldsymbol{h}_s, \boldsymbol{h}_t) = |\boldsymbol{h}_s^\top \boldsymbol{h}_t|. \tag{9}$$

We assume that the attribute signatures should be originated from K independent attributes, so we partition them into clusters, and then the signatures correspond to a same attribute could be put together.

Since the amount of clusters K is unknown in advance, algorithms such as K-means cannot be used here. Inspired by the nonparametric clustering algorithm DP-Means (Kulis and Jordan 2012), we propose an optimization problem:

$$\max_{\mathcal{R}, \{l_k\}} \sum_{k=1}^{K} \sum_{t \in l_k} f(\boldsymbol{h}_t, \boldsymbol{r}_k) - \lambda K, \tag{10}$$

where $\mathcal{R} = \{\boldsymbol{r}_k\}_{k=1}^M$ denotes the collection of K independent attribute signatures, and $l_k = \{t | c_t = k\}$ denotes the indexes of all tasks whose membership variable equals to k. $\lambda = (1-\rho)M$ is a similarity threshold, and $\rho \in [0, 1]$ is a relaxation factor. According to this formula, we can prove that when the maximum similarity between a signature $\boldsymbol{h}_t$ and all existing $\boldsymbol{r}_k$ is smaller than ρM, it is better to create a new cluster with $\boldsymbol{h}_t$ as its associated attribute signature.

Problem (10) can be optimized by iteratively updating the partitions $\{l_k\}$ and the signatures $\mathcal{R}$. When updating the partitions, we add or delete clusters based on the similarity conditions in real time, so we can learn a reasonable K. Finally, these results are used to initialize the Bayesian aggregating model. We put more details in Appendix B.

Empirical Results

To demonstrate the efficacy of our methods, we conduct experiments on three image datasets. Below are the details.

Experiment Setups

The rabbits dataset is a synthetic dataset, which contains pictures with controlled attributes. The yellow flowers and the animals datasets are composed of natural scene pictures from the common visual datasets (Nilsback and Zisserman 2008; Lampert, Nickisch, and Harmeling 2009). We select a subset of the categories for the ease of demonstration, and also to save time and costs during labeling. All tasks are posted on the Amazon Mechanical Turk (AMT) platform.

During label aggregation, We set $\rho = 0.6$ for the rabbits and the yellow flowers datasets based on our experience. For the animals dataset, the results on multiple values of ρ are shown. The model parameters $\boldsymbol{\theta}$ are uniformly initialized and then updated from the data.

Results on the Rabbits Dataset

We generate a series of cartoon rabbit pictures, each of which is composed of three parts: head, ears and body. These rabbits belong to 8 categories, which are decided by three visual attributes: (1) the rabbit is in deep blue or light blue; (2) the rabbit is bigger or smaller in size; (3) the rabbit has long upright ears or short round ears. Each category contains 10 rabbits, their attributes' control parameters are randomly sampled from normal distributions with category specific mean values. To increase the potential diversity of the human responses, the mean values of different categories are close. Fig. 2 shows the examples of these categories.

In each of the labeling tasks, we first show the annotator two rabbit pictures from the two analogical rabbit categories respectively, and then we show six rabbits from the other six categories and solicit the similarities. There exist $8 \times 7/2 = 28$ different arrangements for the 8 categories in a task. For each arrangement the representative pictures for categories are randomly chosen for 6 times, so we post 168 tasks in total. For each task we pay $0.04 + 0.01$ dollars (0.04 dollars to the annotator and 0.01 dollars to the platform).

Item Signatures. We visualize our results in a 3-D coordinate system in Fig. 2. Each axis denotes one crowdsourced attribute; each point denotes one rabbit signature; and different colors denote different rabbit categories. In this figure the signatures roughly fall into 8 clusters, which appear near the 8 vertexes of a cube. This cube is centered at the coordinate origin, and its edges are parallel to the axes. It shows that our model learned 3 independent attributes, and the results successfully match with the three rabbit attributes in our design. For example, cluster A for big, deep blue rabbits with round ears; cluster H for small, light blue rabbits with upright ears, etc. Besides, some rabbit signatures fall outside the clusters.

No.	Pos Examples	Neg Examples	Signatures	Worker Descriptions
1				Solid-color versus variegated petals
2				Shape of flower
3				Shape of petals
4				Number of petal rows
5				Converge of petals
6				Open petals vs closed
7				Visible center vs. non visible
8				Color of the petals
9				Direction of petals curl

Figure 4: Crowdsourced attributes learned from the yellow flower dataset (Best viewed in color).

We find that they are caused by abnormal attribute values. For example, the rabbit *a* has an ear type between round and upright; and the rabbit *b* has a medium size, etc.

Priorities of the Attributes. It is a common phenomenon that different attributes have different priorities. In Fig. 3(a), we present the statistics of the membership variables $\boldsymbol{c}$ to explore the human behaviors when they are selecting attributes. It shows that the attributes about the body size and the ear shape are more likely to be noticed, while the attribute about color is less used. It is probably due to the fact that the difference between the two color types are less obvious than that between the body sizes and that between the ear shapes.

Results on the Yellow Flowers Dataset

The yellow flowers dataset is a subset of the Oxford flower dataset (Nilsback and Zisserman 2008) with 13 yellow flower species. Each category contains 40 flower pictures. When building the tasks, there are 78 different category arrangements, and for each of them we build 4 tasks with randomly selected representative pictures. So we post 312 AMT tasks in total, and each of them costs $0.07 + 0.02$ dollars.

Attribute Signatures. After removing the results that are compatible with less than 4 categories, we visualize 9 crowdsourced flower attributes in Fig. 4. Each of them is demonstrated by a row of the table. Specifically, we present four most representative flower pictures for each attribute, including two positive examples and two negative examples with the most extreme item signature values. For instance, the positive examples for attribute No. 1 come from the marigold and the globe flower, which usually have solid color petals. The negative examples come from the blanket flower and the canna lily, which usually have variegated petals. So the semantic meaning of this attribute could be whether the petals of this flower have solid color. To make the meaning more conspicuous, we find the task that is most likely to use this attribute as the judging reference, and show the response for the descriptive question of this task. For attribute No. 1, annotator describes it as about *solid color versus variegated petals*, which confirms what we suspected.

We also show the bar diagrams of the attribute signatures in the table to give overviews of the attributes. Each bar relates to one flower category. Bars above the line denote positive values, and bars below the line denote negative values. Empty positions relate to the incompatible categories. The relationship between the bar positions and the flower categories are shown in the signature legend in Appendix D.

Priorities of the Attributes. Each crowdsourced attribute describes an independent aspect of the flower appearance, including the shape of petals, color, structure, etc. Fig. 3(b) shows the empirical distribution of the membership variable. Specifically, the color purity and the flower shape are arguably the two most preferred attributes. Then the two attributes about the shape and the amount of the petals are also commonly used. The other five attributes are significantly less frequently used, and these attributes are compatible with fewer categories according to their signatures. However, their meanings are usually more interesting, such as the rarely noticed center size attribute No. 7. These results demonstrate that our method can help to find both the general and commonly used attributes and the specific and infrequently used attributes.

Results on the Animals Dataset

The animals dataset is a subset of the animals with attributes (AwA) dataset (Lampert, Nickisch, and Harmeling 2009) with 16 species, and we select 40 pictures for each category. Compared with the above two datasets, it possesses more cross-category variety. We generate 3 tasks with randomly selected representative pictures for each arrangement. So we post 360 AMT tasks, and each task costs $0.07 + 0.02$ dollars.

Zero-Shot Learning. The main focus of this experiment is to quantitatively evaluate the quality of the crowdsourced attributes. Since we believe that better attributes should contain more cross-category information, we evaluate attributes by conducting zero-shot learning (ZSL) tasks. Specifically, we randomly split the 16 species into two parts, i.e. the source domain and the target domain. Then we train classifiers using the data and labels from the source domain, and transfer them into target domain classifiers with the help of the attributes. During experiments, we use the deep features extracted by a 19 layer convolution neural network (VGG19) (Simonyan and Zisserman 2015) for classification, and the ESZSL algorithm (Romera-Paredes and Torr 2015), which requires binary attributes, to do zero-shot learning. Since each crowdsourced attribute has three possible values, we split it into two binary attributes to fit the learning algorithm. Thus we have $2K$ crowdsourced attributes in total.

To show the performances under different conditions, we train classifiers on the source domains with the amounts of species M_S vary in $[11,\ 12,\ 13,\ 14]$. So the sizes of the target domains M_T vary in $[5,\ 4,\ 3,\ 2]$ correspondingly. The hyperparameter ρ can influence the amount of the crowdsourced attributes, and it is tested using values from $[0.55,\ 0.60,\ 0.65,\ 0.70]$. We run experiments on 100 random domain partitions for each value of M_T and ρ.

Table 1: Classification accuracies on zero-shot learning. M_T means the amount of categories in the target domain.

Attributes	ρ	K	$M_T = 2$	$M_T = 3$	$M_T = 4$	$M_T = 5$
Crowdsouced	0.55	11	$\mathbf{0.793 \pm 0.189}$	$\mathbf{0.644 \pm 0.197}$	$\mathbf{0.572 \pm 0.186}$	$\mathbf{0.442 \pm 0.149}$
	0.60	15	$\mathbf{0.794 \pm 0.162}$	$\mathbf{0.665 \pm 0.174}$	$\mathbf{0.586 \pm 0.152}$	$\mathbf{0.463 \pm 0.146}$
	0.65	20	$\mathbf{0.812 \pm 0.152}$	$\mathbf{0.665 \pm 0.173}$	$\mathbf{0.611 \pm 0.151}$	$\mathbf{0.502 \pm 0.128}$
	0.70	28	$\mathbf{0.822 \pm 0.144}$	$\mathbf{0.673 \pm 0.160}$	$\mathbf{0.592 \pm 0.146}$	$\mathbf{0.478 \pm 0.124}$
AwA-Best $2K$ entries	0.55	11	0.787 ± 0.202	0.604 ± 0.169	0.525 ± 0.140	0.439 ± 0.120
	0.60	15	0.763 ± 0.204	0.622 ± 0.172	0.554 ± 0.142	0.463 ± 0.123
	0.65	20	0.782 ± 0.199	0.603 ± 0.173	0.534 ± 0.142	0.442 ± 0.123
	0.70	28	0.791 ± 0.200	0.626 ± 0.170	0.571 ± 0.141	0.467 ± 0.123
AwA-85 entries	-	-	0.762 ± 0.184	0.596 ± 0.161	0.558 ± 0.128	0.453 ± 0.128

Three types of attributes are evaluated, including the crowdsourced attributes and two baselines formed by the AwA attributes. The average classification accuracies are reported in Tab. 1. When comparing the crowdsourced attributes learned by the Bayesian aggregating model with the complete AwA attributes with 85 entries, it is easy to see that although the crowdsourced attributes have significantly fewer entries, they achieve better transferring performances than the AwA attributes in almost all the situations. We also evaluated the subsets of the AwA attributes. For each domain partition and ρ, we randomly select $2K$ entries (equals to the amount of the crowdsourced attributes) from the 85 AwA attributes for 50 times, and record the highest accuracy achieved by them. Then the average performance over partitions are reported. It shows that with the same number of entries, the crowdsourced attributes can induce performances which are significantly better than those induced by the AwA attributes in all the settings we tested. These results demonstrate that the crowdsourced attributes possess much potential to express the cross-category relationship.

For both the crowdsourced and the AwA attributes, as K increases, the ZSL accuracy usually grows. But we have an interesting observation that in some situations, such as when $M_T = 4$ or 5 for the crowdsourced attributes, the ZSL accuracy stops growing when K is larger than a certain value. Similarly, a subset of the AwA attributes with 56 entries can always induce higher accuracies than that induced by the full set of AwA attributes. This phenomenon implies that it is possible to find the optimal attribute amounts for specific applications, which can help to reduce human labor intensity.

Related Work

Most previous work focuses on extracting attributes from raw features without utilizing crowdsourcing (Rastegari, Farhadi, and Forsyth 2012; Sharmanska, Quadrianto, and Lampert 2012; Marchesotti, Murray, and Perronnin 2015; Huang, Change Loy, and Tang 2016). Parikh and Grauman (2011a) propose a method to interactively discover nameable attributes from humans; and Maji (2012) asks annotators to list differences between images to build a vocabulary of attributes. However, they do not consider the potential noise within the labels and cannot acquire the attribute values for all concerned categories. Patterson and Hays (2012) also collect attributes from crowds by pairwise comparison, but they use the descriptive words during aggregating, which is less robust than the analogical encoding method. On the contrary, Law et al. (2011) design a game to collect the values of attributes from the crowds; and Kovashka and Grauman (2015) utilize crowdsourced labels to find latent factors underlying the human opinions. However, these methods cannot be used to discover novel attributes.

The semantic hashing methods (Salakhutdinov and Hinton 2009) intend to represent items by codes and locate similar items at nearby addresses. The main differences between the semantic hashing codes and the attributes are: (1) the semantic hashing codes are usually hard to interpret; and (2) the attributes are at the category level, while the items in a same category may have different semantic hashing codes. Label aggregation (Raykar et al. 2010; Zhou et al. 2015; Tian and Zhu 2015) is also a popular area in crowdsourcing. Parts of the techniques used in our model, such as the clustering based initialization, are inspired by these works. We are addressing a challenging task with a novel method.

Conclusions

We propose a method to learn human comprehensible attributes with crowdsourcing. We first design an analogical interface to collect relative labels from the crowds. Then we propose a Bayesian model, and an efficient initialization strategy, to aggregate labels and extract concise attributes. The experimental results demonstrate that our methods can discover convincing attributes.

Although we mainly discuss applications on images, the purposed methods have potential to be applied on other data types. The aggregating algorithm is compatible with diverse data types, while we need to design interfaces for specific domains. The interface for images is straightforward, while other types (e.g.,articles) may be harder to comprehend and require specific treatments.

In the future, we will try to combine the attribute learning procedure with zero-shot learning, so that we can build new classifiers only from crowds opinions efficiently. Other possible extensions including introducing the human descriptive responses into the aggregating model, or extend the aggregating method into a nonparametric Bayesian model.

Acknowledgments

The work was supported by the National Basic Research Program (973 Program) of China (No. 2013CB329403), National NSF of China Projects (Nos. 61620106010, 61305066), and the Youth Top-notch Talent Support Program.

References

Cheng, J., and Bernstein, M. S. 2015. Flock: Hybrid crowd-machine learning classifiers. In *CSCW*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Farhadi, A.; Endres, I.; Hoiem, D.; and Forsyth, D. 2009. Describing objects by their attributes. In *CVPR*.

Gentner, D.; Loewenstein, J.; and Thompson, L. 2003. Learning and transfer: A general role for analogical encoding. *Journal of Educational Psychology* 95(2):393–408.

Gomes, R. G.; Welinder, P.; Krause, A.; and Perona, P. 2011. Crowdclustering. In *NIPS*.

Huang, C.; Change Loy, C.; and Tang, X. 2016. Unsupervised learning of discriminative attributes and visual representations. In *CVPR*.

Hwang, C.-L., and Yoon, K. 1981. *Basic Concepts and Foundations*. Springer Berlin Heidelberg. 16–57.

Jayaraman, D., and Grauman, K. 2014. Zero-shot recognition with unreliable attributes. In *NIPS*.

Karaletsos, T.; Belongie, S.; Tech, C.; and Rätsch, G. 2016. Bayesian representation learning with oracle constraints. In *ICLR*.

Kovashka, A., and Grauman, K. 2015. Discovering attribute shades of meaning with the crowd. *IJCV* 114(1):56–73.

Kovashka, A.; Russakovsky, O.; Fei-Fei, L.; and Grauman, K. 2016. Crowdsourcing in Computer Vision. *Foundation and Trends in Computer Graphics and Vision* 10(3):177–243.

Kulis, B., and Jordan, M. 2012. Revisiting k-means: New algorithms via bayesian nonparametrics. In *ICML*.

Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2009. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*.

Law, E.; Settles, B.; Snook, A.; Surana, H.; Ahn, L. V.; and Mitchell, T. 2011. Human computation for attribute and attribute value acquisition. In *CVPR Workshop on Fine-Grained Visual Categorization*.

Maji, S. 2012. Discovering a lexicon of parts and attributes. In *ECCV Workshops and Demonstrations*.

Marchesotti, L.; Murray, N.; and Perronnin, F. 2015. Discovering beautiful attributes for aesthetic image analysis. *IJCV* 113(3):246–266.

Neal, R. M., and Hinton, G. E. 1999. *A View of the EM Algorithm That Justifies Incremental, Sparse, and Other Variants*. MIT Press. 355–368.

Nilsback, M.-E., and Zisserman, A. 2008. Automated flower classification over a large number of classes. In *ICVGIP*.

Norouzi, M.; Mikolov, T.; Bengio, S.; Singer, Y.; Shlens, J.; Frome, A.; Corrado, G. S.; and Dean, J. 2014. Zero-shot learning by convex combination of semantic embeddings. In *ICLR*.

Palatucci, M.; Pomerleau, D.; Hinton, G. E.; and Mitchell, T. M. 2009. Zero-shot learning with semantic output codes. In *NIPS*.

Parikh, D., and Grauman, K. 2011a. Interactively building a discriminative vocabulary of nameable attributes. In *CVPR*.

Parikh, D., and Grauman, K. 2011b. Relative attributes. In *ICCV*.

Patterson, G., and Hays, J. 2012. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*.

Rastegari, M.; Farhadi, A.; and Forsyth, D. 2012. Attribute discovery via predictable discriminative binary codes. In *ECCV*.

Raykar, V. C.; Yu, S.; Zhao, L. H.; Valadez, G. H.; Florin, C.; Bogoni, L.; and Moy, L. 2010. Learning from crowds. *JMLR* 11(2):1297–1322.

Romera-Paredes, B., and Torr, P. 2015. An embarrassingly simple approach to zero-shot learning. In *ICML*.

Salakhutdinov, R., and Hinton, G. 2009. Semantic hashing. *International Journal of Approximate Reasoning* 50(7):969–978.

Sharmanska, V.; Quadrianto, N.; and Lampert, C. H. 2012. Augmented attribute representations. In *ECCV*.

Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Tian, T., and Zhu, J. 2015. Max-margin majority voting for learning from the crowd. In *NIPS*.

Wainwright, M. J., and Jordan, M. I. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning* 1(1-2):1–305.

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

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