Scalable Multi-label Annotation

Jia Deng[1,2], **Olga Russakovsky**[2], **Jonathan Krause**[2], **Michael Bernstein**[2], **Alex Berg**[3], **and Li Fei-Fei**[2]
University of Michigan[1], Stanford University[2], UNC Chapel Hill[3]
jiadeng@umich.edu, {olga, jkrause, msb, feifeili}@cs.stanford.edu, aberg@cs.unc.edu

ABSTRACT

We study strategies for scalable multi-label annotation, or for efficiently acquiring multiple labels from humans for a collection of items. We propose an algorithm that exploits correlation, hierarchy, and sparsity of the label distribution. A case study of labeling 200 objects using 20,000 images demonstrates the effectiveness of our approach. The algorithm results in up to 6x reduction in human computation time compared to the naïve method of querying a human annotator for the presence of every object in every image.

Author Keywords

Human computation; Crowdsourcing

ACM Classification Keywords

H5.m Information Interfaces and Presentation: Miscellaneous

INTRODUCTION

Consider building an AI system which is able to navigate a user's photo album and automatically find all pictures which contain a cat but not a dog, pictures which show both a table and a chair, or pictures which have a boat, sky, and sheep. Building such a system requires first collecting a training set of images with known annotations: each of the images in the training set needs to be labeled with the presence or absence of a dog, cat, table, and all other objects of interest. In another domain, consider building a system which automatically recommends songs to users based on their preferences. Creating this requires collecting a large training set of songs hand-annotated by humans with many musical attributes. A key component of building both of these systems is doing *multi-label annotation*, or acquiring multiple labels from humans for a collection of items.

A key challenge for multi-label annotation is scalability. Suppose there are N inputs which need to be annotated with the presence or absence of K labels. A naïve approach would query humans for each combination of input and label, requiring $N \times K$ queries. However, in real life applications N and K can be very large and the cost of this exhaustive approach quickly becomes prohibitive. For example, state-of-the-art computer vision algorithms use thousands or millions of images for training and evaluation [7] and are interested in determining the presence of 10,000 or even up to 100,000 object classes [6]. The number of queries required in this case is 1,000,000 images × 100,000 objects, which costs $10 million even in the optimistic setting of perfect workers who label at a cost of 10 cents per 1,000 annotations.

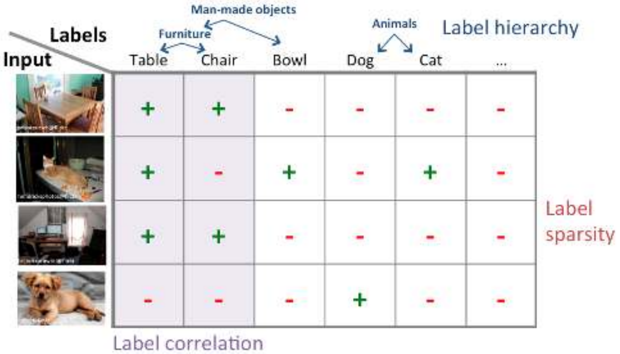

	Table	Chair	Bowl	Dog	Cat	...
	+	+	-	-	-	-
	+	-	+	-	+	-
	+	+	-	-	-	-
	-	-	-	+	-	-

Figure 1. Multi-label annotation becomes much more efficient when considering real-world structure of data: correlation between labels, hierarchical organization of concepts, and sparsity of labels.

In this paper we study strategies for scaling up multi-label annotation, i.e. obtaining labels with a cost substantially smaller than that of the exhaustive naïve approach. This technique is important in multiple domains, such as labeling actions in videos [11], news article topics [15], functional classes of genes [8], musical attributes or emotions in songs [12], semantic classes of scenes [2], product categories customers are likely to buy [21], and categories of web pages [18]. While the problem of acquiring one label has been well studied [20, 9, 22, 19, 16, 5], to our knowledge the challenge of large-scale multi-label annotation has not been addressed before.

We exploit three key observations for labels in real world applications (illustrated in Figure 1).

Correlation. Subsets of labels are often highly correlated. Objects such as a computer keyboard, mouse and monitor frequently co-occur with each other in images. Topics such as economy and finance often co-occur in news articles. Similarly, some labels tend to all be absent at the same time. For example, all objects that require electricity are usually absent in pictures taken outdoors. This suggests that we could potentially "fill in" the values of multiple labels by grouping them into only one query for humans. Instead of checking if dog, cat, rabbit etc. are present in the photo, we check them as a group animal. If the answer is no, then this implies a no for all categories in the group.

Hierarchy. The above example of grouping dog, cat, rabbit etc. into animal has implicitly assumed that labels can be

CHI 2014, April 26 - May 01 2014, Toronto, ON, Canada
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2473-1/14/04$15.00.
http://dx.doi.org/10.1145/2556288.2557011

grouped together and humans can efficiently answer queries about the group as a whole. This brings up our second key observation: humans organize semantic concepts into hierarchies and are able to efficiently categorize at higher semantic levels [17], e.g. humans can determine the presence of an animal in an image as fast as every type of animal individually. This leads to substantial cost savings.

Sparsity. The values of labels for each item tend to sparse, i.e. an image is unlikely to contain more than a dozen types of objects, a small fraction of the tens of thousands of object categories. This enables a rapid elimination of many objects, filling no for many labels very quickly. With a high degree of sparsity, an efficient algorithm can have a cost which grows logarithmically with the number of objects instead of linearly.

In this paper we propose algorithmic strategies that exploit the above intuitions. The key is to select a sequence of queries for humans such that we achieve the same labeling results with only a fraction of the cost of the naïve approach. The main challenges include how to measure cost and utility, how to construct good queries, and how to order them. We present a theoretical analysis and a practical algorithm.

We then perform a case study using our approach on a task of labeling 200 objects in 20,000 images, a total of 4 million labels. We describe our system setup in detail and discuss various design heuristics, including how to frame cost effective queries posted to humans. Experiments demonstrate that our approach is much more scalable than the naïve approach.

RELATED WORK

Acquiring labels as a crowdsourcing task has been extensively studied. The key challenge is making efficient use of resources to achieve quality results. A growing body of work has studied how to estimate worker quality [9], how to combine results from multiple noisy annotators [20, 22, 19], how to model the trade-off between quality and cost [5], how to merge machine and human intelligence [10], as well as how to select the next best item to label [16]. However, they only focus on the single-label case. Multi-label annotation has been practiced in many crowd-powered systems. For example, PlateMate [14] tags all foods in each photo for nutrition estimation. VizWiz [1] labels the presence of objects in images to help blind users. These systems, however, do not address the scalability issue of a large number of labels.

Our framework of optimizing the sequence of queries to fill in values relates to general strategies using iterative steps [13] to limit the search space. For example, Branson et al. study how to select questions for multi-class image classification [4]; this is a special case of our setting where only one class can be present in an image. Our work also draws on research on multi-label classification in crowdsourcing [3]. We exploit a *given* label hierarchy to rapidly eliminate labels, whereas previous work has no access to a hierarchy and cannot issue high level queries outside the label set. Instead, this previous work achieves speed-ups by modeling label co-occurrences.

APPROACH

We first describe a meta algorithm for multi-label annotation, and then customize to make it more efficient. For clarity of exposition and without any loss of generality we use the task of labeling images with the presence of objects as a running example. Here each label represents the presence or absence of an object and takes a value of yes or no. We assume that all labels are binary since any multi-valued label can be represented as a set of mutually exclusive binary labels.

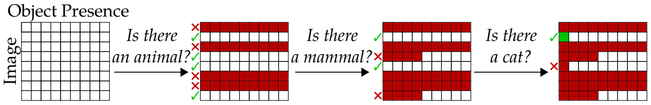

Figure 2. Our algorithm dynamically selects the next query to efficiently determine the presence or absence of every object in every image. Green denotes a positive annotation and red denotes a negative annotation. This toy example illustrates a sample progression of the algorithm for one label (cat) on a set of images.

Algorithm. Our meta algorithm (Algorithm 1) poses a sequence of queries to humans. Each query allows us to fill in values for some labels. We stop when all values are filled. A few sample iterations of the algorithm are shown in Figure 2.

```
Input: An item to be labeled
Output: $K$ labels, each label $+1$ or $-1$
Set values of all $K$ labels to 0 (i.e. $missing$);
while any values are 0 do
    Select a query $Q$ from possible queries $\mathcal{Q}$;
    Obtain an answer $A$ to query $Q$ from humans;
    Set values of some labels to $+1$ or $-1$ given answer $A$;
end
```

Algorithm 1: The meta algorithm for multi-label annotation

In the naïve instantiation of this meta algorithm, we issue one query for each label (i.e. is there a dog in the image). This is clearly not scalable as the cost is $O(NK)$ for N items and K labels. The key to scalability is using additional queries that may fill in multiple values (e.g. if there is no animal with four legs, we know there is no dog and no cat and no rabbit in the image). Moreover, we can exploit the fact that the meta algorithm allows *dynamic* selection of the next query based on the current available information.

A good query should fill in as many values as possible and is easy for humans to answer. In other words, we would like to pick a question with the most *utility* in filling in the values per unit of *cost*. We now make the two notions precise.

Utility. We measure the utility of a query as the expected number of new values filled in over a distribution of items to be labeled. Consider an image with k missing labels. Let $y \in \{-1, 0, +1\}^k$ represent the values of those k labels after using query Q, where -1 means "no," 0 means "unknown" and 1 means "yes." Thus the l_1 norm $\|y\|_1$ is the number of newly acquired labels. The utility of Q is $U(Q) = \mathbf{E}\|y\|_1$.

In practice the utility can be estimated using a "training" set, i.e. an i.i.d. sample of items with ground truth annotations.[1] Suppose we have a set of n training images labeled with the presence or absence of cats, dogs, and other objects. Let s be the number of objects of interest which are "animals," and consider the high-level query "is there an animal present." Let $n-$ be the number of training images with no animals. On these images the query yields s new labels since it reveals that all s animals must be absent. On the other images there are no new labels since it is still unknown which of the s animals are present. Thus, the estimated utility is $\hat{U}(Q) = sn^{-}/n$. This utlity may be high in practice for well-designed queries. In contrast, consider a low-level query such as "is there a cat present." The utility would always be 1, since on every image it reveals one new label: $+1$ if there is a cat, -1 otherwise.

Correlation and sparsity of large label sets leads to high utility of certain queries. For example, when annotating a diverse set of internet images for the presences of couches, desks, sofas, and chairs, designing queries with good utility (e.g., is furniture present?) is easy because the labels are correlated: most internet images that do not have couches also will not have desks. High sparsity means potentially more high utility queries because for most inputs most queries will have a no answer (e.g., most images will not have most of the objects being annotated).

Cost. We measure the cost $C(Q)$ of a query Q as the expected human time it takes to obtain a reliable answer for one item. First, we can empirically measure the average amount of time a human takes to answer a query on a small training set. Next, we might need to consult multiple humans to be confident in the answer. Here we take the majority voting approach and assume a Bernoulli process for querying multiple workers. Again on a small training set we can estimate that the average worker gives a correct answer with probability $p > 0.5$. Then the accuracy of a majority of $2n + 1$ votes is [16]: $\hat{P}_{2n+1} = \sum_{i=n+1}^{2n+1} \binom{2n+1}{i} p^i (1-p)^{2n+1-i}$. Given an acceptable accuracy threshold $1 - \epsilon$, we can find the number of votes needed to reach the threshold, which allows us to calculate $C(Q)$ as a product of the number of workers needed and the average time a worker takes to give an answer.

To be more scalable than the naïve method, it is crucial to find high-utility queries that are also low cost. This is where the hierarchical structure of the label space helps. Hierarchy means many high-level ("is there an animal?") or attribute-like queries (e.g. "is it red?") have low time cost because they are not arbitrary groupings but useful shortcuts in human cognition.

Selection. In Algorithm 1, the query is selected by maximizing utility per unit cost, i.e., $Q^* = \arg\max_Q \hat{U}(Q)/C(Q)$.

EXPERIMENTS

Task and Implementation. We apply this algorithm to the task of labeling images with the presence or absence of many object categories. We use $20,000$ images from ImageNet [7] and Flickr and annotate them with 200 object categories from accordion to zebra. We manually create a hierarchy of these objects which contains 56 internal queries, using high-level categories such as "animals with hooves," "electronics that play sound" or "liquid containers."

We created a user interface shown in Figure 3 for efficient binary labeling of images. A user is given an object category (either one of the target categories or a high-level category) along with positive and negative example images, and is then asked to click on all images from a large candidate pool which contain an instance of this category. We used this interface to query humans using Amazon Mechanical Turk.

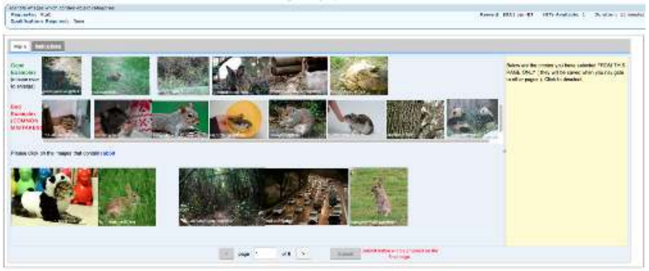

Figure 3. The Amazon Mechanical Turk interface for obtaining human annotations. Here workers are asked to select images which contain a rabbit, and are shown good and bad example images.

We used an early pilot of this algorithm to obtain ground truth annotations on this data, with stringent quality control but potentially suboptimal cost. This allows us to evaluate our algorithm in a controlled setting through simulation. We estimate key simulation parameters (worker confusion matrix, worker response time per image for each query) through real AMT experiments with a sample of 100 images per category, each image labeled 3 workers. Query utility is estimated by the algorithm on the fly using the training set (we use a 10%-90% training-test split). In simulation we enforce a minimum worker accuracy of 75% after filtering of spammers.

Query construction. Before our algorithm can automatically perform query selection, we need to provide a pool of candidate queries. We can leverage general knowledge bases such as WordNet, or specialized ones such as the product taxonomy from eBay. These databases can provide high-level concepts or attributes as candidate queries.

If manual query construction is necessary (e.g. to augment an existing pool), we provide simple heuristics. As discussed above, there are two key components of good queries: high utility and low cost. For high utility the query should be broad in scope (e.g., "is there an animal?", "is there furniture?", "is it sharp?"). To be low cost, the query should be easy for the average human to answer using just salient information in the input. For example, queries such as "are there school supplies?", "motorized vehicles?", "things used to open cans/bottles?" took up to 3 times longer on average than simple queries such as "is there a bug?", " a canine?", "a ball?". Generally, queries should avoid requiring the user to do additional inference beyond the provided input.

Query construction may involve significant effort, but it is a one-time, fixed investment: the label set for a particular application is relatively static, whereas the items to label can be dynamic and infinitely many. The cost saved in labeling many items can easily outweigh the fixed, upfront cost of query construction. Moreover, our method is designed to minimize the effort of query construction as it automatically selects the most effective queries.

[1] We could in principle estimate the utility conditioned on values of existing labels. This is beyond the scope of this paper.

Query: Is there a...[2]	Utility (num labels)	Cost (secs)
mammals that have claws or fingers	12.0	3.0
living organisms	24.8	7.9
mammals	17.6	7.4
creatures without legs	5.9	2.6
land and avian creatures	20.8	9.5

Table 1. The most useful queries at the first iteration of our algorithm. Utility is the expected number of new values for object labels as a result of this query. Cost is the human time needed to obtain the answer with $\geq 95\%$ expected accuracy. Usefulness of a query is utility per unit cost.

Thresh	Accuracy		F1 score		Cost saving
	Naïve	Ours	Naïve	Ours	
95.0	99.64	**99.75±0.00**	75.67	**76.97±0.16**	**3.93±0.00**
90.0	99.29	**99.62±0.00**	60.17	**60.69±0.39**	**6.18±0.01**
85.0	99.25	**99.62±0.00**	59.09	**60.46±0.39**	**6.11±0.01**

Table 2. Our algorithm obtains superior accuracy compared to the naïve brute force approach while being more computationally efficient. Thresh is a parameter of the algorithm (please see text for details).

Some examples of highest-utility queries at the first iteration of our algorithm are shown in Table 1.

Large-scale evaluation. We compare our algorithm to the baseline approach that queries a human for every object in every image (Table 2). We use 3 metrics: (1) accuracy, or the total percentage of correct labels, (2) F1-score, or the harmonic mean of precision and recall on labels from all categories, and (3) reduction of human annotation time of our algorithm compared to the baseline. Error bars are the result of 5 simulations. Threshold is the acceptable level of accuracy; it determines the number of workers needed for each query. Our algorithm obtains up to 6× savings compared to the naïve approach while maintaining superior accuracy.

DISCUSSION AND CONCLUSION

Our algorithm works well in cases where the natural distribution of labels satisfies our assumptions, i.e. when the labels are correlated, sparse, and naturally form a hierarchy. If, on the other hand, the distribution of labels is dense and independent, there is little for our algorithm to exploit. In real world scenarios, though, and as validated by our experiments, exploiting the label distribution can yield significant savings.

REFERENCES

1. Bigham, J. P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R. C., Miller, R., Tatarowicz, A., White, B., White, S., and Yeh, T. VizWiz: Nearly real-time answers to visual questions. In *Proc. UIST* (2010).
2. Boutell, M. R., Luo, J., Shen, X., and Brown, C. M. Learning multi-label scene classification. *Pattern recognition 37*, 9 (2004), 1757–1771.
3. Bragg, J., Mausam, and Weld, D. S. Crowdsourcing multi-label classification for taxonomy creation. In *HCOMP* (2013).
4. Branson, S., Wah, C., Schroff, F., Babenko, B., Welinder, P., Perona, P., and Belongie, S. Visual recognition with humans in the loop. In *ECCV*. 2010.
5. Dai, P., Mausam, and Weld, D. S. Decision-theoretic control of crowd-sourced workflows. In *AAAI* (2010).
6. Dean, T., Ruzon, M. A., Segal, M., Shlens, J., Vijayanarasimhan, S., and Yagnik, J. Fast, accurate detection of 100,000 object classes on a single machine. In *CVPR* (2013).
7. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *CVPR* (2009).
8. Elisseeff, A., and Weston, J. A kernel method for multi-labelled classification. In *NIPS* (2001).
9. Ipeirotis, P. G., Provost, F., and Wang, J. Quality management on amazon mechanical turk. In *HCOMP* (2010).
10. Kamar, E., Hacker, S., and Horvitz, E. Combining human and machine intelligence in large-scale crowdsourcing. In *Proc. AAMAS* (2012).
11. Laptev, I., Marszalek, M., Schmid, C., and Rozenfeld, B. Learning realistic human actions from movies. In *CVPR* (2008).
12. Li, T., and Ogihara, M. Detecting emotion in music. In *ISMIR* (2003).
13. Little, G., Chilton, L. B., Goldman, M., and Miller, R. C. Turkit: tools for iterative tasks on mechanical turk. In *SIGKDD workshop on human computation* (2009).
14. Noronha, J., Hysen, E., Zhang, H., and Gajos, K. Z. PlateMate: crowdsourcing nutritional analysis from food photographs. In *Proc. UIST* (2011).
15. Schapire, R. E., and Singer, Y. Boostexter: A boosting-based system for text categorization. *Machine learning 39*, 2-3 (2000), 135–168.
16. Sheng, V. S., Provost, F., and Ipeirotis, P. G. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *SIGKDD* (2008).
17. Thorpe, S., Fize, D., Marlot, C., et al. Speed of processing in the human visual system. *nature 381*, 6582 (1996), 520–522.
18. Ueda, N., and Saito, K. Parametric mixture models for multi-labeled text. In *NIPS* (2002).
19. Welinder, P., Branson, S., Perona, P., and Belongie, S. J. The multidimensional wisdom of crowds. In *NIPS* (2010).
20. Whitehill, J., Wu, T.-f., Bergsma, J., Movellan, J. R., and Ruvolo, P. L. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS* (2009).
21. Zhang, Y., Burer, S., and Street, W. N. Ensemble pruning via semi-definite programming. *JMLR 7* (2006), 1315–1338.
22. Zhou, D., Platt, J., Basu, S., and Mao, Y. Learning from the wisdom of crowds by minimax entropy. In *NIPS* (2012).

[2] Actual queries are longer and include detailed definitions.