Abstract

In this short paper, we briefly describe some recent progress on statistical decision making for budget allocation in crowdsourcing. We address the budget allocation problem for two important labeling tasks in crowdsourcing: the categorization labeling task and pairwise ranking aggregation. We also show the connections between our work and the “proactive learning” framework proposed by Jaime Carbonell.

1 Introduction

In many real applications, data are usually collected without any innate label. For example, a digital camera will not automatically tag a picture as a portrait or a landscape. A traditional approach for data labeling is to hire a small group of experts to provide labels for the entire set of data. However, for large-scale data, such an approach becomes inefficient and very costly. Thanks to the advent of many online crowdsourcing services, e.g., Amazon Mechanical Turk, a much more efficient way is to post unlabeled data to a crowdsourcing marketplace, where a big crowd of low-paid workers can be hired instantaneously to perform labeling tasks.

Despite its high efficiency and immediate availability, crowd labeling raises many new challenges. Since labeling tasks are tedious and workers are usually non-experts, labels generated by the crowd suffer from low quality. As a remedy, most crowdsourcing services resort to labeling redundancy to reduce the labeling noise, which is achieved by collecting multiple labels from different workers for each data instance. In particular, a crowd labeling process can be described as a two phase procedure:

1. In the first phase, unlabeled data instances are assigned to a crowd of workers and multiple raw labels are collected for each data instance.

2. In the second phase, for each data instance, one aggregates the collected raw labels to infer its true label.

In principle, more raw labels will lead to a higher chance of recovering the true label. However, each raw label comes with a cost: the requester has to pay workers a pre-specified monetary reward for each label they provide, usually, regardless of the label’s correctness. For example, a worker typically earns ten cents by categorizing a website as porn or not. In practice, the requester has only a limited budget, which essentially restricts the total number of raw labels that he/she can collect. This raises a challenging question that is central to crowd labeling: What is the best way to allocate the budget among data instances and workers so that the overall accuracy of aggregated labels is maximized?

The most important factors that determine how to allocate the budget are the intrinsic characteristics of data instances and workers: the labeling difficulty/ambiguity of each data instance and the reliability/quality of each worker. In particular, an instance is less ambiguous if its label can be decided based on common knowledge and the vast majority of reliable workers will provide the same label for it. In principle, we should avoid spending too much of the budget on those easy instances because additional raw labels will not bring much additional information. In contrast, for an ambiguous instance that falls near the decision boundary, even reliable workers will still disagree with each other and generate inconsistent labels. For those ambiguous instances, we are facing a challenging decision problem on how much of the budget we should spend on them. On one hand, it is worth collecting more labels to boost
the accuracy of the aggregate label. On the other hand, because our goal is to maximize the overall labeling accuracy, when the budget is limited, we should simply put those few highly ambiguous instances aside to save money for labeling less difficult instances. In addition to the ambiguity of data instances, the other important factor is the reliability of each worker; undoubtedly, it is desirable to assign more instances to the workers who are reliable. Despite their importance in deciding how to allocate the budget, both data ambiguity and workers’ reliability are unknown parameters at the beginning and need to be updated based on the stream of collected raw labels in an online fashion. This suggests that the budget allocation policy should be dynamic and simultaneously conduct parameter estimation and decision making. In this work, we discuss budget allocation for two important data labeling tasks: (1) labeling categorical data (i.e., each data instance is a multiple choice question with several possible options or classes) and (2) making a comparison of pairs of objects to infer the goal ranking of the dataset.

2 Categorical Data Labeling

For categorical data labeling tasks, we model data ambiguity and workers’ reliability using two sets of random variables drawn from known prior distributions. Then, we formulate the problem into a finite-horizon Bayesian Markov Decision Process (MDP), whose state variables are the posterior distributions of these variables, which are updated by each new label. The Bayesian setting is necessary; in (Chen et al., 2013b) we show that a uniformly optimal policy exists only in the Bayesian setting. Using the MDP formulation, the optimal budget allocation policy for any finite budget level can be readily obtained via dynamic programming (DP). However, DP is computationally intractable for large-scale problems since the size of the state space grows exponentially with budget level. The existing widely-used approximate policies, such as approximate Gittins index rule (Gittins, 1989) or knowledge gradient (KG) (Gupta and Miescke, 1996; Frazier et al., 2008), either have high computational costs or poor performance for our problem. In this paper, we propose a new policy, called optimistic knowledge gradient (Opt-KG). In particular, the Opt-KG policy dynamically chooses the next instance-worker pair based on the optimistic outcome of the marginal improvement on the accuracy, which is a function of state variables. We further propose a more general Opt-KG policy using the conditional value-at-risk measure (Rockafellar and Uryasev, 2002). The Opt-KG policy is computationally efficient, achieves superior empirical performance and has asymptotic theoretical guarantees.

3 Pairwise Ranking Aggregation

Obtaining a set of gold-standard labels for a set of objects is a critical step in learning to rank. For example, when determining how to rank the results returned in response to a web search, the results are often passed through a ranking model that has been learned using a machine learning procedure (Liu, 2009). To achieve this goal, an effective method is to take advantage of crowdsourcing services to query pairwise comparisons among objects (i.e., \( i \) is preferred to \( j \), denoted as \( i \succ j \)). However, the reliability of workers available via crowdsourcing can vary significantly. In addition, seeking pairwise assessments from the crowd can lead to inconsistent pairs (e.g., \( i \succ j \) by one worker and \( j \succ i \) by another worker; or \( i \succ j \), \( j \succ k \) and \( k \succ i \)). Many existing ranking aggregation methods are either incapable of modeling the quality of work by workers or are inadequate for dealing with inconsistent pairs.

To address the challenge of learning a global ranking in a crowdsourced setting, we introduce the Crowd-BT algorithm in (Chen et al., 2013a), which extends the widely used Bradley-Terry model (R.A.Bradley and Terry, 1952) by explicitly incorporating the quality of contributions provided by different workers. The Crowd-BT algorithm can both appropriately weight workers’ contributions by their annotation quality as well as distinguish between spammers and malicious workers; spammers assign random labels, while malicious workers (or poorly informed workers) assign the wrong label most of the time. Beyond appropriately handling error, spam, and malicious inputs, we seek to be budget-conscious; we typically prefer to harness fewer labeled samples while achieving reasonably good accuracy. Thus, we formulate and study an exploration-exploitation tradeoff in crowdsourcing, previously explored
in bandit and reinforcement learning. More precisely, *exploration* refers to using pairs with high-confidence labels to test the quality of workers, while *exploitation* refers to asking for labels for the most uncertain pairs. We carefully model and balance the exploration-exploitation tradeoff and implement it using an efficient online Bayesian updating scheme. Using real-world data from Microsoft’s Bing group, we demonstrate that our budget allocation strategy achieves significant reductions in labeling cost while maintaining accuracy.

4 Connections with Proactive Learning

Budget allocation in crowdsourcing has a lot of connections to and similarities with the proactive learning framework proposed by Jaime Carbonell (e.g., (Donmez and Carbonell, 2008; Yang and Carbonell, 2010)). According to (Donmez and Carbonell, 2008), “Proactive learning is a generalization of active learning designed to relax unrealistic assumptions and thereby reach practical applications. Active learning seeks to select the most informative unlabeled instances and ask an omniscient oracle for their labels, so as to retrain the learning algorithm maximizing accuracy. However, the oracle is assumed to be infallible (never wrong), indefatigable (always answers), individual (only one oracle), and insensitive to costs (always free or always charges the same). Proactive learning relaxes all four of these assumptions, relying on a decision-theoretic approach to jointly select the optimal oracle and instance, by casting the problem as a utility optimization problem subject to a budget constraint.” As we can see, in crowdsourcing, each worker corresponds to a noisy oracle in proactive learning and the budget allocation in crowdsourcing is also a decision problem which needs to dynamically jointly select an oracle and an instance at each stage under a budget constraint. The main difference between learning from crowds and proactive learning is that the goal of the proactive active learning is to label as few instances as possible to learn a good classifier. In contrast, for budget allocation in crowd labeling, the goal is to infer the true labels for as many instances as possible. Despite this difference, proactive learning provides both a useful theoretical framework and technical tools for investigating challenging problems in crowdsourcing.

References

Xi Chen, Paul N. Bennett, Kevyn Collins-Thompson, and Eric Horvitz. 2013a. Pairwise ranking aggregation in a crowdsourced setting. In ACM International Conference on Web Search and Data Mining.


