Get another worker? Active crowdlearning with sequential arrivals.

James Zou JZOU@FAS.HARVARD.EDU
David Parkes PARKES@EECS.HARVARD.EDU

Prior work. Much of the literature on crowdlearning focus on the static case where we have a batch of noisy labels (Raykar, 2010) (Karger, 2011) and need to infer the true labels. The emerging online crowdlearning papers either assumes 1) we can repeatedly access any worker (Yan, 2011), or 2) each worker only labels one item. The decision task is how many labels to obtain for each item (Sheng, 2008).

The contributions of this work are:

- We introduce a new model where the workers arrive sequentially and the decision problem of which items to assign to a worker must be made online.
- We use loopy belief propagation to estimate the *posterior distribution* of the true labels and worker types given the current set reported labels. The current literature use maximum likelihood point estimate of the true label and worker type (usually through EM type approach). In our online setting, it is advantageous to know the variances (uncertainty) as well as the mode of the distribution at every time step.
- Using the approximate posterior distribution over true labels and worker types, we design and evaluate a family of active learning algorithms.

Model set up. We have a fixed set of items $X = \{x_1, ..., x_m\}$ that need to be labeled. The labels are binary $\{0, 1\}$, and initially we do not have any known labels (no gold standard). Workers are employed to label these data. Let y_{ij} denote the label that worker w_i assigns to item x_j. Let $\{y_j^*\}$ denote the (unknown) true labels.

A simple worker model often used in crowdlearning literature (Karger, 2011) (Sheng, 2008) is:

- each worker is an iid sample from a binary distribution of good and bad workers.
- $P(\text{worker} = \text{good}) = \alpha$;
  $P(\text{worker} = \text{bad}) = 1 - \alpha$.
- $P(y_{ij} = y_j^* | w_i \text{ good}) = \beta > 0.5$
  $P(y_{ij} = y_j^* | w_i \text{ bad}) = 0.5$.

We assume that workers arrive sequentially $\{w_1, w_2, ...\}$. Each w_i is an iid sample from the binary distribution above. At each time step, we may assign any item x_j to the current worker w_i to label. Alternatively, we can dismiss w_i and assign any item x_j to the new worker w_{i+1}.

A worker is given only one item to label at a time, and returns the label instantaneously. We assume that if we assign the same item to a worker repeatedly, he will return the same label. A worker may be assigned an arbitrary number of items in total. After he is dismissed, he is no longer available for future tasks (gone for good). This captures many natural settings including: 1) where workers are anonymous and can not be identified once the engagement is discontinued; 2) where workers are busy and we only get one shot to learn from each one.

There is a fixed budget to pay for total of T labeling tasks. It costs the same for any worker to label any item. After collecting all the labels, the designer submits a subset of items along with predicted labels to the oracle: $\{(x_j, y_j) | x_j \in \tilde{X} \subseteq X\}$. From the submissions, he receives reward q_1 for every correctly labeled data, and penalty $-q_2$ for each incorrect prediction. The total reward is $R = q_1 * n_1 - q_2 * n_2$, where $n_1 = |\{x_j | y_j = y_j^*\}|$ and $n_2 = |\{x_j | y_j \neq y_j^*\}|$. The objective is to maximize R.

Result I: posterior inference over labels and worker types. Let W_i and Y_j denote the random variables over the true worker type and item label, respectively. Let $D = \{(x_j, y_{ij})\}$ be the data collected. The posterior is given by

$$P(\{W_i\}, \{Y_j\} | D) = \frac{1}{Z} P(D | \{W_i\}, \{Y_j\}) P(\{W_i\}, \{Y_j\})$$

where Z is the normalization factor.

This can be represented by a Markov random field on a bipartite graph with one node for every worker W_i and one for every item Y_j. The adjacency matrix of the graph, A, is also the assignment matrix: $A_{ij} = 1$ if w_i has labeled x_j and $A_{ij} = 0$ otherwise. The factor associated to the edge between W_i and Y_j is $P(y_{ij} | W_i, Y_j)$.

We run loopy belief propagation on this graph (Murphy, 1999). Message from node W_i to Y_j encapsulates worker w_i's belief on the true label of x_j. And message from Y_j to W_i encodes item x_j's contribution to the posterior belief of the true type of w_i. Before we obtain any labels, the initial marginals are $P(Y_j = 0) = 0.5$ and $P(W_i = \text{good}) = \alpha$. The algorithm quickly converges and the belief at each node approximates the posterior marginals $P(W_i)$ and $P(Y_j)$. By comparing with exact marginals on tractable examples, we verify that the Loopy BP algorithm performs effective inference for our model.

So far the model assumes that we have α and β. In some applications, rather than knowing α and β, we may have some prior distributions over these parameters. This more general model can also be encoded in a Markov Random Field, and we use a modified loopy BP to approximate its posterior marginals.

Result II: active learning. The Bayesian optimal prediction is $y_j = \arg\max_{l\in\{0,1\}} P(Y_j = l)$, where $P(Y_j)$ is the posterior approximated from loopy BP. The expected reward is

$$R(\{y_i\}) = \sum_i [q_1 P(Y_i = y_i) - q_2 P(Y_i \neq y_i)]. \quad (1)$$

To maximize R, the designer should only submit prediction (x_j, y_j) if $q_1 P(Y_j = y_j) > q_2 P(Y_j \neq y_j)$.

Example 1. *Suppose we have a budget to collect 20 labels from a large set of items ($M \gg 20$). One naive strategy is to ask the first worker that comes along to label 20 different items. The probability that a label is correct is $0.5+\alpha(\beta-0.5)$. Suppose the penalty is relatively large: $\frac{q_2}{q_1} > \frac{0.5+\alpha(\beta-0.5)}{0.5-\alpha(\beta-0.5)}$. Then the Bayesian optimal decision is to not submit any predictions at the end. This strategy greedily explores item information to learn the most bits about $\{P(Y_j)\}$. It does not learn anything about the worker type, nor does it learn enough about $P(Y_j)$ for any particular item x_j in order to make a confident prediction.*

Example 2. *Following the same set up, an alternative is to have the first worker label items 1-7, second worker label items 1-7, and the third worker label 1-6. At the end, we can infer accurately the type of each worker (sharpe marginals for $P(W_1), P(W_2), P(W_3)$), but have high variance in item posteriors.*

Example 3. *In the third strategy, we recruit ten workers to label item 1 and ten other workers to label item 2. We can infer the two correct item labels with high probability, but we only get at most $2q_1$ reward.*

We identify three dimensions in the learning problem as illustrated by the examples: item exploration, worker exploration, and reward exploitation. Item and worker exploration greedily learn the item and worker marginals. Reward exploitation greedily maximizes the expected reward. The general approach we adopt for online learning is to select at each step the action that, in expectation, maximizes a weighted combination of these three objectives. Let $M_t < M$ be the number of items with at least one label by time t. Then the action space at t has size at most $2(M_t + 1)$: assign one of the M_t points to either the current worker or to the new worker, or give an unlabeled data to the current or new worker. It doesn't matter which of the $M - M_t$ unlabeled items to pick since they are symmetric a priori.

Item exploration. Let $D(t)$ be the set of labels collected up to time t and $\{P(Y_j, t)\}$ be the item marginals at t. Consider an action, a, that assigns x_j to w_i. There are two outcomes, $y_{ij} = 0$ or $y_{ij} = 1$. Run loopy BP on $D(t) \bigcup (x_i, y_{ij} = 0)$ and $D(t) \bigcup (x_i, y_{ij} = 1)$ to obtain the new marginals $\{P(Y_j, a, 0)\}$ and $\{P(Y_j, a, 1)\}$ respectively. Define $\triangle_Y(a, 0) = \sum_j \parallel P(Y_j, a, 0) - P(Y_j, t) \parallel$ and $\triangle_Y(a, 1) = \sum_j \parallel P(Y_j, a, 1) - P(Y_j, t) \parallel$. The norm is L_2. Set $\triangle_Y(a) = \triangle_Y(a, 0)P(y_{ij} = 0) + \triangle_Y(a, 1)P(y_{ij} = 1)$. $\triangle_Y(a)$ measures the expected gain in item information from action a.

Worker exploration. For each action a and possible outcome, we similarly compute the marginal distributions: $\{P(W_i, a, 0)\}$ and $\{P(W_i, a, 0)\}$. Define $\triangle_W(a, 0) = \sum_i \parallel P(W_i, a, 0) - P(W_i, t) \parallel$ and $\triangle_W(a, 1) = \sum_i \parallel P(W_i, a, 1) - P(W_i, t) \parallel$. The gain in worker information is $\triangle_W(a) = \triangle_W(a, 0)P(y_{ij} = 0) + \triangle_W(a, 1)P(y_{ij} = 1)$.

Reward exploitation. Let $R(t)$ be the expected reward based on $D(t)$ as in Eqn.1. For each action a and outcome, compute the expected reward $R(a, 0)$ and $R(a, 1)$. Define $R(a) = R(a, 0)P(y_{ij} = 0) + R(a, 1)P(y_{ij} = 1)$. Then the gain in reward is $\triangle_R(a) = R(a) - R(t)$.

The algorithm chooses the action a that maximizes $\triangle_R(a) + \gamma_1\triangle_Y(a) + \gamma_2\triangle_W(a)$. The parameters γ_1 and γ_2 determine the relative weight of the objectives, and characterize our algorithmic space. We give analytic characterizations of the algorithm in the asymptotic limits of γ_1 and γ_2. We empirically investigate the optimal choices of the parameters as a function of α, β while allowing it to also depend on time: $\gamma_1(\alpha, \beta, t)$ and $\gamma_2(\alpha, \beta, t)$. We also benchmark the performance of our active learning algorithm against several non-adaptive strategies for assigning items to workers.

References

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

Karger,D., Oh,S., and Shah,D. Iterative learning for reliable crowd-sourcing systems. *Proceedings of NIPS*, 2011.

Yan,Y., Rosales, R., Fung, G., and Dy J. Active Learning from Crowds. *ICML*, 2011.

Sheng,V., Provost,F., and Ipeirotis, P. Get Another Label? Improving Data Quality and Data Mining using Multiple Noisy Labelers. *KDD*, 2008.

Murphy, K., Weiss, Y., Jordan, M. Loopy Belief Propagation for Approximate Inference: An Empirical Study. *UAI*, 1999.