Exploiting Commonality and Interaction Effects in Crowdsourcing Tasks Using Latent Factor Models

Paul Ruvolo
Department of Engineering
Franklin W. Olin College of Engineering
Paul.Ruvolo@olin.edu

Jacob Whitehill
Machine Perception Laboratory
University of California San Diego
jake@mplab.ucsd.edu

Javier R. Movellan
Machine Perception Laboratory
University of California San Diego
movellan@mplab.ucsd.edu

Abstract

Crowdsourcing services such as the Amazon Mechanical Turk [1] are increasingly being used to annotate large datasets for machine learning and data mining applications. The crowdsourced data labels must then be somehow combined to form a final judgment on the "true" label value of each data instance. Recently developed algorithms for combining multiple opinions [13, 12, 17, 4, 9, 15] have shown promising results, but they are lacking in several ways: (1) Each labeler is typically treated independently. In practice, labelers may share much in common, and their labeling proficiency may vary as a function of easily-queried attributes such as age and geographical location. Such *commonality* both among data labelers and among data instances could be better exploited. (2) *Interaction effects* between labelers and data and the reliability of a given label are ignored. For instance, some labelers may be good at labeling particular kinds of data but not others. In this paper, we present a probabilistic model for combining crowdsourced opinions that discovers and exploits both commonality and interaction effects automatically using a latent product-of-factors approach. We evaluate our proposed method on a facial expression labeling task and on a geography knowledge test. Empirical results show increased accuracy of the label estimation compared to competing methods. In addition, the model revealed interesting and useful trends relating the labeler and data features to the probability of correct labeling.

1 Introduction

As machine learning and data mining applications tackle more and more difficult problems, the importance of large-scale and varied datasets of high-quality training data becomes more apparent [8]. In order to meet the large demand for labeled data, researchers have been increasingly relying on crowdsourcing services that allow them to harness vast pools of human labelers at very low cost. Current systems include Pay-per-label services such as Amazon's Mechanical Turk [1] and interactive games such as *Herd It* [2] and the ESP game [14]. However, with the power of crowdsourcing come new challenges as well, e.g.: How should labelers' opinions be weighted when arriving at a final "judgment" on each data label? Could some labelers be particularly skilled at answering some kinds of questions, but not others? Can known properties of the labelers (e.g., age, gender) and data (e.g., subject matter) be exploited to make better inferences?

As crowdsourcing tools have grown in popularity, machine learning algorithms have been developed to address some of these challenges [13, 12, 17, 4, 9]. These algorithms have thus far focused mostly on binary labeling tasks. The high-level premise is that a binary class label Z is a latent random variable that must be inferred for each data instance using the set of observed labels $\mathbf{L}$ obtained from multiple labelers. The probability that a particular labeler i would correctly label a particular data instance j is then computed as an intermediate variable (henceforth called the "correctness probability"), which may be estimated from the labeler's *ability* as well as certain properties of the data themselves, such as a *difficulty* parameter [17] or perhaps a vector of unobserved features for each data instance [15]. The estimation of each Z_j then proceeds using the Expectation-Maximization algorithm (see Section 3). These algorithms vary chiefly in the form and complexity of the data likelihood function. For instance Dawid and Skene [3] model the correctness probability as depending on a latent accuracy attribute of the labeler. A more recent algorithm by Whitehill *et. al.* [17] additionally models a latent difficulty attribute associated with each data instance. Finally, [15] use a generative approach and thus the label likelihood is computed by marginalizing over a set of hidden factors. While such crowdsourcing models have shown promising results compared to heuristic methods such as Majority Vote, they also neglect important phenomena such as *commonality* among labelers and among data instances, and *interaction effects* between labelers and instances. These phenomena may strongly affect the accuracy of estimating the correctness probability, and hence the overall quality of the inferred data labels.

- **Commonality**: Existing work typically treats each labeler (and each data instance) independently. While this approach offers, in a sense, the maximum flexibility, it also runs risk of over-fitting the model parameters. In practice, labelers may share much in common, and their labeling proficiency may vary as a function of a smaller number of easily-queried attributes such as age and geographical location. Similarly, instead of modeling the difficulty of each data instance independently, it could be inferred instead from a set of features (e.g., image resolution on an image labeling task) that are shared among many data. There may even be latent factors that are highly predictive of the correctness probability that cannot easily be foreseen but instead emerge naturally from the data.
- **Interaction effects**: Some labelers may be good at labeling particular kinds of data but not others (*specialization*). For instance, labelers who are electronic music aficionados may have an accuracy in categorizing music as a "trip hop" song that is higher than their performance on a more general musical genre labeling task would predict. Most previous approaches cannot model this because they assume each labeler has only 1 latent accuracy attribute that applies to all instances equally. Another effect is *trickiness*, in the sense that the probability of correct labeling could be non-monotonic in the labeler's skill level.

In this paper, we present a novel crowdsourcing model, based on the product of latent factors, that effectively discovers commonality and interaction effects between labelers and data, and also exploits these phenomena to improve label estimation (Z) accuracy compared to competing methods such as Majority Vote, GLAD [17], and the model of Welinder, et. al [15] (which we call MultiDim in this paper). These latent factors are capable of capturing a wide range of dimensions of variability in the correctness probabilities (e.g. question trickiness and labeler specialization). In addition, the model we propose contributes to the emerging usage of crowdsourcing for behavioral psychology where the experimenter wants to test some hypothesis (for instance using an online questionnaire). In this type of application the aim is not so much to correctly infer the data labels, but to understand the patterns that influence the particular responses of labelers to various data queries. We show that the factors that our model extracts lead to both more accurate labels and insight into the trends that influence the probability of a particular labeler responding correctly to a particular question.

2 Modeling the Labeling Process

We seek to determine the latent class label $Z_j \in \{-1, 1\}$ for each instance j of a dataset of n data instances by querying m different labelers. We assume that we are given access to *a priori* information describing each labeler and instance. Such information is often available in real world labeling tasks via questionnaires (e.g., of gender or age) asked of the labelers or low-level features (e.g., image resolution) extracted from the data instances. We use the notation $L_{ij} \in \{-1, 1\}$ to refer to the response of the ith labeler to the jth instance and the symbol $\mathbf{L}_j$ to refer to the collection

of responses given by the labelers to the jth image. There is no requirement that each labeler label every data instance. A high-level picture of the labeling process according to our model is:

1. The probability that labeler i correctly labels data instance j (the "correctness probability") depends on the interaction between D latent instance and labeler factors, plus the labeler bias (which allows our method to infer interesting effects in in crowdsourced data effects that single factor models [17, 13, 3, 9] cannot capture).
2. Each latent factor is modeled as a linear function of a known set of features with an inferred set of weights (described in Section 2.1). In Section 2.1 we show that these features can be used to help infer commonalities among instances and labelers that affect the likelihood of labelers responses.
3. Expectation-Maximization is used to infer the unknown weights that map the labeler and instance features into the latent factors (described in Section 3).

We assume that the correctness probabilities depend on the interaction between a set of D latent labeler and D latent instance factors as well as labeler bias towards a particular class. For now we will not impose any structure on these latent factors, however, in Section 2.1 we will show how these latent factors can be parameterized to take into account *commonalities* among labelers and instances that may be quantifiable using known attributes of these entities. We use the notation $\alpha_{i,k}$ and $\beta_{j,k}$ (scalar quantities) to refer to the value of the kth labeler factor of labeler i and the kth instance factor of instance j respectively. The values of all k factors associated with the ith labeler and jth instance are referred to as $\boldsymbol{\alpha_i}$ and $\boldsymbol{\beta_j}$ (row vectors), respectively. Additionally we use the symbol γ_i to refer to the bias of the ith labeler. A column vector containing the biases of each labeler is given by $\boldsymbol{\gamma}$. Finally, we let $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ (matrices of row vectors) represent the collection of $\boldsymbol{\alpha_i}$ across all i and $\boldsymbol{\beta_j}$ across all j, respectively.

The likelihood is modeled as the sum of multiplicative interactions between each of the instance and labeler factors. The result is constrained to be between 0 and 1 using the logistic function σ:

$$p(L_{ij} = z_j | z_j, \boldsymbol{\alpha_i}, \boldsymbol{\beta_j}, \gamma_i) \quad = \quad \sigma\left(\sum_{k=1}^{D} (\alpha_{i,k}\beta_{j,k}) + z_j\gamma_i\right) \tag{1}$$

This particular form of the likelihood function exhibits several useful properties. First, it allows us to recover previous models of label quality control as special cases (see Section 2.2). Second, as we show in the next paragraph, it provides a view of maximum likelihood inference of the labeler and instance factors as a low-rank matrix factorization with missing data (where one matrix factor contains the "labeler factors" and one matrix factor contains the "instance factors"). This is a view that has been shown to be effective in the field of collaborative filtering [11].

The correctness probabilities can be expressed in matrix form using the symbols P^1 and P^0, where the i, jth cell of each matrix refers to the probability that the ith labeler correctly labels the jth instance, conditional on the true label being either 1 (in the case of P^1) or 0 (in the case of P^0). By applying Equation 1 the correctness probability matrices can be computed as: $P^1 = \sigma\left(\boldsymbol{\alpha}\boldsymbol{\beta}^\top + \boldsymbol{\gamma}e^\top\right)$ and $P^0 = \sigma\left(\boldsymbol{\alpha}\boldsymbol{\beta}^\top - \boldsymbol{\gamma}e^\top\right)$ where e is an n dimensional vector of all ones.

Notably, it is not possible to model these correctness probabilities using only 1 latent factor, as in [17]. This can be seen by examining the log-odds matrix that describes a particular labeler's probability of responding correctly to a particular instance. In this case the log-odds matrix is [2.944 .8473; 1.3863 1.3863]. A property of our model is that the matrix of log odds is formed by taking the sum of the outer products of the each labeler factor with the corresponding instance factor. Since the rank of the log-odds matrix in this example is greater than 1, it follows that it would take at least 2 factors to reconstruct it as the sum of the outer products of vectors.

In general, we will not know ahead of time who the generalists and specialists are, and thus will have to learn "who is who" through their given responses. However, in the next section we show that if there exist commonalities among labelers that are correlated with measurable properties of the labelers, then we can predict who is a specialist and incorporate that information into our model to make more efficient inference of structure within the labeling task. In Section 4.2 we show an experiment with a real crowdsourcing task where this type of specialization exists, can be predicted based on user demographics, and can be exploited to infer more accurate labels.

2.1 Parameterizing Latent Factors of Labelers and Instances

In order to determine commonalities among instances and labelers we model the latent labeler factors $\boldsymbol{\alpha}$ and latent instance factors $\boldsymbol{\beta}$ as linear functions of a specified set of features and an unknown set of weights. For example the labeler features might include things such as geographic location, musical tastes, and hobbies. An example of an instance feature would be the head pose of a face being coded for facial expression. For these features to help in learning structural relationships in the data, they must be reasonably expected have an influence on the correctness probabilities. As indicated in the figure by the use of shading, we assume that the algorithm has access to a set of known features for each instance and each labeler. The number of labeler features, instance features, and bias features need not be the same. We use the symbol $\boldsymbol{\Phi}_{\boldsymbol{\alpha_i}}$ to refer to a row vector containing the features describing the ith labeler, the symbol $\boldsymbol{\Phi}_{\boldsymbol{\beta_j}}$ to refer to a row vector containing the features describing the jth instance, and the symbol $\boldsymbol{\Phi}_{\boldsymbol{\gamma_i}}$ to refer to a row vector containing features that control the bias of the ith labeler. To refer to the collection of features for all instances or labelers the subscript is omitted (e.g., $\boldsymbol{\Phi}_{\boldsymbol{\alpha}}$ represents a matrix of all labeler features).

The weight matrices relating the latent factors to the features, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, are denoted by W_α and W_β, respectively. Additionally, the weight vector relating the bias features to the labeler bias is denoted by w_γ. Each of the weights is assumed to be normally distributed with known mean and variance. The instance factors, labeler factors, and labeler bias are modeled with the following linear forms: $\boldsymbol{\alpha_i} = \boldsymbol{\Phi}_{\boldsymbol{\alpha_i}} W_\alpha,\ \boldsymbol{\beta_j} = \boldsymbol{\Phi}_{\boldsymbol{\beta_j}} W_\beta,\ \gamma_i = \boldsymbol{\Phi}_{\boldsymbol{\gamma_i}} w_\gamma$.

Notably, the use of features does not reduce the flexibility of our model, as features can be introduced to adjust the accuracy of particular labelers or instances independently of all others (see Section 2.2).

2.2 Special Cases

Several models can be derived as special cases of the model proposed here. For instance the model of Whitehill *et. al.* [17] can be captured using 1 latent factor and setting both $\Phi_\alpha = I_m$ and $\Phi_\beta = I_n$ where we use I_k to indicate the identity matrix of size k. A very similar model to [15] can be recovered by parameterizing with unstructured interacting factors can be formulated using the same parameterization as just described for the GLAD model, but allowing multiple factors for both labelers and instances to be inferred. Additionally, the model due to Dawid and Skene [3] can be captured by associating one factor with each labeler and instance and by setting $\Phi_\alpha = I_m$ and $\Phi_\beta = E_n$ where E_n is an n by n matrix of all ones.

3 Inference

In this section we show how to find maximum *a posteriori* estimates of the model parameters using the Expectation Maximization algorithm [5]. The values of Z – the class labels of the data that the crowdsourcing was supposed to capture in the first place – will be estimated from the last iteration of the E-step using Equation 2. We proceed by deriving the E-step and M-step that allow us to maximize the posterior distribution of model parameters (W_α, W_β, and w_γ) given the data. For brevity we define the symbol $\boldsymbol{\Phi} \equiv (\boldsymbol{\Phi}_{\boldsymbol{\alpha}}, \boldsymbol{\Phi}_{\boldsymbol{\beta}}, \boldsymbol{\Phi}_{\boldsymbol{\gamma}})$ to refer to the collection of instance features, labeler features, and labeler bias features. We also define $W \equiv (W_\alpha, W_\beta, w_\gamma)$ to refer to the labeler factors weights, instance factors weights, and bias weights. Our goal is to find:

$$W^* = \arg\max_W p(W|L,\Phi) = \arg\max_W \sum_{\mathbf{Z}\in\{-1,1\}^n} p(\mathbf{Z})p(W|L,\mathbf{Z},\Phi)$$

3.1 E-Step

Recall that the set of all labels given to a specific instance j be denoted as $\mathbf{L}_j$. The goal of the E-step is to compute the posterior distributions of the hidden class labels given the current setting of the labeler factors, instance factors, and labeler biases.

$$\begin{aligned} p(z_j|\mathbf{L},W,\boldsymbol{\Phi}) &= p(z_j|\mathbf{L_j},W,\boldsymbol{\Phi}) \\ &\propto p(z_j|W,\Phi)p(\mathbf{L_j}|z_j,W,\boldsymbol{\Phi}) \\ &= p(z_j)\prod_i p(L_{ij}|z_j,W,\boldsymbol{\Phi}) \end{aligned} \tag{2}$$

3.2 M-Step

In the M-Step we maximize the expectation of the joint log likelihood of the observed data and the hidden variables given the parameter values where the expectation is with respect to the posterior probabilities of the hidden variables computed in the last iteration of the **E-step**. To simplify our derivation, let p_j^1 and p_j^0 represent the posterior probabilities of the jth instance being 1 or 0 as computed in the last E-step. The function we wish to maximize is called the Q-function. We provide a compact form for computing Q in the following equations:

$$\begin{aligned} Q(W) &= E\left[\ln p(\mathbf{L}, \mathbf{\Phi}, \mathbf{Z}|W)\right] \\ &= E\left[\ln\left(p(\mathbf{\Phi}|\mathbf{Z})\prod_j p(z_j)\prod_i p(L_{ij}|z_j, \mathbf{\Phi}, W)\right)\right] \\ &= \sum_{ij} E\left[\ln p(L_{ij}|z_j, \mathbf{\Phi}, W)\right] + \text{const} \end{aligned} \tag{3}$$

$$E[\ln p(L_{ij}|z_j, \Phi, W)] = p_j^1 \ln\left({P_{i,j}^1}^{L_{ij}}(1-P_{i,j}^1)^{(1-L_{ij})}\right) + p_j^0 \ln\left({P_{i,j}^0}^{(1-L_{ij})}(1-P_{i,j}^0)^{L_{ij}}\right) \tag{4}$$

Where, as defined in Section 2 the symbols P^0 and P^1 refer to the likelihood in Equation 1 in matrix form. The Q function can be maximized using a number of different methods. In this paper we employ the conjugate gradient ascent algorithm [6]. The gradient of Q with respect to the model parameters $(W_\alpha, W_\beta, w_\gamma)$ can be determined by applying the chain rule to Equation 4.

4 Experiments

Here we compare the performance of our model to the approach in [15], the GLAD model, as well as the Majority Vote strategy to predict the class label on two crowdsourcing tasks:

1. Facial Expression Labeling. We show that using features automatically extracted from an image of a face using expression analysis tools can increase the accuracy of inferred labels on the very difficult problem of discriminating social vs. genuine smiles.
2. Testing Users' knowledge of Geography. We show that a user's travel history as well as easily quantifiable attributes about a geography question can be used to more accurately determine geographical relationships. We also show that this task contains specialization where labelers with particular travel histories are adept at answering particular questions.

4.1 Facial Expression Labeling

We compare both the 1, 2, and 3 latent factor versions of our model to two alternatives on a facial expression labeling task, in particular, the discrimination between "Duchenne" and "Non-Duchenne" smiles. We structure the instance features for each face using an automatic smile detection system's output composed with a set of non-linear basis functions. The seven models considered are:

Proposed Model with {1,2,3} Latent Factor(s): this is the model proposed in this document with 1, 2, or 3 latent factors for each labeler and instance. **MultiDim {1,2} Latent Factor(s)**: this is the modeled proposed in [15]. The model learns to place the instances and labelers in either a 1 or 2 dimensional space. **GLAD**: this is the model in [17]. **Majority Vote**: this model assigns posterior probabilities based on the fraction of labels given to the instance of each class.

Each model is evaluated in terms of two metrics: (1) proportion of correctly inferred labels (i.e. the proportion match between the MAP estimated label of each instance and the true label) and (2) two-fold cross validation likelihood. The cross validation likelihood is a metric to quantitatively test whether model is inferring accurate structure in the task and is evaluated as follows: fit each model to half of the labels collected for each experiment, then the model's score is given by the log likelihood assigned to the given labels on the half of the labels unused for fitting the model parameters.

We used the the database of Duchenne smile versus Non-Duchenne smile experiment first reported on in [17]. The database consisted of 160 ground truth labeled images (as determined by two experts) labeled by a total of 20 different mechanical Turkers. Distinguishing a Duchenne smile ("enjoyment" smile) from a Non-Duchenne ("social" smile) has applications in various domains including psychology experiments, human-computer interaction, and marketing research. Reliable coding

Table 1: Proportion correctly inferred labels for the five models considered in this paper on the Duchenne vs. Non-Duchenne smile task. The 3 factor model is the best of all the models considered (shown in bold).

Method	**Prop. Correct 1-factor**	**Prop. Correct 2-factor**	**Prop. Correct 3-factors**
Proposed Model	.749	.772	**.775**
MultiDim [15]	.742	.749	See footnote.
GLAD [17]	.754	N/A	N/A
Majority Vote	.729	N/A	N/A

of Duchenne smiles is a difficult task even for certified experts in the Facial Action Coding System (who only agree about 80% of the time). For each experiment 20% of the labels in the database were removed to get a sense of the variability of the performance of the different models with different training sets. The facial expression recognition experiment was repeated a total of 100 times.

Since we had no demographic data from the labelers with which to infer structure, each labeler's accuracy and bias were parameterized independently. It is likely that there are visual features about each face that contribute to the difficulty of labeling the smile as Duchenne or Non-Duchenne. To test this hypothesis we parameterized the difficulty of each instance using an automatic smile detection system [16]. Additionally we allowed the instance factor estimates to have a non-linear relationship with the smile detector output (e.g. it is possible that an output near the mean is predictive of a more difficulty instance) by composing the smile detector output with a set of radial basis functions (see Figure 1), We then allowed then allowed our model to learn a linear relationship between the features in this space and the instance factors. In addition to the smile detector output each instance had a constant feature designed to allow the model to infer the baseline for each instance factor.

The feature weights, W_β, inferred by our single factor model are visualized in Figure 1. The learned weights indicate that instances with higher smile detector outputs (above the median smile value) are harder to label. A possible explanation is that some people can give the impression of an enjoyment smile simply by exaggerating the activation of a social smile (measured by the automatic smile detector), thereby confusing the Turk workers.

The performance for each model inferring the labels of Duchenne vs. Non-Duchenne is given in Table 1[1]. The 3-factor model performs the best of any of the model tested. It achieves a 2.1% increase in performance over the GLAD model. In terms of variability, the 3-factor model outperformed GLAD on 89% of the 100 randomly selected label sets (recall that 20% of the labels are omitted in each experiment). It is reasonable to suspect that there may be a ceiling effect at work in that expert coders only agree on the label of Duchenne vs. Non-Duchenne label 80% of the time.

In addition to comparing the accuracy of the inferred labels, we also compared our model against the approach in [15] on the cross validation log-likelihood metric. Our system achieves a mean log-likelihood (per label) of -0.4925. The 1-factor model of [15] achieves a mean log-likelihood of -0.5512, while the 2-factor version of that model appears to suffer from overfitting achieving a mean log-likelihood of -0.6755. To assess the impact of the smiledetector output on the ability to infer accurate label likelihoods we also evaluated our model without these features achieving a mean log-likelihood of -0.5403. The difference in mean log-likelihoods for our model with and without the smile detector output was significant ($p < .001$).

4.2 Testing Users' Knowledge of Geography

In this task we tested labelers on their ability to judge spatial relationships between two cities. Specifically, we asked subjects to determine which of two cities was either more westerly or more northerly in terms of longitude or latitude respectively (e.g. "Which city is more westerly: San Diego, California, USA or Reno, Nevada, USA"). We used Amazon's Mechanical Turk to collect 10 responses for each of 100 questions of this form. In addition to collecting labeler responses we also collected information on the travel history of the labelers.

We tested the proposed model by parameterizing the labeler latent factors with the number of continents the labeler visited over the course of his/her lifetime as well as a feature that was set to 1 for all

[1]Not easily performed using code provided by [15]

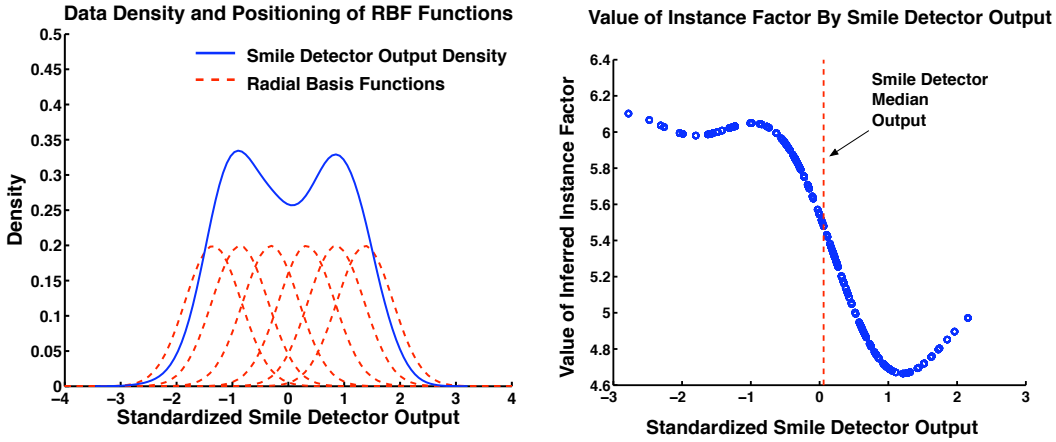

Figure 1: **Left:** The density of smile detector outputs along with the non-linear basis functions (RBFs) used to parameterize the instance factors. **Right:** The first instance factor as a function of detector output. Each point is a particular image from the Duchenne experiment.

labelers. The instance factors were parameterized by the absolute difference in latitude or longitude (whichever was relevant to the question) with the intuition that cities farther apart would be easier to distinguish for latitude/longitude. In addition, we included an instance feature that was set to 1 for all instances. The performance of our proposed model was compared against the alternative models on the area under the ROC metric. Our performance was the best, .9604 (compared to .9514 for MultiDim 1 factor, .8619 for MultiDim 2 factors (possibly due to overfitting), .9494 for GLAD, and .9492 for majority vote). Additionally, the model inferred, as expected, that instances that were more distant along the relevant geographical dimension were easier to label (an average increase in log odds of a correct response of 0.04 per degree of difference in either latitude or longitude depending on which dimension was relevant to the question) and also that labelers who had traveled to more continents were better at determining the correct answers.

Investigating multi-dimensional latent structure: While the 1-factor version of our model inferred interesting structure in the task and was able to use that structure to infer more accurate labels, we studied whether interaction effects between labelers and instances and the correctness probabilities could be inferred with a 2-factor model. In this experiment we were chiefly interested in determining whether there was specialization among the labelers for certain types of geography questions. Specifically, we wanted to know if a person having visited Europe would be more accurate at answering questions about the geographical relationship between two European cities than two cities from another continent. Hence, we added a new labeler feature encoding whether or not the labeler had ever visited Europe and a new instance feature indicating whether at least one of the response options for a question was a European city. First we trained a 1-factor model as before. We computed the log odds of a correct response for two types of labelers (one that had been to two continents including Europe, the other that had been to two continents not including Europe) and two types of instances (one in which the two cities were 5 degrees apart and at least one was in Europe and one that was 5 degrees apart with both cities not in Europe). The computed log odds are shown in Table 2. The 1-factor model inferred that labelers that had been to Europe were better at the task and questions about European cities were in general easier to answer. Since the model only had 1 latent factor for both labelers and instances the model had no ability to infer specialization.

Next we used our proposed model using two latent factors to see if any specialization could be inferred. The table of log odds given in Table 2 suggests that indeed the model is inferring a degree of specialization where labelers that had been to Europe were better than would have otherwise been expected for questions not about European cities. This can be seen by noticing that the increase in log-odds of correctness for a Europe traveler on a Europe question is greater than on a question not about Europe. Additionally, the performance of the two factor model improved slightly from the 1 factor model ($.9624$ area under the ROC vs. $.9604$)

Table 2: Increase in log-odds of a correct response for questions about at least one European city vs. those that did not include a European city for labelers that had been to Europe and who had not been to Europe.

Labeler Type	Increase in Log Odds (1-factor model)	Increase in Log Odds (2-factor model)
Been to Europe	.2378	.2811
Not Been to Europe	.2212	.2034

In addition to comparing the accuracy of the inferred labels, we also compared our model against the approach in [15] on the cross validation log-likelihood metric. Our system achieves a mean log-likelihood (per label) of $-.6416$. The 1-factor model of [15] did not achieve a good fit to the data mean log-likelihood of -0.6947. The 2-factor version of this same model achieved a mean log likelihood value of -0.6489. The higher mean log-likelihood shows that our method infers more accurate labeler response likelihoods than the MultiDim [15] model.

5 Related Work

A multitude of approaches to automatic quality control of crowd-sourced data based on the expectation maximization algorithm have been proposed (e.g. [3, 17, 9]). In contrast to our work, none of these approaches take into account the the multiple dimensions along which instances and labelers may vary.

Two approaches for learning multidimensional structure from the crowds have been proposed. The first is the model proposed in [15]. Our models were developed concurrently (see our tech report for an earlier version of our work [10]). They key difference between the approaches is that we allow the latent factors to depend on measured features of the instances and labelers through a linear transformation, whereas in [15] the labeler and instance factors are learned independently for each instance and labeler. While our model retains the flexibility of the approach in [15] (since we can parameterize each labeler and instance independently) by including the option to impose structure on the learned factors through measured features, our model in principle could infer structure that is both more interpretable (since it is based on clearly defined features) and more effective (since it takes into account prior knowledge of factors likely correlated with accuracy and difficulty) using less data. A more recent approach for learning multidimensional structure in crowdsourced labels is [7] in which a personal classifier is used to model the labeling decision of each labeler. This model has two critical limitations in comparison to ours. Firstly, it does not allow the direct modeling of the probability of correctness (which is the most natural way to express concepts such as trickiness and specialization). Secondly, it does not allow for the inference of instance factors, instead requiring that these factors be specified by the user.

6 Conclusion

Quality control of the labeling process is likely to become an even more crucial issue in the future as the scale of database collection and consumption by machine learning algorithms increases. As more data becomes available the more potential there is for inferring rich latent structure in among labelers and instances. In this document we showed that this structure is found in real crowdsourcing tasks. Examples of this structure include interaction effects (e.g. specialization of certain labelers to particular questions) and commonalities among data instances and labelers that can be predicted based on prior information.

We presented a new model for uncovering such structure by using labeler and instance features to parameterize labeler factors, labeler biases, and instance factors in order to infer latent class without the use of using a pre-labeled subset of ground truth data. We then showed how to exploit this structure to achieve more accurate class labels. Our work also unifies previous work by casting both the GLAD model [17] and Dawid and Skene's model [3] in a common framework. We provided two experiments that validate the power of this new model for improving quality control of large databases. Finally, we showed that our proposed method is useful for uncovering interesting associations between particular features of instances and labelers that influence the probability of correctness.

References

[1] Amazon. Mechanical turk, 2005. http://www.mturk.com.

[2] L. Barrington, D. O'Malley, D. Turnbull, and G. Lanckriet. User-centered design of a social game to tag music. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 7–10. ACM, 2009.

[3] A. Dawid and A. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics*, 28(1):20–28, 1979.

[4] O. Dekel and O. Shamir. Good learners for evil teachers. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(Series B):1–38, 1977.

[6] R. Fletcher and C. Reeves. Function minimization by conjugate gradients. *The computer journal*, 7(2):149–154, 1964.

[7] H. Kajino, Y. Tsuboi, and H. Kashima. A convex formulation for learning from crowds. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2012.

[8] Omron. OKAO vision brochure, July 2008.

[9] V. Raykar, S. Yu, L. Zhao, A. Jerebko, C. Florin, G. Valadez, L. Bogoni, and L. Moy. Supervised Learning from Multiple Experts: Whom to trust when everyone lies a bit. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM New York, NY, USA, 2009.

[10] P. Ruvolo, J. Whitehill, and J. Movellan. Exploiting structure in crowdsourcing tasks via latent factor models. Tech report, Machine Perception Laboratory, 2010.

[11] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20:1257–1264, 2008.

[12] V. Sheng, F. Provost, and P. Ipeirotis. Get another label? improving data quality and data mining using multiple noisy labelers. In *Knowledge Discovery and Data Mining*, 2008.

[13] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods on Natural Language Processing*, 2008.

[14] L. von Ahn and L. Dabbish. Labeling Images with A Computer Game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM Press New York, NY, USA, 2004.

[15] P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems*, 2010.

[16] J. Whitehill, G. Littlewort, I. Fasel, M. Bartlett, and J. Movellan. Toward practical smile detection. *Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[17] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems 22*, pages 2035–2043. 2009.