Exploiting Structure in Crowdsourcing Tasks via Latent Factor Models

Paul Ruvolo, Jacob Whitehill, Javier R. Movellan
Machine Perception Laboratory
UC San Diego, CA

Abstract

Internet crowdsourcing services such as the Amazon Mechanical Turk (1) and the ESP Game (15) have become important tools for the machine learning community by facilitating the distributed labeling of large datasets at little cost. A key challenge when using crowdsourcing to label databases is the need to derive high quality labels by aggregating the responses from labelers of varying reliability over data instances of varying difficulty. Existing algorithms for quality control and label inference (14; 17; 10) suffer several significant shortcomings: (1) Existing methods are incapable of modeling interaction effects between labeler and data items, such as when some labelers have specialized knowledge about a particular subset of items. (2) Existing algorithms assume that labelers' accuracies, as well as data instances' difficulties, are independent. In reality, there may be *a priori* information about labelers (or data instances) that predicts those labelers' accuracy at the labeling task. Analogously, certain features shared among data instances may predict their difficulty of being labeled correctly. In this paper, we present an algorithm that addresses both of these shortcomings. We demonstrate that the proposed algorithm delivers superior accuracy, compared to previous methods, of inferring data labels on a difficult facial expression labeling task. Finally, we show that our proposed model subsumes certain previous models as special cases.

1 Introduction

As machine learning applications tackle more and more difficult problems, the importance of large-scale and varied datasets of high-quality training data becomes more apparent. For instance, the Omron face detector, which represents the current state of the art of face detection, was trained using millions of hand-labeled images (9). 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 (15).

As crowdsourcing tools grow in popularity, algorithms have been developed to assess the quality of, and optimally combine, the labels obtained from such services (14; 13; 17; 4; 10). The high-level premise is that a latent, binary class label Z (e.g., a face is Smiling or Not Smiling for a facial expression labeling task) must be inferred for each data instance (e.g., an image) using the binary labels L obtained from a set of labelers. Many of the existing approaches to this task share a common, iterative 2-step architecture based on the Expectation-Maximization algorithm:

1. Estimate the probability that labeler i's opinion L_{ij} of the binary class of data instance j matches the "true" class label Z_j. (For the rest of this document we refer to this quantity as the "correctness probability".)

2. Use Bayes' rule to compute the posterior distribution over the true latent class label Z_j of each data instance given the observed labels $\{L_{ij}\}$ by assuming that each L_{ij} is generated independently given the correctness probabilities determined in step 1.

These algorithms vary chiefly in the particular dimensions of variability that are considered when estimating the correctness probability for a specific labeler on a specific instance. 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) improves on Dawid and Skene's model by also modeling a latent difficulty attribute associated with each data instance.

Latent Factors to predict the correctness probability: While labeler ability and instance difficulty are arguably important factors, many other factors may exist that previous models ignore. 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. In this work, we present a model of label inference that offers much more flexibility than previous approaches. Two examples of important factors include *trickiness* and *specialization*. A *tricky* instance is one where a moderately skilled labeler will be duped into choosing the wrong class label whereas a more naive labeler would be more likely to choose the correct label by simply guessing. The trickiness phenomenon cannot be captured by previous models because they assume the probability of labeling an instance correctly is monotonic in the labeler's ability. *Specialization* is when labelers are apt at labeling a certain subset of data instances. 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. Previous approaches cannot model this because they assume each labeler has only 1 latent accuracy attribute that applies to all instances equally.

In the current work we allow the correctness probabilities to depend on a larger number of latent labeler and instance factors. These factors do not necessarily have high-level semantic meaning (e.g. difficulty or accuracy); however, they are capable of capturing a wide range of dimensions of variability in the correctness probabilities (including both trickiness and specialization).

Features shared across labelers, and instances: While richer models of the labeling process are attractive due to their ability to more completely represent instance-labeler interactions, increasing the model complexity also runs the risk of overfitting. However, in many crowd-sourcing applications this problem can be mitigated by incorporating *a priori* information about labelers and instances that are correlated with the correctness probabilities. Instead of modeling an additional parameter for each additional labeler or each additional data instance, the latent factors described above could instead be functions of *features* associated with those labelers or data instances. For example, for a particular task it may be that women are more accurate than men and therefore the correctness probability estimate for a previously unseen female labeler should be higher than for a previously unseen male labeler. Our algorithm is able to take into account this type of information (e.g., the gender of the labeler) in order to provide accurate estimates of the correctness probabilities while avoiding overfitting.

2 Modeling the Labeling Process

We wish 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. Each labeler may label a variable number of instances; there is no requirement that each labeler label every data instance. A high-level picture of the characteristics of our model of the labeling process 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 (described in Section 2.1).

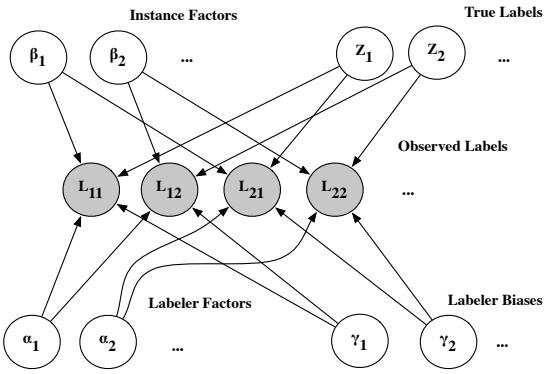

Figure 1: The graphical model of image factors, true image labels, labeler biases, observed labels, and labeler factors is shown. The use of "..." indicates that more variables can occur with each having similar connectivity to the variables pictured. Shaded nodes are random variables whose values are observed (see Section 3).

2. Each latent factor is modeled as a linear function of a known set of features (e.g. demographic information about the labeler or metadata about an image) and an inferred set of weights (described in Section 2.2).
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).

2.1 Modeling the Correctness Probabilites

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. 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. Figure 1 shows the causal structure of the model. The labeler latent factors, instance latent factors, labeler bias, and the true label each have an influence on the observed labels.

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 (denoted as 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 2.3). Second, 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 factor contains the labeler factors and one factor contains the instance factors). This is a view that has been shown to be effective in the field of collaborative filtering (12).

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:

$$P^1 = \sigma\left(\boldsymbol{\alpha}\boldsymbol{\beta}^\top + \boldsymbol{\gamma}e^\top\right) \quad (2)$$

$$P^0 = \sigma\left(\boldsymbol{\alpha}\boldsymbol{\beta}^\top - \boldsymbol{\gamma}e^\top\right) \quad (3)$$

where e is an n dimensional vector of all ones.

We illustrate the utility of this model with an example: Consider a labeling problem with two types of labelers. The first type of labeler is a "generalist" who achieves an 80% accuracy on all instances. The second type labeler is a "specialist" who achieves a 95% accuracy on data instances of his/her specialization, and 70% accuracy on all other instances. For simplicity of exposition, all labelers in the example are assumed to have zero bias. The proposed model can model these accuracy probabilities using a model with two latent factors. The specific values for the latent factors are given in the following table:

Labeler Type	α_1	α_2	Instance Type	β_1	β_2
Specialist	.848	1	Specialist Question	1	2.094
Generalist	1.386	0	Generalist Question	1	0

It is easy to see that the factors in the table above produce the proper correctness probabilities by applying Equation 1.

	Specialist Question	Generalist Question
Specialist	$\sigma(.848 \times 1 + 1 \times 2.094) = .95$	$\sigma(.848 \times 1 + 1 \times 0) = .7$
Generalist	$\sigma(1.386 \times 1 + 0 \times 2.094) = .8$	$\sigma(1.386 \times 1 + 0 \times 0) = .8$

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 some known features that help predict who is a specialist then we can incorporate that information into our model to make more efficient inference of structure within the labeling task.

2.2 Parameterizing Latent Factors of Labelers and Instances

In order to exploit attributes shared across multiple labelers and across multiple data instances, we model the latent label factors $\boldsymbol{\alpha}$ and latent instance factors $\boldsymbol{\beta}$ as a linear function of a specified set of features and an unknown set of weights. The causal structure of the relationships between features, weights, and the latent dimensions of correctness probability is given in Figure 2. 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 latent instance factors, latent labeler factors, and labeler bias are modeled with the following linear forms:

$$\boldsymbol{\alpha_i} = \boldsymbol{\Phi}_{\boldsymbol{\alpha_i}} W_\alpha \quad (4)$$

$$\boldsymbol{\beta_j} = \boldsymbol{\Phi}_{\boldsymbol{\beta_j}} W_\beta \quad (5)$$

$$\gamma_i = \boldsymbol{\Phi}_{\boldsymbol{\gamma_i}} w_\gamma \quad (6)$$

It is important to note that the use of features does not reduce the flexibility of our proposed model, as features can be introduced to adjust the accuracy of one particular labeler or instance independently of all others (see Section 2.3).

2.3 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 $\Phi_\beta = I_n$

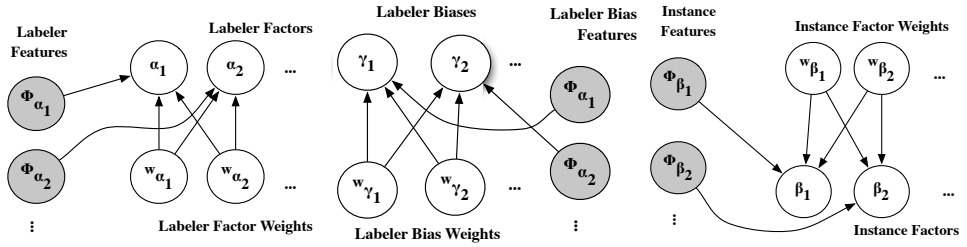

Figure 2: The latent structure of the labeler factors, biases, and image factors is given. The features are assumed to be given (indicated using shading) to the inference algorithm which then infers the weights given to each of these features.

where we use I_k to indicate the identity matrix of size k. 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). 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 $\mathbf{\Phi} \equiv (\mathbf{\Phi_\alpha}, \mathbf{\Phi_\beta}, \mathbf{\Phi_\gamma})$ to refer to the collection of image difficulty features, labeler accuracy 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:

$$W^* = \arg\max_W p(W|L, \Phi) \tag{7}$$

$$= \arg\max_W \sum_{\mathbf{Z}\in\{-1,1\}^n} p(\mathbf{Z})p(W|L, \mathbf{Z}, \Phi) \tag{8}$$

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.

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

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

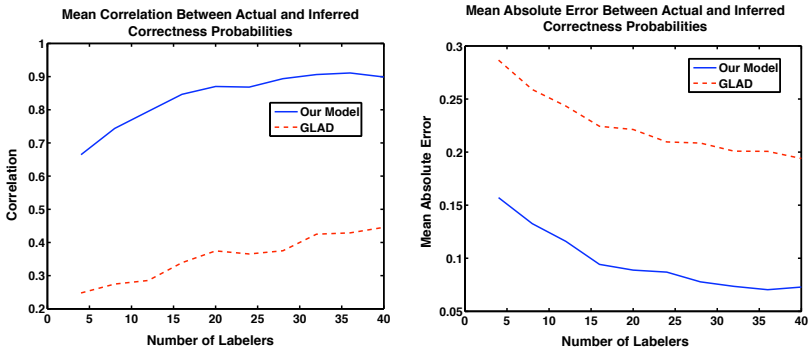

Figure 3: **Left:** the correlation of real and inferred label accuracies as a function of the number of labelers who labeled on average 40% of the 40 total instances. **Right:** the mean absolute error between the inferred and actual label accuracies as a function of the number of labelers who labeled on average 40% of the 40 total instances.

a compact form for computing Q in the following equations:

$$\begin{aligned} Q(W) &= E\left[\ln p(\mathbf{L}, \mathbf{\Phi}, \mathbf{Z}|W)\right] \\ &= E\left[\ln\left(p(\mathbf{\Phi}|\mathbf{Z})\prod_j p(z_j)\prod_i p(L_{ij}|z_j, \mathbf{\Phi}, W)\right)\right] \\ &= \sum_{ij} E\left[\ln p(L_{ij}|z_j, \mathbf{\Phi}, W)\right] + \text{const} \end{aligned} \tag{9}$$

$$\begin{aligned} E[\ln p(L_{ij}|z_j, \Phi, W)] &= p_j^1 \ln\left({P_{i,j}^1}^{L_{ij}}(1-P_{i,j}^1)^{(1-L_{ij})}\right) + \\ &\quad p_j^0 \ln\left({P_{i,j}^0}^{(1-L_{ij})}(1-P_{i,j}^0)^{L_{ij}}\right) \end{aligned} \tag{10}$$

Where as defined in Section 2.1 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 10.

4 Simulations

Here we explore the performance of our model using a set of data drawn from the generative process of the model itself. In this case we know the true correctness probabilities for each labeler-instance combination and thus can compare the correctness probability estimates generated using various models with the true values.

For each simulation we assume 1 latent factor for each of the labelers and 1 latent factor for each of the instances. The labelers are assumed to be unbiased. For each experiment we create a set of labelers (ranging from 4 to 40) with latent factors determined based on a set of randomly generated features, $\mathbf{\Phi}_{\boldsymbol{\alpha_i}}$, and randomly generated weights, W_α. Each vector $\mathbf{\Phi}_{\boldsymbol{\alpha_i}}$ has five dimensions where the first four are labeler specific and generated from a normal distribution with mean zero and .5 standard deviation and the last dimension was constrained to always be 1. The feature weights W_α were also five dimensional where the first 4 dimensions were generated from a normal distribution with mean zero and .5 standard deviation and the last dimension was generated from a normal distribution with mean 1 and standard deviation .5 (the mean of 1 for the last dimension ensures that generally labelers are non-adversarial). The instance features and factor weights were generated identically as in the case of the labelers. For each (labeler, instance) combination there was a 40% probability of a response being generated (thus not all labelers labeled each instance). If a label was generated for a specific combination of labeler and instance then it was generated according to the likelihood in Equation 1.

Both the model due to Whitehill *et. al.* (GLAD) and our model were evaluated on two metrics. First, we correlated each model's concatenated estimates of the correctness probabilities for class 0 and class 1 (i.e. P^0 and P^1) for each labeler instance combination with the true values. Second, we computed the mean absolute deviation between the model's estimates of P^0 and P^1 and the true values. All simulations were repeated 100 times. The results are given in Figure 3. Notice how the proposed model is able to exploit the structural information of the features to provide more efficent estimates of the correctness probabilities than GLAD.

5 Experiments in Facial Expression Labeling

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

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 proportion of correctly inferred labels. This metric was measured by computing the proportion match between the MAP estimated label of each instance and the true label.

The authors of (17) allowed us to use the database of Duchenne smile versus Non-Duchenne smile experiment reported in their paper. The database consisted of 160 ground truth labeled images (as determined by two experts) labeled by a total of 20 different mechanical Turkers. A Duchenne smile ("enjoyment" smile) is distinguished from a Non-Duchenne ("social" smile) through the activation of the *Orbicularis Oculi* muscle around the eyes, which the former exhibits and the latter does not. Distinguishing the two kinds of smiles has applications in various domains including psychology experiments, human-computer interaction, and marketing research. Reliable coding 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). It may be important to allow 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). To accomplish this we passed the smile detector output through a set of radial basis functions (see Figure 4) to create a non-linear basis and 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 4. 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.

Ultimately we are interested in the performance of our method in inferring correct judgments of Duchenne vs. Non-Duchenne. The accuracy for each method is given in Table 1. The 3-factor model performs the best of any 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

Method	Proportion Correct
Proposed Model with 1 Latent Factor	.749
Proposed Model with 2 Latent Factors	.772
Proposed Model with 3 Latent Factors	**.775**
GLAD	.754
Majority Vote	.729

Table 1: Model accuracy and area under the ROC (A') 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).

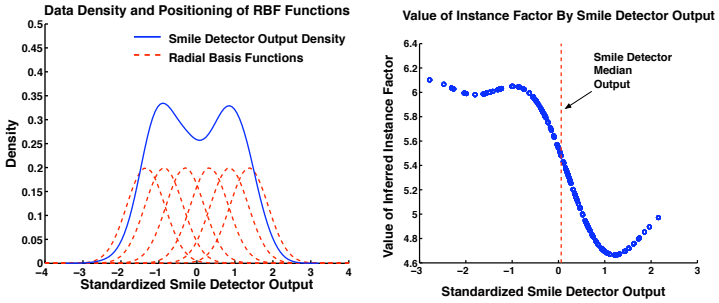

Figure 4: **Left:** The density of smile detector outputs along with the non-linear basis functions (RBFs) used to parameterize the instance factors. **Right:** The first instance factor as a function of detector output. Each point is a particular image from the Duchenne experiment. In this case the model learns that the most difficult images are those that have high smile detector output values.

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.

6 Related Work

In addition to the two point-based maximum likelihood approaches discussed in the Introduction (3; 17) a more recent approach though closely related approach (10) allows the joint training of a logistic regression classifier during the quality control process. It would be a straightforward exercise to add this feature to model presented here. In addition to these maximum likelihood approaches there approaches where posterior inference is based on representing the posterior over model parameters using a collection of samples (e.g. (8; 7; 11)). However, inference in these models is based carried out using variants of MCMC which is likely to suffer from high computational expense, and the need to wait (arbitrarily long) for parameters to "burn in" during sampling.

7 Conclusion

Quality control against adversarial and noisy labelers 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.

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 prelabeled 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 provide a simulation result and an experiment that validate the power of this new model for improving quality control of large databases.

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] V. Johnson. On bayesian analysis of multi-rater ordinal data: An application to automated essay grading. *Journal of the American Statistical Association*, 91:42–51, 1996.

[8] G. Karabatsos and W. H. Batchelder. Markov chain estimation for test theory without an answer key. *Psychometrika*, 68(3):373–389, 2003.

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

[10] 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.

[11] S. Rogers, M. Girolami, and T. Polajnar. Semi-parametric analysis of multi-rater data. *Statistics and Computing*, 2009.

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

[13] 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.

[14] 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.

[15] 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.

[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. fan 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.