Nonparametric Bayesian inference of item-level features in classifier combination

Patrick Stinson[1] **Nikolaus Kriegeskorte**[1]

[1]Zuckerman Institute, Columbia University

Abstract

In classification tasks, examples belonging to the same class can often still differ substantially from one another, and being able to capture such heterogeneity and its impact on classification can be important for aggregating estimates across multiple classifiers. Bayesian models developed so far have relied on a fixed set of latent variables to model these causal factors, which not only introduces the need for model selection but also assumes that each item is governed by the same set of causal factors. We develop a Bayesian model that can infer generic item features by modeling item feature membership as distributed according to an Indian Buffet Process. Despite its flexibility, our model is scalable to a large number of classifiers and examples. We compare our method with models from item response theory and Bayesian classifier combination on black-box crowdsourcing tasks and with neural network instance-dependent models in white-box classifier combination tasks.

1 INTRODUCTION

In classification problems, better performance can often be achieved by combining the predictions of an ensemble of classifiers [Dietterich, 2000, Yuksel et al., 2012] with the underlying intuition that different models may be responsive to different features in a given dataset and the set of predictions from an ensemble of models provides a more comprehensive statistic from which inferences can be done.

Crowdsourcing [Howe, 2006, Callison-Burch and Dredze, 2010] outsources a wide range of problems to humans, many of them requiring either special expertise or reasoning whose nature and scope can be difficult for algorithms to accurately capture (see e.g., [Budd et al., 2021]), especially if such examples aren't well-represented in the training data. Estimates among groups of classifiers, both human and machine, are often correlated (conditional on the ground-truth label) despite the classifiers operating independently [Kim and Ghahramani, 2012, Trick and Rothkopf, 2022] because items with the same ground-truth label are usually still not homogeneous: different items will often have different features which will give rise to different classification patterns that will coincide among classifiers, making it appear as if the classifiers are statistically dependent. As a simple example, a more difficult item will be classified correctly less often than an easier item. A model that does not differentiate among these different latent item features effectively marginalizes over them, thus correlating classifications which when otherwise conditioned on the correct latent variables would be independent.

Fully Bayesian black-box methods exist for modeling the resulting statistical dependencies among the classifiers while keeping items homogeneous [Kim and Ghahramani, 2012, Moreno et al., 2015, Li et al., 2019, Trick and Rothkopf, 2022], but one should expect modeling these marginal distributions to be suboptimal as item-specific information is lost and all items are treated the same when they are not. For example, the classifier outputs for difficult items would be the same as for easy items as long as they belong to the same class. Moreover, these methods require specifying a generative model of (and inferring) these dependencies, which could be complex and without a straightforward closed form approximation, as they arise from marginalizing over an unknown (and likely variable) set of random variables.

Item response theory (IRT) [Lazarsfeld, 1950, Rasch, 1960, Lord et al., 1968, Baker and Kim, 2004] was developed to measure specific latent traits such as ability or attitude in individuals based on their performance on tests whose items were assumed to possess specific latent features such as difficulty or discriminability. Thus, a Bayesian treatment of IRT models provides a means to heterogeneous item methods, and one of our contributions is generalizing the $\{1, 2, 3\}$-PL (parameter logistic) IRT models to classification tasks of higher arity and evaluating their performance on simulated

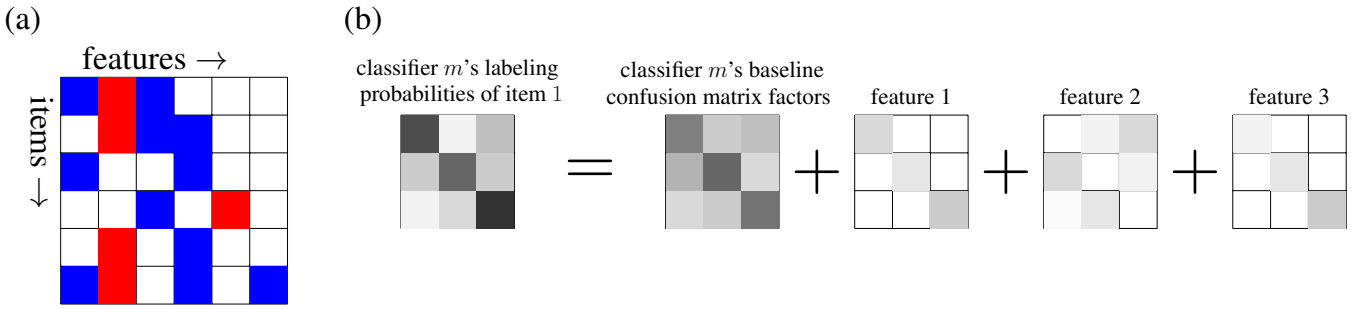

Figure 1: Illustration of our proposed method, item-dependent BCC (idBCC). (a). The presence of shared latent features that can either increase classification accuracy (blue) or decrease classification accuracy (red) is inferred for every item. (b) The confusion matrix of each classifier conditioned on the item is the softmaxed sum of the baseline factors unique to classifier m that determine its classification probabilities in the absence of any latent features (its confusion matrix is just the softmax of those factors over each row) and the (classifier-specific) inferred effects of each latent feature belonging to the item. Conditioning on the label of the item selects a row of the confusion matrix representing the classifier's labling probabilities (features can be shared across items of potentially different ground-truth label).

data and crowdsourcing benchmarks. A potential issue with IRT models, however, is that the number latent features and how they interact is determined a priori instead of inferred from the data and that each item is assumed to possess the same set of latent features.

Item-dependent methods based on neural networks [Rodrigues and Pereira, 2018, Guo et al., 2023, Li et al., 2024] have been developed, but these methods require direct access to the item, which although straightforward in certain tasks such as image recognition, a numerical representation of an item, such as a patient's entire medical chart, or all the information necessary to integrate to make a forecast (along with the appropriate architecture to transform this data into a representation), can be difficult to determine. Additionally, many neural network models require large numbers of data points to be trained adequately, which limits their scope of use.

To address these limitations, we propose a Bayesian nonparametric model that infers black-box item-level features. We model these item-specific feature membership as distributed according to an Indian Buffet Process [Griffiths and Ghahramani, 2005, 2011], so we do not place a priori assumptions on the number of item-specific latent factors and do not assume which items possess which factors. For each factor, we infer a classifier-specific effect, so we do not place a priori assumptions on how each latent factor affects classification (which can vary among classifiers). We compare our model with the competitors on simulated data, black-box crowdsourcing benchmarks, and white-box classifier combination tasks.

2 RELATED WORK

2.1 ITEM RESPONSE THEORY

Item response theory (IRT) [Lazarsfeld, 1950, Rasch, 1960, Lord et al., 1968, Baker and Kim, 2004] models the responses of test takers to items in a test using a few handcrafted parameters, namely the ability of the test taker (in terms of sensitivity $\alpha^{(1)}$ given an item whose ground-truth value is True or specificity $\alpha^{(0)}$ given an item whose ground-truth value is False), the difficulty of the item β, the discriminability of the item γ, and the guessability of the item λ. Indexing the items with n and test takers with m, we have

$$P(x_n^{(m)} = t_n|\theta, t_n) = \lambda_n + (1 - \lambda_n)\sigma(\gamma_n(\alpha_m^{(t_n)} - \beta_n)), \tag{1}$$

where $\theta = \{\alpha, \beta, \gamma, \lambda\}$ and t_n is the ground-truth label of item n. The binary model is generalized to L-ary when $t_n \in \{1, ..., L\}$ with the probability of being incorrect split evenly among the alternative $L - 1$ choices. Thus, a limitation of IRT extended to L-ary classification problems is that the off-diagonal entries of the corresponding confusion matrix $P(x_n^{(m)} = l'|t_n = l)$ are all equal, so the model cannot learn label- or item-conditional class-dependent misclassification rates; e.g., in digit recognition, the model probability for misclassifying an instance of a 1 for a 7 must be the same as that for misclassifying it for an 8.

Equation (1) is called the 3 parameter logistic (3-PL) model. The 2-PL and 1-PL models can be recovered by setting $\lambda_n = 0$ and both $\lambda_n = 0, \gamma_n = 1$, respectively. Bayesian treatments of IRT include Whitehill et al. [2009], Trick et al. [2023] using 1-PL and Han et al. [2024] using various combinations of parameters but evaluating models on two binary labeling tasks separate from crowdsourcing benchmarks.

2.2 BLACK-BOX INDEPENDENT AND DEPENDENT BAYESIAN CLASSIFIER COMBINATION MODELS

IBCC [Kim and Ghahramani, 2012] models each classifier's labeling probability of an item independently based on the underlying (inferred) ground-truth label of the item:

$$x_n^{(m)}|t_n = l \sim \text{Cat}(\pi_{l,\cdot}^{(m)}),$$

where m and n index classifiers and items, respectively, l indexes the ground-truth label, and $\pi^{(m)}$ represents the mth classifier's confusion matrix whose rows are each given a Dirichlet prior:

$$\pi_{l,\cdot}^{(m)} \sim \text{Dir}(\alpha_{l,\cdot}^{(m)}),$$

where

$$\alpha_{l,l'}^{(m)} \sim \text{Exp}(\lambda \mathbb{I}(l = l') + \lambda' \mathbb{I}(l \neq l')),$$

and $\lambda < \lambda'$ is set to reflect an inductive bias that classifiers are better than chance level.

Kim and Ghahramani [2012] also propose a dependent model in which a Markov network models the label-conditional dependencies between each pair of classifiers; however, the model requires computing a partition function and does not scale well to large numbers of classifiers.

Li et al. [2019] develop a variational Bayesian method, EBCC, that approximates this dependency matrix using a low-rank tensor decomposition.

Clustering based BCC (cBCC) [Moreno et al., 2015] uses a Chinese Restaurant Process [Ferguson, 1973, Blackwell and Macqueen, 1973, Teh, 2010] prior to infer a nonparameteric clustering of classifiers. For each classifier in each cluster, the confusion matrices are the same. In a hierarchical version, the intra-cluster classifiers' confusion matrices are distributed according to the same distribution.

2.3 WHITE-BOX ITEM DEPENDENT MODELS

In contrast to black-box models which can be used in any crowdsourcing or classifier combination task, more recent white-box models use neural networks to transform the data underlying a given item (for example, the image in an image recognition task) into a representation that can be used to relate features in a given data point to the labels the classifiers assign to the item. White-box models are therefore limited in scope of use, as some classification tasks cannot easily be represented as a numerical input, and neural network models often cannot be trained well on limited amounts of data.

CrowdLayer Rodrigues and Pereira [2018] learns a simple mapping, such as a linear or affine transformation, from the bottleneck layer of a neural network to each classifier's confusion matrix parameters.

IDNT Guo et al. [2023] uses neural networks to learn separate nonlinear representations of both the classifier's expertise and the features of the item as a function of the item and determines labeling predictions using Bayesian linear regression with a spike-and-slab weight prior.

TAIDTM Li et al. [2024] learns an annotator adjacency graph which is transformed by a graph convolutional network Kipf and Welling [2017] into item-dependent parameterizations for each classifier.

3 MODEL

Our model, which we call idBCC for item-dependent Bayesian Classifier Combination, infers a binary feature membership matrix, $V \in \{0, 1\}^{N \times K}$ paired with each feature's inferred effects on each classifier $\{U_{m,k}\}_{m \in [M], k \in [K]}$ whose dimensionality K is dynamic during inference via the Indian Buffet Process (IBP) Griffiths and Ghahramani [2005, 2011].

We illustrate the idea behind our model in Figure 1. Each item is associated with a set of latent features (including potentially none) represented by 1s in the corresponding row of V. The combination of a given item's latent features, the effects each feature has on each classifier, the (inferred) ground-truth label of the item, and the classifiers' ground-truth label-conditional rating probabilities (i.e., each of their baseline confusion matrices) gives the labeling probabilities of that item for the classifiers.

The IBP is a stochastic process that defines a probability distribution over binary matrices with an infinite number of columns (with only a finite number of columns containing 1s). Modeling feature membership as a realization of an IBP, we are able to infer a variable and unbounded number of causal factors for each item, in contrast to existing models whose causal factors are generally fixed in number and whose number cannot vary across items.

The IBP prior can be derived by first fixing the number of features/columns K and using a Beta-Bernoulli model to generate a $N \times K$ binary matrix:

$$\theta_k \sim \text{Beta}(\alpha/K, 1) \tag{2}$$

$$V_{n,k}|\theta_k \sim \text{Bern}(\theta_k), \tag{3}$$

where α controls the row and column sums of V. Integrating out θ_k gives

$$P(V) = \prod_{k=1}^{K} \frac{\frac{\alpha}{K}\Gamma(N_k + \alpha/K)\Gamma(N - N_k + 1)}{\Gamma(N + 1 + \alpha/K)}, \tag{4}$$

where N_k is the row sum of column k. Taking the limit $K \to \infty$ and arranging the columns in a particular way (see

Griffiths and Ghahramani [2011] for more details), we get

$$P(V) = \frac{\alpha^{K_+} \exp(-\alpha H_N)}{\prod_{h=1}^{2^N-1} K_h!} \prod_{k=1}^{K_+} \frac{(N-N_k)!(N_k-1)!}{N!}, \tag{5}$$

where K_+ is the number of nonzero columns in V, K_h is the number of columns whose entries match the index h expressed as a binary number, and $H_N := \sum_{i=1}^{N} \frac{1}{i}$.

In contrast with models with handcrafted features, for example in the IRT model described in Equation (1), we do not specify a priori how each learned feature impacts the classification probabilities. Instead, in our model, the kth feature has an impact on each classifier that is represented by the matrix $U_{m,k} \in \mathbb{R}_+^{L \times L}$, which can be thought of as an unnormalized confusion matrix factor. Our model's prediction of the mth classifier's labeling of the nth item is determined by softmaxing the sum of the unnormalized confusion matrix factors for classifier m corresponding to the features present in the nth item:

$$x_n^{(m)} | U, V, t_n \sim \text{Cat}(\text{softmax}((\sum_{k=0}^{K} U_{m,k,t_n,l'} V_{n,k})_{l'=1}^{L})). \tag{6}$$

We reserve a bias term for $k = 0$ such that $V_{\cdot,0} = 1$, which equips our model with the standard item-independent baseline confusion-matrix parameterization, on top of which item dependencies can be learned when $K_+ > 0$. Thus, our model can be considered a generalization of IBCC (although our effective prior over confusion matrix rows is not Dirichlet) as well as 1-PL.

To avoid potentially learning spurious features, we place some constraints on the form $U_{m,k}$ can take. Each matrix is constrained to be nonnegative to avoid learning matrices that cancel each other out. Furthermore, since the softmax function is invariant to adding a constant to each term, we constrain the form each $U_{m,k}$ can take to ensure each learned feature has an impact on classification probabilities. This can be achieved by enforcing an inductive bias that each feature has either a positive or negative effect on each classifier's accuracy. For example, to refer to a toy example illustrated in Section 5, a handwritten digit drawn thinly such that the digit's edges don't activate convolutional filters as well as those with thicker edges should give rise to a negative classification accuracy effect regardless of the digit being drawn or the particulars of the specific classifier architecture being used. Thus, the inductive bias is that the feature's effect sign on classification, i.e., whether it is positive or negative, is invariant with respect to the particular item or classifier.

Introducing an indicator variable $s_k \in \{+, -\}$ that indicates a positive/negative feature, a prior over $U_{m,k}$ that satisfies these constaints is

$$U_{m,k>0,l,l'} | s_k \sim \begin{cases} \mathbb{I}(l \neq l')\delta(0) + \mathbb{I}(l = l')\mathcal{N}_+(0, v), s_k = + \\ \mathbb{I}(l = l')\delta(0) + \mathbb{I}(l \neq l')\mathcal{N}_+(0, v), s_k = -, \end{cases} \tag{7}$$

where $\mathcal{N}_+(\cdot, \cdot)$ is a nonnegative (truncated) normal distribution. $U_{\cdot,0}$ corresponds to each classifier's item-independent label-conditional rating (unnormalized log-) probabilities, analogous to π in IBCC, for which on each entry we place a $\mathcal{N}(0, v)$ prior.

Sometimes in our exposition, a clearer notation is to separate U into three separate matrices: $U_{\cdot,0}$, $U^{(\text{pos})} := \{U_{\cdot,k}\}_{\{k:s_k=+\}}$, and $U^{(\text{neg})} := \{U_{\cdot,k}\}_{\{k:s_k=-\}}$, and we do the same for the corresponding binary feature variables: $V^{(\text{pos})} := \{V_{\cdot,k}\}_{\{k:s_k=+\}}$, $V^{(\text{neg})} := \{V_{\cdot,k}\}_{\{k:s_k=-\}}$.

Positive and negative features are distributed according to separate Indian Buffet Processes:

$$V^{(\text{pos})} \sim \text{IBP}(\alpha^{(\text{pos})}),$$
$$V^{(\text{neg})} \sim \text{IBP}(\alpha^{(\text{neg})}).$$

We put an inverse-gamma prior on the variance

$$v \sim IG(\alpha_v, \beta_v)$$

and a prior on the ground-truth labels

$$t_n \sim \text{Cat}(\kappa).$$

Our full model is shown in Figure 2.

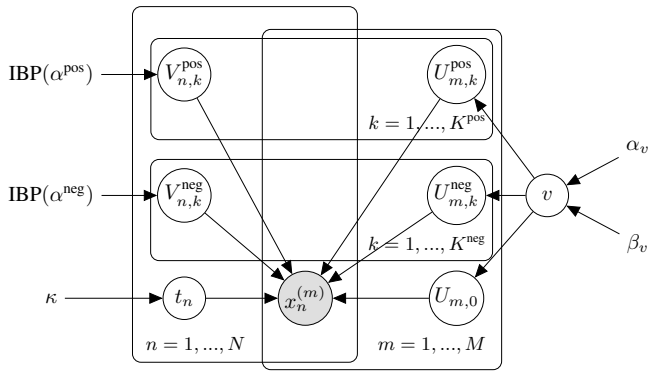

Figure 2: Plate notation of idBCC.

4 INFERENCE

We use Gibbs sampling for inference, so we derive posterior conditional distributions for all latent variables in our model.

The posterior distribution for $U_{m,k,l,l'}$ is log-concave and can be efficiently sampled using an adaptive rejection sam-

pling Gilks and Wild [1992] (ARS) routine:

$$\log p(U_{m,k,l,l'}|\text{rest}) = \left(\sum_{\substack{\{n:t_n=l \\ x_n^{(m)}=l'\}}} V_{n,k} \right) U_{m,k,l,l'} - \quad (8)$$

$$\sum_{\{n:t_n=l\}} \log \sum_{l'=1}^{L} \exp(\sum_{k'=0}^{K} U_{m,k',l,l'} V_{n,k'}) + \log(p(U_{m,k,l,l'}|s_k)).$$

Dealing with missing data requires simply indexing over non-missing entries.

4.1 SAMPLING THE ITEM FEATURES

Gibbs sampling over the binary feature matrix V involves two different steps. For $V_{n,k}$ such that $\sum_{n' \neq n} V_{n',k} > 0$, i.e., another item shares the same feature, we have

$$P(V_{n,k}|x, U, V_{-n,k}, t_n, s_k) \propto P(x|V_{-n,k}, U, V_{n,k}, t_n) * P(V_{n,k}|V_{-n,k}, s_k),$$

where from Equation (4),

$$P(V_{n,k} = 1|V_{-n,k}, s_k) = \frac{V_{\cdot,k}^T 1_N - V_{n,k} + \alpha^{s_k}/K^{s_k}}{N + \alpha^{s_k}/K^{s_k}},$$

where $K^{s_k} := \sum_{k'=1}^{K} \mathbb{I}(s_{k'} = s_k)$.

When $V_{-n,k} = 0$, factor k is replaced by a sample over the posterior distribution of latent features that no other item possesses. First, the number of new features K^{new} is sampled with probability

$$P(K^{\text{new}}|x_n, U, V_{n,-k}, t_n, v, s_k) \propto P(K^{\text{new}}|s_k) * P(x_n|U, [V_{n,-k}, 1_{K^{\text{new}}}], t_n, v, s_k), \quad (9)$$

where $[V_{n,-k}, 1_{K^{\text{new}}}]$ indicates the concatenation of the nth item's latent factors (except the kth one) with K^{new} extra latent factors. From the IBP, the prior $P(K^{\text{new}}|s_k)$ is given by $\text{Pois}(\alpha^{s_k}/N)$. In practice, the probability mass function of the posterior $P(K^{\text{new}}|\text{rest})$ is truncated at some $K_{\max}^{\text{new}}$.

The second term in Equation (9) is marginalized over the possible effects of the new latent factors represented by $U_{\cdot,K+1:K+K^{\text{new}}}$:

$$\begin{aligned} &P(x_n|U, [V_{n,-k}, 1_{K^{\text{new}}}], t_n, v, s_k) = \\ &E_{U_{\cdot,K+1:K+K^{\text{new}}}}[P(x_n|U, U_{\cdot,K+1:K+K^{\text{new}}}, [V_{n,-k}, 1_{K^{\text{new}}}], t_n, v, s_k)] \\ &= \prod_m E_{U_{m,K+1:K+K^{\text{new}}}} \Big[P(x_n^{(m)}|U, U_{m,K+1:K+K^{\text{new}},t_n}, \\ &\qquad [V_{n,-k}, 1_{K^{\text{new}}}], v, s_k) \Big] \\ &= \prod_m \mathbb{E}_{z_{k',l'} \sim \mathcal{N}_+(0,v)} \Bigg[\text{softmax}((\sum_{k=0}^{K} U_{m,k,t_n,l'} V_{n,k} + \\ &\qquad \sum_{k'=1}^{K^{\text{new}}} z_{k',l'} (\mathbb{I}(s_k = +)\mathbb{I}(l' = t_n) + \\ &\qquad \mathbb{I}(s_k = -)\mathbb{I}(l' \neq t_n)))_{l'=1}^{L}) \Bigg]^T x_n^{(m)}, \end{aligned}$$

where m indexes over all classifiers that classified item n.

Note that conditioning on s_k indicates that inference is being done using two separate IBP priors: one for $s_k = +$, the latent factors improving classification accuracy, and $s_k = -$, those detrimental to classification accuracy. Not only does this reflect a more generalized model in which we may expect a different number of positive and negative latent factors, but it also simplifies the next step in inference, which is to evaluate the second term in Equation (9).

Recall from Section 3 that a positive factor ($s_k = +$) results in a truncated normal $N_+(0, v)$ random variable being effectively added to each term on the diagonal of each classifier's confusion matrix factors, and a negative factors results in the same to the off-diagonal entries. For each possible K^{new}, this is done independently K^{new} times. Since the (inferred) label t_n is conditioned on, we only need to calculate the expectation of the softmax function w.r.t. these random variables on the t_nth row of the resulting confusion matrix.

When $s_k = +$ and only one element in the row has a random variable (or sum of RVs) to add, we can express any element of the softmax output as the result of applying the logistic sigmoid function, σ. To use arbitrary variables y and z, and adding z to the lth entry of vector y representing the confusion matrix factor row we have:

$$\frac{\exp(y_l + z)}{\sum_{l' \neq l} \exp(y_{l'}) + \exp(y_l + z)} = \sigma(y_l - \log(\sum_{l' \neq l} \exp(y_{l'})) + z) \quad (10)$$

$$\frac{\exp(y_{l'})}{\sum_{l' \neq l} \exp(y_{l'}) + \exp(y_l + z)} = \frac{\exp(y_{l'})}{\sum_{l' \neq l} \exp(y_{l'})} * \sigma(\log(\sum_{\tilde{l} \neq l} \exp(y_{\tilde{l}})) - y_l - z)). \quad (11)$$

For any real value x we can represent $\sigma(x)$ as a Taylor

expansion at some point μ:

$$\sigma(x) = \sum_{p=0}^{\infty} \frac{1}{p!} \sigma^{(p)}(\mu)(x-\mu)^p. \tag{12}$$

Setting $x := y + \sum_{k=1}^{K^{\text{new}}} z_k$, where $z_k \sim \mathcal{N}_+(0, v)$, then $x = \mu + \sum_k (z_k - m_1)$, where $\mu := y + K^{\text{new}} m_1$ and m_p is the pth moment of $\mathcal{N}_+(0, v)$. The expectation of $\sigma(x)$ is then

$$\mathbb{E}[\sigma(x)] = \sum_{p=0}^{\infty} \frac{1}{p!} \sigma^{(p)}(\mu) \mathbb{E}[(\sum_k (z_k - m_1))^p]. \tag{13}$$

From the multinomial theorem, we have

$$\mathbb{E}[(\sum_k (z_k - m_1))^p] = \sum_{\substack{h_1+h_2+\cdots+h_K=p \\ h_k \in \mathbb{Z}_+}} \binom{p}{h_1, h_2, ..., h_K} m_{h_1} m_{h_2} \cdots m_{h_K}. \tag{14}$$

The moments of a truncated normal distribution and the integer partitions needed can be efficiently calculated [Kelleher and O'Sullivan, 2014, Orjebin, 2014]. Computing derivatives of $\sigma(\mu)$ can be done recursively by noting that $(\sigma^{(p)})' = p(\sigma^{(p)} - \sigma^{(p+1)})$, so differentiation is matrix multiplication in the coefficient space of powers of σ.

For $z \sim \mathcal{N}_-(0, v)$, $E[z^p] = (-1)^p m_p$, so we can compute expectations of $\sigma(y - \sum_k z_k)$ in the same way.

We can thus compute the value of Equation (9) to arbitrary precision for positive latent features. While in general this allows avoiding costly Monte Carlo approximations to marginalizing over the item feature effects, it is particularly important for our method, as it enables our method to scale well with N.

When $s_k = -$, we approximate the expectation of adding independent sums of K^{new} truncated normals to all the off-diagonal terms with the expectation of the softmax when subtracting the sum of K^{new} truncated normals from the diagonal term, thus enabling approximating the expectation again by a Taylor series expansion.

After sampling K^{new}, the effects of the new item features on the classifiers $U_{\cdot,K:K+K^{\text{new}}}$ are sampled via Equation (8).

Finally, the feature variance is updated by

$$v|\text{rest} \sim IG\left(\alpha_v + \frac{ML}{2}(K^{\text{pos}} + (L-1)K^{\text{neg}}), \beta_v + \frac{1}{2}\sum_{m,k,l,l'} U^2_{m,k,l,l'}\right)$$

and the labels updated by

$$P(t_n = l|\text{rest}) \propto P(x_n|U, V_{n,\cdot}, t_n = l) P(t_n = l).$$

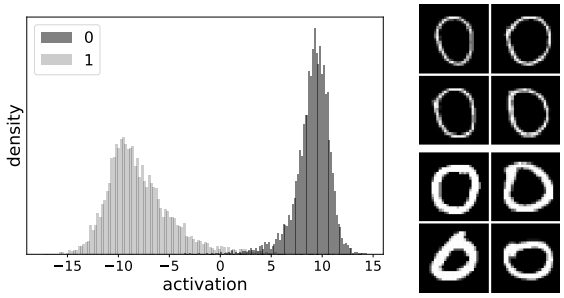

Figure 3: Toy example of item heterogeneity in a classification task. *Left:* A simple CNN binary classifier's activation values for classifying 0s vs 1s in MNIST. *Top right*: Items of label 0 with lowest activation values. The thinness of the digit means the convolutional filters learned do not activate well. *Bottom right*: Items of label 0 with typical activation $\approx$10.

5 EXPERIMENTS

In all experiments, we initialize $U_{\cdot,0}$ to be $2.5I$, $v = 5$, and sample for 1000 iterations, discarding the first 100 iterations and subsampling each 10th iteration to form a final estimate marginalized over subsamples. Our initialization and procedure in the IRT models is done the same way for fair comparison. We set $\alpha^{(\text{pos})} = \alpha^{(\text{neg})} = 10^{-2}$. This created a high threshold for adding new item features in order to prevent inferring spurious features. We truncate our Taylor series to 5 terms and took $K^{\text{new}}_{\text{max}} = 1$ for computational efficiency as we found no instances when using larger $K^{\text{new}}_{\text{max}}$ in which more than one new feature of the same sign was sampled for a particular item. We set $\alpha_v = \beta_v = 10^{-3}$ for a vague prior over the variance. We set $\kappa = 1_L$.

For the IRT models we set the priors (when applicable) to be $\lambda_i \sim \text{U}[0, 1]$, $\gamma_i \sim \text{Lognormal}(0, 1)$, $\alpha^{(l)}_m \sim \mathcal{N}(0, 1)$, $\beta_i \sim \mathcal{N}(0, 1)$. We used an adaptive Metropolis-Hastings method to simulate from the non-log-concave conditional posterior distributions of the 3-PL IRT model, simulating 100 steps each Gibbs step to ensure adequate mixing. Otherwise, the conditional posterior distributions are log-concave and can be sampled using ARS.

5.1 TOY EXAMPLE

To make more concrete our motivation that modeling item heterogeneity can improve classification performance, we illustrate this in a toy example of classifying 0s from 1s in MNIST. We choose a simple architecture for illustration purposes consisting of 20 5x5 convolutional filters followed by a max pooling over the entire feature map. We set the predictive probability of the digit being 0 or 1 proportional to the exponential of the sum of the first half of the max-pooled feature maps and the sum of the last half, respectively

Table 1: Cross-model classification accuracy in simulated data.

rows: inference / cols: generation	1-PL	2-PL	3-PL	idBCC
MV	0.925 ± 0.004	0.701 ± 0.002	0.717 ± 0.007	0.855 ± 0.002
1-PL	0.931 ± 0.004	0.73 ± 0.004	0.801 ± 0.005	0.861 ± 0.003
2-PL	0.926 ± 0.003	0.745 ± 0.004	0.780 ± 0.006	0.859 ± 0.005
3-PL	0.882 ± 0.015	0.740 ± 0.006	0.777 ± 0.007	0.855 ± 0.004
idBCC	$\mathbf{0.935 \pm 0.006}$	$\mathbf{0.780 \pm 0.003}$	$\mathbf{0.824 \pm 0.004}$	$\mathbf{0.865 \pm 0.006}$

Table 2: Model classification accuracy on black-box crowdsourcing benchmarks.

	face	**SP**	**CF**	**web**	**bird**	**MS**
MV	0.637 ± 0.004	0.886 ± 0.001	0.881 ± 0.009	0.729 ± 0.005	0.759 ± 0.000	0.704 ± 0.009
1-PL	0.643 ± 0.002	0.911 ± 0.000	0.887 ± 0.003	0.744 ± 0.004	0.889 ± 0.000	$\mathbf{0.801 \pm 0.001}$
2-PL	0.650 ± 0.002	0.910 ± 0.001	$\mathbf{0.890 \pm 0.000}$	0.769 ± 0.002	0.824 ± 0.000	0.794 ± 0.001
3-PL	0.647 ± 0.002	0.902 ± 0.002	0.885 ± 0.005	0.665 ± 0.000	0.815 ± 0.000	0.790 ± 0.002
IBCC	0.638 ± 0.002	0.916 ± 0.000	0.883 ± 0.000	0.753 ± 0.003	0.889 ± 0.000	0.785 ± 0.005
EBCC	0.638 ± 0.008	0.915 ± 0.000	0.883 ± 0.000	0.769 ± 0.000	0.863 ± 0.015	0.787 ± 0.000
cBCC	$\mathbf{0.651 \pm 0.000}$	0.915 ± 0.000	0.877 ± 0.000	0.782 ± 0.000	0.889 ± 0.000	0.777 ± 0.000
idBCC	$\mathbf{0.651 \pm 0.004}$	$\mathbf{0.918 \pm 0.001}$	0.888 ± 0.005	$\mathbf{0.859 \pm 0.003}$	$\mathbf{0.926 \pm 0.000}$	0.787 ± 0.013

(i.e., it is the softmax of the activation sum over each half of the filters).

From Figure 3, we see that after training, convolutional filters are learned such that the resulting activation values (the sum of the max-pooled 0 feature maps minus the sum of the max-pooled 1 feature maps) are clustered according to the digit, but the classes are not fully separated. Upon inspecting the convolutional filters, we found that the filters that learned to match for 0 (the first half of the set of filters) were better at classification than those learned to match for 1 (possibly because an edge detector that activates for 1 would still activate relatively strongly for many 0s), with a similar clustering of digits as in the figure. However, certain items did not adequately activate the learned convolutional filters due to the thinness of the drawn digit, thus giving rise to label-conditional item heterogeneity.

We wanted to see if 1. our proposed model could infer the heterogeneity in our data purely from the individual outputs of the filters 2. modeling this heterogeneity would lead to better performance. We constructed a black-box dataset consisting of the activations of our filters binarized by thresholding them at their mean activations for 300 data points: 100 points consisting of the 0s that activated the 0 filters the least (the difficult items), 100 points consisting of the 0s that activated the 0 filters the most (the easy items), and 100 points that activated the 1 filters the most. We found our model to approximately correctly identify the item heterogeneity, inferring a negative item feature that 81% on average of the difficult items had as a feature, compared to 12% of the easy 0s and 0% of the easy 1s. Under majority vote, predictive accuracy was 0.947, compared to 0.970 under our proposed model.

5.2 SIMULATED DATA

We compared idBCC and the Bayesian IRT models on simulated data, performing all pairwise comparisons between models, shown in Table 1, where we generated data from the prior of each of our Bayesian models, corresponding to each column in the table, and ran inference under each model, recording the predictive accuracy of each model in the rows. Note that in general performance is highest among the IRT models for data generated by the 3-PL models because, due to a non-zero guessing probabilities λ_n, the base classifiers are strictly more accurate than under the 2- and 1-PL models.

Our results show that even when the generative and inferential models are the same, a simpler (and more flexible, in the case of idBCC) model can often perform better. Even if the structure of the model matches that of the ground truth, if the model has many parameters, it might be difficult to end up in a region of parameter space in which all the parameters are useful to the model. Otherwise, the model effectively marginalizes over a set of nuisance parameters that it must infer, in contrast to idBCC which can remove parameters if they contribute little or negatively to the evidence (or

Table 3: CIFAR10 classification accuracy (Average classifier accuracy of base models: .952)

Training/Testing points	IDNT	TAIDTM	CrowdLayer	idBCC
5000	0.933 ± 0.005	0.787 ± 0.023	$\mathbf{0.963 \pm 0.007}$	0.960 ± 0.003
1000	0.933 ± 0.003	0.481 ± 0.015	0.946 ± 0.001	$\mathbf{0.957 \pm 0.004}$
500	0.930 ± 0.007	0.378 ± 0.022	0.950 ± 0.006	$\mathbf{0.955 \pm 0.004}$
200	0.929 ± 0.014	0.298 ± 0.007	0.952 ± 0.013	$\mathbf{0.956 \pm 0.007}$
100	0.924 ± 0.011	0.200 ± 0.018	0.935 ± 0.007	$\mathbf{0.950 \pm 0.005}$

Table 4: FashionMNIST classification accuracy (Average classifier accuracy of base models: .916)

Training/Testing points	IDNT	TAIDTM	CrowdLayer	idBCC
5000	0.915 ± 0.003	0.917 ± 0.005	0.925 ± 0.003	$\mathbf{0.937 \pm 0.006}$
1000	0.905 ± 0.004	0.915 ± 0.006	0.918 ± 0.004	$\mathbf{0.935 \pm 0.006}$
500	0.898 ± 0.006	0.913 ± 0.007	0.915 ± 0.007	$\mathbf{0.932 \pm 0.005}$
200	$0.895 \pm .0013$	0.873 ± 0.023	0.912 ± 0.006	$\mathbf{0.930 \pm 0.007}$
100	0.882 ± 0.015	0.593 ± 0.039	0.898 ± 0.009	$\mathbf{0.920 \pm 0.005}$

add more if they help). Such nuissance parameters are not limited to the 2-PL and 3-PL models; if, for example, an item's difficulty is close to 0 in the 1-PL model (or if in the 2-PL model its discriminability is very low), not modeling its difficulty (effectively setting its parameterization to 0) could be more beneficial than marginalizing over stochastic inferences of it, which may contain little information.

5.3 BLACK-BOX CROWDSOURCING BENCHMARKS

We next tested our model's performance on several crowdsourcing black-box benchmark datasets [Welinder et al., 2010, Zhao et al., 2012, Rodrigues et al., 2013, Mozafari et al., 2014, Venanzi et al., 2015], which we show in Table 2. Overall, we found that idBCC performs the most robustly of all the methods, performing the best on the majority of the datasets, and still performing close to the top when it did not perform best.

We also found the IRT models to often perform surprisingly well in comparison to state-of-the-art models, which are generally more sophisticated. In particular, 1-PL and 2-PL performed fairly robustly, although they were substantially worse than idBCC on the `web` and `bird` datasets.

5.4 WHITE-BOX BENCHMARKS

We finally compared our method against neural network based white-box methods CrowdLayer [Rodrigues and Pereira, 2018], TAIDTM Guo et al. [2023], and IDNT [Li et al., 2024] in two tasks combining classifications from neural network classifiers. For our base classifiers, we used max-one-hotted predictions from Densenet-bc-L190-k40, PreResnet-110, and Resnet-110 on the test set of CIFAR10[1] and from LeNet-5, AlexNet-Light, VGGNet-16, and InceptionNet-10 on the test set of the FashionMNIST dataset[2]. We used the official implementations for TAIDTM[3] and IDNT[4] and the crowd-kit[5] Python implementation for CrowdLayer.

We did not want to test model performance on data that had been used for training/validation of the base classifiers, so we restricted ourselves to the test set, taking random subsamples of size 100, 200, 500, 1000, and 5000 of base model classifications of the test set. Since the models have access only to the images and the noisy annotations and we are interested in these predictions, the training and test sets are the same. We used the rest of the dataset (the original test test containing 10000 examples) as the validation set for the neural network based models to ensure good model validation. For idBCC, we did not use this validation set, which meant that we used at most half the amount of data as the other methods in every comparison. We show performance for CIFAR10 in Table 3. In general, we found idBCC gave the best performance and retained high performance as the number of datapoints decreased down to 100. We found

[1] downloaded from github.com/GavinKerrigan/conf_matrix_and_calibration

[2] pretrained weights downloaded from github.com/wzyjsha-00/CNN-for-Fashion-MNIST

[3] github.com/tmllab/TAIDTM

[4] github.com/hguo1728/BayesianIDNT

[5] crowd-kit.readthedocs.io

similar performance under FashionMNIST, which is shown in Table 4.

6 DISCUSSION

A limitation of our model is that the features that can be inferred are additive; it may be possible to generalize our model further by allowing for interactions between two features or among features up to some fixed order, or to a generic order using a hierarchical version of the IBP (e.g., James et al. [2024]).

To the best of our knowledge, our model is the first black-box Bayesian instance-dependent model of classifier combination, which we have shown generally performs better compared to competitors both on black-box crowdsourcing tasks as well as white-box classifier combination tasks when there is limited data.

References

F. B. Baker and S. H. Kim. *Item response theory: Parameter estimation techniques*. Marcel Dekker, New York, NY, 2004.

D. Blackwell and J. Macqueen. Ferguson distributions via Polya urn schemes. *The Annals of Statistics*, 1:353–355, 1973.

S. Budd, E. C. Robinson, and B. Kainz. A survey on active learning and human-in-the-loop deep learning for medical image analysis. *Medical Image Analysis*, 71, 2021.

C. Callison-Burch and M. Dredze. Creating speech and language data with Amazon's Mechanical Turk. *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, 2010.

T. G. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems. MCS 2000. Lecture Notes in Computer Science*, pages 1–15. Springer, Berlin, Heidelberg, 2000.

T. Ferguson. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209–230, 1973.

W. R. Gilks and P. Wild. Adaptive rejection sampling for Gibbs sampling. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 41(2):337–348, 1992.

T. L. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. *Advances in Neural Information Processing Systems*, 18, 2005.

T. L. Griffiths and Z. Ghahramani. The Indian Buffet Process: An introduction and review. *Journal of Machine Learning Research*, 12:1185–1224, 2011.

H. Guo, B. Wang, and G. Yi. Label correction of crowdsourced noisy annotations with an instance-dependent noise transition model. *Advances in Neural Information Processing Systems*, 2023.

S. W. Han, O. Adiguzel, and B. Carpenter. Crowdsourcing with difficulty: A bayesian rating model for heterogeneous items. 2024. URL `https://arxiv.org/abs/2405.19521`.

J. Howe. The rise of crowdsourcing. *Wired*, 2006.

L. F. James, J. Lee, and A. Pandey. Bayesian analysis of generalized hierarchical Indian Buffet Processes for within and across group sharing of latent features. 2024. URL `https://arxiv.org/abs/2304.05244`.

J. Kelleher and B. O'Sullivan. Generating all partitions: A comparison of two encodings. 2014. URL `https://arxiv.org/pdf/0909.2331`.

H.-C. Kim and Z. Ghahramani. Bayesian classifier combination. *International Conference on Artificial Intelligence and Statistics*, pages 619–627, 2012.

T. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*, 2017.

P. F. Lazarsfeld. The logical and mathematical foundation of latent structure analysis. In S. A. Stouffer, L. Guttman, A. Suchman, P. Lazarsfeld, A. Star, and J. A. Clausen, editors, *Measurement and Prediction, volume IV of Studies in Social Psychology in World War II*, pages 362–412. Princeton University Press, Princeton, NJ, 1950.

S. Li, X. Xia, J. Deng, S. Ge, and Tongliang Liu. Transferring annotator- and instance-dependent transition matrix for learning from crowds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Y. Li, B. I. P. Rubinstein, and T. Cohn. Exploiting worker correlation for label aggregation in crowdsourcing. *International Conference on Machine Learning*, 2019.

F. M. Lord, M. R. Novick, and A. Birnbuam. *Statistical theories of mental test scores*. Addison-Wesley, 1968.

P. G. Moreno, A. Artes-Rodriguez, Y.-W. Teh, and F. Perez-Cruz. Bayesian nonparametric crowdsourcing. *Journal of Machine Learning Research*, 16:1607–1627, 2015.

B. Mozafari, P. Sarkar, M. Franklin, M. Jordan, and S. Madden. Scaling up crowd-sourcing to very large datasets: A case for active learning. *Proceedings of the VLDB Endowment*, 8(2):125–136, 2014.

E. Orjebin. A recursive formula for the moments of a truncated univariate normal distribution. 2014. URL `https://people.smp.`

uq.edu.au/YoniNazarathy/teaching_projects/studentWork/EricOrjebin_TruncatedNormalMoments.pdf.

G. Rasch. *Studies in Mathematical Psychology: I. Probabilistic Models for Some Intelligence and Attainment Tests*. Nielsen & Lydiche, 1960.

F. Rodrigues, F. Pereira, and B. Ribeiro. Learning from multiple annotators: distinguishing good from random labelers. *Pattern Recognition Letters*, 34(12):1428–1436, 2013.

R. Rodrigues and F. Pereira. Deep learning from crowds. *AAAI*, 2018.

Y.-W. Teh. Dirichlet process. *Encyclopedia of machine learning*, pages 280–287, 2010.

S. Trick and C. Rothkopf. Bayesian classifier fusion with an explicit model of correlation. *AISTATS*, 2022.

S. Trick, C. Rothkopf, and F. Jakel. A normative model for Bayesian combination of subjective probability estimates. *Judgment and Decision Making*, 18:1–20, 2023.

M. Venanzi, O. Parson, A. Rogers, and N. Jennings. The activecrowdtoolkit: An open-source tool for benchmarking active learning algorithms for crowdsourcing research. *In Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.

P. Welinder, S. Branson, P. Perona, and S. Belongie. The multidimensional wisdom of crowds. *Advances in Neural Information Processing Systems*, pages 2424–2432, 2010.

J. Whitehill, T.-F. Wu, J. Bergsma, J. Movellan, and P. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in Neural Information Processing Systems*, 2009.

S. E. Yuksel, J. N. Wilson, and P. D. Gader. Twenty years of Mixture of Experts. *IEEE Transcations on Neural Networks and Learning Systems*, 2012.

B. Zhao, B. I. P. Rubinstein, J. Gemmell, and J. Han. A Bayesian approach to discovering truth from conflicting sources for data integration. *Proceedings of the VLDB Endowment*, 5(6):550–561, 2012.