Topic Model Based Multi-Label Classification

Divya Padmanabhan, Satyanath Bhat, Shirish Shevade, and Y. Narahari

Indian Institute of Science, Bangalore

Abstract. Multi-label classification is a common supervised machine learning problem where each instance is associated with multiple classes. The key challenge in this problem is learning the correlations between the classes. An additional challenge arises when the labels of the training instances are provided by noisy, heterogeneous crowd-workers with unknown qualities. We first assume labels from a perfect source and propose a novel topic model (ML-PA-LDA) where the classes that are present as well as the classes absent generate the latent topics and hence the words. Extensive experimentation on real world datasets reveals the superior performance of the proposed model. We then non-trivially extend our topic model to the scenario where the labels are provided by noisy crowd-workers and refer to this model as ML-PA-LDA-C. With experiments on simulated crowd, the proposed model learns the qualities of the annotators well, even with minimal training data.

1 Introduction

Multi-label classification is a variant of a classification problem wherein an instance $\mathbf{d}$ is associated with multiple classes or labels. There are several areas where multi-label classification finds applications, for example, text classification, image retrieval, etc. Consider the task of classification of documents into several classes such as crime, politics, arts, sports etc. The classes are not mutually exclusive since a document belonging to the 'politics' category may also belong to 'crime'. In the classification of images, an image belonging to 'forest' category may also belong to 'scenery' category, and so on.

One of the solution approaches for multi-label classification is to generate a new label set that is a power set of the original label set, and then use traditional single label classification techniques. The immediate limitation here is an exponential blow-up of the label set and availability of only a small sized training dataset for each of the generated labels. Another approach is to build one-vs-all binary classifiers, where, for each label, a binary classifier is built. This method, however, does not take into account the correlation between the labels.

In the past, topic models [2, 11, 13] have proved to be successful in modeling the process behind generating text documents. The idea is to model latent topics responsible for generating words. Originally, topic models were used in an unsupervised manner and were gradually adapted to the supervised learning setting [15]. The models for single label multi-class classification were then adapted to the multi-label setting [26, 22]. However the topic models developed for the multi-label setting either involve too many parameters [22] or learn the parameters by

heavily depending on iterative optimization techniques [26], thereby making it hard to adapt to the scenario where labels are provided by noisy crowd-workers. Moreover in all these models, the topics and hence words are assumed to be generated depending only on the classes that are present. They do not make use of the information provided by the *absence* of classes. The absence of a class often provides critical information about the words present. For example, a document labeled 'sports' is less likely to have words related to 'astronomy'. Similarly in the images domain, an image categorised as 'portrait' is less likely to have the characteristics of 'scenery'. Needless to say, such correlations are dataset dependent. However a principled analysis must account for such correlations. Motivated by this subtle observation, we introduce a novel topic model for multi-label classification.

Further the problem renders itself more interesting when the labels are procured from multiple heterogenous noisy crowd-workers with unknown qualities. We also refer to crowd-workers as annotators in the paper. In the current era of big data where large amounts of unlabeled data are readily available, obtaining a noiseless source for labels is almost impossible. However it is possible to get instances labeled by several human annotators. The problem becomes harder as now the true labels are unknown and the qualities of the annotators must be learnt to train a model. We non-trivially extend our topic model to this scenario.

Contributions

1. We introduce a novel topic model for multi-label classification; our model has the distinctive feature of exploiting any additional information provided by the absence of classes. We refer to our topic model as ML-PA-LDA (Multi-Label Presence-Absence LDA).
2. If the labels are provided by multiple noisy annotators (from a crowd), we enhance our model to account for heterogenous annotators with unknown qualities. We refer to this enhanced model as ML-PA-LDA-C (ML-PA-LDA with Crowd). A feature of ML-PA-LDA-C is that it does not require an annotator to label all classes for a document. Even partial labeling by the annotators upto the granularity of labels within a document is adequate.
3. We test the performance of ML-PA-LDA on several real world datasets and establish its superior performance over state of the art.
4. Further, we study the performance of ML-PA-LDA-C, with simulated annotators providing the labels for these datasets. In spite of the noisy labels, ML-PA-LDA-C demonstrates excellent performance and the qualities of the annotators learnt approximate closely the true qualities of the annotators.

2 Related Work

Several approaches have been devised for multi-label classification with labels provided by a single source. The most natural approach is the Label Powerset (LP) method [6] which generates a new class for every combination of labels and

then solves the problem using multiclass classification approaches. The main drawback of this approach is the exponential growth in the number of classes, leading to several generated classes having very few labeled instances leading to overfitting. To overcome this drawback, RAndom k-labELsets method (RAkEL) [23] was introduced, which constructs an ensemble of LP classifiers where each classifier is trained with a random subset of k labels. However, the large number of labels still poses challenges. The approach of pairwise comparisons (PW) improves upon the above methods, by constructing C(C-1)/2 classifiers for every pair of classes, where C is the number of classes. Finally a ranking of the predictions from each classifier yields the labels for a test instance. Rank-SVM [9] uses PW approach to construct SVM classifiers for every pair of classes and then performs a ranking.

The previously described approaches are discriminative approaches. Generative models for multi-label classification model the correlation between the classes by mixing weights for the classes [16]. Other probabilistic mixture models include Parametric Mixture Models PMM1 and PMM2 [25]. After the advent of the topic models like Latent Dirichlet Allocation (LDA) [2], extensions have been proposed for multi-label classification such as Wang et al [26]. However in [26], due to the non-conjugacy of the distributions involved, closed form updates cannot be obtained for several parameters and iterative optimization algorithms such as conjugate gradient and Newton Raphson are required to be used in the variational E step as well as M step, introducing additional implementation issues. Adapting this model to the case of crowds would result in enormous complexity. The topic models proposed for multi-label classification in [22] involve far too many parameters which can be learnt effectively only in the presence of large amounts of labeled data. For small and medium sized datasets, the approach suffers from overfitting. Moreover it is not clear how this model can be adapted when labels are procured from crowd-workers with unknown qualities. SLDA [15] is a single label classification technique which works well on multi-label classification when used with the one-vs-all approach. SLDA inherently captures the correlation between classes through the latent topics.

With crowdsourcing gaining popularity due to the availability of large amounts of unlabeled data and difficulty in procuring noiseless labels for these datasets, aggregating labels from the crowd has become an important problem. Raykar et al [18] look at training binary classification models with labels from a crowd with unknown annotator qualities. Being a model for multiclass classification, this model does not capture the correlation between classes and thereby cannot be used for multi-label classification from the crowd. Mausam et al [5] look at multi-label classification for taxonomy creation from the crowd. They construct C classifiers by modeling the dependence between the classes explicitly. The graphical model representation involves too many edges especially when the number of classes is large and hence the model suffers from overfitting. Deng et al [7] look at selecting the instance to be given to a set of crowd-workers. However they do not look at aggregating these labels and developing a model for classification given these labels. In the report [8], Duan et al. look at methods

to aggregate a multi-label set provided by crowd-workers. However, they do not look at building a model for classification for new test instances for which the labels are not provided by the crowd. Recently the topic model, SLDA, has been adapted to learning from the labels provided by crowd annotators [21]. However, like its predecessor SLDA, it is only applicable to the single label setting and not to multi-label classification.

The existing topic models in the literature assume that the presence of a class generates words pertaining to those classes and do not take into account the fact that the absence of a class may also play a role in generating words. In practice, the absence of a class may yield information about occurrence of words. We propose a model for multi-label classification based on latent topics where the presence as well as absence of a class could generate topics. The labels could be procured from multiple sources (e.g. crowd workers) whose qualities are unknown.

3 Proposed Approach for Multi-label Classification: ML-PA-LDA

We now explain our model for multi-label classification. For ease of exposition, we use notations from the text domain. However the model itself is general and can be applied to several domains by suitable transformation of features into words. In our experiments we have applied the model to domains other than text. We will explain the transformation of features to words when we describe our experiments.

Let D be the number of documents in the training set, also known as a corpus. Each document is a set of several words. Let C be the total number of classes in the universe. In multi-label classification, a document may belong to any 'subset' of the C classes as opposed to the standard classification setting where a document belongs to exactly one class. Let T be the number of latent topics responsible for generating words. The set of all possible words is referred to as a vocabulary. We denote by V the size of the vocabulary $\nu = \{\nu_1, \ldots, \nu_V\}$, where ν_j refers to the j^{th} word in ν. Consider a document $\mathbf{d}$ comprising N words $\mathbf{w} = \{w_1, w_2, \ldots, w_N\}$ from the vocabulary ν. Let $\lambda = [\lambda_1, \ldots, \lambda_C] \in \{0, 1\}^C$ denote the true class membership of the document. In our notations, we denote by w_{nj} the value $\mathbb{1}[w_n = \nu_j]$, that is the indicator that the word w_n is the j^{th} word of the vocabulary. Similarly, we denote by λ_{ij}, the indicator that $\lambda_i = j$, where $j = 0$ or 1. The objective is to predict the vector λ for every test document.

Topic Model for the Documents

We introduce a model to capture the correlation between the various classes generating a given document. The presence as well as absence of a class provides additional information about the topics present in a document. We now describe the generative process for each document assuming labels are provided by a perfect source.

1. Draw $\lambda_i \sim \text{Bern}(\xi_i)$ for every class $i = 1, \ldots, C$.
2. Draw $\theta_{i,j,\cdot} \sim \text{Dir}(\alpha_{i,j,\cdot})$ for $i = 1, \ldots, C$, for $j \in \{0, 1\}$, where $\alpha_{i,j,\cdot}$ are the parameters of a Dirichlet distribution with T parameters.
3. For every word w in the document
   (a) Sample $u \sim \text{Unif}\{1, \ldots, C\}$ from one of the C classes.
   (b) Generate a topic $z \sim \text{Mult}(\theta_{u,\lambda_u,\cdot})$, where $\theta_{i,j,\cdot}$ are the parameters of a multinomial distribution in T dimensions.
   (c) Generate the word $w \sim \text{Mult}(\beta_{z\cdot})$ where $\beta_{z\cdot}$ are the parameters of a multinomial distribution in V dimensions.

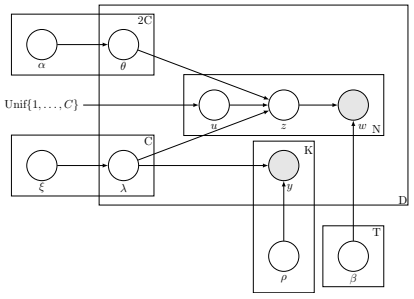
We refer to this model where the true class vector λ is observed for the training documents as ML-PA-LDA (Multi-Label Presence-Absence LDA).

Intuitively, for every class i, its presence or absence (λ_i) is first sampled from a Bernoulli distribution parameterized by ξ_i. The parameter ξ_i is the prior for class i. We capture the correlations across classes through latent topics. The corpus wide distribution $\text{Dir}(\alpha_{i,j,\cdot})$ is the prior for the distribution ($\text{Mult}(\theta_{i,j,\cdot})$) of topics for class i taking the value j. Then the latent class u is sampled, which in turn along with λ_u generates latent topic z. The topic z is then responsible for a word. The same process repeats for the generation of every word in the document.

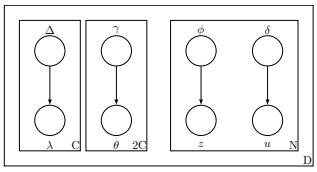
Single Coin Model for the Annotators

When the true labels of the documents are not observed, λ is unknown. Instead noisy versions $y_1, \ldots, y_K$ of λ provided by a set of K independent annotators with heterogenous unknown qualities $\{\rho_1, \ldots, \rho_K\}$ are observed. y_{ji} can be either $0, 1$ or -1. $y_{ji} = 1$ indicates that, according to annotator j, the class i is present while $y_{ji} = 0$ indicates that the class i is absent as per annotator j. $y_{ji} = -1$ indicates that the annotator j has not made a judgement on the presence of class i in the document. This allows for partial labeling upto the granularity of labels even within a document. This flexibility in the modeling is essential, especially when the number of classes is large. ρ_j is the probability with which an annotator reports the ground truth corresponding to each of the classes. ρ_j is not known to the learning algorithm. For simplicity we have assumed the single coin model for annotators and also that the qualities of the annotators are independent of the class under consideration. That is, $P(y_{j1} = 1|\lambda_1 = 1) = P(y_{j1} = 0|\lambda_1 = 0) = \ldots = P(y_{jC} = 1|\lambda_C = 1) = P(y_{jC} = 0|\lambda_C = 0) = \rho_j$. This is a common assumption in literature [18].

The generative process for the documents is depicted pictorially in Figure 1a. The parameters of our model consist of $\pi = \{\alpha\ ,\ \xi\ ,\ \rho, \beta\}$. The observed variables for each document are $\mathbf{d} = \{\mathbf{w}, y_{ji}\}$ for $i = 1, \ldots, C$, $j = 1, \ldots, K$. The hidden random variables are $\Theta = \{\theta, \lambda, u, z\}$. We refer to our topic model trained with labels from annotators as ML-PA-LDA-C (Multi-Label Presence-Absence LDA with Crowd).



(a) Graphical model for ML-PA-LDA-C where labels are provided by the crowd-workers.



(b) Graphical model representation of the variational distribution used to approximate the posterior

Fig. 1: Graphical model representations of our model. For ML-PA-LDA (the non-crowd version), Figure 1a is modified so that the observed variables are w and λ while the random variable y and therefore the parameter ρ is absent.

4 Variational EM for ML-PA-LDA-C

We now detail the steps for estimating the parameters of our proposed model ML-PA-LDA-C. Since ML-PA-LDA-C is a generalization of ML-PA-LDA to adapt to the crowd, we provide the details of the steps for ML-PA-LDA-C and give pointers to highlight the differences with ML-PA-LDA whenever appropriate.

Given the observed words $\mathbf{w}$ and the labels $y_1, \ldots, y_K$ for a document $\mathbf{d}$, where y_j is the label vector provided by annotator j. The objective of the model described above is to obtain $p(\Theta|\mathbf{d})$. Here, the challenge lies in the intractable computation of $p(\Theta|\mathbf{d})$ which arises due to the intractability in the computation of $p(\mathbf{d}|\pi)$. We use variational inference with mean field assumptions [1] to overcome this challenge.

The underlying idea in variational inference is the following. Suppose $q(\Theta)$ is any distribution over Θ for any arbitrary $\Theta = \{\theta, \lambda, u, z\}$ which approximates $p(\Theta|\mathbf{d})$. We refer to $q(\Theta)$ as variational distribution. The underlying variational model is provided in Figure 1b. Observe that,

$$\begin{aligned}
\ln p(\mathbf{d}|\pi) &= \ln \frac{p(\mathbf{d}, \Theta|\pi)}{p(\Theta|\mathbf{d}, \pi)} = \ln \frac{p(\mathbf{d}, \Theta|\pi) q(\Theta)}{q(\Theta) p(\Theta|\mathbf{d}, \pi)} \qquad (1)\\
&= \mathbb{E}_{q(\Theta)} \left[\ln \frac{p(\mathbf{d}, \Theta|\pi) q(\Theta)}{q(\Theta) p(\Theta|\mathbf{d}, \pi)} \right] \\
&= \mathbb{E}_{q(\Theta)} \left[\ln p(\mathbf{d}, \Theta|\pi) - \ln q(\Theta) \right] + \mathbb{E}_{q(\Theta)} \left[\ln q(\Theta) - \ln p(\Theta|\mathbf{d}, \pi) \right] \\
&= \mathcal{L}(\Theta) + \mathbf{KL}(q(\Theta)||p(\Theta|\mathbf{d}, \pi)) \qquad (2)
\end{aligned}$$

Variational inference involves maximizing $\mathcal{L}(\Theta)$ over the variational parameters $\{\Delta, \gamma, \phi, \delta\}$ so that $\mathbf{KL}(q(\Theta)||p(\Theta|\mathbf{d}, \pi))$ also gets minimized. In our notations,

$u_{ni} = \mathbb{1}[u_n = i]$ for $i \in \{1, \ldots, C\}$, $z_{nt} = \mathbb{1}[z_n = t]$ for $t \in \{1, \ldots, T\}$ and Γ denotes the gamma function. From our model (Figure 1a),

$$\begin{aligned}\ln p(\mathbf{d}, \Theta|\pi) &= \ln p(\mathbf{w}, y, \theta, \lambda, u, z|\pi) \\ &= \ln p(\lambda|\xi) + \ln p(\theta|\alpha) + \ln p(u) + \ln p(z|\lambda, u, \theta) + \\ &\quad \ln p(\mathbf{w}|z, \beta) + \ln p(y|\lambda, \rho)\end{aligned}$$

and in turn,

$$\ln p(\lambda|\xi) = \sum_{i=1}^{C} \lambda_i \ln \xi_i + (1 - \lambda_i) \ln(1 - \xi_i) \tag{3}$$

$$\ln p(\theta_{ij}|\alpha) = \ln \Gamma \left(\sum_{t=1}^{T} \alpha_{ijt} \right) - \sum_{t=1}^{T} \ln \Gamma \alpha_{ijt} + \sum_{t=1}^{T} (\alpha_{ijt} - 1) \log \theta_{ijt} \tag{4}$$

$$\log p(u) = \sum_{n=1}^{N} \sum_{i=1}^{C} u_{ni} \log 1/C \tag{5}$$

$$\log p(z|u, \lambda, \theta) = \sum_{n=1}^{N} \sum_{t=1}^{T} \sum_{i=1}^{C} \sum_{j=0}^{1} u_{ni} \lambda_{ij} z_{nt} \log \theta_{ijt} \tag{6}$$

$$\log p(w|z, \beta) = \sum_{n=1}^{N} \sum_{t=1}^{T} \sum_{j=1}^{V} w_{nj} z_{nt} \log \beta_{tj} \tag{7}$$

$$\begin{aligned}\log p(y|\lambda, \rho) = &\sum_{j=1}^{K} \sum_{i=1}^{C} [\lambda_i y_{ji} + (1 - \lambda_i)(1 - y_{ji})] \log \rho_j \\ &+ [(1 - \lambda_i) y_{ji} + \lambda_i (1 - y_{ji})] \log 1 - \rho_j\end{aligned} \tag{8}$$

Assume the following variational distributions (as per Figure 1b) over Θ for a document d. These assumptions on the independence between the latent variables are known as mean field assumptions [1].

$$\begin{aligned}&u^d \sim \text{Mult}(\delta^d), \;\; \lambda_i^d \sim \text{Bern}(\Delta_i^d) \text{ for } i = 1, \ldots, C, \\ &z^d \sim \;\text{Mult}(\phi^d), \;\; \theta_{ij}^d \sim \text{Dir}(\gamma_{ij}^d) \text{ for } i = 1, \ldots, C \text{ and } j = 0, 1\end{aligned}$$

Therefore, for a document d,

$$q(\Theta^d) = \prod_{i=1}^{C} q(\lambda^d) \prod_{i=1}^{C} \prod_{j=0}^{1} q(\theta_{ij}^d) \prod_{n=1}^{N} \prod_{i=1}^{C} q(u_{ni}^d) q(z_{ni}^d)$$

4.1 E-step Updates for ML-PA-LDA-C

The E-step involves computing the document-specific variational parameters $\Theta^d = \{\delta^d, \Delta^d, \gamma^d, \phi^d\}$, for every document d, assuming a fixed value for the parameters $\pi = \{\alpha, \xi, \rho, \beta\}$. As a consequence of the mean field assumptions on

the variational distributions, we get the following update rules for the distributions by maximising $\mathcal{L}(\Theta)$. From now on, when clear from context we omit the superscript d.

$$\begin{aligned}\log q(z) &= \mathbb{E}_{\Theta\setminus z}\left[p(\mathbf{d},\Theta)\right] \propto \mathbb{E}_{u,\lambda,\theta}\left[\log p(z|u,\lambda,\theta)\right] + \log p(w|z,\beta) \\ &\propto \sum_{n=1}^{N}\sum_{t=1}^{T} z_{nt}\left[\sum_{i=1}^{C}\sum_{j=0}^{1}\mathbb{E}[u_{ni}]\mathbb{E}[\lambda_{ij}]\mathbb{E}[\log\theta_{ijt}]\right] \quad (9) \\ &\quad + \sum_{n=1}^{N}\sum_{t=1}^{T} z_{nt}\left[\sum_{j=1}^{V} w_{nj}\log\beta_{tj}\right] \quad (10)\end{aligned}$$

In the computation of the expectation of $\mathbb{E}_{\Theta\setminus z}\left[p(\mathbf{d},\Theta)\right]$ in Eqn 10, the terms in $p(\mathbf{d},\Theta)$ that are a function of z need to be considered as the rest of the terms contribute to the normalizing constant for the density function $q(z)$. Hence expectations of $\log p(z|u,\lambda,\theta)$ (Eqn 6) and $\log p(w|z,\beta)$ (Eqn 7) must be taken with respect to u,λ,θ. Therefore ,

$$\begin{aligned}\log\phi_{nt} &\propto \sum_{i=1}^{C}\sum_{j=0}^{1}\mathbb{E}[u_{ni}]\mathbb{E}[\lambda_{ij}]\mathbb{E}[\log\theta_{ijt}] + \sum_{j=1}^{V} w_{nj}\log\beta_{tj} \quad (11)\\ &= \sum_{i=1}^{C}\sum_{j=0}^{1}\delta_{ni}\Delta_i^j(1-\Delta_i)^{1-j}\mathbb{E}[\log\theta_{ijt}] + \sum_{j=1}^{V} w_{nj}\log\beta_{tj}\end{aligned}$$

Similarly, the updates for the other variational parameters are as follows.

$$\begin{aligned}\log q(u) &= \mathbb{E}_{\Theta\setminus u}\left[\log p(u) + p(z|u,\lambda,\theta)\right] \propto \sum_{n=1}^{N}\sum_{i=1}^{C} u_{ni}\log 1/C \\ &\quad + \sum_{n=1}^{N}\sum_{t=1}^{T}\sum_{i=1}^{C}\sum_{j=0}^{1} u_{ni}\mathbb{E}[\lambda_{ij}]\mathbb{E}[z_{nt}]\mathbb{E}[\log\theta_{ijt}] \quad (12)\end{aligned}$$

Therefore,

$$\log\delta_{ni} \propto \log\frac{1}{C} + \sum_{t=1}^{T}\phi_{nt}\Delta_i\mathbb{E}\left[\log\theta_{i1t}\right] + \phi_{nt}(1-\Delta_i)\mathbb{E}\left[\log\theta_{i0t}\right] \quad (13)$$

$$\begin{aligned}\log\, q(\theta) &= \mathbb{E}_{\Theta\setminus\theta}\left[p(\mathbf{d},\Theta)\right] \propto \mathbb{E}\left[p(\theta|\alpha) + p(z|u,\lambda,\theta)\right] \\ &= \sum_{i=1}^{C}\sum_{j=0}^{1}\sum_{t=1}^{T}(\alpha_{ijt}-1)\log\theta_{ijt} + \sum_{n=1}^{N}\sum_{i=1}^{C}\sum_{j=0}^{1}\sum_{t=1}^{T}\mathbb{E}[u_{ni}]E[z_{nt}]E[\lambda_{ij}]\log\theta_{ijt} \\ &= \sum_{i=1}^{C}\sum_{j=0}^{1}\sum_{t=1}^{T}(\alpha_{ijt}-1)\log\theta_{ijt} + \sum_{n=1}^{N}\sum_{i=1}^{C}\sum_{j=0}^{1}\sum_{t=1}^{T}\delta_{ni}\phi_{nt}\Delta_i^j(1-\Delta_i)^{1-j}\log\theta_{ijt} \\ &= \sum_{i=1}^{C}\sum_{j=0}^{1}\sum_{t=1}^{T}(\gamma_{ijt}-1)\log\theta_{ijt}\end{aligned}$$

where,

$$\gamma_{ijt} = \alpha_{ijt} + (\Delta_i)^j (1 - \Delta_i)^{1-j} \sum_{n=1}^{N} \delta_{ni} \phi_{nt} \tag{14}$$

$$\begin{aligned}
&\log q(\lambda) \propto \mathbb{E}\left[\log p(\lambda|\xi) + \log p(z|u, \lambda, \theta) + \log p(y|\lambda, \rho)\right] \\
&= \sum_{i=1}^{C} \lambda_i \log \xi_i + (1 - \lambda_i) \log(1 - \xi_i) + \sum_{n=1}^{N} \sum_{t=1}^{T} \sum_{i=1}^{C} \sum_{j=0}^{1} \lambda_i^j (1 - \lambda_i)^{1-j} \mathbb{E}[u_{ni}] \mathbb{E}[z_{nt}] \mathbb{E}[\log \theta_{ijt}] \\
&\quad + \sum_{j=1}^{K} \sum_{i=1}^{C} [\lambda_i y_{ji} + (1 - \lambda_i)(1 - y_{ji})] \log \rho_j + [(1 - \lambda_i) y_{ji} + \lambda_i (1 - y_{ji})] \log 1 - \rho_j
\end{aligned}$$

Hence,

$$\log \Delta_i \propto \log \xi_i + \sum_{j=1}^{K} y_{ji} \log \rho_j + (1 - y_{ji}) \log 1 - \rho_j + \sum_{n=1}^{N_d} \sum_{t=1}^{T} \delta_{ni} \phi_{nt} \mathbb{E}\left[\log \theta_{i1t}\right] \tag{15}$$

$$\begin{aligned}
\log(1 - \Delta_i) \propto \log 1 - \xi_i + \sum_{j=1}^{K} (1 - y_{ji}) \log \rho_j + y_{ji} \log(1 - \rho_j) \\
+ \sum_{n=1}^{N_d} \sum_{t=1}^{T} \delta_{ni} \phi_{nt} \mathbb{E}\left[\log \theta_{i0t}\right]
\end{aligned} \tag{16}$$

In all the above update rules, $\mathbb{E}[\log \theta_{ijt}^d] = \psi(\gamma_{ijt}) - \psi(\sum_{t'=1}^{T} \gamma_{ijt'}^d)$, where $\psi(.)$ is the digamma function. Also, terms involving y_{ji} are considered only when $y_{ji} \neq -1$. For the non-crowd model ML-PA-LDA, the variational parameter Δ_i is absent as λ_i is observed. Therefore the E-step boils down to computing the updates for ϕ, δ and γ from Eqns 11, 13 and 14 with Δ_i replaced by λ_i.

4.2 M-step Updates for ML-PA-LDA-C

In the M-step, the parameters ξ, ρ, β and α are estimated using the values of $\Delta^d, \phi^d, \delta^d, \gamma^d$ estimated from the E-step. The function $\mathcal{L}(\Theta)$ in Eqn 1 is maximized with respect to the parameters π yielding the following update equations.
Updates for ξ: for $i = 1, \ldots, C$.

$$\xi_i = \frac{\sum_{d=1}^{D} \Delta_i^d}{D} \tag{17}$$

Intuitively, Eqn 17 makes sense as ξ_i is the probability that any document in the corpus belongs to class i. Δ_i^d is the probability that document d belongs to class i and is computed in the E-step. Therefore ξ_i is an average of Δ_i^d over all

documents.
Updates for ρ: for $j = 1, \ldots, K$:

$$\rho_j = \frac{\sum_{d=1}^{D} \sum_{i=1}^{C} \mathbb{1}\left[y_{ji}^d \neq -1\right] \left[y_{ji}^d \Delta_i^d + (1 - y_{ji}^d)(1 - \Delta_i^d)\right]}{\sum_{d=1}^{D} \sum_{i=1}^{C} \mathbb{1}\left[y_{ji}^d \neq -1\right] \left[y_{ji}^d \Delta_i^d + (1 - y_{ji}^d)(1 - \Delta_i^d) + y_{ji}^d (1 - \Delta_i^d) + (1 - y_{ji}^d)\Delta_i^d\right]} \tag{18}$$

From Eqn 18, we observe that ρ_j is the fraction of times that crowd-worker j has provided a label that is consistent with the probability estimate Δ_i^d over all classes i. The implicit assumption is that every crowd-worker has provided at least one label, otherwise such a crowd-worker need not be considered in the model.
Updates for β: for $t = 1, \ldots, T$; for $j = 1, \ldots, V$:

$$\beta_{tj} = \frac{\sum_{d=1}^{D} \sum_{n=1}^{N_d} w_{nj}^d \phi_{nt}^d}{\sum_{d=1}^{D} N_d} \tag{19}$$

Intuitively, the variational parameter ϕ_{nt}^d is the probability that the word w_n^d is associated with topic t. Having updated this parameter in the E-step, β_{tj} computes the fraction of times the word j is associated with topic t by giving a weight ϕ_{nt}^d to its occurrence in document d.
Updates for α:
There do not exist closed form updates for α parameters. Hence we use Newton Raphson (NR) method to iteratively obtain the solution as follows.

$$\alpha_{ijr}^{t+1} = \alpha_{ijr}^t - \frac{g_r - c}{h_r} \tag{20}$$

where,

$$c = \frac{\sum_{\tau=1}^{T} g_\tau / h_\tau}{z^{-1} + \sum_{\tau=1}^{T} 1/h_\tau}, z = D\psi'\left(\sum_{t'=1}^{T} \alpha_{ijt'}^t\right), h_\tau = -D\psi'(\alpha_{ijr}^t),$$

$$g_r = D\left[\psi\left(\sum_{\tau=1}^{T} \alpha_{ij\tau}^t\right) - \psi\left(\alpha_{ijr}^t\right)\right] + \sum_{d=1}^{D}\left[\psi\left(\gamma_{ijr}^d\right) - \psi\left(\sum_{\tau=1}^{T} \gamma_{ij\tau}^d\right)\right]$$

The M-step updates for β and α involved in ML-PA-LDA (non-crowd version) are same as the updates in the crowd version, ML-PA-LDA-C. The parameter ρ is absent in ML-PA-LDA. Eqn 17, with Δ_i^d replaced by λ_i^d (as in the E-step) is used to update ξ_i. The overall algorithm for learning the parameters is provided in Algorithm 1.

Inference For inference on unseen test documents, only the E-step updates are performed till convergence, for each document. Subsequently, the following rule is used for getting the actual prediction:
Aggregation rule for predicting document labels: Δ_i gives a probabilistic estimate

Algorithm 1 Algorithm for learning the parameters π during training phase.

repeat
  for $d = 1$ to D **do**
    Initialize Θ^d ▷ E-step
    repeat
      Update Θ^d sequentially using Eqns 11, 13, 14, 15 and 16.
    until convergence
  end for ▷ M-step
  Update ξ using Eqn 17
  Update ρ using Eqn 18
  Update β using Eqn 19
  Perform NR updates for α using Eqn 20, till convergence.
until convergence

corresponding to class i. In order to predict the labels of any document, a suitable threshold (say 0.5) can be applied on the value of Δ_i so that if $\Delta_i >$ threshold, the estimate for λ_i, that is, $\hat{\lambda}_i = 1$.

5 Smoothing

In the model described in Section 3, we modeled β to be a parameter that governs the multinomial distributions for generating the words from each topic. In general, a new document can include words that have not been encountered in any of the training documents. The unsmoothed model described earlier does not handle this issue. In order to handle this, we must "smoothen" the multinomial parameters involved [2]. One way to perform smoothing is to model β as a multinomial random variable over the vocabulary ν, with parameters η. Again due to the intractable nature of the computations, we model the variational distribution for β as $\beta \sim \text{Mult}(\chi)$. We estimate the variational parameter χ in the E-step of variational EM using Eqn 21 assuming η is known.

$$\chi_{tj} = \eta_{tj} + \sum_{d=1}^{D} \sum_{n=1}^{N_d} \phi_{nt}^d w_{nj}^d \tag{21}$$

The model parameter η is estimated in the M-step using Newton Raphson method as follows.

$$\eta_{ir}^{t+1} = \eta_{ir}^t - \frac{g_r - c}{h_r} \tag{22}$$

where,

$$c = \frac{\sum_{\tau=1}^{V} g_\tau / h_\tau}{z^{-1} + \sum_{\tau=1}^{T} 1/h_\tau}, \quad z = \psi'\left(\sum_{j'=1}^{V} \eta_{ij'}^t\right), h_r = -\psi'(\eta_{ir}^t),$$

$$g_r = \left[\psi\left(\sum_{j'=1}^{V} \eta_{ij'}^t\right) - \psi\left(\eta_{ir}^t\right)\right] + \left[\psi\left(\chi_{ir}\right) - \psi\left(\sum_{j'=1}^{V} \chi_{ij'}\right)\right]$$

The steps for the derivation are similar to the steps for non-smooth version.

6 Experiments

In order to test the efficacy of the proposed techniques, we evaluate our model on datasets from several domains.

6.1 Dataset Descriptions

We have carried out our experiments on several datasets from the text domain as well as non-text domain. Our code is available on bitbucket [1]. We now describe the datasets and the pre-processing steps below.

Text Datasets In the text domain, we have performed studies on the Reuters-21578, Bibtex and Enron datasets.
Reuters-21578: The Reuters-21578 dataset [14] is a collection of documents with news articles. The original corpus had 10,369 documents and a vocabulary of 29930 words. We performed stemming using the Porter Stemmer algorithm [17] and also removed the stop words. From this set the words which occurred more than 50 times across the corpus were retained and only documents which contained more than 20 words were retained. Finally the most commonly occurring top 10 labels were retained namely acq, crude, earn, fx, grain , interest, money, ship, trade, wheat. This led to a total of 6547 documents and a vocabulary of size 1996. Of these, a random 80% was used as training set and the remaining 20% as test.
Bibtex: The Bibtex dataset [12] was released as part of the ECML-PKDD 2008 Discovery Challenge. The task is to assign tags such as physics, graph, electrochemistry etc to bibtex entries. There are a total of 4880 and 2515 entries in the training set and test respectively. The size of the vocabulary is 1836 and the number of tags is 159.
Enron: The Enron dataset [24] is a collection of emails for which a set of pre-defined categories are to be assigned. There are a total of 1123 and 573 training and test instances respectively with a vocabulary of 1001 words. The total number of email tags are 53.

Non-text Datasets We also evaluate our model on datasets from domains other than text, where the notion of words is not explicit.
Converting real valued features to words: Since we assume a bag-of-words model, we must replace every real-valued feature with a 'word' from a 'vocabulary'. We begin by choosing an appropriate size for the vocabulary. Thereafter, we collect every real number which occurs across features and instances in the corpus into a set. We then cluster this set into V clusters, using the k-means algorithm, where

[1] https://bitbucket.org/divs1202/ml-pa-lda-c

V is the size of the vocabulary previously chosen. Therefore, each real valued feature has a new representative word given by the nearest cluster center to the feature under consideration. The corpus is then generated with this new feature representation scheme.

Yeast: The Yeast dataset [9] contains a set of genes which may be associated with several functional classes. There are 1500 training examples and 917 examples in the test set with a total of 14 classes and 103 real valued features.

Scene: The Scene dataset [4] is a dataset of images. The task is to classify images into the following 6 categories- beach, sunset, fall, field, mountain, urban. The dataset contains 1211 instances in the training set and 1196 instances in the test set with a total of 294 real valued features.

In our experiments, we use the measures, accuracy across classes, micro-f1 score and average class log likelihood on the test sets to evaluate our model. Let TP, TN, FP and FN denote the number of true positives, true negatives, false positives and false negatives respectively with respect to all classes. Then the overall accuracy is computed as:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{23}$$

The micro-f1 is the harmonic of micro-precision and micro-recall where

$$\begin{aligned} \text{micro-precision} &= TP/(TP + FP) \text{ and} \\ \text{micro-recall} &= TP/(TP + FN) \end{aligned} \tag{24}$$

The average class log-likelihood on the test instances is computed as follows:

$$\text{log-l} = \frac{\sum_{d=1}^{D_{test}} \sum_{i=1}^{C} \lambda_i^d \log \Delta_i^d + (1 - \lambda_i^d) \log(1 - \Delta_i^d)}{D_{test} \times C}$$

where D_{test} is the number of instances in the test set.Further details on these measures can be found in the survey [10].

6.2 Results: ML-PA-LDA (Non-Crowd Version)

We run our model first assuming labels from a perfect source.

In Table 1 we compare the performance of our non-annotator model vs other methods such as RAKel, Monte Carlo Classifier Chains (MCC) [19], Binary Relevance Method - Random Subspace (BRq) [20], Bayesian Chain Classifiers (BCC) [27] and SLDA. BCC [27] is a probabilistic method which constructs a chain of classifiers by modeling the dependencies between the classes using a Bayesian network. MCC instead uses a Monte-Carlo strategy to learn the dependencies. BRq improves upon binary relevance methods of combining classifiers by constructing an ensemble. As mentioned earlier RAKel draws subsets of the classes, each of size k and constructs ensemble classifiers. The implementations of RAKel, MCC, BRq and BCC provided by Meka (http://meka.sourceforge.net/) were used. For SLDA the code provided by the authors was used. On the reuters, bibtex and enron datasets, ML-PA-LDA (without the annotators) performs significantly better than SLDA. On scene and yeast datasets, ML-PA-LDA and

SLDA give the same performance. It is to be noted that these datasets, known to be hard, are from the images and biology domains respectively. As can be seen from the table, our model gives a better overall performance than SLDA and also does not require training C binary classifiers. This advantage is a significant one, especially in datasets such as bibtex where the number of classes is 159.

Dataset	RAKel (J48)	MCC	BRq	BCC	SLDA	ML-PA-LDA	ML-PA-LDA-C
reuters	0.881	0.876	0.863	0.867	0.897	**0.969**	**0.942**
bibtex	0.293	0.290	0.309	0.299	0.984	**0.984**	**0.981**
enron	0.402	0.389	0.430	0.411	0.937	**0.939**	**0.938**
scene	0.577	0.580	0.550	0.594	**0.823**	**0.823**	**0.818**
yeast	0.415	0.432	0.462	0.413	**0.767**	**0.767**	**0.767**

Table 1: Comparison of average accuracy of various multi-label classification techniques

We compared the performance of our algorithm with the size of the datasets used for training as well as the number of topics used. The results of our model are shown in Figures 2c, 2f and 2i. An increase in the size of the dataset improves the performance of our model with respect to all the measures in use. Similarly an increase in the number of topics generally improves the measures under consideration. A striking observation is the low accuracy, log likelihood and micro-f1 scores associated with the model when the number of topics = 80 (eight times the number of classes) and the size of the dataset is low (S=25%). This is expected as the number of parameters to be estimated is too large to be learned using very few training examples. However as more training data is available, the model achieves enhanced performance. This observation is consistent with Occam's razor [3].

6.3 Results: ML-PA-LDA-C (Crowd Version)

To verify the performance of the annotator model where the labels are provided by multiple noisy annotators, we simulated 50 annotators with varying qualities. The ρ values of the annotators were sampled from a uniform distribution. For 10 of these annotators, ρ was sampled from $U[0.51, 0.65]$. For another 20 of them, ρ was sampled from $U[0.66, 0.85]$ and for the remaining 20 of them ρ was sampled from $U[0.86, 0.9999]$. This captures the heterogeneity in the annotator qualities. For each document in the training set, a random 10% (= 5) annotators were picked for generating the noisy labels.

In Table 1, we report the performance of the annotator model. We find that the performance of ML-PA-LDA-C is close to that of ML-PA-LDA and many a time better than or at par with SLDA (from Table 1), in spite of having access to only noisy labels. On scene and yeast datasets, ML-PA-LDA-C, ML-PA-LDA and SLDA give the same performance. In Table 2, we compare the

performance of ML-PA-LDA-C and ML-PA-LDA on the reuters dataset under varying amounts of training data. With more training data, both models perform better. We also report 'Ann RMSE' which is the L2 norm of the difference in predicted qualities of the annotators vs the true qualities. Ann RMSE $= \sqrt{\sum_{j=1}^{K} |\hat{\rho}_j - \rho_j|^2 / K}$ where $\hat{\rho}_j$ is the quality of annotator j as predicted by our variational EM algorithm and ρ_j is the true annotator quality which is unknown during training. We find that 'Ann RMSE' decreases as more training data is available showing the efficacy of our model for learning the qualities of the annotators.

% of training set used	ML-PA-LDA avg accuracy	ML-PA-LDA avg microf1	ML-PA-LDA-C avg accuracy	ML-PA-LDA-C avg microf1	Ann RMSE
10	0.949	0.762	0.927	0.616	0.023
30	0.953	0.784	0.930	0.619	0.014
50	0.955	0.787	0.936	0.629	0.011
70	0.961	0.828	0.937	0.650	0.010
100	0.969	0.829	0.942	0.669	0.009

Table 2: Performances of ML-PA-LDA and ML-PA-LDA-C for different sizes of training set, for a fixed number of topics (= 20). Results are shown for Reuters dataset. Similar trend is demonstrated by other datasets (omitted for space).

Similar to the experiment carried out on ML-PA-LDA, we vary the number of topics as well as data-set sizes and compute all the measures used. The plots are shown in Figure 2(first two columns) and help in understanding how T, the number of topics must be tuned depending on the size of the available training set. As in ML-PA-LDA, an increase in the topics as well as dataset size improves the performance of ML-PA-LDA-C in general. Therefore as more training data becomes available, having more number of topics helps.

Adversarial Annotators We also tested the robustness of our model against labels from adversarial or malicious annotators. An adversarial annotator is characterized by a quality parameter $\rho < 0.5$. As in the previous case, we simulated 50 annotators. The ρ values of 10 of them was sampled from $U[0.0001, 0.1]$. For another 15 annotators, ρ was sampled from $U[0.51, 0.65]$. For another 20 of them ρ was sampled from $U[0.66, 0.85]$ and for the remaining 5 of them ρ was sampled from $U[0.86, 0.9999]$. The choice of the proportion of malicious annotators is as per literature [18]. On the Reuters dataset, we obtained an average accuracy of **0.955**, average class log likelihood of **-0.193**, average micro-f1 of **0.793** and an average ann-rmse of **0.002** over five runs, with 40 topics. This shows that even in the presence of malicious annotators, our model remains unaffected and performs well.

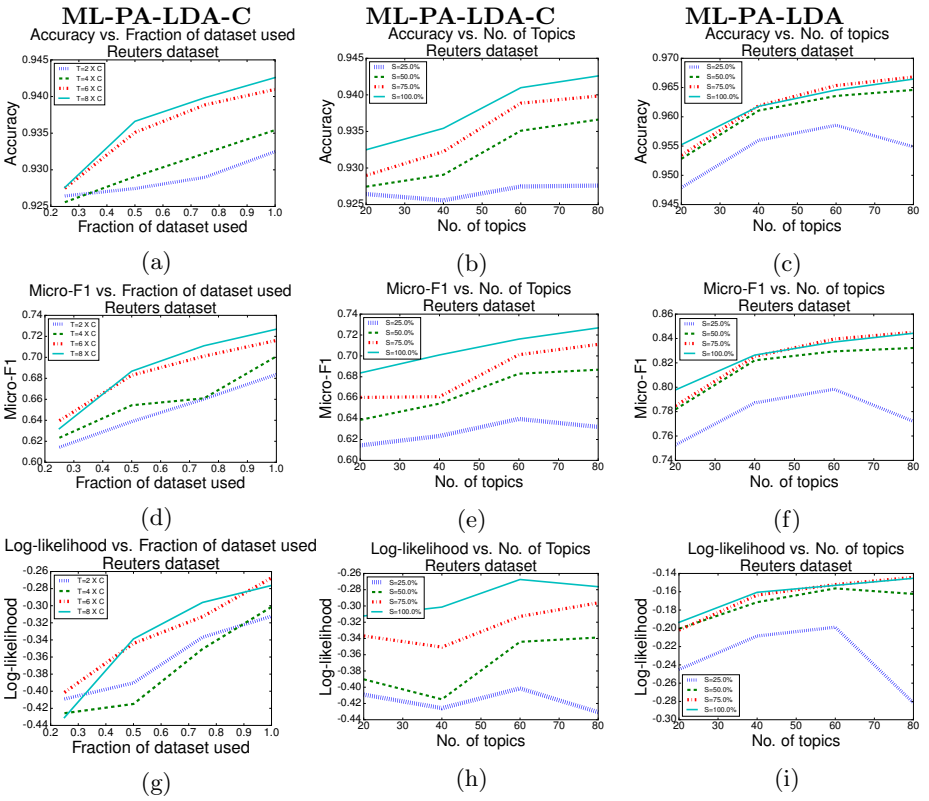


Fig. 2: Performance of ML-PA-LDA and ML-PA-LDA-C on Reuters dataset. T is the number of topics, C is the number of classes and S is the percentage of dataset used for training. The graphs show the trend in the various measures as a function of number of examples in the training set as well as number of topics. Other datasets (omitted for space) follow a similar trend. The last column - Figures 2c, 2f and 2i are the results for the non-crowd version (ML-PA-LDA) whereas all other plots study the performance of the crowd-version (ML-PA-LDA-C)

References

1. C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
2. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 2003.
3. A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam's razor. *Inf. Process. Lett.*, 1987.
4. M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern recognition*, 2004.
5. J. Bragg, Mausam, and D. S. Weld. Crowdsourcing multi-label classification for taxonomy creation. In *HCOMP*, 2013.

6. E. A. Cherman, M. C. Monard, and J. Metz. Multi-label problem transformation methods: a case study. *CLEI Electron. J.*, 2011.
7. J. Deng, O. Russakovsky, J. Krause, M. S. Bernstein, A. Berg, and L. Fei-Fei. Scalable multi-label annotation. In *CHI*, 2014.
8. L. Duan, O. Satoshi, H. Sato, and M. Kurihara. Leveraging crowdsourcing to make models in multi-label domains interoperable. *Technical Report*, 2014.
9. A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *NIPS*, 2001.
10. E. Gibaja and S. Ventura. A tutorial on multilabel learning. *ACM Computing Surveys*, 2015.
11. G. Heinrich. A generic approach to topic models. In *ECML-PKDD*. 2009.
12. I. Katakis, G. Tsoumakas, and I. Vlahavas. Multilabel text classification for automated tag suggestion. In *ECML/PKDD Discovery Challenge*, 2008.
13. R. Krestel and P. Fankhauser. Tag recommendation using probabilistic topic models. *ECML PKDD Discovery Challenge*, 2009.
14. M. Lichman. UCI machine learning repository, 2013.
15. J. D. Mcauliffe and D. M. Blei. Supervised topic models. In *NIPS*. 2008.
16. A. K. McCallum. Multi-label text classification with a mixture model trained by em. In *AAAI 99 Workshop on Text Learning*, 1999.
17. M. F. Porter. An algorithm for suffix stripping. *Program*, pages 130–137, 1980.
18. V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *JMLR*, 2010.
19. J. Read, L. Martino, and D. Luengo. Efficient monte carlo optimization for lulti-label classifier chains. In *ICASSP*, 2013.
20. J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine Learning*, 2011.
21. F. Rodrigues, B. Ribeiro, M. Lourenço, and F. Pereira. Learning supervised topic models from crowds. In *HCOMP*, 2015.
22. T. N. Rubin, A. Chambers, P. Smyth, and M. Steyvers. Statistical topic models for multi-label document classification. *Machine Learning*, 2012.
23. G. Tsoumakas, I. Katakis, and I. Vlahavas. Random k-labelsets for multilabel classification. *TKDE*, 2011.
24. G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas. Mulan: A java library for multi-label learning. *JMLR*, 2011.
25. N. Ueda and K. Saito. Parametric mixture models for multi-labeled text. In *NIPS*, 2003.
26. H. Wang, M. Huang, and X. Zhu. A generative probabilistic model for multi-label classification. In *ICDM*, 2008.
27. J. H. Zaragoza, L. E. Sucar, E. F. Morales, C. Bielza, and P. Larrañaga. Bayesian chain classifiers for multidimensional classification. In *IJCAI*, 2011.