Extreme Multi-label Classification from Aggregated Labels

Yanyao Shen [1] **Hsiang-fu Yu** [2] **Sujay Sanghavi** [1 2] **Inderjit Dhillon** [2 3]

Abstract

Extreme multi-label classification (XMC) is the problem of finding the relevant labels for an input, from a very large universe of possible labels. We consider XMC in the setting where labels are available only for groups of samples - but not for individual ones. Current XMC approaches are not built for such multi-instance multi-label (MIML) training data, and MIML approaches do not scale to XMC sizes. We develop a new and scalable algorithm to impute individual-sample labels from the group labels; this can be paired with any existing XMC method to solve the aggregated label problem. We characterize the statistical properties of our algorithm under mild assumptions, and provide a new end-to-end framework for MIML as an extension. Experiments on both aggregated label XMC and MIML tasks show the advantages over existing approaches.

1. Introduction

Extreme multi-label classification (XMC) is the problem of finding the relevant labels for an input from a very large universe of possible labels. XMC has wide applications in machine learning including product categorization (Agrawal et al., 2013; Yu et al., 2014), webpage annotation (Partalas et al., 2015) and hash-tag suggestion (Denton et al., 2015), where both the sample size and the label size are extremely large. Recently, many XMC methods have been proposed with new benchmark results on standard datasets (Prabhu et al., 2018a; Guo et al., 2019; Jain et al., 2019).

XMC problem, as well as many other modern machine learning problems, often require a large amount of data. As the size of the data grows, the annotation of the data becomes less accurate, and large-scale data annotation with high quality becomes growingly expensive. As a result, modern machine learning applications need to deal with certain types of weak supervision, including partial but noisy labeling and active labeling. These scenarios lead to exploration of advanced learning methods including semi(self)-supervised learning, robust learning and active learning.

In this paper, we study a typical weak supervision setting for XMC named Aggregated Label eXtreme Multi-label Classification (AL-XMC), where only aggregated labels are provided to a group of samples. AL-XMC is of interest in many practical scenarios where directly annotated training data can not be extracted easily, which is often due to the way data is organized. For example, Wikipedia contains a set of annotated labels for every wiki page, and can be used by an XMC algorithm for the task of tagging a new wiki page. However, if one is interested in predicting keywords for a new wiki *paragraph*, there is no such directly annotated data. Similarly, in e-commerce, the attributes of a product may not be provided directly, but the attributes of the brand of the product may be easier to extract. To summarize, it is often easier to get aggregated annotations belonging to a group of samples. This is known as multi-instance multi-label (MIML) (Zhou et al., 2012) problem in the non-extreme label size setting.

AL-XMC raises new challenges that standard approaches are not able to address. Because of the enormously large label size, directly using MIML methods leads to computation and memory issues. On the other hand, standard XMC approaches suffer from two main problems when directly applied to AL-XMC: (1) higher computation cost due to increased number of positive labels, and (2) worse performance due to ignoring of the aggregation structure. In this work, we propose an Efficient AGgregated Label lEarning algorithm (EAGLE) that assigns labels to each sample by learning label embeddings based on the structure of the aggregation. More specifically, the key ingredient of EAGLE follows the simple principle that *the label embedding should be close to the embedding of at least one of the sample points in every positively labeled group*. We first formulate such an estimator, then design an iterative algorithm that takes projected gradient steps to approximate it. As a by-product, our algorithm naturally extends to the non-XMC setting as a new end-to-end framework for the MIML problem. Our main contributions include:

[1]Department of ECE, The University of Texas at Austin, TX, USA. [2]Amazon, CA, USA. [3]Department of Computer Science, The University of Texas at Austin, TX, USA.. Correspondence to: Yanyao Shen <shenyanyao@utexas.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

- We propose to study AL-XMC, which has significant impact for modern machine learning applications. We propose an efficient and robust algorithm EAGLE with low computation cost (**Section 4**) that can be paired with any existing XMC method for solving AL-XMC.
- We provide theoretical analysis for EAGLE and show the benefit of label assignment, the property of the estimator and the convergence of our iterative update (**Section 5**).
- The proposed method can be easily extended to the regular (non-extreme) MIML setting (**Section 6**). Our solution can be viewed as a co-attention mechanism between labels and samples. We empirically show its benefit over previous MIML framework in **Section 7**.

2. Related Work

Extreme multi-label classification (XMC). The most classic and straightforward approach for XMC is based on the One-Vs-All (OVA) method (Yen et al., 2016; Babbar & Schölkopf, 2017; Liu et al., 2017; Yen et al., 2017), which treats each label separately and learns a classifier for each label. OVA has shown to achieve high accuracy, but the computation is too expensive for extremely large label set. Tree-based methods, on the other hand, try to improve the efficiency of OVA by using hierarchical representations for samples or labels (Agrawal et al., 2013; Prabhu & Varma, 2014; Jain et al., 2016; Si et al., 2017; Siblini et al., 2018; Prabhu et al., 2018b). Among these approaches, label partitioning based methods, including Parabel (Prabhu et al., 2018b), have achieved leading performances with training cost sub-linear in the number of labels. Apart from tree-based methods, embedding based methods (Zhang et al., 2018; Chang et al., 2019; You et al., 2019; Guo et al., 2019) have been studied recently in the context of XMC in order to better use the textual features. In general, while embedding based methods may learn a better representation and use the contextual information better than tf-idf, the scalability of these approaches is worse than tree-based methods. Recently, Medini et al. (2019) apply sketching to learn XMC models with label size at the scale of 50 million, and the connection between softmax and negative sampling approach (Jain et al., 2019), hierarchical softmax with partitioned label trees are shown in (Bamler & Mandt, 2020; Wydmuch et al., 2018).

Multi-instance multi-label learning (MIML). MIML (Zhang & Zhang, 2007) is a general setting that includes both multi-instance learning (MIL) (Dietterich et al., 1997; Maron & Lozano-Pérez, 1998) and multi-label learning (MLL) (McCallum, 1999; Zhang & Zhou, 2013). AL-XMC can be categorized as a special MIML setting with extreme label size. Recently, Feng and Zhou (2017) propose the general deep MIML architecture with a 'concept' layer and two max-pooling layers to align with the multi-instance nature of the input. In contrast, our approach learns label representations to use them as one branch of the input. On the other hand, Ilse et al. (2018) adopt the attention mechanism for multi-instance learning. Similar attention-based mechanisms are later used in learning with sets (Lee et al., 2019) but focus on a different problem. Our label assignment based algorithm EAGLE can be viewed as an analogy to the attention-based mechanisms, while having major differences from previous work. EAGLE provides the intuition that attention truly happens between the label representation and the sample representation, while previous methods do not. The idea of jointly considering sample and label space exists in the multi-label classification problems in vision (Weston et al., 2011; Frome et al., 2013). While sharing the similar idea of learning a joint input-label space, our work addresses the multi-instance learning challenges as well as scalability in the XMC setting.

Others. AL-XMC is also related to a line of theoretical work on learning with shuffled labels and permutation estimation (Collier & Dalalyan, 2016; Pananjady et al., 2017b; Abid et al., 2017; Pananjady et al., 2017a; Hsu et al., 2017; Haghighatshoar & Caire, 2017), where the labels of all samples are provided without correspondences. Our work uniquely focuses on an aggregation structure where we know the group-wise correspondences. Our targeted applications have extreme label size that makes even classically efficient estimators hard to compute. Another line of work studies learning with noisy labels (Natarajan et al., 2013; Liu & Tao, 2015), where one is interested in identifying the subset of correctly labeled samples (Shen & Sanghavi, 2019b), but there is no group-to-group structure. More broadly, in the natural language processing context, indirect supervision (Chang et al., 2010; Wang & Poon, 2018) tries to address the label scarcity problem where large-scale coarse annotations are provided with very limited fine-grain annotations.

3. Problem Setup and Preliminaries

In this section, we first provide a brief overview of XMC, whose definition helps us formulate the aggregated label XMC (AL-XMC) problem. Based on the formulation, we use one toy example to illustrate the shortcomings of existing XMC approaches when applied to AL-XMC.

XMC and AL-XMC formulation. An XMC problem can be defined by $\{\mathbf{X}, \mathbf{Y}\}$, where $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the feature matrix for all n samples, $\mathbf{Y} \in \{0, 1\}^{n \times l}$ is the sample-to-label binary annotation matrix with label size l (if sample i is annotated by label k then $\mathbf{Y}_{i,k} = 1$). For the AL-

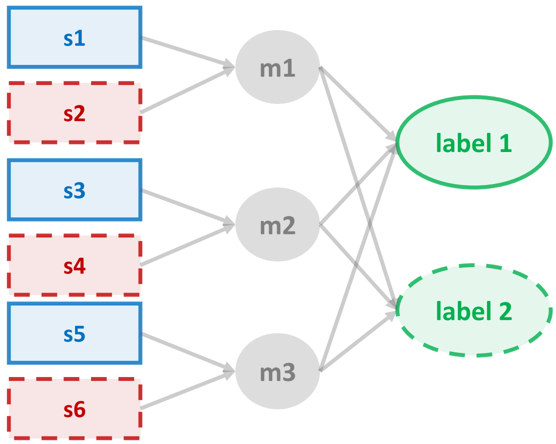


Figure 1: A toy example of AL-XMC. Label 1 & 2 are both tagged for each group in $\{s_1, s_2\}, \{s_3, s_4\}, \{s_5, s_6\}$. Samples s_1, s_3, s_5 all have feature $\mathbf{x}$, while s_2, s_4, s_6 all have feature $-\mathbf{x}$. A good model should identify that s_1, s_3, s_5 and s_2, s_4, s_6 belong to two labels, respectively (order does not matter). Ignoring the aggregation structure makes standard XMC approaches fail completely.

XMC problem, however, such a clean annotation matrix is not available. Instead, aggregated labels are available for subsets of samples. We use m intermediate nodes to represent this aggregation, where each node is connected to a subset of samples and gets annotated by multiple labels. More specifically, AL-XMC can be described by $\{\mathbf{X}, \mathbf{Y}^1, \mathbf{Y}^2\}$, where the original annotation matrix is replaced by two binary matrices $\mathbf{Y^1} \in \{0,1\}^{n \times m}$ and $\mathbf{Y^2} \in \{0,1\}^{m \times l}$. $\mathbf{Y}^1$ captures how the samples are grouped, while $\mathbf{Y}^2$ captures the labels for the aggregated samples. The goal is to use $\{\mathbf{X}, \mathbf{Y}^1, \mathbf{Y}^2\}$ to learn a good extreme multi-label classifier. Let $\bar{g} = \mathtt{nnz}(\mathbf{Y}^1)/m$ denote the average group size. In general, the larger $\bar{g}$ is, the weaker the annotation quality becomes. For convenience, let $\mathcal{N}, \mathcal{M}, \mathcal{L}$ be the set of samples, intermediate nodes, and labels, respectively. Let $\mathcal{N}_j, \mathcal{L}_j$ be the set of samples, labels linked to intermediate node j respectively, $\forall j \in \mathcal{M}$, and $\mathcal{M}_k$ be the set of nodes in $\mathcal{M}$ connected to label $k \in \mathcal{L}$[1]. Let $\mathbf{x}_i^\top$ be the i-th row in $\mathbf{X}$ for $i \in \mathcal{N}_j$, $j \in \mathcal{M}$, and $\mathbf{X}_\mathcal{S}$ be the submatrix of $\mathbf{X}$ that includes rows with index in set $\mathcal{S} \subseteq \mathcal{N}$[2].

Deficiency of existing XMC approaches. Existing XMC approaches can be applied for AL-XMC by treating the product $\mathbf{Y}^1\mathbf{Y}^2$ directly as the annotation matrix, and learning a model using $\{\mathbf{X}, \mathbf{Y}^1\mathbf{Y}^2\}$. While using all labeling information, this simple treatment ignores the possible incorrect correspondences due to label aggregation. To see the problem of this treatment, we take the XMC method Parabel (Prabhu et al., 2018b) as an example. Notice that the deficiency generally holds for all standard XMC approaches, but it is convenient to illustrate for a specific XMC method. Parabel is a scalable algorithm that achieves good performance on standard XMC datasets based on a partitioned label tree: It first calculates label representations $\mathbf{L} = \mathbf{Y}^\top\mathbf{X}$ that summarize the positive samples associated with every label (**step 1**); Next, a balanced hierarchical partitioning of the labels is learned using $\mathbf{L}$ (**step 2**); Finally, a hierarchical probabilistic model is learned given the label tree (**step 3**). Notice that the label embedding that Parabel would use for AL-XMC if we naively use $\mathbf{Y}^1\mathbf{Y}^2$ as the annotation matrix is given by:

$$\mathbf{L} = \left(\mathbf{Y^1Y^2}\right)^\top \mathbf{X}. \tag{1}$$

Consider an example where there are n samples and 2 labels, with $\mathbf{X} \in \mathbb{R}^{n \times d}$, $\mathbf{Y}^1 \in \{0,1\}^{n \times \frac{n}{2}}$, $\mathbf{Y}^2 \in \{0,1\}^{\frac{n}{2} \times 2}$ defined as follows:

$$\mathbf{X} = \mathbf{1}_{\frac{n}{2}} \otimes \begin{bmatrix} \mathbf{x}^\top \\ -\mathbf{x}^\top \end{bmatrix}, \mathbf{Y^1} = \mathbf{I}_{\frac{n}{2}} \otimes \mathbf{1}_2, \mathbf{Y^2} = \mathbf{1}_{\frac{n}{2} \times 2},$$

where $\otimes$ is the Kronecker product, $\mathbf{1}_d(\mathbf{1}_{d_1 \times d_2})$ is an all-ones vector(matrix) with dimension $d(d_1 \times d_2)$ and $\mathbf{I}_d$ is the identity matrix with size d. A pictorial explanation is shown in Figure 1. The embedding calculated using the above $\mathbf{X}, \mathbf{Y^1}, \mathbf{Y^2}$ leads to $\mathbf{L} = \mathbf{0}_{n \times 2}$ and loses all the information. With this label embedding, the clustering algorithm in step 2 and the probabilistic model in step 3 would fail to learn anything useful. However, a good model for the above setting should classify samples with feature close to $\mathbf{x}$ as label 1 and samples with feature close to $-\mathbf{x}$ as label 2 (or vice versa). Such a failure of classic XMC approaches motivates us to provide algorithms that are robust for the AL-XMC problem.

4. Algorithms

The main insight we draw from the toy example above is that ignoring the aggregation structure may lead to serious information loss. Therefore, we propose an efficient and robust label embedding learning algorithm to address this. We start with the guiding principle of our approach, and then explain the algorithmic details.

Given an XMC dataset with aggregated labels, our key idea for finding the embedding for each label is the following:

> *The embedding of label $k \in \mathcal{L}$ should be close to the embedding of **at least one** of the samples in $\mathcal{N}_j$, $\forall j \in \mathcal{M}_k$.*

The closeness here can be any general characterization of similarity, e.g., the standard cosine similarity. According

[1] We slightly abuse the notation and use $\mathcal{N}_j, \mathcal{M}_k, \mathcal{L}_j$ for the corresponding index sets as well.

[2] We summarize all notations in the Appendix.

Algorithm 1 GROUP_ROBUST_LABEL_REPR (GRLR)

Inputs: $\mathcal{M}_k$, $\{\mathcal{N}_j\}_{j\in\mathcal{M}_k}$, $\mathbf{X}$.
Output: Label embedding $\mathbf{e}_k$.
Initialize: Set $\mathbf{e}^0 \leftarrow \texttt{Proj}\left(\sum_{j\in\mathcal{M}_k}\sum_{i\in\mathcal{N}_j}\mathbf{x}_i\right)$.
for $t = 1, \cdots, T$ **do** /* *where* $\texttt{Proj}(\mathbf{x}) := \mathbf{x}/\|\mathbf{x}\|$ */
  for $j \in \mathcal{M}_k$ **do**
    $v_{i,j} \leftarrow \langle \mathbf{e}^{t-1}, \mathbf{x}_i\rangle, \forall i \in \mathcal{N}_j$.
    $a_{i,j} \leftarrow \mathbf{1}\{v_{i,j} == \max_{i'\in\mathcal{N}_j} v_{i',j}\}, \forall i \in \mathcal{N}_j$.
  end for
  $\mathbf{g}^t \leftarrow \texttt{Proj}\left(\sum_{j\in\mathcal{M}_k}\sum_{i\in\mathcal{N}_j} a_{i,j}\mathbf{x}_i\right)$.
  $\mathbf{e}^t \leftarrow \texttt{Proj}\left(\mathbf{e}^{t-1} + \lambda\cdot\mathbf{g}^t\right)$.
end for
Return: $\mathbf{e}^T$.

Algorithm 2 EAGLE

Inputs: $\mathbf{X}, \mathbf{Y}^1 \in \{0,1\}^{n\times m}, \mathbf{Y}^2 \in \{0,1\}^{m\times l}$.
Output: A filtered XMC dataset.
$\mathbf{Y}_{\texttt{filter}} \leftarrow \mathbf{0}^{n\times l}, \mathcal{N} \leftarrow [n], \mathcal{M} \leftarrow [m], \mathcal{L} \leftarrow [l]$.
$\mathcal{N}_j \leftarrow \{i \in \mathcal{N} | \mathbf{Y}^1_{i,j} == 1\}, \forall j \in \mathcal{M}$.
for $k \in \mathcal{L}$ **do**
  $\mathcal{M}_k \leftarrow \{j \in \mathcal{M} | \mathbf{Y}^2_{j,k} == 1\}$.
  $\mathbf{e}_k \leftarrow$GRLR$(\mathcal{M}_k, \{\mathcal{N}_j\}_{j\in\mathcal{M}_k}, \mathbf{X})$.
end for
for $j \in \mathcal{M}$ **do**
  $\mathbf{Y}^{\texttt{filter}}(\arg\max_{i\in\mathcal{N}_j}\langle \mathbf{e}_k, \mathbf{x}_i\rangle, k) \leftarrow 1, \forall k \in \mathcal{L}_j$.
end for
Return: $\{\mathbf{X}, \mathbf{Y}^{\texttt{filter}}\}$.

to this rule, in the previous toy example, the optimal label embedding for both labels is either $[\mathbf{x}, -\mathbf{x}]$ or $[-\mathbf{x}, \mathbf{x}]$, instead of $[\mathbf{0}, \mathbf{0}]$. More formally, the label embedding for label $k \in \mathcal{L}$ is calculated based on the following:

$$\hat{\mathbf{e}}_k = \arg\max_{\mathbf{e}:\|\mathbf{e}\|=1} \sum_{j\in\mathcal{M}_k} \max_{i\in\mathcal{N}_j} \langle \mathbf{x}_i, \mathbf{e}\rangle. \tag{2}$$

where $\mathcal{M}_k, \mathcal{N}_j$ are as defined in Section 3.

The goal of (2) is to find label k's representation that is robust even when every group has samples not related to k. However, finding the estimator in (2) is in general hard, since the number of possible combinations for choosing the maximum in each group is exponential in n. We provide an iterative algorithm that alternates between the following two steps for T times to approximate this estimator: (i) identify the closest sample in each group given the current label embedding, and (ii) update the label embedding based on the direction determined by all currently closest samples. This is formally described in Algorithm 1.

The complete algorithm Efficient AGgregated Label lEarning (EAGLE) is formally described in Algorithm 2, whose output can be directly fed into any standard XMC solver. Given the label embedding and a set of samples connected to the same intermediate node, each positive label is assigned to the sample with highest similarity. Notice that calculating the label embedding using Algorithm 1 is *equivalent* to Parabel's label embedding in (1) if $\bar{g} = 1$, i.e., the standard XMC setting. For general $\bar{g}$,(1) may perform well if each sample in $\mathcal{N}_j$ contributes equally to the label $k \in \mathcal{L}$, for every node $j \in \mathcal{M}_k$. However, this is not always the case.

Complexity. Computational efficiency is one of the main benefits of EAGLE. Notice that for each label, only the samples belonging to its positively labeled groups are used. Let $\bar{d}$ be the average feature sparsity and assume each sample has $O(\log l)$ labels, the positive samples for each label is $O(n\log l/l\cdot\bar{g})$. Therefore, the total complexity for learning all label's embedding is $O(n\bar{d}\log l/l\cdot\bar{g}l) = O(n\bar{d}\bar{g}\log l)$. On the other hand, the time complexity for Parabel (one of the most efficient XMC solver) is $O(n\bar{d}\log l)$ for step 1, $O(l\bar{d}\log l)$ for step 2 and $O(n\bar{d}\log l)$ for step 3 (Prabhu et al., 2018b). Therefore, EAGLE paired with any standard XMC solver for solving AL-XMC adds very affordable pre-processing cost.

5. Analysis

In this section, we provide theoretical analysis and explanations to the proposed algorithm EAGLE in Section 4. We start with comparing two estimators under the simplified regression setting to explain when assigning labels to each sample is helpful. Next, we analyze the statistical property of the label embedding estimator defined in (2) in Theorem 3, and the one-step convergence result of the key step in Algorithm 1 in Theorem 4.

In EAGLE, a learned label embedding is used to assign each label to the 'closest' sample in its aggregated group. Therefore, we start with justifying when label assignment would help. Since the multi-label classification setting may complicate the analysis, we instead analyze a simplified regression scenario. Let $\mathbf{Z} \in \mathbb{R}^{n\times l}$ be the response of all n samples in $\mathbf{X}$. Given $\mathbf{B}^\star \in \mathbb{R}^{d\times l}$, each group in $\mathbf{Z}$ is generated according to

$$\mathbf{Z}_{\mathcal{N}_j} = \mathbf{\Pi}^j(\mathbf{X}_{\mathcal{N}_j}\mathbf{B}^\star + \mathbf{E}^j), j \in \mathcal{M}$$

where $\mathbf{E}^j$ is the noise matrix, and $\mathbf{\Pi}^j$ is an unknown permutation matrix. For simplicity, we assume each group includes g samples and the aggregation structure can be described by $\mathbf{Y}^1 = \mathbf{I}_m \otimes \mathbf{1}_g$, $\mathbf{Y}^2 = \left(\mathbf{I}_m \otimes \mathbf{1}_g^\top\right)\cdot\mathbf{Z}$ with $m = n/g$. If each row in $\mathbf{Z}$ is a one-hot vector, $\mathbf{Y}^2$ becomes a binary matrix and $\{\mathbf{X}, \mathbf{Y}^1, \mathbf{Y}^2\}$ corresponds to the standard AL-XMC problem. Our goal here is to recover the model parameter $\mathbf{B}^\star$, with $\|\mathbf{B}^\star\| = 1$ for convenience. We assume each row in $\mathbf{E}^j$ independently follows $\mathcal{N}(\mathbf{0}, \sigma_e^2\mathbf{I}_l)$, each sample feature is generated according to $\mathbf{x}_i = \bar{\mathbf{x}}_j + \mathbf{d}_i$

for $i \in \mathcal{N}_j, j \in \mathcal{M}$, where $\bar{\mathbf{x}}_j \sim \mathcal{N}(\mathbf{0}, \sigma_1^2 \mathbf{I}_d)$ describes the center of each group, and $\mathbf{d}_i \sim \mathcal{N}(\mathbf{0}, \sigma_2^2 \mathbf{I}_d)$ captures the deviation within the group. Notice that the special case of $\sigma_1 \ll \sigma_2$ corresponds to all samples are i.i.d. generated spherical Gaussians. In the other extreme, $\sigma_2 \ll \sigma_1$ corresponds to samples within each group are clustered well. We consider the following two estimators:

$$\begin{aligned} \hat{\mathbf{B}}_{\texttt{NoAS}} &= \texttt{LR}\left(\cup_{j\in\mathcal{M}} \left\{\left(\mathbf{1}_g^\top \mathbf{X}_{\mathcal{N}_j}, \mathbf{1}_g^\top \mathbf{Z}_{\mathcal{N}_j}\right)\right\}\right) \\ \hat{\mathbf{B}}_{\texttt{AS}} &= \texttt{LR}\left(\cup_{j\in\mathcal{M}} \cup_{i\in\mathcal{N}_j} \left\{(\mathbf{x}_{i'}, \mathbf{z}_i)\right\}\right) \end{aligned} \tag{3}$$

where $\texttt{LR}\left(\{(\mathbf{x}_i, \mathbf{z}_i)\}_{i\in[n]}\right) = \left(\sum_{i\in[n]} \mathbf{x}_i \mathbf{x}_i^\top\right)^{-1} \sum_{i\in[n]} \mathbf{x}_i \mathbf{z}_i^\top$ and $i' = \arg\min_{\bar{i}\in\mathcal{N}_j} \left\|\mathbf{z}_i - \hat{\mathbf{B}}_{\texttt{NoAS}}^\top \mathbf{x}_{\bar{i}}\right\|$. Here, $\hat{\mathbf{B}}_{\texttt{NoAS}}$ corresponds to the baseline approach that learns a model without label assignment. On the other hand, $\hat{\mathbf{B}}_{\texttt{AS}}$ corresponds to the estimator we learn after assigning each output in the group to the closest instance based on the residual norm using $\hat{\mathbf{B}}_{\texttt{NoAS}}$. We have the following result that describes the property of the two estimators:

Theorem 1. *Given the two estimators in (3), let* $\mathcal{R}_1 = \left\|\hat{\mathbf{B}}_{\texttt{NoAS}} - \mathbf{B}^\star\right\|$, $\mathcal{R}_2 = \left\|\hat{\mathbf{B}}_{\texttt{AS}} - \mathbf{B}^\star\right\|$, $\sigma_x = \sqrt{\sigma_1^2 + \sigma_2^2}$, *with* $n \geq c_0 pgd \log^2 d$, *the following holds with high probability (i.e.,* $1 - n^{-c_1}$*):*

$$\mathcal{R}_1 \leq O\left(\sqrt{\frac{1}{p(g\sigma_1^2 + \sigma_2^2)}}\sigma_e\right), \tag{4}$$

$$\mathcal{R}_2 \leq O\left(\sqrt{\frac{1}{pg\sigma_x^2}}\sigma_e\right) + O\left(\sqrt{\frac{\sigma_e^2}{\sigma_x^2} + \mathcal{R}_1^2}\sqrt{\frac{\sigma_e^2}{\sigma_x^2} + 1}\right). \tag{5}$$

We can see the pros and cons of the two estimators from the above theorem. The first term in (5) is the rate achieved by the maximum likelihood estimator with all correspondences given (known $\mathbf{\Pi}^j$s), while the second term is a bias term due to incorrect assignment. This bias term gets smaller as the measurement noise and the estimation error become smaller. In (4), when $\sigma_1 \gg \sigma_2$, $\hat{\mathbf{B}}_{\texttt{NoAS}}$ achieves the same rate as the optimal estimator, but when $\sigma_1 \ll \sigma_2$, the rate goes down from $n^{-1/2}$ to $(n/g)^{-1/2}$. This shows that $\hat{\mathbf{B}}_{\texttt{NoAS}}$ is close to optimal when the clustering quality is high (within group deviation is small), on the other hand, $\hat{\mathbf{B}}_{\texttt{AS}}$ is nearly optimal for all clustering methods, while having an additional bias term that depends on the measurement noise.

Next, we analyze the property of the estimator in (2), since our iterative algorithm tries to approximate it. We assume that for each label $k \in \mathcal{L}$, there is some ground truth embedding $\mathbf{e}_k^\star$, and each sample is associated with one of the labels. For sample i with ground truth label k, its feature vector $\mathbf{x}_i$ can be decomposed as: $\mathbf{x}_i = \mathbf{e}_k^\star + \boldsymbol{\epsilon}_i$. Without loss of generality, we only need to focus on the recovering of single label $k \in \mathcal{L}$. Further, for simplicity, let us assume that both $\mathbf{x}_i$ and $\mathbf{e}_k^\star$ have unit norm measured in Euclidean space. We introduce the following definition that describes the property of the data:

Definition 2. *Define* $\delta = \min_{k_1,k_2} \left\|\mathbf{e}_{k_1}^\star - \mathbf{e}_{k_2}^\star\right\|$ *to be the minimum separation between each pair of ground truth label embeddings. Define* $f(\gamma) = \max_{\mathcal{S}\subset\mathcal{M}, |\mathcal{S}|/|\mathcal{M}|\leq\gamma} \frac{1}{|\mathcal{S}|}\left\|\sum_{i\in\mathcal{S}} \epsilon_i\right\|$ *to be the maximum influence of the noise, for* $\gamma \in [0, 1]$, *and let* $f = f(1)$. *Define* $q = \max_{k_1,k_2} \frac{|\mathcal{M}_{k_1} \cap \mathcal{M}_{k_2}|}{\min\{\mathcal{M}_{k_1}, \mathcal{M}_{k_2}\}}$ *to be the maximum overlap between the set of intermediate nodes associated with two labels.*

Given the above definition, we have the following result showing the property of the estimator in (2):

Theorem 3. *With* $q, \delta > 0$, *the estimator in (2) satisfies:*

$$\langle \mathbf{e}_k^\star, \hat{\mathbf{e}}_k \rangle \geq 1 - rf - (\sqrt{2}r + 2)f^2, \tag{6}$$

where $r = \left(\left[1 - q - \frac{2f}{\delta}\right]_+\right)^{-1} - 1.$

Theorem 3 characterizes the consistency of the estimator as noise goes to zero. It also quantifies the influence of minimum separation as well as maximum overlap between labels. A smaller δ and a larger q both leads to harder identification problem, which is reflected in an increasing r in (6). We then provide the following one-step convergence analysis for each iteration in Algorithm 1.

Theorem 4. *Given label* k *and current iterate* $\mathbf{e}^t$. *Let* $\mathcal{S}_t^{good}$ *be the set of groups where* $\mathbf{e}^t$ *is closest to the sample belongs to label* k. *Denote* $|\mathcal{S}_t^{good}|/|\mathcal{M}_k| = \alpha_t$. *The next step iterate given by Algorithm 1 has the following one-step property:*

$$\langle \mathbf{e}_k^\star, \mathbf{e}^{t+1} \rangle \geq \alpha_t + (1 - \alpha_t)\left(\langle \mathbf{e}_k^\star, \mathbf{e}^t \rangle - \|\mathbf{e}_k^\star - \mathbf{e}^t\|\right) - f$$

and a sufficient condition for contraction is $\langle \mathbf{e}_k^\star, \mathbf{e}^t \rangle \leq 1 - 2\left(\frac{1-f/2}{\alpha_t} - 1\right)^2$.

The key idea of the proof is based on the following fact of our algorithm: if a sample that does not come from label k is selected by $\mathbf{e}^t$, then the distance from the selected incorrect sample to label k's embedding can be controlled by the distance to $\mathbf{e}^t$ and the distance from $\mathbf{e}^t$ to $\mathbf{e}_k^\star$, where the first term is small because of the selection rule. Please find the details of the proof in the Appendix. Theorem 4 shows how each iterate gets closer to the ground truth label embedding. Since our algorithm does not require any assumption on the group size, we do not show the connection between α_t and $\mathbf{e}^t$ (which requires more restrictive assumptions), but instead provide a sufficient condition to illustrate when the

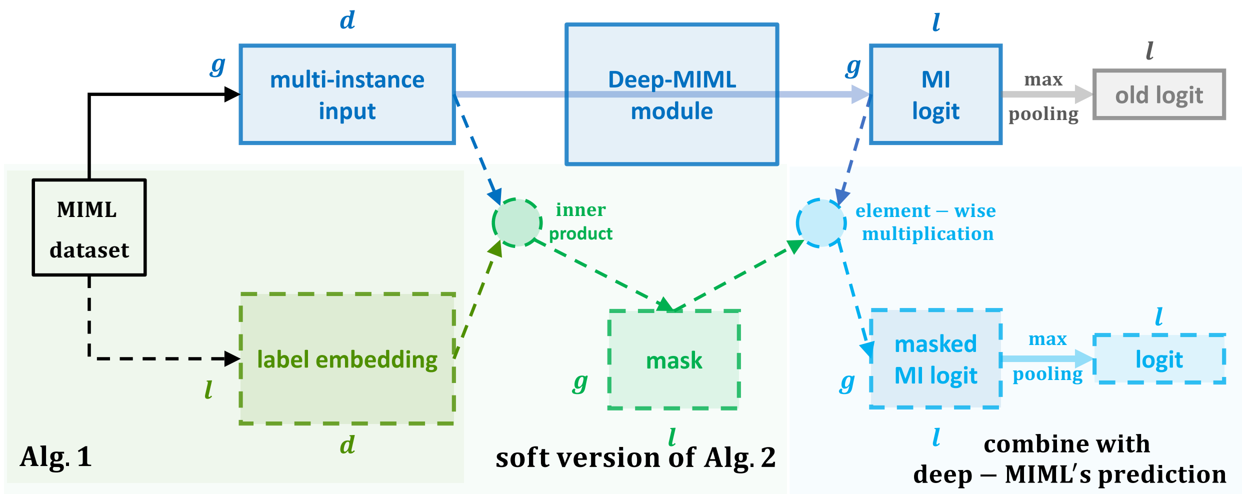


Figure 2: Illustration of extending EAGLE to standard MIML framework. The upper branch is the original deep MIML framework (Feng & Zhou, 2017), where the Deep-MIML module converts $\mathbf{X}$ to $\mathbf{V}_2$ as in (7). In the lower branch, EAGLE first learns label embedding using Algorithm 1 by re-organizing the MIML dataset. Then, a soft-assignment mask is generated based on the inner product between the multi-instance inputs and the label embedding. The masked MI logit is an element-wise multiplication of the mask and the original MI logit. A max-pooling operation over instances on the masked MI logit gives the final logit for prediction, similar to the original deep MIML.

next iterate would improve. Notice that as the group size becomes larger, the signal becomes smaller and α is smaller in general.

6. Extensions

In the previous section, EAGLE is proposed for AL-XMC in the extreme label setting. In the non-extreme case, EAGLE naturally leads to a solution for the general MIML problems.

Deep-MIML Network. Feng and Zhou (2017) propose a general deep-MIML network: given a multi-instance input $\mathbf{X}$ with shape $g \times d$, the network first transforms it into a $g \times k \times l$ tensor $\mathbf{V}_1$ through a fully connected layer and a ReLU layer, where l is the label size and k is the additional dimension called 'concept size' (Feng & Zhou, 2017) to improve the model capacity. A max-pooling operation over all 'concepts' is taken on $\mathbf{V}_1$ to provide a $g \times l$ matrix $\mathbf{V}_2$ (which we call multi-instance logit). Finally, max-pooling over all instances is taken on $\mathbf{V}_2$ to give the final length-l logit prediction $\hat{\mathbf{Y}}$. This network can be summarized as:

$$\mathbf{X} \xrightarrow{\text{FC + ReLU}} \mathbf{V}_1 \xrightarrow[\text{(over concepts)}]{\text{max-pooling}} \mathbf{V}_2 \xrightarrow[\text{(over samples)}]{\text{max-pooling}} \hat{\mathbf{Y}} \quad (7)$$

A co-attention framework. Our idea can be directly applied to modify the deep-MIML network structure, as shown in Figure 2. The main idea is to add a soft-assignment mask to the original multi-instance logit, where this mask mimics the label assignment in EAGLE. After learning the label embedding $\mathbf{L} \in \mathbb{R}^{l\times d}$ from the dataset using Algorithm 1, the mask $\mathbf{M} \in \mathbb{R}^{g\times l}$ is calculated by $\mathbf{M} = g \cdot \texttt{Softmax}\left(\tau \mathbf{X}_{\mathcal{M}_j}\mathbf{L}^\top\right)$ where this softmax operation applies to each column in $\mathbf{M}$. As a result, $\mathbf{M}_{i,j}$ indicates the affinity between instance i and label j. Notice that τ controls the hardness of the assignment, and the special case of $\tau = 0$ corresponds to the standard deep-MIML framework. Interestingly, this mask can also be interpreted as an attention weight matrix, which is then multiplied with the multi-instance logit matrix $\mathbf{V}_2$. While there is other literature using attention for MIL (Ilse et al., 2018), none of the existing methods uses a robust calculation of the label embedding as the input to the attention. The proposed co-attention framework is easily interpretable since both labels and samples lie in the same representation space, with theoretical justifications we have shown in Section 5. Notice that the co-attention framework in Figure 2 can be trained end-to-end.

7. Experimental Results

In this section, we empirically verify the effectiveness of EAGLE from multiple standpoints. First, we run simulations to verify and explain the benefit of label assignment as analyzed in Theorem 1. Next, we run synthetic experiments on standard XMC datasets to understand the advantages of EAGLE under multiple aggregation rules. Lastly, for the natural extension of EAGLE in the non-extreme setting (as

Table 1: Statistics of 4 XMC datasets. 'sample size' column includes training & test set. The last column includes precisions with the clean datasets, which can be thought of as the *oracle* performance given an XMC dataset with aggregated labels.

Dataset	# feat.	# label	sample size	avg samples/label	avg labels/sample	std. precision (P@1/3/5)
EurLex-4K	5,000	3,993	15,539 / 3,809	25.73	5.31	82.71 / 69.42 / 58.14
Wiki-10K	101,938	30,938	14,146 / 6,616	8.52	18.64	84.31 / 72.57 / 63.39
AmazonCat-13K	203,882	13,330	1,186,239 / 306,782	448.57	5.04	93.03 / 79.16 / 64.52
Wiki-325K	1,617,899	325,056	1,778,351 / 587,084	17.46	3.19	66.04 / 43.63 / 33.05

Table 2: Comparing **Baseline**, **EAGLE-0** (EAGLE without label learning) and **EAGLE** on small/mid/large-size XMC datasets with aggregated labels. 'O' stands for oversized model (>5GB). **R-4/10** randomly selects $4/10$ samples in each group and observes their aggregated labels. **C** clusters samples based on hierarchical k-means. The cluster depth is determined based on sample size (8 for EurLex-4k, Wiki-10k and 16 for AmazonCat-13k and Wiki-325k).

		EurLex-4k			Wiki-10k			AmazonCat-13k			Wiki-325k		
		Baseline	EAGLE-0	EAGLE	Baseline	EAGLE-0	EAGLE	Baseline	EAGLE-0	EAGLE	Baseline	EAGLE-0	EAGLE
R-4	P1	76.58	80.16	**80.87**	73.07	78.28	**80.38**	80.62	78.83	**81.34**	34.09	**62.45**	60.60
	P3	61.36	63.49	**65.33**	60.84	63.81	**66.15**	65.81	67.45	**69.97**	33.34	**41.96**	40.23
	P5	49.50	51.01	**52.88**	53.31	54.66	**57.22**	54.06	54.80	**56.98**	26.05	**31.23**	29.80
R-10	P1	62.95	62.58	**67.56**	64.87	65.21	**68.89**	56.42	60.71	**63.47**	O	52.43	**54.81**
	P3	47.72	44.07	**48.03**	51.13	50.33	**53.60**	47.74	49.05	**51.59**	O	34.02	**35.80**
	P5	**37.59**	33.25	35.91	42.89	41.22	**44.68**	**40.74**	38.06	39.47	O	24.84	**26.18**
C	P1	17.94	39.11	**43.11**	16.34	39.13	**40.25**	**74.65**	56.41	56.19	O	45.03	**46.39**
	P3	16.27	28.02	**30.08**	16.08	30.70	**31.21**	**64.85**	48.32	48.07	O	27.98	**28.96**
	P5	14.83	22.46	**23.78**	15.96	25.69	**26.09**	**53.61**	40.01	39.93	O	20.53	**21.27**

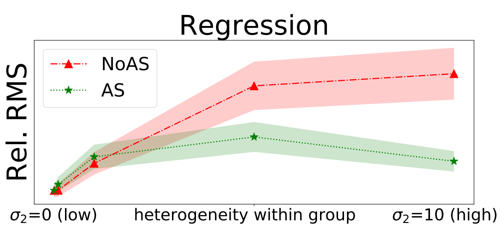


Figure 3: Regression task with aggregated outputs. The advantage of **AS** (estimator w. label assignment) over **NoAS** (estimator w.o. label assignment) matches with the result in Theorem 1. The y-axis is the root mean square (RMS) value normalized by RMS of the maximum likelihood estimator with known correspondences (lower is better).

mentioned in Section 6), we study multiple MIML tasks and show the benefit of EAGLE over standard MIML solution. We include details of the experimental settings and more comparison results in the Appendix.

7.1. Simulations

We design a toy regression task to better explain the performance of our approach from an empirical perspective. Our data generating process strictly follows the setting in Theorem 1. We set $\sigma_1 = \sigma_e = 1.0$ and vary σ_2 from 0.0 to 10.0, which corresponds to heterogeneity within group changes from low to high.

Results. In Figure 3, as the deviation within each sample group increases, **AS** performs much better, which is due to the $\sqrt{g}$ difference in the error rate between (4) and the first term in (5). On the other hand, **AS** may perform slightly worse than **NoAS** in the well-clustered setting, which is due to the second term in (5). See another toy classification task with similar observations in the Appendix.

7.2. Extreme Multi-label Experiments

We first verify our idea on 4 standard extreme classification tasks[3](1 small, 2 mid-size and 1 large), whose detailed statistics are shown in Table 1. For all tasks, the samples are grouped under different rules including: (i) random clustering: each group of samples are randomly selected; (ii) hierarchical clustering: samples are hierarchically clustered using k-means. Each sample in the original XMC dataset belongs to exactly one of the groups. As described in Section 4, EAGLE learns the label embeddings, and assigns every label in the group to one of the samples based on the embeddings. Then, we run Parabel and compare the final performance. Notice that it is possible to assign labels more cleverly, however, we focus on the quality of the label embedding learned through EAGLE hence we stick to this

[3] http://manikvarma.org/downloads/XC/XMLRepository.html

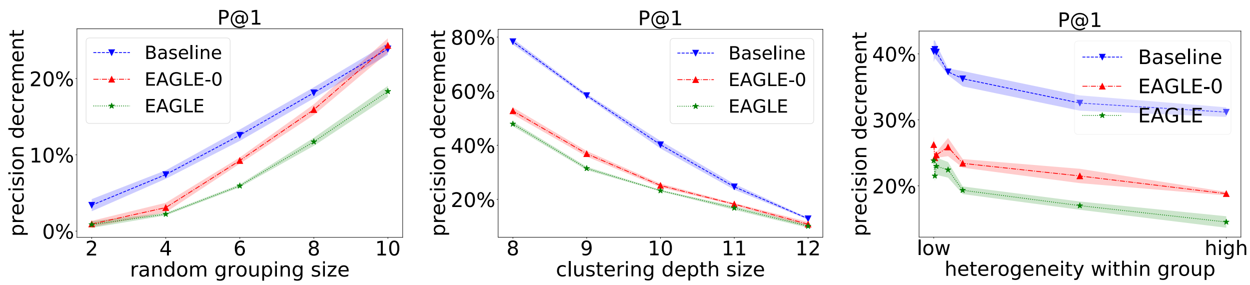


Figure 4: Ablation study: comparing Precision@1 between **Baseline** (no label assignment), **EAGLE-0** (EAGLE without label learning) and **EAGLE** (EAGLE with label learning). The y-axis calculates the percentage of precision decrement over the oracle performance (trained with known sample-label correspondences). We study different factors including **left**: group size in random grouping; **middle**: hierarchical clustering depth size; **right**: heterogeneity within group changes from low (hierarchical clustering) to high (random grouping).

simple assigning rule. We consider (i) **Baseline**: Parabel without label assignment; (ii) **EAGLE-0**: EAGLE without label learning ($T = 0$); and (iii) **EAGLE**: EAGLE with label learning ($T = 20$ by default).

Results. We report the performance using the standard Precision@1/3/5 metrics in Table 2. From the empirical results, we find that **EAGLE** performs better than **EAGLE-0** almost consistently, across all tasks and all grouping methods, and is much better than **Baseline** where we ignore such aggregation structure. **Baseline** performs much better only on AmazonCat-13k with hierarchical clustering, which is because of the low heterogeneity within each cluster, as theoretically explained by our Theorem 1. Notice that the precision on standard AmazonCat-13k achieves 93.04, which implies that the samples are easily separated. Furthermore, we also provide ablation study on EurLex-4K in Figure 4 to understand the influence of group size and clustering rule. We report the decrement percentage over a model trained with known correspondences. As a sanity check, as the size of the group gets smaller and annotation gets finer in Figure 4-(a) & (b), all methods have 0% decrement. More interestingly, in the other regime of more coarse annotations, (a) & (b) show that the benefit of EAGLE-0 varies when the clustering rule changes while the benefit of EAGLE is consistent. The consistency also exists when we change the heterogeneity within group by injecting noise to the feature representation when running the hierarchical clustering algorithm, as shown in Figure 4-(c).

7.3. MIML Experiments

First, we run a set of synthetic experiments on the standard MNIST & Fashion-MNIST image datasets, where we use the raw 784-dimension vector as the representation. Each 'sample' in our training set consists of g random digit/clothing images and the set of corresponding labels (we set $g = 4, 50$). We then test the accuracy on the standard test set. On the other hand, we collect the standard Yelp's customer review data from the web. Our goal is to predict the tags of a restaurant based on the customer reviews. We choose 10 labels with balanced positive samples and report the overall accuracy. Notice that each single review can be splitted into multiple sentences, as a result, we formulate it as an MIML problem similar to (Feng & Zhou, 2017). We retrieve the feature of each instance using InferSent[4], an off-the-shelf sentence embedding method. We randomly collect 20k reviews with 4 sentences for training, 10k reviews with single sentence for validation and 10k reviews with single sentence for testing. We report the top-1 precision.

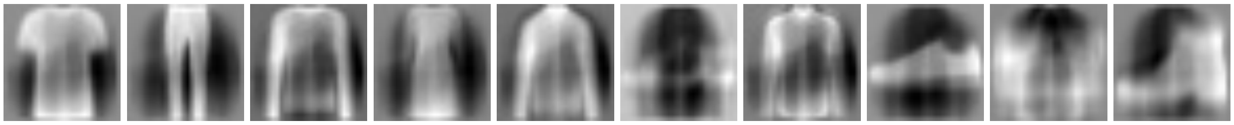
Table 3: Prediction accuracy on multiple MIML tasks.

group size	**MNIST** 4 / 50	**Fashion** 4 / 50	**Yelp** 4
Deep-MIML	94.70/33.33	84.70/19.00	40.69
EAGLE-0	**94.82**/36.10	84.89/27.62	45.82
EAGLE	**94.82/38.46**	**85.09/28.65**	**46.25**

For both the tasks, we use a two-layer feed-forward neural network as the base model, identical to the setting in (Feng & Zhou, 2017). We compare Deep-MIML with the extension of EAGLE-0 and EAGLE for MIML, as illustrated in Figure 2. We first do a hyper-parameter search for Deep-MIML to find the best learning rate and the ideal epoch number (the epoch number with best validation performance). Then we fix those hyper-parameters and use them for all algorithms.

Results. The performance of EAGLE in multiple MIML tasks is shown in Table 3. We see a 5.6% absolute im-

[4]https://github.com/facebookresearch/InferSent



T-shirt Trouser Pullover Dress Coat Sandal Shirt Sneaker Bag Boot

Figure 5: Visualization of learned label embeddings on Fashion-MNIST dataset.

provement over Deep-MIML on the Yelp dataset. There is consistent improvement for the image tasks as well. Notice that the improvement on the large group size is much significant than on the small group size. This is because the original Deep-MIML framework is able to handle the easy MIML tasks well, but is less effective for difficult tasks. Figure 5 visualizes the learned label embedding for Fashion-MNIST dataset. All these results corroborate the advantage of using label embeddings at the beginning of feed-forward neural networks in MIML.

8. Conclusions & Discussion

In this paper, we study XMC with aggregated labels, and propose the first efficient algorithm EAGLE that advances standard XMC methods in most settings. Our work leaves open several interesting issues to study in the future. First, while using positively labeled groups to learn label embedding, what would be the most efficient way to also learn/sample from negatively labeled groups? Second, is there a way to estimate the clustering quality and adjust the hyper-parameters accordingly? Moving forward, we believe the co-attention framework we proposed in Section 6 can help design deeper neural network architectures for MIML with better performance and interpretation.

References

Abid, A., Poon, A., and Zou, J. Linear regression with shuffled labels. *arXiv preprint arXiv:1705.01342*, 2017.

Agrawal, R., Gupta, A., Prabhu, Y., and Varma, M. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *Proceedings of the 22nd International Conference on World Wide Web*, pp. 13–24, 2013.

Babbar, R. and Schölkopf, B. Dismec: Distributed sparse machines for extreme multi-label classification. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 721–729, 2017.

Bamler, R. and Mandt, S. Extreme classification via adversarial softmax approximation. In *International Conference on Learning Representations*, 2020.

Chang, M.-W., Srikumar, V., Goldwasser, D., and Roth, D. Structured output learning with indirect supervision. In *International Conference on Machine Learning*, pp. 199–206, 2010.

Chang, W.-C., Yu, H.-F., Zhong, K., Yang, Y., and Dhillon, I. X-BERT: extreme multi-label text classification using bidirectional encoder representations from transformers. *arXiv preprint arXiv:1905.02331*, 2019.

Collier, O. and Dalalyan, A. S. Minimax rates in permutation estimation for feature matching. *The Journal of Machine Learning Research*, 17(1):162–192, 2016.

Denton, E., Weston, J., Paluri, M., Bourdev, L., and Fergus, R. User conditional hashtag prediction for images. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1731–1740, 2015.

Dietterich, T. G., Lathrop, R. H., and Lozano-Pérez, T. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

Feng, J. and Zhou, Z.-H. Deep MIML network. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., and Mikolov, T. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, pp. 2121–2129, 2013.

Guo, C., Mousavi, A., Wu, X., Holtmann-Rice, D. N., Kale, S., Reddi, S., and Kumar, S. Breaking the glass ceiling for embedding-based classifiers for large output spaces. In *Advances in Neural Information Processing Systems*, pp. 4944–4954, 2019.

Haghighatshoar, S. and Caire, G. Signal recovery from unlabeled samples. *IEEE Transactions on Signal Processing*, 66(5):1242–1257, 2017.

Hsu, D. J., Shi, K., and Sun, X. Linear regression without correspondence. In *Advances in Neural Information Processing Systems*, pp. 1531–1540, 2017.

Ilse, M., Tomczak, J., and Welling, M. Attention-based deep multiple instance learning. In *International Conference on Machine Learning*, pp. 2132–2141, 2018.

Jain, H., Prabhu, Y., and Varma, M. Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 935–944, 2016.

Jain, H., Balasubramanian, V., Chunduri, B., and Varma, M. Slice: Scalable linear extreme classifiers trained on 100 million labels for related searches. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pp. 528–536. ACM, 2019.

Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., and Teh, Y. W. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pp. 3744–3753, 2019.

Liu, J., Chang, W.-C., Wu, Y., and Yang, Y. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 115–124, 2017.

Liu, T. and Tao, D. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):447–461, 2015.

Maron, O. and Lozano-Pérez, T. A framework for multiple-instance learning. In *Advances in Neural Information Processing Systems*, pp. 570–576, 1998.

McCallum, A. Multi-label text classification with a mixture model trained by EM. In *AAAI Workshop on Text Learning*, pp. 1–7, 1999.

Medini, T. K. R., Huang, Q., Wang, Y., Mohan, V., and Shrivastava, A. Extreme classification in log memory using count-min sketch: A case study of Amazon search with 50M products. In *Advances in Neural Information Processing Systems*, pp. 13244–13254, 2019.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, pp. 1196–1204, 2013.

Pananjady, A., Wainwright, M. J., and Courtade, T. A. Denoising linear models with permuted data. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 446–450. IEEE, 2017a.

Pananjady, A., Wainwright, M. J., and Courtade, T. A. Linear regression with shuffled data: Statistical and computational limits of permutation recovery. *IEEE Transactions on Information Theory*, 64(5):3286–3300, 2017b.

Partalas, I., Kosmopoulos, A., Baskiotis, N., Artieres, T., Paliouras, G., Gaussier, E., Androutsopoulos, I., Amini, M.-R., and Galinari, P. LSHTC: A benchmark for large-scale text classification. *arXiv preprint arXiv:1503.08581*, 2015.

Prabhu, Y. and Varma, M. FastXML: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 263–272, 2014.

Prabhu, Y., Kag, A., Gopinath, S., Dahiya, K., Harsola, S., Agrawal, R., and Varma, M. Extreme multi-label learning with label features for warm-start tagging, ranking & recommendation. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 441–449. ACM, 2018a.

Prabhu, Y., Kag, A., Harsola, S., Agrawal, R., and Varma, M. Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In *Proceedings of the 2018 World Wide Web Conference*, pp. 993–1002. International World Wide Web Conferences Steering Committee, 2018b.

Shen, Y. and Sanghavi, S. Iterative least trimmed squares for mixed linear regression. In *Advances in Neural Information Processing Systems*, pp. 6076–6086, 2019a.

Shen, Y. and Sanghavi, S. Learning with bad training data via iterative trimmed loss minimization. In *International Conference on Machine Learning*, pp. 5739–5748, 2019b.

Si, S., Zhang, H., Keerthi, S. S., Mahajan, D., Dhillon, I. S., and Hsieh, C.-J. Gradient boosted decision trees for high dimensional sparse output. In *International Conference on Machine Learning*, pp. 3182–3190. JMLR. org, 2017.

Siblini, W., Kuntz, P., and Meyer, F. Craftml, an efficient clustering-based random forest for extreme multi-label learning. In *International Conference on Machine Learning*, pp. 4664–4673, 2018.

Wang, H. and Poon, H. Deep probabilistic logic: A unifying framework for indirect supervision. *arXiv preprint arXiv:1808.08485*, 2018.

Weston, J., Bengio, S., and Usunier, N. Wsabie: Scaling up to large vocabulary image annotation. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

Wydmuch, M., Jasinska, K., Kuznetsov, M., Busa-Fekete, R., and Dembczynski, K. A no-regret generalization of hierarchical softmax to extreme multi-label classification. In *Advances in Neural Information Processing Systems*, pp. 6355–6366, 2018.

Yen, I. E., Huang, X., Dai, W., Ravikumar, P., Dhillon, I., and Xing, E. Ppdsparse: A parallel primal-dual sparse method for extreme classification. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 545–553, 2017.

Yen, I. E.-H., Huang, X., Ravikumar, P., Zhong, K., and Dhillon, I. Pd-sparse: A primal and dual sparse approach to extreme multiclass and multilabel classification. In *International Conference on Machine Learning*, pp. 3069–3077, 2016.

You, R., Zhang, Z., Dai, S., and Zhu, S. HAXMLNet: Hierarchical attention network for extreme multi-label text classification. *arXiv preprint arXiv:1904.12578*, 2019.

Yu, H.-F., Jain, P., Kar, P., and Dhillon, I. Large-scale multi-label learning with missing labels. In *International Conference on Machine Learning*, pp. 593–601, 2014.

Zhang, M.-L. and Zhou, Z.-H. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2013.

Zhang, W., Yan, J., Wang, X., and Zha, H. Deep extreme multi-label learning. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pp. 100–107. ACM, 2018.

Zhang, Z.-L. and Zhang, M.-L. Multi-instance multi-label learning with application to scene classification. In *Advances in Neural Information Processing Systems*, pp. 1609–1616, 2007.

Zhou, Z.-H., Zhang, M.-L., Huang, S.-J., and Li, Y.-F. Multi-instance multi-label learning. *Artificial Intelligence*, 176 (1):2291–2320, 2012.