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:
• 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; Wydymuch 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; Haghifhatshoar & 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 \( \{X, Y\} \), where \( X \in \mathbb{R}^{n \times d} \) is the feature matrix for all \( n \) samples, \( Y \in \{0, 1\}^{n \times l} \) is the sample-to-label binary annotation matrix with label size \( l \) (if sample \( i \) is annotated by label \( k \) then \( Y_{i,k} = 1 \)). For the AL-
where the original annotation matrix is replaced by two bi-

XMC problem, however, such a clean annotation matrix is
corresponding index sets as well.

YMC approaches can be applied for AL-XMC
includes rows with index in set
for the aggregated samples. The goal is to use
Y captures how the samples are grouped, while
Y is, the weaker the annotation quality be-
classifier. Let $\bar{g} = \text{nanz}(Y^1)/m$ denote the average group size. In
general, the larger $\bar{g}$ is, the weaker the annotation quality be-
comes. For convenience, let $N', M, L$ be the set of samples,
intermediate nodes, and labels, respectively. Let $N_j, L_j$
be the set of samples, labels linked to intermediate node $j$
respectively, $\forall j \in M$, and $M_k$ be the set of nodes in $M$
connected to label $k \in L$. Let $x_j^T$ be the $i$-th row in $X$
for $i \in N_j$, $j \in M$, and $X_S$ be the submatrix of $X$ that
includes rows with index in set $S \subseteq N^2$.

**Deficiency of existing XMC approaches.** Existing XMC
approaches can be applied for AL-XMC by treating the
product $Y^1 Y^2$ directly as the annotation matrix, and learn-
ing a model using $\{X, Y^1 Y^2\}$. While using all labeling
information, this simple treatment ignores the possible in-
correct 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 perfor-
mance on standard XMC datasets based on a partitioned
label tree: It first calculates label representations $L = Y^T X$
that summarize the positive samples associated with every
label (step 1); Next, a balanced hierarchical partitioning of
the labels is learned using $L$ (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 $Y^1 Y^2$ as the annotation matrix
is given by:

$$L = (Y^1 Y^2)^T X. \quad (1)$$

Consider an example where there are $n$ samples and 2 la-

labels, with $X \in \mathbb{R}^{n \times d}$, $Y^1 \in \{0, 1\}^{n \times \frac{d}{2}}$, $Y^2 \in \{0, 1\}^{\frac{d}{2} \times 2}$
defined as follows:

$$X = 1_{\frac{d}{2}} \otimes \begin{bmatrix} x^T \ 0 \end{bmatrix}^T,$$

$$Y^1 = I_{\frac{d}{2}} \otimes 1_2, \ Y^2 = 1_{\frac{d}{2} \times 2},$$

where $\otimes$ is the Kronecker product, $1_d(1_{d_1} \times 1_{d_2})$ is an all-
ones vector(matrix) with dimension $d(1 \times 2d)$ and $I_d$ is the
identity matrix with size $d$. A pictorial explanation is shown
in Figure 1. The embedding calculated using the above
X, $Y^1$, $Y^2$ leads to $L = 0_{n \times 2}$ 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 $x$ as
label 1 and samples with feature close to $-x$ 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 $k \in L$ should be close
to the embedding of at least one of the samples in
$N_j$, $\forall j \in M_k$.

The closeness here can be any general characterization of
similarity, e.g., the standard cosine similarity. According
Algorithm 1 \textsc{Group\_Robust\_Label\_Repr (GRLR)}

\textbf{Inputs:} $\mathcal{M}_k, \{\mathcal{N}_j\}_{j \in \mathcal{M}_k}, \mathbf{X}$.
\textbf{Output:} Label embedding $\mathbf{e}_k$.
\textbf{Initialize:} Set $\mathbf{e}^0 \leftarrow \text{Proj} \left( \sum_{j \in \mathcal{M}_k} \sum_{i \in \mathcal{N}_j} \mathbf{x}_i \right)$.
\textbf{for} $t = 1, \ldots, T$ \textbf{do} /\* where $\text{Proj}(x) := x/\|x\|$ */
\hspace{1em} \textbf{for} $j \in \mathcal{M}_k$ \textbf{do}
\hspace{2em} \text{\textbf{\$\texttt{v}_{i,j}$}} \leftarrow (\mathbf{e}^{t-1} \cdot \mathbf{x}_i), \forall i \in \mathcal{N}_j.
\hspace{2em} \text{\textbf{\$\texttt{a}_{i,j}$}} \leftarrow 1 \{ \text{\textbf{\$\texttt{v}_{i,j}$}} = \max_{i' \in \mathcal{N}_j} \text{\textbf{\$\texttt{v}_{i',j}$}} \}, \forall i \in \mathcal{N}_j.
\hspace{1em} \textbf{end for}
\hspace{1em} \textbf{\textbf{\$\texttt{g}^t$}} \leftarrow \text{Proj} \left( \sum_{j \in \mathcal{M}_k} \sum_{i \in \mathcal{N}_j} \text{\textbf{\$\texttt{a}_{i,j}$}} \mathbf{x}_i \right).
\hspace{1em} \textbf{\textbf{\$\texttt{e}^t$}} \leftarrow \text{Proj} \left( \mathbf{e}^{t-1} + \lambda \cdot \textbf{\$\texttt{g}^t$} \right).
\textbf{end for}
\textbf{Return:} $\mathbf{e}^T$.

Algorithm 2 \textsc{EAGLE}

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

All label’s embedding is $O(n d \log l / l \cdot \bar{g}) = O(n d \bar{g} \log l)$. On the other hand, the time complexity for Parabel (one of the most efficient XMC solver) is $O(n d \log l)$ for step 1, $O(d \log l)$ for step 2 and $O(n d \log l)$ 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

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

where $\mathbf{E}^j$ is the noise matrix, and $\mathbf{I}^j$ is an unknown permutation matrix. For simplicity, we assume each group includes $g$ samples and the aggregation structure can be described by $\mathbf{Y}^1 = \mathbf{I}_m \otimes \mathbf{1}_g, \mathbf{Y}^2 = (\mathbf{I}_m \otimes \mathbf{1}_g^T) \cdot \mathbf{Z}$ 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 $\mathcal{N}(0, \sigma^2 \mathbf{I}_l)$, each sample feature is generated according to $x_i = x_j + d_i$, to this rule, in the previous toy example, the optimal label embedding for both labels is either $[x, -x]$ or $[-x, x]$, instead of $[0, 0]$. More formally, the label embedding for label $k \in \mathcal{L}$ is calculated based on the following:

$$\hat{\mathbf{e}}_k = \arg \max_{\mathbf{e}} \frac{1}{\sum_{j \in \mathcal{N}_j} (\mathbf{x}_i \cdot \mathbf{e})} \sum_{j \in \mathcal{N}_j} (\mathbf{x}_i \cdot \mathbf{e}) = \mathbf{e}.$$ (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 IEarning (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
for \( i \in N_j, j \in M \), where \( \tilde{x}_j \sim \mathcal{N}(0, \sigma_y^2 I_j) \) describes the center of each group, and \( d_i \sim \mathcal{N}(0, \sigma_x^2 I) \) 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:

\[
\hat{B}_{\text{bas}} = \text{LR}( \cup_{j \in M} \{ (1^T_j \mathbf{X}_j, 1^T_j \mathbf{Z}_j) \} )
\]

\[
\hat{B}_A = \text{LR}( \cup_{j \in M} \cup_{i \in N_j} \{ (x_i, z_i) \} )
\]

(3)

where \( \text{LR}( \{ (x_i, z_i) \}_{i \in [n]} ) = \left( \sum_{i \in [n]} x_i z_i^T \right)^{-1} \sum_{i \in [n]} x_i z_i^T \)

and \( i' = \arg \min_{i \in N_j} \| z_i - \hat{B}_{\text{bas}} x_i \| \).

Here, \( \hat{B}_{\text{bas}} \) corresponds to the baseline approach that learns a model without label assignment. On the other hand, \( \hat{B}_A \) corresponds to the estimator we learn after assigning each output in the group to the closest instance based on the residual norm using \( \hat{B}_{\text{bas}} \). We have the following result that describes the properties of the two estimators:

**Theorem 1.** Given the two estimators in (3), let \( R_1 = \| B_{\text{bas}} - B^* \|, \ R_2 = \| B_A - B^* \| \), \( \sigma_x = \sqrt{\sigma_1^2 + \sigma_2^2} \).

with \( n \geq c_0 p d \log^2 d \), the following holds with high probability (i.e., \( 1 - n^{-c} \)):

\[
R_1 \leq O \left( \frac{1}{p(\sigma_1^2 + \sigma_2^2)} \sigma_\epsilon \right),
\]

(4)

\[
R_2 \leq O \left( \frac{1}{p g^2 \sigma_2^2} \sigma_\epsilon + O \left( \frac{\sigma_2^2}{\sigma_2^2} + R_1^2 \sqrt{\frac{\sigma_2^2}{\sigma_2^2} + 1} \right) \right).
\]

(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 \( \Pi^I \)'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 \( \sigma_1 \gg \sigma_2 \), \( B_{\text{bas}} \) achieves the same rate as the optimal estimator, but when \( \sigma_1 \ll \sigma_2 \), the rate goes down from \( n^{-1/2} \) to \( \left( \frac{n}{g} \right)^{-1/2} \). This shows that \( B_{\text{bas}} \) is close to optimal when the clustering quality is high (within group deviation is small), on the other hand, \( B_{A} \) 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 \( k \in L \), there is some ground truth embedding \( e_k^* \), and each sample is associated with one of the labels. For sample \( i \) with ground truth label \( k \), its feature vector \( x_i \) can be decomposed as: \( x_i = e_k^* + \epsilon_i \). Without loss of generality, we only need to focus on the recovering of single label \( k \in L \). Further, for simplicity, let us assume that both \( x_i \) and \( e_k^* \) have unit norm measured in Euclidean space. We introduce the following definition that describes the property of the data:

**Definition 2.** Define \( \delta = \min_{k_i, k_2} \| e_{k_i} - e_{k_2} \| \) to be the minimum separation between each pair of ground truth label embeddings. Define \( f(\gamma) = \max_{S \subset M, |S| \leq \gamma} \| \sum_{i \in S} e_i \| \) to be the maximum influence of the noise, for \( \gamma \in [0, 1] \), and let \( f = f(1) \). Define \( q = \max_{k_1, k_2, \min(|M_{k_1} \cap M_{k_2}|)} \) 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:

\[
\langle e_k^*, \hat{e}_k \rangle \geq 1 - rf - (\sqrt{2r + 2}) f^2,
\]

(6)

where \( r = \left( \frac{1 - q - 2f}{2} \right) \). 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 \( \delta \) and a larger \( q \) both leads to a 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 \( e^t \). Let \( S_k^{\text{good}} \) be the set of groups where \( e^t \) is closest to the sample belongs to label \( k \). Denote \( |S_k^{\text{good}}|/|M_k| = \alpha_k \). The next step iterate given by Algorithm 1 has the following one-step property:

\[
\langle e_k^*, e^{t+1} \rangle \geq \alpha_k + (1 - \alpha_k) \left( \langle e_k^*, e^t \rangle - \| e_k^* - e^t \| \right) - f
\]

and a sufficient condition for contraction is \( \langle e_k^*, e^t \rangle \leq 1 - 2 \left( \frac{1 - f/2}{\alpha_k} - 1 \right)^2 \).

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 \( e^t \), then the distance from the selected incorrect sample to label \( k \)'s embedding can be controlled by the distance to \( e^t \) and the distance from \( e^t \) to \( e_k^* \), 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 \( \alpha_k \) and \( e^t \) (which requires more restrictive assumptions), but instead provide a sufficient condition to illustrate when the
next iterate would improve. Notice that as the group size becomes larger, the signal becomes smaller and $\alpha$ 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 $X$ with shape $g \times d$, the network first transforms it into a $g \times k \times l$ tensor $V_1$ 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 $V_1$ to provide a $g \times l$ matrix $V_2$ (which we call multi-instance logit). Finally, max-pooling over all instances is taken on $V_2$ to give the final length-$l$ logit prediction $Y$. This network can be summarized as:

\[
X \xrightarrow{\text{FC + ReLU}} V_1 \xrightarrow{\text{max-pooling}} V_2 \xrightarrow{\text{max-pooling}} Y \quad (7)
\]

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 $L \in \mathbb{R}^{d \times l}$ from the dataset using Algorithm 1, the mask $M \in \mathbb{R}^{g \times l}$ is calculated by $M = g \cdot \text{Softmax} \left( \tau X \Lambda, L^\top \right)$ where this softmax operation applies to each column in $M$. As a result, $M_{i,j}$ indicates the affinity between instance $i$ and label $j$. Notice that $\tau$ 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 $V_2$. 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
Extreme Multi-label Classification from Aggregated Labels

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.

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

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).

We first verify our idea on small/mid/large-size XMC datasets with similar observations in the Appendix.

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 \( \sigma_1 = \sigma_2 = 1.0 \) and vary \( \sigma_2 \) 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).

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. 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.

Table 3: Prediction accuracy on multiple MIML tasks.

<table>
<thead>
<tr>
<th>group size</th>
<th>MNIST</th>
<th>Fashion</th>
<th>Yelp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep-MIML</td>
<td>94.70/33.33</td>
<td>84.70/19.00</td>
<td>40.69</td>
</tr>
<tr>
<td>EAGLE-0</td>
<td>94.82/36.10</td>
<td>84.89/27.62</td>
<td>45.82</td>
</tr>
<tr>
<td>EAGLE</td>
<td>94.82/38.46</td>
<td>85.09/28.65</td>
<td>46.25</td>
</tr>
</tbody>
</table>

\(^4\)https://github.com/facebookresearch/InferSent
Extreme Multi-label Classification from Aggregated Labels

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


Extreme Multi-label Classification from Aggregated Labels


