Multi-Label Supervised Contrastive Learning

Pingyue Zhang, Mengyue Wu

MoE Key Lab of Artificial Intelligence, AI Institute
X-LANCE Lab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{williamzhangsjtu, mengyuewu}@sjtu.edu.cn

Abstract

Multi-label classification is an arduous problem given the complication in label correlation. Whilst sharing a common goal with contrastive learning in utilizing correlations for representation learning, how to better leverage label information remains challenging. Previous endeavors include extracting label-level presentations or mapping labels to an embedding space, overlooking the correlation between multiple labels. It exhibits a great ambiguity in determining positive samples with different extent of label overlap between samples and integrating such relations in loss functions. In our work, we propose Multi-Label Supervised Contrastive learning (MulSupCon) with a novel contrastive loss function to adjust weights based on how much overlap one sample shares with the anchor. By analyzing gradients, we explain why our method performs better under multi-label circumstances. To evaluate, we conduct direct classification and transfer learning on several multi-label datasets, including widely-used image datasets such as MS-COCO and NUS-WIDE. Validation indicates that our method outperforms the traditional multi-label classification method and shows a competitive performance when comparing to other existing approaches.

Introduction

Multi-label classification (MLC) scenario commonly exists in many machine learning domains such as computer vision (Wang et al. 2017), audio signal processing (Gemmeke et al. 2017), and natural language processing (Yang et al. 2018). Unlike single-label classification, label correlation is more complex and important in MLC. Previous endeavors on MLC have included latent space learning and label correlation modeling (Wang et al. 2017; Zhang et al. 2021; Ma et al. 2021).

As a matter of fact, MLC shares a similar goal with contrastive learning paradigm, where two instances exhibit a relation based on their content, semantics, or label correlation (Zhang et al. 2022). Therefore, extending contrastive learning to MLC is a natural progression and has shown promising results in recent works (Zhang et al. 2022; Malikowski and Mańdziuk 2022; Bai, Kong, and Gomes 2022; Zhou, Kang, and Ren 2022). However, these works mainly focus on the correlation between the sample and its corresponding label set, which may limit the generalization ability of their models due to the sample-label binding and the exclusive use of fully supervised learning. In MulCon (Dao et al. 2021), several label-level embeddings are extracted for each label in the dataset for a single sample. If two samples belong to the same class, their corresponding label-level embeddings, which correspond to the class, are pulled together. Further, contrastive learning can boost multi-label prediction model based on a Gaussian mixture variational autoencoder (C-GMVAE), indicating that contrastive loss can pull together correlated label embeddings and push away unrelated label embeddings in MLC (Bai, Kong, and Gomes 2022).

Different from prior investigations into MLC contrastive learning, our approach focuses primarily on capturing the correlation between samples. In this regard, we specifically analyze the relationships between two sets of labels, instead of directly utilizing the label information itself. By decoupling samples from their associated labels, our intention is to achieve improved performance and enhance generalization capabilities. This departure from binding samples to labels is driven by the aim to enhance overall model effectiveness. Taking inspiration from successful supervised contrastive learning (SupCon) (Khosla et al. 2020) technique, which demonstrates enhanced capability in exploiting label information within single-label classification tasks, we strive to extend these concepts to the realm of MLC. SupCon leverages label information and broadens the concept of positive samples from the anchor to all samples sharing the same label. This approach consistently outperforms traditional cross-entropy loss and shows robustness against data corruptions. However, in the context of MLC, determining positive samples for an anchor proves intricate and ambiguous due to the presence of multiple labels. This complexity stands in contrast to single-label classification, where positive samples can be unambiguously identified by matching their label to that of the anchor.

An intuitive question would be, should we consider one sample as positive when its label set partially overlaps with or exactly matches the anchor’s? We take both situations into consideration and properly define them as “ANY” for partially overlapping, and “ALL” for exactly matching. We then show the possible drawbacks of these two meth-
ods and propose a new multi-label supervised contrastive (MulSupCon) loss function to define positive samples in a finer way, based on the overlap proportion (i.e., 2 out of 3 labels are the same) between one instance and the anchor. By analyzing the gradients, we indicate (i) the optimizing process can be viewed as pushing the anchor’s embedding to the direction calculated by averaging embeddings of each of its classes, where the class embedding is represented by averaging the embedding of samples which belong to it; (ii) our loss function weighs each sample according to how many labels are shared between the anchor and the designated sample. Note that our approach yields a pretrained model. We add an extra linear classifier to tune the model using BCE. Our main contributions are:

1. In this paper, we present a novel approach for incorporating correlations between samples in Multi-Label Classification (MLC) by leveraging the principles of supervised contrastive learning. Our method introduces a novel contrastive loss function, termed “MulSupCon”, which effectively extends the single-label supervised contrastive learning to the multi-label context.

2. We evaluate the performance of MulSupCon by comparing it with other state-of-the-art MLC methods, the commonly adopted MLC loss BCE, and two alternative SupCon variants, on a series of datasets. Our findings reveal a notable enhancement in performance when leveraging MulSupCon.

3. Upon transferring the models trained using MulSupCon and BCE methods respectively, to downstream tasks, our approach showcases superior generalization capabilities.

**Methods**

For a batch of data \( B = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(B)}, y^{(B)})\} \), where \( B \) is the mini-batch size and \( y^{(i)} \) is the multi-label of sample \( i \), where \( y_{j}^{(i)} \) means the \( j \)-th label of sample \( i \). The main challenge of applying supervised contrastive learning to multi-label classification is that for the anchor sample \( i \) it is difficult to determine its corresponding positive samples. We will first introduce the framework of our contrastive learning and give several notations.

**Framework** The general framework is similar to the widely-used MoCo (He et al. 2020) framework. For sample \( i \), we use \( z_q^{(i)} \) and \( z_k^{(i)} \) to denote the \( \ell_2 \)-normalized output of the query model and key model, where the key model is momentum updated. We also use a queue \( Q \) to contain \( z_k \) from previous batches, as mentioned in MoCo.

**Multi-Label Supervised Contrastive Loss** As illustrated in Figure 1, two preliminary ideas to find positive samples include:

1. **ALL**: only those with exactly the same label class \( y \) are considered positive
2. **ANY**: samples with any class overlapping with the anchor \( y^{(i)} \) are positive

\[
\nabla \approx \frac{z^{(1)} + z^{(6)}}{2}
\]

**Figure 1**: Illustration for ALL and ANY and method. Each row represents one sample’s label, where the first row is the anchor and the following ones are samples in \( A \), here \( |A| = 6 \). Each sample’s label is denoted in a one-hot form where the green circle means 1. The row with circles plotted with dotted line means that the corresponding sample is in the negative set \( N \), otherwise the sample is in the positive set \( P \).

**Methods**

The general framework is similar to the widely-used MoCo (He et al. 2020) framework. For sample \( i \), we use \( z_q^{(i)} \) and \( z_k^{(i)} \) to denote the \( \ell_2 \)-normalized output of the query model and key model, where the key model is momentum updated. We also use a queue \( Q \) to contain \( z_k \) from previous batches, as mentioned in MoCo.

**Multi-Label Supervised Contrastive Loss** As illustrated in Figure 1, two preliminary ideas to find positive samples include:

1. **ALL**: only those with exactly the same label class \( y \) are considered positive
2. **ANY**: samples with any class overlapping with the anchor \( y^{(i)} \) are positive

\[
\nabla \approx \frac{z^{(1)} + z^{(6)}}{2}
\]

**Figure 1**: Illustration for ALL and ANY and method. Each row represents one sample’s label, where the first row is the anchor and the following ones are samples in \( A \), here \( |A| = 6 \). Each sample’s label is denoted in a one-hot form where the green circle means 1. The row with circles plotted with dotted line means that the corresponding sample is in the negative set \( N \), otherwise the sample is in the positive set \( P \).

Correspondingly, the positive sets for these two illustrations are

- **ALL**: \( \mathcal{P}^{(i)} = \{m | y^{(m)} = y^{(i)} \} \)
- **ANY**: \( \mathcal{P}^{(i)} = \{m | y^{(m)} \cap y^{(i)} \neq \emptyset \} \)

for ALL and

\[
\mathcal{P}^{(i)} = \{m | y^{(m)} \cap y^{(i)} \neq \emptyset \}
\]

for ANY.

We use \( \mathcal{A}^{(i)} \) to denote indices of all samples involved in calculating contrastive loss (from \( B \) and \( Q \)), \( \mathcal{A}^{(i)} = \mathcal{A}^{(j)}, \forall i, j \in \{1, 2, \cdots, B\} \), and \( |\mathcal{A}^{(i)}| = |B| + |Q| \). Negative set \( \mathcal{N}^{(i)} = \mathcal{A}^{(i)} - \mathcal{P}^{(i)} \).

Following SupCon (Khosla et al. 2020) loss:

\[
\mathcal{L}^{(i)}_{\text{supcon}} = -\frac{1}{|\mathcal{P}^{(i)}|} \sum_{p \in \mathcal{P}^{(i)}} \log \frac{e^{s_p^{(i)} / \tau}}{\sum_{a \in \mathcal{A}^{(i)}} e^{s_a^{(i)} / \tau}}
\]

where \( s_p^{(i)} = \frac{z_q^{(i)} \cdot z_k^{(p)}}{\tau} \). By taking gradient of \( \mathcal{L}^{(i)}_{\text{supcon}} \) with respect to \( s_p^{(i)}, s_n^{(i)} \), \( p \in \mathcal{P}^{(i)}, n \in \mathcal{N}^{(i)} \) we get:

\[
\nabla_{s_p^{(i)}} \mathcal{L}^{(i)}_{\text{supcon}} = \frac{1}{\tau} \left( \frac{1 - e^{s_p^{(i)} / \tau}}{|\mathcal{P}^{(i)}| + C} \right) \cdot \nabla s_p^{(i)} \mathcal{L}^{(i)}_{\text{supcon}} = \frac{1}{\tau} \cdot \frac{e^{s_p^{(i)} / \tau}}{C}
\]

where \( C = \sum_{a \in \mathcal{A}^{(i)}} e^{s_a^{(i)} / \tau} \).

By taking gradient of \( \mathcal{L}^{(i)}_{\text{supcon}} \) with respect to \( z_q^{(i)} \) we get:

\[
\nabla_{z_q^{(i)}} \mathcal{L}^{(i)}_{\text{supcon}} = \sum_{p \in \mathcal{P}^{(i)}} \nabla_{s_p^{(i)}} \mathcal{L}^{(i)}_{\text{supcon}} \cdot z_k^{(p)}
\]

\[
+ \sum_{n \in \mathcal{N}^{(i)}} \nabla_{s_n^{(i)}} \mathcal{L}^{(i)}_{\text{supcon}} \cdot z_k^{(n)}
\]

\[
= \sum_{p \in \mathcal{P}^{(i)}} \frac{1}{\tau} \left( \frac{1 - e^{s_p^{(i)} / \tau}}{|\mathcal{P}^{(i)}| + C} \right) \cdot z_k^{(p)}
\]

\[
+ \sum_{n \in \mathcal{N}^{(i)}} \frac{1}{\tau} \cdot \frac{e^{s_n^{(i)} / \tau}}{C} \cdot z_k^{(n)}
\]
As a result,
\[
\nabla_{\tilde{z}} \mathcal{L}_{\text{supcon}}^{(i)} = \tilde{z} + \hat{z}
\]
\[
\tilde{z} = \frac{-1}{\tau} \frac{1}{|\mathcal{P}(i)|} \sum_{p \in \mathcal{P}(i)} z_k^{(p)}
\]
\[
\hat{z} = \sum_{a \in \mathcal{A}(i)} \frac{1}{\tau} \frac{e^{s(a)}/\tau}{C} \cdot z_k^{(a)},
\]
which means the optimization strategy is pushing \(z_k^{(i)}\) towards the direction of the mean value of all \(z_k^{(p)}\) (scaled by \(\frac{1}{\tau}\)), where \(\hat{z}\) is the weighted average of all embeddings which prevents the collapse of representation.

As a result, for ALL the optimization direction is the mean representation of all samples with exactly the same label. The drawback is that if most samples have some common classes, the averaging process will highlight the information of the common classes and give less weight to others. As illustrated in Figure 1, all positive samples belong to the third class while only three samples belong to the first class. Therefore, the mean \(\nabla\) mainly contains the information of the third class while the first class’s information gains less weight.

Take one of datasets used in our work MS-COCO (Lin et al. 2014) as an example, if the anchor belongs to person, bicycle, car class, then there are less than 800 (1%) samples among over 80k samples which belong to exactly the same class set. Furthermore, there are about 45k, 2.3k, and 8.6k samples belong to person, bicycle, car respectively. The loss function we propose treats sample \(i\) as a separate sample for each class \(y_j^{(i)}\) it belongs to.

For each \(y_j^{(i)} \in y^{(i)}\), we construct a separate positive set:
\[
\mathcal{P}_j = \{m|y_m^{(i)}, y_j^{(i)} \in y^{(m)}\}.
\]

The proposed MulSupCon loss is:
\[
\mathcal{L}^{(i)} = \sum_{y_j^{(i)} \in y^{(i)}} \frac{-1}{|\mathcal{P}_j^{(i)}|} \sum_{p \in \mathcal{P}_j^{(i)}} \log \frac{e^{s_k^{(i)}}/\tau}{\sum_{a \in \mathcal{A}(i)} e^{s(a)}/\tau},
\]
and we illustrate it in Figure 2. The loss function of one batch is:
\[
\mathcal{L} = \frac{1}{|y^{(i)}|} \sum_{i} \mathcal{L}^{(i)}.
\]

Our loss function is a generalized version of SupCon loss in Equation (2), where it reduces to SupCon loss under the single-label circumstance, that is, \(|y^{(i)}| = 1\).

Note that we do not use \(\frac{1}{|y^{(i)}|}\) to weigh each sample as shown in Equation (6):
\[
\frac{1}{|y^{(i)}|} \sum_{y_j^{(i)} \in y^{(i)}} \frac{-1}{|\mathcal{P}_j^{(i)}|} \sum_{p \in \mathcal{P}_j^{(i)}} \log \frac{e^{s_k^{(i)}}/\tau}{\sum_{a \in \mathcal{A}(i)} e^{s(a)}/\tau},
\]

and we show the effect of \(\frac{1}{|y^{(i)}|}\) in the ablation section.

By taking gradient of \(\mathcal{L}^{(i)}\) with respect to \(z_k^{(i)}\) we get:
\[
\nabla_{z_k^{(i)}} \mathcal{L}^{(i)} = \sum_{y_j^{(i)} \in y^{(i)}} \frac{-1}{|\mathcal{P}_j^{(i)}|} \sum_{p \in \mathcal{P}_j^{(i)}} z_k^{(p)}
\]
\[
+ |y^{(i)}| \sum_{a \in \mathcal{A}(i)} \frac{1}{\tau} \frac{e^{s(a)}/\tau}{C} \cdot z_k^{(a)}
\]
\[
= \sum_{y_j^{(i)} \in y^{(i)}} \z_j + |y^{(i)}| \cdot \hat{z},
\]
where
\[
\z_j = \frac{-1}{\tau} \frac{1}{|\mathcal{P}_j^{(i)}|} \sum_{p \in \mathcal{P}_j^{(i)}} z_k^{(p)}
\]
is the mean value of \(z\) from samples which belong to class \(y_j^{(i)}\). If we consider \(\z_j\) as the “class” representation, then the optimization direction of our loss function is the sum value of all its “class” representations with an extra mean negative vector \(\hat{z}\).

From another perspective, we pick a sample \(t\) and denote \(y = y_t^{(i)} \cap y^{(i)}\), then following Equation (7), \(z_k^{(t)}\) will be weighted by
\[
\frac{-1}{\tau} \sum_{y_j^{(i)} \in y^{(i)}} \frac{1}{|\mathcal{P}_j^{(i)}|},
\]
which means more labels sample \(t\) intersects with the anchor, \(z_k^{(t)}\) will gain more weight.

**Experiments**

To assess our approach, we pretrain the model with the MulSupCon loss on various datasets. Subsequently, we employ the pretrained model, excluding its final non-linear projection head \(g\). By default, we introduce a linear layer atop the pretrained model and conduct MLC using the Binary Cross Entropy (BCE) Loss.
<table>
<thead>
<tr>
<th>Dataset</th>
<th># Samples</th>
<th># Labels</th>
<th>Mean L/S</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS-COCO</td>
<td>82081/40137</td>
<td>80</td>
<td>2.93/2.90</td>
</tr>
<tr>
<td>NUS-WIDE</td>
<td>15000/59347</td>
<td>81</td>
<td>2.41/2.40</td>
</tr>
<tr>
<td>Objects365</td>
<td>515233/80000</td>
<td>365</td>
<td>6.48/7.17</td>
</tr>
<tr>
<td>MIRFLICKR</td>
<td>19664/4917</td>
<td>24</td>
<td>3.78/3.76</td>
</tr>
<tr>
<td>PASCAL</td>
<td>5011/4952</td>
<td>20</td>
<td>1.46/1.42</td>
</tr>
<tr>
<td>Bookmarks</td>
<td>60000/27856</td>
<td>208</td>
<td>2.03/2.03</td>
</tr>
<tr>
<td>Mediamill</td>
<td>29804/12373</td>
<td>101</td>
<td>4.54/4.60</td>
</tr>
<tr>
<td>Nus-vec</td>
<td>125449/83898</td>
<td>81</td>
<td>2.40/2.41</td>
</tr>
<tr>
<td>Delicious</td>
<td>12886/3181</td>
<td>983</td>
<td>19.07/18.94</td>
</tr>
<tr>
<td>Scene</td>
<td>1210/1195</td>
<td>7</td>
<td>1.06/1.09</td>
</tr>
<tr>
<td>Yeast</td>
<td>1500/917</td>
<td>14</td>
<td>4.23/4.25</td>
</tr>
</tbody>
</table>

Table 1: Image and Vector dataset statistics. Mean L/S represents mean number of labels per sample. We present the statistics for the training and test sets separately, using a slash to divide them.

Data

We evaluate our method by measuring the multi-label classification performance on several image datasets, including MS-COCO (Lin et al. 2014), NUS-WIDE (Chua et al. 2009), Objects365 (Shao et al. 2019), MIRFLICKR (Huiskes and Lew 2008), and PASCAL-VOC (VOC2007) (Everingham et al. 2007). The dataset statistics are shown in Table 1. Note that for Objects365 we randomly pick a subset for the whole dataset contains tremendous number of images. We also validate on several vector multi-label datasets from Mulan\(^1\) where each sample is pre-processed and released in a vector form.

Metrics

We use six commonly-used metrics to evaluate our method and compare with baseline methods. Suppose that \(y\) is the ground truth label and \(\hat{y}\) is the predicted label (all in one-hot form), we use (1) the mean value of average precision (mAP), (2) the precision of the top-1 predictions (precision@1), (3) macro-F1, (4) micro-F1, (5) Hamming Accuracy \(\hat{y}^T L \hat{y}\), \(L\) is number of class and (6) example-F1 \(2\hat{y}_i \hat{y} \sum_{j=1}^{L} \hat{y}_j / \sum_{j=1}^{L} \hat{y}_j + \sum_{j=1}^{L} \hat{y}_j \).

Settings

For image datasets we use a common encoder architecture: ResNet-50. During the preprocessing process before training, all images are first resized to \(224 \times 224\) before training for easier storage and speedy training. For vector datasets, we use a simple multi-layer perceptron (MLP) which contains three layers of linear, ReLU activation function, and dropout. Note that we train both backbone encoders from scratch without using pretrained parameters.

For image datasets, during the pretraining stage, we train the model for 400 epochs using SGD optimizer with an initial learning rate of 0.1 and a cosine learning rate scheduler. A batch size of 64 is utilized for all datasets except for the Objects365 dataset, where a batch size of 128 is employed. For vector datasets, the model is trained for 150 epochs using the Adam optimizer with an initial learning rate of 0.0004, coupled with a cosine learning rate scheduler. A batch size of 256 is utilized, except for the yeast and scene datasets, where a batch size of 32 is used.

For two kinds of datasets, we all use data augmentation during pretraining because we find out that the performance will drop when the training period is long without a hard data augmentation. During pretraining, for image datasets, we first randomly crop the image and resize it to \(224 \times 224\); then, we apply color jittering as well as randomly converting the image to greyscale. And for vector datasets, we randomly mask 50% input elements.

After the pretraining stage, for image datasets, we both linearly probe and finetune the model to evaluate the performance. To linearly probe the model, we freeze the encoder and use the Adam optimizer with an initial learning rate of 0.01. And to finetune the model, we train the whole model use the Adam optimizer with an initial learning rate of \(1 \times 10^{-4}\). And for vector datasets, we finetune the model and use a learning rate of \(4 \times 10^{-4}\) for the linear layer and \(4 \times 10^{-5}\) for the backbone. For all datasets, we reduce the learning rate by a factor of 0.1 when it plateaus.

As one of the comparison methods, we train the model from scratch using BCE loss for image datasets. We employ identical settings including the optimizer, scheduler, number of epochs, augmentation strategy, and batch size as described in the pretraining process above.

Results

Following previous tradition of evaluating pretraining model like SimCLR (Chen et al. 2020) and SupCon (Khosla et al. 2020), we mainly evaluate our method on two aspects: linear probing/finetuning and transfer learning to downstream tasks, compared with existing state-of-the-art multi-label algorithms, commonly adopted BCE loss for multi-label classification, and the aforementioned ALL and ANY under supervised-contrastive learning paradigm.

Methods of comparison

We present a comprehensive comparative analysis of our proposed method against several established MLC approaches, as shown in Tables 2 to 4. Specifically, the methods under consideration are as follows:

- LaMP (Lanchantin, Sekhon, and Qi 2020): LaMP employs neural message passing techniques to effectively model the joint prediction of multiple labels.
- MPVAE (Bai, Kong, and Gomes 2020): MPVAE adopts a variational autoencoder framework to learn and align probabilistic embedding spaces for both labels and features.
- ASL (Ridnik et al. 2021): ASL introduces an innovative asymmetric loss function that operates differently on positive and negative samples, contributing to improved performance.
- RBCC (Gerych et al. 2021): RBCC focuses on learning a Bayesian network of class dependencies.
- C-GMVAE (Bai, Kong, and Gomes 2022): C-GMVAE combines the strengths of Gaussian mixture variational autoencoders and contrastive learning.

Comparison with existing approaches

We fine-tune the pretrained model on the same dataset. The results presented

\(^1\)http://mulan.sourceforge.net/datasets-mlc.html
Table 2: Comparison with existing approaches. The example-F1 and micro-F1 scores are shown, † means we run the official C-GMV AE codes with our data splits.

<table>
<thead>
<tr>
<th>Metric</th>
<th>media</th>
<th>yeast</th>
<th>scene</th>
<th>bmks</th>
<th>deli</th>
<th>nus</th>
<th>media</th>
<th>yeast</th>
<th>scene</th>
<th>bmks</th>
<th>deli</th>
<th>nus</th>
</tr>
</thead>
<tbody>
<tr>
<td>LaMP</td>
<td>0.624</td>
<td>0.728</td>
<td>0.389</td>
<td>0.372</td>
<td>0.376</td>
<td></td>
<td>0.641</td>
<td>0.716</td>
<td>0.373</td>
<td>0.386</td>
<td>0.472</td>
<td></td>
</tr>
<tr>
<td>MPVAE</td>
<td>0.648</td>
<td>0.751</td>
<td>0.382</td>
<td>0.373</td>
<td>0.468</td>
<td></td>
<td>0.655</td>
<td>0.742</td>
<td>0.375</td>
<td>0.393</td>
<td>0.492</td>
<td></td>
</tr>
<tr>
<td>ASL</td>
<td>0.613</td>
<td>0.770</td>
<td>0.373</td>
<td>0.359</td>
<td>0.468</td>
<td></td>
<td>0.637</td>
<td>0.753</td>
<td>0.354</td>
<td>0.387</td>
<td>0.495</td>
<td></td>
</tr>
<tr>
<td>RBCC</td>
<td>0.605</td>
<td>0.758</td>
<td></td>
<td>-</td>
<td>0.466</td>
<td></td>
<td>0.623</td>
<td>0.749</td>
<td></td>
<td>-</td>
<td>0.490</td>
<td></td>
</tr>
<tr>
<td>C-GMV AE</td>
<td>0.623</td>
<td></td>
<td></td>
<td>-</td>
<td>0.481</td>
<td></td>
<td>0.626</td>
<td></td>
<td></td>
<td>0.481</td>
<td>0.510</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>0.627</td>
<td>0.659</td>
<td>0.787</td>
<td>0.394</td>
<td>0.386</td>
<td>0.484</td>
<td>0.630</td>
<td>0.667</td>
<td>0.773</td>
<td>0.386</td>
<td>0.399</td>
<td>0.511</td>
</tr>
</tbody>
</table>

Table 3: An extension to Table 2, where the macro-F1 and hamming accuracy scores are shown.

<table>
<thead>
<tr>
<th>Metric</th>
<th>macro-F1</th>
<th>Hamming Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LaMP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MPVAE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RBCC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C-GMV AE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Comparison of results on embeddings from image datasets extracted by Imagenet-pretrained ResNet-50.
Table 5: Linear probe results of ALL, ANY, and MulSupCon (ours) method on MS-COCO dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>mAP</th>
<th>HA</th>
<th>example-F1</th>
<th>Macro-F1</th>
<th>Micro-F1</th>
<th>precision@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>BCE</td>
<td>0.694</td>
<td>0.981</td>
<td>0.724</td>
<td>0.654</td>
<td>0.706</td>
<td>0.921</td>
</tr>
<tr>
<td></td>
<td>linear</td>
<td>0.672</td>
<td>0.980</td>
<td>0.700</td>
<td>0.636</td>
<td>0.688</td>
<td>0.916</td>
</tr>
<tr>
<td></td>
<td>finetune</td>
<td>0.708</td>
<td>0.982</td>
<td>0.736</td>
<td>0.662</td>
<td>0.714</td>
<td>0.930</td>
</tr>
<tr>
<td>ANY</td>
<td>BCE</td>
<td>0.524</td>
<td>0.984</td>
<td>0.707</td>
<td>0.505</td>
<td>0.722</td>
<td>0.821</td>
</tr>
<tr>
<td></td>
<td>linear</td>
<td>0.552</td>
<td>0.984</td>
<td>0.701</td>
<td>0.553</td>
<td>0.721</td>
<td>0.822</td>
</tr>
<tr>
<td></td>
<td>finetune</td>
<td>0.566</td>
<td>0.984</td>
<td>0.714</td>
<td>0.557</td>
<td>0.728</td>
<td>0.825</td>
</tr>
<tr>
<td>Ours</td>
<td>BCE</td>
<td>0.180</td>
<td>0.983</td>
<td>0.361</td>
<td>0.133</td>
<td>0.432</td>
<td>0.716</td>
</tr>
<tr>
<td></td>
<td>linear</td>
<td>0.297</td>
<td>0.984</td>
<td>0.441</td>
<td>0.288</td>
<td>0.479</td>
<td>0.784</td>
</tr>
<tr>
<td></td>
<td>finetune</td>
<td>0.333</td>
<td>0.984</td>
<td>0.467</td>
<td>0.311</td>
<td>0.509</td>
<td>0.802</td>
</tr>
<tr>
<td>NUS-WIDE</td>
<td>BCE</td>
<td>0.673</td>
<td>0.962</td>
<td>0.684</td>
<td>0.628</td>
<td>0.692</td>
<td>0.769</td>
</tr>
<tr>
<td></td>
<td>linear</td>
<td>0.694</td>
<td>0.962</td>
<td>0.680</td>
<td>0.644</td>
<td>0.697</td>
<td>0.778</td>
</tr>
<tr>
<td></td>
<td>finetune</td>
<td>0.726</td>
<td>0.965</td>
<td>0.716</td>
<td>0.670</td>
<td>0.726</td>
<td>0.802</td>
</tr>
</tbody>
</table>

Table 6: Comparison of linear probing and finetuning results with BCE on 5 image datasets

(a) Model pretrained on MS-COCO

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>mAP</th>
<th>HA</th>
<th>example-F1</th>
<th>Macro-F1</th>
<th>Micro-F1</th>
<th>precision@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASCAL</td>
<td>BCE</td>
<td>0.865</td>
<td>0.979</td>
<td>0.838</td>
<td>0.809</td>
<td>0.842</td>
<td>0.916</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>0.853</td>
<td>0.977</td>
<td>0.818</td>
<td>0.792</td>
<td>0.824</td>
<td>0.907</td>
</tr>
<tr>
<td>MIRFLICKR</td>
<td>BCE</td>
<td>0.656</td>
<td>0.911</td>
<td>0.673</td>
<td>0.613</td>
<td>0.700</td>
<td>0.873</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>0.722</td>
<td>0.923</td>
<td>0.716</td>
<td>0.668</td>
<td>0.741</td>
<td>0.899</td>
</tr>
</tbody>
</table>

(b) Model pretrained on NUS-WIDE

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>mAP</th>
<th>HA</th>
<th>example-F1</th>
<th>Macro-F1</th>
<th>Micro-F1</th>
<th>precision@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>PASCAL</td>
<td>BCE</td>
<td>0.661</td>
<td>0.964</td>
<td>0.713</td>
<td>0.624</td>
<td>0.713</td>
<td>0.801</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>0.750</td>
<td>0.969</td>
<td>0.749</td>
<td>0.693</td>
<td>0.760</td>
<td>0.853</td>
</tr>
<tr>
<td>MIRFLICKR</td>
<td>BCE</td>
<td>0.716</td>
<td>0.922</td>
<td>0.716</td>
<td>0.662</td>
<td>0.740</td>
<td>0.897</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>0.750</td>
<td>0.927</td>
<td>0.736</td>
<td>0.691</td>
<td>0.758</td>
<td>0.906</td>
</tr>
</tbody>
</table>

Table 7: Transfer to PASCAL and MIRFLICKR datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Weight</th>
<th>mAP</th>
<th>HA</th>
<th>example-F1</th>
<th>Macro-F1</th>
<th>Micro-F1</th>
<th>precision@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS-COCO</td>
<td>Yes</td>
<td>0.663</td>
<td>0.980</td>
<td>0.697</td>
<td>0.630</td>
<td>0.683</td>
<td>0.914</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>0.672</td>
<td>0.980</td>
<td>0.700</td>
<td>0.636</td>
<td>0.688</td>
<td>0.916</td>
</tr>
<tr>
<td>PASCAL</td>
<td>Yes</td>
<td>0.693</td>
<td>0.962</td>
<td>0.676</td>
<td>0.640</td>
<td>0.695</td>
<td>0.772</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>0.694</td>
<td>0.962</td>
<td>0.680</td>
<td>0.644</td>
<td>0.697</td>
<td>0.778</td>
</tr>
<tr>
<td>MIRFLICKR</td>
<td>Yes</td>
<td>0.689</td>
<td>0.919</td>
<td>0.703</td>
<td>0.634</td>
<td>0.725</td>
<td>0.884</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>0.694</td>
<td>0.920</td>
<td>0.705</td>
<td>0.637</td>
<td>0.727</td>
<td>0.888</td>
</tr>
</tbody>
</table>

Table 8: Ablation study results

and NUS-WIDE on two downstream datasets: PASCAL and MIRFLICKR. We use an Adam optimizer with an initial learning rate of 0.04. We freeze the models and linearly probe them on other datasets. Note that in Table 6 BCE method outperforms the performance of linear probing our method on MS-COCO dataset. However, the outcomes outlined in Table 7(a) establish our method’s pronounced superiority over BCE on MIRFLICKR dataset. Despite the fact that directly transferring the model pretrained on the MS-COCO dataset does not yield results as favorable as those achieved by BCE on PASCAL dataset, we observe that the performance surpasses that of BCE when the pre-trained model is first finetuned on MS-COCO and then transferred, achieving an mAP of 0.877. Results in Table 7(b) show that our model consistently outperforms BCE, demonstrating MulSupCon’s stronger generalization ability.
Ablation Study

To further evaluate the MulSupCon loss function, we conducted ablation study.

Loss function We explore an alternative form of the loss function, which incorporates an additional weight \( \frac{1}{|y_{ij}|} \) for each sample, as illustrated in Equation (6). In this part, we conduct experiments on the MS-COCO, PASCAL, and MIRFLICKR datasets to compare the performance difference between using the loss function with or without the weight. The results presented in Table 8 indicate that not incorporating the weight leads to better performance.

Related Work

Multi-label Classification A simple strategy to solve MLC problem is to treat one sample’s labels as several independent labels, which is also the idea of the Binary-Cross-Entropy loss. This strategy ignores the the label dependencies in an image, where (Wang et al. 2016) utilizes RNN along with CNN to address the problem. (Chen et al. 2019) uses Graph Convolutional Network (GCN) to learn label representation by treating label as the node represented by the label’s word embedding. In (Wang et al. 2017), they utilize a spatial transformer layer as well as an LSTM to learn correlation between labels.

To utilize contrastive learning in the MLC problem, MulCon (Dao et al. 2021) propose to learn multiple label-level representations from each sample; that is, for each label in the dataset, they extract one separate representation. For one sample, the representation which represents the class it belongs to is treated as the anchor, while its positives are label-level representations from other samples belonging to the same class. They utilize BCE loss along with contrastive loss. C-GMVAE (Bai, Kong, and Gomes 2022) solves the problem by learning a latent embedding space shared by features and labels, which is similar to multi-modal contrastive learning. They map data and labels to the same probabilistic embedding space and conduct contrastive learning using the loss proposed in SupCon. They treat a sample’s feature as the anchor and embeddings from labels it belongs to as its positives.

Contrastive Learning Recently, representation learning has achieved great success in a variety of research and gained increasing attention, including contrastive representation learning. SimCLR (Chen et al. 2020) advises hard data augmentation and more negative samples for effective representation learning. MoCo (He et al. 2020) uses a queue for contrastive learning with stored negative samples. SupCon (Khosla et al. 2020) improves single-label classification using a modified contrastive loss, allowing multiple positives for one anchor. By incorporating label information, they include all samples with the same label with the anchor sample as the positives and achieve consistent improvement on various classification tasks.

Different from the above works, we neither extract label-level representations for each sample nor align representations of labels and samples. Instead, we use one representation for each sample and mainly consider the correlation between samples, similar to the idea of SimCLR or SupCon. We use the label information in an implicit way, that is, we do not bind one sample to its labels but correlate it to the samples which share some common labels.

Conclusion

MLC is an interesting problem for applying a contrastive learning paradigm since there are a plethora of ways to determine the relations between two multi-label samples. In the current work, we proposed MulSupCon, which provides a finer way to weigh one sample’s relation to the anchor based on the proportional label overlap. We compared with two other intuitive views (ALL and ANY) and the commonly-adopted multi-label BCE loss function on a series of datasets with linear probing, finetuning, and transfer learning. Robust results evaluated with six metrics suggest the effectiveness of MulSupCon. Additionally, our method demonstrates competitive performance when compared to existing approaches in the field. Our ablation study further explores the weighing mechanism in MulSupCon and its influence to the final performance.

Acknowledgements

This work has been supported by National Natural Science Foundation of China (No.92048205), the Key Research and Development Program of Jiangsu Province, China (No.BE2022059), and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102). Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

References


