Dual Relation Semi-Supervised Multi-Label Learning

Lichen Wang, Yunyu Liu, Can Qin, Gan Sun, Yun Fu
Northeastern University, Boston, USA
{wang.lich, liu.yuny, qin.ca, g.sun}@husky.neu.edu, yunfu@ece.neu.edu

Abstract

Multi-label learning (MLL) solves the problem that one single sample corresponds to multiple labels. It is a challenging task due to the long-tail label distribution and the sophisticated label relations. Semi-supervised MLL methods utilize a small-scale labeled samples and large-scale unlabeled samples to enhance the performance. However, these approaches mainly focus on exploring the data distribution in feature space while ignoring mining the label relation inside of each instance. To this end, we proposed a Dual Relation Semi-supervised Multi-label Learning (DRML) approach which jointly explores the feature distribution and the label relation simultaneously. A dual-classifier domain adaptation strategy is proposed to align features while generating pseudo labels to improve learning performance. A relation network is proposed to explore the relation knowledge. As a result, DRML effectively explores the feature-label and label-label relations in both labeled and unlabeled samples. It is an end-to-end model without any extra knowledge. Extensive experiments illustrate the effectiveness and efficiency of our method[1].

Introduction

Real-world objects could have multiple labels (*e.g.*, colors, shapes, textures, and categories). Multi-label learning (MLL) was proposed to predict tens or hundreds of different labels for a single instance. MLL has become an attractive and emerging field (Boutell et al. 2004) as it can be applied in a lot of practical applications (*e.g.*, data mining (Cong et al. 2018), image retrieval (Verma and Jawahar 2017) and image annotation (Verma and Jawahar 2017)).

There are two major challenges. First, the multi-label usually follows the long-tail distribution, which means that different labels appear in different frequencies. Some labels rarely show up (*e.g.*, *Fight* and *Fall down*) while some labels are common (*e.g.*, *Daytime* and *Natural light*). Technologically, deploying more samples in the training stage could solve this problem. However, it is not practical as the long-tail label distribution characteristic, which means it is hard to collect a dataset with enough and balanced information.

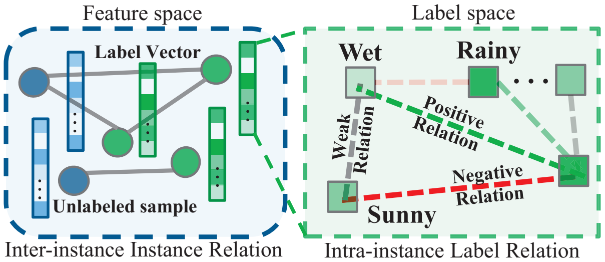

Figure 1: Two major challenges of semi-supervised MLL. 1) The labeled and unlabeled data have a distribution gap in feature space due to long-tail label distribution. 2) The label relations are complicated. (*e.g.*, ***Sunny*** and ***Rainy*** have negative relation, ***Rainy*** and ***Wet*** have positive relation, while ***Wet*** and ***Sunny*** have weak relation).

The scale of the available well-labeled datasets (Duygulu et al. 2002; Wah et al. 2011; Patterson and Hays 2012) is relatively small compared with single-label datasets. Second, the label relations are crucial to improve the MLL performance (Wu et al. 2018b). As illustrated in Figure 1. Some labels have negative relations (*e.g.*, *Sunny* and *Rainy*) which are rare to show up together. While some labels have positive relations (*e.g.*, *Rainy* and *Wet*) which usually appear together, and some labels have weak or no distinctive relations (*e.g.*, *Wet* and *Sunny*). Unfortunately, few datasets have the relation information as prior knowledge. Besides, the relation map is manually defined and task-specific. It is difficult to extend to other MLL tasks, which limits the potential applications of these approaches.

Although there are not enough labeled samples, the related unlabeled samples are easy to get. Consequently, semi-supervised learning (Zhu, Ghahramani, and Lafferty 2003) came up and has achieved great progress in MLL tasks (Dong, Li, and Zhou 2018; Zhaomin et al. 2019). Conventional semi-supervised approaches mainly analyze the data in feature space. However, the distribution of the label and unlabeled features could be different which would affect the final performance. Moreover, most methods are inspired by the single-label classification approaches while ignoring

[1] The code is available in: https://github.com/wanglichenxj/Dual-Relation-Semi-supervised-Multi-label-Learning

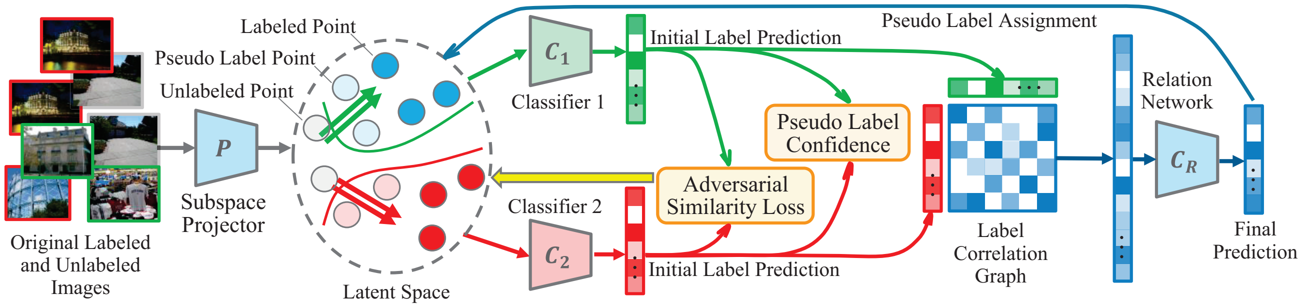

Figure 2: In our model, $P(\cdot)$ and the two classifiers (*i.e.*, $C_1(\cdot)$ and $C_2(\cdot)$) are designed to project samples from the original feature space to a latent subspace for reducing the distribution gap of labeled and unlabeled samples. The two initial predicted labels from $C_1(\cdot)$ and $C_2(\cdot)$ are forwarded to the intra-instance label relation network $C_R(\cdot)$ to further explore the label relations and get final high accurate results. The reliable pseudo labels will be aligned to unlabeled data to increase the learning performance. All modules are optimized simultaneously which is suitable for a wide range of practical applications.

the relations between multiple labels (Nie, Xu, and Li 2012).

In this work, we propose a novel Dual Relation Multi-label Learning (DRML) in semi-supervised manner. DRML includes a novel domain adaptation co-training strategy and a label relation mining module in semi-supervised fashion. It explores both the instance similarity in feature space and the label-label relation in label space simultaneously. Specifically, deploy a two-classifier domain adaptation strategy to align the feature distribution in a latent space. Moreover, it further provides the pseudo label of unlabeled samples to enhance the training performance. Furthermore, a relation network is proposed to utilize the predictions from the two classifiers to learn the label relations. All modules are simultaneously optimized in an end-to-end manner to achieve the highest performance. The major contributions of our work are briefly listed as follows:

- A two-classifier domain adaptation co-training strategy is proposed. It aligns the labeled and unlabeled samples in feature space to improve model accuracy and robustness.
- A label assignment strategy is proposed to generate pseudo labels to the unlabeled data. The assigned samples are further utilized in the training process.
- A graph-based relation network is proposed to learn the label relations for both labeled and unlabeled samples.

Our model fully utilizes the potential of a few networks which are simultaneously optimized in an end-to-end scenario. It is effective, efficient, and easy to extend to a wide of range of real-world semi-supervised applications.

Related Work

Multi-label Learning

MLL recovers multiple labels from a single sample. A lot of real-world applications are related to such problem, including text classification (Ghamrawi and McCallum 2005), image annotation (Kang, Jin, and Sukthankar 2006), and video concept recognition (Qi et al. 2007). Due to the massive amount of label combinations, MLL is more challenging compared with the single-label learning. FastTag (Chen, Zheng, and Weinberger 2013) was proposed to eliminate the negative effect of label noise. (Ge, Yang, and Yu 2018) introduces a fusion approach for MLL. To couple relevant tasks, a modulation module is proposed in (Zhao et al. 2018). However, the scales of MLL (Von Ahn and Dabbish 2004; Duygulu et al. 2002) are relatively small which limits its potential performance. Semi-supervised learning could address this issue by utilizing a small-scale of labeled data and a large-scale of unlabeled data. However, these approaches assume the distribution between labeled and unlabeled data are similar, while large distribution difference could cause a dramatic performance decrease. Label relation information is another crucial aspect for MLL. (Zhaomin et al. 2019) uses a semantic label hierarchy as prior knowledge to improve MLL performance. (Wu et al. 2018a) implements a label semantic structure, which covers different labels and avoids label noise. However, building such kind of label relation knowledge required sophisticated semantic knowledge which is difficult and expensive to get. Moreover, the obtained knowledge is difficult to extend to other datasets. This issue dramatically limits the potential of this strategy for real-world applications. Label embedding (Tai and Lin 2012) explores the label relations by projecting them to a latent space. (Chen et al. 2018) studies the object relations using attention and RNN.

Therefore, we proposed a semi-supervised MLL approach. It learns the label relations from both labeled and unlabeled instances. This makes the learned label relation knowledge more accurate and comprehensive.

Semi-supervised Learning

Semi-supervised learning (SSL) utilizes labeled as well as unlabeled sets in the training process (Zhu 2005). SSL aims to explore extra information from the unlabeled data to enhance the learning performance. (Zhu, Ghahramani, and Lafferty 2003) proposed a continuous relaxation based on the discrete Markov random fields. (Sindhwani, Niyogi, and Belkin 2005) presents a semi-supervised kernel that is suitable for all input space. (Nie, Xu, and Li 2012) introduces an initialization independent method by actively selecting the training set. (Can et al. 2019) deploys a co-training model to address the domain shift problem between source and target data. (Wang, Ding, and Fu 2018b; 2019) generates an distinctive subspace to measure the similarities across source and target frames. (Wang et al. 2018) presents a new generative approach in clustering setting. (Levatić et al. 2017) deploys decision trees and random forests to improve the performance. (Levatić et al. 2018) proposes semi-supervised trees to handle high computational cost and performance degradation issues. However, most of the methods focus on exploring the feature distribution of the unlabeled data.

In MLL scenario, the label relation is crucial. How to explore the label relation from the unlabeled data is still not well explored. In our model, the pseudo label is assigned to unlabeled data and further utilized in the training process which hopefully explores the label relations from the unlabeled samples to enhance the performance.

The Proposed Approach

Preliminaries & Motivation

Given the multi-label training data $\{X_l, Y_l\}$, where $X_l \in \mathbb{R}^{d \times n_l}$ is the feature matrix and $x_i \in \mathbb{R}^d$ represents one instance. n_l is the instance number and d is the feature dimension. $Y_l \in \mathbb{R}^{d_l \times n_l}$ is the label matrix, where d_l is the label dimension. Meanwhile, $X_u \in \mathbb{R}^{d \times n_u}$ and $Y_u \in \mathbb{R}^{d_l \times n_u}$ are the unlabeled feature and label matrix. Specifically, our approach aims to explore X_l, X_u and Y_l to recover Y_u. Since there is feature distribution gap between X_l and X_u, thus, it is natural to learn the feature presentation in a latent subspace where the labeled and unlabeled data can be well aligned. Meanwhile, there are sophisticated relations residing across different labels. To this end, a simple but effective label relation network is proposed to automatically explore the label relation knowledge. These two strategies allow the model to fully utilize the feature-label mapping and label-label relation knowledge from the labeled and unlabeled samples.

Our Approach

Our model (Figure 2) contains a projector $P(\cdot)$, two multi-label classifiers $C_1(\cdot)$ and $C_2(\cdot)$ and a label relation network $C_R(\cdot)$. $P(\cdot)$ projects all the samples into a latent space Z,

$$\begin{aligned} Z_l &= P(X_l), \\ Z_u &= P(X_u), \end{aligned} \tag{1}$$

where $Z_l \in \mathbb{R}^{d_z \times n_l}$ and $Z_u \in \mathbb{R}^{d_z \times n_u}$ are the representations of X_l and X_u in subspace Z, d_z is the dimension of Z. As mentioned above, the feature distributions of X_l and X_u could be different. Directly utilize the original features could cause a negative effect. Inspired by MDA (Saito et al. 2018), we designed a two-classifier domain adaptation framework which achieves domain adaptation and initial multi-label prediction simultaneously. For classification purpose, the loss functions of $C_1(\cdot)$ and $C_2(\cdot)$ are below,

$$L_C(X_l, Y_l) = \frac{1}{2}\left[\|C_1(Z_l) - Y_l\|_{\mathrm{F}}^2 + \|C_2(Z_l) - Y_l\|_{\mathrm{F}}^2\right], \tag{2}$$

where L_C represents the classification errors of $C_1(\cdot)$ and $C_2(\cdot)$. Meanwhile, the representation Z_l and Z_u are also optimized in the training process. Thus, $C_1(\cdot)$, $C_2(\cdot)$ and $P(\cdot)$ are simultaneously trained:

$$\min_{P, C_1, C_2} L_C(X_l, Y_l). \tag{3}$$

By this way, the initial classification results could be obtained. Moreover, the two-classifier structure is able to train the projection $P(\cdot)$ for domain adaptation goal. It aims to align the distribution shift between X_l and X_u in the latent space Z. To achieve this goal, the projection and the classifiers are further trained in an adversarial way. First, when $P(\cdot)$ is fixed, the classifier $C_1(\cdot)$ and $C_2(\cdot)$ are optimized to maximize the classification difference of the unlabeled data X_u. The prediction difference can be obtained by l_1-norm which is shown below,

$$d(f_1, f_2) = \frac{1}{d_l} \sum_{k=1}^{d_l} |f_{1k} - f_{2k}|, \tag{4}$$

where $f_1 \in \mathbb{R}^{d_l \times 1}$ and $f_2 \in \mathbb{R}^{d_l \times 1}$ are the predicted label vector from $C_1(\cdot)$ and $C_2(\cdot)$. f_{1k} and f_{2k} are the k-th entries of the label vector f_1 and f_2. Both l_1- and l_2-norm could be deployed in Eq. (4) while we empirically found out l_1-norm could achieve the best performance. It is a simple yet effective metric for measuring the prediction differences. Then, the objective of updating $C_1(\cdot), C_2(\cdot)$ for maximizing classification difference can be written as follows:

$$\min_{C1, C2} -L_{DA}(X_u) + \lambda L_C(X_l, Y_l), \tag{5}$$

$$L_{DA}(X_u) = d(C_1(Z_u), C_2(Z_u)), \tag{6}$$

where L_{DA} represents the classification difference. $C_1(\cdot)$ and $C_2(\cdot)$ are trained to maximize the classification differences of the unlabeled data, while it still needs to secure the classification performance on labeled samples. Thus, we add the L_C term in the objective, and $\lambda > 0$ is the trade-off parameter which balances the weight between classification difference and accuracy. On the other hand, $P(\cdot)$ tries to update the projection space which minimizes the unlabeled data classification difference. To this end, $P(\cdot)$ can be updated by the following function:

$$\min_{P} L_{DA}(X_u). \tag{7}$$

Projection $P(\cdot)$, classifier $C_1(\cdot)$ and $C_2(\cdot)$ are alternately updated in an adversarial fashion based on Eq. (5) and Eq. (7). By this way, the labeled and unlabeled samples would be gradually aligned in the latent space Z, which could effectively reduce the negative influence of the distribution shift of labeled and unlabeled samples.

The outputs from both classifier $C_1(\cdot)$ and $C_2(\cdot)$ can be the final classification results. Averaging these two prediction results is an efficient strategy. However, as introduced before, label relation and trivial prediction differences are crucial to further improve the learning performance. To this end, we propose a simple but effective label-level relation network, $C_R(\cdot)$, to automatically explore the label relation knowledge. As shown in Figure 2, after the predicted label f_1 and f_2 are obtained, we designed a label relation graph R_i by multiplying f_1 and the transposition of f_2 as $R_i = f_1 \times f_2^{\top}$, where $R_i \in \mathbb{R}^{d_l \times d_l}$ is the relation matrix and d_l is the label dimension. The obtained R_i is reshaped to a $\mathbb{R}^{d_l^2 \times 1}$ vector and forwarded to a fully connected relation network $C_R(\cdot)$. $C_R(\cdot)$ further predicts the multi-label result based on R_i. To this end, the objective of the relationship network is shown below:

$$L_R = \sum_{i=1}^{n} \|y_i - C_R\big(C_1(P(x_i)) \cdot C_2(P(x_i))^{\top}\big)\|_2^2, \quad (8)$$

where x_i and $y_i \in \mathbb{R}^{d_l \times 1}$ are a training sample and its ground truth multi-label vector of x_i. In this framework, the elements in R_i are the multiplication of each pair of the predicted labels, which could be considered as a dot-product similarity metric of the pairwise labels (including the similarity with itself). By this way, $C_R(\cdot)$ explores the latent relation knowledge residing inside the training data based on the obtained similarities, and further refine the predicted label from $C_1(\cdot)$ and $C_2(\cdot)$ to improve performance. In the training procedure, $C_R(\cdot)$ is trained simultaneously with the other networks which is shown as follow:

$$\min_{P,C_1,C_2,C_R} \frac{\alpha}{2} L_C + (1-\alpha) L_R, \quad (9)$$

where α is the trade-off parameter which balances the weight between initial prediction error and the relation network prediction error. Jointly optimizing $C_{1,2}(.)$ and $C_R(.)$ by combining their losses together could 1) control the training of $C_{1,2}(.)$ to predict initial labels and 2) intentionally force $C_R(.)$ to capture the label relations based on the initial labels from $C_{1,2}(.)$. This strategy balances the update processing between $C_{1,2}(.)$ and C_R to further help each other in the training stage and achieve a promising performance at last. α is set to 0.5 as default. We have observed that slightly tuning α near 0.5 does increase the performance a little, and cross validation could be employed for automatic parameter tuning. Since the improvement is not significant, thus, we set $\alpha = 0.5$ which avoids the parameter tuning procedure.

$C_R(.)$ can be easily deployed for labeled samples. Meanwhile, in semi-supervised learning scenario, we assume that the unlabeled samples also include informative and comprehensive label relation knowledge. To this end, we utilize predicted labels from partial unlabeled samples as pseudo labels in the training process. By this way, $C_R(.)$ could further explore the correlation from the unlabeled samples and increase the learning performance. Since the prediction results of $C_1(\cdot)$ and $C_2(\cdot)$ are not reliable at the beginning of the training procedure. To this end, we first trained $C_1(\cdot)$ and $C_2(\cdot)$ for 50 to 100 iterations before we involve the pseudo label strategy in the complete training procedure.

Table 1: MLL performance

Data	Method	Pre	Rec	F1	N-R	mAP
Corel	LR	0.2859	0.3211	0.3025	128	0.3630
	SSMLDR	0.2741	0.3366	0.3022	143	0.3410
	FastTag	0.3123	0.3657	0.3369	143	0.3871
	ML-PGD	0.2575	0.2911	0.2732	122	0.3727
	SAE	0.2962	0.3442	0.3184	141	0.3823
	AG2E	0.3011	0.3520	0.3245	**157**	0.3568
	Ours	**0.3154**	**0.3775**	**0.3437**	148	**0.4127**
ESP	LR	0.3793	0.2038	0.2653	215	0.3440
	SSMLDR	0.3298	0.1885	0.2399	226	0.3156
	FastTag	0.4011	0.1927	0.2617	208	0.3904
	ML-PGD	0.3239	0.2012	0.2482	210	0.4077
	SAE	0.3861	0.1743	0.2402	194	0.3842
	AG2E	0.3548	0.1525	0.2133	213	0.3730
	Ours	**0.4373**	**0.2189**	**0.2918**	**227**	**0.4105**
IAP	LR	0.4287	0.2041	0.2765	199	0.4211
	SSMLDR	0.3491	0.2520	0.2927	229	0.3981
	FastTag	0.4346	0.2267	0.2980	227	0.4596
	ML-PGD	0.4132	0.2441	0.3011	230	0.4674
	SAE	0.3537	0.2282	0.2774	213	0.4309
	AG2E	0.3829	0.2330	0.2897	229	0.4353
	Ours	**0.4570**	**0.2531**	**0.3258**	**230**	**0.5148**
SUN	LR	0.6209	0.1473	0.2457	102	0.6807
	SSMLDR	0.6879	0.1700	0.2726	102	0.6723
	FastTag	0.6816	0.1473	0.2457	102	0.6914
	ML-PGD	0.7110	0.1614	0.2631	101	**0.7087**
	SAE	0.7183	0.1638	0.2668	98	0.7012
	AG2E	0.7685	0.1765	0.2871	99	0.6778
	Ours	**0.7906**	**0.1793**	**0.2923**	**102**	0.6800
CUB	LR	0.2010	0.0239	0.0428	157	0.0638
	SSMLDR	0.3410	0.0473	0.0832	178	0.2329
	FastTag	0.2147	0.0359	0.0615	167	0.3144
	ML-PGD	0.3334	0.0451	0.0794	155	0.3288
	SAE	0.3383	0.0514	0.0908	196	0.3255
	AG2E	0.3409	0.0531	0.0911	190	0.3106
	Ours	**0.3714**	**0.0548**	**0.0955**	**202**	**0.3542**
AWA	LR	0.8798	0.0821	0.1500	75	0.8626
	SSMLDR	0.7812	0.0858	0.1546	67	0.8346
	FastTag	0.7861	0.0949	0.1694	72	0.8791
	ML-PGD	0.5395	0.0635	0.1136	57	0.9121
	SAE	**0.9683**	**0.0957**	**0.1742**	73	0.9397
	AG2E	0.8483	0.0827	0.1507	73	0.9033
	Ours	0.8689	0.0835	0.1523	**75**	**0.9441**

Compared with single-label learning, we cannot simply determine the confidence of the predicted labels. To handle this problem, we fully utilize the two-classifier structure and measure the prediction differences between $C_1(\cdot)$ and $C_2(\cdot)$. Specifically, all target data are sent to $C_1(\cdot)$, $C_2(\cdot)$ and $C_R(\cdot)$ and achieve the predictions. The prediction differences are calculated by Eq. (10). Unlike the previous prediction difference in (6), we use l_2-norm which is shown as follows:

$$D(z_i) = \|C_1(z_i) - C_2(z_i)\|_2^2, \quad (10)$$

where z_i is the i-th sample representation in space Z. Due to the difference among the datasets (*e.g.*, feature scale, label numbers and labels formats), the pseudo label strategy is deployed a little differently for different datasets. Take CUB dataset (Wah et al. 2011) for example, we set a threshold value $d = 1$. If $D(Z_i) \leq d$, we will select the testing instance x_i with a pseudo label. For the other dataset, we select $\mathcal{K}$ samples with the lowest differences from the whole predictions. $\mathcal{K}$ is normally set to 10 to 20. After that, we give them the pseudo labels and add them to the training set for the next training loop. We deploy fully connected networks in our implementation. Other deep networks can be used to attain higher performance. In our implementation, $P(\cdot)$ is an one-layer fully-connected linear network. $C_1(\cdot)$ and $C_2(\cdot)$ are both one-layer fully-connected network with a Sigmoid activation after the last layer. $C_R(\cdot)$ is an one-layer fully-connected network with Sigmoid activation after the last layer.

Our model contains four networks which are jointly opti-

Table 2: MLL performance on augmented label sets

Data	Methods	Pre	Rec	F1	N-R	mAP
Corel-A	LR	0.2842	0.2304	0.2545	103	0.3762
	SSMLDR	0.3036	0.2791	0.2908	134	0.3660
	FastTag	0.3329	0.3145	0.3234	136	0.4127
	ML-PGD	0.3245	0.3011	0.3124	140	0.4275
	SAE	0.3168	0.3037	0.3101	128	0.4192
	AG2E	0.3273	0.3172	0.3221	143	0.3985
	Ours	**0.3345**	**0.3671**	**0.3500**	**147**	**0.4315**
ESP-A	LR	0.3848	0.1256	0.1894	178	0.3913
	SSMLDR	0.3253	0.1697	0.2231	202	0.3357
	FastTag	0.3886	0.1531	0.2197	196	**0.4254**
	ML-PGD	0.3713	0.1184	0.1795	162	0.4211
	SAE	0.3153	0.1425	0.1966	156	0.4050
	AG2E	0.3518	0.1492	0.2095	196	0.4030
	Ours	**0.4202**	**0.1744**	**0.2465**	**209**	0.4121

Table 3: Zero-shot MLL performance

Data	Method	Pre	Rec	F1	N-R	mAP
SUN	LR	0.7047	0.1548	0.2539	97	0.6616
	SSMLDR	0.6637	0.1481	0.2422	95	0.6581
	FastTag	0.6906	0.1522	0.2494	90	0.6706
	ML-PGD	0.7037	0.1471	0.2433	95	0.6829
	SAE	0.6978	0.1710	0.2747	**100**	0.6513
	AG2E	0.7125	0.1618	0.2637	88	0.6693
	Ours	**0.7512**	**0.1794**	**0.2896**	97	**0.6924**
CUB	LR	0.2600	0.0307	0.0549	160	0.2693
	SSMLDR	0.2926	0.0383	0.0677	166	0.2329
	FastTag	0.2231	0.0434	0.0726	143	0.2967
	ML-PGD	0.2392	0.0365	0.0635	117	0.3178
	SAE	0.2552	0.0469	0.0798	**167**	0.3102
	AG2E	0.2808	0.0481	0.0821	163	0.2693
	Ours	**0.2981**	**0.0486**	**0.0835**	153	**0.3338**
AWA	LR	0.7555	0.0766	0.1392	66	0.8809
	SSMLDR	0.7017	0.0764	0.1378	66	0.7858
	FastTag	0.8610	0.0912	0.1649	81	0.8918
	ML-PGD	0.4338	0.0623	0.1091	49	0.8677
	SAE	0.9015	**0.0926**	**0.1679**	78	0.8918
	AG2E	0.8247	0.0811	0.1476	71	0.8874
	Ours	**0.9023**	0.0832	0.1524	**81**	**0.8985**

mized in a minimax strategy, which brings in several advantages. First, it is an end-to-end model without the requirement of any other prior knowledge, which is easy to train and compatible for a lot of real-world applications. Second, the performance is stable and robust since the domain adaptation strategy is able to well align the distribution shift across the labeled and unlabeled data. Third, the label-level correlation is explored by the relation network in both labeled and unlabeled samples. Forth, our approach can be directly deployed for more testing data samples without any other optimization operations which are more simple and efficient compared with graph-based semi-supervised approaches.

Experiment

Multi-label Datasets

We evaluate our model on six fine-grained multi-label datasets. **ESP Game** (Von Ahn and Dabbish 2004) has $18,689$ training images and $2,081$ testing images which is labeled by an ESP interactive gaming system. **Corel5K** (Duygulu et al. 2002) is an image dataset of the CDs. There are $4,500$ training samples and 499 testing samples. It is represented by a 260-dimensional semantic description vector in binary format. **IAPRTC-12** (Grubinger et al. 2006) includes images of actions, animals, landscapes and other objects. It has $19,627$ training images and $1,962$ testing images. The label is represented by a 291-dimensional vector in binary format. Each sample has 5.72 labels in average. **CUB** (Wah et al. 2011) has $8,800$ training images and $1,440$ testing images. This dataset contains 200 birds. The label information can be described by a 312-dimensional vector in binary format. Each instance has 31.39 labels in average. **SUN** (Patterson and Hays 2012) contains $12,900$ training images and $1,440$ testing images such as *bakery*, *ballroom* and *balcony*. There are 717 scene classes in total. These labels are assigned by multiple trained labors. Each instance has 6.31 labels in average. **AWA** (Lampert, Nickisch, and Harmeling 2014) contains $24,295$ training images and $6,180$ testing images. This dataset consists of 50 animal species. Each instance has 15 labels in average.

We directly deploy the visual descriptors provided by (Guillaumin et al. 2009) for Corel5K, IAPRTC and ESP Game datasets. A pre-trained VGG Networks (Simonyan and Zisserman 2014) based on ImageNet is set as feature extractor for SUN, CUB and AWA datasets.

Experimental Setup

We evaluate our approach associated with several state-of-the-art representative MLL methods. **Least Square Regression (LR)** directly learns a linear regression model from the feature space to the label spaces. **Semi-Supervised Multi-Label Dimension Reduction (SSMLDR)** (Guo et al. 2016) explores the information in both labeled and unlabeled data. To enhance the model robustness, it designs a specific label propagation strategy. **Fast Image Tagging (FastTag)** (Chen, Zheng, and Weinberger 2013) introduces two linear mappings to obtain the whole tags based on the incomplete tags. These two mappings are regularized in one loss function. **Multi-Label learning using a Mixed Graph (ML-PGD)** (Wu, Lyu, and Ghanem 2015) introduces a label dependencies model. It constructs a mixed graph and takes the similarity of instance level with class co-occurrence into consideration. **Semantic AutoEncoder (SAE)** (Kodirov, Xiang, and Gong 2017) introduces an effective auto-encoder model to recover labels. It also proposes an additional reconstruction constraint. **Adaptive Graph Guided Embedding (AG2E)** (Wang, Ding, and Fu 2018a) proposes an adaptive graph strategy. It jointly obtains the similarity graph and predicts multiple labels in a semi-supervised fashion.

Since $C_R(\cdot)$ depends on the performance of $C_1(\cdot)$ and $C_2(\cdot)$, we train $C_1(\cdot)$ and $C_2(\cdot)$ for 50 epochs prior the training of other networks. When $C_1(\cdot)$ and $C_2(\cdot)$ become gradually stable, we begin to train $C_1(\cdot)$, $C_2(\cdot)$ and $C_R(\cdot)$ simultaneously. We repeat the training procedure until $C_R(\cdot)$ achieves a stable performance. After that, we add the pseudo label assignment section in the training procedure. Due to the difference label formats of the datasets, the pseudo label assignment approach is slightly different between different dataset. For CUB and AWA datasets, the pseudo label is the original output of $C_R(\cdot)$. For SUN (Patterson and Hays 2012) dataset, the pseudo label is represented by the combinations of $\{0, 0.33, 0.66, 1\}$ due to its unique label format. For other datasets, the pseudo label is binary.

We deploy the metrics proposed in (Guillaumin et al. 2009) for evaluation. The precision (Pre) P and the re-

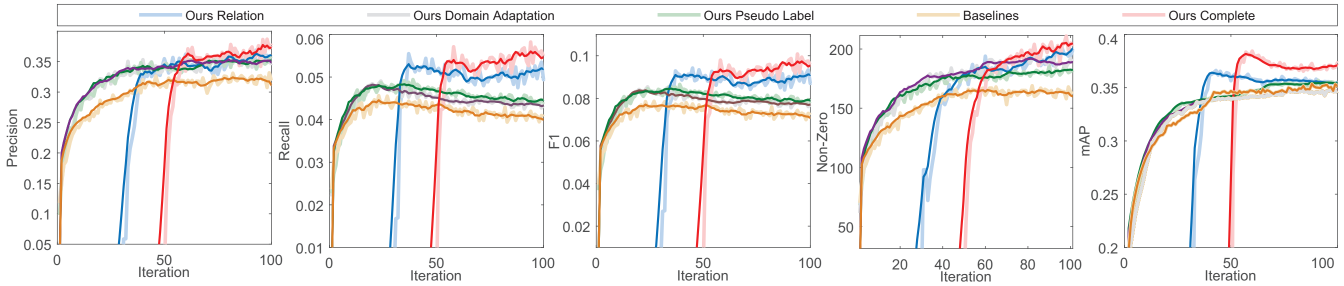

Figure 3: Ablation study. MLL performance along with training iterations in the CUB dataset. Different color indicates different models. **Red**: Our complete model. **Blue**: without domain adaptation and pseudo labeling. **Purple**: without relation network and pseudo labeling. **Green**: without domain adaptation and relation network. **Yellow**: only domain adaptation.

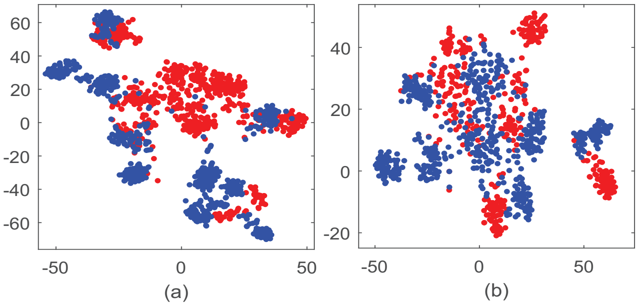

Figure 4: Visualization of 10 labelled (blue) and unlabelled (red) data before (a) and after (b) the projection.

call (Rec) R are calculated. $P = \frac{t_p}{t_p+f_p}$ and $R = \frac{t_p}{t_p+f_n}$, where t_p is true-positive, f_p is false-positive, and f_n is false-negative. For easy comparison, we calculate the F1-score, the harmonic mean of R and P, where $F1 = 2\frac{P \times R}{P+R}$. The number of labels with a non-zero recall (N-R) value and the mean average precision (mAP) (Wu, Lyu, and Ghanem 2015) are further used for comprehensive evaluation. In all metrics, the higher value, the better performance.

Conventional & Zero-shot MLL

The result of conventional MLL is shown in Table 1. Our approach surpasses other baselines in most evaluations. Furthermore, (Wu, Lyu, and Ghanem 2015) proposes an augmented label set for Corel5K and ESP Game datasets. It increases average label number of Corel5K from 3.40 to 4.84, and the ESP from 4.69 to 7.27. We evaluate our model based on these label sets. The results (Table 2) indicates that our approach still achieves the best performance in most of the matrices.

We further apply our method to zero-shot MLL scenario which means the classes in training and testing sets are non-overlapping, while they still share same the multi-labels (*e.g.*, *horse* and *Zebra*). It is more difficult because of the larger distribution gap between the two sets. We evaluate our approach based on SUN, CUB and AWA datasets. The default training and testing splits are provided. The specific splits are $645/72$, $40/10$ and $150/50$ respectively. For CUB, there are 4 different ways to split. We test the model once for each split and calculate the average performance. For SUN and AWA datasets, we test the model for 5 times and calculate the mean performance.

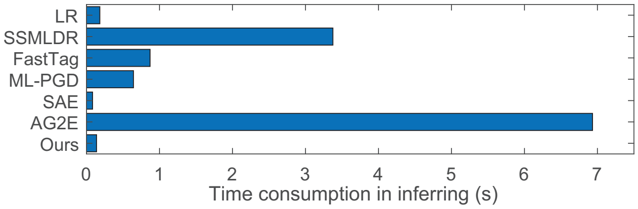

Figure 5: Time consumption in inferring process.

Table 3 indicates that our approach outperforms other methods which demonstrates that our approach is accurate and robust. In the real world, this is helpful since collecting the images from all possible classes is impossible. We also notice that our approach cannot achieve the highest performance in AWA dataset. We consider this in the following reasons. 1), AWA samples that belong to the same class share only one consistent semantic description, thus, it is difficult to comprehensively learn the feature-label relations; 2), there are limited relation information learned by CDN due to the consistent label issue.

Ablation Study

We run our model with and without CDN and the domain adaptation strategy on CUB dataset. Figure 3 illustrates the performance with the iteration increasing and details are introduced in the caption. The result illustrates that all the strategies can effectively improve the performance respectively, and the combination of all the proposed approaches do help each other and dramatically improve/stabilize the performance. We further visualize the original and projected features of 10 labelled (blue circle) and 10 unlabelled(yellow circle) classes from CUB dataset (Figure 4) by t-SNE (Van Der Maaten 2014). It illustrates that two distribution gap becomes smaller which demonstrates the effectiveness of the domain adaptation strategy.

Figure 5 shows the time consumption of each method in the test stage. Due to the simple feed-forward network struc-

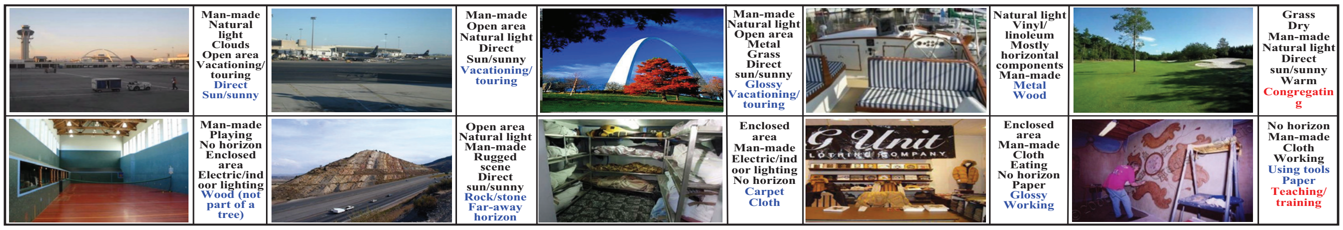

Figure 6: Samples of recovered labels from SUN dataset. Black font denotes correct prediction. Blue font denotes labels that do not exist in the ground truth but match our judgments. Red font denotes incorrect prediction.

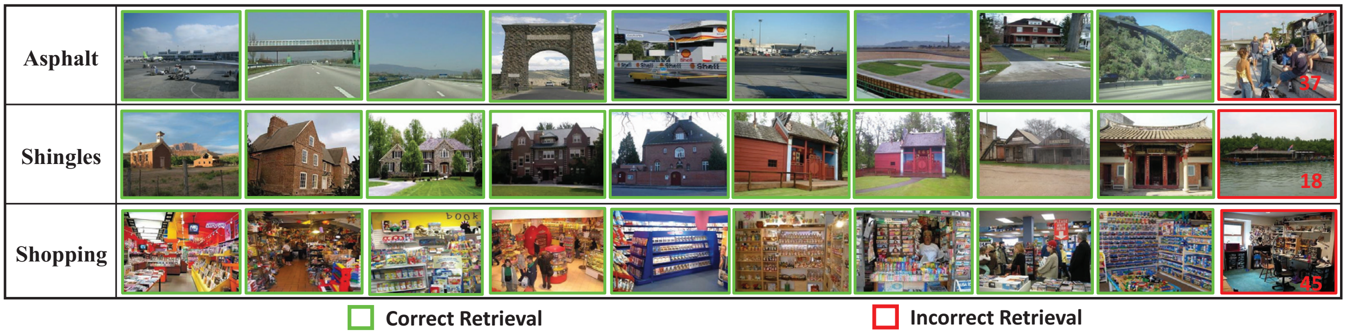

Figure 7: Image retrieval result of SUN dataset in the zero-shot scenario. Each row shows the images with the highest corresponding label scores. Green and red boxes indicate correct and incorrect retrieval, respectively. For each target label, we show incorrect retrieval result and its score rankings on the image right corner.

ture and the usage of GPU acceleration, even if the computational cost is a little higher, our approach only spends an average of 0.12 seconds to infer 2081 test samples which is the second fastest method. It indicates that our approach fits well for large-scale real-world applications.

Image Retrieval

The goal of Image retrieval is to find the images from a large dataset which contain one or more specific labels. We obtain the labels of each candidate image using our well-trained model then rank the labels by the predicted scores. We still deployed the zero-shot split setting since it is a more practical setting for most real-world applications.

Figure 7 shows the images extracted from the dataset based on specific labels. We select incorrect results and mark the images with the ranking numbers on the bottom corner. We can see that most images are correctly retrieved with only a few errors. First, the adjective and verb labels are easier to retrieve than noun labels. And noun labels are easier to make a mistake when the image has a similar label and similar color (*e.g.*, *ice* and *running water*). Second, the model always retrieval the scene from very similar class. Such as *Asphalt*, the model prefers to retrieve all field scenes on the road and in the airport. We can study these in the future.

Image Annotation

We evaluate image annotation performance on SUN dataset in Figure 6. We set different colors to indicate different labels and details can be found in the caption. We observe that most recovered labels are correct in our experiment. Under some circumstances, our model even discovered “new” labels. This proves that our approach is robust, effective and is able to recover the vast majority of labels with high accuracy. Furthermore, our model can discover the missing and error labels of the ground truth.

Conclusion

In this paper, we proposed a Dual Relation Multi-label Learning (DRML) approach for Semi-supervised manner. DRML explores the instance-level relations across labeled and unlabeled samples in feature space. Meanwhile, it further explores the label-level relations residing in each sample in label space. A projector associated with two-classifier has been designed to align the distribution gap. A pseudo label assignment strategy is proposed to further improve learning performance. A relation network has been further deployed to automatically explore the relations across labels. All modules are optimized simultaneously. Experiments demonstrate the efficiency and effectiveness of DRML. Ablation study has demonstrated the necessities of all proposed strategies for high accuracy.

Acknowledgement. This research is supported by the U.S. Army Research Office Award W911NF-17-1-0367.

References

Boutell, M. R.; Luo, J.; Shen, X.; and Brown, C. M. 2004. Learning multi-label scene classification. *Pattern Recognition* 37(9):1757–1771.

Can, Q.; Lichen, W.; Yulun, Z.; and Yun, F. 2019. Generatively inferential co-training for unsupervised domain adaptation. In *ICCV Workshop*.

Chen, S.; Chen, Y.; Yeh, C.; and Wang, Y. F. 2018. Order-free RNN with visual attention for multi-label classification. In *AAAI*.

Chen, M.; Zheng, A.; and Weinberger, K. 2013. Fast image tagging. In *ICML*, 1274–1282.

Cong, Y.; Sun, G.; Liu, J.; Yu, H.; and Luo, J. 2018. User attribute discovery with missing labels. *Pattern Recognition* 73:33–46.

Dong, H.; Li, Y.; and Zhou, Z. 2018. Learning from semi-supervised weak-label data. In *AAAI*.

Duygulu, P.; Barnard, K.; de Freitas, J. F.; and Forsyth, D. A. 2002. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, 97–112.

Ge, W.; Yang, S.; and Yu, Y. 2018. Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning. In *CVPR*.

Ghamrawi, N., and McCallum, A. 2005. Collective multi-label classification. In *CIKM*, 195–200.

Grubinger, M.; Clough, P.; Müller, H.; and Deselaers, T. 2006. The IAPR TC12 benchmark: A new evaluation resource for visual information systems. In *OntoImage*.

Guillaumin, M.; Mensink, T.; Verbeek, J.; and Schmid, C. 2009. Tagprop: Discriminative metric learning in nearest neighbor models for image annotation. In *ICCV*, 309–316.

Guo, B.; Hou, C.; Nie, F.; and Yi, D. 2016. Semi-supervised multi-label dimensionality reduction. In *ICDM*, 919–924.

Kang, F.; Jin, R.; and Sukthankar, R. 2006. Correlated label propagation with application to multi-label learning. In *CVPR*, volume 2, 1719–1726.

Kodirov, E.; Xiang, T.; and Gong, S. 2017. Semantic autoencoder for zero-shot learning. In *CVPR*, 3174–3183.

Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2014. Attribute-based classification for zero-shot visual object categorization. *TPAMI* 36(3):453–465.

Levatić, J.; Ceci, M.; Kocev, D.; and Džeroski, S. 2017. Semi-supervised classification trees. *JIIS* 49(3):461–486.

Levatić, J.; Kocev, D.; Ceci, M.; and Džeroski, S. 2018. Semi-supervised trees for multi-target regression. *Information Sciences* 450:109–127.

Nie, F.; Xu, D.; and Li, X. 2012. Initialization independent clustering with actively self-training method. *Trans. on Cybernetics* 42(1):17–27.

Patterson, G., and Hays, J. 2012. SUN attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2751–2758.

Qi, G.; Hua, X.; Rui, Y.; Tang, J.; Mei, T.; and Zhang, H. 2007. Correlative multi-label video annotation. In *Multimedia*, 17–26.

Saito, K.; Watanabe, K.; Ushiku, Y.; and Harada, T. 2018. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 3723–3732.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*.

Sindhwani, V.; Niyogi, P.; and Belkin, M. 2005. Beyond the point cloud: from transductive to semi-supervised learning. In *ICML*, 824–831.

Tai, F., and Lin, H. 2012. Multilabel classification with principal label space transformation. *Neural Computation* 24(9):2508–2542.

Van Der Maaten, L. 2014. Accelerating t-SNE using tree-based algorithms. *JMLR* 15(1):3221–3245.

Verma, Y., and Jawahar, C. 2017. Image annotation by propagating labels from semantic neighbourhoods. *IJCV* 121(1):126–148.

Von Ahn, L., and Dabbish, L. 2004. Labeling images with a computer game. In *SIGCHI*, 319–326.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001.

Wang, Q.; Ding, Z.; Tao, Z.; Gao, Q.; and Fu, Y. 2018. Partial multi-view clustering via consistent GAN. In *ICDM*, 1290–1295.

Wang, L.; Ding, Z.; and Fu, Y. 2018a. Adaptive graph guided embedding for multi-label annotation. In *IJCAI*, 2798–2804.

Wang, L.; Ding, Z.; and Fu, Y. 2018b. Learning transferable subspace for human motion segmentation. In *AAAI*, 4195–4202.

Wang, L.; Ding, Z.; and Fu, Y. 2019. Low-rank transfer human motion segmentation. *TIP* 28(2):1023–1034.

Wu, B.; Chen, W.; Sun, P.; Liu, W.; Ghanem, B.; and Lyu, S. 2018a. Tagging like humans: Diverse and distinct image annotation. In *CVPR*, 7967–7975.

Wu, B.; Jia, F.; Liu, W.; Ghanem, B.; and Lyu, S. 2018b. Multi-label learning with missing labels using mixed dependency graphs. *IJCV* 1–22.

Wu, B.; Lyu, S.; and Ghanem, B. 2015. ML-MG: multi-label learning with missing labels using a mixed graph. In *ICCV*, 4157–4165.

Zhao, X.; Li, H.; Shen, X.; Liang, X.; and Wu, Y. 2018. A modulation module for multi-task learning with applications in image retrieval. In *ECCV*.

Zhaomin, C.; Xiushen, W.; Peng, W.; and Yanwen, G. 2019. Multi-label image recognition with graph convolutional networks. In *CVPR*.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. D. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 912–919.

Zhu, X. 2005. Semi-supervised learning literature survey.