Multi-label Ranking from Positive and Unlabeled Data

Atsushi Kanehira and Tatsuya Harada
The University of Tokyo
7-3-1 Hongo Bunkyo-ku, Tokyo, Japan
{kanehira, harada}@mi.t.u-tokyo.ac.jp

Abstract

In this paper, we specifically examine the training of a multi-label classifier from data with incompletely assigned labels. This problem is fundamentally important in many multi-label applications because it is almost impossible for human annotators to assign a complete set of labels, although their judgments are reliable. In other words, a multi-label dataset usually has properties by which (1) assigned labels are definitely positive and (2) some labels are absent but are still considered positive. Such a setting has been studied as a positive and unlabeled (PU) classification problem in a binary setting. We treat incomplete label assignment problems as a multi-label PU ranking, which is an extension of classical binary PU problems to the well-studied rank-based multi-label classification. We derive the conditions that should be satisfied to cancel the negative effects of label incompleteness. Our experimentally obtained results demonstrate the effectiveness of these conditions.

1. Introduction

Multi-label classification treats a problem that allows samples to take more than one label. Although the simplest solution for multi-label classification is training an independent classifier per class, a trained model is well known to have low classification performance when there is a correlation between classes [7]. For this reason, a multi-label learning method, which incorporates label dependency, is needed. In recent years, many studies have specifically addressed multi-label learning [7], [19], [6]. Furthermore, there are widely diverse applications in many domains including computer vision [5], [27], [18].

To collect a dataset for multi-label classification, researchers generally use crowdsourcing. An alternative is to collect data in a semi-automatic way as in [26]. In most cases, the obtained labels will be incomplete but reliable because it is almost impossible to assign a full set of labels to describe images completely in the real world. For instance, let us consider the case in which human annotators attach labels to the leftmost image in Fig. 1. They might identify ”dog” and ”chair,” and then assign them because they are the main components. However, besides these, ”sofa,” ”carpet,” and ”box” can also be used. In addition, numerous other possible correct answers exist such as scene, breed of dog, and attribute.

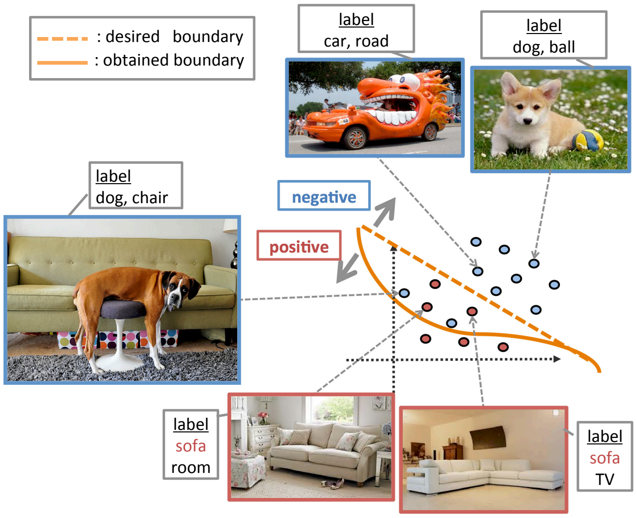

Figure 1. Multi-label dataset tends to have partially labeled samples. Absent labels are regarded as negative, and it affects classification performance.

As in the example presented above, the obtained dataset has properties by which (1) assigned labels are definitely positive and (2) absent labels are not necessarily negative. Because conventional multi-label learning models ignore this incompleteness and because they regard unlabeled objects as negative, their performance will be affected as in the case of the right-hand side of Fig. 1. Therefore, an incomplete label assignment problem is fundamentally important and critical in multi-label learning, which should be solved.

Our goal in this paper is to propose a method that enables us to train a classifier consistently from data with incompletely assigned labels. We deal with the setting as follows:

1. Assigned labels are definitely positive.
2. Absent labels are not necessarily negative.

3. Lastly, samples are allowed to take more than one label.

Settings (1) and (2) have been studied as positive and unlabeled (PU) classification problems in a binary case [15], [22]. Moreover, setting (3) is a multi-label classification setting. In this work, we deal with a multi-label PU ranking problem to treat the setting, which includes all of (1), (2), and (3); then, by extending an analysis for binary PU classification [10] to a multi-label problem, we derive the conditions under which the loss function should be satisfied to have consistency even if assigned labels are incomplete. The main contributions of this work are as follows:

1. We derive the conditions that should be satisfied to cancel the negative effects of label incompleteness in multi-label PU ranking.
2. We demonstrate the effectiveness of these conditions using experiments on several multi-label datasets.

In Sec.1, we describe the goals and contributions of this work. We then discuss related works in Sec. 2. In Sec. 3, we describe the settings of multi-label rankings. In Sec. 4, we explain an extension of the analyses for binary PU to multi-label problems and describe the conditions under which the loss function is satisfied. Several experiments on synthetic datasets and image annotation datasets are explained in Sec. 5 to investigate the efficacy of the derived conditions. Experimental results are discussed in Sec. 6. Then, we conclude our work in Sec. 7.

2. Related work

2.1. Multi-label ranking

Multi-label classification problems have been studied in recent years. One of the most common approaches is based on label ranking. Label ranking is aimed at ranking all positive classes higher than negative ones by minimizing rank loss. Rank loss, originally proposed by [13], has been studied well [16]. Actually, [9] relaxed the constraint condition to allow the application of the algorithm to large-scale data. In the computer vision domain, many algorithms have been proposed based on label ranking. [27] proposed a learning model, WSABIE, that embeds image and word features to a common space by optimizing the weighted rank loss. [4], [1] used rank loss for the object recognition task. In addition, [17] trained a deep convolutional neural network for a multi-label task by replacing the softmax loss with the weighted rank loss proposed by [27]. However, all of these works assume that all labels are assigned completely and do not deal directly with label incompleteness.

2.2. PU Classification

The problem of training classifiers from positive samples and unlabeled samples is called PU classification. Some studies have addressed this problem [22]. Actually, [15] constructed a probabilistic model only from observable samples and estimated a proper classifier using it. In addition, [10] analyzed a binary PU classification problem and revealed that PU classification can be cast as a cost-sensitive learning, which changes the weight of the penalty per class. Using a symmetric non-convex function as a surrogate loss makes it possible to learn consistently. However, these studies emphasized only binary classification problems and did not assume that samples take more than one label.

2.3. Learning from incompletely labeled data

Some works have attempted to address label incompleteness in multi-label learning as label deficits. [3] tried to eliminate the influence of label deficits in the optimization process by adding a regularization term to rank loss, which forces the difference between scores for positive and negative labels to be group sparse. Then, [20] extended [15] to a multi-label setting by considering the label dependency. [28] dealt with weak labels in a multiple-instance, multi-label learning setting. Subsequently, [23] used a conditional restricted Boltzmann machine to denoise the label deficit. However, these studies did not mention that the condition loss function should be satisfied.

3. Multi-label ranking

In this section, we describe the setting of the multi-label ranking problem. Let $\mathcal{X}$ be a sample space and $\mathcal{Y}=\{0,1\}^m$ be the possible set of labels, where m denotes the number of classes. y_i denotes the status of the sample in terms of the i-th class: if $y_i=1$, then the i-th class is positive for a given sample, and if $y_i=0$, then it is negative. A dataset having N samples $S=\{(\mathbf{x}_1,\mathbf{y}_1),(\mathbf{x}_1,\mathbf{y}_1),...,(\mathbf{x}_N,\mathbf{y}_N)\}$ is generated from an unknown distribution on $\mathcal{X}\times\mathcal{Y}$. A score function is defined as $\mathbf{f}(\mathbf{x})=(f_1(\mathbf{x}),f_2(\mathbf{x}),...,f_m(\mathbf{x})):\mathcal{X}\to\mathbb{R}^m$. In an empirical risk minimization framework, algorithms are used to minimize the expectations of the loss function over the (sample, label) space. In other words, the following $\mathbf{f}^*=\mathrm{argmin}\,\mathcal{L}(\mathbf{f})$ is computed:

$$\mathcal{L}(\mathbf{f}) = \mathbb{E}_{\mathbf{xy}}[L(\mathbf{f}(\mathbf{x}),\mathbf{y})]. \tag{1}$$

$L(\mathbf{f}(\mathbf{x}),\mathbf{y})$ indicates the loss function that takes a label and a score for the sample as input. Although some loss functions including 0-1 subset loss and Hamming loss are proposed, we treat the rank loss, which is commonly used in multi-label learning.

3.1. Rank loss

Rank loss imposes a penalty on a classifier when a pair of labels is incorrectly ranked. It can be defined as follows:

$$L_{\mathrm{rank}}(\mathbf{f}(\mathbf{x}),\mathbf{y}) = \sum_{\{i,j:y_i=1,y_j=0\}} [[f_i<f_j]]+\frac{1}{2}[[f_i=f_j]], \tag{2}$$

where i,j are the indices of the class. The i-th element of $\mathbf{f}(\mathbf{x})$, which means the score for the i-th class, is denoted by f_i, omitting the dependency for $\mathbf{x}$. $[[\cdot]]$ is the indicator function that takes a value of 1 when the conditions inside the brackets are met; otherwise, it is 0. From (1), the loss function that should be minimized is

$$\begin{aligned}&\mathcal{L}_{\text{rank}} = \mathbb{E}_{\mathbf{xy}}[L_{\text{rank}}(\mathbf{f}(\mathbf{x}),\mathbf{y})]\\&=\sum_{\mathbf{y}\in\mathcal{Y}}P(\mathbf{y})\mathbb{E}_{\mathbf{x}|\mathbf{y}}[L_{\text{rank}}(\mathbf{f}(\mathbf{x}),\mathbf{y})]\\&=\sum_{\mathbf{y}\in\mathcal{Y}}P(\mathbf{y})\sum_{\{i,j:y_i=1,y_j=0\}}\mathbb{E}_{\mathbf{x}|\mathbf{y}}\left[[[f_i<f_j]]+\frac{1}{2}[[f_i=f_j]]\right].\end{aligned} \tag{3}$$

By swapping two summations, we can rewrite (3) as

$$\begin{aligned}&\mathcal{L}_{\text{rank}}\\&=\sum_{\{i,j:y_i=1,y_j=0\}}P(y_i=1,y_j=0)\mathbb{E}_{\mathbf{x}|y_i=1,y_j=0}\left[[[f_i<f_j]]+\frac{1}{2}[[f_i=f_j]]\right].\end{aligned} \tag{4}$$

Here, we define the mis-rank rate as

$$\begin{aligned}&R(i,j) = \mathbb{E}_{\mathbf{x}|y_i=1,y_j=0}\left[[[f_i<f_j]]+\frac{1}{2}[[f_i=f_j]]\right]\\&=P(f_i<f_j \mid y_i=1,\, y_j=0)+\frac{1}{2}P(f_i=f_j \mid y_i=1,y_j=0).\end{aligned} \tag{5}$$

The mis-rank rate $R(i,j)$ is the probability that the computed score of a sample for the positive label is smaller than that for the negative label. Using this, we express the rank loss $\mathcal{L}_{\text{rank}}$ as

$$\begin{aligned}\mathcal{L}_{\text{rank}} &= \sum_{\{i,j:y_i=1,y_j=0\}}P(y_i=1,y_j=0)R(i,j)\\&=\sum_{1\le i<j\le m}P(y_i=1,y_j=0)R(i,j)+P(y_i=0,y_j=1)R(j,i).\end{aligned} \tag{6}$$

We can consider rank loss $\mathcal{L}_{\text{rank}}$ as the expectation of the mis-rank rate $R(i,j)$ over all possible pairs of labels. Our interest is to minimize it in the PU setting.

4. Multi-label PU ranking

As described above, multi-label PU ranking is a problem of training a label-ranking-based multi-label classifier from a dataset, which has properties in which (1) labels assigned to samples are definitely positive and (2) absent labels are not necessarily negative.

In this section, we extend the analysis for binary PU classification [10] to a multi-label setting and derive the following:

1. A multi-label PU ranking problem can be cast as a cost-sensitive learning using positive and unlabeled data.
2. Applying a surrogate loss for optimization with incompletely labeled data leads to error from a correct one, which can be cancelled by selecting a symmetric surrogate loss function such as a ramp loss or a sigmoid loss.

4.1. Appropriately weighted cost

In this section, we explain how a multi-label PU ranking problem can be cast as a cost-sensitive learning, which means that we should weight the loss function appropriately. Cost-sensitive learning [14] is a type of learning method that incorporates the mis-classification cost. Cost-sensitive "ranking" using rank loss is given naturally as

$$\begin{aligned}&\mathcal{L}_{\text{rank}}\\&=\sum_{1\le i<j\le m}c_{ij}P(y_i=1,y_j=0)R(i,j)+c_{ji}P(y_i=0,y_j=1)R(j,i).\end{aligned} \tag{7}$$

where c_{ij} is the weight of the penalty for mis-ranking for the pair $y_i=1,y_j=0$. We aim at minimizing (6) with incompletely labeled samples. In our PU setting, the mis-rank rate $R(i,j)$ in (6) cannot be estimated directly from the data because there are no negative labels. For this reason, we introduce a pseudo-mis-rank rate $R_X(i,j)$, which can be estimated from the data, and then consider the optimization of (6) via this quantity. Here, we define the pseudo-mis-rank rate $R_X(i,j)$ as

$$\begin{aligned}&R_X(i,j)\\&=P(f_i<f_j \mid s_i=1,s_j=0)+\frac{1}{2}P(f_i=f_j \mid s_i=1,s_j=0).\end{aligned} \tag{8}$$

Therein, $s_i\in\{0,1\}$ shows whether the label of the i-th class is assigned or not. $s_i=1$ and $s_i=0$ respectively mean that the i-th class is labeled and not labeled. The pseudo-mis-rank rate $R_X(i,j)$ represents the possibility that the ranking score of a sample for an assigned label is smaller than that for an absent label. We can estimate this quantity from incompletely labeled data. If we ignore label incompleteness, then the expected loss as actually minimized is not (6) but

$$\begin{aligned}&\hat{\mathcal{L}}_{\text{rank}}\\&=\sum_{1\le i<j\le m}P(s_i=1,s_j=0)R_X(i,j)+P(s_i=0,s_j=1)R_X(j,i).\end{aligned} \tag{9}$$

Similarly to [10], we assume that unlabeled data are generated from a marginal distribution. This is called the "case-controlled" setting in PU classification [25]. This condition is expressed as $P(y_i=1|s_i=0)=P(y_i=1)$. Furthermore, we make the assumption that the deficit of a positive label is not biased on the sample space. This condition is represented as $P(\mathbf{x}|s_i{=}1){=}P(\mathbf{x}|y_i{=}1)$. By these assumptions, the pseudo-mis-rank rate $R_X(i,j)$ can be written using $R(i,j)$ as follows (see Appendix A):

$$R_X(i,j)=(1-\pi_{ij})R(i,j)+\pi_{ij}R_{\text{-}X}(i,j), \tag{10}$$

where

$$\begin{aligned}&R_{\text{-}X}(i,j)\\ &=P(f_i<f_j \mid y_i=1,y_j=1)+\frac{1}{2}P(f_i=f_j \mid y_i=1,y_j=1),\end{aligned} \tag{11}$$

and

$$\pi_{ij}=P(y_j=1 \mid y_i=1). \tag{12}$$

$R_{\text{-}X}(i,j)$ denotes the penalty imposed on mistakes attributable to the existence of positive labels contained in absent labels. To be more specific, in the case of learning from samples with incompletely assigned labels, rank loss imposes a penalty not for (positive, negative) but for (assigned, not assigned) label pairs. However, these pairs contain (positive, positive) pairs, which should not be included in the penalty. Here, we represent this excessively imposed penalty and its ratio as $R_{\text{-}X}(i,j)$ and π_{ij}, respectively. Therefore, (10) can be interpreted as a decomposition into two losses: the loss that should be given and the loss that should not be given. Transforming (10), we obtain

$$R(i,j)=\frac{1}{1-\pi_{ij}}(R_X(i,j)-\pi_{ij}R_{\text{-}X}(i,j)). \tag{13}$$

By substituting this into (6) and using the relation

$$\begin{aligned}&R_{\text{-}X}(i,j)+R_{\text{-}X}(j,i)\\ &=P(f_i>f_j \text{ or } f_i=f_j \text{ or } f_i<f_j \mid y_i=1,y_j=1)\\ &=1,\end{aligned} \tag{14}$$

we can obtain the following equation (see Appendix B):

$$\begin{aligned}&\mathcal{L}_{\text{rank}}\\ &=\sum P(y_i=1)R_X(i,j)+P(y_j=1)R_X(j,i)-P(y_i=1,y_j=1)\\ &=\sum c_{ij}P(s_i=1,s_j=0)R_X(i,j)+c_{ji}P(s_i=0,s_j=1)R_X(j,i)\\ &\quad -P(y_i=1,y_j=1).\end{aligned} \tag{15}$$

Here,

$$c_{ij}=\frac{P(y_i=1)}{P(s_i=1,s_j=0)}. \tag{16}$$

Compared with (9), the multi-label PU ranking problem can be cast as a cost-sensitive learning. In fact, minimizing the rank loss function weighted by c_{ij} with incompletely labeled data implies that the rank loss with completely labeled data, which should be originally minimized, can be minimized. $P(s_i{=}1,s_j{=}0)$ is the ratio of the samples in a dataset that meets the condition $s_i{=}1,s_j{=}0$. This can be estimated from the training dataset. In addition, $P(y_i{=}1)$ can be estimated using the methods proposed in [11], [2].

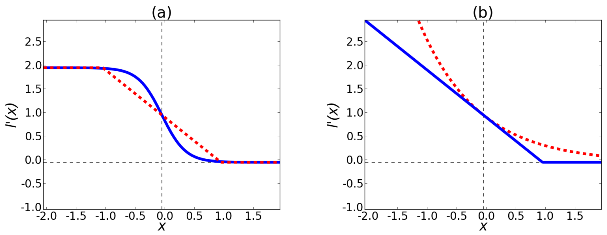

Figure 2. (a) Ramp loss (red dashed line) and sigmoid loss (blue solid line) meet the described condition, while (b) exponential loss (red dashed line) and hinge loss (blue solid line) do not.

4.2. Symmetric surrogate loss

In this section, we derive the condition for surrogate loss used in the optimization process. Mis-rank rate R can be written as the expectation of the 0-1 function over the sample space as described below:

$$R(i,j)=\mathbb{E}_{\mathbf{x}|y_i=1,y_j=0}[l_{0\text{-}1}(f_i-f_j)],\, l_{0\text{-}1}(x)=\begin{cases}1 \text{ (if } x<0)\\ \frac{1}{2} \text{ (if } x=0)\\ 0 \text{ (otherwise)}\end{cases}$$

Similarly,

$$R_X(i,j)=\mathbb{E}_{\mathbf{x}|s_i=1,s_j=0}[l_{0\text{-}1}(f_i-f_j)].$$

Because direct optimization of this expectation including the 0-1 loss is intractable, the surrogate loss function $l'(x)$ is generally used instead of the 0-1 loss. For example, the pseudo-mis-rank rate is written as follows by applying the hinge loss function l'_{H}, which is used in many popular models (e.g., support vector machines), as the surrogate loss.

$$R_X(i,j)\approx\mathbb{E}_{\mathbf{x}|s_i=1,s_j=0}[l'_{\text{H}}(f_i-f_j)],\, l'_{\text{H}}(x)=\begin{cases}1-x \text{ (if } x<1)\\ 0 \text{ (otherwise)}\end{cases}$$

When fully labeled data are used, by applying surrogate loss to rank loss (6), we obtain

$$\begin{aligned}&\mathcal{L}'_{\text{rank}}\\ &=\sum P(y_i=1,y_j=0)E_{\mathbf{x}|y_i=1,y_j=0}[l'(f_i-f_j)]\\ &\quad +P(y_i=0,y_j=1)E_{\mathbf{x}|y_i=0,y_j=1}[l'(f_j-f_i)].\end{aligned} \tag{17}$$

By replacing R_X in (15) with the expectation of surrogate loss,

$$
\begin{aligned}
&\mathcal{L}''_{\text{rank}}\\
=&\sum P(y_i{=}1)E_{\mathbf{x}|s_i=1,s_j=0}[l'(f_i{-}f_j)]\\
&+P(y_j{=}1)E_{\mathbf{x}|s_i=0,s_j=1}[l'(f_j{-}f_i)]-P(y_i{=}1,y_j{=}1)\\
=&\sum P(y_i{=}1,y_j{=}0)E_{\mathbf{x}|y_i=1,y_j=0}[l'(f_i{-}f_j)]\\
&+P(y_i{=}0,y_j{=}1)E_{\mathbf{x}|y_i=0,y_j=1}[l'(f_j{-}f_i)]\\
&-P(y_i{=}1,y_j{=}1)\big(1-E_{\mathbf{x}|y_i=1,y_j=1}[l'(f_i{-}f_j)+l'(f_j{-}f_i)]\big)
\end{aligned}
\tag{18}
$$

is obtained. From the two equations (17) and (18), we observe the relation $\mathcal{L}''_{\text{rank}}{=}\mathcal{L}'_{\text{rank}}{+}(\text{Error})$. It means that utilizing surrogate loss in the optimization process with incompletely labeled data causes an error. If we select the surrogate loss function $l'(\cdot)$ to meet $l'(f_i{-}f_j){+}l'(f_j{-}f_i){=}1$, then this error is cancelled and $\mathcal{L}''_{\text{rank}}{=}\mathcal{L}'_{\text{rank}}$ is obtained. Convex functions such as hinge loss and exponential loss (Fig. 2(b)) do not meet this condition. Symmetric non-convex functions such as ramp loss and sigmoid loss (Fig. 2(a)) satisfy it.

For example, let us consider a case in which data are completely labeled and separable. In other words, $\min \mathcal{L}_{\text{rank}}{=}0$. In this case,

$$\text{argmin } \mathcal{L}'_{\text{rank-hinge}}{=}\text{argmin } \mathcal{L}'_{\text{rank-ramp}}.$$

Completely identical hyperplanes are obtained for both the ramp loss and the hinge loss. Considering the PU setting, from the discussion presented above, we obtain

$$\text{argmin } \mathcal{L}''_{\text{rank-ramp}}{=}\text{argmin } \mathcal{L}'_{\text{rank-ramp}}.$$

On the other hand,

$$\text{argmin } \mathcal{L}''_{\text{rank-hinge}}{\neq}\text{argmin } \mathcal{L}'_{\text{rank-hinge}}$$

This relation indicates that the obtained boundary differs from the optimal one when using hinge loss.

5. Experiment

To investigate the efficacy of the conditions stated in the previous section, we conducted experiments on three datasets: synthetic dataset, MSCOCO [24], and NUS-WIDE [8].

5.1. Setting

Classifiers were trained from data containing a deficit on positive labels at a rate from 0% to 80%. The accuracy was evaluated on fully labeled data. The label deficit is given from the following steps:

1. Determine the total number of deficit labels N_{noise} by multiplying the deficit rate by the total number of positive labels.
2. Determine the number of deficit samples N^c_{noise} for class c from a multinomial distribution to meet $N_{\text{noise}}{=}\sum_c N^c_{\text{noise}}$.
3. Choose N^c_{noise} samples labeled as c at random and remove their label.

Table 1. Methods used in the experiments. Each method corresponds to whether condition 1 (use of weighted loss function) and condition 2 (use of symmetric loss function) are met or not.

	Baseline	Method 1	Method 2	Method 3 (proposed)
Condition 1 (weighted loss)		✓		✓
Condition 2 (symmetric loss)			✓	✓

We used the mean average precision in terms of samples as an evaluation criterion. Through all the experiments, hinge loss and ramp loss were used in the non-symmetric and symmetric surrogate loss function, respectively. Score functions were linear, and their weights were updated using stochastic gradient descent. Specifically for the score function $\mathbf{f}(\mathbf{x}){=}\mathbf{W}^T\mathbf{x}$ and loss function $\frac{1}{N}\sum_{n=1}^{N} l(\mathbf{f}(\mathbf{x}_n),\mathbf{y}_n)$, models were updated as

$$\mathbf{W}^{(t+1)}{=}\mathbf{W}^{(t)}{-}\eta^{(\tau)}\frac{\partial l(\mathbf{f}(\mathbf{x}_n),\mathbf{y_n})}{\partial \mathbf{W}},$$

where N denotes the number of training samples, and $\eta^{(\tau)}$ and τ denote the learning rate and the number of iterations, respectively. In this experiment, we reduced the learning rate as $\eta^{(\tau)}{=}\eta^{(0)}/\sqrt{\tau}$.

5.2. Synthetic dataset

A synthetic dataset was generated as in the following procedure, which was performed for each sample:

1. The number of relevant labels n is sampled from a Poisson distribution.
2. For n times, relevant class c is sampled from a multinomial distribution.
3. The number of feature samplings k is sampled from a Poisson distribution.
4. For k times, feature $\mathbf{x}$ is sampled from a multinomial distribution parametrized per class and their summation is used as a feature of the sample.

Ten thousand samples were generated and then divided into 8,000 for training and 2,000 for testing. The parameters of the multinomial distributions were sampled from a uniform distribution. In addition, L2 normalization was applied to all samples.

Experiment A. First, to show the influence of the derived condition on classification accuracy, we evaluated four methods, corresponding to whether condition 1 (use of appropriately weighted loss function) and condition 2 (use of

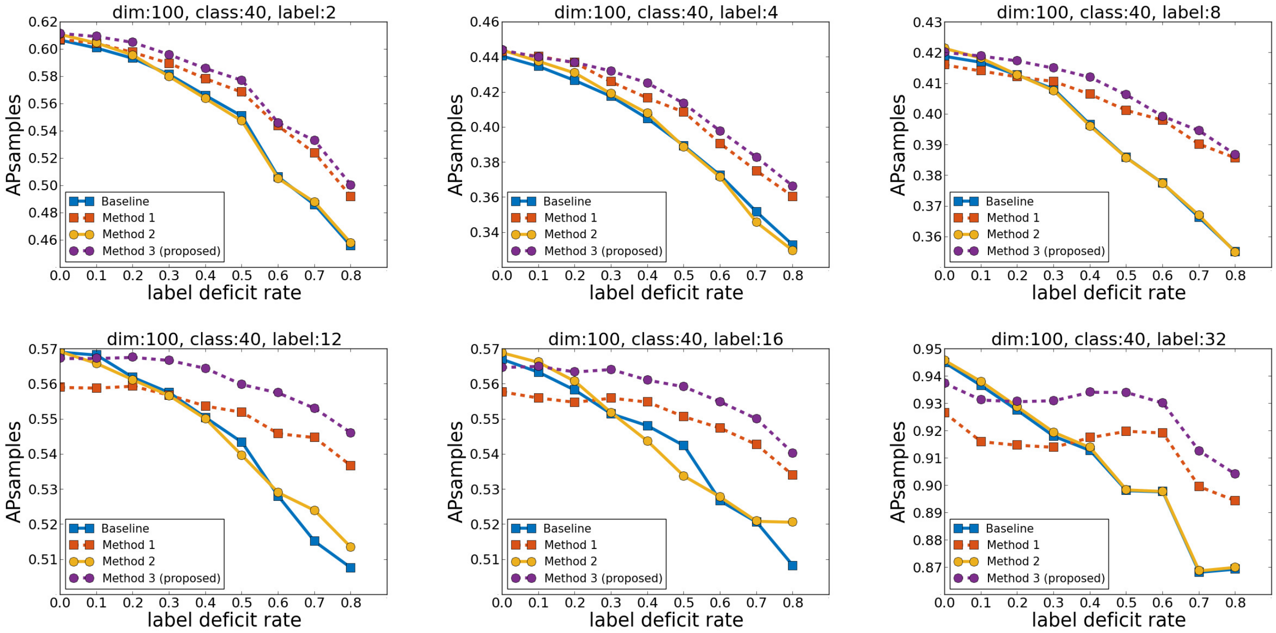

Figure 3. Results of Experiment A on a synthetic dataset. Each figure from the left corresponds to a different mean number of labels (2, 4, 8, 12, 16, 32). The method that meets both conditions has robustness to the label deficit.

symmetric loss function) are met or not (Table 1). In this experiment, the prior $P(y_i{=}1)$ in (16) was given from fully labeled training samples. We fixed the number of classes to 40 and the dimension of a feature to 100. The mean number of labels for each sample was chosen from (2,4,8,12,16,32). The result is presented in Fig. 3. In all settings, Method 3, which satisfies both conditions, could learn most robustly. As the number of labels increased, the difference in accuracy between Method 3 and the other methods became large.

Experiment B. Then, we evaluated the accuracy of Method 3 when combined with prior estimation, which is necessary for application to a real problem. To this end, we compared three methods: Method 1 (baseline), Method 3 with a prior given from a fully labeled dataset (denoted as optimal in result), and Method 3 with an estimated prior (denoted as estimated). To estimate class prior $P(y_i{=}1)$ for class i, we applied [11] based on distribution matching [12] for every class. Because we cannot know negative samples, this algorithm is designed to estimate prior $\theta{=}p(y{=}1)$ by partially matching marginal distribution $p(\mathbf{x})$ and class conditional distribution $p(\mathbf{x}|y{=}1)$ weighted by prior probability. To this end, the algorithm attempts to minimize the Pearson (PE) divergence in terms of θ. Formally, we obtain

$$\begin{aligned}\theta^*&{=}\mathrm{argmin}\ \mathrm{PE}\\ &{=}\mathrm{argmin}\frac{1}{2}\int\left(\frac{\theta p(\mathbf{x}|y{=}1)}{p(\mathbf{x})}{-}1\right)^2 p(\mathbf{x})\mathbf{dx}.\end{aligned} \tag{19}$$

Instead of a direct minimization of the PE divergence, the tightest lower bound is minimized. For more details, please see [11]. We changed the synthetic dataset from three points of view: (1) the number of classes, (2) the number of labels, and (3) the dimension of the features. Each setting is described as follows:

- Choose the number of labels from (2, 4, 8), while the number of classes and the dimension of the features are fixed respectively to 80 and 100.
- Choose the number of classes from (40, 80, 160), while the number of labels and the dimension of the features are fixed respectively to 4 and 100.
- Choose the dimension of the features from (50, 100, 150), while the number of classes and the number of labels are fixed respectively to 80 and 4.

The results of each setting correspond respectively to Fig. 4, Fig. 5, and Fig. 6. As shown in Fig. 4, when few labels were assigned for a fixed number of classes (2 labels out of 80 classes), and when the label deficit rate was low, the performance of the method with the estimated prior was high, sometimes even better than that with the optimal prior, and it worsened as the deficit rate increased. The result of the experiment using samples with more labels (4 labels out of 80 classes) showed that, although the performance was low when the label deficit rate was low, it was comparable to the method with the optimal prior as the deficit increased. If many labels were attached (8 labels out of 80 classes), the prior estimation failed and high accuracy could not be achieved. A similar tendency was observed from an experiment in which the number of classes varied (Fig. 5). These

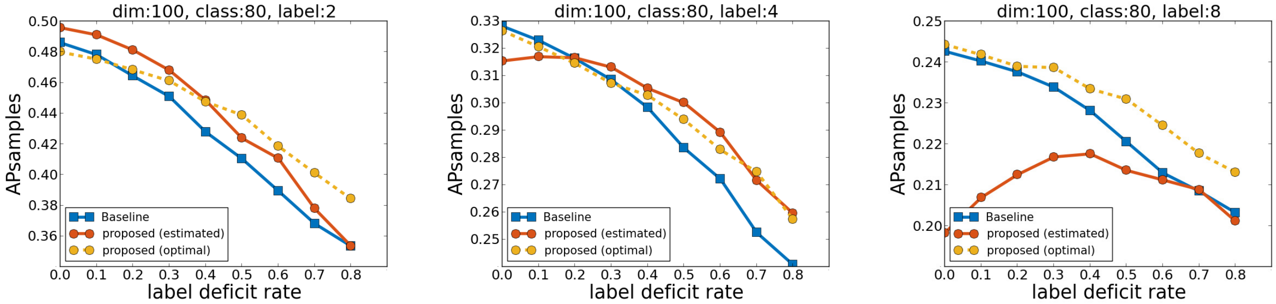

Figure 4. Results of Experiment B. Each figure from the left corresponds to a different mean number of labels (2, 4, 8).

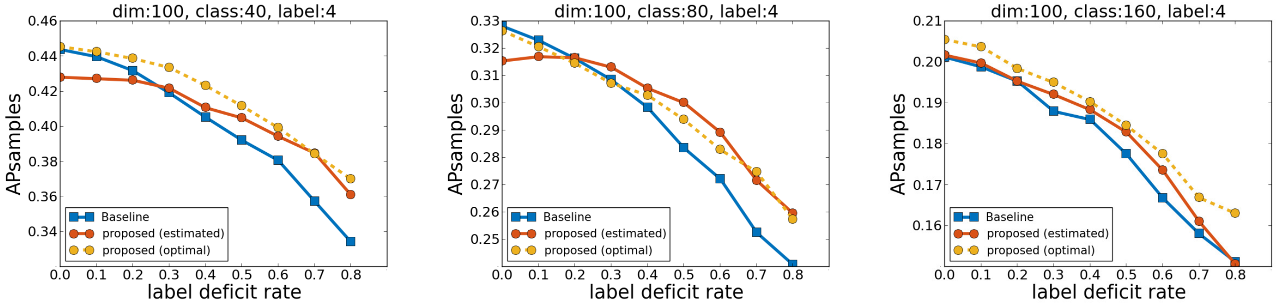

Figure 5. Results of Experiment B. Each figure from the left corresponds to a different number of classes (40, 80, 160).

results implied that the rate of (the number of labels)/(the number of classes) in the data is a key to the success of our method combined with prior estimation. From the results of the experiment of changing the dimension of the features shown in Fig. 6, it is apparent that the accuracy of the estimated method approached the optimal one as the dimensions increased.

5.3. Image Annotation Dataset

We conducted experiments on the image annotation datasets MSCOCO [24] and NUS-WIDE [8]. The settings of each experiment are given as follows:

MSCOCO: This dataset contains segmentation information and captions in addition to object labels. In this experiment, we used only images and object tags attached to them. We eliminated duplicated tags to a single image. We used 82,783 samples for training and 40,504 samples for testing. The number of classes was 80. The average number of labels was 2.95 per sample.

NUS-WIDE : This dataset includes Flickr images with 5,018 types of tags. We used 81 classes predetermined in the dataset and 161,789 samples for training and 107,859 samples for testing. The average number of labels was 1.76 per sample.

As the image feature, we used the activation of the 7th layer of AlexNet [21] pre-trained with the ILSVRC2012 dataset. The dimensions of the visual features were 4,096. We compared the same methods as those used in the previous experiment: Method 1 (baseline), Method 3 (optimal), and Method 3 (estimated). In these experiments, we estimated the prior using a naive method, which uses the ratio of a labeled sample per class in the training dataset. The experimental results for MSCOCO and NUS-WIDE are presented in Fig. 7 and Fig. 8, respectively. In both datasets, Method 3 with the estimated prior was able to improve slightly the accuracy in every deficit rate even though the prior estimation was naive.

6. Discussion

The results of Experiment A on the synthetic dataset showed that, as both (A) the label deficit rate and (B) the number of labels assigned to the samples increased, the difference in accuracy between the methods with condition 1 (Method 1 and Method 3) and those without it (baseline and Method 2) increased. This result can be explained using the following analysis. As provided in (9), let $\hat{\mathcal{L}}_{\mathrm{rank}}$ denote the loss function when we do not change the weight per class. The error from true loss $\mathcal{L}_{\mathrm{rank}} - \hat{\mathcal{L}}_{\mathrm{rank}}$ can be decomposed into the following: (a) A term proportional to the probability that samples are unlabeled even if they are positive. (b) A term proportional to the probability that both labels within a pair of labels are assigned (see Appendix C). (a) and (b) respectively correspond to (A) and (B).

Furthermore, if condition 1 is met, the difference in accuracy between the method with condition 2 and that without it becomes large when samples have many labels. That result derives from the error caused by surrogate loss in the PU setting data, where $\mathcal{L}''_{\mathrm{rank}} - \mathcal{L}'_{\mathrm{rank}}$ (provided in (17) and (18)) is proportional to the probability that both labels

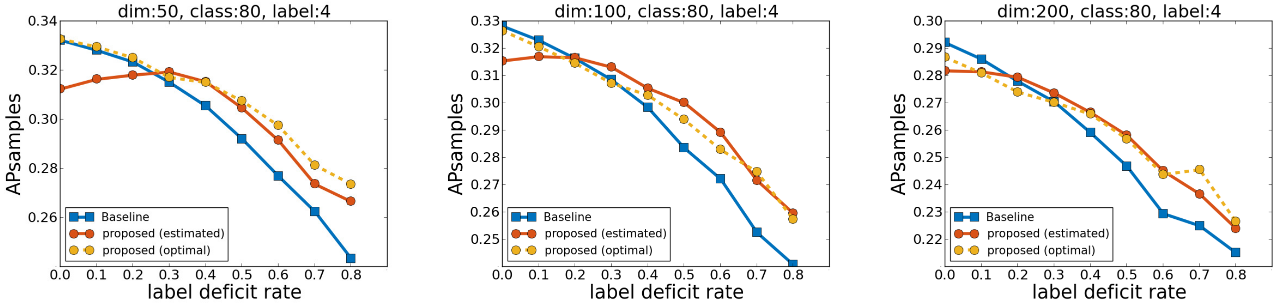

Figure 6. Results of Experiment B. Each figure from the left corresponds to a different number of features (50, 100, 150).

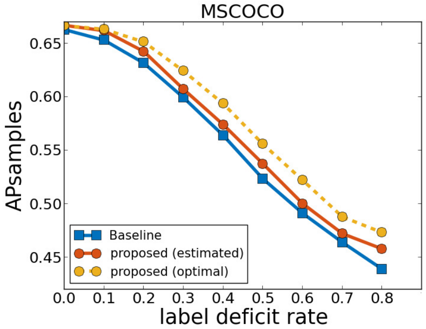

Figure 7. Experimental result for the MSCOCO dataset.

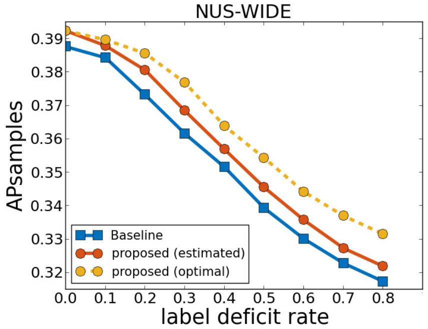

Figure 8. Experimental result for the NUS-WIDE dataset.

are positive from every pair.

The reason for the low performance of Method 1, even when the label deficit rate was low, can be attributed to the fact that an error occurs when the “case-controlled” condition is not satisfied, which we assumed to derive the conditions. In contrast to Method 1, Method 3 still showed high performance because the error was low when symmetric loss was used. Particularly, when a label’s deficit rate is 0%, or the labels are assigned completely, this error can be cancelled using symmetric loss (see Appendix D).

In many situations with Experiment B on the synthetic dataset, the accuracy near a 0% deficit rate was low because the prior estimation method we used did not assume that the labels were complete. Our method works when the label deficit rate is greater than about 20%, which is likely in realistic data. To treat both complete and incomplete labels using the same method, it is necessary to invent more flexible prior estimation methods, which is a subject for our future work. We conjecture that the reason for the prior estimation failing when numerous labels were attached is that positive samples and negative samples for one class tend to share other class’s label, which causes low separability in feature space. For a task of classifying (dog, cat, human), if we collect images in which they exist together with high probability, the sample labeled as (cat, human) and (dog, cat, human) might have similar features. The rate of such a sample in a dataset increases as the number of labels increases for a fixed number of classes. It is extremely difficult to estimate the prior from such low-separability data. The performance of a method with prior estimation that sometimes overcomes the optimal one can be thought to estimate the distribution from which data are generated more accurately than taking a ratio of the samples in a dataset without a label deficit.

In the experiments on image annotation datasets, we estimated the class prior in a naive manner because the computational costs of existing algorithms are high. Despite its simplicity, our method slightly improved the accuracy, but an efficient estimation algorithm is needed for more improvement when applied to large-scale data.

7. Conclusion

In this paper, we specifically examined the training of a multi-label classifier from incompletely labeled data, which is essential for multi-label training. Regarding this problem as a multi-label PU classification problem, we extended the binary PU classification to label-ranking-based multi-label learning. By analyzing this problem, we derived two conditions for training classifiers consistently even if only parts of the relevant labels are obtained: (1) use of appropriately weighted loss function and (2) use of symmetric surrogate loss. We conducted experiments on several datasets and also demonstrated the efficacy of these conditions.

8. Acknowledgment

This work was supported by CREST, JST. The authors would like to thank Prof. Masashi Sugiyama (The University of Tokyo) for the useful discussions.

References

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *CVPR, 2013*.

[2] G. Blanchard, G. Lee, and C. Scott. Semi-supervised novelty detection. *The Journal of Machine Learning Research*, 11:2973–3009, 2010.

[3] S. S. Bucak, R. Jin, and A. K. Jain. Multi-label learning with incomplete class assignments. In *CVPR, 2011*.

[4] S. S. Bucak, P. K. Mallapragada, R. Jin, and A. K. Jain. Efficient multi-label ranking for multi-class learning: application to object recognition. In *ICCV, 2009*.

[5] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *PAMI*, 29(3):394–410, 2007.

[6] Y.-N. Chen and H.-T. Lin. Feature-aware label space dimension reduction for multi-label classification. In *NIPS, 2012*.

[7] W. Cheng, E. Hüllermeier, and K. J. Dembczynski. Bayes optimal multilabel classification via probabilistic classifier chains. In *ICML, 2010*.

[8] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. Nus-wide: a real-world web image database from national university of singapore. In *CIVR, 2009*.

[9] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7:551–585, 2006.

[10] M. C. Du Plessis, G. Niu, and M. Sugiyama. Analysis of learning from positive and unlabeled data. In *NIPS, 2014*, pages 703–711.

[11] M. C. Du Plessis and M. Sugiyama. Class prior estimation from positive and unlabeled data. *IEICE TRANSACTIONS on Information and Systems*, 97(5):1358–1362, 2014.

[12] M. C. Du Plessis and M. Sugiyama. Semi-supervised learning of class balance under class-prior change by distribution matching. *Neural Networks*, 50:110–119, 2014.

[13] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *NIPS, 2001*.

[14] C. Elkan. The foundations of cost-sensitive learning. In *IJCAI, 2001*.

[15] C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *SIGKDD, 2008*.

[16] W. Gao and Z.-H. Zhou. On the consistency of multi-label learning. *Artificial Intelligence*, 199:22–44, 2013.

[17] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe. Deep convolutional ranking for multilabel image annotation. *arXiv preprint arXiv:1312.4894*, 2013.

[18] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *ICCV, 2009*.

[19] A. Kapoor, R. Viswanathan, and P. Jain. Multilabel classification using bayesian compressed sensing. In *NIPS, 2012*.

[20] X. Kong, Z. Wu, L.-J. Li, R. Zhang, P. S. Yu, H. Wu, and W. Fan. Large-scale multi-label learning with incomplete label assignments. In *SIAM, 2014*.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS, 2012*.

[22] W. S. Lee and B. Liu. Learning with positive and unlabeled examples using weighted logistic regression. In *ICML, 2003*.

[23] X. Li, F. Zhao, and Y. Guo. Conditional restricted boltzmann machines for multi-label learning with incomplete labels. In *AISTATS, 2015*.

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV, 2014*.

[25] A. Menon, B. V. Rooyen, C. S. Ong, and B. Williamson. Learning from corrupted binary labels via class-probability estimation. In *ICML, 2015*.

[26] L. Von Ahn and L. Dabbish. Labeling images with a computer game. In *CHI, 2004*.

[27] J. Weston, S. Bengio, and N. Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *IJCAI, 2011*.

[28] S.-J. Yang, Y. Jiang, and Z.-H. Zhou. Multi-instance multi-label learning with weak label. In *IJCAI, 2013*.

A.

From two conditions $P(y_i{=}1|s_i{=}0){=}P(y_i{=}1)$ and $P(\mathbf{x}|s_i{=}1){=}P(\mathbf{x}|y_i{=}1)$,

$$\begin{aligned}
&P(f_i{<}f_j \mid s_i{=}1,s_j{=}0)\\
&{=}P(f_i{<}f_j \mid y_i{=}1,s_j{=}0)\\
&{=}P(y_j{=}0 \mid y_i{=}1,s_j{=}0)P(f_i{<}f_j \mid y_i{=}1,y_j{=}0,s_j{=}0)\\
&{+}P(y_j{=}1 \mid y_i{=}1,s_j{=}0)P(f_i{<}f_j \mid y_i{=}1,y_j{=}1,s_j{=}0)\\
&{=}P(y_j{=}0 \mid y_i{=}1)P(f_i{<}f_j \mid y_i{=}1,y_j{=}0)\\
&{+}P(y_j{=}1 \mid y_i{=}1)P(f_i{<}f_j \mid y_i{=}1,y_j{=}1).
\end{aligned} \tag{20}$$

Similarly,

$$\begin{aligned}
&\frac{1}{2}P(f_i{=}f_j \mid s_i{=}1,s_j{=}0)\\
&{=}\frac{1}{2}P(y_j{=}0 \mid y_i{=}1)P(f_i{=}f_j \mid y_i{=}1,y_j{=}0)\\
&{+}\frac{1}{2}P(y_j{=}1 \mid y_i{=}1)P(f_i{=}f_j \mid y_i{=}1,y_j{=}1).
\end{aligned} \tag{21}$$

Combining them, we obtain

$$\begin{aligned}
&R_X(i,j)\\
&{=}P(f_i{<}f_j \mid s_i{=}1,s_j{=}0){+}\frac{1}{2}P(f_i{=}f_j \mid s_i{=}1,s_j{=}0)\\
&{=}(1{-}\pi_{ij})R(i,j){+}\pi_{ij}R_{-X}(i,j).
\end{aligned} \tag{22}$$

B.

From (13),

$$\begin{aligned}
&P(y_i{=}1,y_j{=}0)R(i,j)\\
&{=}\frac{P(y_i{=}1,y_j{=}0)}{P(y_j{=}0|y_i{=}1)}\Big\{R_X(i,j){-}P(y_j{=}1|y_i{=}1)R_{-X}(i,j)\Big\}\\
&{=}P(y_i{=}1)R_X(i,j){-}P(y_i{=}1,y_j{=}1)R_{-X}(i,j),
\end{aligned} \tag{23}$$

Plugging the equation above and (14) into (6), we obtain

$$\begin{aligned}
&\mathcal{L}_{\text{rank}}\\
&= \sum_{1\le i<j\le m} P(y_i{=}1,y_j{=}0)R(i,j){+}P(y_i{=}0,y_j{=}1)R(j,i)\\
&= \sum_{1\le i<j\le m} P(y_i{=}1)R_X(i,j){+}P(y_j{=}1)R_X(j,i)\\
&\quad{-}P(y_i{=}1,y_j{=}1)\Big\{R_{-X}(i,j){+}R_{-X}(j,i)\Big\}\\
&= \sum_{1\le i<j\le m} P(y_i{=}1)R_X(i,j){+}P(y_j{=}1)R_X(j,i)\\
&\quad{-}P(y_i{=}1,y_j{=}1),
\end{aligned} \tag{24}$$

C.

Because

$$\begin{aligned}
&P(y_i{=}1){-}P(s_i{=}1,s_j{=}0)\\
&{=}P(y_i{=}1,s_i{=}0){+}P(y_i{=}1,s_i{=}1){-}P(y_i{=}1,s_i{=}1,s_j{=}0)\\
&{=}P(y_i{=}1,s_i{=}0){+}P(s_i{=}1,s_j{=}1),
\end{aligned} \tag{25}$$

from (15) and (9), we obtain

$$\begin{aligned}
&\mathcal{L}_{\text{rank}}{-}\hat{\mathcal{L}}_{\text{rank}}\\
&= \sum_{1\le i<j\le m} P(y_i{=}1)R_X(i,j){+}P(y_j{=}1)R_X(j,i){-}\text{const}\\
&- \sum_{1\le i<j\le m} P(s_i{=}1,s_j{=}0)R_X(i,j){+}P(s_i{=}0,s_j{=}1)R_X(j,i)\\
&= \sum_{1\le i<j\le m} P(s_i{=}1,s_j{=}1)\Big\{R_X(i,j){+}R_X(j,i)\Big\}\\
&\quad{+}P(y_i{=}1,s_i{=}0)R_X(i,j){+}P(y_j{=}1,s_j{=}0)R_X(j,i)\\
&\quad{+}\text{const}.
\end{aligned} \tag{26}$$

The first term is proportional to the ratio of samples having both i-th and j-th label. Second and third terms are proportional to ratio of samples, which is not labeled even if they are positive.

D.

When labels are given completely, the loss function which should be minimized is (6),

$$\begin{aligned}
&\mathcal{L}_{\text{rank}}\\
&= \sum_{1\le i<j\le m} P(y_i{=}1,y_j{=}0)R(i,j){+}P(y_i{=}0,y_j{=}1)R(j,i).
\end{aligned} \tag{27}$$

However, the loss function derived based on "case-controlled" assumption is (15)

$$\begin{aligned}
&\mathcal{L}_{\text{rank}}\\
&{=}\sum P(y_i{=}1)R_X(i,j){+}P(y_j{=}1)R_X(j,i){-}P(y_i{=}1,y_j{=}1)
\end{aligned} \tag{28}$$

Considering the case in which label deficit does not exist, because $R(i,j){=}R_X(i,j)$,

$$\begin{aligned}
&\mathcal{L}_{\text{rank-false}}\\
&{=}\sum P(y_i{=}1)R(i,j){+}P(y_j{=}1)R(j,i){-}P(y_i{=}1,y_j{=}1).
\end{aligned} \tag{29}$$

Their mutual difference is

$$\begin{aligned}
&\mathcal{L}_{\text{rank-false}}{-}\mathcal{L}_{\text{rank}}\\
&{=}\sum P(y_i{=}1,y_j{=}1)\Big\{R(i,j){+}R(j,i){-}1\Big\}.
\end{aligned} \tag{30}$$

This error is proportional to the joint probability that both i-th class and j-th class are positive. However, this error is cancelled if we use symmetric surrogate loss.