Machine Learning (2020) 109:2369–2388
https://doi.org/10.1007/s10994-020-05915-2

Binary classification with ambiguous training data

Naoya Otani[1] · **Yosuke Otsubo**[1] · **Tetsuya Koike**[1] · **Masashi Sugiyama**[2,3]

Received: 16 April 2020 / Revised: 29 July 2020 / Accepted: 19 September 2020 /
Published online: 3 November 2020

Abstract
In supervised learning, we often face with *ambiguous* (A) samples that are difficult to label even by domain experts. In this paper, we consider a binary classification problem in the presence of such A samples. This problem is substantially different from semi-supervised learning since unlabeled samples are not necessarily difficult samples. Also, it is different from 3-class classification with the positive (P), negative (N), and A classes since we do not want to classify test samples into the A class. Our proposed method extends binary classification with reject option, which trains a classifier and a rejector simultaneously using P and N samples based on the 0-1-c loss with rejection cost c. More specifically, we propose to train a classifier and a rejector under the 0-1-c-d loss using P, N, and A samples, where d is the misclassification penalty for ambiguous samples. In our practical implementation, we use a convex upper bound of the 0-1-c-d loss for computational tractability. Numerical experiments demonstrate that our method can successfully utilize the additional information brought by such A training data.

Keywords Ambiguous samples · Classification with reject option · Binary classification

1 Introduction

Supervised learning has been successfully deployed in various real-world applications, such as medical diagnosis (Bar et al. 2015; Wang et al. 2016; Esteva et al. 2017) and manufacturing systems (Park et al. 2016; Ren et al. 2017). However, when the amount of labeled data is limited, current supervised learning methods still do not work reliably (Pesapane et al. 2018).

To efficiently obtain labeled data, domain knowledge has been used in many application areas (Ren et al. 2017; Cruciani et al. 2018; Konishi et al. 2019; Bejnordi et al.

Editors: Kee-Eung Kim and Vineeth N.Balasubramanian

✉ Naoya Otani
Naoya.Otani@nikon.com

[1] Nikon Corporation, Research and Development Division, 471, Nagaodai-cho, Sakae-ku, Yokohama-city, Kanagawa 244-8533, Japan

[2] RIKEN Center for Advanced Intelligence Project, Tokyo, Japan

[3] The University of Tokyo, Graduate School of Frontier Sciences, Chiba, Japan

2017). However, as some studies have pointed out (Wagner et al. 2005; Li et al. 2016; Shahriyar et al. 2018), there are often *ambiguous* samples that are substantially difficult to label even by domain experts.

The goal of this paper is to propose a novel classification method that can handle such ambiguous data. More specifically, we consider a binary classification problem where, in addition to positive (P) and negative (N) samples, ambiguous (A) samples are available for training a classifier.

Naively, we may consider employing 3-class classification methods for the P, N, and A classes. However, since we classify test samples only in the P or N class, not in the A class, naive 3-class methods cannot be directly used in our problem. Moreover, they cannot utilize the information that the A class exists between the P and N classes. Another related approach is classification with reject option (Bartlett and Wegkamp 2008; Cortes et al. 2016), where ambiguous test samples are not classified into the P or N classes, but rejected. However, classification methods with reject option do not consider ambiguous samples in the training phase and thus they cannot be employed in the current scenario.

Semi-supervised learning may also be related to the current problem, where unlabeled data is used for training a classifier in addition to P and N data (Odena 2016; Sakai et al. 2017). In semi-supervised learning, unlabeled samples are P and N samples that have not yet been labeled and they are not necessarily difficult samples to be labeled. On the other hand, in our target problem of classification with ambiguous data, ambiguous data are typically distributed in the intersection of the P and N classes. Thus, since the problem setups are intrinsically different, merely using semi-supervised learning methods in the current problem may not be promising.

Classification with imperfect labeling (Cannings et al. 2020) allows incorrect labels in training data, so it can be useful to deal with a dataset where annotators forcibly give positive or negative labels to all samples. Open-set classification (Scheirer et al. 2012) detects samples classified into none of the training classes and classifies them into the "unknown" class, so we can apply it when we deal with a dataset where annotators skip labeling hard-to-label samples and we classify test samples into the P, N or A class. However, those two approaches are still different from our problem setting in terms of labels in input and output data. Table 1 summerizes the problem setting of different approaches.

To effectively solve the problem of classification with ambiguous data, we propose to extend classification with reject option that trains a classifier and a rejector simultaneously using P and N samples based on the 0-1-c loss with rejection cost c (Cortes et al. 2016). Our proposed method trains a classifier and a rejector under the 0-1-c-d loss using P, N, and A samples, where d is the misclassification penalty for ambiguous samples. Then, in the test phase, we use the trained classifier to assign P or N labels to test samples. However, in the same way as the 0-1-c loss, directly performing optimization with the 0-1-c-d loss is cumbersome due to its discrete nature. To cope with this problem, we introduce a convex upper bound of the 0-1-c-d loss and use it in our practical implementation. Through experiments, we demonstrate that the proposed method can improve the test classification accuracy by utilizing A samples in the training phase. We also consider a simple heuristic that we randomly relabel ambiguous samples into the positive or negative class and apply classification with reject option. We show that the heuristic is essentially equivalent to a special case of the proposed method, and thus it can be an easy-to-implement alternative to the proposed method.

The rest of this paper is organized as follows. We briefly review supervised learning in Sect. 2. Then, we define our problem setting and describe the details of our proposed

Table 1 Problem settings of related and our methods

Methods	Labels in training data	Labels predicted in test phase	Relationship among classes
Binary classification	Positive	Positive	None
	Negative	Negative	
3-class classification	Class 1	Class 1	None
	Class 2	Class 2	
	Class 3	Class 3	
Classification with reject option	Positive	Positive	Rejected samples are in P/N mixed regions
	Negative	Rejected	
		Negative	
Semi-supervised learning	Positive	Positive	Unlabeled samples belong to P or N
	Unlabeled	Negative	
	Negative		
Classification with imperfect labeling	Positive	Positive	Training data can contain incorrect labels
	Negative	Negative	
Open-set classification	Positive	Positive	Test data can contain neither positive nor negative samples
	Negative	Negative	
		Unknown	
Our proposal	Positive	Positive	Ambiguous samples are in P/N mixed regions
	Ambiguous	Negative	
	Negative		

method in Sect. 3 . In Sect. 4, we experimentally evaluate its performance. Finally, in Sect. 5, we summarize our contributions and describe future works.

2 Supervised classification

In this section, we first define the standard supervised classification problem and then review its standard solution.

2.1 Formulation

Let $x \in \mathcal{X}$ be an input point and $y \in \mathcal{Y} = \{1, -1\}$ denote a binary label, which corresponds to the positive and negative classes, respectively. Suppose that we are given a set of positive and negative samples $\{(x_i, y_i)\}_{i=1}^N$ drawn independently from the probability distribution with density $p(x, y)$ defined on $\mathcal{X} \times \mathcal{Y}$. Let $h : \mathcal{X} \to \mathbb{R}$ denote a discriminant function, with which a class label is predicted for test input point x as $\hat{y} = \mathrm{sign}(h(x))$.

The goal is to train the discriminant function h so that the expected misclassification rate is minimized. Let us define the 0-1 loss as

$$L_{01}(h, x, y) = 1_{yh(x) \leq 0}, \tag{1}$$

where 1_A is the indicator function that takes 1 if statement A is true and 0 otherwise. Then, we can express this problem as

$$\begin{aligned} h^* &= \operatorname*{argmin}_h R(h), \\ R(h) &= \mathbb{E}_{p(x,y)}\left[L_{01}(h, x, y)\right], \end{aligned}$$

where h^* denotes the optimal discriminant function and $\mathbb{E}_{p(x,y)}$ denotes the expectation over $p(x, y)$. In practice, since we do not know the true density $p(x, y)$, we usually use the empirical distribution to approximate the expectation:

$$\hat{R}(h) = \frac{1}{N} \sum_{i=1}^{N} L_{01}(h, x_i, y_i). \tag{2}$$

Based on Eqs. (1) and (2), we can formulate various classification methods depending on loss functions (Sugiyama 2015). In the rest of this section, we introduce the support vector machine (SVM) (Vapnik 1995), which is one of the most basic algorithms of binary classification.

2.2 Support vector machine (SVM)

Because optimization with $L_{01}(h, x, y)$ is computationally intractable, it is not practical to optimize the empirical risk $\hat{R}(h)$ directly. To overcome this problem, the hinge loss, an upper bound of $L_{01}(h, x, y)$ called the hinge loss, defined by

$$L_{\mathrm{H}}(h, x, y) = \max(1 - yh(x), 0), \tag{3}$$

has been introduced as its surrogate. Since the hinge loss is convex, optimization can be reduced to a convex program. Further introducing the L2 regularization, basis functions

$\phi_1(x), \ldots, \phi_N(x)$, and slack variables $\xi = (\xi_1, \ldots, \xi_N)^\top$ with $^\top$ being the transpose, the following quadratic program can be obtained as a dual optimization problem:

$$(\hat{w}, \hat{\xi}) = \underset{(w,\xi)}{\operatorname{argmin}} \left[\frac{\lambda}{2}\|w\|^2 + \frac{1}{N}\sum_{i=1}^{N} \xi_i \right] \quad \text{s.t.} \begin{pmatrix} \xi_i \geq 1 - y_i h_i \\ \xi_i \geq 0 \end{pmatrix} \text{for } i = 1, \ldots, N, \tag{4}$$

where $w = (w_1, \ldots, w_N)^\top$ are the coefficients of the discriminant function, $\lambda > 0$ is the L2 regularization parameter, and h_i is the value of the discriminant function at sample point x_i given by $h_i = \sum_{j=1}^{N} w_j \phi_j(x_i)$. The resulting discriminant function is given by $h(x;\hat{w}) = \sum_{j=1}^{N} \hat{w}_j \phi_j(x)$.

3 Classification with ambiguous data

In this section, we formulate our target problem called *classification with ambiguous data* (CAD) and propose a new method for solving the CAD.

3.1 Formulation

We consider three class labels, i.e,. positive, ambiguous, and negative: $y \in \mathcal{Y}_0 = \{1, 0, -1\}$. Suppose that we are given a set of positive, ambiguous, and negative samples $\{(x_i, y_i)\}_{i=1}^{N}$ drawn independently from the probability distribution with density $p_0(x, y)$ defined on $\mathcal{X} \times \mathcal{Y}_0$. Our goal is still to learn a discriminant function that classifies test samples into either the positive or negative class (not in the ambiguous class). Our key question in this scenario is if we can utilize the ambiguous training data to improve the classification accuracy of the discriminant function.

In this section, we develop a new method based on a method of *classification with reject option* (CRO) (Cortes et al. 2016). For this reason, before deriving the new method, we first review the CRO method.

3.2 Classification with reject option by SVM (CRO-SVM)

Cortes et al. (2016) introduced a rejection function $r : \mathcal{X} \to \mathbb{R}$ to identify the regions with high risk for misclassification, in addition to discriminating the positive and negative classes. When the rejection function takes a positive value, the corresponding sample is accepted and is classified into the positive or negative class by classifier h; otherwise, the sample is rejected and is not classified. When a sample is rejected, the rejection cost c is incurred, which trades off the risk of misclassification and the cost of rejection. To realize this idea, the *0-1-c loss* was introduced:

$$L_{01c}(h, r, x, y) = 1_{yh(x)\leq 0} 1_{r(x)>0} + c 1_{r(x)\leq 0}. \tag{5}$$

When $c = 0$, all samples are rejected because the loss function does not incur any cost. On the other hand, when $c \geq 0.5$, no samples are rejected because the expectation of the 0-1 loss is less than 0.5; in that case, the 0-1-c loss is reduced to the 0-1 loss. Therefore,

effectively, we only consider c such that $0 < c < 0.5$. The rejection function and the discriminant function are simultaneously learned from training data.

Similarly to the 0-1 loss, the 0-1-c loss has discrete nature and thus its direct optimization is computationally intractable. To avoid the discontinuity, the following surrogate loss called the *max-hinge (MH) loss* was introduced:

$$L_{\mathrm{MH}}(h, r, x, y) = \max\left(1 + \frac{\alpha}{2}(r(x) - yh(x)), c(1 - \beta r(x)), 0\right), \tag{6}$$

where $\alpha, \beta > 0$ are the hyperparameters to control the shape of the surrogate loss.

In the same manner as the original SVM, introducing the L2 regularization, basis functions, and slack variables yields the following quadratic program:

$$\begin{aligned}(\hat{w}, \hat{u}, \hat{\xi}) = \underset{(w,u,\xi)}{\operatorname{argmin}} & \left[\frac{\lambda}{2}\|w\|^2 + \frac{\lambda'}{2}\|u\|^2 + \frac{1}{N}\sum_{i=1}^{N}\xi_i\right] \\ \text{s.t.} & \begin{pmatrix} \xi_i \geq 1 + \frac{\alpha}{2}(r_i - y_i h_i) \\ \xi_i \geq c(1 - \beta r_i) \\ \xi_i \geq 0 \end{pmatrix} \text{for } i = 1, \ldots, N,\end{aligned} \tag{7}$$

where $w = (w_1, \ldots, w_N)^\top$ are the coefficients of the discriminant function, $u = (u_1, \ldots, u_N)^\top$ are the coefficients of the rejection function, $\lambda, \lambda' > 0$ are the L2 regularization parameters, and h_i and r_i denote the values of the discriminant function and rejection function at sample point x_i given by $h_i = \sum_{j=1}^{N} w_j \phi_j(x_i)$ and $r_i = \sum_{j=1}^{N} u_j \phi_j(x_i)$, respectively. The resulting discriminant function and rejection function are given by $h(x;\hat{w}) = \sum_{j=1}^{N} \hat{w}_j \phi_j(x)$ and $r(x;\hat{u}) = \sum_{j=1}^{N} \hat{u}_j \phi_j(x)$.

We refer to this method as CRO-SVM.

3.3 Proposed method: classification with ambiguous data by SVM (CAD-SVM)

To handle ambiguous training data in the SVM formulation, we extend the 0-1-c loss to the *0-1-c-d loss* defined as

$$L_{01\mathrm{cd}}(h, r, x, y) = 1_{y^2=1}\left(1_{yh(x)\leq 0} 1_{r(x)>0} + c 1_{r(x)\leq 0}\right) + d 1_{y=0} 1_{r(x)>0}. \tag{8}$$

Tables 2 and 3 compare the behavior of the 0-1-c loss and the 0-1-c-d loss. For positive and negative samples, the 0-1-c-d loss behaves the same as the 0-1-c loss. On the other hand, for ambiguous samples, the 0-1-c-d loss incurs penalty d when they are classified into the positive or negative class. Therefore, ambiguous samples tend to be classified into

Table 2 The 0-1-c loss function

Label y	Judgement (h, r)		
	Positive	Rejected	Negative
	$h > 0$	$r \leq 0$	$h \leq 0$
	$r > 0$		$r > 0$
Positive $y = 1$	0	c	1
Negative $y = -1$	1	c	0

Table 3 The 0-1-*c*-*d* loss function

Label y	Judgement (h, r)		
	Positive	Ambiguous	Negative
	$h > 0$	$r \leq 0$	$h \leq 0$
	$r > 0$		$r > 0$
Positive $y = 1$	0	c	1
Ambiguous $y = 0$	d	0	d
Negative $y = -1$	1	c	0

the ambiguous class if we employ the 0-1-*c*-*d* loss. Compared to the CRO formulation, where a rejector cannot be learned explicitly from positive and negative samples, the CAD utilizes ambiguous samples to learn a rejector explicitly.

The above discussion may mislead us as if we are just solving a 3-class problem with the positive, ambiguous, and negative classes. However, we do not classify test samples into the ambiguous class, but only into the positive and negative classes. To solve the CAD problem, we utilize a binary discriminant function h and a rejection function r, as in the CRO formulation reviewed above. More specifically, we train h and r under the 0-1-*c*-*d* loss, and we only use h in the test phase to classify test samples into the positive and negative classes. Thanks to the interplay between h and r in the 0-1-*c*-*d* loss, we can utilize ambiguous data to train h through r.

Similarly to the 0-1-*c* loss, we consider the following convex upper bound of the 0-1-*c*-*d* loss called the *max-hinge-ambiguous (MHA) loss* as a surrogate to avoid its discrete nature:

$$\begin{aligned} L_{01\mathrm{cd}}(h, r, x, y) &\leq 1_{y^2=1} L_{\mathrm{MH}}(h, r, x, y) + d 1_{y=0} \max(1 + \beta r(x), 0) \\ &= y^2 \max\left(1 + \frac{\alpha}{2}(r(x) - y h(x)), c(1 - \beta r(x)), 0\right) \\ &\quad + (1 - y^2) \max(d(1 + \beta r(x)), 0) \\ &\leq y^2 \max\left(1 + \frac{\alpha}{2}(r(x) - y h(x)), \eta c(1 - \beta r(x)), 0\right) \\ &\quad + (1 - y^2) \max(\eta d(1 + \beta r(x)), 0) \\ &\equiv L_{\mathrm{MHA}}(h, r, x, y), \end{aligned} \tag{9}$$

where $\eta \geq 1$ is the hyperparameter to control the shape of the surrogate loss. See Fig. 1 for its visualization.

Then, in the same way as the CRO-SVM, we have the following quadratic program:

$$\begin{aligned} (\hat{w}, \hat{u}, \hat{\xi}) = \underset{(w,u,\xi)}{\operatorname{argmin}} & \left[\frac{\lambda}{2}\|w\|^2 + \frac{\lambda'}{2}\|u\|^2 + \frac{1}{N}\sum_{i=1}^{N} \xi_i\right] \\ \text{s.t.} & \begin{pmatrix} \xi_i \geq y_i^2\left(1 + \frac{\alpha}{2}(r_i - y_i h_i)\right) \\ \xi_i \geq y_i^2 \eta c(1 - \beta r_i) \\ \xi_i \geq (1 - y_i^2)\eta d(1 + \beta r_i) \end{pmatrix} \quad \text{for } i = 1, \ldots, N. \end{aligned} \tag{10}$$

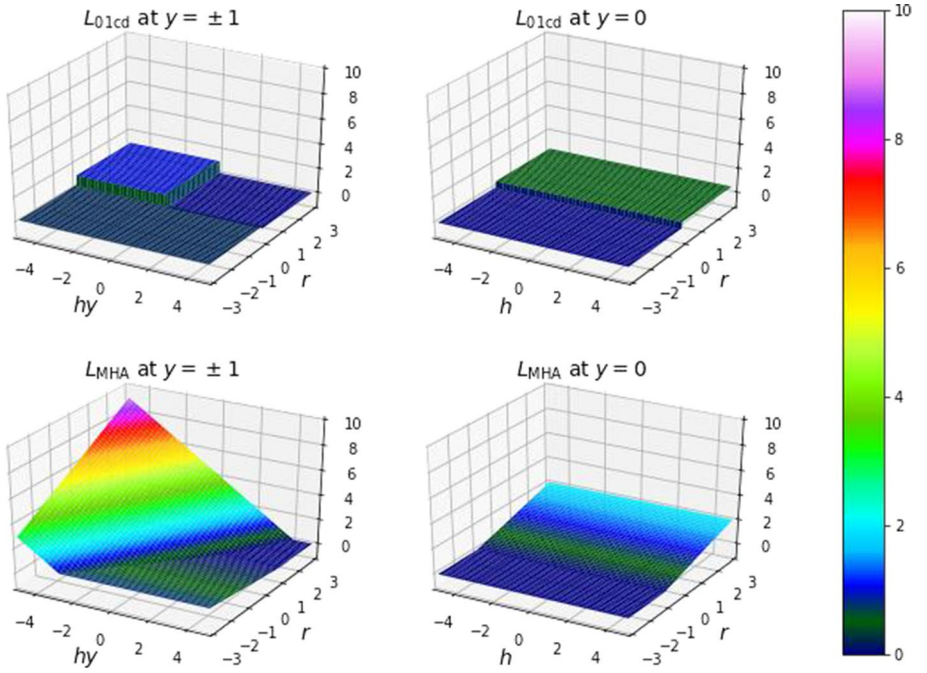

Fig. 1 The 0-1-c-d loss L_{01cd} and its surrogate loss L_{MHA} for the penalty values $(c, d) = (0.2, 0.5)$

This is our proposed method called CAD-SVM. The computational complexity of the CAD-SVM depends on implementation of the quadratic program. It naively costs $O(N^3)$, but if we use a fixed number of basis functions, the complexity reduces to $O(N)$.

The MHA loss depends on the choice of hyperparameters (α, β, η). To find good hyperparameter values, let us analyze the 0-1-c-d loss first.

For each $x \in \mathcal{X}$, let $\pi_+(x) = p_0(y = 1|x)$, $\pi_0(x) = p_0(y = 0|x)$, and $\pi_-(x) = p_0(y = -1|x)$, where $\pi_+(x) + \pi_0(x) + \pi_-(x) = 1$. Then the following lemma shows how c and d are related to $\pi_+(x)$ and $\pi_-(x)$ for the optimal classifier and rejector (its proof is available in Appendix 1):

Lemma 1 *For each $x \in \mathcal{X}$, let*

$$\left(h^*_{01cd}, r^*_{01cd}\right) = \operatorname*{argmin}_{(h,r)} \mathbb{E}_{p_0(y|x)}\left[L_{01cd}(h, r, x, y)\right]. \tag{11}$$

Then

$$\begin{cases} \operatorname{sign}(h^*_{01cd}) = 1, \ \operatorname{sign}(r^*_{01cd}) = 1 & \text{if } \pi_+ \geq \dfrac{d + (1 - c - d)\pi_-}{c + d}, \\ \operatorname{sign}(h^*_{01cd}) = -1, \ \operatorname{sign}(r^*_{01cd}) = 1 & \text{if } \pi_- \geq \dfrac{d + (1 - c - d)\pi_+}{c + d}, \\ \operatorname{sign}(r^*_{01cd}) = -1 & \text{otherwise.} \end{cases} \tag{12}$$

Figure 2 illustrates the above results. This shows that when π_+ (or π_-) is large (i.e., imbalanced classification), the rejector accepts the sample and the classifier classifies

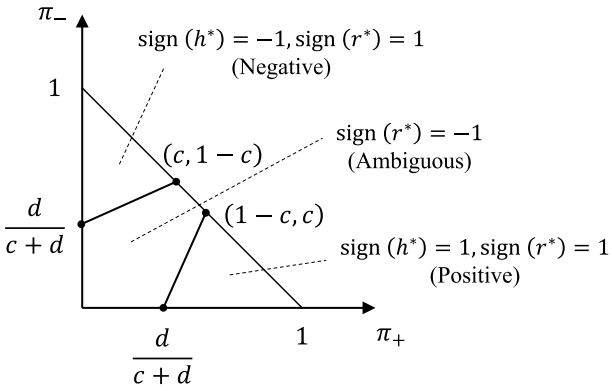

Fig. 2 Optimal solutions for the 0-1-c-d loss

that sample into the positive (or negative) class. On the other hand, when both π_+ and π_- are not so large, the rejector rejects the sample.

Next, based on the above lemma, we have the following theorem for the MHA loss (its proof is given in Appendix 2):

Theorem 1 *For each $x \in \mathcal{X}$, let*

$$\left(h^*_{\mathrm{MHA}}, r^*_{\mathrm{MHA}}\right) = \operatorname*{argmin}_{(h,r)} \mathbb{E}_{p_0(y|x)}\left[L_{\mathrm{MHA}}(h, r, x, y)\right]. \tag{13}$$

Then, for

$$\alpha^* = 2(1-2c), \quad \beta^* = 1+2c, \quad \eta^* = \frac{2}{1+2c}, \tag{14}$$

the signs of $(h^*_{\mathrm{MHA}}, r^*_{\mathrm{MHA}})$ match those of $(h^*_{\mathrm{01cd}}, r^*_{\mathrm{01cd}})$.

Based on the above theorem, we use Eq. (14) as hyperparameter values in our experiments in the next section and demonstrate that they work well in practice. Nevertheless, given that the above theorem is valid only for the optimal solutions, we may cross-validate better hyperparameter values around Eq. (14) to further improve the classification performance. Note that Eq. (14) does not include d.

4 Numerical experiments

In this section, we report experimental results.

4.1 Datasets

For experiments, we use a toy dataset, a public dataset, and an in-house dataset.

4.1.1 Toy dataset

To understand the behavior of our method and related methods, we created a toy classification problem and applied the methods to it. The problem contains three regions: the positive, negative, and mixed regions (see Fig. 3). The positive and negative regions are clearly separable, whereas the mixed region has no good discriminant function. For this problem, we want to clearly discriminate the positive and negative regions, with the influence of the mixed region avoided. We also studied how the proportion of ambiguous samples influences the results by changing the proportion of ambiguous samples in the mixed region r. The number of all samples is 400, the total number of positive and negative samples in the separable regions is 200, and the total number of positive and negative samples in the mixed region is $200(1-r)$. Therefore, the expected maximum accuracy is $\{1+(1-r)\times 0.5\}/\{1+(1-r)\}$.

4.1.2 Public datasets (PD1, PD2, PD3)

We processed a regression dataset, the Boston Housing Dataset (Harrison and Rubinfeld 1978), to convert it to a classification dataset with ambiguous data. The original dataset consists of 13 features $\xi_i \in \mathbb{R}^{13}$ associated with the average house prices $\zeta_i \in \mathbb{R}$ for 506 districts, where $1 \le i \le 506$ denotes the sample number. We annotated all the samples to be positive, negative, and ambiguous according to the following procedure:

PD1 (P/N/A separable): Simply, the samples with $\zeta_i > 23$ were labeled as positive, the samples with $\zeta_i < 19$ were labeled as negative, and the other samples were labeled as ambiguous. The numbers of samples were 190, 173, and 143 for the positive, negative, and ambiguous classes, respectively.
PD2 (P/N/A mixed): The samples with $\zeta_i > 23$ and the samples with $\zeta_i < 19$ were still labeled as positive and negative, respectively, but the remaining 143 samples were randomly labeled as positive, negative, or ambiguous.
PD3 (separable and mixed): We considered a hyperplane $v \cdot \xi = 0$ in the feature space and divided all the samples into two parts, the mixed part $\{i | v \cdot \xi_i \ge 0\}$ and the separable part $\{i | v \cdot \xi_i < 0\}$. The coefficients of the hyperplane v are selected so that the averages

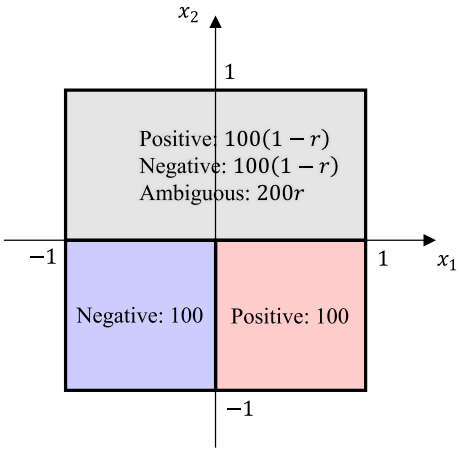

Fig. 3 Toy dataset. The lower-left and lower-right regions are negative and positive regions, while the upper region is the mixed region. The numbers indicate the numbers of samples in each region. For each region, samples are distributed uniformly

of ζ_i over the samples in both parts were approximately matched. For the mixed part, each sample is randomly labeled as positive, negative or ambiguous, whereas for the separable part, the samples with $\zeta_i > 21$ were labeled as positive and the others were labeled as negative. The numbers of samples in the separable region were 83, 0, and 87, and those in the mixed region were 114, 107, and 115 for the positive, negative, and ambiguous classes, respectively.

Figure 4a–c and e–g show the 2-dimensional plots of the above datasets visualized by the principle component analysis (PCA) and the locality preserving projection (LPP) (He and Niyogi 2004). On the whole, ambiguous points are located between positive and negative points.

4.1.3 In-house dataset from a cell culture process (ID)

As a real-world application, we prepared an in-house cell culture dataset. This dataset contained 124 fields of view (FOV). For each FOV, 2 images were acquired: one was in the middle, and the other was at the end of the culturing process. Each middle image was analyzed by the image processing software, CL-Quant (Alworth et al. 2010), and converted to 8 morphological features such as the average brightness and average area of cells.

Each final image was annotated by experts. If the cells in the image overall looked healthy/damaged, the image was labeled as positive/negative. However, some images contained both healthy and damaged cells, and they were labeled as ambiguous. The numbers of samples were 41, 59, and 24 for positive, negative, and ambiguous, respectively.

Our motivation was to predict the final state of each FOV annotated by the experts, using morphological features obtained in the middle of the culturing process. If we could predict it accurately, the culturing cost would be saved by aborting the culturing process where the cells would be damaged while keeping the healthy cells cultured. Therefore, we should focus on reducing misclassification between the positive and negative classes and we hope that information from ambiguous samples could be useful to improve the

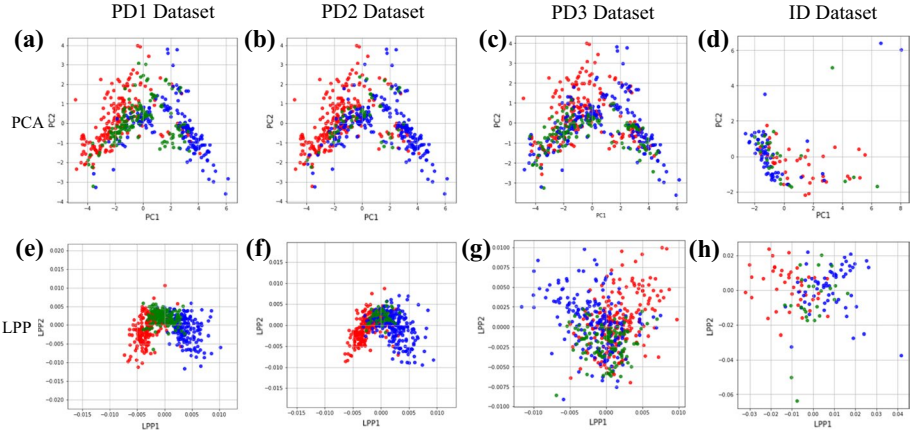

Fig. 4 Two-dimensional visualization of the PD1, PD2, PD3, and ID datasets by the PCA and the LPP. Red, blue, and green points correspond to positive, negative and ambiguous samples, respectively (Color figure online)

prediction accuracy. That was why test samples did not have the ambiguous label and were not classified into the ambiguous class in our scenario. Though ambiguous samples may occur in the test phase in the actual curturing process, we did not care which class they were classified into.

In the same manner as the PD1, PD2, and PD3 datasets, this dataset was also visualized by the PCA and the LPP in Fig. 4d and h, respectively. They show that ambiguous points are roughly located between positive and negative points.

4.2 Experimental settings

Using the above datasets, we compared the classification performance of the SVM, the SVM-RL (random label), the LapSVM (Belkin et al. 2006), the two-step SVM, the CRO-SVM, the CRO-SVM-RL, and the CAD-SVM. The SVM-RL employs the SVM algorithm, but ambiguous samples are randomly relabeled as positive or negative, which effectively utilizes information of the ambiguous samples. The LapSVM is a semi-supervised learning method based on the SVM, which employs a regularization term defined by the graph Laplacian. Ambiguous samples are treated as unlabeled samples in the LapSVM. The two-step SVM is the method which learns the rejection function and the discriminant function sequentially—first the rejection function is learned to judges whether the sample is ambiguous or not, and then the discriminant function is learned only using the samples which are not rejected by the rejection function. When the rejection function is learned, class weights c and d are applied to non-ambiguous (i.e., positive and negative) and ambiguous samples, respectively. The CRO-SVM-RL is the CRO-SVM with random labels for ambiguous samples in the same manner as SVM-RL.

For each method, 500 test runs were performed by changing the training and test datasets which were randomly divided from the original dataset. The dividing ratio of the training and test dataset was 4:1 for the ID dataset and 1:2 for the other datasets. For each test run, 5-fold cross validation was performed to determine the parameters below. For validation and in the test phase, only positive and negative samples were applied to the discriminant function, thus we was able to evaluate the binary classification accuracy.

Note that our goal was to maximize the binary classification accuracy in the test phase, and that was equivalent to minimizing the expected 0-1 loss. Though we trained the models using various loss functions such as the hinge loss, the MH loss, and the MHA loss, the 0-1 loss was minimized by cross validation.

In total, we had 10 hyperparameters $(\lambda, \lambda', \sigma, \sigma', \tau, c, d, \alpha, \beta, \eta)$, where (λ, λ') were the L2 regularization parameters, σ was the width of the Gaussian radial basis function in the basis functions $\phi_i(x) = \exp\left(-\frac{\|x-x_i\|^2}{2\sigma^2}\right)$, σ' was the hyperparameter of the weight matrix W of the graph Laplacian as $W_{ij} = \exp\left(-\frac{\|x_i-x_j\|^2}{2\sigma'^2}\right)$ (only in the LapSVM), τ was the coefficient of the graph Laplacian regularization (only in the LapSVM), (c, d) were the hyperparameters of the 0-1-c loss (in the CRO-SVM and CRO-SVM-RL), the 0-1-c-d loss (in the CAD-SVM) or the class weights (in the two-step SVM), and (α, β, η) were the hyperparameters of the MH and MHA loss functions (in the CRO-SVM, CRO-SVM-RL and CAD-SVM). We used 5-fold cross validation in terms of the classification accuracy to choose the hyperparameters from $(\lambda, \lambda') \in \{10^{-3}, 10^{-5}, 10^{-7}\}$, $(\sigma, \sigma') \in \{10^{0.5}, 10^{0.75}, 10^{1}\}$, $\tau \in \{10^{-1}, 10^{-2}, 10^{-3}\}$, $c \in \{0.03, 0.06, 0.20, 0.45\}$, and $d \in \{0.03, 0.06, 0.20, 0.50\}$. The other hyperparameters (α, β, η) were determined by Eq. (14). Quadratic programming problems were solved using cvxopt (Andersen et al. 2018).

4.3 Results

Figure 5 shows an example of the discriminant results for the toy dataset with $r = 0.5$. The upper half of the domain was the mixed region of positive and negative samples and samples in that region are expected to be classified into the ambiguous class. On the other hand, the lower half of the domain was perfectly separable into the positive and negative regions and samples in those regions are expected to be discriminated accurately (see Fig. 3). The SVM made some misclassifications in the lower half of the domain, which were caused by the mixed region. The SVM-RL and LapSVM could not reduce the number of misclassifications, though they utilized information of ambiguous samples. This suggests that ambiguous samples should not be simply relabeled to positive or negative samples or should not be treated as unlabeled samples. The two-step SVM could not also reduce the number of misclassifications and it could not learn the ambiguous region. Thus, it would be disadvantageous to learn the ambiguous region by combining positive and negative classes. The CRO-SVM and CRO-SVM-RL successfully learned the ambiguous region but that did not lead to learning more accurate discriminant functions. The CAD-SVM also successfully learned the ambiguous region and then it was able to learn a more accurate discriminant function. Overall, the CAD-SVM was shown to be the most appropriate method in this toy experiment.

Tables 4 shows the test accuracy of the toy dataset by changing the parameters r which are the propotion of ambiguous samples in the mixed region. When the proportion of the ambiguous samples was small, the proposed method did not work effectively since the number of ambiguous samples was too small. On the other hand, when the proportion of the ambiguous samples was large, it also did not show better performance. This was because, under this condition, only small numbers of positive and negative samples existed in the mixed region. Therefore, methods which do not utilize ambiguous samples showed reasonably good enough performance. However, when the proportion of ambiguous samples was $r = 0.5$, the CAD-SVM achieved a better score than the other methods.

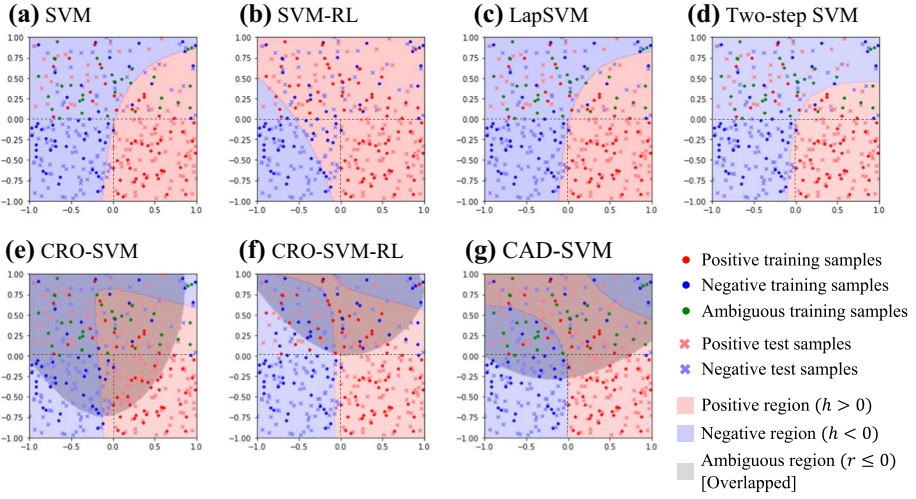

Fig. 5 An example of discriminant results for the toy dataset

Table 4 Test accuracy for the toy dataset by changing the ratio r of ambiguous samples in the mixed region

r	0.1	0.3	0.5	0.7	0.9
SVM	**0.739**	0.767	0.814	0.853	0.931
SVM-RL	**0.739**	0.766	0.812	0.850	0.926
LapSVM	**0.739**	0.766	0.814	0.850	0.930
Two-step SVM	**0.737**	**0.771**	0.816	0.854	**0.935**
CRO-SVM	0.736	0.767	**0.820**	0.857	0.931
CRO-SVM-RL	0.735	0.768	0.817	**0.861**	0.926
CAD-SVM	0.736	0.767	**0.822**	0.857	0.932

The boldface numbers show the best and equivalent results with 5% t-test

Table 5 summarizes the test accuracy of each method for the PD1, PD2, PD3, and ID datasets, respectively. For the PD1 dataset, the CAD-SVM was not superior to other methods since this condition was similar to the toy dataset with higher r. For the PD2 dataset, though the dataset had a mixed region, the CAD-SVM also did not show good performance. However, for the PD3 dataset, which had a mixed region and a separable region, the CAD-SVM achieved the best performance among the compared methods. It is suggested that our proposed method, the CAD-SVM, works effectively when the dataset has both of a mixed region and a separable boundary between the positive and negative classes. The CAD-SVM utilizes ambiguous samples to learn a rejector which rejects mixed regions, thus it would be able to focus on learning a separable region. The CRO-SVM and the CRO-SVM-RL also learn a rejector from positive and negative samples, but the CAD-SVM would be superior since the CAD-SVM can explicitly learn a rejector using ambiguous samples. For the ID dataset, the CAD-SVM performed better than the other method, except for the CRO-SVM-RL. Overall, ambiguous samples can improve the binary classification accuracy under some conditions, and the CAD-SVM is one of the solutions that can utilize such ambiguous samples.

4.4 Discussions

Through all the experiments, the CRO-SVM-RL gave as good performance as the CAD-SVM. As detailed in Appendix 3, we can show the following relations between the CRO-SVM-RL and the CAD-SVM:

Table 5 Test accuracy for the PD1, PD2, PD3, and ID datasets, where ± denotes the standard deviation

	PD1	PD2	PD3	ID
SVM	**0.924 ± 0.019**	**0.828 ± 0.019**	0.635 ± 0.029	0.803 ± 0.089
SVM-RL	0.918 ± 0.021	**0.827 ± 0.020**	0.629 ± 0.030	0.806 ± 0.088
LapSVM	0.921 ± 0.020	0.824 ± 0.021	0.629 ± 0.031	0.802 ± 0.088
Two-step SVM	0.918 ± 0.024	0.823 ± 0.021	0.632 ± 0.030	0.790 ± 0.091
CRO-SVM	**0.922 ± 0.020**	**0.827 ± 0.020**	0.635 ± 0.029	**0.812 ± 0.090**
CRO-SVM-RL	0.917 ± 0.026	**0.828 ± 0.022**	0.632 ± 0.029	**0.820 ± 0.086**
CAD-SVM	0.921 ± 0.020	**0.828 ± 0.019**	**0.639 ± 0.028**	**0.818 ± 0.089**

The boldface numbers show the best and equivalent results with 5% t-test

1. The 0-1-c loss function for the randomly labeled (RL) dataset, in which we randomly relabeled ambiguous samples as positive or negative, reduces to the 0-1-c-d loss with $d = \frac{1}{2} - c$.
2. For the RL dataset, the MH loss can be regarded as a surrogate loss of the 0-1-c-d loss with $d = \frac{1}{2} - c$, and it is calibrated under the conditions of Eq. (14).

Thus, though it is a simple heuristic, the CRO-SVM-RL is essentially equivalent to the CAD-SVM except for that the hyperparameter d is fixed to $\frac{1}{2} - c$. In practice, the CRO-SVM-RL is easier to implement and thus it may be used as an alternative to the CAD-SVM. However, we note that the CRO-SVM-RL alone does not provide rich theoretical insights that we have shown in Sect. 3.

Our goal for each experiment was minimizing the expected 0-1 loss in the test phase. Therefore, the SVM was naively an appropreate method since the hinge loss was calibrated to the 0-1 loss. Nevertheless, though the CRO-SVM-RL and the CAD-SVM minimized a surrogate of the 0-1-c-d loss in the training phase, they were able to achieve the better accuracy than the SVM in the test phase. It is suggested that the providing a reject option and incorporating ambiguous training samples could work as a kind of regularization, but further studies will be needed to clarify its mathematical properties.

We note that ambiguous samples are intrinsically hard-to-label samples even by experts, so they usually contain little information for classifying positive and negative samples, and thus we cannot expect large improvement to the binary classification accuracy. Nevertheless, our proposed method achieved statistically significant improvements for some cases.

5 Conclusion

In this study, we aimed to reduce the labeling cost and improve the classification accuracy by allowing labelers to give "ambiguous" labels for difficult samples. We extended a method of classification with reject option and proposed a novel classification method named the CAD-SVM that uses the 0-1-c-d loss. We derived a surrogate loss for the 0-1-c-d loss, which allowed us to convert the optimization problem into a convex quadratic program. We carried out numerical experiments and showed that ambiguous labels can be effectively used to improve the classification accuracy. We also showed that the CRO-SVM-RL, in which we randomly relabeled ambiguous samples to be positive or negative and applied classification with reject option, can be a practical alternative to the proposed method since it is essentially equivalent to the proposed method.

Though our proposed method was based on the SVM, it would be more useful if it can be applied to other models especially deep neural networks. However, further experimental studies will be needed to confirm if a naive application of the proposed MHA loss works well in practice. Indeed, it is known that changing models can cause other problems such as overfitting (Kiryo et al. 2017). Moreover, for deep neural networks, though we usually use the softmax cross entropy as the loss function, even the 0-1-c loss function has not been extended to the softmax cross entropy. So, analyzing the influence of changing loss functions is also an important issue to be further investigated.

In addition to the experimental analysis with more complex models, our future study will conduct theoretical analysis of the proposed method such as statistical consistency and the rate of convergence. Extending the proposed loss function to semi-supervised

problems, imperfect labeling problems or multi-class problems is also a promising direction to be pursued.

Acknowledgements The authors would like to thank Yasujiro Kiyota and Momotaro Ishikawa for providing the in-house dataset.

Appendix 1: Proof of Lemma 1

We calculate the expectation value of $L_{01\mathrm{cd}}$ as follows:

$$\begin{aligned}
&\mathbb{E}_{y\sim p_0(y|x)}[L_{01\mathrm{cd}}(h,r,x,y)] \\
&\quad = \pi_+ L_{01\mathrm{cd}}(h,r,x,+1) + \pi_0 L_{01\mathrm{cd}}(h,r,x,0) + \pi_- L_{01\mathrm{cd}}(h,r,x,-1) \\
&\quad = \begin{cases} d\pi_0 + \pi_- & \text{if } \operatorname{sign}(h) = 1, \operatorname{sign}(r) = 1, \\ d\pi_0 + \pi_+ & \text{if } \operatorname{sign}(h) = -1, \operatorname{sign}(r) = 1, \\ c(\pi_+ + \pi_-) & \text{otherwise.} \end{cases}
\end{aligned} \tag{15}$$

Then, the minimum of the expectation value is

$$\begin{aligned}
&\min_{(h,r)} \mathbb{E}_{y\sim p_0(y|x)}[L_{01\mathrm{cd}}(h,r,x,y)] \\
&\quad = \min\left(d\pi_0 + \pi_-, d\pi_0 + \pi_+, c(\pi_+ + \pi_-)\right) \\
&\quad = \begin{cases} d\pi_0 + \pi_- & \text{if } \pi_+ \geq \frac{d+(1-c-d)\pi_-}{c+d}, \\ d\pi_0 + \pi_+ & \text{if } \pi_- \geq \frac{d+(1-c-d)\pi_+}{c+d}, \\ c(\pi_+ + \pi_-) & \text{otherwise.} \end{cases}
\end{aligned} \tag{16}$$

From the comparison of Eqs. (15) and (16), the optimal (h, r) subject to (π_+, π_-) are determined. □

Appendix 2: Proof of Theorem 1

At the condition of Eq. (14), the expectation value of the loss is

$$\begin{aligned}
&\mathbb{E}_{y\sim p_0(y|x)}[L_{\mathrm{MHA}}(h,r,x,y)] \\
&\quad = \pi_+ \max\left(1 + (1-2c)(r(x) - h(x)), \frac{2c}{1+2c} - 2cr(x), 0\right) \\
&\qquad + \pi_- \max\left(1 + (1-2c)(r(x) + h(x)), \frac{2c}{1+2c} - 2cr(x), 0\right) \\
&\qquad + \pi_0 \max\left(\frac{2d}{1+2c} + 2dr(x), 0\right).
\end{aligned} \tag{17}$$

To find the minimum of the expectation value, we derive a linear programming problem considering $r(x)$, $h(x)$ as independent variables. As shown in Fig. 6, we can calculate the boundary conditions subject to (h, r) as,

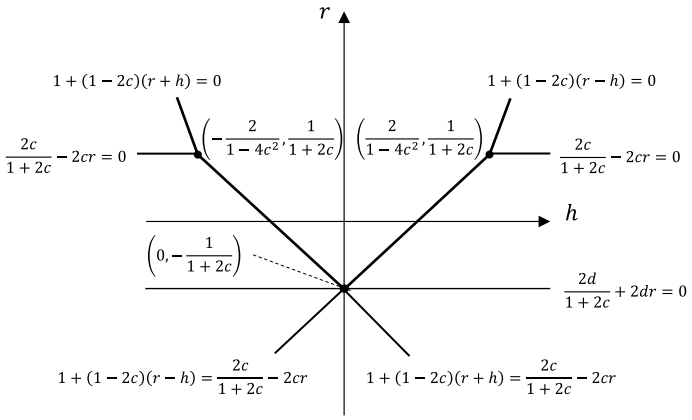

Fig. 6 The boundary conditions in the linear programming problem of Eq. (17)

$$\mathbb{E}_{y\sim p_0(y|x)}[L_{\mathrm{MHA}}(h,r,x,y)] = \begin{cases} \frac{4}{1+2c}(d\pi_0+\pi_-) & \text{if } (h,r) = \left(\frac{2}{1-4c^2}, \frac{1}{1+2c}\right), \\ \frac{4}{1+2c}(d\pi_0+\pi_+) & \text{if } (h,r) = \left(-\frac{2}{1-4c^2}, \frac{1}{1+2c}\right), \\ \frac{4}{1+2c}c(\pi_+ + \pi_-) & \text{if } (h,r) = \left(0, -\frac{1}{1+2c}\right). \end{cases} \tag{18}$$

Thus, we determine the minimizers of (h, r) subject to (π_+, π_-).

$$(h^*_{\mathrm{MHA}}, r^*_{\mathrm{MHA}}) = \underset{(h,r)}{\operatorname{argmin}}\, \mathbb{E}_{y\sim \mathrm{Pr}_0(y|x)}[L_{\mathrm{MHA}}(h,r,x,y)], \tag{19}$$

$$\begin{cases} h^*_{\mathrm{MHA}} = \frac{2}{1-4c^2} > 0, \quad r^*_{\mathrm{MHA}} = \frac{1}{1+2c} > 0 & \text{if } \pi_+ \geq \frac{d+(1-c-d)\pi_-}{c+d}, \\ h^*_{\mathrm{MHA}} = -\frac{2}{1-4c^2} < 0, \quad r^*_{\mathrm{MHA}} = \frac{1}{1+2c} > 0 & \text{if } \pi_- \geq \frac{d+(1-c-d)\pi_+}{c+d}, \\ h^*_{\mathrm{MHA}} = 0, \quad r^*_{\mathrm{MHA}} = -\frac{1}{1+2c} < 0 & \text{otherwise.} \end{cases} \tag{20}$$

□

Appendix 3: Relation between the CRO-SVM-RL and the CAD-SVM

For a dataset which contains positive, negative and ambiguous samples, we define a RL (randomly labeled) dataset as a dataset in which ambiguous samples are randomly relabeled into the positive or negative classes.

Theorem 2 *The risk of the* 0-1-c *loss function for the RL dataset is equal to the risk of the* 0-1-c-d *loss with* $d = \frac{1}{2} - c$ *for the original dataset.*

Proof Let a relabeled label be $z \in \{-1, 1\}$ which satisfies

$$\Pr(z = 1|x) = \pi_+(x) + \tfrac{1}{2}\pi_0(x), \qquad \Pr(z = -1|x) = \pi_-(x) + \tfrac{1}{2}\pi_0(x). \tag{21}$$

Thus, we can calculate the risk of the 0-1-c loss for the relabeled label z as

$$\begin{aligned}
&\mathbb{E}_{z\sim \Pr(z|x)}[L_{01c}(h,r,x,z)] \\
&\quad = \left(\pi_+(x) + \tfrac{1}{2}\pi_0(x)\right)(1_{h<0}1_{r\geq 0} + c1_{r<0}) + \left(\pi_-(x) + \tfrac{1}{2}\pi_0(x)\right)(1_{h>0}1_{r\geq 0} + c1_{r<0}) \\
&\quad = \pi_+(x)(1_{h<0}1_{r\geq 0} + c1_{r<0}) + \pi_-(x)(1_{h>0}1_{r\geq 0} + c1_{r<0}) + \pi_0(x)\left(\tfrac{1}{2} - c\right)1_{r\geq 0} + c\pi_0(x) \\
&\quad = \mathbb{E}_{y\sim p_0(y|x)}\left[L_{01cd}(h,r,x,y)\big|_{d=\frac{1}{2}-c}\right] + \pi_0(x)c.
\end{aligned} \tag{22}$$

Therefore, the risk is equal to that of the 0-1-c-d loss for the original label y. Note that the second term in the right-hand side is constant with respect to h and r. □

Theorem 3 *The MH loss for the RL dataset is convex and an upper bound of the* 0-1-c-d *loss with* $d = \frac{1}{2} - c$ *for the original dataset.*

Proof For the ambiguous label, we can calculate the expectation value of the MH loss function over the relabeling process as

$$\begin{aligned}
&\mathbb{E}_{z\sim \Pr(z|x,y=0)}[L_{MH}(h,r,x,z)] \\
&\quad = \tfrac{1}{2}L_{MH}(h,r,x,z=1) + \tfrac{1}{2}L_{MH}(h,r,x,z=-1) \\
&\quad = \tfrac{1}{2}\max\left(1 + \tfrac{\alpha}{2}(r(x) - h(x)), c(1-\beta r(x)), 0\right) \\
&\qquad + \tfrac{1}{2}\max\left(1 + \tfrac{\alpha}{2}(r(x) + h(x)), c(1-\beta r(x)), 0\right) \\
&\quad \geq L_{01cd}(h,r,x,y=0).
\end{aligned} \tag{23}$$

Since $L_{MH}(h,r,x,z=\pm 1)$ is convex, the expectation is also convex. For positive and negative labels, it can be calculated in the same manner. □

Theorem 4 *The expectation of the risk of the MH loss for the RL dataset is calibrated to the* 0-1-c-d *loss with* $d = \frac{1}{2} - c$ *for the original dataset under the conditions of Eq.* (14).

Proof We minimize the expectation of the MH loss with respect to (r, h) for the RL dataset as

$$\begin{aligned}
&\min_{(h,r)} \mathbb{E}_{z\sim \Pr(z|x)}[L_{MH}(h,r,x,z)] \\
&\quad = \min_{(h,r)} \left[\pi_+(x)L_{MH}(h,r,x,z=1) + \pi_-(x)L_{MH}(h,r,x,z=-1)\right. \\
&\qquad \left. + \pi_0(x)\tfrac{1}{2}\left(L_{MH}(h,r,x,z=1) + L_{MH}(h,r,x,z=-1)\right)\right] \\
&\quad = \min_{(h,r)}\left[\frac{2}{1+2c} \times \begin{cases} 2\pi_+ + \pi_0 & (h = -\frac{2}{1-4c^2}, r = \frac{1}{1+2c}) \\ 2\pi_- + \pi_0 & (h = \frac{2}{1-4c^2}, r = \frac{1}{1+2c}) \\ 2c & (h = 0, r = -\frac{1}{1+2c}) \end{cases}\right].
\end{aligned} \tag{24}$$

Then, we calculate the minimizers as

$$\begin{aligned}&\operatorname*{argmin}_{(h,r)} \mathbb{E}_{z\sim \Pr(z|x)}[L_{\mathrm{MH}}(h,r,x,z)]\\ &= \begin{cases} (-\frac{2}{1-4c^2}, \frac{1}{1+2c}) & \text{if } \pi_- \geq (1-2c)+\pi_+,\\ (\frac{2}{1-4c^2}, \frac{1}{1+2c}) & \text{if } \pi_+ \geq (1-2c)+\pi_-,\\ (0, -\frac{1}{1+2c}) & \text{otherwise.}\end{cases}\end{aligned} \tag{25}$$

The derived minimizers are consistent with Lemma 1 with $d = \frac{1}{2} - c$. □

References

Alworth, S. V., Watanabe, H., & Lee, J. S. (2010). Teachable, high-content analytics for live-cell, phase contrast movies. *Journal of Biomolecular Screening*, *15*(8), 968–977.

Andersen, MS., Dahl, J., & Vandenberghe, L. (2018). Cvxopt: A python package for convex optimization. http://cvxopt.org/.

Bar, Y., Diamant, I., Wolf, L., Lieberman, S., Konen, E., & Greenspan, H. (2015). Chest pathology detection using deep learning with non-medical training. In *2015 IEEE 12th international symposium on biomedical imaging (ISBI)* (pp. 294–297), IEEE.

Bartlett, P. L., & Wegkamp, M. H. (2008). Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, *9*, 1823–1840.

Bejnordi, B. E., Veta, M., Van Diest, P. J., Van Ginneken, B., Karssemeijer, N., Litjens, G., et al. (2017). Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, *318*(22), 2199–2210.

Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, *7*(Nov), 2399–2434.

Cannings, T. I., Fan, Y., & Samworth, R. J. (2020). Classification with imperfect training labels. *Biometrika*, *107*(2), 311–330.

Cortes, C., DeSalvo, G., & Mohri, M. (2016). Learning with rejection. In *International conference on algorithmic learning theory* (pp. 67–82). Springer.

Cruciani, F., Cleland, I., Nugent, C., McCullagh, P., Synnes, K., & Hallberg, J. (2018). Automatic annotation for human activity recognition in free living using a smartphone. *Sensors*, *18*(7), 2203.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., et al. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, *542*(7639), 115–118.

Harrison, D., & Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, *5*(1), 81–102.

He, X., & Niyogi, P. (2004). Locality preserving projections. *Advances in neural information processing systems 16* (pp. 153–160). Cambridge, MA: MIT Press.

Kiryo, R., Niu, G., Du Plessis, M. C., Sugiyama, M. (2017). Positive-unlabeled learning with non-negative risk estimator. In *Advances in neural information processing systems* (pp. 1675–1685).

Konishi, K., Mimura, M., Nonaka, T., Sase, I., Nishioka, H., & Suga, M. (2019). Practical method of cell segmentation in electron microscope image stack using deep convolutional neural network. *Microscopy*, *68*(4), 338–341.

Li, Y., Wu, B., Ghanem, B., Zhao, Y., Yao, H., & Ji, Q. (2016). Facial action unit recognition under incomplete data based on multi-label learning with missing labels. *Pattern Recognition*, *60*, 890–900.

Odena, A. (2016). Semi-supervised learning with generative adversarial networks. arXiv:1606.01583.

Park, J. K., Kwon, B. K., Park, J. H., & Kang, D. J. (2016). Machine learning-based imaging system for surface defect inspection. *International Journal of Precision Engineering and Manufacturing-Green Technology*, *3*(3), 303–310.

Pesapane, F., Codari, M., & Sardanelli, F. (2018). Artificial intelligence in medical imaging: threat or opportunity? Radiologists again at the forefront of innovation in medicine. *European Radiology Experimental*, *2*(1), 35.

Ren, R., Hung, T., & Tan, K. C. (2017). A generic deep-learning-based approach for automated surface inspection. *IEEE Transactions on Cybernetics*, *48*(3), 929–940.

Sakai, T., du Plessis, M. C., Niu, G., & Sugiyama, M. (2017). Semi-supervised classification based on classification from positive and unlabeled data. In *Proceedings of the 34th international conference on machine learning-volume 70, JMLR. org* (pp. 2998–3006).

Scheirer, W. J., de Rezende, Rocha A., Sapkota, A., & Boult, T. E. (2012). Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(7), 1757–1772.

Shahriyar, S. A., Alam, K. M. R., Roy, S. S., & Morimoto, Y. (2018). An approach for multi label image classification using single label convolutional neural network. In *2018 21st international conference of computer and information technology (ICCIT)* (pp. 1–6). IEEE.

Sugiyama, M. (2015). *Introduction to statistical machine learning*. Amsterdam: Morgan Kaufmann.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Berlin: Springer.

Wagner, J., Kim, J., & André, E. (2005). From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification. In *2005 IEEE international conference on multimedia and expo* (pp. 940–943). IEEE.

Wang, D., Khosla, A., Gargeya, R., Irshad, H., & Beck, A. H. (2016). Deep learning for identifying metastatic breast cancer. arXiv:1606.05718.