曖昧な訓練データを用いた二値分類の適用

大谷直也，大坪洋介，小池哲也，杉山　将

An Application of Binary Classification using Ambiguous Training Data

Naoya OTANI, Yosuke OTSUBO, Tetsuya KOIKE and Masashi SUGIYAMA

教師あり学習において，しばしば専門家にとってもラベル付けが難しいデータ（曖昧なデータ）が存在する．本稿では，曖昧なデータが存在する状況下での二値分類問題を検討し，社内で取得した細胞培養データに適用する．曖昧なデータはラベル付けが難しいという情報を持っているため，半教師あり学習におけるラベルなしデータとは異なる扱いが必要である．また，曖昧なデータは訓練データに存在するものの，テスト時は正負の二値に分類するため，正負と曖昧なクラスによる三値分類問題とも異なる．我々の提案手法は，リジェクト付き分類を拡張する形で定式化した．具体的には，リジェクト付き分類は，リジェクトコスト c を有する0-1-c 損失に基づいて，分類器とリジェクタを同時に学習する方法であるが，我々は0-1-c-d 損失として曖昧なデータに対する誤分類ペナルティ d を導入し，分類器とリジェクタを同時に学習する方法を提案した．計算の容易性の観点から，我々は0-1-c-d 損失の凸の上界となる代理損失を用いて実装を行った．細胞培養データに対する数値実験を通じて，曖昧なデータから得られる情報を，二値分類問題に有効に活用できることを示した．

In supervised learning, ambiguous (A) samples that are difficult to label even by domain experts are often encountered. In this study, we consider a binary classification problem using such A samples and apply our in-house datasets of a cell culture process. This problem is substantially different from semi-supervised learning because unlabeled samples are not necessarily difficult samples. Furthermore, it is different from the three-class classification involving positive (P), negative (N), and A classes because the test samples are not to be classified as the A class. Our proposed method extends binary classification with a reject option, which trains a classifier and a rejector simultaneously using P and N samples based on the 0–1-c loss with a rejection cost, c. More specifically, we propose to train a classifier and a rejector based on the 0–1-c-d loss using P, N, and A samples, where d is the misclassification penalty for A samples. In our practical implementation, we use the convex upper bound of the 0–1-c-d loss to achieve computational tractability. Numerical experiments using the in-house datasets demonstrate that our method can successfully utilize the additional information resulting from such A training data.

Key words 曖昧なサンプル，リジェクト付き分類，二値分類
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 [1] and manufacturing systems [2]. However, when the amount of labeled data is limited, current supervised learning methods become unreliable [3].

To efficiently obtain labeled data, domain knowledge has been used in many applications [2], [4]. However, as indicated in some studies [5], [6], ambiguous (A) samples that are substantially difficult to label even by domain experts are often encountered.

The goal of this study is to propose a novel classification method that can manage A samples. Specifically, we consider a binary classification problem where, in addition to positive (P) and negative (N) samples, A samples are available for training a classifier. Because of the characteristics of A samples, they are assumed to be located near the boundary between P and N classes.

We may consider employing three-class classification methods for the P, N, and A classes. However, because we intend to classify test samples only in the P or N class, not in the A class, naive three-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 a reject option [7], [8], where A test samples are not classified into P or N classes but as rejected (R). However, classification methods

with a reject option do not consider A samples in the training phase; hence, they cannot be employed in our problem.

Semi-supervised learning may be related to the current problem, where unlabeled (U) data, in addition to P and N data, are used to train a classifier [9]. In semi-supervised learning, U samples are P and N samples that have not yet been labeled, and they are not necessarily difficult samples to be labeled. By contrast, A samples in our target problem are typically distributed at the intersection of P and N classes. Thus, as the problem setups are intrinsically different, merely using semi-supervised learning methods in the current problem may not be optimal. Our problem and related methods are summarized in Table 1.

Table 1 Problem settings of related and our methods.

Methods	Labels in training data	Labels predicted in test phase	Relationship among classes
Binary classification	P / N	P / N	None
Three-class classification	Class 1 Class 2 Class 3	Class 1 Class 2 Class 3	None
Classification with reject option	P / N	P / R / N	R samples are in P/N mixed regions
Semi-supervised learning	P / U / N	P / N	U samples belong to P or N
Our proposal	P / A / N	P / N	A samples are in P/N mixed regions

To effectively solve the classification problem involving A data, we propose to extend classification with a reject option that trains a classifier and a rejector simultaneously using P and N samples based on the 0-1-*c* loss with a rejection cost, *c* [8]. The proposed method trains a classifier and a rejector based on the 0-1-*c*-*d* loss using P, N, and A samples, where *d* is the misclassification penalty for A samples. Then, in the test phase, we use the trained classifier to assign P or N labels to the test samples. Through experiments using an in-house cell culture dataset, we demonstrate that the proposed method can improve the test classification accuracy by using A samples in the training phase.

2 Formulation

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

2.1. Preliminary

We consider three class labels, namely, P, A, and N: $y \in \mathcal{Y}_0 = \{1, 0, -1\}$. We assume that we are assigned a set of P, A, and N samples $\{(x_i, y_i)\}_{i=1}^{N}$ drawn independently from a probability distribution with density $p_0(x, y)$ defined on $\mathcal{X} \times \mathcal{Y}_0$. Let $h : \mathcal{X} \to \mathbb{R}$ denote a discriminant function, with which a class label is predicted to be P or N (not predicted to be A) for a test input point, *x*, as $\hat{y} = \mathrm{sign}(h(x))$. Our goal is to learn a discriminant function that accurately classifies the test samples (not in the A class). Our key question in this scenario is whether we can utilize A training data to improve the classification accuracy of the discriminant function.

Hence, we develop a new method based on classification with a reject option (CRO) [8]. We first review the CRO method before deriving the new method.

2.2. Classification with Reject Option using Support Vector Machine (CRO-SVM)

Cortes *et. al.* [8] introduced a rejection function, $r : \mathcal{X} \to \mathbb{R}$, in addition to the discriminant function, to identify regions with a high risk for misclassification. When the rejection function yields a positive value, the corresponding sample is classified into the P or N class by using classifier *h*; otherwise, the sample is rejected and not classified. When a sample is rejected, a rejection cost, *c*, is incurred, which trades off the risk of misclassification. To realize this idea, the *0-1-c loss* is introduced:

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

where 1_A is the indicator function that yields 1 if statement *A* is true and 0 otherwise. When $c = 0$, all samples are rejected because the loss function does not incur any cost. By contrast, when $c \ge 0.5$, no samples are rejected because the expectation of the 0-1 loss, $1_{yh(x) \le 0}$, is less than 0.5; thus, the 0-1-*c* loss is reduced to the 0-1 loss. Therefore, we only consider *c* such that $0 < c < 0.5$.

Based on the 0-1-*c* loss, the problem is expressed as

$$\begin{aligned} (h^*, r^*) &= \underset{(h,r)}{\mathrm{argmin}}\, R(h, r), \\ R(h, r) &= \mathbb{E}_{p_0(x,y)}\left[L_{01c}(h, r, x, y) \right], \end{aligned} \tag{2}$$

where h^* and r^* denote the optimal discriminant function and rejection function, respectively, and $\mathbb{E}_{p_0(x,y)}$ denotes the expectation over $p_0(x, y)$. In practice, because the true density, $p_0(x, y)$, is unknown, we typically use the empirical dis-

tribution to approximate the expectation:

$$\widehat{R}(h)=\frac{1}{N}\sum_{i=1}^{N}L_{01c}(h,r,x_i,y_i). \tag{3}$$

Because of the discrete nature of the 0–1-*c* loss, its direct optimization is computationally intractable. To avoid discontinuity, the following surrogate loss, known as the max-hinge (MH) loss, is introduced:

$$L_{\mathrm{MH}}(h,r,x,y)= \max\left(1+\frac{\alpha}{2}\big(r(x)-yh(x)\big),c\big(1-\beta r(x)\big),0\right), \tag{4}$$

where $\alpha, \beta > 0$ are the hyperparameters used to control the shape of the surrogate loss. The surrogate loss is an extension of the hinge loss, which is employed in a support vector machine (SVM) [10].

Further, introducing 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 yields the following quadratic program:

$$\left(\hat{w},\hat{u},\hat{\xi}\right)=\underset{(w,u,\xi)}{\operatorname{argmin}}\left[\frac{\lambda}{2}\|w\|^2+\frac{\lambda'}{2}\|u\|^2+\frac{1}{N}\sum_{i=1}^{N}\xi_i\right] \quad \text{s.t.}\begin{pmatrix}\xi_i\geq 1+\frac{\alpha}{2}(r_i-y_ih_i)\\ \xi_i\geq c(1-\beta r_i)\\ \xi_i\geq 0\end{pmatrix}\text{for } i=1,\ldots,N, \tag{5}$$

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; h_i and r_i denote the values of the discriminant function and rejection function at sample point x_i expressed as $h_i=\sum_{j=1}^{N}w_j\phi_j(x_i)$ and $r_i=\sum_{j=1}^{N}u_j\phi_j(x_i)$, respectively. The resulting discriminant and rejection functions are expressed as $h(x;\hat{w})=\sum_{j=1}^{N}\hat{w}_j\phi_j(x)$ and $r(x;\hat{u})=\sum_{j=1}^{N}\hat{u}_j\phi_j(x)$, respectively.

We refer to this method as CRO-SVM.

2.3. Proposed Method: Classification with A Data using SVM (CAD-SVM)

To manage A training data in the SVM formulation, we extend the 0–1-*c* loss to the *0–1-c-d loss*, as Eq. (6):

Table 2 0–1-*c* loss function.

Judgment (h, r) / Label y	P $h>0$ $r>0$	R $r\leq 0$	N $h\leq 0$ $r>0$
P: $y=1$	0	c	1
N: $y=-1$	1	c	0

Table 3 0–1-*c*-*d* loss function.

Judgment (h, r) / Label y	P $h>0$ $r>0$	R $r\leq 0$	N $h\leq 0$ $r>0$
P: $y=1$	0	c	1
A: $y=0$	d	0	d
N: $y=-1$	1	c	0

$$L_{01\mathrm{cd}}(h,r,x,y)= 1_{y^2=1}\left(1_{yh(x)\leq 0}1_{r(x)>0}+c1_{r(x)\leq 0}\right)+d1_{y=0}1_{r(x)>0}. \tag{6}$$

Tables 2 and 3 present comparisons of the behaviors of the 0–1-*c* and 0–1-*c*-*d* losses, respectively. For the P and N samples, the 0–1-*c*-*d* loss behaves the same as the 0–1-*c* loss. In contrast, for the A samples, the 0–1-*c*-*d* loss incurs penalty *d* when they are classified as the P or N class. Therefore, A samples tend to be classified into the A class if we employ the 0–1-*c*-*d* loss. Unlike the CRO formulation, CAD utilizes A samples to learn a rejector explicitly.

This discussion may mislead us as if we are just solving a three-class problem involving P, N, and A classes. However, we do not classify the test samples into the A class, but only into the P and N classes. To solve the CAD problem, we utilize a binary discriminant function, *h*, and a rejection function, *r*, as in the CRO formulation reviewed earlier. We train *h* and *r* based on the 0–1-*c*-*d* loss, and we use only *h* in the test phase to classify the test samples into P and N classes. Owing to the interplay between *h* and *r* in the 0–1-*c*-*d* loss, we can utilize A samples to train *h* through *r*.

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

$$\begin{aligned}L_{01\mathrm{cd}}(h,r,x,y)&\leq 1_{y^2=1}L_{\mathrm{MH}}(h,r,x,y)+d1_{y=0}\max\big(1+\beta r(x),0\big)\\ &=y^2\max\left(1+\frac{\alpha}{2}\big(r(x)-yh(x)\big),c\big(1-\beta r(x)\big),0\right)+\big(1-y^2\big)\max\big(d\big(1+\beta r(x)\big),0\big)\\ &\leq y^2\max\left(1+\frac{\alpha}{2}\big(r(x)-yh(x)\big),c\big(1-\beta r(x)\big),0\right)+\big(1-y^2\big)\max\big(\eta d\big(1+\beta r(x)\big),0\big)\\ &\equiv L_{\mathrm{MHA}}(h,r,x,y),\end{aligned} \tag{7}$$

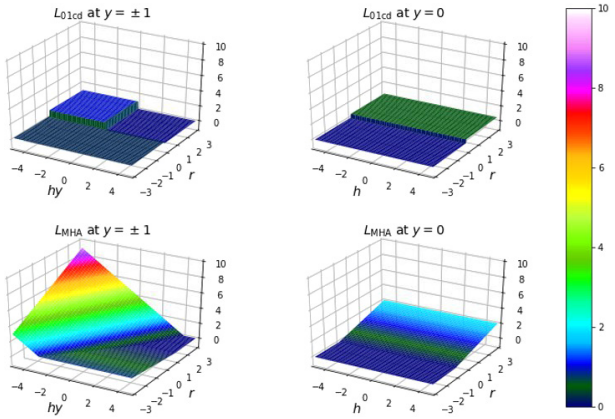

Fig. 1 0–1-*c*-*d* loss, L_{01cd}, and its surrogate loss, L_{MHA}.

where $\eta \geq 1$ is a hyperparameter that controls the shape of the surrogate loss (see Fig. 1 for the visualization).

Next, similar to CRO-SVM, we have the following quadratic program:

$$\left(\hat{w},\hat{u},\hat{\xi}\right) = \underset{(w,u,\xi)}{\operatorname{argmin}} \left[\frac{\lambda}{2}\|w\|^2 + \frac{\lambda'}{2}\|u\|^2 + \frac{1}{N}\sum_{i=1}^{N}\xi_i\right]$$

$$\text{s.t.} \begin{pmatrix} \xi_i \geq y_i^2\left(1+\frac{\alpha}{2}\left(r_i - y_i h_i\right)\right) \\ \xi_i \geq y_i^2 \eta c\left(1-\beta r_i\right) \\ \xi_i \geq \left(1-y_i^2\right)\eta d\left(1+\beta r_i\right) \end{pmatrix} \tag{8}$$

$$\text{for } i = 1,\ldots,N.$$

This formula expresses our proposed method, CAD-SVM.

To select hyperparameters (α,β,η), we can apply the following theorem (its proof is available in [11]):

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

$$\left(h_{01cd}^*, r_{01cd}^*\right) = \underset{(h,r)}{\operatorname{argmin}} \mathbb{E}_{p_0(y|x)}\left[L_{01cd}\left(h,r,x,y\right)\right], \tag{9}$$

and

$$\left(h_{MHA}^*, r_{MHA}^*\right) = \underset{(h,r)}{\operatorname{argmin}} \mathbb{E}_{p_0(y|x)}\left[L_{MHA}\left(h,r,x,y\right)\right]. \tag{10}$$

Then, for

$$\alpha^* = 2(1-2c), \quad \beta^* = 1+2c, \quad \eta^* = \frac{2}{1+2c}, \tag{11}$$

the signs of $\left(h_{MHA}^*, r_{MHA}^*\right)$ *match those of* $\left(h_{01cd}^*, r_{01cd}^*\right)$.

In the next section, we demonstrate that this method is feasible. It is noteworthy that Eq. (11) does not include *d*.

3 Numerical Experiments

In this section, we report the experimental results obtained using an in-house dataset from a cell culture process. A detailed performance evaluation and a comparison with baseline methods on other datasets have been reported in [11].

3.1. Dataset

For real-world applications, we prepared an in-house cell-culture dataset. This dataset contains 124 fields of view (FOVs). For each FOV, images were acquired three times at T = 99, 279, and 459 h. All images for each FOV were analyzed using an image processing software, CL-Quant [12], and converted to eight morphological features, such as the average brightness and average area of cells. Based on the final image for each FOV (T = 459 h), each FOV was annotated by experts. If the cells in the image appeared healthy/damaged, the image was labeled as P/N. Otherwise, the experts assign A labels to samples that cannot be confidently classified as healthy or damaged. The numbers of samples for the P, N, and A classes were 41, 59, and 24, respectively. Our goal was to predict the final state of each FOV (annotated by the experts in this simulation) using morphological features obtained from each time point of the culturing process. In total, we trained and evaluated three types of datasets (Datasets 1, 2, and 3), corresponding to the time point of the input images, T. For Datasets 1 and 2, the images from which we extracted the input features and those from which we annotated the output labels were different; this is illustrated in Fig. 2.

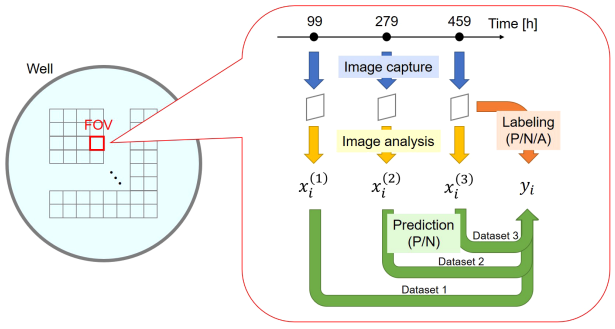

Fig. 2 Schematic image of datasets. We created three types of datasets and evaluated corresponding models.

3.2. Experimental Settings

Using the aforementioned datasets, we compared the classification performance of the SVM, SVM-RL (random label), LapSVM [13], two-step SVM, CRO-SVM, CRO-SVM-RL, and CAD-SVM.

For each method, 1500 test runs were performed by changing the training and test datasets, which were randomly selected from the original dataset. The ratio of the

training and test datasets was 4:1. For each test run, five-fold cross-validation was performed to determine the relevant parameters. For validation and in the test phase, only P and N samples were applied to the discriminant function; hence, we were able to evaluate the binary classification accuracy.

We determined 10 hyperparameters $(\lambda, \lambda', \sigma, \sigma', \tau, c, d, \alpha, \beta, \eta)$, where σ is the width of the Gaussian radial basis function in the basis function $\phi_i(x) = \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right)$, σ' is the hyperparameter of the weight matrix, W, of the graph Laplacian expressed as $W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma'^2}\right)$ (only in the LapSVM), and τ is the coefficient of the graph Laplacian regularization (only in the LapSVM). The hyperparameters (α, β, η) were determined by using Eq. (11), and other hyperparameters were selected via five-fold cross-validation (see [11] for details). The experimental procedure applied for each dataset and method is summarized in Algorithm 1.

Input: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, $\mathcal{H} = \{(\lambda_\xi, \lambda'_\xi, \sigma_\xi, \sigma'_\xi, \tau_\xi, c_\xi, d_\xi)\}_{\xi=1}^{\Xi}$
For $j = 1, \ldots, 1500$ **do**:
  $\mathcal{D}_{\text{train}} \leftarrow$ (random set of $0.8N$ records in $\mathcal{D}$)
  $\mathcal{D}_{\text{test}} \leftarrow \mathcal{D} \setminus \mathcal{D}_{\text{train}}$
  $\left(\mathcal{D}_{\text{train}}^{(1)}, \ldots, \mathcal{D}_{\text{train}}^{(5)}\right) \leftarrow$ (random set that equally divide $\mathcal{D}$ into 5)
  For $k = 1, \ldots, 5$ **do**:
    $\mathcal{D}_{\text{valid}} \leftarrow \mathcal{D}_{\text{train}}^{(k)}$
    $\mathcal{D}_{\text{subtrain}} \leftarrow \mathcal{D}_{\text{train}} \setminus \mathcal{D}_{\text{valid}}$
    For $\xi = 1, \ldots, \Xi$ **do**:
      Create a model using $\mathcal{D}_{\text{subtrain}}$ under hyperparameter setting $\mathcal{H}_\xi$
      $a_\xi^{(k)} \leftarrow$ (binary classification accuracy of $\mathcal{D}_{\text{valid}}$)
  $\overline{a_\xi} \leftarrow \frac{1}{5}\sum_{k=1}^{5} a_\xi^{(k)}$
  $\xi^* \leftarrow \text{argmax}_\xi \overline{a_\xi}$
  Create a model using $\mathcal{D}_{\text{train}}$ under hyperparameter setting $\mathcal{H}_{\xi^*}$
  $a_j \leftarrow$ (binary classification accuracy of $\mathcal{D}_{\text{test}}$)
Calculate the mean and standard deviation of a_j

Algorithm 1 Experimental procedure for each dataset and method.

3.3. Results

Table 4 summarizes the test accuracy of each method. The CAD-SVM showed statistically significant improvements over the other methods, particularly in the earlier stages of the culturing process. In the earlier stages, the input data contained few or inaccurate information; therefore, utilizing A samples would be beneficial. However, because the input data contained almost complete information during the final state, the information of A samples need not be utilized. If the information of A samples is intrinsically meaningless, then the SVM would be a better solution as it utilizes the hinge loss directly based on the 0–1 loss (*i.e.*, the binary classification accuracy). Overall, the CAD-SVM is a promising method for utilizing A samples.

Table 4 Test accuracy for each timepoint, where ± denotes standard deviation. Boldfaced numbers represent the best and comparable results with 5% t-test.

	Dataset 1 (T = 99)	Dataset 2 (T = 279)	Dataset 3 (T = 459)
SVM	0.732 ± 0.092	0.799 ± 0.088	**0.941 ± 0.049**
SVM-RL	0.730 ± 0.096	0.805 ± 0.088	0.929 ± 0.058
LapSVM	0.731 ± 0.091	0.801 ± 0.089	0.931 ± 0.055
Two-step SVM	0.733 ± 0.097	0.788 ± 0.090	0.931 ± 0.054
CRO-SVM	0.747 ± 0.095	**0.814 ± 0.087**	**0.939 ± 0.050**
CRO-SVM-RL	0.740 ± 0.097	**0.818 ± 0.085**	0.920 ± 0.063
CAD-SVM	**0.755 ± 0.094**	**0.819 ± 0.087**	0.937 ± 0.051

4 Conclusion

In this study, we aimed to reduce labeling cost and improve classification accuracy by allowing labelers to provide A labels for difficult samples. We extended a classification method with a reject option and proposed a novel classification method, named CAD-SVM, which uses the 0–1-*c*-*d* loss. We derived a surrogate loss for the 0–1-*c*-*d* loss, thereby allowing us to convert the optimization problem into a convex quadratic program. We conducted numerical experiments and demonstrated that A labels can be effectively used to improve the classification accuracy.

Although our proposed method was based on the SVM, it would be more useful if it is applicable to other models, particularly to deep neural networks. In future studies, we will conduct a theoretical analysis of the proposed method in terms of the statistical consistency and convergence rate. Extending the proposed loss function to semi-supervised, imperfect labeling, or multiclass problems is also a promising direction for future research.

References

[1] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature,* vol. 542, no. 7639, pp. 115–118, 2017.

[2] R. Ren, T. Hung and K. C. Tan, "A generic deep-learning-based approach for automated surface inspection," *IEEE transactions on cybernetics,* vol. 48, no. 3, pp. 929–940, 2017.

[3] F. Pesapane, M. Codari and F. Sardanelli, "Artificial intelligence in medical imaging: threat or opportunity? Radiologists again at the forefront of innovation in medicine," *European radiology experimental,* vol. 2, no. 1, p. 35, 2018.

[4] K. Konishi, M. Mimura, T. Nonaka, I. Sase, H. Nishioka and M. Suga, "Practical method of cell segmentation in electron microscope image stack using deep convolutional

neural network," *Microscopy*, vol. 68, no. 4, pp. 338–341, 2019.

[5] Y. Li, B. Wu, B. Ghanem, Y. Zhao, H. Yao and Q. Ji, "Facial action unit recognition under incomplete data based on multi-label learning with missing labels," *Pattern Recognition,* vol. 60, pp. 890–900, 2016.

[6] S. A. Shahriyar, K. M. R. Alam, S. S. Roy and Y. Morimoto, "An approach for multi label image classification using single label convolutional neural network," in *2018 21st international conference of computer and information technology (ICCIT)*, 2018.

[7] P. L. Bartlett and M. H. Wegkamp, "Classification with a reject option using a hinge loss," *Journal of Machine Learning Research*, vol. 9, no. Aug, pp. 1823–1840, 2008.

[8] C. Cortes, G. DeSalvo and M. Mohri, "Learning with rejection," in *International Conference on Algorithmic Learning Theory*, 2016.

[9] T. Sakai, M. C. du Plessis, G. Niu and M. Sugiyama, "Semi-supervised classification based on classification from positive and unlabeled data," in *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[10] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Berlin, Germany: Springer-Verlag, 1995.

[11] N. Otani, Y. Otsubo, T. Koike and M. Sugiyama, "Binary classification with ambiguous training data," *Machine Learning*, vol. 109, pp. 2369–2388, 2020.

[12] S. V. Alworth, H. Watanabe and J. S. J. Lee, "Teachable, high-content analytics for live-cell, phase contrast movies," *Journal of biomolecular screening*, vol. 15, no. 8, pp. 968–977, 2010.

[13] M. Belkin, P. Niyogi and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of machine learning research*, vol. 7, no. Nov, pp. 2399–2434, 2006.

大谷直也　Naoya OTANI
研究開発本部　数理技術研究所
Mathematical Sciences Research Laboratory
Research & Development Division

大坪洋介　Yosuke OTSUBO
研究開発本部　数理技術研究所
Mathematical Sciences Research Laboratory
Research & Development Division

小池哲也　Tetsuya KOIKE
研究開発本部　数理技術研究所
Mathematical Sciences Research Laboratory
Research & Development Division

杉山　将　Masashi SUGIYAMA
理化学研究所
RIKEN
東京大学
The University of Tokyo


大谷直也
Naoya OTANI


大坪洋介
Yosuke OTSUBO


小池哲也
Tetsuya KOIKE


杉山　将
Masashi SUGIYAMA