BINARY CLASSIFICATION ONLY FROM UNLABELED DATA BY ITERATIVE UNLABELED-UNLABELED CLASSIFICATION

Hirotaka Kaji[1]*, and Masashi Sugiyama*[2,3]

[1]Frontier Research Center, Toyota Motor Corp., Shizuoka, Japan
[2]Center for Advanced Intelligence Project, RIKEN, Tokyo, Japan
[3]Graduate School of Frontier Sciences, The University of Tokyo, Chiba, Japan

ABSTRACT

Unlabeled-unlabeled (UU) classification (du Plessis et al. 2013) allows us to train a binary classifier from two sets of unlabeled data with different class priors. In this paper, we go beyond this scenario and try to train a binary classifier only from a *single* set of unlabeled data. Our key idea is to iteratively perform UU classification: We initially split the original single unlabeled dataset into two disjoint datasets and perform UU classification. We then split the original unlabeled dataset in a different way based on the obtained classifier, perform UU classification, and repeat this process until convergence. We numerically show that the classification accuracy tends to be improved over iterations. Finally, we apply our iterative UU classification method to a real-world drowsiness prediction problem and demonstrate its usefulness.

Index Terms— Unlabeled-unlabeled classification, Iteration

1. INTRODUCTION

Machine learning with big data has been highly successful in various application domains [1]. However, even in the era of big data, collecting a large number of *labeled* data is not straightforward, e.g., in medicine, biometrics, and manufacturing, since data annotation is highly costly. To reduce the labelling costs, *weakly-supervised classification* [2] has been actively researched. In weakly-supervised classification, data which are less informative but less expensive [3, 4] than fully labeled data are used for training a classifier. *Semi-supervised classification* [5, 6, 7, 8] is the most well-known example of weakly-supervised classification, where unlabeled data are utilized. Other examples include multiple instance classification [9], partial label classification [10], positive-unlabeled classification [11, 12], complementary classification [13], similar-unlabeled classification [4], and unlabeled-unlabeled (UU) classification [14]. Among them, UU classification requires the least supervision. More specifically, UU classification methods train a classifier only from two sets of unlabeled data having different class proportions[1]. UU classification is inherently unsupervised, but a classifier is trained to separate two classes with theoretical guarantee. Experimentally, UU classification was shown to perform much better than unsupervised clustering techniques such as k-means [15] and spectral clustering [16].

However, even though annotation is not needed in UU classification, obtaining two sets of unlabeled data having different class proportions is sometimes not straightforward. For example, in our application of drowsiness prediction, we may simply divide a physiological time-series record into two sections and hope that drowsy-awake ratios are significantly different in each section. However, since the transition of a person's awake and drowsiness states may change only gradually, the drowsy-awake ratios in two consecutive sections may not be that much different. More generally, when only a single set of unlabeled data is given, it is not obvious how to divide it into two datasets so that class proportions are significantly different.

The goal of this paper is to give a practical method to perform UU classification only from a single set of unlabeled data. Our basic idea is to iteratively perform UU classification: Given a single set of unlabeled data, we split it into two disjoint subsets and perform UU classification. We then split the original unlabeled dataset in a different way based on the obtained classifier, perform UU classification, and repeat this process until convergence. We discuss heuristics for initialization and improving the performance over iterations, and experimentally demonstrate the effectiveness of the proposed iterative UU (iUU) classification method.

2. UNLABELED-UNLABELED CLASSIFICATION

In this section, we formulate the problem of *unlabeled-unlabeled (UU) classification* [14] and review its solution.

Problem Formulation: Let $\boldsymbol{x} \in \mathbb{R}^d$ be a d-dimensional pattern and $y \in \{+1, -1\}$ be its class label. Let $p(\boldsymbol{x}, y)$ and

[1]Note that the different class proportions need not be known, but significantly different class proportions tend to yield better classification performance [14].

$p'(\boldsymbol{x}, y)$ be unknown joint probability densities. UU classification is a way to train a classifier from two sets of unlabeled data,

$$\mathcal{X} := \{\boldsymbol{x}_i\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}) = \sum_y p(\boldsymbol{x}, y) = \sum_y p(\boldsymbol{x}|y)p(y),$$

$$\mathcal{X}' := \{\boldsymbol{x}'_i\}_{i=1}^{n'} \overset{\text{i.i.d.}}{\sim} p'(\boldsymbol{x}) = \sum_y p'(\boldsymbol{x}, y) = \sum_y p'(\boldsymbol{x}|y)p'(y),$$

which share the same class-conditional densities but have different class-priors: $p(\boldsymbol{x}|y) = p'(\boldsymbol{x}|y)$ and $p(y) \neq p'(y)$.

For test joint density $q(\boldsymbol{x}, y)$, the goal of UU classification is to obtain from $\mathcal{X}$ and $\mathcal{X}'$ the Bayes optimal classifier given by $\mathrm{sign}(q(y = +1|\boldsymbol{x}) - q(y = -1|\boldsymbol{x}))$. Throughout this paper, we assume that the test joint density shares the same class-conditional density $q(\boldsymbol{x}|y) = p(\boldsymbol{x}|y)$ with uniform test class prior $q(y = +1) = q(y = -1) = \frac{1}{2}$.

It was shown [14] that the above Bayes optimal classifier can be equivalently expressed as $C\mathrm{sign}[p(\boldsymbol{x}) - p'(\boldsymbol{x})]$, where $C = \mathrm{sign}[p(y = +1) - p'(y = +1)]$. Although C is unidentifiable only from unlabeled data, it takes either $+1$ or -1 and thus still the optimal boundary between two classes can be obtained without C: $g^*(\boldsymbol{x}) = \mathrm{sign}[p(\boldsymbol{x}) - p'(\boldsymbol{x})]$.

A naive way to estimate $g^*(\boldsymbol{x})$ is to estimate densities $p(\boldsymbol{x})$ and $p'(\boldsymbol{x})$ separately from $\mathcal{X}$ and $\mathcal{X}'$ and plug the estimated densities in $g^*(\boldsymbol{x})$. However, such a plug-in approach is not accurate enough to estimate $g^*(\boldsymbol{x})$. A slightly more sophisticated approach is to directly estimate the density difference $p(\boldsymbol{x}) - p'(\boldsymbol{x})$ from $\mathcal{X}$ and $\mathcal{X}'$ [17] and plug the estimated density difference in $g^*(\boldsymbol{x})$. However, this is still a plug-in approach. The most direct approach is to estimate the entire quantity $g^*(\boldsymbol{x})$ from $\mathcal{X}$ and $\mathcal{X}'$. This can be achieved by the *direct sign density difference* (DSDD) method [14], which is reviewed below.

Direct Sign Density Difference Estimation: The *Fenchel-dual lower bound* of the L_1-distance between $p(\boldsymbol{x})$ and $p'(\boldsymbol{x})$ is given, for any function f such that $|f(\boldsymbol{x})| \leq 1$ for all $\boldsymbol{x}$, as

$$\int |p(\boldsymbol{x}) - p'(\boldsymbol{x})| \mathrm{d}\boldsymbol{x} \geq \int f(\boldsymbol{x})(p(\boldsymbol{x}) - p'(\boldsymbol{x})) \mathrm{d}\boldsymbol{x}.$$

We can easily confirm that $f(\boldsymbol{x}) = g^*(\boldsymbol{x})$ achieves the equality, i.e., the above Fenchel-dual lower bound can be maximized by $g^*(\boldsymbol{x})$, which is the quantity we want to estimate in UU classification.

Let us consider parametric model $f_{\boldsymbol{\alpha}}(\boldsymbol{x})$ for function f, and learn parameter $\boldsymbol{\alpha}$ to maximize the above Fenchel-dual lower bound. In practice, we minimize an negated regularized empirical criterion given by

$$J(\boldsymbol{\alpha}) = \frac{1}{n'} \sum_{i=1}^{n'} R(f_{\boldsymbol{\alpha}}(\boldsymbol{x}'_i)) - \frac{1}{n} \sum_{j=1}^{n} R(f_{\boldsymbol{\alpha}}(\boldsymbol{x}_j)) + \frac{\lambda}{2} \|\boldsymbol{\alpha}\|^2,$$

where $\lambda \geq 0$ is the regularization constant, $\|\cdot\|$ denotes the ℓ_2-norm, and $R(z)$ is a clipping function to fulfill $|f_{\boldsymbol{\alpha}}(\boldsymbol{x})| \leq 1$ for all $\boldsymbol{x}$. In our implementation, we use smoothed clipping function $R(z) = \tanh(z)$. As $f_{\boldsymbol{\alpha}}(\boldsymbol{x})$, we employ a Gaussian kernel model given by $f_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \boldsymbol{\alpha}^\top \boldsymbol{\varphi}(\boldsymbol{x})$, where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{n+n'})^\top$, $\boldsymbol{\varphi}(\boldsymbol{x})$ is the $(n + n')$-dimensional vector whose ℓ-th element is given by $\exp\left(\frac{-\|\boldsymbol{x} - \boldsymbol{c}_\ell\|^2}{2h^2}\right)$, $\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{n+n'}\} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_{n'}\}$, and $h > 0$. We may subsample $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_{n'}\}$ if we want to reduce the number of parameters when $n + n'$ is too large.

Since $J(\boldsymbol{\alpha})$ is non-convex, a (stochastic) gradient method is used to find a local minimizer: at the t-th iteration, new solution $\boldsymbol{\alpha}_t$ is obtained from the previous solution $\boldsymbol{\alpha}_{t-1}$ as $\boldsymbol{\alpha}_t \leftarrow \boldsymbol{\alpha}_{t-1} - \eta \nabla J(\boldsymbol{\alpha}_{t-1})$, where $\eta > 0$ is a step size and gradient $\nabla J(\boldsymbol{\alpha})$ is given by

$$\nabla J(\boldsymbol{\alpha}) = \frac{1}{n'} \sum_{i=1}^{n'} (1 - \tanh(\boldsymbol{\alpha}^\top \boldsymbol{\varphi}(\boldsymbol{x}'_i))^2) \boldsymbol{\varphi}(\boldsymbol{x}'_i) - \frac{1}{n} \sum_{j=1}^{n} (1 - \tanh(\boldsymbol{\alpha}^\top \boldsymbol{\varphi}(\boldsymbol{x}_j))^2) \boldsymbol{\varphi}(\boldsymbol{x}_j) + \lambda \boldsymbol{\alpha}.$$

3. ITERATIVE UU CLASSIFICATION

Even though annotation is not needed in UU classification, obtaining two sets of unlabeled data having different class proportions is sometimes not straightforward. In this section, we propose a practical method to perform UU classification only from a single set of unlabeled data.

The key idea of our proposed method, called *iterative UU* (iUU) classification, is to employ a classifier obtained by UU classification in the next step to split the original unlabeled dataset into two datasets. We initially split the unlabeled dataset into two subsets in some way, e.g., by the k-means clustering method [15]. Then we perform UU classification and obtain a classifier. We use the obtained classifier to split the original unlabeled dataset in a *different* way than the previous step. However, similar (or the same) partition may be obtained, which slows down (or stops) the learning process. To accelerate the evolution, we randomly exchange part of samples between two subsets, with mixture rate $0 < \pi < 0.5$. In experiments, we will employ DSDD as a UU classification method, and we call its iterative variant the *iterative direct sign density difference (iDSDD)* method. A pseudo code of iUU classification is described in Algorithm 1. When two unlabeled datasets have rather similar class priors, UU classification tends to perform poorly [14], iUU classification may also be used to improve the classification performance in such a scenario.

Experiment: We employ benchmark datasets to demonstrate the practical usefulness of iUU classification. In the following experiments, we set the maximum number of iterations to $t_{\max} = 20$. As an optimizer of the cost function, the momentum method [18] with $\eta = 0.01$ and $\alpha = 0.9$ was adopted. Parameters in the Gaussian kernel model were ini-

Algorithm 1 Iterative UU classification

Require: a single set of unlabeled samples $\mathcal{X}$, the maximum number of iterations $t_{\max}$, and mixture rate $\pi \in (0.0, 0.5)$
Ensure: model parameters

```
Split X into X_0^+ and X_0^- by using, e.g.,k-means.
Initialize counter t = 1.
repeat
    Randomly divide X_{t-1}^+ into X_+ and X'_+ with ratio π : 1 − π.
    Randomly divide X_{t-1}^- into X_− and X'_− with ratio 1 − π : π.
    X_t = X_+ ∪ X_−, X'_t = X'_+ ∪ X'_−
    Perform UU classification for X_t and X'_t, and give labels +1 and −1 to all unlabeled samples.
    Divide X_t ∪ X'_t into X_t^+, X_t^− based on the labels.
    t ← t + 1.
until t ≥ t_max
```

tialized to $\boldsymbol{\alpha} = \mathbf{0}$. Hyperparameters, the regularization constant λ and the bandwidth of the Gaussian kernel h, were selected from $\{0.001, 0.01, 0.1, 1\}$ and $\{0.5, 1, 2, 5, 10\}$ based on 5-fold cross validation with grid search, respectively. Performance of iUU classification was investigated for mixture rate $\pi \in \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45\}$.

Since by UU classification, we cannot know exact labels, but only know the decision boundary between the positive and negative classes, we used the labeling error rate (LER) for performance evaluation: $\mathrm{LER} := \min(\mathrm{MCR}, 1 - \mathrm{MCR})$, where MCR denotes the misclassification rate.

We applied iUU classification to nine benchmark datasets taken from the IDA benchmark repository [19] and the UCI machine learning repository [20]. In this experiment, two scenarios, UU classification from two unlabeled datasets with similar class priors initialized with DSDD (*Scenario 1*) and from a single unlabeled dataset initialized with k-means (*Scenario 2*), were investigated. We considered Scenario 1 with $p(y = +1) = 0.4$ and $p'(y = +1) = 0.6$. We set the number of training samples to $n = n' = 100$. As test samples, we used 100 samples from each of the positive and negative classes in general, while 20 samples for the Heart dataset and 25 samples for the Ionosphere dataset due to lack of samples. All feature vectors were normalized to have zero mean and unit standard deviation.

Table 1 shows the average and standard deviation of the LER for the benchmark problems over 100 trials. In Scenario 1, we can confirm that the proposed method with small π (0.05 to 0.2) improves the LERs for many cases. On the other hand, the effectiveness of the proposed method for Scenario 2 is weaker than Scenario 1. However, the average LER tends to be reduced for several cases such as the Diabetes dataset, although there was no statistically significant improvement. Overall, we empirically found that iUU classification works effectively for many benchmark datasets.

4. REAL-WORLD DROWSINESS PREDICTION

In this section, we demonstrate the usefulness of iUU classification on real-world drivers' drowsiness prediction based on heart beat information. Since time-consuming manual annotation is necessary to develop a drowsiness predictor based on supervised classification, a drowsiness predictor which can be trained only from unlabeled samples has a huge impact in automotive applications.

Drivers' Drowsiness Dataset: We briefly introduce the drivers' drowsiness dataset [21]. Three healthy males played a driving simulator along an expressway around 100 km/h with overtaking other cars, until an expert observed their strong drowsiness or they finished the whole driving task (about 150 km distance) [2]. The dataset is made up by input vector $\boldsymbol{x}$ and the drowsiness score. Input vector $\boldsymbol{x}$ is composed of seven features such as LF (the spectral power of the low frequency component (0.04-0.15 Hz)) extracted from electrocardiogram (ECG) measured by a physiological amplifier. These features were computed from the peak-to-peak interval of the R-wave which is the largest wave of ECG at 60 seconds intervals with 120-second sliding windows, and were normalized to have zero mean and unit standard deviation for each subject. A drowsiness score based on facial expressions [22] from 1 ("Not sleepy") to 5 ("Very sleepy") was given every 60 seconds by experts. In this dataset, each subject performed the experiment 10 times, and only five trials were annotated and remaining five trials were unlabeled.

Experiment: Since the transition of a person's awake and drowsy states during driving is gradually changed in the course of nature, it can be expected that dividing the sequence of data into two at arbitrary time generates datasets with slightly different class proportions. In order to evaluate the effectiveness of the proposed method, we conducted an experiment in comparison with supervised classification. We used the samples of the annotated five trials of each subject for the experiment. In order to simulate a more realistic situation, at first, we marged the time series of five trials into a single data sequence for each subject. Then, we converted the drowsiness scores from 1 to 2 into the positive class ("awake") and from 3 to 5 into the negative class ("drowsy") and obtained binary label y.

We used 280 samples from the beginning of each sequence because of the limitation of samples. Then, the data sequence was divided into two training datasets ($n = n' = 100$) and a test dataset ($n^{\mathrm{test}} = 80$). The sequences were split in three ways to generate different class proportions. A conceptual diagram of generating datasets is shown in Fig. 1. As a result, we obtained nine patterns of training and test datasets from three subjects. For iUU classification, two training datasets were treated as the unlabeled datasets. On the other hand, two training datasets were merged into a

[2]The design of experiments was approved and conducted according to Toyota Motor Corporation's ethical guidelines.

Table 1: Comparison of the mean and standard deviation of the LER over 100 trials. The results of initial and final iteration ($t_{\max} = 20$) with the mixture rate π from 0.05 to 0.45 are indicated. The improved and deteriorated results according to the paired t-test at the significance level 5% are specified by bold and italic faces.

Senario 1

Benchmark	Initial	$\pi = 0.05$	$\pi = 0.1$	$\pi = 0.15$	$\pi = 0.2$	$\pi = 0.25$	$\pi = 0.3$	$\pi = 0.35$	$\pi = 0.4$	$\pi = 0.45$
Australian	.266(.144)	**.202**(.078)	**.201**(.086)	**.198**(.074)	**.182**(.052)	**.180**(.043)	**.185**(.043)	**.186**(.040)	*.305*(.128)	*.399*(.104)
Banana	.373(.085)	.381(.074)	*.391*(.073)	*.405*(.067)	*.411*(.072)	*.433*(.046)	*.436*(.048)	*.439*(.044)	*.432*(.050)	*.452*(.060)
Diabetes	.370(.073)	**.340**(.055)	**.341**(.053)	**.344**(.056)	**.343**(.057)	**.345**(.050)	.360(.056)	**.350**(.047)	.381(.073)	*.420*(.073)
German	.457(.048)	**.438**(.047)	**.441**(.043)	**.442**(.044)	**.440**(.038)	**.441**(.040)	**.443**(.042)	.449(.045)	*.471*(.034)	*.465*(.037)
Heart	.255(.111)	**.211**(.087)	**.199**(.071)	**.199**(.069)	**.200**(.069)	**.198**(.062)	**.189**(.052)	**.201**(.067)	.272(.118)	*.373*(.112)
Image	.350(.073)	.356(.073)	.360(.071)	.353(.070)	*.374*(.067)	*.375*(.067)	*.387*(.056)	*.398*(.051)	*.398*(.056)	*.415*(.063)
Ionosphere	.362(.110)	**.320**(.083)	**.312**(.077)	**.307**(.074)	**.302**(.068)	**.296**(.064)	**.301**(.067)	**.326**(.087)	.343(.090)	*.397*(.090)
Twonorm	.041(.050)	**.025**(.011)	**.025**(.010)	**.025**(.011)	**.025**(.011)	**.026**(.012)	**.028**(.012)	**.030**(.016)	.037(.019)	*.333*(.176)
Waveform	.194(.083)	.194(.062)	.204(.064)	.208(.070)	*.214*(.066)	*.218*(.068)	*.221*(.068)	*.204*(.065)	*.226*(.077)	*.343*(.133)

Senario 2

Benchmark	Initial	$\pi = 0.05$	$\pi = 0.1$	$\pi = 0.15$	$\pi = 0.2$	$\pi = 0.25$	$\pi = 0.3$	$\pi = 0.35$	$\pi = 0.4$	$\pi = 0.45$
Australian	.207(.085)	**.189**(.064)	**.186**(.060)	**.182**(.054)	**.180**(.050)	**.178**(.036)	**.176**(.030)	.187(.046)	*.303*(.120)	*.396*(.109)
Banana	.447(.037)	.447(.036)	.448(.033)	.448(.034)	.445(.038)	.447(.037)	.451(.039)	.443(.044)	.437(.049)	.444(.051)
Diabetes	.369(.058)	.361(.050)	.361(.049)	.363(.055)	.358(.051)	.361(.058)	.356(.049)	.369(.057)	*.388*(.064)	*.429*(.061)
German	.454(.042)	**.445**(.042)	**.440**(.044)	**.439**(.042)	**.442**(.041)	**.440**(.049)	**.440**(.041)	.449(.045)	*.464*(.039)	*.469*(.041)
Heart	.191(.060)	.192(.061)	.190(.065)	.187(.056)	.188(.049)	.197(.050)	.197(.058)	.199(.055)	*.244*(.087)	*.382*(.105)
Image	.390(.053)	.386(.046)	.389(.047)	.390(.043)	.388(.043)	.392(.040)	.397(.045)	.394(.045)	.408(.051)	.411(.057)
Ionosphere	.285(.057)	*.291*(.058)	*.293*(.058)	*.292*(.056)	*.294*(.064)	*.293*(.062)	*.301*(.066)	*.316*(.080)	*.345*(.089)	*.399*(.100)
Twonorm	.023(.011)	*.025*(.010)	*.025*(.011)	*.025*(.011)	*.026*(.011)	*.026*(.012)	*.028*(.013)	*.030*(.016)	*.053*(.081)	*.359*(.170)
Waveform	.186(.048)	*.195*(.055)	*.197*(.058)	*.202*(.057)	*.205*(.057)	*.204*(.056)	*.205*(.053)	*.208*(.066)	*.235*(.077)	*.324*(.129)

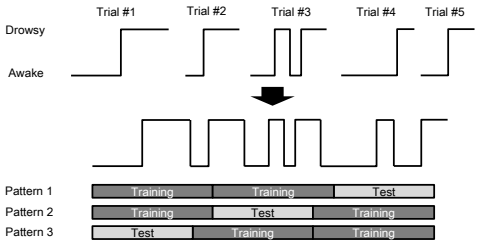

Fig. 1: The conceptual diagram of generating the datasets.

single *labeled* dataset for supervised classification. We set $t_{\max} = 20$ and $\pi = 0.1$ for iDSDD. As a baseline of supervised classification, the support vector machine (SVM) [23] was employed. LIBSVM [24] was used as an implementation. The settings of SVM were as follows: The Gaussian kernel was used. Hyperparameters, the cost parameter C and the kernel parameter γ, were selected from $\{0.5, 1, 2, 5, 10\}$ and $\{0.001, 0.01, 0.1, 1\}$ through 5-fold cross validation with grid search, respectively. These training and test sets may have class imbalance [25], so we also evaluated the SVM with class weights (referred to as wSVM hereafter). In this experiment, we computed the class weights by the ratio of the number of positive and negative samples to compensate for the class imbalance. The F-measure, which is the harmonic mean of the precision and recall, was used to evaluate the performance of each method.

Table 2 indicates the F-measures for each condition and the average for each method, where DSDD and iDSDD mean the initial and final iterations of iDSDD, respectively. The bold faces denote the best and comparative methods in terms of the average F-measure over nine conditions according to the paired t-test at a significance level of 5%. Surprisingly, the result of iDSDD did not only enhance the initial state, but stood comparison with the SVMs with/without class weights. This result indicates that iDSDD is promising both in improving the classification performance and reducing the labeling costs in drowsiness prediction.

Table 2: Comparison of the mean and standard deviation of the F-measure over 9 conditions. "Subj.1-1" means that the dataset was generated from Subject 1 by Pattern 1.

Dataset	DSDD	iDSDD	SVM	wSVM
Subj.1-1	.567	.573	.459	.600
Subj.1-2	.562	.625	.551	.512
Subj.1-3	.644	.835	.730	.718
Subj.2-1	.456	.761	.725	.725
Subj.2-2	.573	.600	.631	.732
Subj.2-3	.475	.644	.718	.684
Subj.3-1	.576	.570	.448	.587
Subj.3-2	.407	.446	.600	.485
Subj.3-3	.444	.691	.473	.634
Mean(Std)	.532(.079)	**.639**(.114)	**.593**(.116)	**.631**(.092)

5. CONCLUSION

In this paper, we proposed a practical unsupervised classification method called iterative UU (iUU) classification. We demonstrated the effectiveness of iUU classification for a real-world drowsiness prediction. We believe that this method is a promising way to reduce a burden for collecting labeled data in various applications. In future work, we will investigate other initialization methods and model representations and to add heuristics to further improve the performance.

The authors thank Masahiro Kato at the University of Tokyo for furitful discussion.

6. REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, The MIT Press, 2016.

[2] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National Science Review*, vol. 5, no. 1, pp. 44–53, 2018.

[3] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, 2009, pp. 1003–1011.

[4] H. Bao, G. Niu, and M. Sugiyama, "Classification from pairwise similarity and unlabeled data," in *Proceedings of 35th International Conference on Machine Learning*, 2018, pp. 452–461.

[5] O. Chapelle, B. Schlkopf, and A. Zien, *Semi-Supervised Learning*, The MIT Press, 1st edition, 2010.

[6] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of 20th International Conference on Machine Learning*, 2003, pp. 912–919.

[7] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proceedings of 10th International Workshop on Artificial Intelligence and Statistics*, 2005, pp. 57–64.

[8] 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 34th International Conference on Machine Learning*, 2017, pp. 2998–3006.

[9] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, no. 1-2, pp. 31–71, 1997.

[10] T. Cour, B. Sapp, and B. Taskar, "Learning from partial labels," *Journal of Machine Learning Research*, vol. 12, pp. 1501–1536, 2011.

[11] C. Elkan and K. Noto, "Learning classifiers from only positive and unlabeled data," in *Proceedings of 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 213–220.

[12] M. C. du Plessis, G. Niu, and M. Sugiyama, "Analysis of learning from positive and unlabeled data," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 703–711.

[13] T. Ishida, G. Niu, and M. Sugiyama, "Learning from complementary labels," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 5644–5654.

[14] M. C. du Plessis, G. Niu, and M. Sugiyama, "Clustering unclustered data: Unsupervised binary labeling of two datasets having different class balances," in *Proceedings of 2013 Conference on Technologies and Applications of Artificial Intelligence*, 2013, pp. 1–6.

[15] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967, vol. 1, pp. 281–297.

[16] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[17] M. Sugiyama, T. Kanamori, T. Suzuki, M. C. du Plessis, S. Liu, and I. Takeuchi, "Density-difference estimation," *Neural Computation*, vol. 25, no. 10, pp. 2734–2775, 2013.

[18] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Networks*, vol. 12, no. 1, pp. 145–151, 1999.

[19] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for adaboost," *Machine Learning*, vol. 42, no. 3, pp. 287–320, 2001.

[20] D. Dua and E. Karra Taniskidou, "UCI machine learning repository," 2017.

[21] H. Kaji, H. Yamaguchi, and M. Sugiyama, "Multi task learning with positive and unlabeled data and its application to mental state prediction," in *Proceedings of 2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 2301–2305.

[22] H. Kitajima, N. Numata, K. Yamamoto, and Y. Goi, "Prediction of automobile driver sleepiness (1st report, rating of sleepiness based on facial expression and examination of effective predictor indexes of sleepiness)," *Trans. of the Japan Society of Mechanical Engineers. C*, vol. 63, no. 613, pp. 3059–3066, sep 1997.

[23] V. N. Vapnik, *Statistical Learning Theory*, Wiley-Interscience, 1998.

[24] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, 2011.

[25] N. Japkowicz, "The class imbalance problem: Significance and strategies," in *Proceedings of 2000 International Conference on Artificial Intelligence*, 2000, pp. 111–117.