https://doi.org/10.2352/EI.2022.34.10.IPAS-381

Machine Learning with Blind Imbalanced Domains

Hiroshi Kuwajima; DENSO CORPORATION, Tokyo, Japan
Masayuki Tanaka, Masatoshi Okutomi; Tokyo Institute of Technology, Tokyo, Japan

Abstract

Recently machine learning is used in various applications and has shown success. Machine learning is good at learning the overall characteristics of massive training data. However, for real-world applications, training data often include multiple domains, and some domains have higher importance or risks. In this paper, we first propose a new problem setting: machine learning with blind imbalanced domains. In the proposed problem, the domain assignment of samples is unknown and imbalanced in the training data, and the performance is evaluated for each domain in the test data. Second, we propose an approach for that problem in classification tasks. The proposed approach combines center loss and weighted mini-batch sampling based on distances between samples and centroids in the deep feature space. Experiments on one minor domain and two minor domain settings using three handwritten digit databases (MNIST, EMNIST, and USPS) show that our proposed approach outperforms possible solutions using related methods. Remarkably our approach improves the accuracy in the minor domain by more than 1% on average. Furthermore, it can be inductively estimated that our proposed approach works on multiple domains given the successful results on one and two minor domains.

Introduction

Deep learning [1] techniques are rapidly advanced recently and becoming a necessary component for widespread systems. Deep networks are usually trained to minimize the average of sample losses. It means that the optimization process considers only major domain samples and neglects the minor domain samples.

In practice, training data contain samples from various domains. Domains include different individuals in handwritten character recognition, different locations and environmental conditions in automated driving, different translators in translation tasks, and different speakers with dialects and cadences in the speech recognition tasks. In industrial applications, small sample data are sometimes critical. For example, accidents, *e.g.*, in automated driving and credit authorization, are critical but rare cases. Those accident samples are much smaller than normal samples. For example, for automated driving, the data at rainy midnight are usually smaller than the data at sunny daytime. In contrast, the accident risk at rainy midnight is presumed to be much larger than that at the sunny daytime. We refer the minor domains to the domains associated with the small training samples. The major domains are the domains corresponding to the dominant training samples. Thus, it is essential to improve the performance on minor domains while maintaining that on major domains.

Figure 1 illustrates distributions of major and minor domains in the deep feature. The minor domain samples distribute far from the major domain samples in the typical random mini-batch generation [2, 3]. Then, the performance on the minor domain samples tends low. However, in the safety-critical systems, the performance on the minor domains is also essential. If the domain of each sample is known, then we can easily apply domain-balanced sampling during the training. However, in many practical situations, the domain information is blind.

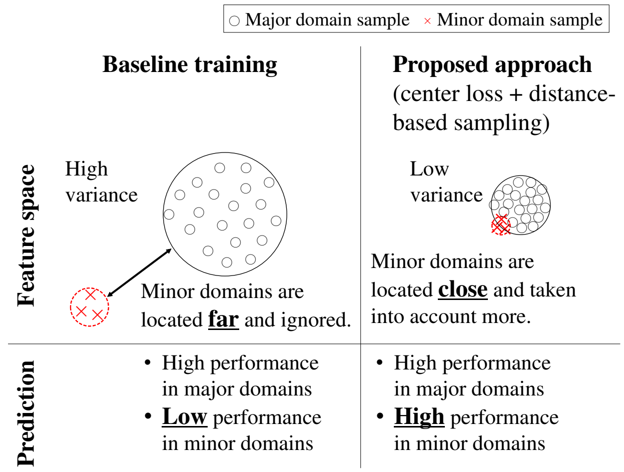


Figure 1: Machine learning with imbalanced domains.

In this work, we first mathematically define the problem of machine learning with blind imbalanced domains. Many domain adaptation techniques [4, 5] are only for the non-blind domain problem, in which we know the domain information of samples. It follows that we cannot apply such techniques to the blind domain problem. On the other hand, as mentioned above, if we can detect the domain information, we can apply domain-balanced sampling [6]. However, we will experimentally show that the minor domain sample detection using anomaly detection [7, 8] does not work well. Therefore in this work, we apply the center loss [9] and the deep feature distance-based sampling for a mini-batch generation to improve the performance of the blind minor domain samples. Our contributions are twofold:

- We introduce and formalize a new problem setting: machine learning with blind imbalance domains.
- We identify and empirically show that the combination of center loss and distance-based sampling is effective for the machine learning with blind imbalance domains in classification tasks.

This paper is organized as follows. First, *Related Works* section introduces center loss, distance-based sampling, and other related works. Then, we propose and formalize a new problem setting: machine learning with blind imbalanced domains, and observe the effect of domain imbalance in *Problem Setting and Ob-*

(a) Major domain samples (b) Minor domain samples

Figure 2: Examples of different domains in image processing.

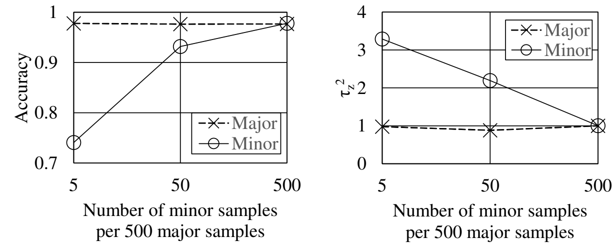
servation section. Based on the observation, we propose an effective countermeasure specialized for classification tasks in *Method* section. Then, *Experiments and Discussion* section demonstrates the advantage of our method through thorough experiments. Finally, *Conclusion* section summarizes our work and suggests future works.

Related Works

Center loss [9] is a regularizer to make samples and centroids (class means) in the deep feature closer. Contrastive center loss [10] is an extended center loss to maximize the deep feature variance between classes. Contrastive loss [11] selects positive, *e.g.*, same class, sample pairs and negative sample pairs. Triplet loss [12] selects triplets of 1) anchor samples, 2) positive samples, and 3) negative samples that make up positive sample pairs (1 and 3) and negative sample pairs (1 and 4). Then, contrastive loss and triplet loss minimize and maximize the distances between the deep features of the positive and the negative sample pairs, respectively. This paragraph shows that it is common to minimize and maximize the deep feature variance for similar and dissimilar samples, respectively. However, we are interested only in minimizing the deep feature variance in this work, and we use center loss.

Weighted sampling and loss weighting control the number of samples and significance of losses based on the characteristics of each sample, respectively. Hard negative mining [13] is a method to backpropagate only the selected hard samples. SMOTE [14] is a data augmentation [15] technique for imbalanced classes [16]. Hard negative mining aggressively selects real hard samples with large losses, whereas SMOTE generates augmented samples to compensate the class imbalance. However, selecting and generating only hard or minority samples in high-dimensional spaces suffer from concentration on the sphere [17] and noises [18]. Distance weighted sampling [18] directly addressed this problem by selecting negative samples at various distances. It uses sampling weight based on the inverse of sample probability. A loss weighting technique of focal loss [19] originally addresses the foreground-background imbalance in training object detectors by down-weighting already well-classified examples.

Domain adaptation [4, 5] is a research area to address the performance degradation when a machine learning model is trained in a source domain and tested in another target domain. In domain adaptation, we know the domain labels of samples, *i.e.*, training samples are always from the source domain, and test samples



(a) Performance (b) Domain Separation

Figure 3: Imbalanced domains in different numbers of minor samples.

are always from the target domain. We call such conditions a non-blind domain setting, and it does not apply to our blind domain setting. Deep supervised domain adaptation [4] is a supervised (class known) approach. It optimizes the feature extractor to minimize the distance between source and target samples with the same class closer and maximize that with different classes. Maximum classifier discrepancy [5] is an unsupervised (class unknown) approach. It assumes two classifiers for a shared feature extractor and optimizes the classifiers and the feature extractor to maximize and minimize the discrepancy between these classifiers, respectively.

Problem Setting and Observation

Here, we formalize the machine learning with blind imbalanced domains. Let a training sample, a label, and a domain label of the sample be x, y, and z, respectively. The joint probability of the training sample and the label with multiple domains can be expressed by a mixture distribution:

$$p(x,y) = \sum_{z=0}^{N_z-1} p(z)p(x,y|z), \tag{1}$$

where N_z is the number of domains. We say the non-blind domain if $p(z)$ is known. If $p(z)$ is unknown, then it is a blind domain problem. If the variance of $p(z)$ is small, then the distribution of domains is balanced. We say the imbalanced domains for the large variance of $p(z)$. In simple two-domain cases, $p(z=0) \gg p(z=1)$ is the imbalanced domain problem. If the domains are balanced, then $p(z=0) \simeq p(z=1)$. To evaluate the machine learning with blind imbalanced domains, we introduce the domain-wise performance PERF_z, which is the performance on a domain z.

To simplify the discussion, we consider only two domains, *i.e.*, a major domain and a minor domain. We focus on classification tasks as an example and emulate the multi-domain data with three handwritten digit databases MNIST [20], EMNIST [21], and USPS [22], with all images resized to 32×32. From those three datasets, we can generate six pairs of the major and minor domains. Figure 2 depicts an example pair of the major domain (MNIST) and the minor domain (EMNIST), for which we can observe different handwriting. For each domain pair, we train LeNet [23] with the activation function ReLU [24, 25] for handwritten digit recognition (classification task). The domain-wise accuracy ACC_z is an example of PERF_z in classification tasks. Figure 3a shows the average of six pairs of major and minor accuracies for the number of the minor domain samples while fixing the number of the major domain samples as 500.

$f_{\theta,\phi}(x) = (h_\phi \circ g_\theta)(x)$ denotes a trained network separated into a feature extractor g_θ and a classifier h_ϕ. We use LeNet as $f_{\theta,\phi}$, and the network from input to the second last full connection (F6) layer of LeNet as g_θ. Let $g_\theta(x)$ and $\mu_y = E_{x\sim p(x|y)}[g_\theta(x)]$ be a sample deep feature and the centroid deep feature of class y, respectively. We define the distance between a sample deep feature and the centroid deep feature as $d = \|g_\theta(x) - \mu_y\|_2$. We know the class y for a training sample x. Therefore, the above centroid distance d of a training sample x is computed based on the centroid deep feature of class y. Figure 4 shows three density plots for the number of the minor domain samples while fixing the number of the major domain samples as 500. Each plot is the density of centroid distance on major domain samples from MNIST, minor domain samples from EMNIST, and all samples.

Figures 3a and 4 show that the minor performance increases and the minor domain samples locate close to the centroid when the number of the minor domain samples increases. Therefore, minor performance and the distance between the minor domain sample and the centroid are correlated. We define domain separation τ_z^2 to evaluate the closeness of domain samples as the normalized second order central moment [26] for domain z. We define the relative domain separation of the domain z as

$$\tau_z^2 = \frac{E_{x,y\sim p(x,y|z)}\left[\|g_\theta(x) - \mu_y\|_2^2\right]}{E_{x,y\sim p(x,y)}\left[\|g_\theta(x) - \mu_y\|_2^2\right]} \quad (2)$$

for classification tasks. Figure 3b shows the average of six pairs of major and minor domain separations for the number of the minor domain samples while fixing the number of the major domain samples as 500. Figures 3a and 3b show a clear negative correlation between the performance and the domain separation.

Method

As the previous section shows, balancing major and minor samples is critical in the imbalanced domain cases. In the non-blind situation, we can apply weighted data sampling to balance the imbalanced domains. However, in the blind imbalanced domain cases, we cannot apply the balanced sampling directly because the domain of each sample is unknown. The straightforward approach is a combination of anomaly detection and balanced sampling. In such an approach, samples are classified into the major and minor domains by anomaly scores. Then, we can apply balanced sampling based on classified domains. However, this straightforward approach does not work well, to be shown in *Experiments and Discussion* section, since anomaly detection of minor domains is not easy. This section builds a practical approach for machine learning with blind imbalanced domains in classification tasks.

The previous section also showed a negative correlation between the performance and the domain separation, i.e., variance. Thus in our approach, we minimize the variance of deep features instead of classifying the domains. For that purpose, we use center loss and distance-based sampling [27]. Center loss reduces the variance in the deep feature as introduced in *Related Works* section. The purpose of distance-based sampling is to pick up many samples in minor domains. We saw that the minor domain samples locate far from centroids in *Problem Setting and Observation* section. We assume that the samples far from the centroid in the deep feature have high probabilities of being in the minor domain. With higher weights for the samples far from the centroid in the deep feature, distance-based sampling generates mini-batches expected to contain such samples [28]. For that purpose, we first model sample probability $q(d)$ as a function of centroid distance d. Then, we hold the centroid distance d for all samples throughout training and estimate the model parameters of $q(d)$ based on it. If we select Gaussian distribution, we estimate the sample mean $\bar{d}$ and the sample variance s_d^2 from d; if we select exponential distribution, we estimate the rate parameter λ_d from d. Then, we generate a mini-batch $\mathscr{B}$ with sample weights $q(d)^{-1}$, the inverse of sample probability, so that we uniformly select samples both from major and minor domains under the blind domain setting. Finally, we update only a specific part of d corresponding to $\mathscr{B}$ avoiding recalculation of entire d. In typical machine learning, if the domains are imbalanced, then τ_z^2 slowly decreases for small $p(z)$ because $p(x,y|z)$ is discounted. In our method, center loss decreases τ_z^2 regardless of domains; then, distance-based sampling increases $p(z)$ to decrease τ_z^2 for minor domains z.

Algorithm 1: Combining center loss and distance-based mini-batch sampling for the machine learning with blind imbalanced domains in classification tasks.

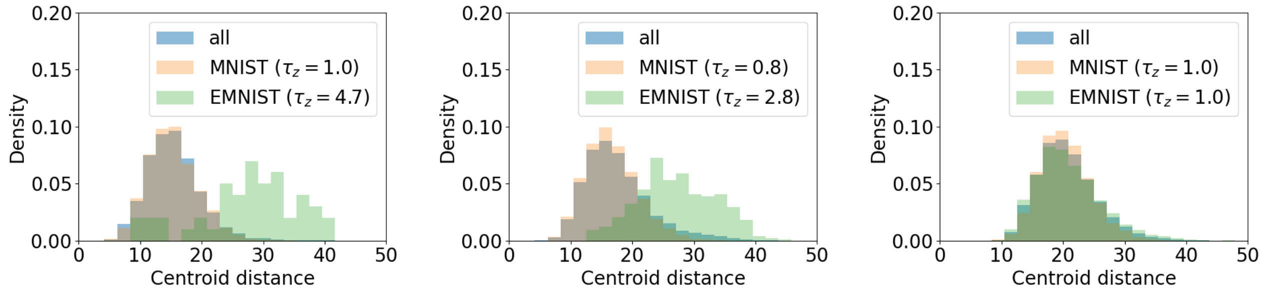
Input: training data $\{(x_i, y_i)\}$, network $f_{\theta,\phi} = h_\phi \circ g_\theta$, momentum coefficient α

Output: network parameters θ (feature extraction) and ϕ (classification)

```
initialize θ and φ
initialize centroid deep feature {μ_c}
initialize centroid distance {d_i}
repeat
    estimate parameters of distribution q(d) from {d_i}
    sample mini-batch 𝓑 = {(x_j, y_j)} with weight
      q(d_j)^{-1}
    update θ and φ w.r.t. classification loss and center
      loss on 𝓑
    update μ_c w.r.t. center loss on 𝓑
    update d_j ← α d_j + (1 − α) ||g_θ(x_j) − μ_{y_j}||_2 for each
      (x_j, y_j) ∈ 𝓑
until training ends;
```

Now i, j, and c denote the indices of all training samples, the indices of the training samples in the mini-batch $\mathscr{B}$, and the indices of classes, respectively. In a training iteration, we update network parameters θ and ϕ, centroid deep features for all classes $\{\mu_c\}$, and the centroid distances $\{d_j\}$ only for the samples in the mini-batch $\mathscr{B}$. First, we update the network parameters θ and ϕ by backpropagating classification loss, *e.g.*, softmax [29] cross entropy loss, and center loss $\frac{1}{2}\sum_{j\in\mathscr{B}} \|g_\theta(x_j) - \mu_{y_j}\|_2^2$. Next, we update $\{\mu_c\}$ through backpropagation of center loss. Then, we update $\{d_j\}$ only for the samples in $\mathscr{B}$ based on $\|g_\theta(x_j) - \mu_{y_j}\|_2^2$ with momentum. We apply momentum with coefficient α to the centroid distance to avoid oscillations. We show the pseudo code of our method in Algorithm 1.

Regular SGD (stochastic gradient descent) algorithm is sampling *without* replacement [30] where once samples are selected, the sampler will not select these samples again. SGD also ensures the selection of all samples in an epoch. On the other hand, our



(a) 5 minor samples (b) 50 minor samples (c) 500 minor samples

Figure 4: Number of minor samples and transition of centroid distance.

Table 1: Results in the two-domain setting.

		M/E	E/M	M/U	U/M	E/U	U/E	Average
Random	Major	0.9835	0.9839	0.9832	0.9647	0.9849	0.9655	0.9776
	Minor	0.6153	0.6148	0.9342	0.9186	0.7141	0.6466	0.7406
Input LOF sampling	Major	0.9822	0.9837	0.9842	0.9639	0.9851	0.9636	0.9771
	Minor	0.6276	0.6420	0.9332	0.9223	0.7115	0.5911	0.7380
Feature LOF sampling	Major	0.9838	0.9844	0.9844	0.9657	0.9845	0.9644	0.9779
	Minor	0.6219	0.6393	0.9321	0.9089	0.7136	0.5998	0.7359
Cross entropy sampling	Major	0.9817	0.9850	0.9852	0.9635	0.9853	0.9630	0.9773
	Minor	0.6319	0.6292	0.9331	0.9135	0.7228	0.6079	0.7397
Distance-based sampling	Major	0.9834	0.9834	0.9828	0.9639	0.9846	0.9604	0.9764
	Minor	0.6287	0.6188	0.9231	0.9050	0.6908	0.6065	0.7288
Focal loss	Major	0.9834	0.9818	0.9820	0.9641	0.9848	0.9644	0.9768
	Minor	0.6186	0.6225	0.9321	0.9162	0.7204	0.6324	0.7404
Center loss	Major	0.9903	0.9909	**0.9911**	0.9692	**0.9907**	0.9706	0.9838
	Minor	0.6976	0.7046	**0.9377**	0.9424	0.8002	0.7621	0.8074
Center loss + distance-based sampling	Major	**0.9910**	**0.9911**	0.9903	**0.9711**	0.9906	**0.9709**	**0.9842**
(proposed)	Minor	**0.7330**	**0.7372**	0.9363	**0.9493**	**0.8242**	**0.7632**	**0.8239**

approach uses weighted sampling to intend a single minor sample selected multiple times, and we opt for sampling *with* replacement [31]. Thus Algorithm 1 does not consider batch and epoch numbers. Instead, we define the number of iterations for an epoch as the training dataset size divided by the batch size to track the training progress.

Experiments and Discussion

As shown in *Method* section, our method consists of center loss and distance-based sampling. In this section, we compare our proposed method with combinations of existing approaches. We configure the combinations of loss methods and sampling methods. Loss methods include focal loss [19] and center loss [9]; sampling methods use local outlier factor (LOF) [7] on input and deep feature, cross entropy loss, and centroid distance. Finally, we confirm the overall performance of our method and the effect of center loss and distance-based sampling through experiments.

Besides the blind domain setting, we use a similar experimental setting in *Problem Setting and Observation* section. We train LeNet with ReLU for handwritten digit recognition (classification task) and use the deep feature $g_\theta(x)$ at the F6 layer. We perform experiments in pairwise and triplet domains in MNIST, EMNIST, and USPS datasets abbreviated as M, E, and U. An example pairwise domain M/E and an example triplet domain M/E,U denote a major domain MNIST with a minor domain EMNIST and a major domain MNIST with two minor domains EMNIST and USPS, respectively. We select the M/E pair as the representative domain setting for drawing figures. Major domains have 500 samples/class, and minor domains have 5 samples/class, *i.e.*, we have approximately $5,000$ images in total. We measure the accuracy after 100 epochs with batch size 128, *i.e.*, the number of iterations is approximately $5,000/128$ per epoch.

Now we describe the hyperparameters and design alternatives of compared methods. We empirically selected exponential distribution to model $q(d)$ in distance-based sampling. Exponential distributions also model LOF scores and cross entropy losses. We use momentum with coefficient $\alpha = 0.9$ to update $d(x)$. We compute the input LOF scores on the inputs (samples) x only once at the beginning of training. In contrast, we compute the feature LOF scores on the deep features $g_\theta(x)$ for each epoch because θ

Table 2: Results in the three-domain setting.

		M/E,U	E/M,U	U/M,E	Average
Random	Major	0.9827	0.9841	0.9646	0.9771
	Minor	0.6294 0.9250	0.6810 0.7332	0.8939 0.6556	0.7725
Input LOF sampling	Major	0.9831	0.9848	0.9644	0.9774
	Minor	0.6109 0.9180	0.6871 0.7393	0.8768 0.6149	0.7664
Feature LOF sampling	Major	0.9818	0.9835	0.9625	0.9759
	Minor	0.6150 0.9230	0.6616 0.7242	0.8855 0.6390	0.7619
Cross entropy sampling	Major	0.9822	0.9849	0.9621	0.9764
	Minor	0.6283 0.9219	0.6960 0.7415	0.8868 0.6340	0.7749
Distance-based sampling	Major	0.9821	0.9842	0.9623	0.9762
	Minor	0.6233 0.9250	0.6648 0.7373	0.8900 0.6246	0.7681
Focal loss	Major	0.9831	0.9838	0.9634	0.9768
	Minor	0.6063 0.9198	0.7122 0.7514	0.8934 0.6507	0.7766
Center loss	Major	0.9904	0.9906	0.9704	0.9838
	Minor	0.7070 **0.9331**	0.7493 0.8077	**0.9336** 0.7526	0.8261
Center loss +distance-based sampling	Major	**0.9910**	**0.9907**	**0.9705**	**0.9840**
(proposed)	Minor	**0.7340** 0.9307	**0.7796** **0.8236**	0.9279 **0.7818**	**0.8392**

is updated. Focal loss uses a focusing parameter $\gamma = 2$.

Tables 1 and 2 show the accuracy for each pair and triplet in the two-domain setting, *i.e.*, one major domain and one minor domain, and the three-domain setting, *i.e.*, one major domain and two minor domains, respectively. All single experiments are executed 4 times and averaged. In minor accuracy, our proposed approach performs best with significant improvements (2.0% to 3.5%) on more than half of pairs and triplets in the two- and three-domain settings; it is comparable (within $\pm 0.6\%$ to the best method) in the rest settings. In major accuracy, we achieved the best major accuracy except for 2 pairs in the two-domain setting (degradation was only 0.01% and 0.08% for these 2 pairs). Although distance-based sampling, which discounts major samples, is superficially regarded as harmful to the major accuracy, experimental results confirm no significant performance loss for the major domains. Also, on average, our approach outperformed all the other methods in minor accuracy by a large margin (1.65% in the two-domain setting and 1.57% in the three-domain setting) and in major accuracy. Therefore, experiments show that our approach is effective in machine learning with blind imbalanced domains. It can be inductively estimated that our proposed approach works for multiple domains, given the experimental results on the two- and three-domain settings. We discuss the detail of the experimental results in the following paragraphs. Experiments ensure that the simplest approach using LOF sampling to detect minor domains for domain-balanced sampling does not work.

In Table 1, our proposed approach outperformed all other methods on M/E, E/M, and E/U with significant minor accuracy improvements of 3.5%, 3.2%, and 2.4%, respectively. On M/U, U/M, and U/E, our proposed approach performs comparably to other methods with minor accuracy deviation between -0.2% and 0.7%. In Table 2, our approach outperformed all other methods on E/M,U with considerable minor accuracy improvements of 3.1% and 1.6%. On M/E,U, and U/M,E, the accuracy of one minor domain is substantially improved (2.7% and 2.9%), but that of the other minor domain was slightly degraded (-0.24% and -0.57%).

We confirmed the effects of center loss and distance-based sampling separately. In the averaged results at the rightmost columns in Tables 1 and 2, center loss performs best among baseline (random sampling with classification loss only) and focal loss. On the other hand, in most conditions, distance-based sampling is not better performed among other sampling methods, *i.e.*, input LOF sampling, feature LOF sampling, and cross entropy sampling. However, in our proposed approach, the combination of center loss and distance-based sampling outperforms all other methods on average for major and minor domain accuracy. Notably, applying distance-based sampling in addition to center loss improves average minor domain accuracy by more than 1.5%. We observe that distance-based sampling works very well only after bringing samples closer together for a sharp contrast.

Conclusion

This paper introduced a new problem setting, machine learning with blind imbalanced domains, and formalized it. In that problem, we assume that the training data consist of imbalanced samples from different domains. The practical problem is to improve the performance on minor domains as well as that on major domains because high-risk minor domains have importance in specific kinds of applications, *e.g.*, safety-critical systems. Then, we proposed an effective approach for the problem on classification tasks, the combination of center loss and distance-based mini-batch sampling. Our approach outperformed other relevant approaches in the accuracy on minor domains with significant improvement for more than half of the experimental settings without hurting that on major domains.

Future works include 1) building theorem-proof of the analytical advantage of our proposed approach in machine learning with blind imbalance domains, 2) causal analysis on the datasets where our approach worked very well, and it just performed comparably, 3) adding relevant data augmentation in distance-based mini-batch sampling instead of simply oversampling minor samples, and 4) experiments using more realistic datasets such as minor accident samples in major regular driving samples for automated driving systems.

References

[1] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[2] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[3] Jack Kiefer, Jacob Wolfowitz, et al. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466, 1952.

[4] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5715–5725, 2017.

[5] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018.

[6] Paula Branco, Luis Torgo, and Rita P Ribeiro. Rebagg: Resampled bagging for imbalanced regression. In *Second International Workshop on Learning with Imbalanced Domains: Theory and Applications*, pages 67–81. PMLR, 2018.

[7] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

[8] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[9] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016.

[10] Ce Qi and Fei Su. Contrastive-center loss for deep neural networks. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2851–2855. IEEE, 2017.

[11] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[12] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, pages 84–92. Springer, 2015.

[13] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 118–126, 2015.

[14] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[15] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

[16] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[17] Frederik J Simons, FA Dahlen, and Mark A Wieczorek. Spatiospectral concentration on a sphere. *SIAM review*, 48(3):504–536, 2006.

[18] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2840–2848, 2017.

[19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[20] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[21] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[22] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[23] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[24] Kunihiko Fukushima. Visual feature extraction by a multilayered network of analog threshold elements. *IEEE Transactions on Systems Science and Cybernetics*, 5(4):322–333, 1969.

[25] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

[26] Geoffrey Grimmett and David Stirzaker. *Probability and random processes*. Oxford university press, 2020.

[27] Mahito Sugiyama and Karsten Borgwardt. Rapid distance-based outlier detection via sampling. *Advances in Neural Information Processing Systems*, 26:467–475, 2013.

[28] Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *international conference on machine learning*, pages 1–9. PMLR, 2015.

[29] Christopher M Bishop. Pattern recognition. *Machine learning*, 128(9), 2006.

[30] Daniel G Horvitz and Donovan J Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685, 1952.

[31] Yves Tillé. *Sampling algorithms*. Springer, 2006.

Author Biography

Hiroshi Kuwajima received his master's degree from Osaka University, Osaka, Japan in 2008 and joined Microsoft Development, Tokyo, Japan. He was a visiting researcher at Stanford University, CA, USA from 2013 to 2015. He is currently a project manager at DENSO CORPORATION, Aichi, Japan.

Masayuki Tanaka received his Ph.D. degree from Tokyo Institute of Technology, Tokyo, Japan in 2003 and joined Agilent Technology. He was a research scientist at Tokyo Institute of Technology from 2004 to 2008, a visiting scholar at Stanford University from 2013 to 2014. He is currently an associate professor at Tokyo Institute of Technology.

Masatoshi Okutomi received his master's degree from Tokyo Institute of Technology in 1983 and joined Canon Inc., Tokyo, Japan. He was a visiting research scientist at Carnegie Mellon University, PA, USA from 1987 to 1990. He received his Ph.D. degree from Tokyo Institute of Technology in 1993. He is currently a Professor at Tokyo Institute of Technology.