Solve lack of label data : Partial classification by geometric attribute of coreset selection

Hyeongu Kang [*1]

Abstract

The performance of the neural network model cannot be guaranteed if label data is insufficient. There are studies that have been solved without relying on the neural network model in the lack of label data, but there are unrealistic assumptions that they already know about prior knowledge of dataset. In this study, we solve it by pseudo-labelling the unlabeled data through high-accuracy classification that does not require a learning process. We propose a new classification method using coreset selection's geometric attribute. This method classified 10 to 40 times of unlabeled data with 95% or more accuracy by labeled data for small image datasets.

1. Introduction

Recently, deep learning (DL) model has been achieving performances in various fields based on a large amount of labeled data. However, as the DL model increases in the data required for learning, how to solve the labeling cost has become an important topic. Semi-supervised learning (SSL) is one of the ways to solve labeling cost, which assumes insufficient label data and many unlabeled data situations. This study focuses on image classification through SSL.

SSL is largely divided into consistency regularization and pseudo labeling(1). In this paper, we focus on pseudo labeling. Consistency regularization is preferred due to its high performance in recent image classification studies. However, both are necessary because dataset's diversity characteristic is different that can be obtained through each method. Consistency regularization has limitations in directly utilizing unlabeled data, even though there are various augmentation method for label data. On the other hand, pseudo labeling may have confirmation bias, but the information of the unlabeled data can be directly used. Two methods are not a contradictory method. The potential of pseudo labeling can be seen through a study(2) that solved confirmation bias well and performed better than consistency regularization.

In pseudo labeling, it is important to prevent the confirmation bias well. Confirmation bias means that the incorrect prediction about the unlabeled sample would deteriorate the performance of the neural network model. Existing studies introduce a threshold for reliability for the prediction of DL models(3), a regulatory term for soft labeling(4). Most measures assume that the neural network has sufficient accuracy. However, the accuracy of the neural network model is greatly reduced when the label data is insufficient. As an example, when 13 CNNs are applied to MNIST datasets, an accuracy of model is just 30% when 100 label data is given. Measures to prevent confirmation bias are also meaningless when the performance of the model itself cannot be guaranteed. One study successfully prevents confirmation bias by using prior knowledge about ratios of classes as regulatory terms by assuming that the dataset will be class-balance dataset(2). However, it is difficult to expect to know whether the dataset is class-balance, or prior knowledge of class distribution. Therefore, more realistic methods are needed.

This study proposes a classification method that does not require a learning process through the coreset-selection, which is one of active learning. Active learning is a method of sampling data that is useful for learning a model. Coreset selection does not rely on neural network-based models by utilizing distance information from data. Furthermore, it will be shown that each sampled data through the coreset selection is representation of the unlabeled data from a geometric perspective. Lastly, classification will be conducted based on geometric relationships. At this time, the unsupervised representation learning will be applied to simplify the geometric relationship of the subgraph because active learning is difficult to apply to high-dimensional data. In this study, the convolution autoencoder (CAE) is applied to reflect the structural information of image data. The whole process is shown in Figure 1. First, feature extraction is performed with CAE for each image data. Coreset selection is performed through the feature, and classification will be performed through the geometric relationship of subgraphs. After that, the DL model will be trained through data from coreset selection and new classification. Shortage of label data is solved by retaining classifed unlabeled data with high accuracy.

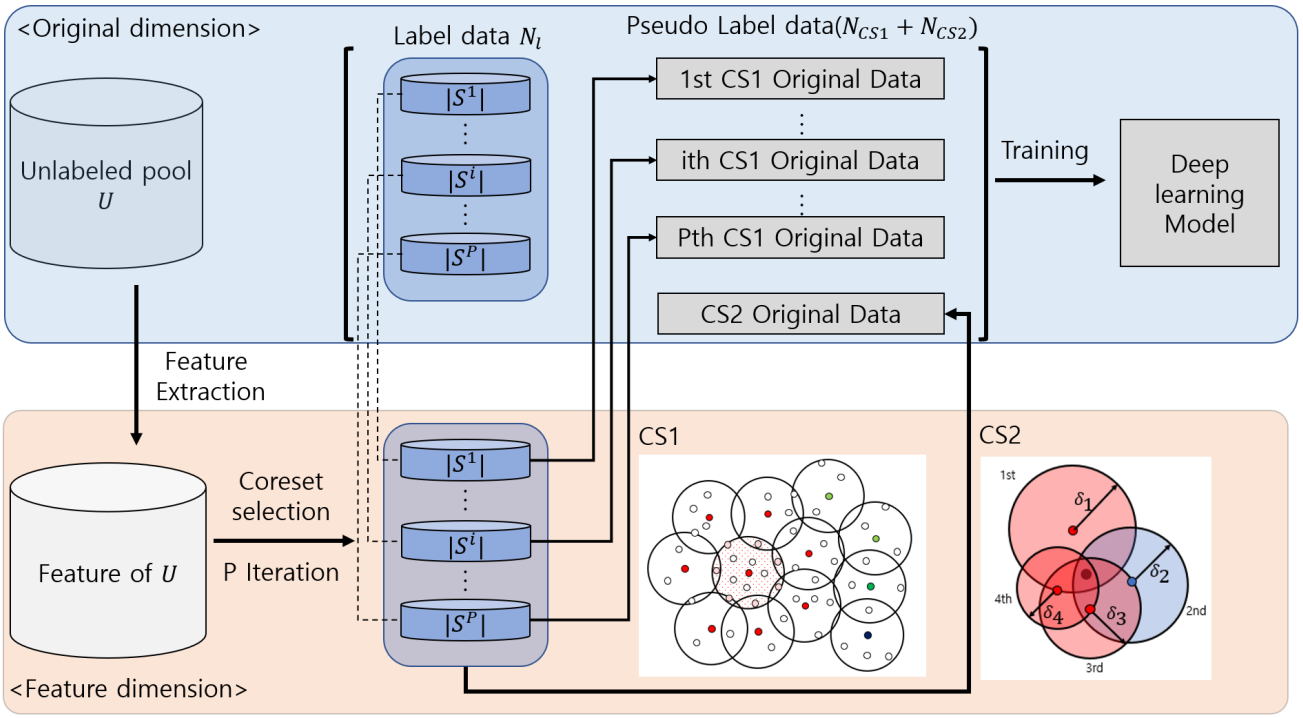

Figure 1. Whole process of solving lack of label data

2. Related work

2.1. Pseudo labeling

Pseudo labeling utilizes predictions about unlabeled data for DL model training. In the early stage when confirmation bias was not solved, pseudo labeling was limited to fine-tuning of the model(5). Since then, various measures have been taken to solve the confirmation bias, such as introducing uncertainty weight(6), and soft labeling. There is a limitation that each measure relies on a neural network-based model that requires learning. The performance of the neural network based model is unreliable when label data is insufficient. Existing methods are meaningful only when the model is sufficiently learned and performance is guaranteed. There is a way to resolve the confirmation bias without relying on the neural network model. In addition to soft labeling, Arazo's study effectively prevented confirmation bias well by using mix-up data augmentation, minimum batch, and utilize dataset's class ratio information by regularization term(2). This study performed better than consistency regulation on CIFAR 10/100, SHVN datasets. However, there are limitations in Arazo's research. In this study, by assuming a class-balance scenario, the class ratio information of the dataset is utilized as a regulatory term. This method cannot be applied if prior knowledge of the class ratio of the dataset isn't given.

2.2. Active learning

Active learning(7)(AL) is also a method of reducing the labeling cost. AL samples the most useful data from the unlabeled dataset to reduce the labeling cost as much as possible while maintaining performance. There are three main types of AL(8). This study focuses on pool-based sampling that selects important data from a given dataset. Pool-based sampling AL selects data according to the acquisition strategy. Acquisition strategy is divided by uncertainty-based, expected-based and diversity-based approach. Uncertainty-based approaches measure uncertainty for each data according to the neural network model(9)(10). The expect-based approach also selects the data that is expected to improve the model performance the most (11), and relies on the neural network model. Diversity-based approach is a method of screening data that can guarantee the diversity of a dataset. The most representative method is a coreset selection(11). Coreset selection proved that selecting center of subgraphs with minimum radius covering all data is same as maximizing performance of model. While most ALs rely on the performance of neural network models, corset selection relies only on distance information between data. In this study, a subgraph formed between coreset selection will be utilized.

AL has a limitation that it is hard to apply to high-dimensional data, and does not have the ability to calculate low-dimensional features from high-dimensional data(7). There are studies that apply low-dimensional feature ex-

traction through DL to address this limitation. DL models are easy to apply to high-dimensional data and extract feature without human intervention. In the case of DBL, autoencoder extract low dimentional feature from high dimentional data and then sampling data which is representing dataset(12). In the case of CEAL, active learning is used for sampling the unreliable data of CNN's model, and pseudo-labeling is performed only when the degree of reliability is above a threshold(13). For BCNNs, the bayesian neural network model is utilized to calculate uncertainty and use it as an acquisition strategy(9). However, the above studies become unreliable as performance of the neural network model unreliable. Acquisition strategy based on a neural network model that has not been sufficiently learned is not reliable.

There are studies that extract the feature of high-dimensional data through the unsupervised deep learning method and then apply AL. Prior works mainly extract low-dimensional image data through autoencoder(12)(13). VAL samples data focusing on diversity in latent space through a variational autoencoder(14). This method can produce its original performance even if label data is insufficient. Therefore, in this study, we will also apply feature extraction through unsupervised model to apply high-dimensional image data.

3. Method

Notation is refered from Arazo's research(2). Let unlabeled set $D_u = \{x_i\}_{i=1}^{N_u}$ and labeled set $D_l = \{(x_i, y_i)\}_{i=1}^{N_l}$ s.t. $N = N_l + N_u$. The two new classification methods presented in this study will be called CS (Coreset selection-based classification) 1 and CS2, respectively. CS1 performs classification in the form of a hard label by utilizing the geometric relationship between subgraphs in contact. CS2 calculates a class- probability vector h_{CS2} according to the degree of overlap of subgraphs for each data x_i when performing multiple iterations of the coreset selection. Details will be described in later.

To prevent confirmation bias, we apply Mix-up data augmentation and setting a minimum number of labeled samples per mini-batch. Mix-up data augmentation is a powerful regulatory method that combines data augmentation and label smoothing(4). Confirmation bias is well prevented when Mix-up data augmentation and setting a minimum batch size of labeled samples are applied simultaneously(2). We will make a difference in the number of epoch and batch size between label data and pseudo labeled data by CS1 and CS2 during CNN model learning.

3.1. Coreset selection

Coreset selection samples data that maximizes the expected model performance. This is the same as sampling data that have a minimum radius δ when constructing subgraphs that can cover the entire data with a given sampling size N_l(11). Each sampling data is widely spread, regardless of the density of the dataset. For this reason, sampling point u_i reflects an overall dataset. Coreset selection only depend on the distance between the data. Therefore, coreset selection's performance are maintained even when label data is insufficient.

AL has difficulty expanding to high-dimensional data such as images and text(7). For this reason, most studies on AL focus on low-dimensional problems or utilize features of high-dimensional data. To apply AL in the latter case, since AL does not have the ability to extract features of data, it is necessary to apply additional methods. Since DL have the ability of feature engineering of high dimensional data without human intervention, DL is utilized for AL.(8).

3.2. Dimension reduction

Dimension reduction through feature extraction is required to apply coreset selection to high-dimensional image data. The CS1 method requires two-dimensional or three-dimensional features in that it utilizes the adjacency of subgraphs. CS1 can be applied when class of center whose encountered subgraph are all the same. The process will be described later. The number of being able to classified N_{CS1} have a trade-off relationship with accuracy of CS1. The higher the number of subgraphs encountered, the stronger the conditions and the higher the precision. However, N_{CS1} decreases as the conditions are strengthened. The dimension of the feature controls the number of subgraphs encountered. In the case of two dimension, the number of subgraphs available to contact for each subgraph is at most 8, but in the case of three dimensions, the number increases to 20 to 30. As the dimension increases, the number of encountered subgraphs increases exponentially. Even if MNIST data of 784 dimensions is reduced to 10 dimensions, it is still high. In this study, the dimension of the feature is reduced to two dimensions.

In this study, convolution autoencoder(CAE) is used to reduce dimensions. In this study, it is assumed that when data are mapping to low dimension by CAE, the information of each class will be well-preserved and clustered for each class. Autoencoder not only capture the repetitive structure but also play the role of dimension reduction(15). In addition, when classification was performed for features by the k-nearest neighbor, CAE was one of the most accurate unsupervised methods with an accuracy of about 85 percent(16). From this, we can infer that CAE makes features clustering for each class.

3.3. Coreset selection based classification (CS1 & CS2)

Extract the two-dimensional feature of train dataset by CAE. After that, the coreset selection is performed based on the distance between features. At this time, the euclian distance is applied. N_l labeled data are the results of sampling through coreset selection. When a total of p coreset selections are performed, Sp s.t. $p \in \{1, \ldots, P\}$ is called a data set sampled in p_{th} iterations of coreset selection, and $u_i^p \in S_p$ s.t. $i \in \{1, \ldots, |S_p|\}$ are each sampling point. In this case, $N_l = \sum_{i=1}^{P} |S_i|$ is satisfied. Let the radius of subgraph G_i^p centered on each u_i^p be δ_p. Coreset selection is the same as K-centers algorithm. It samples the unlabeled data $x_k \in D_u/S_p$ as u_{i+1}^p, which minimizes the radius δ_p of subgraph G_{i+1}^p. Thus, δ_p is $max_{i,k} dist(u_i, x_k)$ s.t. $i \in \{1, \ldots, N_l\}$ and $k \in \{1, \ldots, N_u\}$. The density of each subgraph G_i^p would be measured by the number of data x_k^p $\in$G$_i^p$.

The CS1 method is classified through a geometric relationship between subgraphs. For convenience, we will simplify the notation to $S_p = S, u_i^p = u_i, G_i^p = G_i, \delta_p = \delta$. SSL assumes that the data x_1, x_2 are close in dense region, then the associated label y_1, y_2 are also similar(17). Data x_{ij} belonged to G_i can be assumed that label is same as u_i's label s.t. $j \in \{1, \ldots, n_i | n_i$ is the number of unlabel data in $G_i\}$. Thus, u_i could represent unlabeled data x_{ij}. Furthermore, we assump that a radius δ is small enough to cover the dataset tightly. Let G_{ik} be the subgraph which has connection with G_i s.t. $k \in \{1, \ldots, m_i | m_i$=number of subgraph which has connection with G_i. At this time, if the classes of u_i and u_{ik} are different, it can be inferred that G_i and G_k are in the boundary area of different classes from a geometric perspective. Contrariwise, it will be located in the center of a particular class when all the classes of u_i^k and u_i are the same. That is, when $u_i = u_{i1} = \ldots = u_{im_i}$ and density of subgraph $G_i > \alpha$ is satisfied, the unlabeled data x_{ij} belonging to subgraph G_i can be classified as same as u_i. is the hyperparameter for the subgraph density. In addition, when a few subgraphs overlap, it is necessary to prevent accidental misclassification when u_i is not representation of subgraph G_i but all of encountered subgraph's class is same. The number of encountered subgrphs m_i is also added as a condition to be greater than or equal to the hyperparameter M.

CS2 can be applied when subgraphs overlap by performing the coreset selection multiple times. All unlabeled data x_k are included in one or more subgraphs at one iteration. If the coreset selection is performed a total of P times, all unlabeled x_k will be belonged to at least P subgraphs. The more x_k belongs to a subgraph whose center is a specific class, the higher the probability that x_k have the same class. In addition, the probability are inverse proportion to the radius δ_p of each subgraph. For each class, we calculate

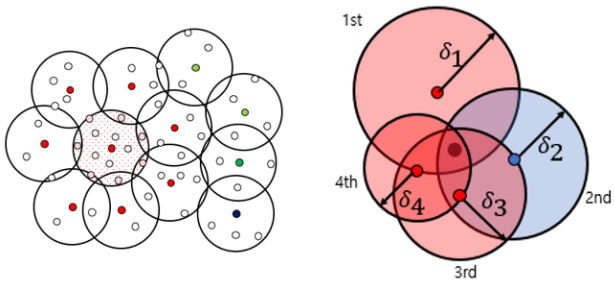

Figure 2. Concept of CS1(Left) and CS2(Right)

probability of class via Softmax based on the number of times $s^p = (s_{c_1}^p, \ldots s_{c_J}^p$ s.t. J is the number of class) and δ_p. We use ratio of radius δ^P compared to $min(\delta^P)$ to prevent the size of the radius from being very large and the probability value of each become low. To give weight to the recent iteration, we will add the weight of $\frac{i}{P}$ for the i_{th} Iteration. Hard label is given only when the highest probability is greater than threshold β

$$p(y = c_j | x_k^p) = \frac{\sum_{p=1}^{P} e^{s_{c_j}^p / \delta^{p'}}}{\sum_{j=1}^{J} \sum_{p=1}^{P} e^{s_{c_j}^p / \delta^{p'}}} [3.1]$$

$$s.t. \delta^{p'} = \frac{\delta^p}{min(\delta^p)} * \frac{i}{P}, k \in \{1, ..., K^P\}$$

4. Experiment

4.1. Datasets and training parameter

We will verify the performance of CS1 and CS2 with five datasets: MNIST, Fashion-MNIST, EMNIST-letter, and CIFAR10/100. Each dataset has 10 classes, except EMNIST-letter and CIFAR 100. EMNIST-letter and CIFAR 100 have 26 and 100 classes, respectively. MNIST and Fashion-MNIST consist of a total of $60K$ training images and 10K test images, respectively. EMNIST-letter consists of $124.8K$ and $20.8K$. Images of three datasets have a resolution of 28x28. CIFAR 10/100 consists of $50K$ and $50K$ training images and 10K and 10K test data, respectively. Image of dataset are color and have a resolution of 32x32. The case of sampling at once for $N_l = 1K$ and the case of sampling sampling 100 data for 10 iteration are tested. The result of each dataset may partially vary depending on the performance of CAE, so the number of detailed N_l cases will be checked only with MNIST. The CAE model utilizes a well-known 13-CNN model structure, and the decoder part is configured in reverse order. The DL model is 13 CNN.

The same hyperparameters were applied for each dataset. In the case of the convolution autoencoder, lr = 0.001 and Adam is applied as an optimizer to learn train dataset for 100 epochs. After that, in learning the 13-CNN model, label dataset and pseudo labeled dataset by CS1 and CS2

were separately learned. When we train 13-CNN model, we applied the same hyperparameters. According to the research of Arazo(2), the value of hyperparameter of mix-up data augmentation was set to 4. In addition, the train batch size of label dataset was set to 8 and the train batch size of CS1 and CS2 was set to 100, respectively. In addition, 13-CNN model are trainbed by labeled data for 5 epoch first, and then utilize pseudo labeled data classified by CS1 and CS2 for 10 epoch.

4.2. Performance of CS1 & CS2

The performance of CS1 and CS2 for each dataset is shown in Table 1. There are no restrictions in CS1, and the threshold β of CS2 was set to 0.5.

CS1 has at least 95% of accuracy in MNIST and Fashion EMNIST, which are low-resolution datasets. CS2's performance is not good except for MNIST and Fashion MNIST datasets. The accuracy and $N_{CS2}$of the CS2 method change depending on how threshold β is set. Details will be covered in 4.3. However, it can be seen that performance deteriorates significantly on the high-resolution dataset, CIFAR 10/100. It is seems to that they lost class information during dimension reduction due to limitations in the performance of the CAE model. MNIST, Fashion-MNIST, and EMNIST-Digit with low resolution contained class information even when two-dimensional features were extracted through 13-CNN-based CAE, but not for CIFAR 10 and CIFAR 100. CIFAR 10 /100 needs to be reviewed for ResNet 18 or another dimension reduction and clustering method.

We can check the relationship between various sampling sizes $|S^i|$ and the number of iteration p when N_L is given. Considering that the performance of CS1 and CS2 is currently good in MNIST Dataset, it is seems to satisfy the assumption already. At this time, the threshold β of CS2 was set to 0.5.

The performance of CS1 and CS2 is similar. In most cases, the accuracy reaches 99%. In terms of accuracy, the size of N_l and whether Iteration exist or not does not affect significantly. Both CS1 and CS2 methods can be applied for 10 to 40 times of unlabeled train data for given sampling size N_l. After CS1 is applied, CS2 can be additionally applied to the unlabeled data. In this case, as the N_l is increased, the size of the additional N_{CS2} is reduced. From this, CS1 and CS2 perform classification for unlabeled data under similar conditions, but areas a little bit different. As N_l increases, the difference in classification between the two methods decreases. Efficiency of the CS1 and CS2 methods gradually decreases as N_l increases. This is because classification is impossible even if the sampling size is increased for the boundary or where different class data are overlapped.

N_{CS1} increases when dividing N_l by P iterations. This can be explained through the snapshot of coreset selection process. There is no difference between sampling 100 data and sampling 10 data for 10 iteration respectively in the perspective of coreset selection. Only the number of times CS1 is applied for each moment of coreset selection is different. That is, it is the same as taking a snapshot for p times about the relationship of subgraph during a total of N_l coreset selection. The CS2 method can also be applied when there are a sufficient number of iterations. As the number of iterations increases, more unlabel data would be classified.

Dataset	N_l ($\|S^i\|$, P)	CS1 Only (N_{cs1}, Acc)		CS2 Only (N_{cs2}, Acc)	
MNIST	1000 (1000, 1)	11992(x12)	99.59(%)	X	X
(1 x 28 x 28)	1000 (100, 10)	15305(x15)	99.97(%)	14237(x14)	94.19(%)
FashionMNIST	1000 (1000, 1)	4217(x4)	96.84(%)	X	X
(1 x 28 x 28)	1000 (100, 10)	6492(x6)	95.56(%)	5641(x5)	90.48(%)
EMNIST-Letter	1000 (1000, 1)	4697(x4)	93.52(%)	X	X
(1 x 28 x 28)	1000 (100, 10)	7732(x7)	95.16(%)	0	0(%)
CIFAR10	1000 (1000, 1)	15(x0.01)	66.67(%)	X	X
(3 x 32 x 32)	1000 (100, 10)	46(x0.05)	73.91(%)	2250(x2)	20.56(%)
CIFAR100	1000 (1000, 1)	9(x0.01)	33.33(%)	X	X
(3 x 32 x 32)	1000 (100, 10)	51(x0.05)	76.46(%)	0	0(%)

Table 1. Performance of CS1 and CS2 for each datasets

However, the sampling size for each iteration should not be lowered than a certain amount. Even though it is at the boundary between classes, there would be cases where the classes of encountered subgraphs are all the same by chance. It could deteriorates the accuracy of CS1 and CS2 significantly. To prevent this, increase the sampling size is helpful. The larger the sampling size, the smaller the radius δ. Small δ increases probability both of that only the same class belongs in the subgraph and the number of encountered subgraphs m_i. However, for fixed N_l, it is necessary to appropriately adjust the sampling size $|S_i|$ with the iteration. We will check whether it is possible to prevent this case by limiting the number of subgraphs M encountered or limiting the density α of the subgraphs.

4.3. Sensitivity test for hyperparameter M, α, β

The sensitivity test for the number of encountered subgraphs M in CS1, the subgraph density α, and the threshold β in CS2 is conducted. The accuracy and the number of classifications are calculated according to the value of each hyperparameter when $N_l = 1000$. In order to visualize the difference in performance, the Fashion MNIST dataset, which has lower accuracy than MNIST was used. The resulys is Figure 3-6.

When CS1 was applied with $N_l = 1000$ and $P = 1$ on the Fashion MNIST dataset without additional constraints, N_{CS1} was 5131 and the accuracy reached 97.3%. As M was added as a condition, N_{CS1} decreased, and accuracy are increased. However, when $M = 6$, N_{CS1} reduced by 1,616, but the accuracy increased just 0.4%. The M-limiting

N_l ($\|S^i\|$, P)	CS1 Only (N_{CS1}, Acc)		CS2 Only (N_{CS2}, Acc)		Both ($N_{CS1} + N_{CS2}$, Acc)	
50 (50,1)	738(x14)	99.04(%)	X	X	X	X
100 (100, 1)	3886(x38)	99.84(%)	X	X	X	X
250 (250, 1)	4024(x16)	98.03(%)	X	X	X	X
500 (500, 1)	9354(x18)	99.66(%)	X	X	X	X
750 (750, 1)	10430(x14)	99.75(%)	X	X	X	X
1000 (1000, 1)	11992(x12)	99.59(%)	X	X	X	X
100 (10, 10)	4505(x45)	100(%)	4268(x42)	99.5(%)	4971(+466)	99.02(%)
250 (25, 10)	8874(x35)	99.92(%)	8244(x33)	99.68(%)	9961(+1087)	99.56(%)
500 (50, 10)	11486(x22)	99.85(%)	10672(x21)	98.2(%)	12200(+714)	98.97(%)
750 (75, 10)	13012(x17)	99.84(%)	13810(x18)	99.17(%)	13012(+260)	99.84(%)
1000 (100, 10)	15305(x15)	99.78(%)	14237(x14)	94.19(%)	15305(+252)	99.47(%)

Table 2. Performance of CS1 and CS2 in MNIST

condition does not have a significant effect on accuracy compared to the decrease in N_{CS1}. Subgraph's density α is similar. Accuracy improves with the addition of the α condition, but N_{CS1} decreases exponentially. Even until $\alpha = 0.8$ where N_{CS1} is gradually reduced, N_{CS1} decreases about 800 in increasing accuracy by 1%

Furthermore, the conditions for M and α do not completely prevent bad case of CS1 that occur with low probability. In CS1, when sampling size $|S^i|$ for each session is set small, there is a case where the class of encountered subgraph's center are the same even though it is not the representation of a subgraph by chance. Through Figure 4, it can be confirmed that the accuracy of 1st and 5th period suddenly decreases. When the m_i α condition is added, the effect depends on the situation. The fundamental problem of the above phenomenon is the insufficient performance of the representation learning method. Multiple classes may be mapped to the same low-dimensional area when representation learning fails to cluster. A place where several class data are aggregated may have a high density even at the boundary. Even if the α condition is strengthened as shown in Figure 5, the accuracy may decrease. On the one hand, even if several classes are united, it is difficult to satisfy the conditions of CS1. However, the number of subgraphs encountered is at most eight on two dimensions. That is, there is a possibility that all center of subgraphs u_i which is not representing the subgraph's data sampled by chance are the same. The possibility can be reduced by adding the number M of subgraphs encountered as a condition, but it is not a perfect solution. This can be confirmed through the trend of accuracy change for 1st and 5th period in Figure 5. In addition to improving the performance of representation learning, it is also a method to reduce the size of radius δ by increasing the size of $|S^i|$. As the size of δ decreases, the gap between subgraphs decreases, enabling more detailed class distinction. In addition, the number of subgraphs encountered increases, reducing the possibility of overlapping classes of subgraphs due to chance.

In addition, sensitive tests were conducted on MNIST, Fashion MNIST, and EMNIST to analyze the condition of threshold β in CS2. In all datasets, N_{CS2} and accuracy have a trade-off relationship. In MNIST and Fashion MNIST datasets with the same number of classes, the trend of change in accuracy and N_{CS2} is similar. Each part painted in red is a point that accuracy reaches to 95%. We can consider that β is stricter than the probability of data belongs to each class. However, the appropriate value of β varies for each data set. Considering the results of Table 1, it seems to depend on how well the CAE reflects the information of each dataset.

The appropriate β is influenced by the number of iterations and the number of classes. As the number of iterations P increases, $max(h_{CS2})$ increase as the subgraph information of the periphery are reflected repeatedly. It can be confirmed that the appropriate β decreases significantly as the number of classes increases. We can check that from the case of EMNIST. Classes that have never been included also have large probability. Let's suppose that there is a data belonging to class 1. When the data belongs to class 1 subgraph only once, the probability of class 1 according to the CS2 method is just 23% in 10 class dataset. As the number of classes increases, the probability measured by the CS2 method become decrease further. It is necessary to take measures to generalize the β for various datasets in the future.

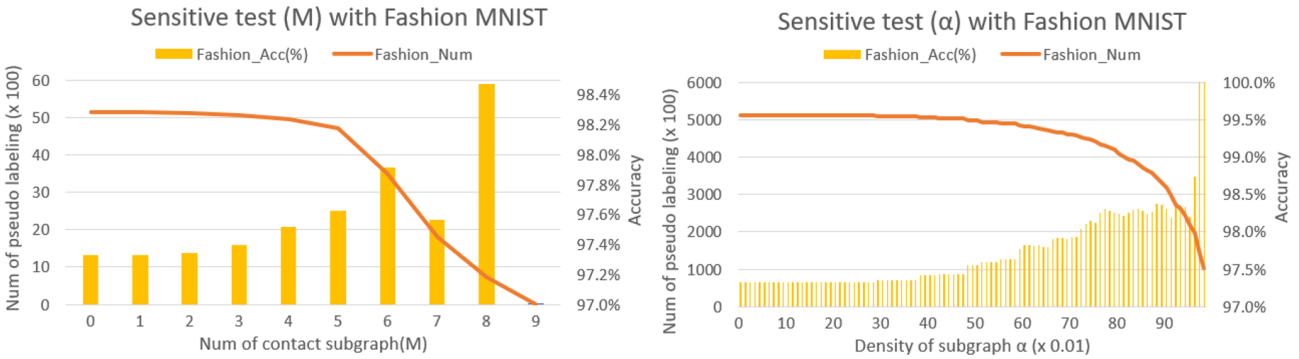

Figure 3. CS1 Sensitive test when $N_l = 1000, P = 1$

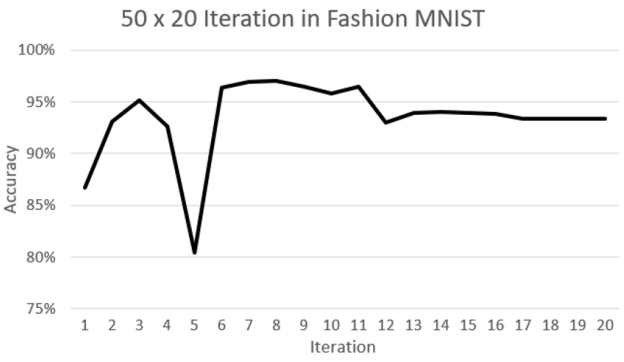

Figure 4. Accurancy of CS1 when $|S^i| = 50, P = 20$

4.4. Limitation

The accuracy for test dataset become poor even if using pseudo labeled data with higher accuracy than 99% via CS1 and CS2. There are three interpretations about this. First, the data classified through the CS1 and CS2 methods are data belonging to the center of the class, and all have similar characteristics. Therefore, training similar data does not have a significant impact on the DL model. Second, the balance of classified unlabel data is broken. The methods of CS1 and CS2 are effective when the feature extracted by CAE is well distinguished from other classes. In the case of MNIST, the classification was good for class of 0, 4, and 6. But in the case of other classes, just a few cases satified condition of CS1 and CS2. Imbalanced label data causes overfitting rather than generalization of the model. In particular, although not included in this paper, we can check that 13-CNN model's performance become same or rather decreased when CS1 and CS2 had 100% accuracy. Third, the data of the misclassified minority class has a great adverse effect on model learning. Due to the lack of performance of CAE, the features were similar, but the original data can differ greatly. As the different class is misclassified as same as major of pseudo labeled class, it can intensify confirmation bias.

5. Future work

The biggest limitation is that the CS1 and CS2 methods can be applied only to datasets with low resolution. It is necessary to find other methods to map high dimension data to a low level with clustering for each class. In this paper, we applied to the 13 CNN model. In the future, it is necessary to apply the unsupervised presentation learning method for high dimensional image data such as ResNet 18 and DGI. Next problem is that the high-accuracy pseudo-labeled data obtained through the CS1 and CS2 methods does not lead to improved performance of the DL model. In the future, it is necessary to solve imbalance classification about unlabeled data and confirmation bias due to the misclassification. The former may be partially resolved by applying a better representation learning method. If the feature extraction is clustered to be well distinguished for each class, CS1 and CS2 can be applied to various class data. In other way, by assuming that the class ratio of the label data selected through the coreset selection is similar to the original dataset, the influence of the unbalanced pseudo labeling data can be regularized. In addition, data augmentation may be performed only for classes that the CS1 and CS2 methods have not been applied. Confirmation bias could be solved by applying outlier detection to a small amount of misclassification data. In the case of training incorrectly labeled data, it was confirmed that the loss value jumped significantly. Therefore, if the value of loss is limited to below a certain threshold, misclassification may be prevented.

In addition, there are several ways for developing CS2. When the number of iterations was increased for a given N_l, the number of classifiable data increased. It could be expanded to iterated for every N_l coreset selection. Above all, since a subgraph is formed for each iteration, each point is included in at least N_l subgraphs. This will allow us to maximize the number of iterations and generalize the threshold β and ensure great performance for various datasets. In addition, if only the CS2 method is considered, there is no need to lower the dimension of the feature to two dimensions. The methods of CS1 and CS2 focused on classification in

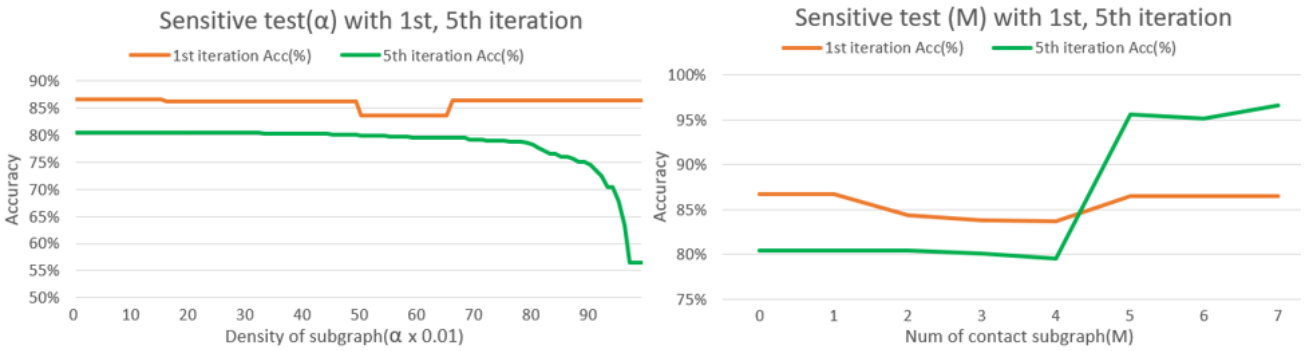

Figure 5. Sensitive test of bad case in CS1

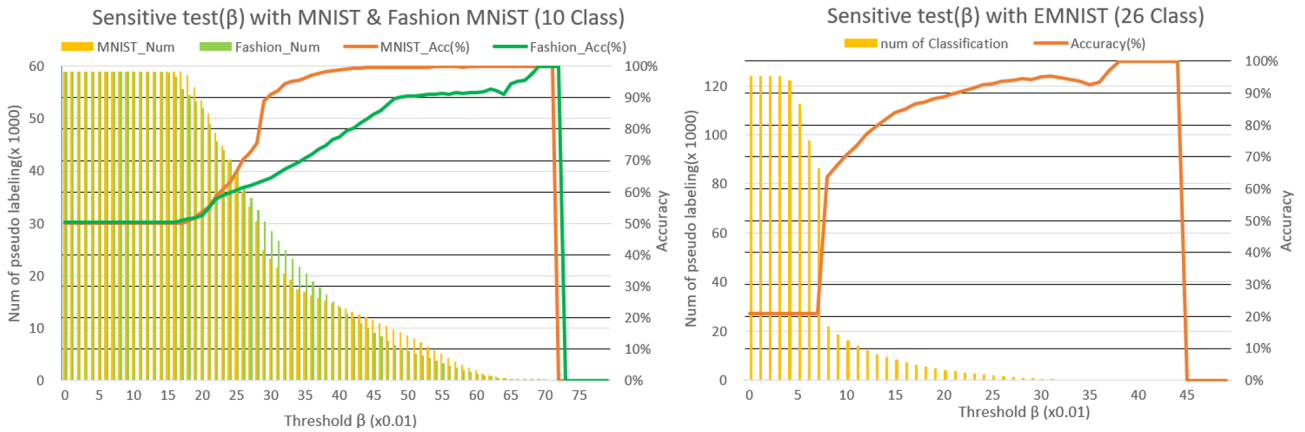

Figure 6. CS2 Sensitive test in MNIST, Fashion MNIST and EMNIST

N_1 ($\|S^i\|$, P)	CS1 N_{CS1}	CS2 + N_{CS2}	Total Acc	CNN Acc (N_1 only)	CNN Acc (+N_{CS1})	CNN Acc (+N_{CS2})
100 (10, 10)	4505	466	99.02	49.84	32.87	31.65
250 (25, 10)	8874	1087	99.56	54.45	39.23	44.52
500 (50, 10)	11486	714	98.97	68.26	52.35	58.03
750 (75, 10)	13012	260	99.84	63.03	54.77	55.42
1000 (100, 10)	15305	252	99.47	67.60	58.17	53.12

Table 3. Model performance when using pseudo labeled data by CS1 and CS2

train dataset. However, if the size of the train dataset is large enough to serve as a population, the representation learning method learned in the train dataset can also be applied in the test dataset. It means that CS1 and CS2 can be applied to test dataset.

6. Conclusion

This study presents a new classification CS1 and CS2 using the subgraph of coreset selection from a geometric perspective. On the small image dataset MNIST, Fashion MNIST, and EMNIST datasets, we can perform partial classification with 95% accuracy for several times of unlabeled data for the given Label data N_l. When representation learning method suitable for datasets is applied, the efficiency become 10 to 40 times higher, and the accuracy is also close to 99%. To ensure the performance of CS1 and CS2, the trade-off between sampling size $|S|$ and the number of iterations P should be considered. The hyperparameter M should be applied to prevent misclassification in the CS1 situation. In two dimensions, $M = 6$ is appropriate. On the other hand, the density α of the subgraph is not very useful. Threshold β is more conservative hyperparameter than the probability that the actual class belongs to. β should be set lower as the number of classes increases and the number of iterations decreases. There are still many things to improve, but it is meaningful in that it is a new classification method to solve the lack of label data.

References

[1] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[2] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

[3] Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6912–6920.

[4] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[5] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896.

[6] Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng MaXiaoyu Tao, and Nanning Zheng. Transductive semi-supervised deep learning using min-max features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 299–315.

[7] Simon Tong. *Active learning: theory and applications*. Stanford University, 2001.

[8] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021.

[9] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pages 1183–1192. PMLR.

[10] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2016.

[11] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

[12] Peng Liu, Hui Zhang, and Kie B Eom. Active deep learning for classification of hyperspectral images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(2):712–724, 2016.

[13] Yibao Sun, Jun Li, Wei Wang, Antonio Plaza, and Zeqiang Chen. Active learning based autoencoder for hyperspectral imagery classification. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 469–472. IEEE.

[14] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981.

[15] Yasi Wang, Hongxun Yao, and Sicheng Zhao. Auto-encoder based dimensionality reduction. *Neurocomputing*, 184:232–242, 2016.

[16] Petr Hurtik, Vojtech Molek, and Irina Perfilieva. Novel dimensionality reduction approach for unsupervised learning on small datasets. *Pattern Recognition*, 103:107291, 2020.

[17] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.