On-Line Training with Guide Data: Shall We Select the Guide Data Randomly or Based on Cluster Centers?

Yuya Kaneda, Qiangfu Zhao, and Yong Liu
Dept. of Computer Science and Engineering
The University of Aizu
Aizu-Wakamatsu, Fukushima, Japan
{d8161108, qf-zhao, yliu}@u-aizu.ac.jp

***Abstract*—To retrain an existing multilayer perceptron (MLP) on-line using newly observed data, it is necessary to incorporate the new information while preserving the performance of the network. This is known as the "plasticity-stability" problem. For this purpose, we proposed an algorithm for on-line training with guide data (OLTA-GD). OLTA-GD is good for implementation in portable/wearable computing devices (P/WCDs) because of its low computational cost, and can make us more independent of the internet. Results obtained so far show that, in most cases, OLTA-GD can improve an MLP steadily. One question in using OLTA-GD is how we can select the guide data more efficiently. In this paper, we investigate two methods for guide data selection. The first one is to select the guide data randomly from a candidate data set G, and the other is to cluster G first, and select the guide data from G based on the cluster centers. Results show that the two methods do not have significant difference in the sense that both of them can preserve the performance of the MLP well. However, if we consider the risk of "instantaneous performance degradation", random selection is not recommended. In other words, cluster center-based selection can provide more reliable results for the user during on-line training.**

***Keywords*-Decision Boundary Making; Multilayer Perceptron; Backpropagation Algorithm; On-Line Learning; Guide Data;**

I. Introduction

Many portable/wearable computing devices (P/WCDs) such as smartphones and smart watches have been developed and used by many users in their daily lives. Various applications are embedded in P/WCDs, so the application market is increasing rapidly in recent years. In this study, we consider to use machine learning models to develop more intelligent applications on the P/WCDs. However, the computational resources of the P/WCD are limited, for example, the CPU power and memory space are less than personal computers, and the battery is limited. High performance machine learning models usually require high calculation cost and large memory space. Therefore, it is difficult to use high performance models in a P/WCD.

To solve the problem, we have proposed the decision boundary making (DBM) algorithm that can be used to design compact and high performance machine learning models [1]. The basic idea of DBM is to reconstruct the decision boundary (DB) of a high performance machine learning model using a small machine learning model. This study uses a support vector machine (SVM) [2] for the high performance model, and a single hidden layer multilayer perceptron (MLP) [3] as the small model. Previous experimental results show that the performance of the MLPs obtained using DBM are better than or comparable to the SVM, and the MLPs are much smaller [1].

Currently, our study focuses on on-line learning to upgrade the performance of a model obtained by the DBM algorithm. During on-line learning, new data are observed in real time, and the user feedbacks can be used as the teacher signals. Updating the model using the newly observed data, we may expect that the model performance can be improved. However, due to limited computing resources and battery, the on-line learning algorithm should require low computing cost to update the model in a P/WCD. Moreover, the learning algorithm should be stable. That is, the performance of the model should not be degraded frequently during on-line learning. Therefore, it is necessary to propose some stable on-line learning algorithms that require low computing cost.

In this study, we have proposed an on-line training algorithm with guide data (OLTA-GD) [4]. OLTA-GD is good for implementation in P/WCDs because of its low computational cost. The basic concern of OLTA-GD is to avoid over-fitting of the model to the new data. When a new datum comes, we calculate the average gradient of the objective function with respect the new datum and the guide data, and update the model using the average gradient. Here, the guide data together are used as a damper to avoid over-fitting. Results obtained so far show that, in most cases, OLTA-GD can improve a pre-trained MLP steadily. One question in using OLTA-GD is how we can select the guide data more efficiently. In this paper, we investigate two methods for guide date selection. The first one is to select the guide data randomly from a candidate dataset G, and the other is to cluster G first using the k-means algorithm [5], and select the guide data from G based on the cluster centers. Results show that the two methods do not have significant difference in the sense that both of them can preserve the performance of the MLP well. However, if we consider the risk of "instantaneous performance degradation", the random selection is not recommended. In other words, the cluster center-based selection can provide more reliable results for the user during on-line training.

The structure of this paper is as follows. Section II briefly introduces the DBM algorithm. Section III explains the proposed OLTA-GD algorithm in detail. Section IV shows the objective, design and environments of experiments in this paper. Section V gives the experimental results and discussions. Section VI is conclusion of this paper.

II. A Brief Review of Decision Boundary Making Algorithm

The DBM algorithm was proposed to design compact and high performance machine learning models in our earlier study [1]. The basic idea of the DBM algorithm is to re-generate the DB of a high performance model. If we can reconstruct the DB using a low-cost model, the low-cost model will also have high performance. For that, we obtain the low-cost model using a new training set generated around the DB of the high performance model. We use a small MLP for the low-cost model, and an SVM model as the high performance model. Fig. 1 shows the brief flow of the learning steps.

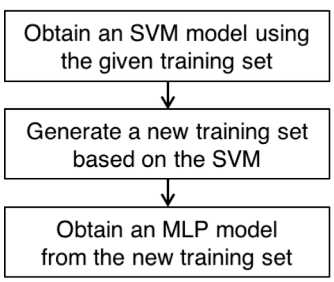

Figure 1. The learning flow of the DBM algorithm. An SVM model is obtained from the given training set, a new training set formed to the DB of the SVM model is generated, and then an MLP model is obtained using the new training set.

The principle of the DBM algorithm is to generate the new training set. For the generation, we define four parameters and two conditions. The parameters are $\epsilon \in \mathbf{R}$, $N \in \mathbf{N}$, $\delta_{DB} \in \mathbf{R}$, and $\delta_{outlier} \in \mathbf{R}$. We generate N data in the ϵ-neighborhood of each support vector (SV). Suppose that a datum X is generated in the neighborhood of a SV P, we add the datum X into the new training set if the datum X satisfies Eq. (1), where $f_{SVM}(X)$ is the output value of the SVM for datum X. The new training set is consisted from the given training set, SVs of the SVM model, and newly generated data. However, sometimes there are noisy data in the given training set. We remove the noisy data before adding to the new training set by Eq. (2), where $y \in \{-1, 1\}$ is the label of a datum X. Note that we consider two-class problems only in this paper.

$$\delta_{DB} \leq |f_{SVM}(X)| \leq |f_{SVM}(P)| \tag{1}$$

$$f_{SVM}(X) \times y < -\delta_{outlier} \tag{2}$$

III. On-Line Training with Guide Data

To update a model steadily in real time on a P/WCD, we proposed an on-line training algorithm with guide data (OLTA-GD). The OLTA-GD updates a model initialized by the DBM algorithm. When an observed datum $X_{observed}$ is coming, we update the model with the observed datum $X_{observed}$ and guide data by Eq. (3), where $w^{(t)}$ is a weight of MLP at time t, α is the learning rate, $X_{guide,i}$ is the i-th guide data, $\mathbf{W}^{(t)}$ is the weight vector of the MLP at time t, $h_{MLP}(X, \mathbf{W})$ is the objective function such as error function in backpropagation (BP) algorithm [9] for a datum X and a weight vector $\mathbf{W}$, and N_{guide} is the number of guide data. Here, the guide data are used together as a damper to avoid over-fitting to $X_{observed}$. To reduce the computational cost and improve the performance of the MLP stably, we need to use a proper value for N_{guide}.

$$w^{(t+1)} = w^{(t)} - \frac{\alpha}{N_{guide}+1}\Big(\sum_{i=1}^{N_{guide}} \frac{\partial h_{MLP}(X_{guide,i}, \mathbf{W}^{(t)})}{\partial w} + \frac{\partial h_{MLP}(X_{observed}, \mathbf{W}^{(t)})}{\partial w}\Big) \tag{3}$$

How do we select the guide data for OLTA-GD? We investigate two guide data generation methods in this paper, namely random selection and cluster center-based selection. For the guide data, we use the new training set generated in the DBM learning for the candidate set G. The new training set has the given training set and many generated data. These guide data generation methods select data from the candidate set G. Moreover, after model updating, we add the observed datum $X_{observed}$ into the candidate set G to update the set through learning. Thus, the number of data increases in G over time. The detailed selection methods and the number of guide data N_{guide} are explained in the sub-sections.

A. Random Selection for the Guide Data

The random selection method picks up guide data randomly from the candidate set G. In this method, the idea is very simple and the data selection cost is very low. We set a value directly to the N_{guide}, therefore the number of guide data N_{guide} is one of given parameters in this method. We use N_{guide} data and an observed datum $X_{observed}$ to update the model. For updating the candidate set G, we simply add the observed datum $X_{observed}$ into the set G after updating the model.

B. Cluster Center-based Selection for the Guide Data

This method uses k-means algorithm to partition the candidate set G. After obtaining an MLP model in the DBM algorithm, we partition the candidate set G into k clusters and add the centers into the corresponding clusters. When an observed datum $X_{observed}$ is received, the cluster center-based selection method picks up a datum from each cluster. However, we do not use the datum which the observed datum belongs. From the condition, we use the observed datum and $k-1$ guide data for model updating.

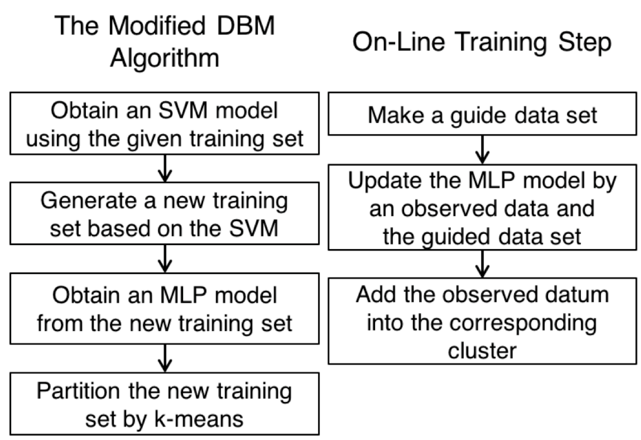

Figure 2. This figure shows the steps of the modified DBM algorithm for the guide data generation method, and the on-line learning steps.

The modified DBM algorithm and the on-line learning steps are shown in Fig. 2, and the guide data selection by the cluster center-based selection is given in below.

Reset the guide data set U_{guide} to empty.
Classify the observed datum $X_{observed}$, and suppose it belongs to the p-th cluster.
for $i = 1$ to k **do**
    if $i = p$ **then**
        Continue
    end if
    Pick up a datum X_{guide} from the i-th cluster of the candidate set G.
    Add X_{guide} to U_{guide}.
end for

We update the MLP model using the guide data set U_{guide} and the observed datum $X_{observed}$. The number of the guide data N_{guide} becomes $k-1$. Therefore, the N_{guide} in this method is decided by the parameter k of the k-means algorithm. When we update the candidate set G, we add the observed datum $X_{observed}$ into the p-th cluster of the set G.

Table I
PARAMETERS OF PUBLIC DATABASES FROM [6].

	Number of Classes (N_c)	Number of Features (N_d)	Number of Data (N_t)
Mushroom	2	22	8,123
Ozone	2	72	2,536
QSAR	2	41	1,055
Seismic	2	18	2,584

Table II
MACHINE SPECS AND ENVIRONMENTS.

Machine	Apple iMac 21.5-inch, Late 2013
OS	Mac OS X 10.9
CPU	Intel Core i5 2.7GHz
Memory	8GB
Program Language	C++
Compiler	Apple LLVM version 6.0

IV. EXPERIMENTAL DESIGN

This paper investigates the accuracy and stability of two guide data selection methods for OLTA-GD by experiments on some public databases. The databases are taken from the machine learning repository of the University of California at Irvine [6]. We use four databases that are mushroom (Mushroom), ozone level detection (Ozone), QSAR biodegradation (QSAR) [7], and seismic bumps (Seismic) [8]. The parameters of the databases are shown in Table I, and the computer configuration used in the experiment is given in Table II.

Our study focuses on on-line training to update an MLP model initialized by the DBM algorithm. In the experiments, we divide the training data set into two sub-sets. The first one, denoted by $U_{off\text{-}line}$, is used for DBM learning. The second one, denoted by $U_{on\text{-}line}$ is used for OLTA-GD or on-line BP learning. Each datum in $U_{on\text{-}line}$ is accessed one-by-one in the on-line training process. A datum is used only once for on-line training. For comparison, we consider the following three methods:

- DBM: MLP model learned by the DBM algorithm, and the BP algorithm for on-line learning.
- DBM-RGD: MLP model learned by the DBM algorithm, and updated by the OLTA-GD with BP algorithm generated by the random selection method (RGD).
- DBM-CGD: MLP model learned by the DBM algorithm, and updated by the OLTA-GD with BP algorithm generated by the cluster center-based selection method (CGD).

To evaluate the performance, we conducted 40 times of 5-fold cross validation [10], and calculated recognition rate (RR) based on the confusion matrix for test set and the accuracy reduction counts (ARC) for several different reduction levels. RR is used to measure the overall performance of the trained MLP, and ARC is used to measure the stability of the on-line training process. For RR, the higher the better,

Table III
RR ALTERATIONS BETWEEN BEFORE AND AFTER ON-LINE LEARNING IN MUSHROOM DATABASE.

	RR (%)			RF(%)
	Before		After(6,400)	
DBM	95.7	→	98.6	+2.9
DBM-RGD (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	95.7	→	98.9	+3.2
DBM-CGD (1, 2, 3, 4, 6)	95.7	→	98.9	+3.2
DBM-CGD (5, 7, 8, 9, 10)	95.7	→	99.0	**+3.3**

Table IV
RR ALTERATIONS BETWEEN BEFORE AND AFTER ON-LINE LEARNING IN OZONE DATABASE.

	RR (%)			RF(%)
	Before		After(1,929)	
DBM	95.8	→	97.0	+1.2
DBM-RGD(1, 2, 3)	95.8	→	**97.1**	**+1.3**
DBM-RGD (4, 5, 6, 7, 8, 10)	95.8	→	97.0	+1.2
DBM-RGD(9)	95.8	→	96.9	+1.1
DBM-CGD(1, 3)	95.8	→	**97.1**	**+1.3**
DBM-CGD (2, 4, 5, 6, 7, 8, 9, 10)	95.8	→	97.0	+1.2

and for ARC, the lower is better. The results were averaged over 40 $\times$ 5 runs.

In the experiments, all data were normalized by scaling. After normalization, each feature takes value in [-1, 1]. For the training set, we used only 100 data for $U_{off\text{-}line}$, and all others were assigned to $U_{on\text{-}line}$.

In the BP-based MLP learning, the learning rate was set to 0.5, the number of off-line learning epoch was 1,000, the number of on-line learning epoch was 1 for each new datum, the number of input neurons was N_d, the number of hidden neurons was 10, and the number of output neurons was 1.

For SVM, we used soft-margin SVM, and sequential minimal optimization [11] algorithm for training the SVM model. We used a radial basis function ($\kappa(x_1, x_2) = \exp(-||x_1 - x_2||^2)$) for the kernel function, and the parameter C was set to 1. In DBM learning, we set the parameters as $N = 10$, $\epsilon = 0.1$, $\delta_{DB} = 0.1$, and $\delta_{outlier} = 0.2$. These parameters were used in previous study.

For guide data selection, we changed the number N_{guide} from 1 to 10 with a step size 1. For the cluster center-based method, we defined k as $N_{guide} + 1$. We would like to confirm the effect of N_{guide} on the performance of the MLP.

V. EXPERIMENTAL RESULTS AND DISCUSSION

Tables III-VI show averaged RR results of each method before and after on-line learning for all databases. The values in parentheses of after are the times of on-line learning in the database, and the bold values mean the best RR among the methods. In the tables, we also indicate rising or falling (RF) value for the RR results. If the RF value is positive, the RR after on-line learning is increased, but the RR is

Table V
RR ALTERATIONS BETWEEN BEFORE AND AFTER ON-LINE LEARNING IN QSAR DATABASE.

	RR (%)			RF(%)
	Before		After(744)	
DBM	81.0	→	78.2	-2.8
DBM-RGD(1)	81.0	→	80.8	-0.2
DBM-RGD(2, 4)	81.0	→	81.5	+0.5
DBM-RGD (3, 6, 8, 9, 10)	81.0	→	81.7	+0.7
DBM-RGD(5)	81.0	→	**81.8**	**+0.8**
DBM-RGD(7)	81.0	→	81.6	+0.6
DBM-CGD(1)	81.0	→	80.2	-0.8
DBM-CGD(2)	81.0	→	81.3	+0.3
DBM-CGD(3)	81.0	→	81.2	+0.2
DBM-CGD (4, 5, 7, 9, 10)	81.0	→	81.5	+0.5
DBM-CGD(6, 8)	81.0	→	81.6	+0.6

Table VI
RR ALTERATIONS BETWEEN BEFORE AND AFTER ON-LINE LEARNING IN SEISMIC DATABASE.

	RR (%)			RF(%)
	Before		After(1,968)	
DBM	91.0	→	**93.2**	**+2.2**
DBM-RGD(1)	91.0	→	93.0	+2.0
DBM-RGD(2, 3)	91.0	→	92.9	+1.9
DBM-RGD(4, 5)	91.0	→	92.8	+1.8
DBM-RGD(6, 7, 8, 9, 10)	91.0	→	92.7	+1.7
DBM-CGD(1)	91.0	→	93.1	+2.1
DBM-CGD(2)	91.0	→	92.9	+1.9
DBM-CGD(3, 4, 5)	91.0	→	92.8	+1.8
DBM-CGD(6, 7, 9)	91.0	→	92.7	+1.7
DBM-CGD(8)	91.0	→	92.6	+1.6
DBM-CGD(10)	91.0	→	92.5	+1.5

decreased if the value is negative. For DBM-RGD and DBM-CGD, N_{guide} is denoted in parentheses. Figs. 3-6 are RR transition graph of averaged results through on-line learning for each database. In the graphs, we only show DBM-RGD and DBM-CGD with $N_{guide} = 8$ setting. For other settings, the trends of transitions are similar to $N_{guide} = 8$ setting. And Figs. 7-10 reveal averaged results of ARCs in 1% and

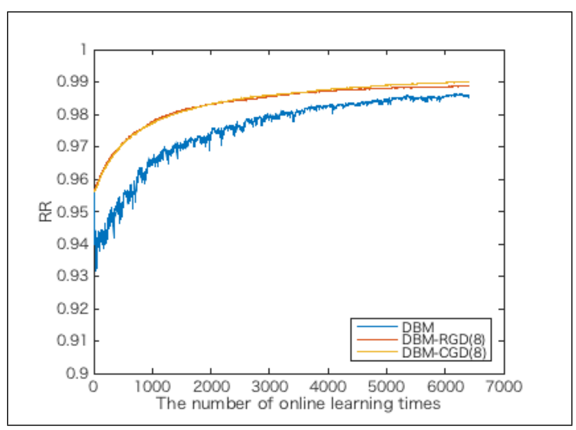

Figure 3. Accuracy transition graph for each method with $N_{guide} = 8$ setting in Mushroom database.

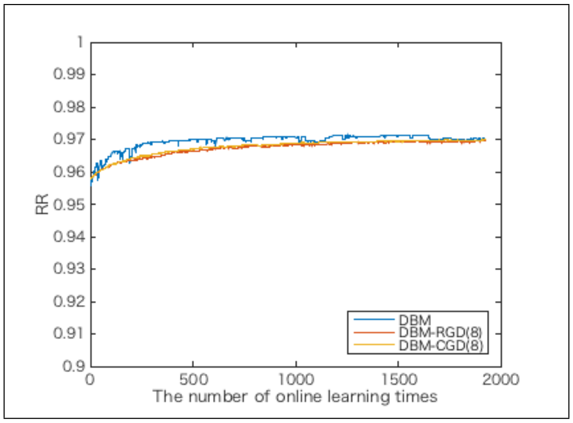

Figure 4. Accuracy transition graph for each method with $N_{guide} = 8$ setting in Ozone database.

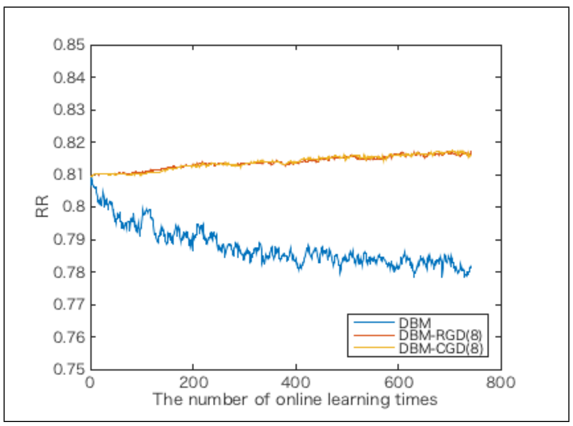

Figure 5. Accuracy transition graph for each method with $N_{guide} = 8$ setting in QSAR database.

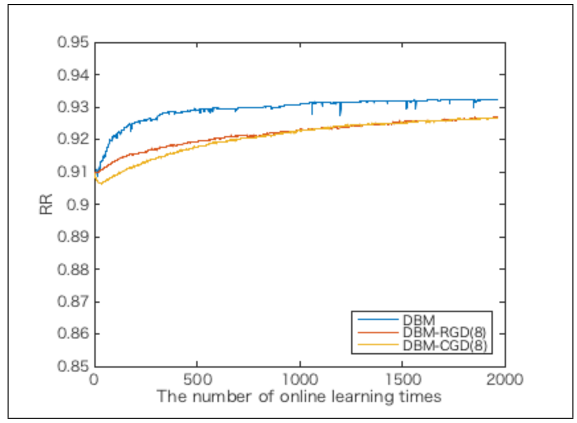

Figure 6. Accuracy transition graph for each method with $N_{guide} = 8$ setting in Seismic database.

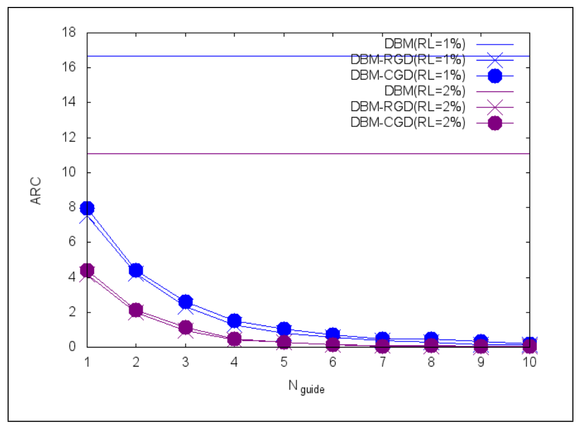

Figure 7. Line graphs for averaged accuracy reduced counts by N_{gd} values in Mushroom database.

2% reduced levels (RLs) by changing N_{guide} values for every database. In the figures, the less averaged ARCs mean better results than larger ARC values.

In comparison among the methods, DBM-RGD or DBM-CGD with better N_{guide} setting reveals better or equivalent performance of DBM in all databases from Tables III-VI and Figs. 3-6. Only RR results, DBM performance is almost the same as DBM-RGD/DBM-CGD in most cases. However, in QSAR database, DBM decreases the RR, but DBM-RGD and DBM-CGD can upgrade the RR through on-line training. From results of stability, DBM-RGD or DBM-CGD with better N_{guide} setting shows better than DBM in all cases. If we use small N_{guide} value, such as 1 or 2, DBM-RGD and DBM-CGD sometimes become unstable results than DBM. However, if we use larger values than a certain value (e.g. 8, 9, and 10), DBM-RGD or DBM-CGD is the most stable results. Therefore, on-line learning using guide data is a good training algorithm for our purpose.

As for better guide data selection, the random selection and the cluster center-based selection show almost same results in many cases, but the random selection becomes worse stability than the cluster center-based selection in some cases. In most cases, the accuracy and stability of the two selection methods are same. However, in Ozone database, the ARCs of DBM-RGD for 1% become the worst among the methods. For the 2% counts, DBM-RGD with better N_{guide} setting becomes better than DBM. On the other hand, DBM-CGD with better N_{guide} setting shows the best results. Thus, in this case, we should use the cluster center-based selection method for guide data selection. For other case, we should use the random selection method because the calculation cost of the random selection is less than the cluster center-based selection. For real applications, the hybrid selection of the two methods may be better method if we know the condition of this case. When data status is identified as the special case, we will use the cluster center-based selection method for the guide data selection, and we use the random selection in other case. The hybrid system may become more applicable method for our purpose. We will investigate the condition of the case in the future.

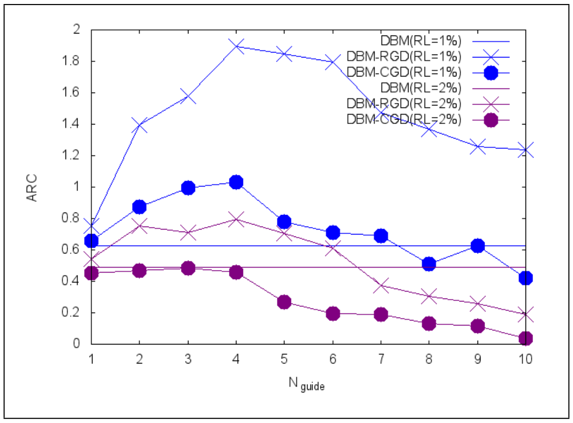

Figure 8. Line graphs for averaged accuracy reduced counts by N_{gd} values in Ozone database.

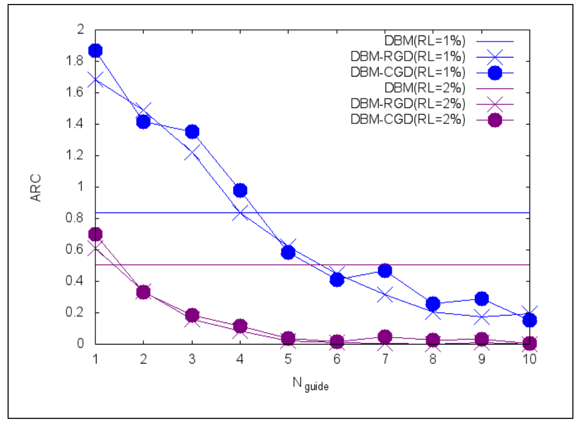

Figure 10. Line graphs for averaged accuracy reduced counts by N_{gd} values in Seismic database.

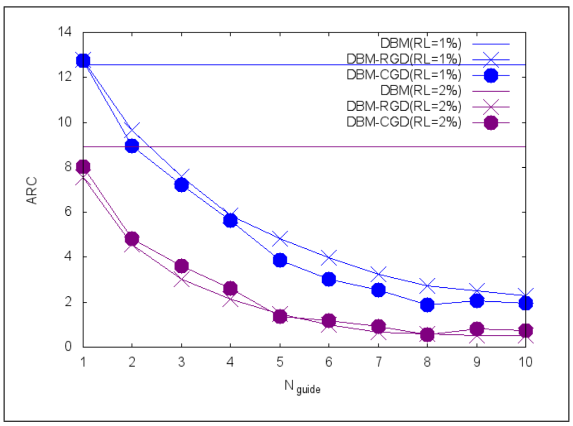

Figure 9. Line graphs for averaged accuracy reduced counts by N_{gd} values in QSAR database.

As for better N_{guide} setting, around 8 is better setting than others to update the model on P/WCD environments. In RR results, the RR increases until a certain value in some cases, and it decreases the RR with larger N_{guide} setting in some cases. Moreover, the calculation cost of the OLTA-GD is proportional to N_{guide}. From the two reasons, we should use less value for N_{guide}. However, from Figs. 7-10, the counts are almost converged around $N_{guide} = 8$ to 10 in many cases. The most important point of this study is stability. Therefore, from the three reasons, we conclude that $N_{guide} = 8$ is better setting in this study.

VI. Conclusion

In this paper, we have investigated two guide data selection methods for OLTA-GD to update a model in real time. One is random selection, and the other is cluster center-based selection. The OLTA-GD updates a model initialized by the DBM algorithm using average gradients of observed data and the guide data set. For the guide data selection methods, the methods pick up guide data from candidate set G which is obtained in the DBM algorithm and added each observed datum through on-line training. The random selection method picks up guide data from the candidate set G randomly, and the cluster center-based selection method gets guide data set from each cluster of the set G partitioned by the k-means algorithm. From the experimental results, the two methods have almost the same performance, but in some cases, the cluster center-based selection method is more stable than the random selection. To upgrade model stably in all cases, we would like to investigate the condition of the special case in the future.

For other future works, we will consider a method to add more effective observed data into the candidate set G. In current method, we add the observed data without conditions. Therefore, the size of G may become huge over on-line learning times, then it will require large storage on P/WCDs. To reduce the storage usage, we would like to propose a method to remove ineffective data from the set G in the future.

Acknowledgement

This work was supported by Grant-in-Aid for JSPS Fellows Grant Number 15J10477.

References

[1] Y. Kaneda, Y. Pei, Q. Zhao, and Y. Liu, "Improving the performance of the decision boundary making algorithm via outlier detection," *Journal of Information Processing*, vol. 23, no. 4, pp. 497–504, 2015.

[2] V. N. Vapnik, *Statistical learning theory*, 1st ed. Wiley, Sep. 1998.

[3] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, Jul. 1989.

[4] Q. Z. Yuya Kaneda, “A study on an on-line learning for multilayer perceptorn using guide data,” in *The 30th Annual Conference of the Japanese Society for Artificial Intelligence 2016*, Jun 2016, pp. 3E4–4.

[5] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press, 1967, pp. 281–297.

[6] K. Bache and M. Lichman”, “UCI machine learning repository,” 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[7] K. Mansouri, T. Ringsted, D. Ballabio, R. Todeschini, and V. Consonni, “Quantitative structure - activity relationship models for ready biodegradability of chemicals,” *Journal of Chemical Information and Modeling*, vol. 53, pp. 867–878, 2013.

[8] S. M. and W. L., “Application of rule induction algorithms for analysis of data collected by seismic hazard monitoring systems in coal mines,” *Archives of Mining Sciences*, vol. 55, no. 1, pp. 91–114, 2010.

[9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 10 1986.

[10] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI’95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 1137–1143.

[11] J. Platt, “Sequential minimal optimization: A fast algorithm for training support vector machines,” Microsoft Research, Tech. Rep. MSR-TR-98-14, April 1998.