Incremental/Decremental Learning for LS-SVM

Wei-Han Lee[1], Bong Jun Ko[1], Shiqiang Wang[1], Changchang Liu[1], Kin K. Leung[2]
[1]IBM T. J. Watson Research Center, Yorktown Heights, NY, USA
[2]Imperial College London, UK

***Abstract*—In this paper, we present a novel incremental and decremental learning method for the least-squares support vector machine (LS-SVM). The goal is to adapt a pre-trained model to changes in the training dataset, without retraining the model on all the data, where the changes can include addition and deletion of data samples. We propose a *provably exact* method where the updated model is exactly the same as a model trained from scratch using the entire (updated) training dataset. Our proposed method only requires access to the updated data samples, the previous model parameters, and a unique, fixed-size matrix that quantifies the effect of the previous training dataset. Our approach can significantly reduce the storage requirement of model updating, preserve the privacy of unchanged training samples without loss of model accuracy, and enhance the computational efficiency. Experiments on real-world image dataset validate the effectiveness of our proposed method.**

I. Introduction

Machine learning (ML) models in image/vision applications usually need to be trained with a large amount of data. In many practical scenarios of tactical coalitions, the samples in the training dataset can change over time, due to the addition of new data samples (e.g., those that have been collected and labeled recently) and removal of existing data samples (e.g., those that are too noisy or incorrectly labeled). Retraining the model from scratch every time when there is a change in the training dataset is too time-consuming. It is more efficient to update the model by including or excluding the influence of specific data samples, which is known as *incremental and decremental learning*.

Least-squares support vector machine (LS-SVM) has been broadly applied in various ML tasks and image/vision applications [2]–[4]. The benefit of LS-SVM is that there exists an analytical solution to the optimal model parameters for a given training dataset. However, it is still computationally intensive to recompute the model parameters on the entire dataset when only a few samples in the dataset change.

In this paper, we propose a novel approach that uses analytical expressions to *exactly update* the LS-SVM model using the added and deleted data samples only. Our approach utilizes an auxiliary matrix that is defined to capture some essential information in the previous training dataset, which has a much smaller size than the original dataset and does *not* grow with the size of the dataset. The updated model is *provably the same as* a model trained on the entire dataset. Hence, the proposed model updating process is much faster and requires less storage than retraining the model from scratch while retaining the same model accuracy. In addition, because our model updating algorithm does not require knowledge of those raw data samples that remain unchanged (i.e., not added or deleted), it preserves the privacy of unchanged data samples in a distributed system setting.

An extended version of this work has been accepted at IEEE ICIP 2019 [1].
This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

II. Least Squares Support Vector Machine

The goal of LS-SVM is to learn a model that assigns the correct label to a data sample with respect to minimizing the squared error in the assigned labels. Formally, it learns a function $f : \mathbb{R}^M \to \mathbb{R}$ which maps each data sample $\boldsymbol{x}$ to a label y with the optimal classifier [5]:

$$\boldsymbol{w}^* = \text{argmin}_{\boldsymbol{w}\in\mathbb{R}^J} \rho\|\boldsymbol{w}\|^2 + \sum\nolimits_{n=1}^{N} (\boldsymbol{w}^T \vec{\boldsymbol{\phi}}(\boldsymbol{x}_n) - y_n)^2 \quad (1)$$

where N is the number of data samples, $\boldsymbol{x}_n \in \mathbb{R}^M$ represents the n-th data sample, M is the dimension of each data sample. The vector $\boldsymbol{w} \in \mathbb{R}^M$ represents a possible model and $\boldsymbol{w}^*$ is the optimal model that satisfies (1). For a training dataset containing N samples, we define an $M \times N$ matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]$ and a vector $\boldsymbol{y} = [y_1, y_2, \cdots, y_N]$. The function $\vec{\boldsymbol{\phi}}(\boldsymbol{x_i})$ is a kernel function that maps the original data $\boldsymbol{x}_i$ into a (possibly) higher-dimensional space (with J dimensions) for linearly separating the data samples [5]. We also define $\boldsymbol{\Phi} = \boldsymbol{\Phi}(\boldsymbol{X}) = [\vec{\boldsymbol{\phi}}(\boldsymbol{x}_1), \vec{\boldsymbol{\phi}}(\boldsymbol{x}_2), \cdots, \vec{\boldsymbol{\phi}}(\boldsymbol{x}_N)]$. The objective function in (1) has an optimal solution given by [5]:

$$\boldsymbol{w}^* = \boldsymbol{\Phi}[\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \rho\boldsymbol{I}_N]^{-1}\boldsymbol{y} = [\rho\boldsymbol{I}_J + \boldsymbol{\Phi}\boldsymbol{\Phi}^T]^{-1}\boldsymbol{\Phi}\boldsymbol{y} \quad (2)$$

where $\boldsymbol{I}_k$ denotes the identity matrix of size k (for given k).

III. Proposed Model Updating Approach

We present our proposed exact incremental and decremental learning approach as follows. We first consider the general case where we need to update the LS-SVM model by adding N_a data samples $(\boldsymbol{X}_a, \boldsymbol{y}_a)$ to the training dataset and removing N_r data samples $(\boldsymbol{X}_r, \boldsymbol{y}_r)$ from the training dataset at the same time. Our main result is in Theorem 1, according to which we can update the model exactly without the need of accessing the unchanged data samples.

Theorem 1. *For a given model*

$$\boldsymbol{w} = \boldsymbol{\Phi}[\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \rho\boldsymbol{I}_N]^{-1}\boldsymbol{y} \quad (3)$$

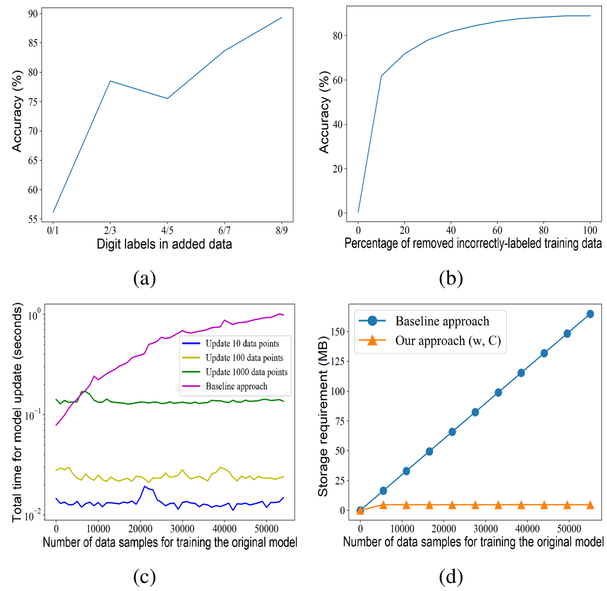

Fig. 1. Experimentation results: (a) model accuracy when incrementally adding each group of data; (b) model accuracy after removing different percentages of incorrectly-labeled training data; (c) total time for model update/retraining; (d) storage requirement.

and auxiliary matrix

$$\boldsymbol{C} = \boldsymbol{\Phi}[\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \rho \boldsymbol{I}_N]^{-1}\boldsymbol{\Phi}^T, \tag{4}$$

when adding new training data $(\boldsymbol{X}_a, \boldsymbol{y}_a)$ *and removing existing training data* $(\boldsymbol{X}_r, \boldsymbol{y}_r)$*, we can compute the new values of* $\boldsymbol{w}$ *and* $\boldsymbol{C}$ *using*

$$\boldsymbol{w}_{\text{new}} = \boldsymbol{w} + (\boldsymbol{C} - \boldsymbol{I}_J)\boldsymbol{\Phi}_c \left(\rho \boldsymbol{I} - \boldsymbol{\Phi}_c'^T(\boldsymbol{C} - \boldsymbol{I}_J)\boldsymbol{\Phi}_c\right)^{-1} \left(\boldsymbol{\Phi}_c'^T \boldsymbol{w} - \boldsymbol{y}_c\right)$$

$$\boldsymbol{C}_{\text{new}} = \boldsymbol{C} + (\boldsymbol{C} - \boldsymbol{I}_J)\boldsymbol{\Phi}_c \left(\rho \boldsymbol{I} - \boldsymbol{\Phi}_c'^T(\boldsymbol{C} - \boldsymbol{I}_J)\boldsymbol{\Phi}_c\right)^{-1} \boldsymbol{\Phi}_c'^T \left(\boldsymbol{C} - \boldsymbol{I}_J\right)$$

where we define $\boldsymbol{\Phi}_c = (\boldsymbol{\Phi}_a, \boldsymbol{\Phi}_r)$*,* $\boldsymbol{\Phi}_c' = (\boldsymbol{\Phi}_a, -\boldsymbol{\Phi}_r)$*, and* $\boldsymbol{y}_c = (\boldsymbol{y}_a, -\boldsymbol{y}_r)$*.*

IV. Experimentation Results

The performance of our proposed model updating method is evaluated using experiments on MNIST dataset [6]. We train an LS-SVM to classify even vs. odd digits.

We first consider the incremental learning process where we separate the training data into five groups according to their labels (digits in the MNIST dataset), i.e., 0&1, 2&3, 4&5, 6&7 and 8&9. The initial model is trained with digits 0 and 1. We sequentially add each group of data to update the model using our proposed approach, to evaluate the scenario where data in different subcategories are added over time. The model accuracy is evaluated on the entire MNIST test dataset which includes all the digits. From Fig. 1(a), we observe that the model accuracy generally increases after adding a new group of data, which shows that our method does not forget the previously learned knowledge. The minor decrease of accuracy after adding digits 4 and 5 is due to different similarities between unlearned digits and learned digits.

To investigate the effectiveness of our proposed decremental learning method, we assign 50% of the original training data with randomly assigned erroneous labels based on which we train an initial model. Then, we update the model by gradually removing the incorrectly-labeled training data. From Figure 1(b), we observe that the accuracy increases as more incorrectly-labeled training data samples are removed.

We then compare the time needed for incremental learning (for one batch with a given size) with the time needed in the baseline approach that retrains the model from scratch. All the time measurements are collected on a MacBook Pro with 2.6 GHz Intel Core i7, 32 GB memory. We use a given amount (up to $55,000$) of training samples to train the original model. Then, we incrementally update the model with a given batch size of new data chosen from the remaining $5,000$ training samples in the MNIST dataset. Note that the computational complexity of incremental and decremental model update is the same, so we only focus on incremental update here. For the baseline approach, we consider a batch size of 10 new data samples. The results are shown in Fig. 1(c), where we see that our proposed approach significantly outperforms the baseline in most cases (note the logarithmic scale in the y-axis), confirming the benefit of our approach.

The baseline approach needs to store all the data used to train the original model, in order to retrain from scratch. This is in contrast to our approach which only needs to store the vector $\boldsymbol{w}$ and matrix $\boldsymbol{C}$, which significantly reduces the storage requirement, as shown in Fig. 1(d).

V. Conclusion

We have proposed a model updating process for LS-SVM. By leveraging a unique auxiliary matrix, the proposed approach can update the model incrementally or decrementally without direct access to the previous training data. Our proposed method can achieve exact model updating while protecting the privacy of non-updated training data. The effectiveness of our approach has been confirmed through experiments.

References

[1] W.-H. Lee, B. J. Ko, S. Wang, C. Liu, and K. K. Leung, "Exact incremental and decremental learning for ls-svm," in *26th IEEE International Conference on Image Processing (ICIP)*, Sept. 2019.

[2] Johan AK Suykens, Tony Van Gestel, and Jos De Brabanter, *Least Squares Support Vector Machines*, World Scientific, 2002.

[3] MN Vimalkumar, K Helenprabha, and A Surendar, "Classification of mammographic image abnormalities based on EMO and LS-SVM techniques," *Research Journal of Biotechnology*, vol. 12, no. 1, pp. 35–40, 2017.

[4] Yuefang Gao, Xin Shan, Zexi Hu, Dong Wang, Ya Li, and Xuhong Tian, "Extended compressed tracking via random projection based on MSERs and online LS-SVM learning," *Pattern Recognition*, vol. 59, pp. 245–254, 2016.

[5] Sun Yuan Kung, *Kernel Methods and Machine Learning*, Cambridge University Press, 2014.

[6] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.