

# An Incremental Learning Classification Algorithm based on Forgetting Factor for eHealth Networks

Li Yang<sup>1</sup>, Kun Wang<sup>1</sup>, Chenhan Xu<sup>1</sup>, Chunsheng Zhu<sup>2</sup>, Yanfei Sun<sup>1</sup>

<sup>1</sup> Key Lab of Broadband Wireless Communication and Sensor Network Technology, Ministry of Education, Nanjing University of Posts and Telecommunications, China.

Emails: islyang@foxmail.com, kwang@njupt.edu.cn, xchank@outlook.com, njsyf@vip.163.com

<sup>2</sup> Department of Electrical and Computer Engineering, The University of British Columbia, Canada.

Email: cszhu@ece.ubc.ca

**Abstract**—The advances of network technology and mobile communication technology are making eHealth possible. In eHealth systems, physiological data and relevant context-aware data are acquired continuously and in real time. At the same time, such large-scale data results in huge challenges in the aspect of real-time big data processing since eHealth data appears in the form of data stream. Therefore, we propose a novel incremental learning algorithm, namely  $\alpha$ -SVMSGD, which improves the SVMSGD (Support Vector Machine-Stochastic Gradient Descent) algorithm by updating the training data with the continuous data stream. Besides, this  $\alpha$ -SVMSGD may handle the problem that original SVMSGD cannot further mine the useful information in unclassified data. In  $\alpha$ -SVMSGD, the process of training data updating is completed by introducing the concept of forgetting mechanism, in which the forgetting factor  $\alpha$  is introduced to weed out useless training data.  $\alpha$ -SVMSGD is applied into ambient assisted living communications, and further incorporated into the data filtering layer of a local data processing architecture (LDPA) to reduce data redundancy. Simulation results confirm that the proposed algorithm is a promising data redundancy solution for classification without loss of accuracy in the case of real-time data stream.

**Keywords**—eHealth, Incremental learning, Support vector machine, Stochastic gradient descent, Forgetting factor

## I. INTRODUCTION

Due to the hospital capacity and medical staff are limited concerning the increasing treatment requests, traditional health care services can hardly satisfy growing population's needs. Under this background, for the benefits of big data technique, a new kind of eHealth service monitoring people's lives with intelligent device is developing rapidly. In this current era of big data [1], the Internet transmits a great deal of data, followed with the data storage and data processing by servers or clouds. Besides, mobile networks collect lots of data all around people's lives. Due to the limitations of traditional data processing methods, they are often used to describe those complex or large data sets in the network.

Specifically, the growing amount of data collected by mobile eHealth networks is more and more pervasive with the development of hardware. Meanwhile, collecting nodes are required to get more data [2] and the number of them tends to increase in networks. All the above factors increase the scale of network and the volume of data transferred in the e-Health network.

Since the energy and functionality of mobile nodes are limited, data has to be aggregated and processed in a central sever. However, with the increment of network size which leads to the emergence of more powerful functions, it is likely to cause that the central server is not capable of analyzing all the data due to various factors (e.g., routing blocking resulting from malicious nodes or network congestions). For this reason, how to efficiently process these data is a very important problem. In our previous work, we proposed the framework of a local data processing architecture (LDPA) to provide the idea of quantifying the result of data analysis in ambient assisted living communications (AAL) [3]. However, data redundancy issue still exists in data filtering layer (DFL) of LDPA. To deal with this problem, this paper utilizes an improved SVMSGD (Support Vector Machine-Stochastic Gradient Descent) [4] algorithm by introducing the concept of forgetting factor  $\alpha$  to update training data. Besides, in original SVMSGD, some past information in classifier may be new classifications, and SVMSGD cannot further process this kind of data. We then propose an incremental learning algorithm  $\alpha$ -SVMSGD and further incorporate it into data filtering layer (DFL) of LDPA for mining useful information from unclassified data.

To this end, we mainly focus on efficient data processing and mining useful data classification from the original data and increasing data to save storage space and reduce the probability of data loss due to network congestion. The contributions of our work are summarized as follows:

- A forgetting factor method is proposed to mine useful information from the unclassified data. It can mine new useful information from these useless or incremental data, and reduce history sample storage and the scale of training sample set.
- In  $\alpha$ -SVMSGD, we adopt adaptive method to adjust the value of  $\alpha$ . We set a threshold for  $\alpha$  firstly, and then calculate the error between initial threshold and  $\alpha$  value of sample which has been training for several times. Finally, we choose  $\alpha$  with maximum error weights as a new threshold and begin the next round of the training of new data sample.

The rest of paper is organized as follows. Section II reviews related work. Section III presents the overview of LDPA.

Section IV describes the  $\alpha$ -SVMSGD algorithm in detail. Section V shows the performance evaluation and Section VI concludes the paper.

## II. RELATED WORK

Many researchers focus on algorithms and strategies for big data mining. In the work [5] authors proposed a method of a Keyword-Aware Service Recommendation to provide personal service recommendation. This method uses keywords to identify users' interests and introduces a user-based Collaborative Filtering algorithm for appropriate recommendations. In the work [6] authors presented an incremental learning method based on LS-SVM. This method uses self-adaptive to pure the sample. It prunes threshold and sample increment according to the performance of classifier is good or poor. In [7], the authors proposed a RVNS queue architecture(IRQA) which contains three layers, aiming to improve the reliability of data transmission and the speed of data processing. Authors of [8] proposed a model-matching algorithm based on improved BP which allows users to extract information according to the order of data's arriving time. Besides, the algorithm can improve the matching speed and accuracy. A dynamic assignment scheduling algorithm was proposed in [9] which is used to improve performance of the parallel machines. It also built a stream query graph to calculate the weight of every edge.

The other aspect concentrates on methodologies for a mass of data storage. In the work [10], authors raised a method called privacy-aware cross-cloud service composition. This method proposes evaluations of services through QoS history records to increase the credibility. Also, it adopts a k-means algorithm as the filter for typical historical records. In the work [11], authors presented a model about efficient storage based on code erasing. In this model, a huge mass of data is divided into all kinds of storage nodes. In order to improve the robustness of the whole storage system, different coding parameters can be set up by different users. Liang et al. presented a digital hormone based classification algorithm [12]. For the classifier can be updated without a big sample-buffer, which makes it can save space when stores temporary records. Besides, it predicts the accuracy of class label. In [13], authors presented an architecture of home M2M networks according to the service ranges and the type of applications. A confidential and lightweight data discovery and dissemination protocol was proposed in [14] which can ensure authenticity of broadcast data item. In the work [15], authors developed LI-BLINEAR as an easy-to-use tool to process large sparse data. In order to solve the optimization problems of support vector machines, the authors proposed an effective stochastic sub-gradient descent algorithm in [4]. In [16], authors proposed a finite sample analysis method to prove that ASGD takes a huge number of samples when it chooses learning rate. These three algorithms are closely related to our proposal. Therefore, we will make comparisons in Section V.

All in all, even many researches on above two directions have been implemented, there are very few classification algorithms for processing and storage of raw data and lost data. To solve these problems, we propose a novel incremental learning

algorithm based on forgetting factor, namely  $\alpha$ -SVMSGD. According to trained classifier model with SVMSGD to train incremental data. We use forgetting factor to decide whether to delete or retain data. Thus it can reduce history sample storage and the scale of the training sample set. In this way, we realize the incremental learning, and constantly improve the adaptability and accuracy of the model.

## III. OVERVIEW OF LDPA

LDPA is a three-layer architecture as shown in Fig. 2. The first layer is Data Gathering Layer (DGL) which is responsible for collecting and storing relevant collected data. The second layer is Data Filtering Layer (DFL). Data entering DFL are divided into static data and real-time data after filter. Static data reflects sensors' status, while a real-time data stream reflects quality of life. For limited buffer space and increasing complexity of risk function, a classifier is designed in this layer. The third layer is Data Analyzing layer (DAL). In this layer, data will be reorganized into a neighborhood structure.

In DGL, there is a distributed sensor architecture deployed in elderly people's living environment. Data collected by sensors will gather and be processed in local. Servers receive the data that is realigned according to timestamps from sensors. They adopt a buffer cleaning mechanism because the buffer spaces are limited.

In DFL, there is a filter designed and a relevant data set introduced to give reference. Packets are decapsulated at the moment they enter DFL. At the same time, filter checks representative data in terms of retrieval range from relevant data set. This mechanism can ensure the data completeness and accuracy. There is also a level division method in this layer. All the data in solution space  $\zeta$  of RVNS will be classified into a corresponding level, and we call that RVNS queue construction. DAL is a layer of reorganizing data into neighborhood structure. The idea of risk function is introduced firstly. Reduced Variable Neighborhood Search (RVNS) is used to reconstruct neighborhood structure and select a local optimal solution. For detailed design of RVNS queue, two factors, the size of solution space and the interval of neighborhood reconstruction, can influence its effectiveness. At last, we need to take care of the terminal condition of RVNS. Different conditions have different processing method. If the former is considered as terminal condition, problems still exist. In this case, LDPA needs to send a report of which results must be at its peak. Also, the traditional terminal condition needs to be changed. If the latter is considered as terminal condition, at the moment we achieve the maximum time, RVNS stops.

What we focus in this paper is to solve data redundancy of classifier part in DFL. Therefore, a proposed  $\alpha$ -SVMSGD algorithm will be described in detail accordingly.

## IV. $\alpha$ -SVMSGD ALGORITHM

### A. Algorithm design

In this section, we mainly introduce the main structure of our proposed algorithm. This structure, as shown in Fig. 2, contains two parts, training module and classification module. Training module is responsible for the training and updating

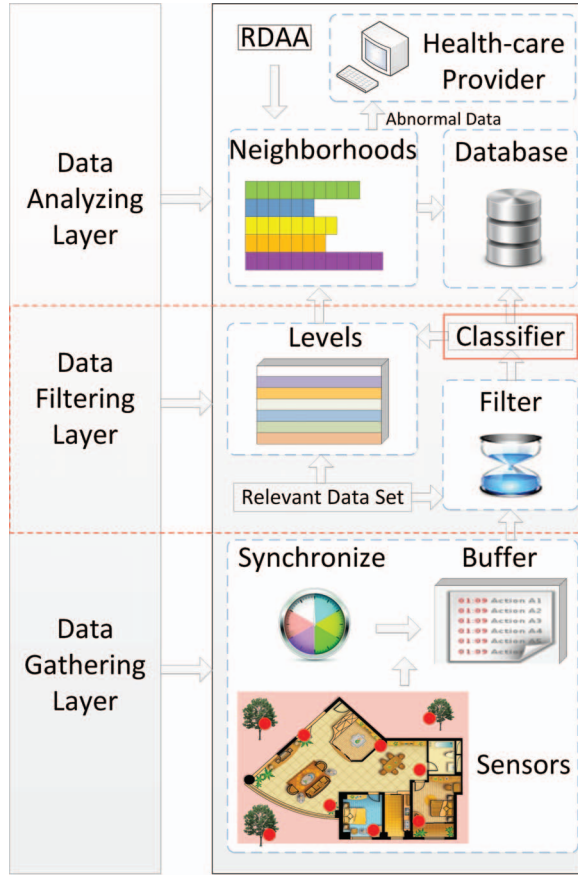


Fig. 1: Architecture of LDPA

the classifier model while classification module is responsible for showing the classification and providing the test data set.

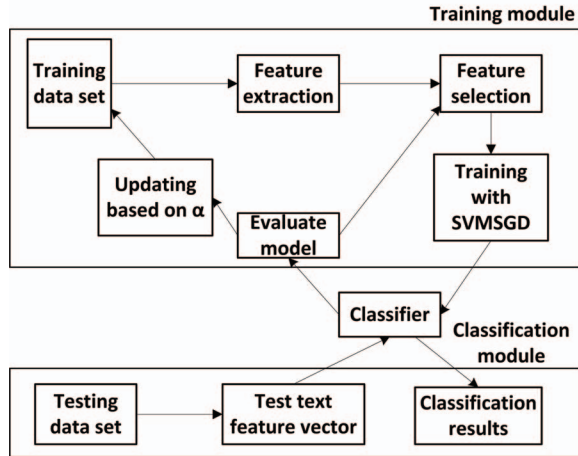


Fig. 2: Structure of  $\alpha$ -SVMSGD Algorithm

In training module, characteristic of data samples need to be extracted firstly, and then some characteristics are selected to train with SGD. In this way, the training of classifier model has been done. After this, the model needs to be evaluated with test data set. At the same time, classification module needs to provide test data set for the classifier model to evaluate the model. Test data set also needs to extract features similar as training data set. Then we can see the classification results in classification module. Especially, we provide  $\alpha$ -SVMSGD algorithm based on forgetting factor to evaluate the classifier

model in training module.

We use the following definitions to describe our proposed  $\alpha$ -SVMSGD algorithm.

**Definition 1.** The form of objective function:

$$obj(\omega) = (1/2\lambda\omega^2 + c \sum_{i=1}^n l(z_i, \omega)) \quad (1)$$

$$y_i(\omega^T x_i + b) - 1 \geq 0, i = 1, 2, \dots, n$$

where  $\omega$  is the parameter,  $(z_1, \dots, z_n)$  are the training examples,  $z_i = (x_i, y_i)$ ,  $y \in +1, -1$  is classification marker,  $1/2\lambda\omega^2$  is a regularization term,  $c$  is the penalization parameter, and  $l(z_i, \omega)$  is the loss function.

**Definition 2.** The form of loss function:

$$l(z_i, \omega) = \lambda\omega^2 + \max(0, 1 - y\omega^T f(x)) \quad (2)$$

where  $f(x) \in R^d$ ,  $\lambda > 0$ ,  $f(x)$  maps  $x$  to a  $d$ -dimensional separable space.

**Definition 3.** SGD selects a single example  $z$  and uses the equation (4) to update  $w$ :

$$\begin{cases} \omega_{t+1} = \omega_t - \gamma_t \lambda \omega_t & y_t \omega_t^T f(x_t) > 1 \\ \omega_{t+1} = \omega_t - \gamma_t \lambda \omega_t - y_t f(x_t) & otherwise \end{cases} \quad (3)$$

where  $\gamma_t$  is learning rate of the  $t$  step,  $\gamma$  is a regularization constant.

**Definition 4.** According to normal Karush-Kuhn-Tucker, we redefine some parameters in order to get more information from a mass of data. Optimal hyper plane satisfies the following condition.

$$\omega = \sum_i \alpha_i y_i H(x_i), 0 \leq \alpha_i \leq 1, \sum_i \alpha_i y_i = 0 \quad (4)$$

where  $\alpha_i = r_i/T_i$ ,  $r_i$  represents the number of support vectors for the  $i$ th sample after training.  $T_i$  represents the total number of test set training.  $\alpha_i$  denotes the forgetting factor which represents the supporting vector ratio of the  $i$ th sample after  $T$  times training of test set.

### B. Algorithm description

In this section, we introduce an incremental learning algorithm based on forgetting factor based on SVMSGD algorithm, which we call  $\alpha$ -SVMSGD. Firstly, it uses normal SVM to train classifier. Then, the trained classifier is used to predict the data. At each iteration, the SVMSGD processes a single sample in random [12]. If the data is not in any category, they need to be trained again in the classifier. This method can constantly optimize the classifier model. Then, the model will adapt to the new data environment. The mechanism can be divided into three steps. First we need to determine classifier and classifications of feature vector. The second step is to train new data sample based on forgetting. The third is to decide



**Algorithm 1**  $\alpha$ -SVMSGD Algorithm

---

**Inputs:**  $\lambda, l(), c$   
**Outputs:**  $\omega, \alpha$

---

**Procedure: Function**  $\alpha$ -SVMSGD( )

- 1 **Begin**
- 2 **Initialize**  $\beta, \gamma, \delta, n$
- 3  $obj(w) = 1/2\lambda\omega^2 + c \sum_{i=1}^n l(z_i, \omega)$  //Train classifier
- 4 SVM.SGD( )
- 5 **for**( $T=1$  to  $n$ )
- 6     **if**  $\alpha_i = 0$  or  $\delta \leq \alpha_i \leq 1$
- 7         StoreSample( )
- 8     **else if**  $\beta \leq \alpha_i < \delta$
- 9         StoreSample( $\alpha_i > \delta$ )
- 10    **else** Delete( $0 < \alpha_i < \gamma$ )
- 11    **end if**
- 12     $e_i = P - \alpha_i$
- 13    //Calculate the difference between training
- 14    sample and set threshold, where  $P = \{\beta, \gamma, \delta\}$
- 15    **end for**
- 16     $\alpha = max\{\beta, \gamma, \delta\}$  //Update  $\alpha$
- 17 **End**

---

how to deal with these data samples, delete, retain, or others. Here we explain each step of this algorithm in detail.

**S1: The determination of classifier and classifications of feature vector**

**S11** According to the number of classifiers, a relatively large sample set (usually need to have a positive and negative sample according to *Definition 1*) is established.

**S12** Determine the classifications of feature vectors. Training steps are shown as follows:

- Use classifier to classify the characteristics of all sample set;
- The relative values of each feature vector in each classifier are calculated separately according to the selected kernel function;
- Compute covariance matrix space according to the relative values of these features;
- Householder transforms to covariance matrix space;
- Calculate feature coefficient;
- Obtain model parameter according to *Definition 1, 2*.

**S2: Train new data sample based on forgetting factor**

**S21** Randomly choose subsets of the new data samples according to *Definition 3*.

**S22** Judge the correctness of the classification of sample set  $B_1$  according to the initial classifier  $\tau^1$ , and  $B_1$  is divided into test error set  $B_{err}$  and test set  $B_{ok}$ .

**S23** If  $B_{err}$  is empty, iteration is over. Choose a new sample set for training. If  $B_{err}$  is not empty, let the union of the support vector set  $A_{sv}^1$  in the original data sample and  $B_{err}$  be a new training set, then we can get a new classifier  $\tau^2$  and support vector set. In order to build a new incremental sample

set  $B_1'$  for the newly generated classifier, it needs to merge the remaining sample from set  $A$  with the set  $B_{ok}$ .

**S24** Repeat the steps above.

**S3: Delete and retain data sample**

In this step, we use *Definition 4* and get forgetting factor  $\alpha_i$  of each sample set. Specific methods are as follows:

**S31** In the step *S22*, each sample in  $B_{err}$  will be assigned to 0, that is  $r_i = 0$ ; each sample in  $B_{ok}$  will be assigned to 1, that is  $r_i = 1$ .

**S32** After  $T$  times training, we can get the value of the forgetting factor  $\alpha_i$ .

**S33** According to the prediction incremental learning mechanism based on the forgetting factor  $\alpha_i$ , the data sample is deleted or preserved.

We set up three threshold  $\beta, \gamma, \delta$  and adjust the ratio of delete and retention of data samples through adaptive technology flexibly. Processes are shown as follows:

Step 1 : set three threshold  $\beta = 0.3, \gamma = 0.4, \delta = 0.7$ ; (when  $\alpha$  is less than 0.3, we hold that the support vector (SV) ratio of data sample is very low, then we take these data samples as non-support vector (NSV). When  $\alpha$  is between 0.3 and 0.7, these data may be support vector samples, or non-supporting vector samples. When  $\alpha$  is greater than 0.7, we regard these data as support vector data.)

Step 2 : train data sample and get the  $\alpha$  values of the data sample;

Step 3 : use the following formula to calculate the updated error  $e_i$  between the sample and the set threshold after  $n$  times of training, and select maximum error weight of  $\alpha$  as a new threshold, so as to update the value of  $\beta, \gamma$  and  $\delta$ .

$$e_i = P - \alpha_i (1 \leq i \leq n) \quad (5)$$

where  $P$  is the set threshold  $\{\beta, \gamma, \delta\}$ .

Step 4: according to the rules to remove or retain data, rules are as follows:

If  $0 < \alpha_i < \beta$ , it means that after many times training, the support vector (SV) ratio of data sample is very low. We take these data samples as non-support vector (NSV), so deleting them will reduce the storage of raw data and improve the training speed.

If  $\beta \leq \alpha_i < \delta$ , select a sample  $\alpha \geq \gamma$  as the test set of next iteration, and accelerate the convergence speed of the support vector set.

If  $\delta \leq \alpha_i \leq 1$ , directly retain data set sample as the next test sample. The pseudo-code of  $\alpha$ -SVMSGD is shown in Algorithm 1.

## V. PERFORMANCE EVALUATION

## A. Simulation settings

This section shows the performance of  $\alpha$ -SVMSGD algorithm through three groups of simulations. In this paper, a server is simulated using python. We use two data sets, mnist9 [17] and rcv1 [18]. Among them, mnist9 contains 60000 data and rcv1 contains 8000 data. Mnist9 is mainly used in Fig. 3(a) and Fig. 3(b) while rcv1 is mainly used in Fig. 4(a) and Fig. 4(b). We set  $\lambda$  to  $10^{-5}$ .

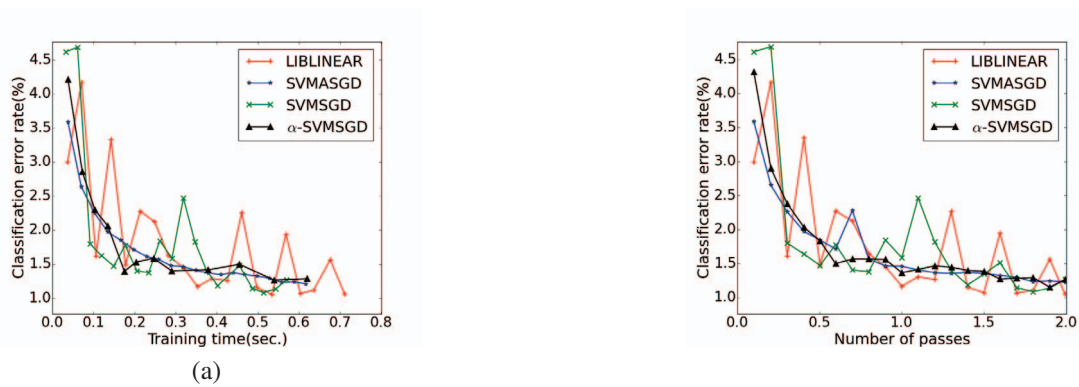


Fig. 3: (a) Classification error rate VS. training time (b) Classification error rate VS. number of passes

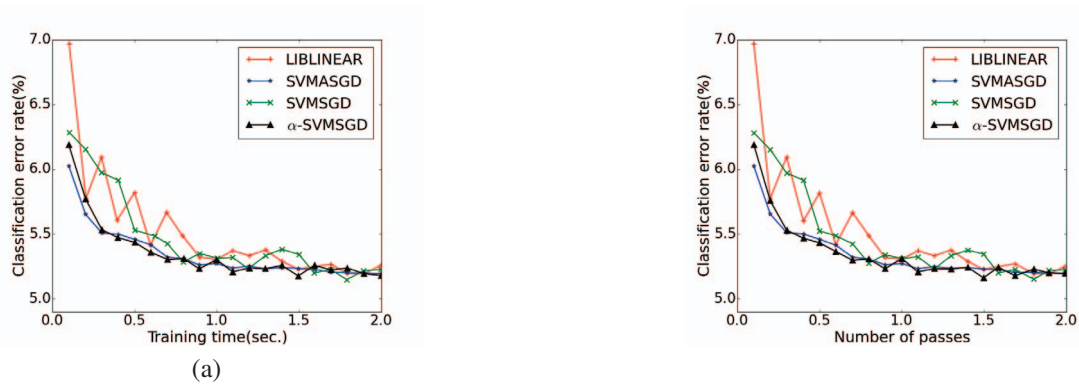


Fig. 4: (a) Classification error rate VS. training time (b) Classification error rate VS. number of passes

There are three groups of simulation being designed. The first group is to compare classification error rate with training time, number of passes and training size. We compare  $\alpha$ -SVMMSGD algorithm with LIBLINEAR [15], SVMMSGD [4] and SVMASGD [16]. The results are shown in Fig. 3-5. The second group is to compare NSV-rate changes with sample subset size in different  $\beta$ , and the results are shown in Fig. 6. The third group is to compare training speed among  $\alpha$ -SVMMSGD, SVMMSGD and SVMASGD and the results are shown in Fig. 7.

## B. Simulation results

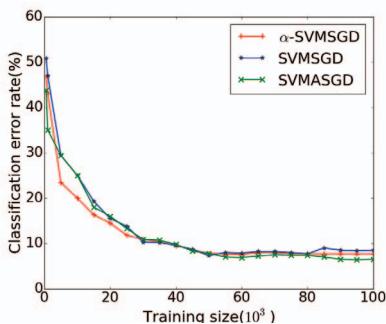


Fig. 5: Classification error rate VS. training size

1) **Comparison of Classification error rate:** Fig. 3(a) shows that data sample classification error rates of four algorithms are decreasing with training time. When the training

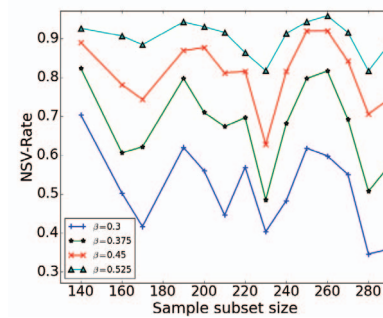


Fig. 6: NSV-rate VS. sample subset size in different values of  $\beta$

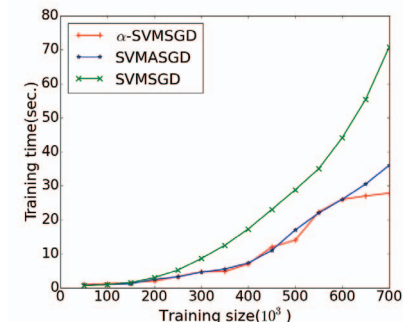


Fig. 7: Training time VS. training size

time is not more than 0.2, we can see that the lines of four algorithms classification error rate are all violent fluctuations, but SVMASGD and  $\alpha$ -SVMMSGD are relatively stable. With the extension of training time,  $\alpha$ -SVMMSGD is superior to the

other algorithms. Especially, when the training time is among 0.4 and 0.5, we can find it is better than SVMASGD.

Fig. 3(b) shows the changes of data sample classification error rate with number of passes. Training time is linear related to number of passes, so Fig. 3(b) is similar to Fig. 3(a). We can see that the effectiveness of  $\alpha$ -SVMSGD and SVMASGD is more stable than other two algorithms. But when the number of passes is between 0.5 and 1, our proposed algorithm is much better.

It is shown in Fig. 4(a) that classification error rates of four algorithms are decreasing with training time.  $\alpha$ -SVMSGD and SVMASGD are more stable than the other two. With the extension of training time,  $\alpha$ -SVMSGD is sometimes superior to SVMASGD. Especially, when the training time is in 1.5.

Similar to Fig. 3(b), we can clearly find that  $\alpha$ -SVMSGD has better stability and low error rate than the other three algorithms in Fig. 4(b), especially when the number of passes is less than 1.

Fig. 5 shows the classification error rate of three algorithms are decreasing with sample training size. We can see that when the data sample training size is less than 30000, the classification size is much better than the other two algorithms. With the increasing scale of data sample, the effect of  $\alpha$ -SVMSGD is similar to the others.

2) **NSV-rate changes with different  $\beta$** : Fig. 6 shows the relationship between sample subset and the ratio of NSV in different value of  $\beta$ . It is quite clear that the ratio of NSV will decrease with the decrease of  $\beta$ . Also, the value of  $\beta$  is not as small as possible. If it is too small, the ratio of NSV is instability.

3) **Training speed comparison**: It is illustrated in Fig. 7 that training time of three algorithm are all increasing with data sample size. As data sample training size is getting larger, the server of mobile eHealth networks will be under more pressure. Therefore, training time is increased. The effect of  $\alpha$ -SVMSGD is much better than SVMASGD and SVMSGD, especially when the size of data sample is larger.

## VI. CONCLUSION

The eHealth systems are faced with the challenges of rapid analysis and processing of increasing number of data. To solve the data redundancy issue in data filtering layer of our previous work LDPA, an incremental learning algorithm based on forgetting factor, i.e.,  $\alpha$ -SVMSGD, was proposed for mobile eHealth networks. Based on SVMSGD algorithm, we introduced forgetting factor mechanism to mine useful information from unclassified data, since in original SVMSGD, some past information in classifier may be new classifications. When more incremental data are collected from nodes, we put them into the classifiers which are already trained by SVMSGD. Forgetting factor method was utilized to decide how to deal with these redundant data. This method not only can improve the speed of data processing, but also can save storage space.

## ACKNOWLEDGEMENT

The authors would like to thank NSFC (61572262, 61100213, 61571233, 61373135, 61572172); ZTE

Research Cooperation Project (ZTE20160106); SFDPH (20113223120007); NSF of Jiangsu Province (BK20141427), NUPT (NY214097); Priority Academic Program Development of Jiangsu Higher Education Institutions; Open research fund of Key Lab of Broadband Wireless Communication and Sensor Network Technology (NUPT), Ministry of Education (NYKL201507); Huawei Innovation Research Program (YB2014010048).

## REFERENCES

- [1] M. Viceconti, P. Hunter, and R. Hose, "Big Data, Big Knowledge: Big Data for Personalized Healthcare," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 4, pp. 1209-1215, 2015.
- [2] A. Hristova, A. M. Bernardos, and J. R. Casar, "Context-aware services for ambient assisted living: A case-study," *Proceedings of the First International Symposium on Applied Sciences on Biomedical and Communication Technologies*, 2008, pp. 1-5.
- [3] K. Wang, Y. Shao, L. Shu, et al., "LDPA: a local data processing architecture in ambient assisted living communications," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 56-63, 2015.
- [4] S. Shalev-Shwartz, Y. Singer, N. Srebro, et al., "Pegasos: Primal estimated sub-gradient solver for svm," *Mathematical programming*, vol. 127, no. 1, pp. 3-30, 2011.
- [5] S. Meng, W. Dou, X. Zhang, et al., "Kasr: a keyword-aware service recommendation method on mapreduce for big data applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3221-3231, 2014.
- [6] Y. Liu and Y. Liu, "Incremental learning method of least squares support vector machine," *IEEE International Conference on Intelligent Computation Technology and Automation*, 2010, pp. 529-532.
- [7] K. Wang, Y. Shao, L. Shu, et al., "Mobile big data fault-tolerant processing for eHealth networks," *IEEE Network*, vol. 30, no. 1, pp. 36-42, 2016.
- [8] K. Wang, L. Zhuo, L. Shu, et al., "A model-matching algorithm based on improved BP over out-of-order streams," in *Proc. IEEE ICC*, 2014, pp. 1849-1853.
- [9] K. Wang, Y. Yu, and B. Liu, "DAS: A dynamic assignment scheduling algorithm for stream computing in distributed applications," in *Proc. IEEE GLOBECOM*, 2013, pp. 1654-1659.
- [10] W. Dou, X. Zhang, J. Liu, et al., "Hiresome-ii: towards privacy-aware cross-cloud service composition for big data applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 2, pp. 455-466, 2013.
- [11] J. Yu, F. Jiang, T. Zhu, "RTIC-C: A Big Data System for Massive Traffic Information Mining," *IEEE International Conference on Cloud Computing and Big Data*, 2013, pp. 395-402.
- [12] H. Liang, L. Jin, L. Zhao, "A new data stream classification algorithm," *2013 International Conference on Measurement, Information and Control*, 2013, pp. 477-481.
- [13] Y. Zhang, R. Yu, W. Yao, et al., "Home M2M Networks: Architectures, Standards, and QoS Improvement," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 44-52, 2011.
- [14] D. He, S. Chan, Y. Zhang, et al., "Lightweight and Confidential DataDiscovery and Dissemination for Wireless Body Area Networks," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 2, pp. 440-448, 2014.
- [15] R. Fan, K. Chang, C. Hsieh, et al., "LIBLINEAR: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871-1874, 2008.
- [16] W. Xu, "Towards optimal one pass large scale learning with averaged stochastic gradient descent," *Technical report, arXiv:1107.2490*, 2011.
- [17] Y. LeCun, L. Bottou, Y. Bengio, et al., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.
- [18] D. Lewis, Y. Yang, T. Rose, et al., "RCV1: A new benchmark collection for text categorization research," *Journal of Machine Learning Research*, vol. 5, pp. 361-397, 2004.