

A New Algorithm for Reducing Dimensionality of L1-CSVM Use Augmented Lagrange Method

Mingzhu Cui, Liya Fan*

School of Mathematics Science, Liaocheng University, Liaocheng, China

Email: *fanliya63@126.com

How to cite this paper: Cui, M.Z. and Fan, L.Y. (2022) A New Algorithm for Reducing Dimensionality of L1-CSVM Use Augmented Lagrange Method. *Journal of Applied Mathematics and Physics*, 10, 21-30.

<https://doi.org/10.4236/jamp.2022.101003>

Received: December 2, 2021

Accepted: January 8, 2022

Published: January 11, 2022

Copyright © 2022 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Principal component analysis and generalized low rank approximation of matrices are two different dimensionality reduction methods. Two different dimensionality reduction algorithms are applied to the L1-CSVM model based on augmented Lagrange method to explore the variation of running time and accuracy of the model in dimensionality reduction space. The results show that the improved algorithm can greatly reduce the running time and improve the accuracy of the algorithm.

Keywords

Support Vector Machine, Dimensionality Reduction, Augmented Lagrange Method, Classification

1. Introduction

Support Vector Machine (SVM) [1] is a machine learning method based on statistical learning and optimization theory, first proposed by Vapnik *et al.* in 1995. SVM is a widely used binary classification model. The basic idea is to find a hyperplane and divide sample points into two categories. The basic idea of Principal Component Analysis (PCA) [2] [3] is to transform the original multi-variable data into new variables with fewer than before through linear transformation. Considering the dimensionality reduction, the data can still maintain the maximum separability between samples and the recent reconstruction of samples. However, PCA uses vector data, which requires a large amount of calculation in the process of singular value decomposition. In order to improve the disadvantage of large PCA calculation, YE proposed a generalized low rank approximation of matrices (GLRAM) method in 2005 [4] [5]. The data set used in this method is a collection of matrices. The optimization process is to use the

method of bilinear transformation for iteration to minimize the reconstruction error, which can lead to lower computational cost.

Sparse representation optimization is widely used. L1 norm is usually used to regularize penalty terms to improve the classification accuracy of network models. The L1 norm uses fewer features for classification by selecting. L1 norm is considered to be an effective method to reduce the influence of outliers [6] [7] [8]. In 2014, Nie *et al.* used Augmented Lagrange method (ALM) [9] [10] in SVM [11] to solve the classification problem of linear big data. In 2019, Chauhan *et al.* summarized the linear SVM [12] and pointed out that the algorithm of Nie *et al.* was the fastest method to solve the original problem. In 2019, Yan *et al.* proposed the algorithm for ALM to solve L1-CSVM [13].

Based on the study of the above theory, this paper takes feature dimension reduction as the starting point to enhance the performance of image binary classification algorithm. PCA and GLRAM are introduced to improve the algorithm model of L1-CSVM based on ALM. The experimental results show that when calculating the L1-CSVM model based on dimensionality reduction, it can significantly reduce the calculation time and generally improve the accuracy due to the reduction of the number of data features.

The following contents of this article are as follows. In Section 2, PCA and GLRAM are introduced. In Section 3, PCA-L1-CSVM and GLRAM-L1-CSVM are described. In Section 4, the content and results of comparative experiments on MNIST and Fashion-MNIST datasets are introduced in detail. Finally, the conclusion of this paper is in Section 5.

2. Related Work

This section mainly introduces two dimensionality reduction algorithms: PCA and GLRAM.

2.1. PCA

Given data $T = \{(x_1, x_2, \dots, x_m)\} \in R^{d \times m}$, Let's say we have m samples, and each sample corresponds to d attributes. Take any centralized data $\tilde{x}_i \in \mathbb{R}^d$, A new coordinate system composed of orthonormal basis vectors is obtained by projection transformation $\{v_1, v_2, \dots, v_{d'}\} (d' \leq d)$, $\|v_i\|_2 = 1$, $v_i^T v_j = 0 (i \neq j)$. The projection of \tilde{x}_i in my new coordinates is $s_i = (s_{i1}; s_{i2}; \dots; s_{id'})$, meet $s_{ij} = v_j^T \tilde{x}_i$ and $\hat{x}_i = \sum_{j=1}^{d'} s_{ij} v_j$. Therefore, the following optimization model can be established

$$\begin{aligned} & \sum_{i=1}^m \left\| \sum_{j=1}^{d'} s_{ij} v_j - \tilde{x}_i \right\|_2^2 \\ & \text{s.t. } \|v_i\|_2 = 1, v_i^T v_j = 0 (i \neq j) \end{aligned} \quad (1)$$

The orthogonal coordinate system can be obtained by solving (1), from

which low-dimensional characteristic data $s_i = (s_{i1}; s_{i2}; \dots; s_{id'}) \in R^{d' \times m}$ can be obtained. Thus, the original high-dimensional data set T is transformed into sparse representation data set $T^r = \{(r_i, y_i)\}_{i=1}^m \in R^{d'} \times \{\pm 1\}$.

2.2. GLRAM

GLRAM minimizes the reconstruction error of the new matrix and the original matrix by reducing the dimension of the matrix data, and uses the low-rank matrix set to approximate the original matrix set to complete data compression. Given data $T = \{(X_i, y_i)\}_{i=1}^m \in R^{r \times c} \times \{\pm 1\}$, Let's say we have m samples, and each sample corresponds to $r \times c$ attributes. We want to find two orthogonal transformation matrices $L \in R^{r \times q_1}$, $P \in R^{c \times q_2}$ and $M_i \in R^{q_1 \times q_2}, i = 1, \dots, m$, make LM_iP^T as close as possible to X_i .

The following optimization model is established

$$\min_{\substack{L \in R^{r \times q_1}; L^T L = I_{q_1} \\ P \in R^{c \times q_2}; P^T P = I_{q_2} \\ M_i \in R^{q_1 \times q_2}, i=1, \dots, m}} \sum_{i=1}^m \|X_i - LM_iP^T\|_F^2 \tag{2}$$

Adopt iterative method, fix P first, use $M_L = \sum_{i=1}^m X_i P P^T X_i^T$ get the q_1 largest eigenvalue pairs $L \in R^{r \times q_1}$ of M_L . Then fix L , use $M_P = \sum_{i=1}^m X_i^T L L^T X_i$ get the q_2 largest eigenvalue pairs $P \in R^{c \times q_2}$ of M_P . And finally, solve M_i through $M_i = L^T X_i P$. Repeat the solution until the requirements are met. Then get new dimension reduction dataset $T' = \{(X_1, X_2, \dots, X_m)\} \in R^{q_1 \times q_2}, i = 1, \dots, m$. In this paper, each sample after dimensionality reduction is pulled into column vectors to form sample matrix $T^r = \{(r_i, y_i)\}_{i=1}^m \in R^{d'} \times \{\pm 1\}$ similar to section 2.1.

3. Proposed Model

The algorithm proposed in this chapter is based on article (13), mainly introduces two improved models: PCA-L1-CSVM and GLRAM-L1-CSVM. The first model uses data set is $T = \{(x_i, y_i)\}_{i=1}^m \in R^d \times \{\pm 1\}$. The second model uses data set is $T = \{(X_i, y_i)\}_{i=1}^m \in R^{r \times c} \times \{\pm 1\}$. $A = [x_1^+, \dots, x_{m_1}^+]$, $B = [x_1^-, \dots, x_{m_2}^-]$, $X = [A, B]$ represents the matrix of the positive sample, the matrix of the negative sample, and the matrix of the population sample. $I_1 = \{i \in \{1, \dots, m_1\} : y_i = 1\}$ and $I_2 = \{j \in \{1, \dots, m_2\} : y_j = -1\}$ represents the subscript set of positive and negative data respectively.

3.1. PCA-L1-CSVM

PCA-L1-CSVM uses dimension reduction dataset $T^r = \{(r_i, y_i)\}_{i=1}^m \in R^{d'} \times \{\pm 1\}$. The following planning model is constructed

$$\begin{aligned} \min_{w,b,r_i,v_i,\xi_i} & \frac{1}{2}\|w\|^2 + C\sum_{i \in I} \xi_i \\ \text{s.t.} & y_i(w^T r_i + b) \geq 1 - \xi, \xi_i \geq 0, i = 1, \dots, m, \\ & \min_{r_i} \sum_{i=1}^m \left\| \sum_{j=1}^{d'} r_{ij} v_j - \tilde{x}_i \right\|_2^2 \quad \text{s.t.} \quad \|v_i\|_2 = 1, v_i^T v_j = 0 (i \neq j). \end{aligned} \tag{3}$$

where $C > 0$ is a penalty parameter, the second condition of the constraint is to use PCA dimensionality reduction for the original data set. First, PCA is used to calculate the data after dimensionality reduction, and then the model is represented as an unconstrained model

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^m \max(1 - y_i(w^T r_i + b), 0) + 1 \tag{4}$$

Because bias has little effect on model performance, bias is often ignored in recent studies, and $r_i^T \leftarrow [r_i^T, 1] \in R^{d'+1}$, $w^T \leftarrow [w^T, b] \in R^{d'+1}$ for convenience, this section $d'+1$ is denoted as d' . The model (4) was rewritten as an unconstrained quadratic programming model

$$\min_w f(w) := \frac{1}{2}\|w\|^2 + C\sum_{i=1}^m \max(1 - y_i(w^T r_i), 0) \tag{5}$$

Model (5) is expressed in matrix form

$$\min_w \frac{1}{2}\|w\|^2 + C \max\|0, Bw + d\| \tag{6}$$

where

$$B = - \begin{bmatrix} y_1 r_1^T \\ \vdots \\ y_m r_m^T \end{bmatrix} \in R^{m \times d'}, \quad d = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in R^{d'}.$$

Introducing variable $s \in R^{d'}$, we obtain the following constrained optimization problem

$$\begin{aligned} \min_w & \frac{1}{2}\|w\|^2 + P(s) \\ \text{s.t.} & s = Bw + d, \end{aligned} \tag{7}$$

and $P(s) = C \max\|0, s\|$.

Consider the ALM function of model (7) above:

$$L_\sigma(w, s, \lambda) = \frac{1}{2}\|w\|^2 + P(s) - \lambda^T (s - Bw - d) + \frac{\sigma}{2}\|s - Bw - d\|^2$$

At iteration k , solve

$$\min_{w,s} L_{\sigma^k}(w^k, s^k, \lambda^k), \tag{8}$$

obtain (w^{k+1}, s^{k+1}) , and then update the Lagrange multiplier λ^{k+1} .

3.2. GLRAM-L1-CSVM

PCA-L1-CSVM uses dimension reduction dataset

$T' = \{(X_1, X_2, \dots, X_m)\} \in R^{q_1 \times q_2}, i \in I$. The following planning model is constructed

$$\begin{aligned} \min_{w, b, r_i, v_i, \xi_i} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i \in I} \xi_i \\ \text{s.t.} \quad & y_i (w^T r_i + b) \geq 1 - \xi, \quad \xi_i \geq 0, i = 1, \dots, m, \end{aligned} \quad (9)$$

$$\min_{r_i} \sum_{i=1}^m \|X_i - LM_{i,R} P^T\|_F^2 \quad \text{s.t.} \quad L^T L = I_{q_1}, P^T P = I_{q_2}$$

where $C > 0$ is a penalty parameter, different from the second constraint condition in formula (3), the second constraint of formula (9) is to use GLRAM to reduce the dimension of the original data. First, GLRAM is used to reduce the dimension of the data, and pull the data matrix into column vectors. Then, the model is transformed into an unconstrained optimization model as (4). Then, the following operations are solved using the same method as in Section 3.1.

4. Experiment and Result Analysis

In order to test the effectiveness of **Algorithm 1** and GLRAM-L1-CSVM, a series of comparative experiments are conducted on MNIST and fashion-MNIST data in this section. Each dataset contains 10 image types, each of which has a pixel size of 28×28 . Eight groups of data were randomly selected from each data set and divided into four groups for binary comparison test. This experiment only uses the training set data of each kind of data set and divides it into training set and test set. The size of the training set is 3000, 4000 and 5000 respectively, accounting for 50%, 66% and 83% of the total data set, and the rest is the test set. In the experiment, the eigenvalues of 81, 100 and 121 after dimensionality reduction were used. All the features in the table refer to the features of the improved algorithm.

The experimental platform used in this experiment is matlab R2014B, and the Windows specification is Windows 10. The device specifications are as follows: The processor is Inter(R) Core (TM) I7-10700 (2.90 GHz), and the onboard RAM is 8 GB.

Algorithm 1. PCA-L1-CSVM.

Step 1. Initialize, given parameters $\rho, \sigma \in (0, 1)$, $\eta_0 > 0$, $\eta_1 > 0$, select w_0 , set $k = 0$, given error $\varepsilon > 0$ and maximum number of iterations N .

Step 2. PCA is used to reduce the dimension of the data set.

Step 3. By calculating (8), get w^{k+1} and use Moreau-Yosida regularization obtain $s^{k+1} = s^*(w^{k+1})$.

Step 4. Update the Lagrange multiplier $\lambda^{k+1} = \lambda^k - \sigma_k (s^{k+1} - Bw^{k+1} - d)$.

Step 5. Select $\sigma_{k+1}, \sigma_k \leq \sigma_{k+1} < +\infty$, go to step 2.

A bar chart is used to represent the comparison of the accuracy of the first two groups of binary classification experiments of different algorithms, and a broken line chart is used to represent the running time of each algorithm in the first two groups of experiments. The bar charts and line charts from left to right of each image are the data results of dimensionality reduction to 81, 100, 121, respectively. The average results of each algorithm under the latter two groups of experiments are shown in the table. The data in bold is the best data, the time unit is second, and the precision unit is %.

4.1. Experimental Results on MNIST Datasets

The experimental results on the MNIST dataset are presented.

According to **Figure 1**, **Figure 2**, we can find that the two improved algorithms generally improve the accuracy of operation. According to **Figure 3**, **Figure 4**, it can be found that the original algorithm takes a long time, while the two improved algorithms can significantly shorten the running time. According to **Table 1**, it can also be concluded that the improved algorithm can generally improve accuracy. **Table 2** shows that the time of the two improved algorithms is about 10% of the original algorithm. Comparing the two improved algorithms, it can be found that GLRAM shows better results in both accuracy and time than PCA.

4.2. Experimental Results on Fashion-MNIST Datasets

Next, the experimental results on the Fashion-MNIST dataset are presented.

It can be seen from **Figure 5** and **Figure 6** that the two improved algorithms generally improve the accuracy of operation. The accuracy of GLRAM is improved obviously and has stability. As can be seen from **Figure 7**, **Figure 8**, the two improved algorithms can shorten the running time. It can be seen from **Table 3** that the improved algorithm can improve the accuracy on the whole, the results of PCA are floating, and the overall results of GLRAM are optimal in comparison of the three results. It can be seen from **Table 4** that the time of the two improved algorithms is about 14% of the original algorithm. Comparing the two improved algorithms, the algorithm with GLRAM is better than that with PCA in accuracy and time.

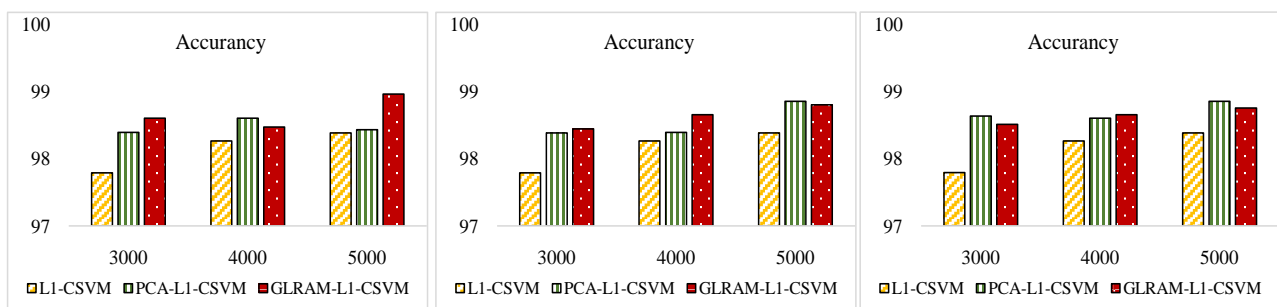


Figure 1. Precision comparison of the first set of MNIST data.

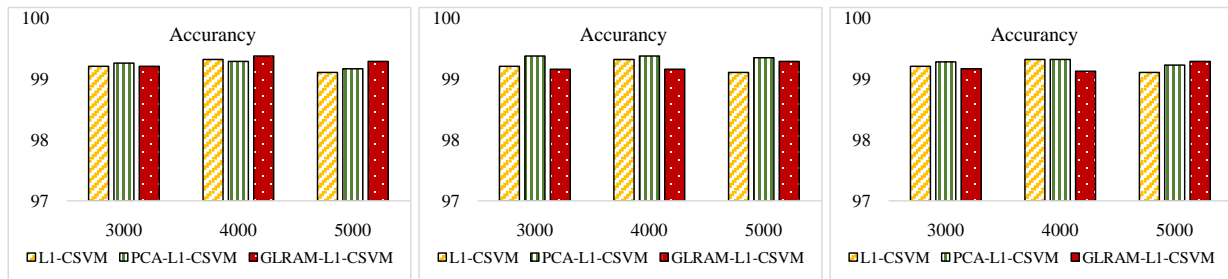


Figure 2. Precision comparison of the second set of MNIST data.

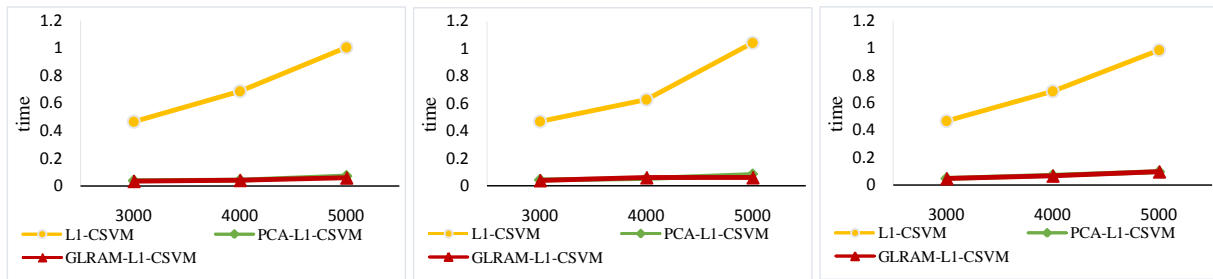


Figure 3. The running time of the first set of MNIST data.

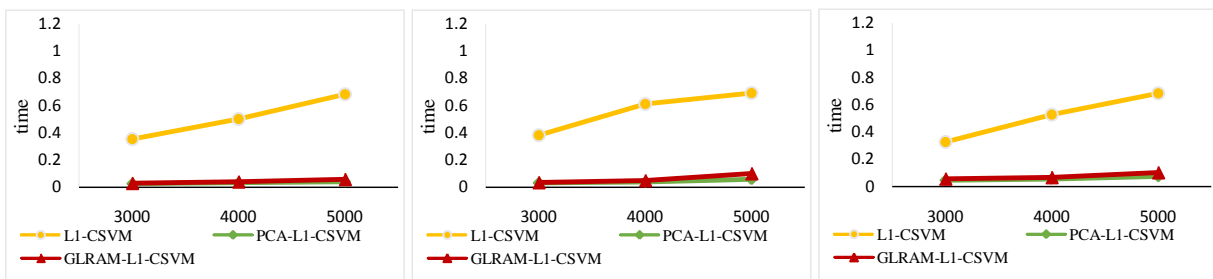


Figure 4. The running time of the second set of MNIST data.

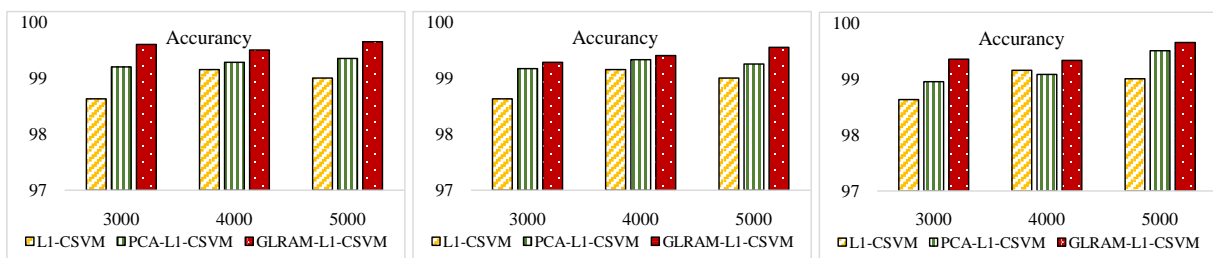


Figure 5. Precision comparison of the first set of Fashion-MNIST data.

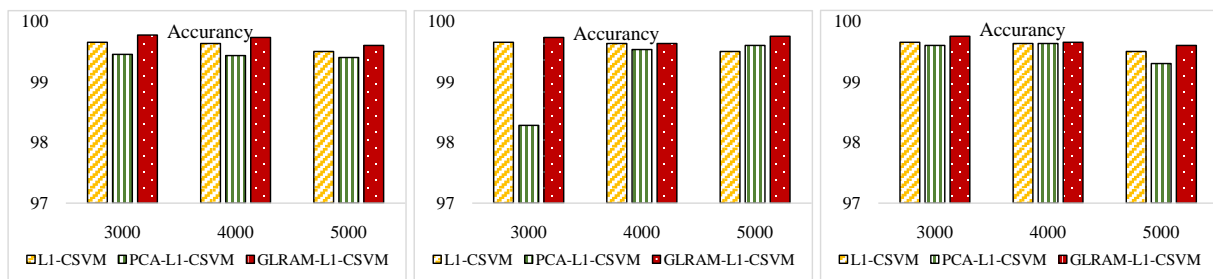


Figure 6. Precision comparison of the second set of Fashion-MNIST data.

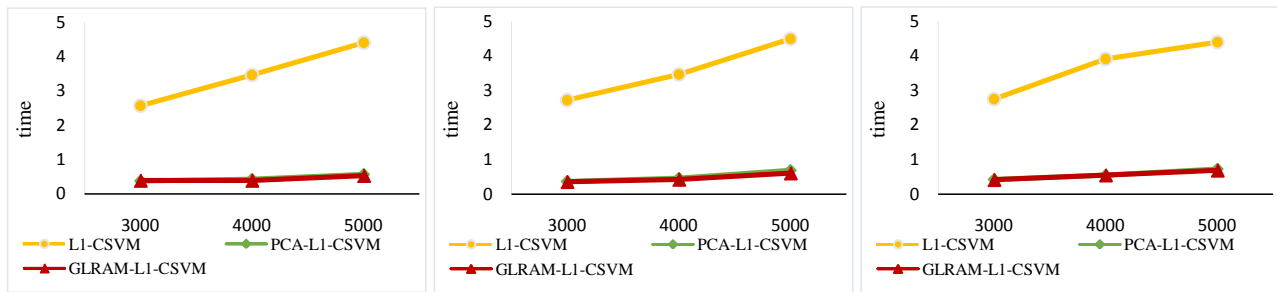


Figure 7. The running time of the first set of Fashion-MNIST data.

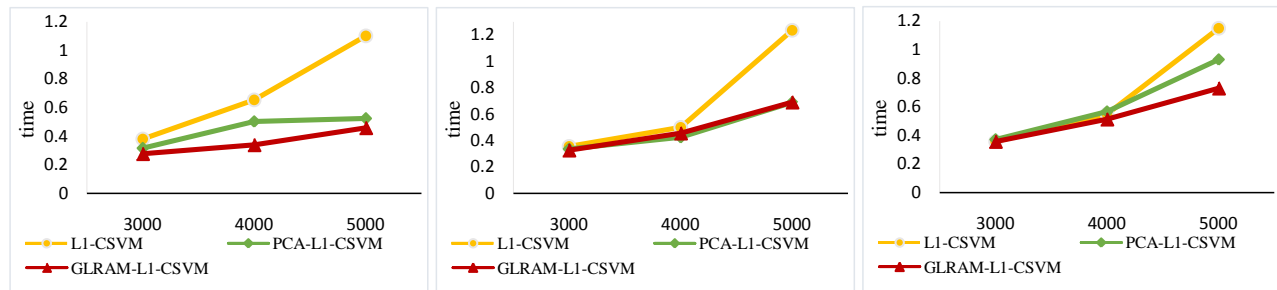


Figure 8. The running time of the second set of Fashion-MNIST data.

Table 1. Average Test accuracy about MNIST data.

TN	Feature	L1-CSVM	PCA-L1-CSVM	GLRAM-L1-CSVM
3000	81	98.67	98.95	99.09
	100	98.67	98.96	99.02
	121	98.67	99.08	99.00
4000	81	98.91	99.02	99.00
	100	98.91	98.98	99.11
	121	98.91	99.05	99.13
5000	81	99.00	99.16	99.32
	100	99.00	99.40	99.24
	121	99.00	99.37	99.21

Table 2. Average Test Time about MNIST data.

TN	Feature	L1-CSVM	PCA-L1-CSVM	GLRAM-L1-CSVM
3000	81	0.3212	0.0362	0.0295
	100	0.3177	0.0397	0.0376
	121	0.3088	0.0446	0.0448
4000	81	0.5138	0.0424	0.0401
	100	0.4980	0.0496	0.0522
	121	0.5114	0.0644	0.0641
5000	81	0.9036	0.0657	0.0544
	100	0.9132	0.0729	0.0584
	121	0.8660	0.0914	0.0906

Table 3. Average accuracy about Fashion-MNIST data.

TN	Feature	L1-CSVM	PCA-L1-CSVM	GLRAM-L1-CSVM
3000	81	99.27	99.59	99.80
	100	99.27	99.54	99.62
	121	99.27	99.44	99.63
4000	81	99.55	99.58	99.74
	100	99.55	99.64	99.67
	121	99.55	99.48	99.63
5000	81	99.50	99.65	99.78
	100	99.50	99.55	99.73
	121	99.50	99.65	99.78

Table 4. Average Test Time about Fashion-MNIST data.

TN	Feature	L1-CSVM	PCA-L1-CSVM	GLRAM-L1-CSVM
3000	81	1.4102	0.1978	0.2080
	100	1.4857	0.1947	0.1926
	121	1.5022	0.2272	0.2272
4000	81	1.9142	0.2294	0.2128
	100	1.9132	0.2491	0.2306
	121	2.1364	0.2964	0.3006
5000	81	2.4396	0.3044	0.3072
	100	2.4949	0.3720	0.3294
	121	2.4380	0.3917	0.3807

5. Conclusion

In this paper, the ALM is applied to solve two models: PCA-L1-CSVM and GLRA-L1-CSVM. The main contribution of this paper is as follows: Firstly, principal component analysis or GLRAM method is introduced to reduce dimension of data set. Then, the data after dimensionality reduction is dichotomized using L1-CSVM. Experimental results show that the improved model can greatly reduce the running time while maintaining a general increase in accuracy. In the future, we hope to use ALM to solve the original problems of other models.

Acknowledgements

Supported by Natural Science Foundation of Shandong Province, Funded Project Nos. ZR2020MA026, ZR2018BF010.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Cortes, C. and Vapnik, V. (1995) Support Vector Networks. *Machine Learning*, **20**, 273-297. <https://doi.org/10.1007/BF00994018>
- [2] Mohammed, A.A., Minhas, R., Wu, Q.M.J. and Sid-Ahmed, M.A. (2011) Human Face Recognition Based on Multidimensional PCA and Extreme Learning Machine. *Pattern Recognition*, **44**, 2588-2597. <https://doi.org/10.1016/j.patcog.2011.03.013>
- [3] Majumdar, A. (2009) Image Compression by Sparse PCA Coding in Curvelet Domain. *Signal, Image and Video Processing*, **3**, 27-34. <https://doi.org/10.1007/s11760-008-0056-5>
- [4] Ye, J.P. (2005) Generalized Low Rank Approximations of Matrices. *Machine Learning*, **61**, 167-191. <https://doi.org/10.1007/s10994-005-3561-6>
- [5] Ahmadi, S. and Rezghi, M. (2020) Generalized Low-Rank Approximation of Matrices Based on Multiple Transformation Pairs. *Pattern Recognition*, **108**, Article ID: 107545. <https://doi.org/10.1016/j.patcog.2020.107545>
- [6] Wang, H., Lu, X., Hu, Z. and Zheng, W. (2013) Fisher Discriminant Analysis with l1-Norm. *IEEE Transactions on Cybernetics*, **44**, 828-842. <https://doi.org/10.1109/TCYB.2013.2273355>
- [7] Ye, Q., Zhao, H., Yang, X. and Ye, N. (2017) L1-Norm Distance Minimization Based Fast Robust Twin Support Vector K-Plane Clustering. *IEEE Transactions on Neural Networks and Learning Systems*, **29**, 4494-4503. <https://doi.org/10.1109/TNNLS.2017.2749428>
- [8] Li, C.H., Shao, Y.H. and Deng, N.Y. (2015) Robust l1-Norm Non-Parallel Proximal Support Vector Machine. *Optimization*, **65**, 1-15. <https://doi.org/10.1080/02331934.2014.994627>
- [9] Han, D.Y., Tang, Q.H., Zhang, Z.K., Yuan, L. and Li, J. (2021) An Efficient Augmented Lagrange Multiplier Method for Steelmaking and Continuous Casting Production Scheduling. *Chemical Engineering Research and Design*, **168**, 169-192. <https://doi.org/10.1016/j.cherd.2021.01.035>
- [10] Shi, C.X. and Yang, G.H. (2018) Augmented Lagrange Algorithms for Distributed Optimization over Multi-Agent Networks via Edge-Based Method. *Automatica*, **94**, 55-62. <https://doi.org/10.1016/j.automatica.2018.04.010>
- [11] Nie, F., Huang, Y., Wang, X. and Huang, H. (2014) New Primal SVM Solver with Linear Computational Cost for Big Data Classifications. *Proceedings of the 31st International Conference on Machine Learning*, **32**, 505-513.
- [12] Chauhan, V.K., Dahiya, K. and Sharma, A. (2019) Problem Formulations and Solvers in Linear SVM: A Review. *Artificial Intelligence Review*, **52**, 803-855. <https://doi.org/10.1007/s10462-018-9614-6>
- [13] Yan, Y.Q. and Li, Q.N. (2019) An Efficient Augmented Lagrangian Method for Support Vector Machine. arXiv e-prints, arXiv:1912.06800.