ISSN 2616-5805 Vol. 5, Issue 3: 67-74, DOI: 10.25236/AJMS.2024.050310

Least Squares Generalization-Memorization Machines

Shuai Wang[1,a,*]

[1]*School of Mathematics and Statistics, Hainan University, Haikou, China*
[a]*wangshuai282615@163.com*
[*]*Corresponding author*

Abstract: *In this paper, a new generalized memory mechanism is proposed, which makes it possible to partition the training set accurately without changing the equation constraints of the original least squares loss by using a new memory influence function that allows the model to avoid overfitting. We propose Least Squares Generalization-Memorization Machines (LSGMM) and give a memory influence function suitable for this model. Experimental results show that our LSGMM has better generalization performance and significant memory capability compared to the least squares support vector machine. Meanwhile, it has a significant advantage in terms of time cost compared with other memory models.*

Keywords: *Least squares support vector machine, Generalization-memorization mechanism, Generalized memory, Memorization kernel*

1. Introduction

Expected risk minimization is one of the most important goals of machine learning. However, due to the finite and unknown nature of the data, the expected risk cannot be calculated directly [1]. According to the law of large numbers, it is known that when the sample capacity tends to infinity, the empirical risk will tend to be the expected risk [2], so the current machine learning models mainly use the empirical risk to approximate the expected risk. However, the excessive pursuit of minimizing the empirical risk can easily lead to the phenomenon of overfitting [3], so how to minimize the empirical risk while ensuring the predictive ability of the model has become an important research topic.

In 1995, Vapnik and Cortes [4] first proposed a Support Vector Machine (SVM) based on statistical learning theory. When all training samples are classified correctly, at this point, the model is said to achieve zero empirical risk, or the model is said to have memory capability. In 2021, Vapnik and Izmailov [5] first introduced memory mechanisms into SVMs, proposing memory theory. The theory introduced two weighted radial bases (RBF) kernel functions in the dyadic problem of SVMs[6] and the concept of the generalization-memory kernel, which theoretically proved that SVMs could improve their generalization performance by achieving zero empirical risk. Wang and Shao[7]further proposed more general generalization-memorization machines on this basis Generalisation-Memorization Machine (GMM), which introduce the memory mechanism in the primal problem of SVM instead of the dual problem, gives the primal problem of memory-theoretic SVM based on memory theory, and illustrates the conditions and ways to achieve zero-empirical risk for SVM from the theoretical point of view. However, like SVM, the model still has high training time costs in the face of large-scale data. In addition, the model leads to a large training cost in achieving zero-experience risk due to its complex memory mechanism.

Therefore, from the perspective of minimizing the empirical risk and based on the memory theory, this paper, on the one hand, transforms the convex quadratic programming problem into the problem of solving a system of linear equations by introducing the least squares term and the memory term, which significantly reduces the time cost of the SVM based on the memory theory to process large-scale data. On the other hand, this paper also proposes a new memory influence function, which makes our proposed Least Squares Generalization-Memorization Machines (LSGMM) achieve zero empirical risk, provides new ideas for memory theory to explore data on other models further, and enriches the theory and methods of machine learning.

The following section provides a brief overview of the most recently proposed memory theories and memory kernel functions and reviews the GMM model. Section 4 describes the construction and solution

ISSN 2616-5805 Vol. 5, Issue 3: 67-74, DOI: 10.25236/AJMS.2024.050310

of the LSGMM model proposed in this paper and gives the new objective function and memory mechanism. Section 5 gives numerical experiments to validate the proposed LSGMM model. Finally, we summarise the main points and innovations of this paper and provide ideas for subsequent research.

2. Research status

The leading indicator of the generalization ability of a machine learning model is the expected risk. According to the law of large numbers, when the sample size tends to be infinite, the empirical risk will tend to be the expected risk, so the empirical risk is generally used in machine learning models to approximate instead of the expected risk, which is difficult to measure directly. Therefore, the expected risk can be minimized by minimizing the error of the model on the training data, i.e., by 'remembering' the laws embedded in the training data. Based on this, a large number of scholars have researched machine learning 'memory' models based on traditional and neural networks.

In traditional machine learning models, Vapnik and Izmailov[6] first proposed a theory of memory for SVMs in 2021. The authors proposed a generalized memory kernel based on a weighted form of two RBF kernels and applied it to SVMs. Zero empirical risk can be achieved by appropriately tuning the parameters of the generalization memory kernel, which remember the errors generated by the training samples. Wang and Shao[7] proposed a Hard Generalization-Memorization Machine (HGMM) and a Soft Generalization-Memorization Machine (SGMM). They then innovatively introduced principles of generalized memory decision-making and memory modeling, and theoretically demonstrated that the model can achieve strong generalization performance even with zero empirical risk. However, although these methods reduces the model's error on training data, they do not effectively improve the efficiency of SVM in handling large-scale data, which inevitably leads to high training time costs.

Neural networks, as a prevalent direction of machine learning in recent years, have been extensively studied for zero experience risk. Li et al.[8] proposed a method of forgetting-free learning, which is based on convolutional neural networks[9] and combines knowledge distillation and parameter fine-tuning to increase robustness while reducing the experience risk. Kim et al.[10] proposed an incremental learning model based on regularisation for image classification tasks, which reduces the impact of migrating new knowledge on old knowledge through maximum entropy regularisation, thus improving the model's risk resistance.Lopez-Paz et al.[11] proposed a model based on maximum entropy regularisation to reduce the effect of migrating new knowledge on old knowledge, thus improving the model's risk resistance. Recent studies [12,13] have shown that deep neural networks can achieve almost zero empirical risk with good generalization performance. However, it consumes vast computational power, memory, storage, etc., due to its lower interpretability and longer training time.

In summary, due to the low interpretability on neural networks, the classification-related studies on Least Squares Support Vector Machine (LSSVM) [14] memory theory still remain under-explored. Therefore, based on memory theory and SVM models, the proposed LSGMM in this paper focuses on exploring how to improve and optimize some of the problems of existing memory mechanisms and memory models. This paper proposes a novel memory mechanism which contains two memory models in the least squares sense, i.e., the LSGMM. While ensuring zero empirical risk, its learning rate is much faster than GMM and SVM.

3. Review

In this paper, we delve into a classification problem situated in $\mathbb{R}^n$. The collection of training samples is represented by $\mathrm{T} = \{(x_i, y_i) | i = 1, 2, ..., m\}$, in which $x_i \in \mathbb{R}^n$ denotes the input and $\mathrm{y}_i \in \{-1, +1\}$ signifies the corresponding true labels. These training samples and their true values are systematically organized into a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ and a diagonal matrix $\mathbf{Y}$ where the diagonal elements are given by $\mathbf{Y}_{ii} = y_i$.

3.1. Memory kernel function

Soft-interval SVM makes two classes of sample points with maximum intervals between them by constructing two parallel hyperplanes. However, it is difficult to achieve zero empirical risk for SVM models using a linear kernel. To solve this problem, Vapnik and Izmailov achieved the correct

ISSN 2616-5805 Vol. 5, Issue 3: 67-74, DOI: 10.25236/AJMS.2024.050310

classification of all training samples by introducing a memory kernel function in the dyadic problem of SVM as follows

$$K_{mg}(\boldsymbol{x},\boldsymbol{x}_i)=(1-\tau)e^{-\sigma^2(\boldsymbol{x}-\boldsymbol{x}_i)^2}+\tau e^{-\sigma_*^2(\boldsymbol{x}-\boldsymbol{x}_i)^2}, \tag{1}$$

where $\sigma_* \gg \sigma > 0$, $0 \leqslant \tau \leqslant 1$ denotes the weights and the τ parameter is used to balance the weights between generalization and memory. Specifically, the generalization memory kernel is considered to use a weighted combination containing both the generalization RBF kernel and the memory RBF kernel. However, the original problem of SVM with the introduction of the memory kernel function is not analyzed, and sufficient explanatory properties are lacking.

3.2. Generalization-memorization machines

Wang et al.[7] further proposed a GMM based on the principle of large margins, which can easily obtain zero empirical risk. Specifically, the decision function of HGMM is expressed as follows

$$f(\boldsymbol{x})=\operatorname{sgn}\left(\langle \boldsymbol{w},\varphi(\boldsymbol{x})\rangle+b+\sum_{i=1}^{m}y_i c_i \delta(\boldsymbol{x}_i,\boldsymbol{x})\right), \tag{2}$$

where , $\boldsymbol{w}\in\mathbb{R}^n$, $b\in\mathbb{R}$, $\operatorname{sgn}(\cdot)$ denotes the sign function. Given a new sample point $\hat{x}$, if $f(\hat{x})\geqslant 0$, it is classified as positive and if $f(\hat{x})<0$, it is classified as negative. c_i represents the memory cost and $\delta(\boldsymbol{x}_i,\boldsymbol{x})$ represents the memory influence function, which is mainly used to measure the influence of the memory sample on the surrounding points. The study assumes that samples that are very similar to $\boldsymbol{x}_i$ and $\boldsymbol{x}$ belong to the same category, i.e., the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}$ is measured. In general, the memory influence function may vary. e.g.

$$\delta(\boldsymbol{x}_i,\boldsymbol{x}_j)=e^{-\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{2\sigma^2}},\quad \sigma>0, \tag{3}$$

$$\delta(\boldsymbol{x}_i,\boldsymbol{x}_j)=\max\{\rho-\|\boldsymbol{x}_i-\boldsymbol{x}_j\|,0\},\quad \rho>0, \tag{4}$$

$$\delta(\boldsymbol{x}_i,\boldsymbol{x}_j)=\begin{cases}\|\boldsymbol{x}_i-\boldsymbol{x}_j\|, & \text{if } \|\boldsymbol{x}_i-\boldsymbol{x}_j\|\leqslant\varepsilon,\ \varepsilon>0,\\ 0, & \text{else.}\end{cases} \tag{5}$$

To explore the original problem of generalized memory models and the implications for generalization, HGMM constructed a model of categorization under the theory of memory, i.e.

$$\begin{aligned}&\min_{\boldsymbol{w},b,\boldsymbol{c}}\quad \frac{1}{2}\|\boldsymbol{w}\|^2+\frac{\lambda}{2}\|\boldsymbol{c}\|^2\\ &\text{s.t.}\quad y_i\left(\langle \boldsymbol{w},\varphi(\boldsymbol{x}_i)\rangle+b+\sum_{j=1}^{m}y_j c_j\delta(\boldsymbol{x}_i,\boldsymbol{x}_j)\right)\geqslant 1,\quad i=1,...,m,\end{aligned} \tag{6}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product, $\varphi(\cdot)$ is the mapping, λ is the positive parameter, and $\mathbf{c}=(c_1,c_2,...,c_m)^{\top}$ is the memory cost of the training sample. Equation 6 should have as little memory cost as possible for the training samples while maximizing the interval between the two classes. This equation 6 is a quadratic programming problem with the below dual

$$\begin{aligned}&\min_{\boldsymbol{\alpha}}\quad \frac{1}{2}\boldsymbol{\alpha}^{\top}\mathbf{Y}\left(\mathbf{K}(\mathbf{X},\mathbf{X})+\frac{1}{\lambda}\Delta\Delta^{\top}\right)\mathbf{Y}\boldsymbol{\alpha}-1^{\top}\boldsymbol{\alpha}\\ &\text{s.t.}\quad 1^{\top}\mathbf{Y}\boldsymbol{\alpha}=0,0\leqslant\boldsymbol{\alpha},\end{aligned} \tag{7}$$

where $\boldsymbol{\Delta}\in\mathbb{R}^{m\times m}=\delta(\boldsymbol{x}_i,\boldsymbol{x}_j),\ i,j=1,2,...,m$, $\alpha\in\mathbb{R}^m$ is a vector of Lagrange multipliers and $\mathbf{K}(\cdot,\cdot)=\langle\varphi(\cdot),\varphi(\cdot)\rangle$ is a kernel matrix. In summary, both HGMM and SGMM Both can achieve the correct classification of all training data, but HGMM gives the original problem and is more interpretable. Moreover, the memory mechanism of HGMM is also applicable to other linear classifiers. However，

ISSN 2616-5805 Vol. 5, Issue 3: 67-74, DOI: 10.25236/AJMS.2024.050310

when dealing with data containing noise, SGMM is more adaptable than HGMM. Finally, although the two models improve the robust performance of SVM, they do not improve the ability of SVM on handling large-scale data and do not involve the memory mechanism in the regression problem.

4. Least Squares Generalization-Memorization Machines

In this section, LSGMM is introduced, which combines the HGMM model with the least squares term and introduces a new memory term in the objective function to transform the inequality constraints in the optimization Equation 6 into equality constraints. LSGMM searches for the optimal classification hyperplane by minimizing the memory cost and the weighted memory term, and helps reduce the computational complexity of the HGMM model. The optimization problem for LSGMM is shown below:

$$\begin{aligned} &\min_{\boldsymbol{w},b,\boldsymbol{c}} \quad \frac{1}{2}\|\boldsymbol{w}\|^2+\frac{\gamma}{2}\|\boldsymbol{c}\|^2+\lambda\sum_{i=1}^{m}c_i\delta(\boldsymbol{x}_i,\boldsymbol{x}_k) \\ &\text{s.t.} \quad y_i(\langle \boldsymbol{w},\varphi(\boldsymbol{x}_i)\rangle+b+y_ic_i\delta(\boldsymbol{x}_i,\boldsymbol{x}_k))=1,\ i=1,...,m, \end{aligned} \tag{8}$$

where $\boldsymbol{x}_k$ denotes the sample centroid corresponding to the label of the training sample $\boldsymbol{x}_i$.In order to achieve zero empirical risk for LSGMM, we construct a new memory influence function with the following form:

$$\delta(\boldsymbol{x}_i,\boldsymbol{x}_j)=\begin{cases}\dfrac{\rho}{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|}, & \text{if } x_i\neq x_j,\ \rho>0,\\ 1, & \text{else.}\end{cases} \tag{9}$$

In LSGMM, we memorize the training samples by finding the training samples closest to the test sample points. The Lagrangian function corresponding to Equation 8 is

$$\begin{aligned} L(\boldsymbol{w},b,\boldsymbol{c},\boldsymbol{\alpha})=&\frac{1}{2}\|\boldsymbol{w}\|^2+\frac{\gamma}{2}\|\boldsymbol{c}\|^2+\lambda\sum_{j=1}^{m}c_i\delta(\boldsymbol{x}_i,\boldsymbol{x}_k)+\sum_{i=1}^{m}\alpha_i \\ &(1-y_i(\langle \boldsymbol{w},\varphi(\boldsymbol{x}_i)\rangle+b+y_ic_i\delta(\boldsymbol{x}_i,\boldsymbol{x}_k))). \end{aligned} \tag{10}$$

Based on the fact that the partial derivative of $L(\boldsymbol{w},b,\boldsymbol{c},\boldsymbol{\alpha})$ with respect to $\boldsymbol{w},b,c_i,\alpha_i$ is zero, it follows

$$\begin{cases}\dfrac{\partial L}{\partial \boldsymbol{w}}=\boldsymbol{w}-\sum_{i=1}^{m}\alpha_iy_i\varphi(\boldsymbol{x}_i)=0,\\ \dfrac{\partial L}{\partial b}=\sum_{i=1}^{m}\alpha_iy_i=0,\\ \dfrac{\partial L}{\partial c_i}=\gamma\xi_i+\lambda\delta(\boldsymbol{x}_i,\boldsymbol{x}_j)-\alpha_i\delta(\boldsymbol{x}_i,\boldsymbol{x}_j)=0,\\ \dfrac{\partial L}{\partial \alpha_i}=1-y_i(\langle \boldsymbol{w},\varphi(\boldsymbol{x}_i)\rangle+b+y_ic_i\delta(\boldsymbol{x}_i,\boldsymbol{x}_j))=0.\end{cases} \tag{11}$$

The above collation gives

$$\begin{pmatrix}\mathbf{Y}\left(\mathbf{K}(\mathbf{X},\mathbf{X})+\dfrac{1}{\gamma}\mathbf{D}\mathbf{D}^{\top}\right)\mathbf{Y} & \mathbf{Y}\mathbf{1}\\ 1^{\top}\mathbf{Y} & 0\end{pmatrix}\begin{pmatrix}\boldsymbol{\alpha}\\ \mathrm{b}\end{pmatrix}=\begin{pmatrix}1+\dfrac{\lambda}{\gamma}\mathbf{D}\mathbf{D}^{\top}1\\ 0\end{pmatrix} \tag{12}$$

where $\mathbf{D}\in\mathbb{R}^{m\times m}$ is a diagonal matrix and $\mathbf{D}_{ii}=\delta(\boldsymbol{x}_i,\boldsymbol{x}_k)\,(i=1,2,...,m)$. Finally, after solving Equation 12, the corresponding decision function is obtained by solving

$$f(\boldsymbol{x})=\operatorname{sgn}(\langle \boldsymbol{w},\varphi(\boldsymbol{x})\rangle+b+y_ic_i\delta(\boldsymbol{x},\boldsymbol{x}_k)), \tag{13}$$

where y_i is the label of the training sample $\boldsymbol{x}_i$ with the closest Euclidean distance to the new

ISSN 2616-5805 Vol. 5, Issue 3: 67-74, DOI: 10.25236/AJMS.2024.050310

sample $\boldsymbol{x}$, and $\boldsymbol{x}_k$ denotes the class center of the class in which $\boldsymbol{x}_i$ is located. LSGMM determines the labels of the newly arrived sample points based on the positivity and negativity of Equation 13.

5. Experiment

This section uses several calibration datasets from UCI, and Table 1 provides detailed information. We analyze the performance of the LSGMM model on various benchmark datasets, as well as its execution time on large datasets. The classical LSSVM model uses a linear kernel, while the SVM^m and HGMM models use a linear generalization kernel and an RBF memory kernel. In contrast, our LSGMM model uses a linear kernel. All these models were implemented using MATLAB 2017a on a PC equipped with an Intel Core Duo processor (dual 4.2 GHz) and 32 GB RAM. For the RBF kernel memory kernel parameters, we tested the weights from the set $\{2^{-8}, 2^{-7}, ..., 2^{8}\}$. The other models use the same set of weighting parameters. Our comparison is conducted by evaluating the memory performance of the linear kernel in the LSGMM models on a number of small datasets and benchmarking the linear kernel in the LSSVM.

Table 1: Details of benchmark datasets.

ID	Name	m	n
(a)	Cleveland	173	13
(b)	Ionosphere	351	34
(c)	New-thyroid	215	4
(d)	Parkinsons	195	22
(e)	Sonar	208	60
(f)	TicTacToe	958	27
(g)	Vowel	988	13
(h)	Wisconsin	683	9
(i)	German	1000	20
(j)	Shuttle	1829	9
(k)	Segment	2308	19
(l)	Waveform	5000	21
(m)	TwoNorm	7400	20
(n)	IJCNN01	49990	22

5.1. Memory Capacity

In order to assess the ability of the LSGMM model to achieve zero empirical risk, Tables 2 and 3 show the maximum training and testing accuracies achieved by the LSGMM model, respectively. The experimental results from Table 3 reveals that the LSGMM model performs best in terms of testing accuracy when using Equation 9. The reason why LSGMM does not achieve 100% training accuracy is due to the irreversibility of D in these functions. Different memory selection influence functions lead to different training effects. Among the various influence functions, Equation 9 produces the highest test accuracy for most datasets. Therefore, in subsequent experiments, we adopt Equation 9 as the basis of our LSGMM model.

Table 2: Training results of LSGMM and LSSVM based on generalized memory kernel.

ID	LSSVM	LSGMM^1	LSGMM^2	LSGMM^3	LSGMM^4
(a)	96.39±0.47	96.40±0.79	94.83±3.10	98.25±0.90	100.00±0.00
(b)	89.46±0.47	100.00±0.00	91.45±2.01	100.00±0.00	100.00±0.00
(c)	94.08±0.86	100.00±0.00	99.09±2.03	100.00±0.00	100.00±0.00
(d)	91.42±1.15	100.00±0.00	97.64±2.25	100.00±0.00	100.00±0.00
(e)	87.99±1.36	100.00±0.00	86.85±2.73	100.00±0.00	100.00±0.00
(f)	98.33±0.25	100.00±0.00	98.33±1.44	100.00±0.00	100.00±0.00
(g)	95.04±0.34	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
(h)	96.16±0.61	100.00±0.00	97.23±1.29	100.00±0.00	100.00±0.00

^{m},^{1},^{2},^{3} ,^{4}with the Equation 1,3, 4,5 and 9.

ISSN 2616-5805 Vol. 5, Issue 3: 67-74, DOI: 10.25236/AJMS.2024.050310

Table 3: Test results of LSGMM and LSSVM based on generalized memory kernel.

ID	LSSVM	LSGMM[1]	LSGMM[2]	LSGMM[3]	LSGMM[4]
(a)	94.82±4.18	95.28±3.40	94.72±4.81	95.33±1.71	95.36±2.60
(b)	88.30±3.46	90.08±4.29	94.61±3.64	88.61±1.67	89.75±4.63
(c)	93.66±3.41	98.64±1.18	98.57±1.28	98.72±1.90	98.73±1.90
(d)	88.40±7.44	97.49±1.49	97.07±1.93	96.42±3.03	96.30±1.62
(e)	79.48±2.85	86.11±5.32	87.46±3.01	86.94±2.20	88.47±2.63
(f)	98.33±1.00	98.33±1.70	98.33±0.93	98.33±0.93	98.33±1.13
(g)	95.04±2.08	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
(h)	96.18±2.45	97.36±1.33	97.37±0.82	97.08±1.00	96.94±1.65

[m],[1],[2],[3],[4]with the Equation 1,3, 4,5 and 9.

5.2. Time performance

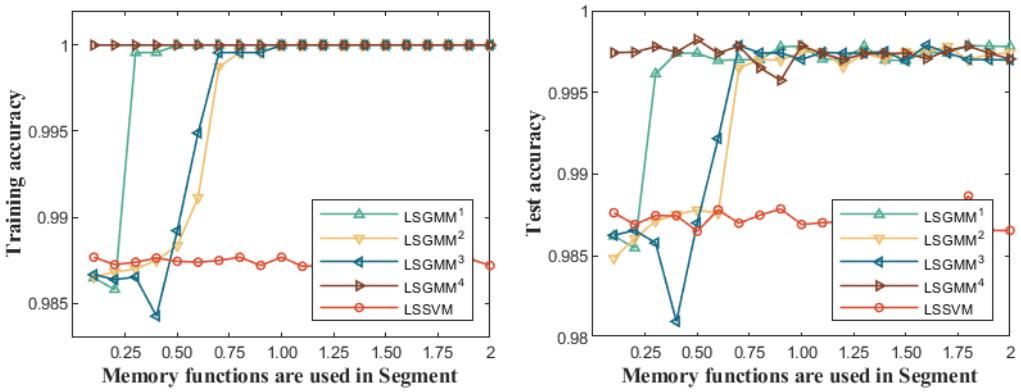
To address the time cost of LSGMM on large-scale data, this experiment records the running time of LSGMM, SVM[m] and HGMM under the optimal parameters on six datasets with sample sizes larger than 1000. The training accuracy and testing accuracy under the corresponding parameters are also recorded. For each dataset, about 70% of the samples are randomly selected for training, and the remaining 30% of the samples constitute the test set. From Table 4, it is clear that LSGMM took less time to ensure 100% training accuracy and obtained test accuracies with little difference compared to the other two memory models. On dataset (n), HGMM and SVM[m] limit the model solving due to lack of memory, while LSGMMM can run normally, thus demonstrating the superiority of the model.

Table 4: Accuracy and time to train and test linear classifiers on benchmark datasets.

ID	SVM[m]	HGMM	LSGMM[4]	SVM	HGMM	LSGMM[4]
	Training results			Test results		
(i)	100.00±0.00	100.00±0.00	100.00±0.00	76.10±1.77	78.33±3.53	75.06±2.55
				0.198s	0.196s	**0.105s**
(j)	100.00±0.00	100.00±0.00	100.00±0.00	99.95±0.12	100.00±0.00	100.00±0.00
				0.738s	0.663s	**0.291s**
(h)	100.00±0.00	100.00±0.00	100.00±0.00	99.83±0.18	99.88±0.19	99.77±0.21
				0.997s	1.256s	**0.393s**
(l)	100.00±0.00	100.00±0.00	100.00±0.00	90.16±0.81	88.27±0.61	83.24±0.67
				5.835s	6.682s	**1.656s**
(m)	100.00±0.00	100.00±0.00	100.00±0.00	98.02±0.25	97.98±0.26	95.09±0.56
				15.793s	18.531s	**3.895s**
(n)	100.00±0.00	100.00±0.00	100.00±0.00	*	*	97.37±0.08
						435.870s

**Indicates insufficient memory to run.*

5.3. Memory Parameter Effects



ISSN 2616-5805 Vol. 5, Issue 3: 67-74, DOI: 10.25236/AJMS.2024.050310

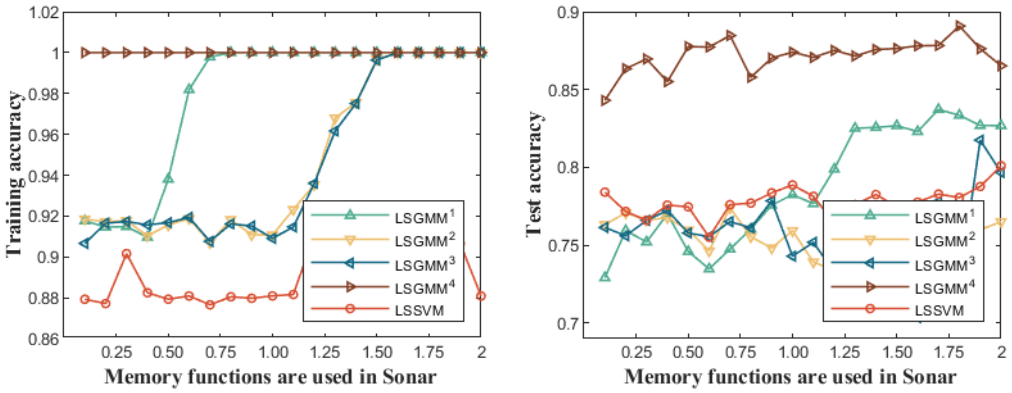


Figure 1: Training (left)/testing (right) accuracy with different influence functions.

In order to analyze the effect of different parameters of different memory kernels on the model, the parameter range of the memory kernel was reset to $\{0, 0.1, ..., 2\}$ in this experiment, and the training and testing sets were set unchanged. Figure 1 uses the LSSVM model as a benchmark and shows the effects of different parameters of different memory kernels on the training and testing accuracies, respectively. It is easy to find from this figure that for the training data, the different choices of memory kernel parameters will directly affect whether the model can achieve 100% training accuracy, i.e., zero empirical risk. Secondly, the use of Equation 9 in the LSGMM model can achieve stable testing and training results. Finally, after choosing appropriate memory kernel parameters, LSGMM can guarantee zero empirical risk while testing accuracy is higher than LSSVM. In conclusion, with these experimental results, demonstrating the degree of influence of memory kernel on the model under different parameters, we validate the performance advantage of our memory mechanism over LSSVM.

6. Conclusion

In this paper, we present two novel innovations on the traditional LSSVM framework. Our contributions include proposing a substitution of the LSSVM objective function, which improves the performance; and introducing a new memory generalization kernel that effectively integrates the complete memory of the training data, achieving zero training error. As a result of these innovations, LSGMM models exhibit superior generalization accuracy while maintaining the same computational complexity. Specifically, they still involve solving systems of linear equations with corresponding dimensions, as in the current LSSVM implementation. In addition, they require less time and higher costs to memorize training samples than existing memory models.

References

[1] Rosenfeld E, Ravikumar P, Risteski A. The risks of invariant risk minimization[J]. arXiv preprint arXiv: 2010.05761, 2020.
[2] Li T, Beirami A, Sanjabi M, et al. Tilted empirical risk minimization[J]. arXiv preprint arXiv: 2007.01162, 2020.
[3] Rice L, Wong E, Kolter Z. Overfitting in adversarially robust deep learning[C]//International Conference on Machine Learning, 2020. PMLR.
[4] Cortes C, Vapnik V. Support-vector networks[J]. Machine Learning, 1995, 20: 273-297.
[5] Vapnik V, Izmailov R. Reinforced SVM method and memorization mechanisms[J]. Pattern Recognition, 2021, 119: 108018.
[6] Ding X, Liu J, Yang F, et al. Random radial basis function kernel-based support vector machine[J]. Journal of the Franklin Institute, 2021, 358(18): 10121-10140.
[7] Wang Z, Shao Y. Generalization-Memorization Machines[J]. arXiv preprint arXiv: 2207.03976, 2022.
[8] Li Z, Hoiem D. Learning without forgetting[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 40(12): 2935-2947.
[9] Li Z, Liu F, Yang W, et al. A survey of convolutional neural networks: analysis, applications, and prospects[J]. IEEE transactions on neural networks and learning systems, 2021, 33(12): 6999-7019.

ISSN 2616-5805 Vol. 5, Issue 3: 67-74, DOI: 10.25236/AJMS.2024.050310

[10] Kim D, Bae J, Jo Y, et al. Incremental learning with maximum entropy regularization: Rethinking forgetting and intransigence[J]. arXiv preprint arXiv: 1902.00829, 2019
[11] Lopez-Paz D, Ranzato M. Gradient episodic memory for continual learning[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems, NY, USA, 2017.
[12] Arpit D, Jastrzębski S, Ballas N, et al. A closer look at memorization in deep networks[C]// International Conference on Machine Learning, 2017: 233-242.
[13] Feldman V. Does learning require memorization? A short tale about a long tail[C]//Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, 2020.
[14] Suykens J A K, Vandewalle J. Least squares support vector machine classifiers[J]. Neural processing letters, 1999, 9: 293-300.