Neurocomputing

Learning Nonseparable Sparse Regularizers via Multivariate Activation Functions

--Manuscript Draft--

Manuscript Number:	NEUCOM-D-24-00007
Article Type:	Regular article
Section/Category:	Neural Networks
Keywords:	Sparse regularization; multivariate activation function; deep learning framework
Corresponding Author:	Xin Xu Peking University CHINA
First Author:	Xin Xu
Order of Authors:	Xin Xu
	Zhouchen Lin
Abstract:	Sparse regularization is a widely embraced technique in high-dimensional machine learning and signal processing. Existing sparse regularizers, however, are predominantly hand-crafted and often separable, making them less adaptable to data and potentially hindering performance. In this paper, we present a novel approach aiming at learning nonseparable (multivariate) sparse regularizers. We leverage the proximal gradient algorithm to transform the challenge of acquiring nonseparable sparse regularizers into the task of learning multivariate activation functions. We further establish the necessary conditions that these activation functions should satisfy. Our contribution culminates in the introduction of MAF-SRL, a deep network designed to learn multivariate activation functions within existing deep learning frameworks. To our knowledge, this research marks the first endeavor to learn nonseparable sparse regularizers. Extensive experiments conducted on benchmark datasets underscore the superiority of regularizers learned through MAF-SRL. They exhibit significantly enhanced performance in terms of both accuracy and sparseness compared to existingsparse regularizers.

Learning Nonseparable Sparse Regularizers via Multivariate Activation Functions

Xin Xu[a], Zhouchen Lin[a,b,c,*]

[a] *National Key Lab of General AI, School of Intelligence Science and Technology, Peking University, China*
[b] *Institute for Artificial Intelligence, Peking University, China*
[c] *Peng Cheng Laboratory, China*

Abstract

Sparse regularization is a widely embraced technique in high-dimensional machine learning and signal processing. Existing sparse regularizers, however, are predominantly hand-crafted and often separable, making them less adaptable to data and potentially hindering performance. In this paper, we present a novel approach aiming at learning nonseparable (multivariate) sparse regularizers. We leverage the proximal gradient algorithm to transform the challenge of acquiring nonseparable sparse regularizers into the task of learning multivariate activation functions. We further establish the necessary conditions that these activation functions should satisfy. Our contribution culminates in the introduction of MAF-SRL, a deep network designed to learn multivariate activation functions within existing deep learning frameworks. To our knowledge, this research marks the first endeavor to learn nonseparable sparse regularizers. Extensive experiments conducted on benchmark datasets underscore the superiority of regularizers learned through MAF-SRL. They exhibit significantly enhanced performance in terms of both accuracy and sparseness compared to existing sparse regularizers.

*Corresponding author
Email addresses: xux20@stu.pku.edu.cn (Xin Xu), zlin@pku.edu.cn (Zhouchen Lin)

Learning Nonseparable Sparse Regularizers via Multivariate Activation Functions

Xin Xu[a], Zhouchen Lin[a,b,c,*]

[a] *National Key Lab of General AI, School of Intelligence Science and Technology, Peking University, China*
[b] *Institute for Artificial Intelligence, Peking University, China*
[c] *Peng Cheng Laboratory, China*

Abstract

Sparse regularization is a widely embraced technique in high-dimensional machine learning and signal processing. Existing sparse regularizers, however, are predominantly hand-crafted and often separable, making them less adaptable to data and potentially hindering performance. In this paper, we present a novel approach aiming at learning nonseparable (multivariate) sparse regularizers. We leverage the proximal gradient algorithm to transform the challenge of acquiring nonseparable sparse regularizers into the task of learning multivariate activation functions. We further establish the necessary conditions that these activation functions should satisfy. Our contribution culminates in the introduction of MAF-SRL, a deep network designed to learn multivariate activation functions within existing deep learning frameworks. To our knowledge, this research marks the first endeavor to learn nonseparable sparse regularizers. Extensive experiments conducted on benchmark datasets underscore the superiority of regularizers learned through MAF-SRL. They exhibit significantly enhanced performance in terms of both accuracy and sparseness compared to existing sparse regularizers.

Keywords: Sparse regularization, multivariate activation function, deep learning framework

*Corresponding author
Email addresses: xux20@stu.pku.edu.cn (Xin Xu), zlin@pku.edu.cn (Zhouchen Lin)

1. Introduction

Sparse regularization is a powerful and widely adopted strategy for tackling challenges in high-dimensional machine learning and signal processing problems. Its effectiveness is well-established through practical applications and rigorous theoretical investigations, as exemplified by the success of techniques like LASSO (Fonti & Belitser, 2017; Kim & Paik, 2019; Celentano et al., 2023).

One of the remarkable strengths of sparse regularization lies in its dual functionality—it simultaneously performs parameter estimation and feature selection. This unique characteristic produces results that are not only informative but also highly interpretable, as it identifies critical variables. Moreover, it effectively mitigates overfitting by eliminating redundant features. These attributes have propelled sparse regularization to remarkable achievements across diverse domains, spanning machine learning and signal processing. Additionally, extensive theoretical research has bolstered its efficacy, complemented by the development of efficient optimization methods, simplifying its practical implementation.

Despite its widespread adoption, a plethora of sparse regularizers have been introduced to facilitate the generation of sparse solutions. The ℓ_0 (pseudo-)norm, which quantifies the number of non-zero elements, serves as the most intuitive form of sparse regularization, with the primary aim of promoting solution sparsity. Unfortunately, problems involving ℓ_0 norm regularization are typically classified as NP-hard (Natarajan, 1995; Hillar & Lim, 2013; Atserias & Müller, 2020; Hirahara, 2022), posing significant computational challenges. Consequently, the ℓ_1 norm has emerged as the predominant surrogate for the ℓ_0 norm [(Candes et al., 2008; Tsagkarakis et al., 2018; Li et al., 2022b). This convex alternative substantially simplifies the optimization process, although it is essential to recognize that ℓ_1 regularization, while advantageous, may not consistently yield sufficiently sparse solutions and can introduce notable estimation bias (Fan & Li, 2001; Issa & Gastpar, 2018; Varno et al., 2022).

To overcome these limitations, a multitude of alternative sparse regularizers have been proposed and systematically analyzed. These include the smoothly

clipped absolute deviation (SCAD) (Fan & Li, 2001; Li et al., 2020; Kadhim, 2023), log penalty (Candes et al., 2008; Zhang et al., 2020; Prater-Bennette et al., 2022), capped ℓ_1 (Zhang, 2010; Chen et al., 2019; Sriramanan et al., 2022), minimax concave penalty (MCP) (Zhang, 2010; Jiang et al., 2019; Liao et al., 2023), ℓ_p penalty with p in the range of $(0, 1)$ (Zhang et al., 2014; Bore et al., 2019; Li et al., 2022a), and the difference between ℓ_1 and ℓ_2 norms (Lou et al., 2015; Wu et al., 2018; Moayeri et al., 2022). It is noteworthy that a majority of these regularizers operate in a separable manner, potentially limiting their ability to capture interactions among vector entries and affecting their performance.

In a related context, it is worth mentioning that, to the best of our knowledge, existing sparse regularizers are primarily manually designed. This inherent characteristic raises concerns about their seamless alignment with underlying models to effectively promote sparsity or their suitability for data characteristics to achieve optimal performance. Consequently, practical approaches often involve experimenting with multiple existing sparse regularizers and selecting the most effective one, a process that can be cumbersome in practice. The only learning based sparse regularizer was proposed by Wang et al. (Wang et al., 2021; Ohn & Kim, 2022). However, the learnt sparse regularizer is separable, hence may not fully exploit the interaction among the entries of the vector to be regularized, preventing it from achieving even better performance.

To address these issues, this paper focuses on learning nonseparable sparse regularizers. Our main contributions can be summarized as follows:

- Leveraging the proximal gradient algorithm, we establish a bridge between nonseparable multivariate regularizers and multivariate activation functions. Notably, a substantial portion of existing sparse regularizers is separable. To our knowledge, this work is the first to tackle the challenge of learning nonseparable (multivariate) sparse regularizers.

- We derive conditions that multivariate activation functions must satisfy to qualify as proper nonseparable sparse regularizers, offering a principled framework for effective regularization.

- We introduce MAF-SRL, a novel deep network that learns multivariate activation functions. This approach allows us to implicitly obtain the desired nonseparable sparse regularizers, seamlessly integrating them into various machine learning tasks.

Extensive experiments showcase that the nonseparable sparse regularizers learned by MAF-SRL significantly outperform all existing representative sparse regularizers in terms of both classification accuracy and sparsity.

2. Related Works

Sparse regularization has gained significant attention in various research fields due to its ability to promote sparsity in estimation. One of the most commonly used sparse regularizers is the ℓ_1 norm (Candes et al., 2008; Tsagkarakis et al., 2018; Wang et al., 2022). However, it has been observed that estimation with the ℓ_1 norm can be biased (Fan & Li, 2001; Issa & Gastpar, 2018; Wang et al., 2022) and may not always result in a sufficiently sparse solution. As a result, researchers have been motivated to design more general sparse regularizers.

In this regard, previous work (Fan & Li, 2001; Li et al., 2020) has proposed that an ideal regularizer should possess three desired properties: unbiasedness, sparsity, and continuity. The smoothly clipped absolute deviation (SCAD) (Fan & Li, 2001; Li et al., 2020; Kadhim, 2023) was introduced as the first regularizer to satisfy these properties. For a vector variable $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n$, SCAD is defined as $\mathcal{L}(\mathbf{x}; \lambda, \gamma) = \sum_{i=1}^{n} \ell(x_i; \lambda, \gamma)$, where

$$\ell(x_i; \lambda, \gamma) = \begin{cases} \lambda |x_i|, & \text{if } |x_i| \leq \lambda, \\ \frac{2\gamma\lambda|x_i| - x_i^2 - \lambda^2}{2(\gamma-1)}, & \text{if } \lambda < |x_i| < \gamma\lambda, \\ \lambda^2(\gamma+1)/2, & \text{if } |x_i| \geq \gamma\lambda, \end{cases}$$

where $\lambda > 0$ and $\gamma > 2$. SCAD is a two-parameter function composed of three pieces. Subsequently, researchers proposed another regularizer called minimax concave penalty (MCP) in (Zhang, 2010; Jiang et al., 2019; Liao et al., 2023),

which has two pieces. MCP is formulated as $\mathcal{L}_\gamma(\mathbf{x};\lambda) = \sum_{i=1}^{n} \ell_\gamma\left(x_i;\lambda\right)$, with

$$\ell_\gamma\left(x_i;\lambda\right) = \begin{cases} \lambda\left|x_i\right| - x_i^2/(2\gamma), & \text{if } \left|x_i\right| \leq \gamma\lambda, \\ \gamma\lambda^2/2, & \text{if } \left|x_i\right| > \gamma\lambda, \end{cases}$$

where parameter $\gamma > 1$. Additionally, the log penalty (Candes et al., 2008; Zhang et al., 2020; Prater-Bennette et al., 2022) was introduced as a generalization of the elastic net family, defined as $\mathcal{L}(\mathbf{x};\gamma) = \sum_{i=1}^{n} \ell\left(x_i,\gamma\right)$, where

$$\ell\left(x_i;\gamma\right) = \frac{\log\left(\gamma\left|x_i\right| + 1\right)}{\log(\gamma+1)},$$

and $\gamma > 0$. The log penalty allows for obtaining the entire continuum of penalties from $\ell_1\left(\gamma \to 0_+\right)$ to $\ell_0(\gamma \to \infty)$ (Mazumder et al., 2011; Xu et al., 2017; Pardo-Simon, 2023). Another approximation of the ℓ_0 norm is the capped ℓ_1 (Zhang, 2008; Chen et al., 2019; Sriramanan et al., 2022), defined as

$$\mathcal{L}(\mathbf{x};a) = \sum_{i=1}^{n} \min\left(\left|x_i\right|, a\right),$$

where $a > 0$. Notably, when $a \to 0$, $\sum_i \min\left(\left|x_i\right|, a\right)/a \to \|\mathbf{x}\|_0$. Furthermore, some concise forms of other norms, such as ℓ_p with $p \in (0,1)$ (Xu et al., 2012; Sharif et al., 2018; Li et al., 2022a), have been considered as alternatives to improve ℓ_1. The ℓ_p norm is expressed as

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} \left|x_i\right|^p\right)^{1/p}.$$

Additionally, sparse regularizers can be combined to form new regularizers, such as the ℓ_{1-2} penalty (Yin et al., 2015; Ming et al., 2019; Chen et al., 2021; Liu & Yu, 2023), which is the difference between the ℓ_1 and ℓ_2 norms, and the combined group and exclusive sparsity (CGES) (Yoon & Hwang, 2017; Bui et al., 2021; Tang et al., 2023).

While all the above sparse regularizers are handcrafted, (Wang et al., 2021) first proposed a strategy to learn sparse regularizers. They utilized the relationship between regularizers and activation functions via the proximal operator. Then learning the regularizers can be converted to learning the activation functions. This paper is a significant extension of (Wang et al., 2021) although inherits some ingredients from (Wang et al., 2021).

It is worth noting that except for the ℓ_p norms where $p \neq 0, 1$, all existing sparse regularizers, including those learnt (Wang et al., 2021), are separable. This implies that they are composed of sums of functions of individual entries of a given vector. While separable regularizers have been widely used, their inability to fully exploit interactions among the vector entries may limit their effectiveness in achieving better performance. Therefore, in this paper, we aim to learn non-separable sparse regularizers.

3. The Proposed MAF-SRL Approach

3.1. Connection between Sparse Regularizer and Activation Function

When solving a learning model of the form

$$\min_{\mathbf{x}} \phi(\mathbf{x}), \tag{1}$$

it is often necessary to add a regularizer $g(\mathbf{x})$ to the objective function and solve the regularized problem

$$\min_{\mathbf{x}} [\phi(\mathbf{x}) + g(\mathbf{x})] \tag{2}$$

instead. This is done to address challenges in solving (1), such as non-unique solutions or to incorporate prior information about the desired solution, such as sparsity. By adding an appropriate regularizer, the original problem becomes well-posed, allowing us to obtain solutions with desired properties.

When ϕ is L-smooth, meaning that it satisfies the following condition,

$$\|\nabla\phi(\mathbf{x}) - \nabla\phi(\mathbf{y})\|_F \leq L\|\mathbf{x} - \mathbf{y}\|_F, \tag{3}$$

where L is called the Lipschitz constant in the sequel, a common algorithm for solving problem (2) is the proximal gradient method (Lu et al., 2015). When applied to (2), the iterations of the proximal gradient method are as follows:

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \arg\min_{\mathbf{x}} \phi\left(\mathbf{x}^{(k)}\right) + \left\langle \nabla\phi\left(\mathbf{x}^{(k)}\right), \mathbf{x} - \mathbf{x}^{(k)} \right\rangle \\ &\quad + \frac{L}{2}\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\|_F^2 + g(\mathbf{x}) \\ &= \arg\min_{\mathbf{x}} \frac{L}{2}\left\|\mathbf{x} - \mathbf{x}^{(k)} + \frac{1}{L}\nabla\phi\left(\mathbf{x}^{(k)}\right)\right\|_F^2 + g(\mathbf{x}). \end{aligned} \tag{4}$$

Let $\mathbf{r}^{(k)} = \mathbf{x}^{(k)} - \frac{1}{L}\nabla\phi\left(\mathbf{x}^{(k)}\right)$, then solving (4) requires solving the following optimization problem:

$$\mathrm{Prox}_{\alpha g}(\mathbf{r}^{(k)}) = \arg\min_{\mathbf{x}} \left[\frac{1}{2}\|\mathbf{x} - \mathbf{r}^{(k)}\|_F^2 + \alpha g(\mathbf{x})\right], \tag{5}$$

where $\mathrm{Prox}_{\alpha g}(\cdot)$ is the proximal operator associated with the function $g(\cdot)$ and $\alpha > 0$ is a parameter. Therefore, the solution to (2) can be obtained through the following iteration:

$$\mathbf{x}^{(k+1)} = \mathrm{Prox}_{L^{-1}g}\left(\mathbf{x}^{(k)} - \frac{1}{L}\nabla\phi\left(\mathbf{x}^{(k)}\right)\right). \tag{6}$$

It is worth noting that proximal operators are monotone (Lu et al., 2015), regardless of the convexity of g. This property allows them to serve as activation functions in deep neural networks (DNNs). Conversely, *some* activation functions can be viewed as proximal operators of regularizers, although this inverse correspondence has only been explored in the univariate case (Li et al., 2019; Bibi et al., 2019; Combettes & Pesquet, 2020). This limitation may arise from the fact that, up to now, only univariate activation functions have been widely used.

In the case of a non-decreasing univariate activation function $\xi(x) : \mathbb{R} \to \mathbb{R}$, we can derive the corresponding univariate regularizer as follows (Li et al., 2019):

$$g(x) = \int_0^x \left(\xi^{-1}(y) - y\right) dy = \int_0^x \xi^{-1}(y)dy - \frac{1}{2}x^2, \tag{7}$$

where $\xi^{-1}(y)$ represents the inverse function of $\xi(y)$. This relationship between univariate regularizers and activation functions is well-known (Li et al., 2019; Bibi et al., 2019; Combettes & Pesquet, 2020; Wang et al., 2021). However, Equation (7) only gives the expression for univariate regularizers. In the following, we extend this deduction from the univariate case to the multivariate case.

It is worth noting that any multivariate regularizer can be approximated using the following form (Chen & Chen, 1995):

$$\sum_{i=1}^{M} q_i g\left(\mathbf{a}_i^T\mathbf{x} + b_i\right), \tag{8}$$

where M, g, $\boldsymbol{q}$, $\mathbf{A}$, and $\mathbf{b}$ are appropriately chosen parameters. Here, $\mathbf{A} = (\mathbf{a}_1, \cdots, \mathbf{a}_M)$, $\boldsymbol{q} = (q_1, \cdots, q_M)^T$, and $\mathbf{b} = (b_1, \cdots, b_M)^T$. Equation (8) can be seen as a neural network with only one hidden layer.

To simplify the computation of parameters in (8) and avoid the need for high accuracy in modeling the regularizer, we set $M = n$. Inspired by (7), we propose a parameterization of the regularizer as follows:

$$\mathcal{G}(\mathbf{x}) = \sum_{i=1}^{n} q_i \int_0^{\mathbf{a}_i^T \mathbf{x} + b_i} \widehat{\xi}^{-1}(y) dy - \frac{1}{2}\|\mathbf{x}\|^2, \tag{9}$$

where $\widehat{\xi}(y)$ is a monotonically non-decreasing univariate activation function. Furthermore, we define a multivariate activation function:

$$\xi(\mathbf{x}) = \mathbf{A}^{-T}\left[\widehat{\xi}\left((\mathbf{A}\mathrm{diag}(\boldsymbol{q}))^{-1}\mathbf{x}\right) - \mathbf{b}\right] : \mathbb{R}^n \to \mathbb{R}^n, \tag{10}$$

where $\hat{\xi}$ is applied entry-wise to the vector $(\mathbf{A}\mathrm{diag}(\boldsymbol{q}))^{-1}\mathbf{x}$. Based on this, we have the following theorem.

Theorem 1. *Given the multivariate activation function ξ in (10), for any $\mathbf{A} = (\mathbf{a}_1, \cdots, \mathbf{a}_n)$, $\boldsymbol{q} = (q_1, \cdots, q_n)^T$, and $\mathbf{b} = (b_1, \cdots, b_n)^T$, such that ξ is well defined, the solution to the proximal operator*

$$\mathbf{y} = \underset{\mathbf{y}}{\mathrm{argmin}} \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|^2 + \mathcal{G}(\mathbf{y}) \tag{11}$$

with $\mathcal{G}$ given in (9) is exactly

$$\mathbf{y} = \xi(\mathbf{x}).$$

Proof: The optimality condition of (11) is:

$$\mathbf{0} \in \sum_{i=1}^{n} q_i \widehat{\xi}^{-1}\left(\mathbf{a}_i^T \mathbf{y} + b_i\right)\mathbf{a}_i - \mathbf{x}. \tag{12}$$

Since $\mathbf{A}$ is invertible, its columns are independent. Furthermore, since all q_i's are non-zero, we can uniquely represent $\mathbf{x}$ as

$$\mathbf{x} = \sum_{i=1}^{n} q_i \beta_i \mathbf{a}_i = \mathbf{A}\,\mathrm{diag}(\boldsymbol{q})\boldsymbol{\beta}, \tag{13}$$

where $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_n)^T$. Then $\widehat{\xi}^{-1}\left(\mathbf{a}_i^T\mathbf{y} + b_i\right) = \beta_i, i = 1, \cdots, n$, provides a solution to (12), and we have

$$\mathbf{a}_i^T\mathbf{y} = \widehat{\xi}\,(\beta_i) - b_i, \quad i = 1, \cdots, n, \tag{14}$$

which can be written in matrix form as

$$\mathbf{A}^T\mathbf{y} = \widehat{\xi}(\boldsymbol{\beta}) - \mathbf{b}. \tag{15}$$

Therefore, the solution to (11) is given by

$$\begin{aligned}\mathbf{y} =& \mathbf{A}^{-T}(\widehat{\xi}(\boldsymbol{\beta}) - \mathbf{b}) \\ =& \mathbf{A}^{-T}\left[\widehat{\xi}\left((\mathbf{A}\,\mathrm{diag}(\boldsymbol{q}))^{-1}\mathbf{x}\right) - \mathbf{b}\right] \\ =& \xi(\mathbf{x}).\end{aligned} \tag{16}$$

■

Thus, a connection is established between the non-separable multivariate regularizer $\mathcal{G}(\mathbf{x})$ and the multivariate activation function $\xi(\mathbf{x})$ through the multivariate proximal operator. For instance, by choosing a multivariate regularizer, we can uniquely determine the multivariate activation function as its proximal operator. Conversely, if we choose a multivariate activation function in the form of (10), where the parameters satisfy certain conditions (to be specified in Section 3.2 after $\hat{\xi}$ is parameterized), then the multivariate regularizer is also uniquely determined. With this analysis, learning a multivariate regularizer $\mathcal{G}(\mathbf{x})$ can be transformed into learning a multivariate activation function $\xi(\mathbf{x})$ that satisfies certain conditions.

3.2. Structure of the Activation Function

In order to learn the activation function $\xi(\mathbf{x})$, we need to learn the parameters $\mathbf{A}$, $\boldsymbol{q}$, and $\mathbf{b}$, as well as the univariate function $\hat{\xi}$. To ensure that $\mathcal{G}(\mathbf{x})$ serves as a sparse regularizer, the proximal operator $\xi(\mathbf{x})$ should be monotone and map a neighborhood of $\mathbf{0}$ to $\mathbf{0}$ (i.e., $\mathbf{0} \in \xi^{-1}(\mathbf{0})$).

For ease of learning, we can first learn

$$\xi(\mathbf{x}) = \widehat{\mathbf{A}}^T[\widehat{\xi}(\mathrm{diag}(\widehat{\boldsymbol{q}})\widehat{\mathbf{A}}\mathbf{x}) - \mathbf{b}] \tag{17}$$

and then obtain $\mathbf{A} = \widehat{\mathbf{A}}^{-1}$ and $\boldsymbol{q} = \frac{1}{\widehat{\boldsymbol{q}}}$, where the reciprocal is computed entry-wise. By defining $\widehat{\mathbf{x}} = \widehat{\mathbf{A}}\mathbf{x}$, we have

$$\begin{aligned}\langle \xi(\mathbf{x}) - \xi(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle =& \left\langle \widehat{\mathbf{A}}^T[\widehat{\xi}(\mathrm{diag}(\widehat{\boldsymbol{q}})\widehat{\mathbf{A}}\mathbf{x}) - \mathbf{b}] - \widehat{\mathbf{A}}^T[\widehat{\xi}(\mathrm{diag}(\widehat{\boldsymbol{q}})\widehat{\mathbf{A}}\mathbf{y}) - \mathbf{b}], \mathbf{x} - \mathbf{y} \right\rangle \\ =& \langle \widehat{\xi}(\mathrm{diag}(\widehat{\boldsymbol{q}})\widehat{\mathbf{x}}) - \widehat{\xi}(\mathrm{diag}(\widehat{\boldsymbol{q}})\widehat{\mathbf{y}}), \widehat{\mathbf{x}} - \widehat{\mathbf{y}} \rangle.\end{aligned} \tag{18}$$

Since $\widehat{\xi}$ is non-decreasing and entry-wise, the above expression is non-negative for all $\mathbf{x}$ and $\mathbf{y}$ if and only if $\widehat{\boldsymbol{q}} > \mathbf{0}$. Thus, $\xi(\mathbf{x})$ is monotone if $\widehat{\boldsymbol{q}} > \mathbf{0}$. However, even with $\widehat{\boldsymbol{q}} > \mathbf{0}$, the regularizer $\mathcal{G}(\mathbf{x})$ may still be non-convex due to its second term $-\frac{1}{2}\|\mathbf{x}\|^2$. If we want $\mathcal{G}(\mathbf{x})$ to be convex and $\widehat{\xi}$ is differentiable, we can require $\sum_{i=1}^{n} \frac{q_i}{\widehat{\xi}'(\mathbf{a}_i^T\mathbf{y}+b_i)}\mathbf{a}_i\mathbf{a}_i^T \succcurlyeq \mathbf{I}$. However, since convexity is not required for $\mathcal{G}(\mathbf{x})$, this condition is not enforced during the learning process. Actually, we only require that $\mathcal{G}(\mathbf{x})$ is non-negative. Without loss of generality, we can fix $\widehat{\xi}(0) = 0$ by allowing $\mathbf{b}$ to compensate for the offset of $\widehat{\xi}$.

It is easy to see that if we choose $\mathbf{b} = \mathbf{0}$ and $\widehat{\xi}(x) = 0$ for $x \in [-b, a]$, where $a, b > 0$, then $\xi(\mathbf{x}) = \mathbf{0}$ when $\|\mathbf{x}\|_2 \leq \frac{\min(a,b)}{\max_i\{|\widehat{q}_i|\|\widehat{\mathbf{a}}_i\|_2\}}$, where $\widehat{\mathbf{A}} = (\widehat{\mathbf{a}}_1, \widehat{\mathbf{a}}_2, \cdots, \widehat{\mathbf{a}}_n)^T$. Therefore, it is easy to make $\xi(\mathbf{x})$ all zero. To make part of $\xi(\mathbf{x})$ zero, we need $\mathbf{u} = \widehat{\xi}(\mathrm{diag}(\widehat{\boldsymbol{q}})\widehat{\mathbf{A}}\mathbf{x})$ to not be all zeros. Most entries of $\mathbf{u}$ should be zeros to ensure that $\xi(\mathbf{x}) = \widehat{\mathbf{A}}^T\mathbf{u}$ is sparse. This requires $\widehat{\mathbf{A}}$ to be a sparse matrix. In summary, the parameters $\mathbf{A}$, $\boldsymbol{q}$, and $\mathbf{b}$, as well as the function $\hat{\xi}$, should satisfy the following conditions:

$$1.\ \widehat{\boldsymbol{q}} > 0; \tag{19}$$

$$2.\ \widehat{\mathbf{A}} \text{ is sparse and invertible;} \tag{20}$$

$$3.\ \widehat{\xi} \text{ is non-decreasing and } \widehat{\xi}(0) = 0; \tag{21}$$

$$4.\ \mathcal{G}(\mathbf{x}) \geq 0. \tag{22}$$

In the following, we investigate how to satisfy conditions (21) and (22).

Since $\widehat{\xi}$ is a function, we need to parameterize it first. We use a piecewise linear function to approximate it as (Wang et al., 2021) does, denoted as $\widehat{\xi}_{(\mu_1,\mu_2)}(x)$

with two sets of learnable parameters (μ_1, μ_2):

$$\widehat{\xi}_{(\mu_1,\mu_2)}(x) = \begin{cases} \eta_2 (x - \delta_2) + \eta_1 (\delta_2 - \delta_1), & \delta_2 \leq x, \\ \eta_1 (x - \delta_1), & \delta_1 \leq x < \delta_2, \\ 0, & -\delta_1 \leq x < \delta_1, \\ \eta_1 (x + \delta_1), & -\delta_2 \leq x < -\delta_1, \\ \eta_2 (x + \delta_2) + \eta_1 (\delta_1 - \delta_2), & x < -\delta_2, \end{cases} \tag{23}$$

where $x \in \mathbb{R}$, $0 \leq \delta_1 \leq \delta_2$, and $\eta_1, \eta_2 > 0$ are learnable parameters, ensuring that $\widehat{\xi}$ is non-decreasing. Here, $\mu_1 = (\eta_1, \delta_1)$ and $\mu_2 = (\eta_2, \delta_2)$. The inverse function $\widehat{\xi}^{-1}_{(\mu_1,\mu_2)}(y)$ can be computed as follows:

$$\widehat{\xi}^{-1}_{(\mu_1,\mu_2)}(y) = \begin{cases} \frac{y - \eta_1(\delta_2 - \delta_1)}{\eta_2} + \delta_2, & \eta_1 (\delta_2 - \delta_1) \leq y, \\ \frac{y}{\eta_1} + \delta_1, & 0 \leq y < \eta_1 (\delta_2 - \delta_1), \\ [-\delta, \delta], & y = 0, \\ \frac{y}{\eta_1} - \delta_1, & -\eta_1 (\delta_2 - \delta_1) \leq y < 0, \\ \frac{y - \eta_1(\delta_2 - \delta_1)}{\eta_2} - \delta_2 & y < -\eta_1 (\delta_2 - \delta_1). \end{cases} \tag{24}$$

Therefore, the function $g(x)$ in Equation (7), learned by parameterized activation function $\widehat{\xi}^{-1}_{(\mu_1,\mu_2)}(y)$, can be derived as:

$$g(x) = \begin{cases} \left(\frac{1}{2\eta_2} - \frac{1}{2}\right) x^2 \\ + \left(\delta_2 - \frac{\eta_1(\delta_2 - \delta_1)}{\eta_2}\right) x \\ + \frac{\eta_1(\eta_1 - \eta_2)}{2\eta_2} (\delta_2 - \delta_1)^2, & x \geq \eta_1 (\delta_2 - \delta_1), \\ \left(\frac{1}{2\eta_1} - \frac{1}{2}\right) x^2 + \delta_1 x, & 0 \leq x < \eta_1 (\delta_2 - \delta_1), \\ g(-x). & x < 0. \end{cases} \tag{25}$$

It is observed that $g(x)$ is symmetric about the y-axis. When $x = 0$, $g(x) = 0$. To ensure that $\mathcal{G}(\mathbf{x})$ is nonnegative, we may require that $g(x) \geq 0$. So the problem of choosing $g(x)$ is the same as that in (Wang et al., 2021). By the deduction in (Wang et al., 2021), the conditions for (21) and (22) are as follows:

$$\begin{aligned} &\eta_1 > 0, 1 \geq \eta_2 > 0, \\ &\delta_2 \geq \delta_1 \geq \max\left\{0, \tfrac{\eta_1 - 1}{\eta_1}\delta_2\right\}. \end{aligned} \tag{26}$$

Finally, the constraints for the learnable parameters are conditions (19), (20), and (26).

3.3. Learning the Activation Function

Given an objective function ϕ, we can design a neural network architecture to implicitly learn the regularizer $\mathcal{G}$ based on (6) (where g is replaced by $\mathcal{G}$). By our design, the proximal operator $\mathrm{Prox}_{L^{-1}\mathcal{G}}$ is equivalent to a multivariate activation function. We can rewrite (6) as:

$$\mathbf{x}^{(k+1)} = \xi_{(\widehat{\mathbf{A}},\widehat{\boldsymbol{q}},\mathbf{b},\mathcal{U})}\left(\mathbf{x}^{(k)} - \frac{1}{L}\nabla\phi\left(\mathbf{x}^{(k)}\right)\right). \tag{27}$$

Here, $\xi_{(\widehat{\mathbf{A}},\widehat{\boldsymbol{q}},\mathbf{b},\mathcal{U})}$ is a multivariate activation function parameterized by $\widehat{\mathbf{A}}, \widehat{\boldsymbol{q}}, \mathbf{b}$, and $\mathcal{U}$, where $\mathcal{U} = \{\mu_1, \mu_2\}$, as shown in (17).

Equation (27) represents the k-th layer of our designed network. The parameters can be learned using the projected gradient method since they are constrained. Automatic differentiation in deep learning platforms allows us to compute the gradient efficiently, so we only need to focus on computing the projection onto the constraints.

Directly projecting the parameters $\mathcal{U} = (\eta_1, \eta_2, \delta_1, \delta_2)^T$ onto (26) is difficult. Following (Wang et al., 2021), we can first project (η_1, η_2) and then project (δ_1, δ_2) after fixing (η_1, η_2).

For completeness, we provide the results of how to compute the projections in (Wang et al., 2021) below. The projection of (η_1, η_2) is formulated as:

$$\eta_1 = \max\{\eta_1, \epsilon\}, \quad \eta_2 = \min\{\max\{\eta_1, \epsilon\}, 1\}, \tag{28}$$

where ϵ is a small positive value. After fixing (η_1, η_2), we can project (δ_1, δ_2) onto

$$\mathcal{S}_\delta = \left\{(\delta_1, \delta_2) \mid \delta_2 \geq \delta_1 \geq \max\left\{0, \frac{\eta_1 - 1}{\eta_1}\delta_2\right\}\right\}. \tag{29}$$

More specifically, when $0 < \eta_1 \leq 1$, the projection $\mathrm{Proj}\,(\delta_1, \delta_2)$ of (δ_1, δ_2) onto $\mathcal{S}_\delta$ is given by:

$$\mathrm{Proj}\,(\delta_1, \delta_2) = \begin{cases} (\delta_1, \delta_2), & \delta_1 \geq 0, \delta_2 \geq 0, \delta_1 \leq \delta_2, \\ (0, \delta_2), & \delta_1 < 0, \delta_2 > 0, \\ (0, 0), & \delta_2 \leq \min\{0, -\delta_1\}, \\ \left(\frac{\delta_1+\delta_2}{2}, \frac{\delta_1+\delta_2}{2}\right), & \delta_1 \geq |\delta_2|. \end{cases} \tag{30}$$

When $\eta_1 > 1$, the projection of (δ_1, δ_2) onto $\mathcal{S}_\delta$ becomes:

$$\text{Proj}\left(\delta_1, \delta_2\right) = \begin{cases} \left(\delta_1, \delta_2\right), & \delta_2 \geq 0, \frac{\eta_1 - 1}{\eta_1}\delta_2 \leq \delta_1 \leq \delta_2, \\ \left(\rho_1\delta_1 + \rho_2\delta_2, \rho_2\delta_1 + \rho_3\delta_2\right), & \frac{\eta_1}{1-\eta_1}\delta_2 < \delta_1 < \frac{\eta_1 - 1}{\eta_1}\delta_2, \\ (0, 0), & \delta_2 \geq 0, \delta_1 \leq \frac{\eta_1}{1-\eta_1}\delta_2, \\ (0, 0), & \delta_2 \leq \min\left\{0, -\delta_1\right\}, \\ \left(\frac{\delta_1+\delta_2}{2}, \frac{\delta_1+\delta_2}{2}\right), & \delta_1 \geq \left|\delta_2\right|, \end{cases} \tag{31}$$

where the parameters $\{\rho_1, \rho_2, \rho_3\}$ are given by $\rho_1 = \frac{(\eta_1-1)^2}{\eta_1^2+(\eta_1-1)^2}$, $\rho_2 = \frac{\eta_1(\eta_1-1)}{\eta_1^2+(\eta_1-1)^2}$, and $\rho_3 = \frac{\eta_1^2}{\eta_1^2+(\eta_1-1)^2}$.

When dealing with condition (19), we project $\widehat{\boldsymbol{q}}$ onto the set $\{\mathbf{v}|\mathbf{v} \geq \epsilon\mathbf{1}\}$ to ensure its invertibility, where ϵ is a small positive number and $\mathbf{1}$ is an all-one vector. For condition (20), we manually specify the sparsity (percentage of non-zero weights) of $\widehat{\mathbf{A}}$ to be between 30% and 50%, which guarantees that $\widehat{\mathbf{A}}$ is both sparse and invertible. The invertibility of $\widehat{\mathbf{A}}$ is not an issue when it is not too sparse since it is somewhat random.

Since our method for learning the sparse regularizer is based on learning a multivariate activation function, we refer to it as MAF-SRL (Multivariate Activation Functions for Sparse Regularizer Learning). Algorithm 1 summarizes the MAF-SRL process, where we also make the Lipschitz constant L learnable instead of manually estimating it. After training, we obtain the optimal solution $\mathbf{x}^*$, the learned parameters $\widehat{\mathbf{A}}, \widehat{\boldsymbol{q}}, \mathbf{b}, \mathcal{U}$, as well as the sparse output. Although the sparse regularizer can also be obtained as it has the same parameters as the activation function, we do not explicitly write it down since the sparse solution has already been obtained.

4. Experiments

4.1. Experimental Analysis on Classification

To evaluate the performance of our proposed method, we conduct experiments on several real-world public classification datasets. We begin by selecting a

Algorithm 1 Sparse Regularizers Learning via Multivariate Activation Functions (MAF-SRL)

Input: A differentiable function $\phi(\mathbf{x})$, the number of layers N, a set of parameters $\mathbf{x}$ related to training data that need to be solved.

Output: The optimal solution $\mathbf{x}^*$, learned parameters $\widehat{\mathbf{A}}, \widehat{\boldsymbol{q}}, \mathbf{b}, \mathcal{U}$.

Initialize learnable parameters $\widehat{\mathbf{A}}_{(0)}, \widehat{\boldsymbol{q}}_{(0)}, \mathbf{b}_{(0)}, \mathcal{U}_{(0)}$, Lipschitz constant $L^{(0)}$, and the counter $l = 0$.

Initialize $\mathbf{x}_{(0)} = \xi_{\left(\widehat{\mathbf{A}}, \widehat{\boldsymbol{q}}, \mathbf{b}, \mathcal{U}\right)}\left(\mathbf{x} - \frac{1}{L^{(0)}} \nabla \phi(\mathbf{x})\right)$.

repeat

  for $i = 1$ **to** N **do**

    $\mathbf{x}_{(i)} = \xi_{\left(\widehat{\mathbf{A}}_{(i)}, \widehat{\boldsymbol{q}}_{(i)}, \mathbf{b}_{(i)}, \mathcal{U}_{(i)}\right)}\left(\mathbf{x}_{(i-1)} - \frac{1}{L^{(i-1)}} \nabla \phi\left(\mathbf{x}_{(i-1)}\right)\right)$.

  end for

  Update $\widehat{\mathbf{A}}_{(l)}, \widehat{\boldsymbol{q}}_{(l)}, \mathbf{b}_{(l)}, \mathcal{U}_{(l)}$ with projected gradient descent, where the loss function $\mathcal{L}(\widehat{\mathbf{A}}, \widehat{\mathbf{q}}, \mathbf{b}, \mathcal{U}) = \frac{1}{2} \left\|\mathbf{x}_{(N)} - \mathbf{x}\right\|_F^2$.

  Update counter $l = l + 1$.

until convergent

return $\mathbf{x}^* = \mathbf{x}_{(N)}$, $\widehat{\mathbf{A}}, \widehat{\boldsymbol{q}}, \mathbf{b}, \mathcal{U}$.

backbone network with the ReLU function $f(x) = \max(0, x)$ as the univariate activation function. One-hot encoding is employed to represent different classes. The softmax activation function is applied to the output layer, and the loss function used is the cross entropy. To ensure a fair comparison, we use the same backbone network on each specific dataset.

The baselines we compare with MAF-SRL are backbone networks with existing hand-crafted sparse regularizers added to their loss functions. For MAF-SRL, the ϕ in Algorithm 1 refers to the loss function of the backbone network.

We implement the models using Tensorflow. The model weights are initialized with random values drawn from a normal distribution. The size of the minibatch depends on the scale of the datasets. We set the number of layers in MAF-SRL

as $N = 16$ and fix the learning rate at $lr = 0.1$. To obtain more reliable results, we run the training process five times for each experiment. The experiments are repeated 20 times, and the average performance is reported. Accuracy and weight sparsity (the ratio of nonzero weights) are used as evaluation metrics.

4.1.1. Baselines and Datasets

We compare our MAF-SRL with several representative state-of-the-art sparse regularizers. ResNet50 is used as the backbone network. The following regularizers are considered: ℓ_1 (Candes et al., 2008; Tsagkarakis et al., 2018), ℓ_{1-2} (Yin et al., 2015; Ming et al., 2019), sparse group lasso (SGL) (Simon et al., 2013), combined group and exclusive sparsity (CGES) (Yoon & Hwang, 2017), smoothly clipped absolute deviation (SCAD) (Fan & Li, 2001), capped-ℓ_1 (Zhang, 2010), log-sum penalty (LSP) (Candes et al., 2008), minimax concave penalty (MCP) (Zhang et al., 2010), and deep sparse regularizer learning (DSRL) (Wang et al., 2021).

We selected several public classification datasets for our experiments:

- **Fashion-MNIST** (Xiao et al., 2017): This dataset consists of a training set with 60,000 instances and a test set with 10,000 examples. Each example is a 28×28 grayscale image associated with one of 10 classes.
- **MNIST** (LeCun et al., 1998): This dataset consists of 70,000 grayscale images of handwritten digits, which can be classified into 10 classes. It includes 60,000 training instances and 10,000 test samples.
- **DIGITS** (Netzer et al., 2011): This is a toy dataset of handwritten digits, composed of 1,797 grayscale images.
- **CIFAR-10** (Krizhevsky et al., 2009): This dataset consists of 60,000 color images belonging to 10 classes, with 6,000 images per class.
- **CIFAR-100** (Krizhevsky et al., 2009): This dataset is similar to CIFAR-10 but contains 100 categories instead. It consists of 60,000 color images divided into 100 classes, with 600 images per class.

- **Sensorless Drive Diagnosis (SDD)** (Bayer et al., 2013): This dataset is downloaded from the UCI repository and contains 58,508 examples obtained under 11 different operating conditions.

- **PENDIGITS** (Alimoglu & Alpaydin, 1997): This dataset is composed of 10,992 grayscale images of handwritten digits 0-9. It includes 7,494 training instances and 3,498 test samples.

- **Caltech-101** (Fei-Fei et al., 2004): This dataset consists of images from 101 object categories, with a varying number of images per category. Most images are of medium resolution, around 300×300 pixels.

4.1.2. Experimental Results and Analysis

To evaluate the effectiveness of different regularization methods for deep neural networks, we use two key metrics: prediction accuracy and weight sparsity in the backbone network. In general, a higher accuracy reflects better classification performance, while a lower weight sparsity indicates stronger regularization that helps to prevent overfitting.

Table 1 provides a comprehensive comparison of all tested models on various datasets. The results demonstrate that our proposed MAF-SRL method consistently outperforms other baseline methods in terms of both accuracy and sparsity. Specifically, MAF-SRL achieves the highest accuracy and the lowest weight sparsity on all tested datasets, demonstrating its effectiveness in improving the generalization and interpretability of deep neural networks.

This superior performance can largely be attributed to the learned multivariate sparse regularizer adopted in MAF-SRL. Unlike traditional hand-crafted regularization methods, this flexible approach can effectively capture the intrinsic correlations among different features and reduce their redundancy while retaining their discriminative power. Moreover, MAF-SRL has the capability to adaptively adjust the strength of regularization based on the complexity and characteristics of different datasets, resulting in better generalization of the model. Furthermore, our proposed method can significantly reduce the number of parameters in the

Table 1: Performance of different methods on the datasets. The weight sparsity is the ratio of nonzero weights in the backbone network.

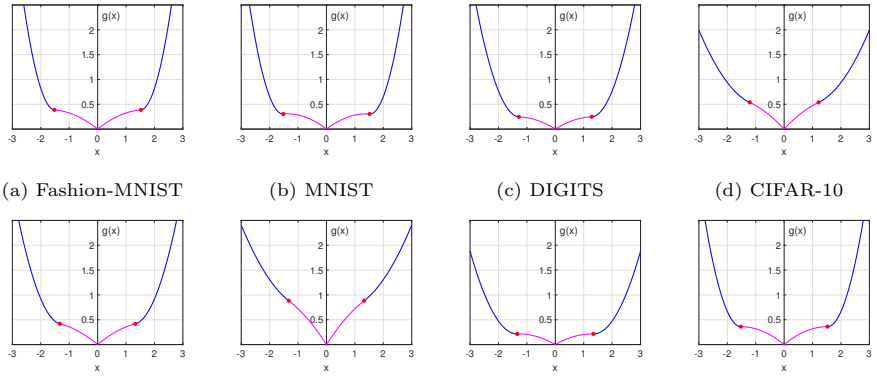
Dataset	Measure	ℓ_1	ℓ_{1-2}	SGL	CGES	SCAD	capped-ℓ_1	LSP	MCP	DSRL	MAF-SRL (ours)
Fashion-MNIST	accuracy	0.9124	0.9281	0.8924	0.8873	0.8671	0.8982	0.9031	0.9127	0.9358	**0.9421**
	weight sparsity	0.2398	0.4363	0.4218	0.2819	0.5728	0.6629	0.2763	0.3397	0.2546	**0.1537**
MNIST	accuracy	0.9642	0.9538	0.9863	0.9837	0.9824	0.9563	0.9563	0.9623	0.9816	**0.9921**
	weight sparsity	0.1727	0.2735	0.1029	0.2013	0.1197	0.1126	0.0928	0.3328	0.1596	**0.0629**
DIGITS	accuracy	0.8638	0.8837	0.8542	0.8837	0.8682	0.8538	0.8772	0.8831	0.8941	**0.9028**
	weight sparsity	0.3387	0.2928	0.2901	0.4283	0.4419	0.2765	0.5319	0.4019	0.2079	**0.1774**
CIFAR-10	accuracy	0.8238	0.8188	0.8092	0.8542	0.8452	0.8562	0.8458	0.8229	0.8643	**0.8759**
	weight sparsity	0.6784	0.5829	0.5429	0.4492	0.5186	0.6294	0.5529	0.3165	0.3081	**0.2396**
CIFAR-100	accuracy	0.7329	0.7219	0.6872	0.7239	0.6549	0.7129	0.7278	0.7362	0.7511	**0.7769**
	weight sparsity	0.5587	0.4982	0.8829	0.7623	0.4927	0.6549	0.5498	0.4892	0.4672	**0.3225**
SDD	accuracy	0.9829	0.9669	0.9539	0.9827	0.9567	0.9632	0.9862	0.9685	0.9846	**0.9941**
	weight sparsity	0.3092	0.4294	0.2397	0.4962	0.2981	0.3982	0.5729	0.4839	0.2153	**0.1703**
PENDIGITS	accuracy	0.9852	0.9902	0.9762	0.9683	0.9719	0.9629	0.9739	0.9827	0.9816	**0.9958**
	weight sparsity	0.6931	0.3397	0.6791	0.3018	0.2973	0.7538	0.5392	0.4492	0.2371	**0.1778**
Caltech-101	accuracy	0.9733	0.9758	0.9883	0.9901	0.9632	0.9857	0.9683	0.9775	0.9816	**0.9949**
	weight sparsity	0.3679	0.4133	0.5582	0.6271	0.3036	0.2279	0.3864	0.4272	0.2361	**0.1762**

backbone network, thereby improving the computational efficiency of the model, which is particularly important for practical applications where computational resources are often limited.

In addition to evaluating the performance of different regularization methods, we also provide visualizations and further analysis to gain insights into our proposed MAF-SRL method.

Figure 1 showcases the learned univariate function $g(x)$ for various datasets. We also report the learned parameters in $\widehat{\xi}_{(\eta_1,\eta_2,\delta_1,\delta_2)}(x)$, highlighting the points $x = \pm\eta_1(\delta_2 - \delta_1)$ in red. It is interesting to observe that g exhibits non-convex behavior, particularly within the interval $[-\eta_1(\delta_2-\delta_1), \eta_1(\delta_2-\delta_1)]$. This characteristic suggests that our learned sparse regularizer possesses a flexible and adaptive nature, allowing it to effectively capture complex relationships in the data.

Furthermore, we delve into the effect of the number of layers N on the performance of our learned sparse regularizer, as shown in Figure 2. By varying the layer number from 2 to 30 while keeping the learning rate fixed at 0.1, we



(a) Fashion-MNIST (b) MNIST (c) DIGITS (d) CIFAR-10

(e) CIFAR-100 (f) SDD (g) PENDIGITS (h) Caltech-101

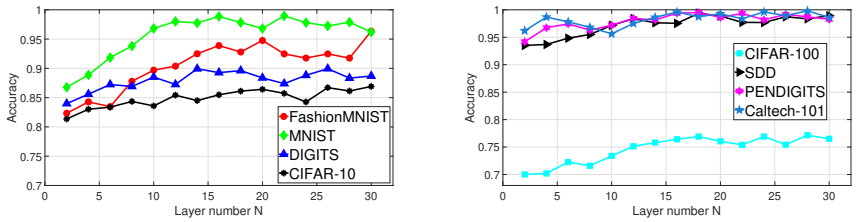
Figure 1: The learned univariate function $g(x)$ given in (25) on different datasets for classification tasks. Its associated parameters are as follows: (a) $\eta_1 = 1.37, \eta_2 = 0.22, \delta_1 = 0.46, \delta_2 = 1.57$. (b) $\eta_1 = 1.46, \eta_2 = 0.24, \delta_1 = 0.44, \delta_2 = 1.48$. (c) $\eta_1 = 1.35, \eta_2 = 0.34, \delta_1 = 0.36, \delta_2 = 1.31$. (d) $\eta_1 = 1.41, \eta_2 = 0.62, \delta_1 = 0.62, \delta_2 = 1.49$. (e) $\eta_1 = 1.33, \eta_2 = 0.36, \delta_1 = 0.48, \delta_2 = 1.47$. (f) $\eta_1 = 1.51, \eta_2 = 0.64, \delta_1 = 0.89, \delta_2 = 1.77$. (g) $\eta_1 = 1.34, \eta_2 = 0.45, \delta_1 = 0.33, \delta_2 = 1.33$. (h) $\eta_1 = 1.44, \eta_2 = 0.27, \delta_1 = 0.47, \delta_2 = 1.53$.

analyze how increasing the depth impacts the model's accuracy. The results reveal a general trend where the accuracy improves with a greater number of layers and stabilizes when $N > 16$. Consequently, we have chosen $N = 16$ as the optimal layer number for our previous experiments.

These additional visualizations and analyses provide valuable insights into the behavior and adaptability of our proposed MAF-SRL method. They demonstrate the unique characteristics of the learned sparse regularizer and its ability to capture intricate patterns within diverse datasets. Such understanding enables us to leverage the strengths of MAF-SRL for improved regularization and performance enhancement in deep neural networks.

4.2. Exploring Multi-View Clustering with MAF-SRL

To address the task of multi-view clustering, we apply our proposed MAF-SRL algorithm to the task of multi-view clustering, using the experimental setup and datasets described in (Wang et al., 2021). The aim of multi-view clustering is to cluster data based on multiple views of the same set of objects. We consider



(a) FashionMNIST, MNIST, DIGITS, CIFAR-10 (b) CIFAR-100, SDD, PENDIGITS, Caltech-101

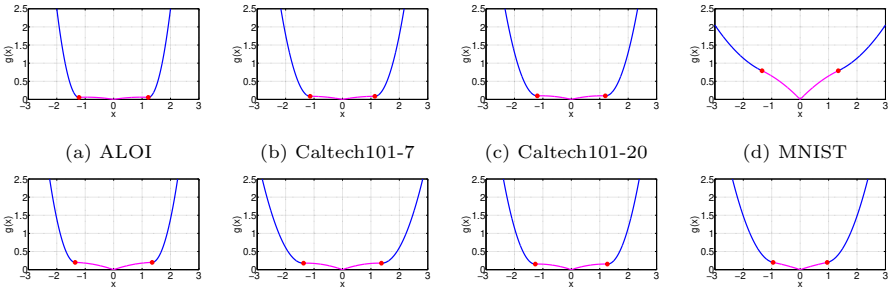
Figure 2: The relationship between classification performance (accuracy) and the number of layers N in the proposed MAF-SRL.

multi-view data $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{v}$, where each view $\mathbf{x}_i$ has n samples and d_i features, and we aim to learn a cluster indicator $\mathbf{y} \in \{0,1\}^n$ by optimizing an affinity matrix $\mathbf{W}$ from the multi-view similarity matrices $\mathcal{W} = \{\mathbf{W}_i\}_{i=1}^{v}$.

To solve this problem, we use a simple optimization framework that minimizes the Frobenius norm between $\mathbf{W}$ and a convex combination of the individual view affinity matrices, subject to a learned sparse regularizer $g(\cdot)$. We separately optimize the weights $\boldsymbol{\alpha}$ using the ADMM algorithm, and then compute the optimal solution for $\mathbf{W}$ using the MAF-SRL framework.

Our proposed MAF-SRL method employs an activation function, as described in equation (10), to facilitate the effective modeling of non-linear relationships within the data. To ensure optimal performance, we carefully initialize the parameterized activation function by tuning the values of η_1 and η_2 to 1.0, while setting δ_1 and δ_2 to 1.0 and 2.0 respectively. It is worth noting that these initialization values may vary across different datasets, as the sparse regularizers are learned in a data-driven manner.

The activation function $\xi(\mathbf{x})$ beautifully encapsulates the essence of our approach. By leveraging the learned parameters $\{\eta_1, \eta_2, \delta_1, \delta_2\}$, which adapt to the characteristics of individual datasets, we can effectively capture the intricate relationships and patterns present in the data. Figure 3 visually demonstrates the remarkable power of the activation function $\xi(\mathbf{x})$, showcasing the learned univariate function $g(x)$. These visualizations provide valuable insights into the behavior and effectiveness of the MAF-SRL approach across a range of test



(a) ALOI (b) Caltech101-7 (c) Caltech101-20 (d) MNIST

(e) NUS-WIDE (f) MSRC-v1 (g) ORL (h) Youtube

Figure 3: The learned univariate function $g(x)$ on different datasets for multi-view clustering. Its associated parameters are as follows: (a) $\eta_1 = 1.16, \eta_2 = 0.11, \delta_1 = 0.13, \delta_2 = 1.18$. (b) $\eta_1 = 1.15, \eta_2 = 0.18, \delta_1 = 0.15, \delta_2 = 1.14$. (c) $\eta_1 = 1.22, \eta_2 = 0.21, \delta_1 = 0.19, \delta_2 = 1.17$. (d) $\eta_1 = 1.49, \eta_2 = 0.68, \delta_1 = 0.81, \delta_2 = 1.71$. (e) $\eta_1 = 1.22, \eta_2 = 0.15, \delta_1 = 0.27, \delta_2 = 1.38$. (f) $\eta_1 = 1.28, \eta_2 = 0.31, \delta_1 = 0.28, \delta_2 = 1.35$. (g) $\eta_1 = 1.28, \eta_2 = 0.21, \delta_1 = 0.26, \delta_2 = 1.25$. (h) $\eta_1 = 1.12, \eta_2 = 0.33, \delta_1 = 0.26, \delta_2 = 1.11$.

datasets specifically designed for clustering tasks.

It's worth emphasizing that all the learned parameters of the activation functions strictly adhere to the conditions specified in equation (26). This ensures the validity and reliability of our approach in generating meaningful and accurate clustering results. By diligently adhering to these conditions, we guarantee that our activation function effectively captures the intrinsic structure of the data, ultimately leading to improved clustering performance.

In order to thoroughly evaluate the performance of our proposed MAF-SRL method, we conducted extensive experiments on various test datasets. Table 2 presents the comprehensive results of these experiments, showcasing the clustering accuracy metrics ACC, NMI, and ARI for different sparse regularizers. Notably, our MAF-SRL approach consistently outperforms all the manually designed sparse regularizers that were included in the comparison. This remarkable improvement in performance highlights the effectiveness and superiority of our proposed method.

Another significant aspect of our approach is the sparsity of the learned sparse regularizers, which directly influences the efficiency and interpretability of

Table 2: Clustering accuracy with different sparse regularizers, where the best performance is highlighted in bold.

Dataset	Metrics	ℓ_1	ℓ_{1-2}	SGL	CGES	SCAD	capped-ℓ_1	LSP	MCP	DSRL	MAF-SRL (ours)
ALOI	ACC	0.6538	0.7146	0.6637	0.7129	0.6329	0.7792	0.7349	0.7839	0.7871	**0.8374**
	NMI	0.7473	0.7639	0.6993	0.7742	0.6395	0.7983	0.7539	0.7439	0.7872	**0.8849**
	ARI	0.6521	0.6029	0.5983	0.6849	0.6175	0.6844	0.6833	0.5948	0.6172	**0.7219**
Caltech101-7	ACC	0.7129	0.6674	0.8129	0.7749	0.8022	0.7375	0.8355	0.7983	0.8382	**0.8935**
	NMI	0.6639	0.7174	0.6893	0.7112	0.7329	0.7121	0.6899	0.7019	0.6162	**0.7495**
	ARI	0.5948	0.6849	0.5584	0.6439	0.5992	0.6217	0.6549	0.6493	0.6192	**0.7042**
Caltech101-20	ACC	0.7753	0.6139	0.7399	0.6539	0.6926	0.7893	0.7837	0.8127	0.7292	**0.8539**
	NMI	0.6783	0.6648	0.7129	0.7583	0.6929	0.6327	0.6349	0.6649	0.6823	**0.8003**
	ARI	0.6938	0.7093	0.6928	0.5947	0.6548	0.7326	0.7133	0.7247	0.7381	**0.7749**
MNIST	ACC	0.8031	0.7749	0.7388	0.6928	0.7449	0.8022	0.7837	0.7762	0.8563	**0.8636**
	NMI	0.7233	0.6649	0.6538	0.6644	0.6291	0.7129	0.7459	0.7837	0.7562	**0.8329**
	ARI	0.7749	0.6938	0.6554	0.7034	0.7592	0.7783	0.6846	0.7206	0.7541	**0.8022**
NUS-WIDE	ACC	0.5053	0.4927	0.4872	0.4551	0.4029	0.3993	0.4892	0.4463	0.4032	**0.5339**
	NMI	0.1948	0.2984	0.3336	0.2841	0.3083	0.2943	0.2988	0.2875	0.2651	**0.3992**
	ARI	0.3294	0.3847	0.2998	0.3736	0.2988	0.4539	0.3352	0.4029	0.1551	**0.4873**
MSRC-v1	ACC	0.8392	0.7539	0.7733	0.8024	0.7938	0.8147	0.8473	0.7939	0.8343	**0.8893**
	NMI	0.6547	0.7749	0.7328	0.7459	0.7201	0.6993	0.7023	0.6994	0.7701	**0.7938**
	ARI	0.6839	0.7055	0.7649	0.7351	0.6694	0.7639	0.7627	0.6891	0.6963	**0.8036**
ORL	ACC	0.8732	0.7793	0.8265	0.8004	0.8501	0.8837	0.7322	0.7837	0.8362	**0.9038**
	NMI	0.7935	0.8118	0.7894	0.8227	0.8092	0.7684	0.8106	0.7787	**0.9132**	0.8547
	ARI	0.6839	0.7128	0.8005	0.7739	0.6845	0.7493	0.6949	0.7239	0.7571	**0.8458**
Youtube	ACC	0.4297	0.4471	0.5076	0.5309	0.4727	0.5574	0.5157	0.6029	0.4213	**0.6249**
	NMI	0.4893	0.5092	0.4474	0.3981	0.4983	0.5427	0.4971	0.4925	0.2703	**0.5882**
	ARI	0.1939	0.3742	0.2947	0.3903	0.3131	0.2992	0.3832	0.3981	0.1833	**0.4585**

the backbone network. Table 3 specifically quantifies the sparsity by calculating the ratio of nonzero weights in the backbone network. This metric provides valuable insights into the degree of compactness and simplicity achieved by our approach.

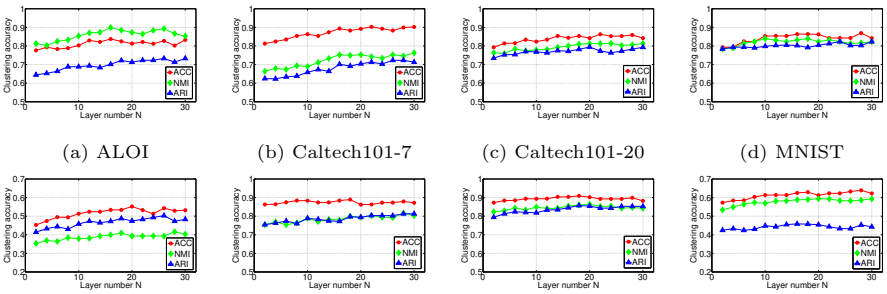
Our experimental results clearly demonstrate that the MAF-SRL method attains superior clustering accuracy while simultaneously ensuring a reasonable level of sparsity within the learned regularizers. This balance between accuracy and sparsity is crucial, as it allows us to effectively capture the essential characteristics of the data while maintaining the interpretability and efficiency of the model.

Table 3: The weight sparsity of different methods on the datasets for multi-view clustering.

Dataset	ℓ_1	ℓ_{1-2}	SGL	CGES	SCAD	capped-ℓ_1	LSP	MCP	DSRL	MAF-SRL (ours)
ALOI	0.2849	0.2283	0.2937	0.2827	0.2948	0.2301	0.2529	0.2729	0.0817	**0.0803**
Caltech101-7	0.2039	0.2312	0.2574	0.2739	0.2029	0.2938	0.2993	0.2395	**0.0827**	0.0901
Caltech101-20	0.2648	0.2947	0.2357	0.3003	0.2995	0.3021	0.2854	0.2836	0.0782	**0.0715**
MNIST	0.2029	0.2227	0.2518	0.2922	0.1988	0.2022	0.2837	0.1962	0.0666	**0.0588**
NUS-WIDE	0.2936	0.3315	0.3079	0.3128	0.2975	0.3287	0.3762	0.3529	0.1187	**0.1125**
MSRC-v1	0.2531	0.3128	0.2483	0.2839	0.2617	0.2491	0.2148	0.2455	0.0820	**0.0802**
ORL	0.1321	0.1732	0.1206	0.1995	0.2012	0.1883	0.1381	0.1937	0.0287	**0.0262**
Youtube	0.2947	0.2348	0.2865	0.2917	0.2649	0.2833	0.2846	0.2753	**0.0709**	0.0851

An important aspect of our research is the investigation of the performance of the MAF-SRL method for clustering tasks with varying numbers of layers. To thoroughly evaluate the impact of layer number on clustering accuracy, we conducted experiments and obtained insightful results, as illustrated in Figure 4. In these experiments, we examined the clustering performance metrics ACC, NMI, and ARI while systematically varying the number of layers in the MAF-SRL model. The layer number was incrementally increased from 2 to 30, and a fixed learning rate lr of 0.15 was utilized throughout. Notably, the results demonstrate a significant pattern: as the number of layers increases, the clustering accuracy generally improves. This observation aligns with our expectations, as the successive addition of layers allows for more complex and abstract representations to be learned by the model. Consequently, the model becomes increasingly capable of capturing intricate patterns and relationships within the data, leading to enhanced clustering accuracy. However, it is noteworthy that there is a point of saturation in the accuracy improvement trend. Specifically, the performance stabilizes when the number of layers exceeds 16. Beyond this point, the additional layers do not contribute significantly to further accuracy improvements. This finding suggests that there is an optimal point of layer number for achieving the best clustering performance with the MAF-SRL method.

The MAF-SRL approach excels at clustering tasks by utilizing data-driven sparse regularizers and our proposed multivariate activation functions. Our experimental results show that the best clustering accuracy is achieved with



(a) ALOI (b) Caltech101-7 (c) Caltech101-20 (d) MNIST

(e) NUS-WIDE (f) MSRC-v1 (g) ORL (h) Youtube

Figure 4: The relations among clustering accuracy (ACC, ARI and NMI) and layer number in $\{2, 4, \cdots, 30\}$ of the proposed method MAF-SRL.

the most sparse outputs (lowest percentage of nonzero weights), demonstrating the effectiveness of the learned sparse regularizers. These regularizers are more robust when applied to various datasets due to their ability to learn a data-driven sparse representation of similarity matrices. MAF-SRL's adaptability and efficiency in learning such sparse representations using multivariate activation functions make it a promising solution for tackling complex clustering tasks across different domains. Furthermore, the learned sparse regularizer exhibits strong generalization capability, which makes MAF-SRL practical for real-world applications.

5. Conclusion

In this paper, we propose MAF-SRL, a method for learning non-separable multivariate sparse regularizers implicitly. Following (Wang et al., 2021), we establish a correspondence between multivariate sparse regularizers and multivariate activation functions through the proximal operator, thereby converting the learning of a multivariate sparse regularizer into the learning of a multivariate activation function. We derive the conditions that the parameters of the multivariate activation function should satisfy and employ the projected gradient method to train these parameters. Experimental results demonstrate that our MAF-SRL framework achieves higher accuracy and sparser weights compared to

existing hand-crafted sparse regularizers.

References

Alimoglu, F., & Alpaydin, E. (1997). Combining multiple representations and classifiers for pen-based handwritten digit recognition. In *Proceedings of the Fourth International Conference on Document Analysis and Recognition* (pp. 637–640 vol.2). volume 2.

Atserias, A., & Müller, M. (2020). Automating resolution is NP-hard. *Journal of the ACM*, *67*, 1–17.

Bayer, C., Enge-Rosenblatt, O., Bator, M., & Mönks, U. (2013). Sensorless drive diagnosis using automated feature extraction, significance ranking and reduction. In *2013 IEEE 18th Conference on Emerging Technologies Factory Automation (ETFA)* (pp. 1–4).

Bibi, A., Ghanem, B., Koltun, V., & Ranftl, R. (2019). Deep layers as stochastic solvers. In *7th International Conference on Learning Representations, ICLR*.

Bore, J. C., Ayedh, W. M. A., Li, P., Yao, D., & Xu, P. (2019). Sparse autoregressive modeling via the least absolute LP-norm penalized solution. *IEEE Access*, *7*, 40959–40968.

Bui, K., Park, F., Zhang, S., Qi, Y., & Xin, J. (2021). Structured sparsity of convolutional neural networks via nonconvex sparse group regularization. *Frontiers in applied mathematics and statistics*, *6*, 529564.

Candes, E. J., Wakin, M. B., & Boyd, S. P. (2008). Enhancing sparsity by reweighted L1 minimization. *Journal of Fourier Analysis and Applications*, *14*, 877–905.

Celentano, M., Montanari, A., & Wei, Y. (2023). The lasso with general gaussian designs with applications to hypothesis testing. *The Annals of Statistics*, *51*, 2194–2220.

Chen, M., Wang, Q., Chen, S., & Li, X. (2019). Capped $l1$-norm sparse representation method for graph clustering. *IEEE Access*, *7*, 54464–54471.

Chen, T., & Chen, H. (1995). Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, *6*, 911–917.

Chen, Y., Yamagishi, M., & Yamada, I. (2021). A generalized moreau enhancement of l12-norm and its application to group sparse classification. In *2021 29th European Signal Processing Conference (EUSIPCO)* (pp. 2134–2138). IEEE.

Combettes, P. L., & Pesquet, J.-C. (2020). Deep neural network structures solving variational inequalities. *Set-Valued and Variational Analysis*, (pp. 1–28).

Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, *96*, 1348–1360.

Fei-Fei, L., Fergus, R., & Perona, P. (2004). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 Conference on Computer Vision and Pattern Recognition Workshop* (pp. 178–178). IEEE.

Fonti, V., & Belitser, E. (2017). Feature selection using lasso. *VU Amsterdam Research Paper in Business Analytics*, *30*, 1–25.

Hillar, C. J., & Lim, L.-H. (2013). Most tensor problems are NP-hard. *Journal of the ACM*, *60*, 1–39.

Hirahara, S. (2022). Np-hardness of learning programs and partial mcsp. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)* (pp. 968–979). IEEE.

Issa, I., & Gastpar, M. (2018). Computable bounds on the exploration bias. In *2018 IEEE International Symposium on Information Theory (ISIT)* (pp. 576–580). IEEE.

Jiang, H., Zheng, W., Luo, L., & Dong, Y. (2019). A two-stage minimax concave penalty based method in pruned adaboost ensemble. *Applied Soft Computing*, *83*, 105674.

Kadhim, A. A. S. (2023). The smoothly clipped absolute deviation (scad) penalty variable selection regularization method for robust regression discontinuity designs. In *AIP Conference Proceedings*. AIP Publishing volume 2776.

Kim, G.-S., & Paik, M. C. (2019). Doubly-robust lasso bandit. *Advances in Neural Information Processing Systems*, *32*, 5877–5887.

Krizhevsky, A., Hinton, G. et al. (2009). Learning multiple layers of features from tiny images.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*, 2278–2324.

Li, G., Ma, C., & Srebro, N. (2022a). Pessimism for offline linear contextual bandits using lp confidence sets. *Advances in Neural Information Processing Systems*, *35*, 20974–20987.

Li, J., Fang, C., & Lin, Z. (2019). Lifted proximal operator machines. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 4181–4188). volume 33.

Li, X. P., Shi, Z.-L., Liu, Q., & So, H. C. (2022b). Fast robust matrix completion via entry-wise l0-norm minimization. *IEEE Transactions on Cybernetics*, .

Li, Z., Wan, C., Tan, B., Yang, Z., & Xie, S. (2020). A fast DC-based dictionary learning algorithm with the scad penalty. Elsevier.

Liao, X., Wei, X., & Zhou, M. (2023). Minimax concave penalty regression for superresolution image reconstruction. *IEEE Transactions on Consumer Electronics*, .

Liu, Z., & Yu, S. (2023). Alternating direction method of multipliers based on l20-norm for multiple measurement vector problem. *arXiv preprint arXiv:2303.10616*, .

Lou, Y., Yin, P., He, Q., & Xin, J. (2015). Computing sparse representation in a highly coherent dictionary based on difference of L1 and L2. *Journal of Scientific Computing*, *64*, 178–196.

Lu, C., Zhu, C., Xu, C., Yan, S., & Lin, Z. (2015). Generalized singular value thresholding. In *Proceedings of the AAAI Conference on Artificial Intelligence*. volume 29.

Mazumder, R., Friedman, J. H., & Hastie, T. (2011). Sparsenet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association*, *106*, 1125–1138.

Ming, D., Ding, C., & Nie, F. (2019). A probabilistic derivation of LASSO and L1-2-norm feature selections. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 4586–4593). volume 33.

Moayeri, M., Banihashem, K., & Feizi, S. (2022). Explicit tradeoffs between adversarial and natural distributional robustness. *Advances in Neural Information Processing Systems*, *35*, 38761–38774.

Natarajan, B. K. (1995). Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, *24*, 227–234.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning.

Ohn, I., & Kim, Y. (2022). Nonconvex sparse regularization for deep neural networks and its optimality. *Neural computation*, *34*, 476–517.

Pardo-Simon, L. (2023). Splitting hairs with transcendental entire functions. *International Mathematics Research Notices*, *2023*, 13387–13425.

Prater-Bennette, A., Shen, L., & Tripp, E. E. (2022). The proximity operator of the log-sum penalty. *Journal of Scientific Computing*, *93*, 67.

Sharif, M., Bauer, L., & Reiter, M. K. (2018). On the suitability of Lp-norms for creating and preventing adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Simon, N., Friedman, J., Hastie, T., & Tibshirani, R. (2013). A sparse-group lasso. *Journal of Computational and Graphical Statistics*, *22*, 231–245.

Sriramanan, G., Gor, M., & Feizi, S. (2022). Toward efficient robust training against union of lp threat models. *Advances in Neural Information Processing Systems*, *35*, 25870–25882.

Tang, A., Niu, L., Miao, J., & Zhang, P. (2023). Training compact dnns with l-12 regularization. *Pattern Recognition*, *136*, 109206.

Tsagkarakis, N., Markopoulos, P. P., Sklivanitis, G., & Pados, D. A. (2018). L1-norm principal-component analysis of complex data. *IEEE Transactions on Signal Processing*, *66*, 3256–3267.

Varno, F., Saghayi, M., Rafiee Sevyeri, L., Gupta, S., Matwin, S., & Havaei, M. (2022). Adabest: Minimizing client drift in federated learning via adaptive bias estimation. In *European Conference on Computer Vision* (pp. 710–726). Springer.

Wang, G., Donhauser, K., & Yang, F. (2022). Tight bounds for minimum l1-norm interpolation of noisy data. In *International Conference on Artificial Intelligence and Statistics* (pp. 10572–10602). PMLR.

Wang, S., Chen, Z., Du, S., & Lin, Z. (2021). Learning deep sparse regularizers with applications to multi-view clustering and semi-supervised classification.

IEEE Transactions on Pattern Analysis and Machine Intelligence, *44*, 5042–5055.

Wu, S., Li, G., Deng, L., Liu, L., Wu, D., Xie, Y., & Shi, L. (2018). L1-norm batch normalization for efficient training of deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *30*, 2043–2051.

Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.

Xu, J., Chi, E., & Lange, K. (2017). Generalized linear model regression under distance-to-set penalties. In *Advances in Neural Information Processing Systems* (pp. 1385–1395).

Xu, Z., Chang, X., Xu, F., & Zhang, H. (2012). $l\_\{1/2\}$ regularization: A thresholding representation theory and a fast solver. *IEEE Transactions on Neural Networks and Learning Systems*, *23*, 1013–1027.

Yin, P., Lou, Y., He, Q., & Xin, J. (2015). Minimization of L2 for compressed sensing. *SIAM Journal on Scientific Computing*, *37*, A536–A563.

Yoon, J., & Hwang, S. J. (2017). Combined group and exclusive sparsity for deep neural networks. In *International Conference on Machine Learning* (pp. 3958–3966).

Zhang, C.-H. et al. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, *38*, 894–942.

Zhang, M., Ding, C., Zhang, Y., & Nie, F. (2014). Feature selection at the discrete limit. In *Proceedings of the AAAI Conference on Artificial Intelligence*. volume 28.

Zhang, T. (2008). Multi-stage convex relaxation for learning with sparse regularization. *Advances in Neural Information Processing Systems*, *21*, 1929–1936.

Zhang, T. (2010). Analysis of multi-stage convex relaxation for sparse regularization. volume 11.

Zhang, Y., Zhang, H., & Tian, Y. (2020). Sparse multiple instance learning with non-convex penalty. Elsevier.

Declaration of interests

☒The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: