Results in Applied Mathematics 14 (2022) 100256

Contents lists available at ScienceDirect

Results in Applied Mathematics

journal homepage: www.elsevier.com/locate/results-in-applied-mathematics

On two recent nonconvex penalties for regularization in machine learning

Sujit Vettam [a,1], Majnu John [b,c,*]

[a] The University of Chicago Booth School of Business, Chicago, IL, United States of America
[b] Departments of Mathematics and of Psychiatry, Hofstra University, Hempstead, NY, United States of America
[c] Feinstein Institutes of Medical Research, Northwell Health System, Manhasset, NY, United States of America

ARTICLE INFO

Article history:
Received 16 February 2022
Received in revised form 7 March 2022
Accepted 9 March 2022
Available online 26 March 2022

Keywords:
Regularization
Nonconvex penalties
Deep learning
Convolutional neural network

ABSTRACT

Regularization methods are often employed to reduce overfitting of machine learning models. Nonconvex penalty functions are often considered for regularization because of their near-unbiasedness properties. In this paper, we consider two relatively new penalty functions: Laplace and arctan, and show how they fit into certain recently introduced statistical and optimization frameworks. We also compare empirically the performance of the two new penalty functions with existing penalty functions utilized as regularizers of deep neural networks and convolutional neural networks on seven different datasets.

1. Introduction

Overfitting a model to training dataset is a common issue in many machine learning methods. A commonly used approach to reduce overfitting is by penalizing the model-parameters, which is accomplished by adding a penalty function, $\boldsymbol{p}_\alpha(\cdot)$, to the loss function $\mathcal{L}_n$. Widely used penalty functions include convex functions such as Lasso [1] and L_2 penalty [2,3], and nonconvex functions such as smoothly clipped absolute deviation (SCAD) [4] and minimax convex penalty (MCP) [5]. Convex penalties are suitable from an optimization perspective because if the loss function is also convex then the overall optimization problem is convex and hence any local minimum of the objective function is a global minimum. However, it is known in the literature that convex penalties such as Lasso and L_2 yield biased estimators of the model parameters [4]. This feature of the convex penalties motivated the development of nonconvex penalties such as SCAD and MCP, which provide unbiased or nearly unbiased estimators, at least for parameters with large values. Recently, [6] showed that when the loss function satisfies the restricted strong convexity (RSC) property, the local minimum of the overall objective function with nonconvex penalties such as SCAD and MCP which satisfy a set of regularity conditions, are essentially as good as a global optimum from a statistical perspective.

In this paper we study two relatively new nonconvex penalties – Laplace [7] and arctan [8] penalties, respectively – that have appeared in recent literature:

$$\boldsymbol{p}_\varepsilon(\boldsymbol{w}) = \sum_{j=1}^{q} p_\varepsilon(w_j) = \sum_{j=1}^{q} (1 - \varepsilon^{|w_j|}), \ \ \varepsilon \in (0, 1), \tag{1}$$

* Correspondence to: 350 Community Drive, Manhasset, NY 11030, United States of America.
E-mail addresses: sjv@chicagobooth.edu, svettam@uchicago.edu (S. Vettam), mjohn5@northwell.edu, majnu.john@hofstra.edu (M. John).

[1] This work was completed when the first author was a student at University of Chicago Booth School of Business. Currently employed in private sector.

https://doi.org/10.1016/j.rinam.2022.100256

$$\boldsymbol{p}_\gamma(\boldsymbol{w}) = \sum_{j=1}^{q} p_\gamma(w_j) = \sum_{j=1}^{q} \frac{2}{\pi}\arctan(\gamma\,|w_j|),\ \ \gamma > 0, \tag{2}$$

where $\boldsymbol{w} = (w_1, \ldots, w_q)$ is the parameter vector of the learning model that is being estimated during the training phase. Both penalties have not yet been widely used in high-dimensional statistical learning approaches such as deep learning and to the best of our knowledge arctan penalty has not been used *at all* so far in published literature for applications of high-dimensional statistical models. The main goals of this paper are: 1) to show how the new penalties fit in with recently introduced statistical and optimization frameworks; 2) to study empirically their utility in deep learning models.

First, in Section 2, we show that both Laplace and arctan penalties fit into the theoretical framework presented in [6], where the corresponding loss function satisfy the restricted strong convexity (RSC) property, thereby guaranteeing the good behavior of optimization algorithms with the new penalties. Next, in Section 3 we show how the two penalties Laplace and arctan fit into an optimization framework known as difference-of-convex learning recently studied in the literature [9]. In Section 4, we study the empirical performance of regularization methods in a setting for which the loss function is also known to be nonconvex [10], viz. deep neural networks. We compare the empirical performance of Laplace and arctan penalties with that of Lasso, L_2, SCAD and MCP on seven datasets; three of the datasets employed deep neural networks and four of the datasets employed convolutional neural network architecture.

2. Theoretical-guarantees when the loss function is restricted strongly convex

RSC of loss functions provides non-flat curvature guarantees in relevant directions for objective functions with decomposable regularizers (see section 9.3 in [11]). The RSC condition that we require is the one given in [6], stated as follows at an optimal parameter value $\boldsymbol{w}^*$ for the empirical loss function $\mathcal{L}_n$:

$$\langle \nabla\mathcal{L}_n(\boldsymbol{w}^* + \Delta) - \nabla\mathcal{L}_n(\boldsymbol{w}^*), \Delta\rangle \geq \begin{cases} \alpha_1\|\Delta\|_2^2 - \tau_1\frac{\log p}{n}\|\Delta\|_1^2, & \forall\|\Delta\|_2 \leq 1, \\ \alpha_2\|\Delta\|_2 - \tau_2\sqrt{\frac{\log p}{n}}\|\Delta\|_1, & \forall\|\Delta\|_2 \geq 1, \end{cases} \tag{3}$$

where $\alpha_j > 0$, $\tau_j \geq 0$, $j = 1, 2$. There are many examples of empirical loss functions including certain nonconvex loss functions that satisfy RSC [6,12].

In addition to the RSC property on $\mathcal{L}_n$, if the penalty function $p_\lambda(t)$ satisfies the regularity conditions P_1-P_5 stated below, then the Theorem 1 in [6] guarantees that squared-l_2 error of a stationary point $\tilde{\boldsymbol{w}}$ with respect to a global optimum point $\boldsymbol{w}^*$ scales as $(k\log q)/n$; here n is the sample size, q is the dimension of the problem and k is the number of non-zero elements in $\boldsymbol{w}^*$. Optimization algorithms are generally designed to reach a stationary point, but not necessarily a global optimum point. The importance of the above-mentioned theorem is that it guarantees that any such stationary point reached by a generic optimization algorithm will lie within statistical precision of the global optimum point for loss functions satisfying RSC and penalty functions satisfying P_1-P_5. Conditions P_1-P_5 are stated as follows. The notation $\alpha = \varepsilon$ for the Laplace penalty and $\alpha = \gamma$ for the arctan penalty.

(P_1) $p_\alpha(0) = 0$ and $p_\alpha(t)$ is symmetric around zero. (P_2) $p_\alpha(t)$ is increasing for $t \in [0, \infty)$. (P_3) For $t > 0$, the function $u_\alpha(t) = p_\alpha(t)/t$ is non-increasing in t. (P_4) The function p_α is differentiable for all $t \neq 0$ and subdifferentiable at $t = 0$, with $\lim_{t\to 0+} p'_\alpha(t) = L$. ($P_5$) There exists $\mu > 0$ such that $p_\alpha(t) + (\mu t^2/2)$ is convex. (P_1) is easy to verify for both Laplace and arctan penalties. Below, we verify the remaining four properties for the two penalty functions. We will need the following two lemmas.

Lemma 3.1. *For $\varepsilon \in (0, 1]$ and $x \geq 0$,*

$$\varepsilon^x \leq \frac{1}{1 - x\log\varepsilon}. \tag{4}$$

Proof. This can be easily proved by letting $\log z = y = -x\log\varepsilon$ and by using the inequality $z - 1 \geq \log z$ for all $z > 0$ (in particular for $z \geq 1$). □

Lemma 3.2. *For $y \geq 0$,*

$$\frac{y}{1 + y^2} \leq \arctan(y) \leq y. \tag{5}$$

(5) is a well-known fact of arctan function. We now verify the remaining properties.

(P_2): In the Laplace case, it is easy to see that $p'_\varepsilon(t) = -\log(\varepsilon)\varepsilon^t$ is positive for $\varepsilon \in (0, 1)$ and $t \geq 0$. In the arctan case, for $t \geq 0$,

$$p'_\gamma(t) = \frac{2\gamma}{\pi}\frac{1}{1 + \gamma^2 t^2}$$

is positive for $\gamma > 0$. Hence $p_\gamma(t)$ is increasing for $t \in [0, \infty)$.

(P_3): In the Laplace case, for $t > 0$, since for $t > 0$,

$$u'_\varepsilon(t) = \frac{t\left[-(\log\varepsilon)\varepsilon^t\right] - \left[1-\varepsilon^t\right]}{t^2} = \frac{\varepsilon^t\left[1 - t\log\varepsilon\right] - 1}{t^2}$$

it suffices to show that the numerator $\varepsilon^t\left[1 - t\log\varepsilon\right] - 1 \leq 0$ for $t > 0$. But this follows from Lemma 3.1 above. In the arctan case, for $t > 0$,

$$u'_\gamma(t) = \frac{2}{\pi t^2}\left[\frac{\gamma t}{1+\gamma^2 t^2} - \arctan(\gamma t)\right].$$

Thus $u'_\gamma(t) \leq 0$ by Lemma 3.2 and hence $p_\gamma(t)/t$ is non-increasing.

(P_4): In the Laplace case, it is easy to see that any point in the interval $[\log\varepsilon, -\log\varepsilon]$ is a subgradient of $p_\varepsilon(t)$ at $t = 0$. For arctan penalty, $\lim_{t\to 0+} p'_\gamma(t) = L$ with $L = 2\gamma/\pi$. Any point in the interval $[-2\gamma/\pi, 2\gamma/\pi]$ is a subgradient of $p_\gamma(t)$ at $t = 0$.

(P_5): $\mu = (\log\varepsilon)^2$ will work in the Laplace case and $\mu = 2\gamma^2/\pi > 0$ will work in the arctan case. This completes verification of the properties P_1-P_5 for Laplace and arctan penalties.

3. Difference-of-convex optimization framework

A framework for nonconvex optimization problems which is more general and less algorithm-specific is the difference-of-convex (DC) programming setting, where the key idea is to write a nonconvex function as the difference of two convex functions [13–15]. Recently, [9] studied regularized optimization problems by exploiting some special structure associated with such problems. [9] focused on the d(irectional)-stationary solutions of the optimization problem because such solutions are sharpest kind among all types of stationary solutions. When a few assumptions made in [9] hold, one of the key propositions proved in [9] (Proposition 3.1) roughly states that a d-stationary point is a unique minimizer of a convex loss function with a nonconvex penalty function appended to it. In order for the above-mentioned proposition from [9] to hold for Laplace and arctan penalties, we verify below that $h_\varepsilon(x) = (-\log\varepsilon)|x| - (1-\varepsilon^{|x|})$ and $h_\gamma(x) = \gamma|x| - \arctan(\gamma|x|)$ are indeed both of the class LC^1. For more details on notation see [9].

Laplace case: We will show the derivative of h_ε is Lipschitz with Lipschitz constant $(\log\varepsilon)^2$.

$$h'_\varepsilon(x) = (-\log\varepsilon)\text{sign}(x)(1-\varepsilon^{|x|}).$$

When $x \geq 0, y \geq 0$:

$$\begin{aligned}
|h'_\varepsilon(x) - h'_\varepsilon(y)| &= (-\log\varepsilon)|\varepsilon^x - \varepsilon^y| \\
&= (\log\varepsilon)^2|(x-y)\varepsilon^u|,\ u \in (\min\{x,y\}, \max\{x,y\}) \\
&\leq (\log\varepsilon)^2|x-y|,\ \text{since } |\varepsilon^u| \leq 1.
\end{aligned}$$

When $x \geq 0, y \leq 0$ ($z = -y$, say):

$$\begin{aligned}
|h'_\varepsilon(x) - h'_\varepsilon(y)| &= |\log\varepsilon||(\varepsilon^x - 1) + (\varepsilon^z - 1)| \\
&= |\log\varepsilon||x(\log\varepsilon)\varepsilon^{u_1} + z(\log\varepsilon)\varepsilon^{u_2}|,\ u_1 \in (0,x),\ u_2 \in (0,z), \\
&= (\log\varepsilon)^2\left(x\varepsilon^{u_1} + z\varepsilon^{u_2}\right) \\
&\leq (\log\varepsilon)^2(x+z),\ \text{since } x \geq 0, z \geq 0,\ \text{and } 0 \leq \varepsilon^{u_1} \leq 1, 0 \leq \varepsilon^{u_2} \leq 1 \\
&= (\log\varepsilon)^2|x+z|,\ \text{since } x+z \geq 0 \\
&= (\log\varepsilon)^2|x-y|,\ \text{since } z = -y.
\end{aligned}$$

Remaining cases $x \leq 0, y \geq 0$ and $x \leq 0, y \leq 0$ can be proved similarly.

arctan case: We will show the derivative of h_γ is Lipschitz with Lipschitz constant γ^2.

$$h'_\gamma(x) = \gamma\,\text{sign}(x)\left(1 - \frac{1}{1+\gamma^2x^2}\right) = \gamma\,\text{sign}(x)\frac{\gamma^2x^2}{1+\gamma^2x^2}.$$

When $x \geq 0, y \geq 0$:

$$h'_\gamma(x) - h'_\gamma(y) = \gamma\left[\frac{\gamma^2x^2 - \gamma^2y^2}{(1+\gamma^2x^2)(1+\gamma^2y^2)}\right] = \gamma^2(x-y)\left[\frac{\gamma x + \gamma y}{(1+\gamma^2x^2)(1+\gamma^2y^2)}\right].$$

Hence

$$|h'_\gamma(x) - h'_\gamma(y)| = \gamma^2|x-y|\left[\frac{\gamma x}{(1+\gamma^2x^2)(1+\gamma^2y^2)} + \frac{\gamma y}{(1+\gamma^2x^2)(1+\gamma^2y^2)}\right]$$
$$\leq \gamma^2|x-y|\left[\frac{1}{2}+\frac{1}{2}\right] = \gamma^2|x-y|.$$

When $x \geq 0, y \leq 0$:

$$h'_\gamma(x) - h'_\gamma(y) = \gamma\left[\frac{\gamma^2x^2}{1+\gamma^2x^2} + \frac{\gamma^2y^2}{1+\gamma^2y^2}\right]$$

Hence

$$|h'_\gamma(x) - h'_\gamma(y)| = \gamma^2\left||x|\left(\frac{\gamma|x|}{1+\gamma^2|x|^2}\right) + |y|\left(\frac{\gamma|y|}{1+\gamma^2|y|^2}\right)\right|$$
$$\leq \frac{\gamma^2}{2}\,||x|+|y|| = \frac{\gamma^2}{2}|x-y| \leq \gamma^2|x-y|.$$

Proof for other cases follows similarly. The above verifications show that Laplace and arctan penalties fit within the difference-of-convex penalties considered in [9].

4. Empirical study with deep neural network models

In this section we empirically evaluate the performance of the new penalties in deep neural network setting for which the loss function is known to be nonconvex [10]. Although there are a few other nonconvex penalties existing in the literature, we focused on only comparing with SCAD and MCP in the empirical study because none of the other penalties satisfied all regularity conditions (P_1-P_5) presented in [6]. We assessed the performance of regularized deep neural networks with the nonconvex penalty functions presented in this paper, by applying them on seven datasets (MNIST, FMNIST, RCV1, CIFAR-10, CIFAR-100, SVHN and ImageNet). We used convolutional neural networks with nonconvex regularization on four of the seven datasets: CIFAR-10, CIFAR-100, SVHN and ImageNet. More details about the datasets may be found in arXiv version of the paper [16].[2,3]

The optimal weights of the fitted deep neural networks were estimated by minimizing the total cross entropy loss function. We used batch gradient descent algorithm with early stopping. To avoid the vanishing/exploding gradients problem, the weights were initialized to values obtained from a normal distribution with mean zero and variance $4/(n_i + n_{(i-1)})$ where n_i is the number of neurons in the ith layer [17,18]. Rectified linear units (ReLU) function was used as the activation function. The training data was randomly split into multiple batches. During each epoch, the gradient descent algorithm was sequentially applied to each of these batches resulting in new weights estimates. At the end of each epoch, the total validation loss was calculated using the validation set. When twenty consecutive epochs failed to improve the total validation loss, the iteration was stopped. The maximum number of epochs was set at 250. The weights estimate that resulted in the lowest total validation loss was selected as the final estimate. Since there was a random aspect to the way the training sets were split into batches, the whole process was repeated three times with seed values 1, 2, and 3. The reported test error rates are the median of the three test error rates obtained using each of these seed values. A triangular learning rate schedule was used because it produced the lowest test error rates [19]. The learning rates varied from a minimum of 0.01 to a maximum of 0.25. For all penalty functions the optimal λ was found by fitting models with logarithmically equidistant values in a grid. We used Python ver. 3.6.7rc2 and TensorFlow ver.1.12.0 for the calculations.

The CNN architecture that we used consisted of three convolutional "blocks" followed by a hidden "block". Each of the three convolutional blocks consisted of a convolutional layer with 96, 128 and 256 filters respectively, kernel size of 5, stride value of 1, and "same" padding, followed by batch normalization with momentum value for the moving average set to 0.9. This was further followed by a ReLU activation layer and finally a max-pooling layer with kernel-size of 3, stride value of 2, and "same" padding. In the hidden block, the data was first flattened and then passed through two fully connected hidden layers of 4096 nodes with ReLU activation function. Finally, the signals were classified into various categories by calculating the sparse softmax cross entropy values between logits and labels. Since MCP did not perform as well as the other penalties in the DNN analyses, we restricted our focus to only b = 5 value for the MCP function, for CNN analyses. All the results from data analyses are summarized in Table 1. *Conclusions:* The results based on new nonconvex penalty functions were comparable to or better than other regularizations in 5 out of the 7 datasets.

[2] We had independently discovered the penalties in Eqs. (1) and (2), only to discover after an earlier version of the paper was posted on arXiv that the penalties had appeared in different guises in previous literature. We heartily acknowledge the credit due to the original discoverers in the previous literature.

[3] The OLS asymptotic theory in this arXiv paper is currently under review as a separate paper in a different journal.

Table 1
Median test error rates at optimal λ.

Penalty function	Datasets, DNN			Datasets, CNN			
	MNIST	FMNIST	RCV1	CIFAR-10	CIFAR-100	SVHN	ImageNet
None	1.87 (0.11)	11.94 (0.13)	14.66 (0.32)	19.58 (0.24)	50.64 (0.25)	6.63 (0.10)	42.85 (0.19)
L_1 (Lasso)	1.24 (0.05)	10.06 (0.21)	12.97 (0.15)	14.71 (0.15)	43.22 (0.36)	5.32 (0.06)	41.32 (0.42)
L_2 (Ridge)	1.23 (0.01)	10.15 (0.02)	13.77 (0.33)	14.32 (0.25)	41.11 (0.61)	5.29 (0.11)	41.64 (0.28)
SCAD (a = 3.7)	1.80 (0.07)	11.45 (0.23)	13.96 (0.03)	18.56 (0.42)	49.50 (0.39)	6.51 (0.04)	42.12 (0.20)
MCP (b = 1.5)	1.60 (0.22)	11.39 (0.15)	13.33 (0.22)	*x*	*x*	*x*	x
MCP (b = 5)	1.67 (0.21)	11.39 (0.12)	14.44 (0.45)	18.47 (0.49)	50.22 (0.52)	6.33 (0.15)	41.99 (0.30)
MCP (b = 20)	1.65 (0.16)	11.33 (0.04)	14.36 (0.37)	*x*	*x*	*x*	*x*
Laplace ($\varepsilon = 10^{-7}$)	1.23 (0.05)	9.98 (0.25)	12.94 (0.21)	18.87 (0.36)	49.79 (0.69)	6.43 (0.14)	42.22 (0.15)
arctan (γ = 1)	1.26 (0.04)	9.87 (0.13)	13.41 (0.12)	14.61 (0.06)	43.58 (0.04)	5.37 (0.05)	40.98 (0.20)
arctan (γ = 100)	1.25 (0.03)	9.93 (0.23)	13.81 (0.32)	14.39 (0.17)	43.33 (0.26)	5.19 (0.09)	42.61 (0.09)

Standard errors are given in parentheses

Codes availability

Sample python codes used for the CNN analysis in the paper are posted on the following github page: 'github.com/mjohn5/dnn_laplace_arctan_regularization'. Different regularizations can be implemented by changing the 'regtype' values within the codes.

Acknowledgments

Majnu John's research was partially funded by the following NIMH, United States of America grants: R01MH117646 (PI: Lencz), R34MH120790 (PI: Birnbaum), R01MH120594 (PI: Kane and Robinson), RF1MH122886 (PI: Barber), R01MH117172 (PI: DeChoudhury).

References

[1] Tibshirani R. Regression shrinkage and selection via the lasso. J R Stat Soc Ser B Stat Methodol 1996;58:267–88.
[2] Hoerl AE, Kennard R. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics 1970;12:55–67.
[3] Frank I, Friedman J. A statistical view of some chemometrics regression tools. Technometrics 1993;35:109–48.
[4] Fan J, Li R. Variable selection via nonconcave penalized likelihood and its oracle properties. J Amer Statist Assoc 2001;96:1348–60.
[5] Zhang CH. Nearly unbiased variable selection under minimax concave penalty. Ann Statist 2010;38:894–942.
[6] Loh P, Wainwright MJ. Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. J Mach Learn Res 2015;16:559–616.
[7] Trzasko J, Manduca A. Highly undersampled magnetic resonance image reconstruction via homotopic L_0-minimization. IEEE Trans Med Imaging 2009;28:106–21.
[8] Wang Y, Zhu L. Variable selection and parameter estimation with the Atan regularization method. J Probab Stat 2016;1–12, Article ID: 6495417.
[9] Ahn M, Pang J-S, Xin J. Difference-of-convex learning: Directional stationarity, optimality, and sparsity. SIAM J Optim 2017;27:1637–65.
[10] Bishop CM. Pattern recognition and machine learning. New York: Springer; 2006.
[11] Wainwright MJ. High-dimensional statistics: a non-asymptotic viewpoint. Cambridge University Press; 2019.
[12] Loh P, Wainwright MJ. High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. Ann Statist 2012;40:1637–64.
[13] Le Thi HA, Pham DT. DC programming and DCA: thirty years of developments. Math Program 2018;69:5–68.
[14] Le Thi HA, Huynh VN, Pham DT. Stochastic difference-of-convex algorithms for solving nonconvex optimization problems. 2019, arXiv eprint, arXiv:1911.04334.
[15] Pham DT, Souad EB. Algorithms for solving a class of nonconvex optimization problems. Methods of subgradients. In: Hiriart-Urruty JB, editor. Fermat days 85: Mathematics for Optimization. North-Holland Mathematics Studies, vol. 129, 1986, p. 249–71.
[16] Vettam S, John M. Regularized deep learning with nonconvex penalties. 2019, arXiv eprint, arXiv:1909.05142.
[17] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. PMLR 2010;9:249–56.
[18] He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. 2015, arXiv eprint, arXiv:1502.01852.
[19] Smith LN. Cyclical learning rates for training neural networks. 2017, arXiv eprint, arXiv:1506.01186v6.