Chapter 9
Meta-learning with Logistic Regression for Multi-classification

Wenfeng Wang, Jingjing Zhang, and Bin Hu

Abstract The current classifiers and basic learners for few-shot meta-learning is based on distance rules and a series of linear classifiers, such as ridge regression, and linear support vector machine. This study introduces a nonlinear basic learner-logistic regression to improve meta-learning through fast convergence in learning downstream tasks and obtaining the global optimal solution. The Woodbury identity is utilized to express our advantages in a small number of samples. This helps to reduce the consumption in the process of matrix operation. The prototype network and residual network are employed as embedding models. The performance on data sets CIFAR-FS, FC100 and MiniImagenet demonstrate the competitiveness of our method.

9.1 Introduction

Meta-learning has been widely used in various fields [1]. Particularly, the model-agnostic meta-learning can be combined into unsupervised learning, few-shot learning and reinforcement learning [2]. These learning systems can adopt tasks to train and test and achieve the objective of meta-learning that minimize the generalization error loss [3–5]. The goal of meta-learning is to learn a function through a set of learning algorithms, as model-agnostic meta-learning which is widely used recently [6]. Maximum likelihood estimation is a method for us to find the maximum value of the log-likelihood function to form an unconstrained optimization problem.

W. Wang (✉) · J. Zhang
Shanghai Institute of Technology, Shanghai 201418, China
e-mail: wangwenfeng@nimte.ac.cn

W. Wang
Interscience Institute of Management and Technology, Bhubaneswar 752054, India

B. Hu
Changsha Normal University, Changsha 410111, China

R. Kountchev et al. (eds.), *New Approaches for Multidimensional Signal Processing*,
Smart Innovation, Systems and Technologies 270,
https://doi.org/10.1007/978-981-16-8558-3_9

In this paper, we also use the way of task training and we mainly focus on the maximum likelihood estimation of our model [3–5]. We mainly use it to update parameters, so that our objective function can find its global optimal solution, which can greatly reduce the training time of the model and the model achieve better training effect within the allowable range. And we adopt residuals network as our embedding model [7, 8].

The goal of the present study is to achieve the stability of the algorithm, minimize the training error in the training process, and at the same time achieve good generalization ability through test. For the parameter trajectories of logistic regression, we mainly form an unconstraint convex optimization problem, it is unlike SVM which adapts a constraint convex optimization problem [4]. We can use iterative reweighted least square method (IRLS) to get the solver of model [5].

9.2 Proposed Method

9.2.1 Problem Formulation

We have mainly undertaken the experiment on two data set—CIFAR-FS and FC100 and experiment on three forms of K ways N shot (5-way-5-shot, 5-way-1-shot, 5-way-2-shot) for classification. On the one hand, our method is mainly divided into two stages. One is the basic learner stage, which is mainly about learning how to calculate the value of w^i completed by logistic regression differentiation. As shown in Fig. 9.1, w^i are the weights of the linear classifier. The second is the meta-learning stage, which needs to improve the learning ability through back propagation error.

We mainly use meta-learning for few-shot learning gradient-based methods, using gradient descent methods to adapt new tasks [9, 10]. Meta-learning enables a few steps of gradient descent to obtain good parameters in parameter space. In logistic regression, the maximum likelihood estimation can be transformed into a minimum unconstrained optimization problem [11]. Meanwhile, logical regression has closed solution like ridge regression [5]. Our method requires a large amount of computation, which requires GPU to calculate the gradient and the solution of the model. As shown in the following Fig. 9.1, we have depicted the overview of our method; it illustrates 1-way 3-shot classification tasks and we adapt logistic regression method as our classifier. The embedding features of the training samples can be learned and obtaining the corresponding weights and testing examples are same. A task is a tuple for fewshot. Finally, the errors are minimized by the meta-learner.

We have traced back to the previous work of the meta-learning framework, explored the convex base learner again, and proposed the base learner [12] of logistic regression. And we compare it with other convex base learners, such as linear SVM and ridge regression.

According to the two components of the previous meta-learning algorithm, namely the base learner and the meta-learner [12], meta-learning is learning to learn, and it

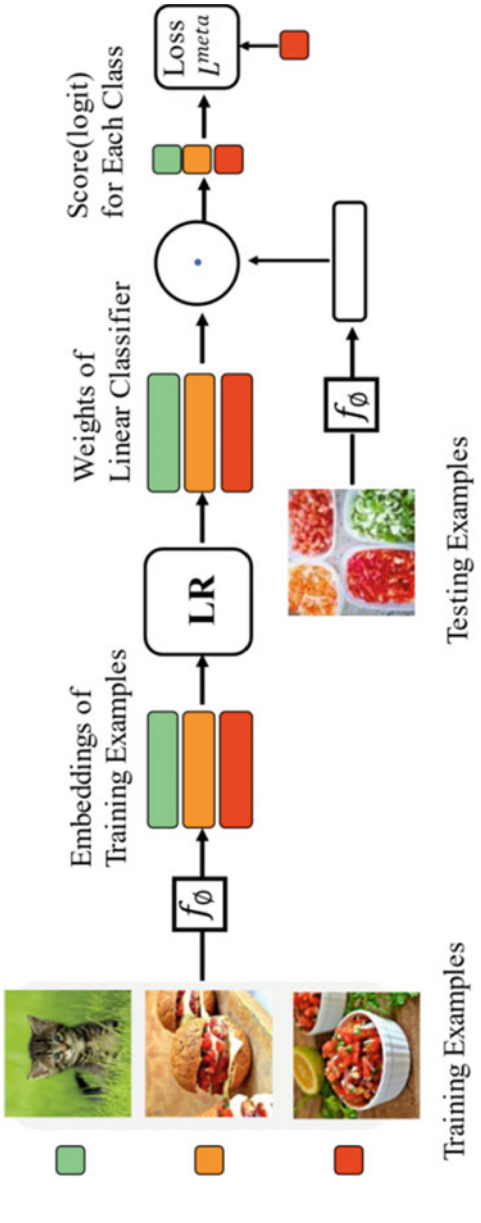

Fig. 9.1 A general overview of our method

is a good way to improve learning skills [13]. The goal of meta-learning is to make the base learning algorithm adapt well to new episodes.

Given a data set $S = \{x_i, y_i\}_{i=1}^{n}$, which includes a meta-training set and a meta-test set, the meta-training set and a meta-test set also include a training set and a test set, but we named it support set and query set. The support set is used for training, and the query set is used for testing so that they construct a task for training. In this paper, there are a group of tasks that is used as a meta-training set $I = \{(D_i^{train}, D_i^{test})\}_{i=1}^{I}$, $D_i^{train} \cap D_i^{test} = \varnothing$. The embedded model is parameterized mainly through $\varnothing$ that mainly uses the support set of the meta-training set. Given J tasks for meta-test $J = \{(D_i^{train}, D_i^{test})\}_{j=1}^{J}$. As we have shown that Fig. 9.2 explains the partition process of data set. The data set is mainly composed of two parts, one is the test set, the other is the training set. At the same time, the test and training set includes support set and query set.

In this paper, the base learner is to estimate the parameter θ of $f(x;\theta)$, here we use the method of university function approximation [14] $y = f(x;\theta)$, and base learner $\mathcal{B}$ is used to achieve better generalization ability. We write it as:

$$\theta = \mathcal{B}\left(D^{train}; \varnothing\right) = \underset{\theta}{argmin}\mathcal{L}^{base}\left(D^{train}; \theta, \varnothing\right) + R(\theta) \tag{9.1}$$

where $\mathcal{L}^{base}$ is the loss function which is computed by the base learner, such as the negative log-likelihood function. As we all know, $R(\theta)$ is a regularization of a function which plays a great important to generalize the loss [15]. As with most meta-learning methods, we regard the training program as episodes, so each episode

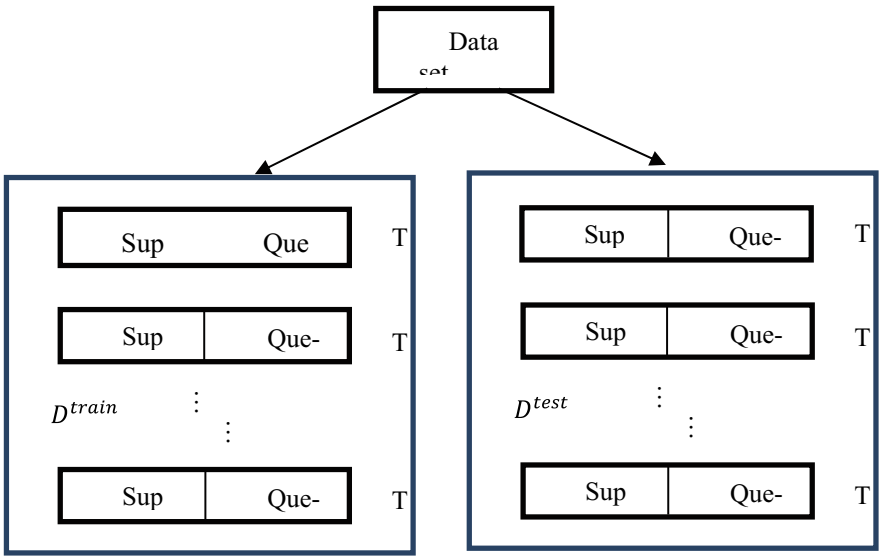

Fig. 9.2 The partition of data set

can be regarded as a small sample classification problem. Usually, the classification of small samples adopts the classification method of K-way and n-shot [16]. Here, we need to consider the values of K and N. Generally, $N = \{1, \ldots, n\}$. In the above, we have described the tasks, a task (or episode) $\top_i = (D_i^{train}, D_i^{test})$. Simultaneously, $D_i^{train} \cap D_i^{test} = \varnothing$ and D_i^{val} also disjoint with them.

9.2.2 *Efficient Logistic Regression Convex Optimization*

The base learner is mainly based on the principle of logistic regression, which is an unconstrained optimization problem. Therefore, we need to discuss the first-order and second-order optimality condition [17, 18], and we first give the unconstrained optimization problem:

$$\theta = \mathcal{B}\left(D^{train}; \varnothing\right) = argmin - \sum_{i=1}^{N} lnp(Y_i|X_i, w_1, \ldots, w_M) + \frac{\lambda}{2} w^T w \tag{9.2}$$

where λ is the regularization and $D^{train} = \{(x_n, y_n)\}$, Y_i is the labels of dataset, $\theta = \{w_k\}_{k=1}^{K}$. Because our objective function is differentiable and convex and there is the quality that if the objective function is continuously differentiable, a practical optimality judgment condition can be obtained by virtue of the property of continuous differentiable function.

Theorem 9.1 (The necessary condition of first order) *If x^* is the local optimal solution of the unconstrained optimization problem* [19], *then* $\nabla f(x^*) = 0$.

Theorem 9.2 (The sufficient condition of second order) *When you suppose that point x^* is the local optimal solution of the unconstrained optimization problem, and if f(x) is continuously differentiable for second order in the neighborhood of point x^*, then*

$$\nabla f\left(x^*\right) = 0 \; and \; \nabla^2 f\left(x^*\right) > 0 \tag{9.3}$$

where $\nabla^2 f(x^*)$ represents Hessian matrix is positive defined, then x^* is a strictly local optimal solution of f(x).

Now we consider the logistic regression multi-class classification problem. Given data have a total of M classes, and each sample x_i corresponds to a vector (or one-hot label) $y_i = [y_{i1}, \ldots, y_{iM}]^T$ of M dimension. Each element of y_i is 0 or 1: If x_i belongs to m-th class, then $y_{im} = 1$, and all other elements are 0. The multinomial logistic regression model uses the following soft-max function as the sample x of the conditional probability belongs to the m class [20].

$$p(y_m = 1|x) = \frac{exp(w_m^T x)}{\sum_{j=1}^{M} exp((w_j^T x))} \tag{9.4}$$

where $w_1, \ldots, w_M$ are the parameters of our model.

We use the following distribution:

$$p(y_m = 1|x) = \sigma\left(w_m^T x\right) = \frac{exp(w_m^T x)}{1 + \sum_{j=1}^{M-1} exp(w_j^T x)}, m = 1, \ldots, M-1 \tag{9.5}$$

$$p(y_M = 1|x) = 1 - \sigma\left(w_m^T x\right) = \frac{1}{1 + \sum_{j=1}^{M-1} exp(w_j^T x)} \tag{9.6}$$

The likelihood function of a single sample is:

$$p(Y_i|X_i, w_1, \ldots, w_M) = \prod_{m=1}^{M} p(y_{im} = 1|x_i)^{y_{im}} \tag{9.7}$$

Therefore, the likelihood function for the meta-training set is:

$$p(Y_i|X_i, w_1, \ldots, w_M) = \prod_{i=1}^{N} \prod_{m=1}^{M} p(y_{im} = 1|x_i)^{y_{im}} \tag{9.8}$$

And we can get the log-likelihood function:

$$lnp(Y_i|X_i, w_1, \ldots, w_M) = \sum_{i=1}^{N} \sum_{m=1}^{M} y_{im} lnp(y_{im} = 1|x_i) \tag{9.9}$$

Newton's-Method and Solving Unconstrained Optimization Problems

Newton's method is a descent method. The difference between Newton's method and gradient descent method lies in the choice of descent direction [21, 22]. For unconstrained optimization problem:

$$min f(x) \tag{9.10}$$

Assuming that f is a convex function and second-order differentiable (the domain is an open set), then the second-order Taylor approximation of f(x) near x is:

$$\widehat{f}(x + v) = f(x) + g(x)^T v + \frac{1}{2} v^T H(x) v \tag{9.11}$$

where $g(x) = \nabla f(x)$ is a gradient, $H(x) = \nabla^2 f(x)$ is a Hessian matrix. Must be noted that the above is only a quadratic approximation, not a complete Taylor expansion.

If x is regarded as a constant, then the above expression is a quadratic function of v, minimized with respect to v, making the gradient zero:

$$g + Hv = 0 \rightarrow v = -H^{-1}g \tag{9.12}$$

It is the Newton step. Since H is positive definite, its inverse is also positive definite,

$$g^T \Delta x_{nt} = -gH^{-1}g \tag{9.13}$$

Unless g = 0, Δx_{nt} is the descent direction. When f is a quadratic function, $x+\Delta x_{nt}$ is its minimum point; As f approaches quadratic, $x + \Delta x_{nt}$ is a good estimate of its minimum point [23]; Since f is quadratic differentiable, the quadratic approximation is very accurate around the minimum value, and $x + \Delta x_{nt}$ is a good estimate of the minimum point [24]. The steps of Newton's method are similar to those of gradient descent, except that the direction of descent is $\Delta x_{nt} = -H^{-1}g$.

There's an objective function (9.2). We should judge whether our goal function is positive definite or not. So let's calculate the gradient:

$$\lambda w + \sum_{i=1}^{N} \frac{-y_i x_i exp(-y_i w^T x_i)}{1 + exp(-y_i w^T x_i)} = \lambda w + \sum_{i=1}^{N} -y_i x_i [1 - \sigma(y_i w^T x)] \tag{9.14}$$

$$g_k = \lambda w_k + \sum_{i=1}^{N} -y_i x_{ik} [1 - \sigma(y_i w^T x_i)] \tag{9.15}$$

where w_l is the lth element of w, and x_{ik} is the kth element of sample x_i, $\sigma(y_i w^T x)$ is sigmoid function. To calculate the Hessian matrix, we need:

$$\frac{\partial \sigma(y_i w^T x_i)}{\partial w_l} = \frac{exp\left(-y_i w^T x_i\right)}{\left[1 + exp\left(-y_i w^T x_i\right)\right]^2}(y_i x_{il}) = \sigma\left(y_i w^T x_i\right)[1 - y_i w^T x_i](y_i x_{il}) \tag{9.16}$$

Let's calculate the elements in k row of the Hessian matrix, k, l = 0, 1…, K. When $k \neq l$,

$$\begin{aligned} H_{kl} = \frac{\partial g_k}{\partial w_l} &= \sum_{i=1}^{N} y_i x_{il} \frac{\sigma\left(y_i w^T x_i\right)}{\partial w_l} \\ &= \sum_{i=1}^{N} \sigma\left(y_i w^T x_i\right)[1 - \sigma\left(y_i w^T x_i\right)(y_i x_{il})(y_i x_{il})] \end{aligned}$$

$$= \sum_{i=1}^{N} \sigma\left(w^T x_i\right)[1 - \sigma\left(w^T x_i\right)] x_{il} x_{ik} \tag{9.17}$$

When $k = l$,

$$H_{kl} = \frac{\partial g_k}{\partial w_l} = \lambda + \sum_{i=1}^{N} \sum_{i=1}^{N} \sigma\left(w^T x_i\right)[1 - \sigma\left(w^T x_i\right)] x_{il} x_{ik} \tag{9.18}$$

Noting the matrix $X = [x_1, x_2, \ldots, x_N]$, $A_{ii} = \sigma\left(w^T x_i\right)\left[1 - \sigma\left(w^T x_i\right)\right]$, the Hessian matrix of (9.2) is

$$H = \lambda I + \sum_{i=1}^{N} \sigma\left(y_i w^T x_i\right)\left[1 - \sigma\left(y_i w^T x_i\right)\right] x_i x_i^T = \lambda I + \sum_{i=1}^{N} A_{ii} x_i x_i^T = \lambda I + X A X^T \tag{9.19}$$

where A is a diagonal matrix of order N, whose elements in i row and i column are A_{ii}, $A_{ii} > 0$.

Because $u^T H u = \lambda u^T u + (X^T u)^T A\left(X^T u\right) > 0, \forall u \neq 0$, so H is positive definite, function (9.2) is a convex function, problem $min - \sum_{i=1}^{N} \ln[1 + \exp(-y_i w^T x_i)] + \frac{\lambda}{2} w^T w$ for unconstrained convex optimization problem.

9.2.3 Approach to the Objective of Meta-learning

When we want to solve unconstrained optimization problems [25], before we do that, we must determine this is a convex optimization problem. The convex function is determined by the Hessian matrix of the objective function $\mathcal{L}^{base}$, for which the Hessian matrix $H = \frac{\partial^2 \theta(w)}{\partial w \partial w^T}$ is positive defined.

$$\begin{aligned} \theta &= \mathcal{B}\left(D^{train}; \varnothing\right) = \underset{\theta}{argmin} \mathcal{L}^{base}\left(D^{train}; \theta, \varnothing\right) + R(\theta) \\ &= argmin - \sum_{i=1}^{N} lnp(Y_i | X_i, w_1, \ldots, w_M) + \frac{\lambda}{2} w^T w \end{aligned} \tag{9.20}$$

We can confirm that the Hessian matrix of our objective function satisfies the condition of the theorem.

And in order to obtain a closed solution, we must consider using an iterative method to solve it. In there we adopt iteratively reweighted least squares (IRLS) method to optimize the problem, the following iteration [26]:

$$w^i = w^{i-1} - H^{-1} g \tag{9.21}$$

H is the Hessian matrix of objective function. The number of Newton steps related to the Hessian matrix can be obtained by the second-order Taylor approximation of the objective function. Among them, the ith iteration updates the parameters

$$H_i = \lambda I + XAX^T, g_i = \lambda w - XAt \tag{9.22}$$

$t_i = \frac{y_i[1-\sigma(y_i w^T x_i)]}{A_i}$, $A = \sigma(w^T X)\left[1 - \sigma(w^T X)\right]$, σ is the sigmoid function, g_i is the gradient. So the formula can be obtained by substituting (9.22) into (9.21) that we can compute:

$$w^i = (XAX^T + \lambda I)^{-1} XAz \tag{9.23}$$

where

$$z = \left(X^T w^{i-1} + t\right) \tag{9.24}$$

$$z_i = X^T w^{i-1} + t_{i-1} = X^T w^{i-1} + \frac{y_i[1 - \sigma(y_i w^T x_i)]}{A_i} \tag{9.25}$$

$$A_i = \sigma(w^T x_i)\left[1 - \sigma(w^T x_i)\right] \tag{9.26}$$

$min - \sum_{i=1}^{N} lnp(Y_i|X_i, w_1, \ldots, w_M)$ also called the cross-entropy error function of logistic regression multi-classification [27].

Although there are many options for measuring losses, here we use a negative log-likelihood function to measure losses, which are same as in the paper of prototype network [28, 29]. The negative log-likelihood function can measure the performance of the meta-test sample, and we think it is very effective way to adopt this function.

$$L^{meta}\left(D^{test}; \theta, \varnothing, \alpha\right) = \sum_{(x,y)\in D^{test}} [-\alpha w^i f_{\varnothing}(x) + log \sum_k exp(\alpha w^j f_{\varnothing}(x))] \tag{9.27}$$

where $\theta = \mathcal{B}\left(D^{train}; \varnothing\right) = \{w^j\}_{j=1}^{K}$ and α is a parameter which can be learned from the process.

9.3 Results and Discussions

In this paper, we mainly use Resnet and prototypical networks as our embedding model. When experiment on the CIFAR and FC100 data set, the network architecture: R64-MP-DB(0.9,1)-R160-MP-DB(0.9,1)-R320-MP-DB(0.9,2)-R640-MP-DB(0.9,2). We initially set the learning rate to 0.1 and change to 0.006 at epoch

Table 9.1 Comparison of other algorithms on CIFAE-FS and FC100. Average few-shot classification accuracies (%) which on the backbone Resnet12. 'R2D2' and Ridge stand for ridge regression but for two different forms. 'LR' stands for the logistic regression

		CIFAR-FS		FC100	
Model	Backbone	1-shot (%)	5-shot (%)	1-shot (%)	2-shot (%)
R2D2	Resnet12	55.52	71.81	31.18	36.91
Ridge	Resnet12	55.19	71.56	31.24	37.03
SVM	Resnet12	55.53	71.33	31.35	37.41
LR	Resnet12	55.60	71.91	31.14	36.78

20. The use of such parameters here is in full compliance with the criteria of gradient descent. We referred to the corresponding parameter settings in the Meta-learning of different- able convex optimization [2]. In order to make full use of the device's availability and available memory space, we tried to set epochs as 20 for many times, which was a wise choice because the GPU often needed to carry out a lot of calculations in the case of many tasks, which would cost a lot of time. The minibatch consists of 8 episodes and every epoch consists of 1000 episodes. And Table 9.1 shows the result of our method and make a comparison to other base learners.

As shown in Table 9.1, LR as our base learner can achieve better performance and be more stable when we use CIFAR-FS data set. As shown in Figs. 9.3 and9.4, we compare four base learners with the same k-way n-shot(5-way 1-shot; 5-way 2-shot; 5-way 5-shot) on CIFAR-FS data set and FC100 data set, MiniImagent data set, it depicts our method can stably get the results. But when we use data set FC100, we find that SVM method will be more efficient to test tasks. In this way, although logistic regression method in FC100 data set doesn't get enough good results but it can confirm that it can be stable for classification. At the same time, it also reflects the authenticity of experiments, the whole operation process is you don't know FC100 data gathering in the effect of the LR algorithm accuracy is lower than the other. It is believed that LR meta-learning has better stability than the other three kinds of algorithms, so it can be as our further exploration work, we can explore that the logistic regression meta-learning algorithm better adapts to all of the downstream tasks. However, when we use MiniImagenet data set to achieve our method, the base learner of SVM becomes the lowest of accuracy in Table 9.2. And LR as the base learner will get 62.48% accuracy with 5-way 5-shot. As shown in Table 9.1, the more samples there are, the higher the accuracy will be. 5-shot means there gives five samples, and 2-shot means there gives only two samples. Therefore, these two samples and five samples will be more accurate than one sample; either a 5-way 10-shot or a 5-way 15-shot (Table 9.2).

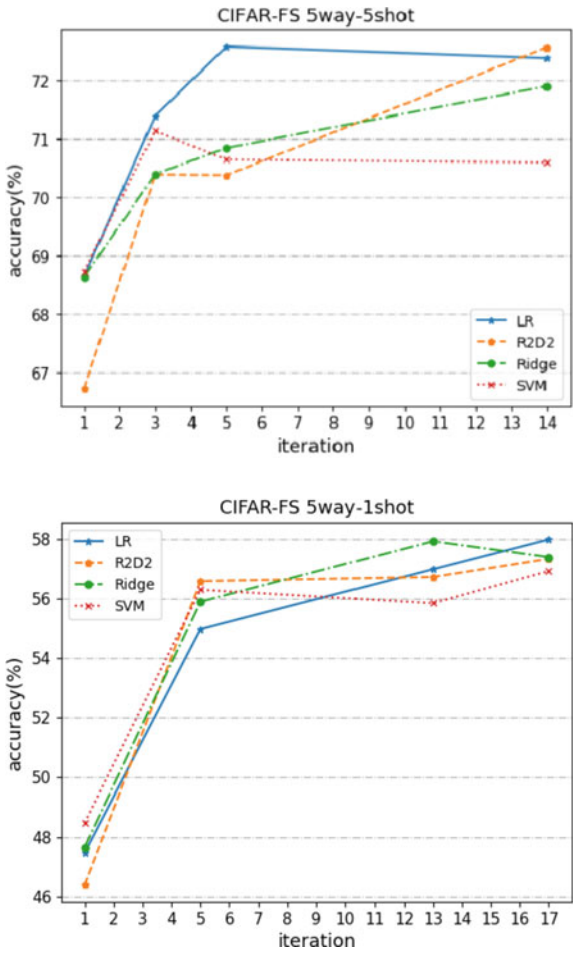

Fig. 9.3 Comparison for four base learners with the same k-way n-shot on CIFAR-FS data set

9.4 Conclusion

In this paper, we mainly show that the performance of logistic regression as the base learner and compare it to other base learners. Our method principally considers the unconstrained optimization problem, and the closed-form solution can be obtained through the iterative method. Moreover, experiments have been carried out on all three data sets, which are fully reflected in the figure above. Finally, we make the conclusion that logistic regression method can stably run than other base learners when there are less epochs as you can see in Figs. 9.3, 9.4, and 9.5. And we just adopt 3 ways to experiment with our convex base learner, it can be seen, our method performs well in CIFAR-FS. At the running level, we further save the time to run our process, because data set is great and the process will be long and complex. It is also an effective way to classification as a base learner after embedding features.

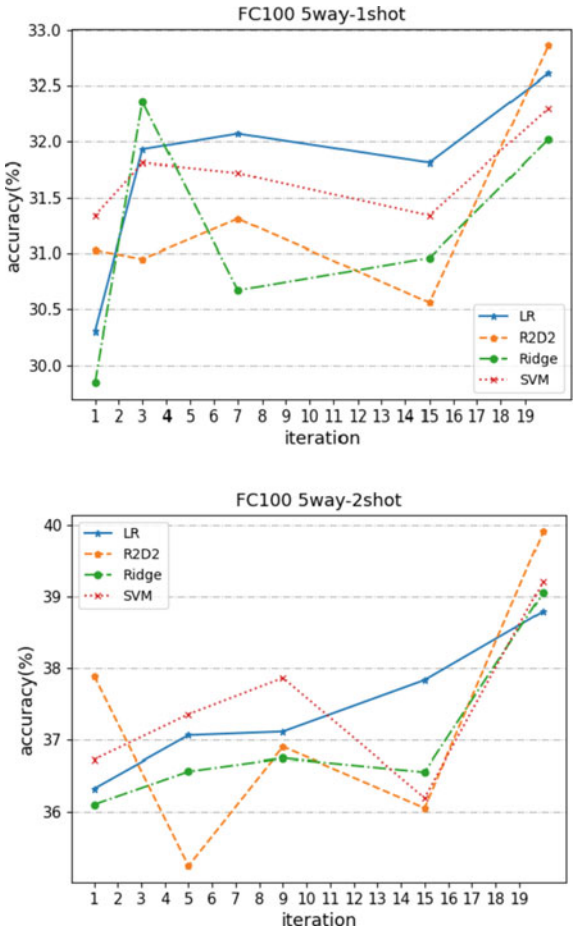

Fig. 9.4 Comparison for four base learners with the same k-way n-shot on FC100 data set

Table 9.2 Comparison of other algorithms on MiniImagenet dataset. Average few-shot classification accuracies (%) which on the backbone 64-64-64-64. 'R2D2' and Ridge stand for ridge regression but for two different forms. 'LR' stands for the logistic regression

	MiniImagenet	
Model	Backbone	5-way 5-shot (%)
R2D2	64-64-64-64	62.38
Ridge	64-64-64-64	62.18
SVM	64-64-64-64	60.59
LR	64-64-64-64	62.48

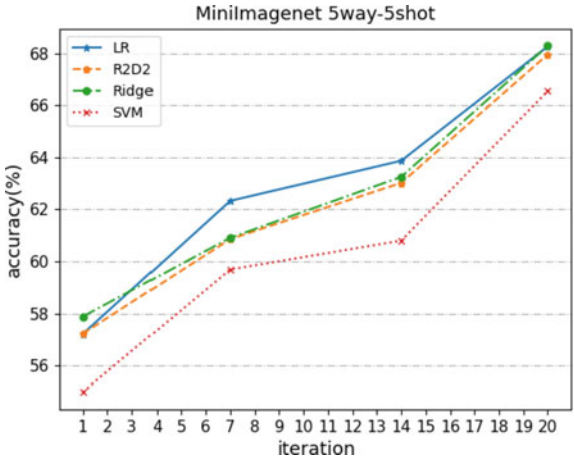

Fig. 9.5 Comparison for four base learners with the same 5-way 5-shot on MiniImagenet data set

Acknowledgements This research was supported by the Shanghai High-Level Base-Building Project for Industrial Technology Innovation (1021GN204005-A06).

References

1. Abramson, N., Braverman, D.J., Sebestyen, G.S.: Pattern recognition and machine learning. Publ. Am. Stat. Assoc. **103**(4), 886–887 (2006)
2. Lee, K., et al.: Meta-learning with differentiable convex optimization. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2019)
3. Menard, S.: Logistic regression. Am. Stat. **58**(4), 364 (2004)
4. Hsu, C.W., Lin, C.J.: A comparison of methods for multiclass support vector machines. IEEE Trans. Neural Netw. **13**(2), 415–425 (2002)
5. Cottle, R.W., Olkin, I.: Closed-form solution of a maximization problem. J. Global Optim. **42**(4), 609–617 (2008)
6. Yuan, K., Ling, Q., Yin, W.: On the convergence of decentralized gradient descent. SIAM J. Optim. **26**(3) (2013)
7. Demiralp, C., Scheidegger, C.E., Kindlmann, G.L., et al.: Visual embedding: a model for visualization. IEEE Comput. Graph. Appl. **34**(1) (2014)
8. Xu, Z., Chen, X., Tang, W., et al.: Meta weight learning via model-agnostic meta-learning. Neurocomputing **432**(7587) (2020)
9. Li, Z., Zhou, F., Fei, C., et al.: Meta-SGD: Learning to Learn Quickly for Few-Shot Learning (2017)
10. Rich, C.: Multitask learning. Mach. Learn. (1997)
11. Silvestre, L.: On the differentiability of the solution to the Hamilton-Jacobi equation with critical fractional diffusion. Adv. Math.-N. Y. **226**(2), 2020–2039 (2009)
12. Bertinetto, L., et al.: Meta-learning with Differentiable Closed-Form Solvers (2018)
13. Chen, Y., Guan, C., Wei, Z., et al.: MetaDelta: A Meta-Learning System for Few-Shot Image Classification (2021)
14. Vilalta, R., Giraud-Carrier, C., Brazdil, P.: Meta-learning. In: Data Mining & Knowledge Discovery Handbook (2005)

15. Bartlett, P.L., Helmbold, D.P., Long, P.M.: Gradient descent with identity initialization efficiently learns positive-definite linear transformations by deep residual networks. Neural Comput. (2019)
16. Dan, L.I., Gao, H.Y., Chen, S., et al.: A proximal gradient method for solving a class of bilevel programming problem. J. Dalian Univ. (2019)
17. Nichol, A., Achiam, J., Schulman, J.: On First-Order Meta-learning Algorithms (2018)
18. Guo-Xun, et al.: A comparison of optimization methods and software for large-scale L1-regularized linear classification. J. Mach. Learn. Res. **11**(11), 3183–3234 (2010)
19. Dontchev, A.L., Rockafellar, R.T.: Solution Mappings for Variational Problems. Springer, New York (2014)
20. Song, T., Song, Y., Wang, Y., et al.: Residual network with dense block. J. Electron. Imaging **27**(PT.2), 053036.1–053036.9 (2018)
21. Stachowiak, M.K.: Cross-entropy method in application to the sirc model. Algorithms **13**(11), 281 (2020)
22. Boyd, S., Crusius, C., Hansson, A.: Advances in convex optimization: theory, algorithms, and applications. IFAC Proc. Vol. **30**(9), 365–393 (1997)
23. Hosmer, D.W., Hosmer, T., Le, C.S., et al.: A comparison of goodness-of-fit tests for the logistic regression model. Stat. Med. **16**(9), 965–980 (2015)
24. Wu, J., Chen, S.P., Liu, X.Y.: Efficient hyperparameter optimization through model-based reinforcement learning. Neurocomputing (2020)
25. Lai, N., Kan, M., Han, C., et al.: Learning to learn adaptive classifier-predictor for few-shot learning. IEEE Trans. Neural Netw. Learn. Syst. (99), 1–13 (2020)
26. Jian, G., Zhang, L., Xiao, X.: Log-Sigmoid nonlinear Lagrange method for nonlinear optimization problems over second-order cones. J. Comput. Appl. Math. **229**(1), 129–144 (2009)
27. Rafi, R., Tang, B., Du, Q., et al.: Attention-based domain adaptation for hyperspectral image classification. In: IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium. IEEE (2019)
28. Pahde, F., Puscas, M., Klein, T., et al.: Multimodal prototypical networks for few-shot learning. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE (2021)
29. Rokach, L.: Ensemble-based classifiers. Artif. Intell. Rev. **33**(1–2), 1–3 (2010)