A Multilayer Framework for Online Metric Learning

Wenbin Li, Yanfang Liu, Jing Huo, Yinghuan Shi, Yang Gao, Senior Member, IEEE, Lei Wang, Senior Member, IEEE, and Jiebo Luo, Fellow, IEEE

Abstract—Online metric learning (OML) has been widely applied in classification and retrieval. It can automatically learn a suitable metric from data by restricting similar instances to be separated from dissimilar instances with a given margin. However, the existing OML algorithms have limited performance in real-world classifications, especially, when data distributions are complex. To address this issue, this article proposes a multilayer framework for OML to capture the nonlinear similarities among instances. Different from the traditional OML, which can only learn one metric space, the proposed multilayer OML (MLOML) takes an OML algorithm as a metric layer and learns multiple hierarchical metric spaces, where each metric layer follows a nonlinear layer for the complicated data distribution. Moreover, the forward propagation (FP) strategy and backward propagation (BP) strategy are employed to train the hierarchical metric layers. To build a metric layer of the proposed MLOML, a new Mahalanobis-based OML (MOML) algorithm is presented based on the passive-aggressive strategy and one-pass triplet construction strategy. Furthermore, in a progressively and nonlinearly learning way, MLOML has a stronger learning ability than traditional OML in the case of limited available training data. To make the learning process more explainable and theoretically guaranteed, theoretical analysis is provided. The proposed MLOML enjoys several nice properties, indeed learns a metric progressively, and performs better on the benchmark datasets. Extensive experiments with different settings have been conducted to verify these properties of the proposed MLOML.

Index Terms—Interpretability, metric layer, nonlinearity, online metric learning (OML), passive-aggressive (PA) strategy.

I. INTRODUCTION

LEARNING a meaningful and quality metric on the original instances is crucial to many classification and retrieval applications. In recent decades, many metric learning methods based on the Mahalanobis distance function and bilinear similarity function have been proposed. Mahalanobis distance-based methods [1], [2], [3], [4], [5], [6], [7], [8] refer to learning a real-valued distance matrix with a symmetric positive semi-definite (PSD) constraint. Bilinear similarity-based methods [9], [10], [11] aim to learn a form of bilinear similarity matrix without the PSD constraint. Moreover, there are two kinds of constraints, i.e., pairwise and triplet constraints, that have been widely used in these metric learning methods. A pairwise constraint consists of two similar or dissimilar instances, while a triplet constraint is of the form \((x, x^+, x^-)\), where instance \(x\) is similar to instance \(x^+\), but is dissimilar to instance \(x^-\).

In many real-world applications, a lot of data is streaming data which is continuously produced in time, such as wind power data [12], credit data [13], and ADs click data [14]. Online learning algorithm investigates how to learn in a streaming setting [15], [16], [17]. Therefore, metric learning algorithms should be able to learn metrics in an online manner, i.e., online metric learning (OML). In fact, multiple OML algorithms have been proposed [9], [18], [19], [20], [21], [22]. However, the existing OML algorithms mainly pay attention to rapid constraints construction [9], [21] or low update complexity [19], [20], [21], while rarely consider the learning ability in the case where all the labeled streaming data cannot be observed. In addition, most of these OML algorithms only learn one linear metric space which cannot learn well-refined metrics for complicated nonlinear data distribution.

To tackle the above limitations of the existing OML algorithms, we propose a multilayer OML (MLOML) framework, which is nonlinear and explainable. In this framework, we attempt to design a metric-algorithm-based layer, which is stacked by several OML algorithms along with the corresponding nonlinear layers (e.g., ReLU, Sigmoid, tanh). In this way, our proposed MLOML is able to learn a progressively refined metric space by learning another new metric in the former learnt feature space (see Fig. 1). Specifically, in MLOML,
OML algorithm, as a result, the parameters of each metric layer in MLOML is a relatively independent layers are repeatedly stacked multiple times. It is worth noting that each metric layer in MLOML is a relatively independent OML algorithm, as a result, the parameters of each metric layer can be innovatively updated according to its own local loss during forward propagation (FP). It means that it is possible to train such a metric-algorithm-based layer by only using the FP strategy. The advantages of this FP updating are: 1) the parameter updating is immediate, unlike the delayed updating of the commonly used backward propagation (BP) and 2) when additional BP is adopted, FP updating can vastly accelerate the convergence. Note that the second advantage has a similar effect of layer-wise unsupervised pretraining [23], [24], [25]. However, there are fundamental differences. The existing layer-wise training is unsupervised and only acts as a pretraining operation (or a regularizer [25]), which is not end-to-end. In contrast, the FP updating in the proposed MLOML is supervised and serves the primary training mode rather than a pretraining role (elaborated in Section IV-D), which is end-to-end. In fact, these two updating strategies (i.e., FP and BP) can be combined to train this metric-algorithm-based layer. Ideally, FP updating can explore new feature spaces sequentially, while BP updating can amend the exploration further.

Furthermore, to achieve a low computational cost when performing MLOML, a new general Mahalanobis-based OML (MOML) algorithm is proposed as the metric layer of MLOML. Since all the labeled streaming data cannot be observed in an online manner, MOML uses the one-pass triplet construction [21] instead of triple constraints obtained in advance. Simultaneously, MOML has a convex objective function inspired by passive-aggressive (PA) learning and enjoys a closed-form solution at each step. We also derive a theoretical regret bound for MOML to prove its convergence. Through stacking MOML hierarchically, the ability to learn feature representation progressively can be explainable and guaranteed.

Our main contributions can be summarized as follows.

1) A MLOML framework is developed for streaming data through FP strategy or BP strategy, such that a metric space is learned progressively and deeply, i.e., exploring and learning a new metric in a nonlinear transformation space sequentially.

2) Taking MOML as a metric layer, MLOML has theoretical guarantees so that the classification performance will be improved or at least well maintained as the depth of the layers increases.

3) MLOML is simple yet effective, as verified by extensive experiments.

**II. RELATED WORK**

OML enjoys several practical and theoretical advantages, making it widely studied and applied in data mining tasks, which can be roughly divided into two categories: 1) Mahalanobis distance-based and 2) bilinear similarity-based methods. In bilinear similarity-based methods, an online algorithm for scalable image similarity (OASIS) [9] is proposed based on the PA algorithm, aiming to learn a similarity metric without PSD constraint. Following a similar setting as OASIS, sparse OML (SOML) [10] learns a diagonal matrix instead of a full matrix to deal with the high-dimensional data. Online multiple kernel similarity (OMKS) [11] has been proposed to handle the multimodal data. Through adopting an off-diagonal $\ell_1$ norm to the similarity matrix, sparse online relative similarity (SORS) [26] can obtain a sparse result. Online similarity learning via low rank and group sparsity (OSLREGS) [27] is designed to address the over-fitting problem for big data by detecting the feature redundant in the metric matrix and constraining the remaining matrix to a low-rank space.

In the second kind of Mahalanobis distance-based method, pseudo-metric online learning algorithm (POLA) [18] introduces the successive projection operation to update a pseudo-metric and map it onto a PSD cone. As an extended version of information-theoretic metric learning online (ITML-Online) [4], LogDet exact gradient online (LEGO) [19] updates a Mahalanobis metric based on LogDet regularization and gradient descent. Regularized distance metric learning (RDML) [20] with appropriate constraints has a provable regret bound. Mirror descent for metric learning (MDML) [28] is a unified approach that updates a Mahalanobis metric by composite objective mirror descent. Bellet and Habrard [29] utilize an adaptation of the notion of algorithmic robustness [30] to derive generalization bounds for metric learning. Low-rank similarity metric learning (LRSML) [31] uses SVD-based projection to solve the challenging high-dimensional learning task, and then employs alternating direction method of multipliers (ADMMs) [32] to optimize the model. Based on the PA algorithm [15], scalable large margin OML (SLMOML) [33] adopts the LogDet divergence to maintain the closeness between two successively learned Mahalanobis matrices, and utilizes the hinge loss to enforce a large margin between relatively dissimilar samples. Fast low-rank metric learning (FLRML) [34] is an unconstrained optimization on the Stiefel manifold to handle datasets with both high dimensions and large numbers of instances. Large-margin distance metric learning (LMDML) [35] employs the principle of margin maximization and stochastic gradient descent (SGD) method to learn the distance metric with PSD constraint. These methods almost assume that the pairwise or triplet constraints can be obtained in advance except RDML, which exactly receives two adjacent samples as a pairwise constraint at each time. In view of adapting the pairwise and triplet constraints to streaming data, Li et al. [21] present a one-pass triplet construction strategy and design OPML and COPML algorithms with low time complexity. By incorporating a smoothed Wasserstein metric distance, evolving metric learning (EML) [36] can handle the instance and feature evolutions simultaneously.
However, the above-mentioned metric algorithms only learn one linear metric space which cannot learn well-refined metrics for complicated nonlinear data distribution. In order to solve this issue, we propose MLOML which is developed based on a newly designed MOML. Compared with the above OML algorithms, MLOML has the following advantages: 1) MLOML is hierarchical and can learn feature representation progressively (i.e., better and better) through FP and BP strategies; 2) MLOML not only has theoretical guarantees by stacking MOML algorithm as its metric layer but also is nonlinear by employing nonlinear functions; and 3) MLOML enjoys a stronger learning ability than traditional OML algorithms with the same amount of data.

III. OUR FRAMEWORK

Our goal is to design a novel MLOML framework for streaming data, which is stacked by metric-algorithm-based layers along with the corresponding nonlinear layers (e.g., ReLU, Sigmoid, tanh). The framework is illustrated in Fig. 2.

A. Multilayer Online Metric Learning

In this section, we propose and explain our MLOML in detail. MLOML is made up of multiple metric layers and nonlinear layers, in which one metric layer is an OML algorithm and one nonlinear layer is ReLU, Sigmoid, or tanh. To ensure the progressively learning ability of MLOML, we should guarantee the convexity of each metric layer, which can easily guarantee the convergence of each layer. Therefore, a new MOML algorithm is designed specifically. MOML has a convex objective function and enjoys a closed-form solution. Moreover, a tight regret bound of MOML is also proved (see Theorem 2).

Specifically, MOML is built on triplet-based constraints \((x, x^+, x^-)\), where instance \(x\) is similar to instance \(x^+\), but is dissimilar to instance \(x^-\), and these triplets can encode the proximity comparison information. Therefore, MLOML is also learnt from triplet constraints. For computational efficiency, a one-pass triplet construction strategy presented by OPML [21] is also employed to construct triplets rapidly which can solve the inability to observe all the streaming data and its labels. In brief, for each new sample, two latest samples from both the same and different classes in the past samples are selected. By using this strategy, triplets can be constructed in an online manner. There are two types of layers in MLOML, which are OML layer and nonlinear layer, where MLOML-r, MLOML-s, and MLOML-t correspond to MLOML with the ReLU, Sigmoid, and tanh layers, respectively. If we design a three-layer MLOML-r model, there should be three OML layers in this model. Moreover, each OML layer is followed by a ReLU layer except the last OML layer (i.e., the third OML layer). This principle is also satisfied for the MLOML-s and MLOML-t models.

A loss layer can also be added, which can give a global adjustment of the entire metric-algorithm-based model via BP. To adequately use the effect of each local metric layer, the local loss is also utilized to update all the former layers (i.e., the loss of the \(i\)th metric layer can be used to update the \(1\)st to the \((i - 1)\)th layers). In this way, vanishing gradient problem can also be alleviated. The novel loss function can be formulated as follows:

\[
\Gamma = \frac{1}{2} \Gamma_{\text{triplet}} + \sum_{i=1}^{n} w_i \Gamma_{\text{local}} + \frac{\lambda}{2} \sum_{i=1}^{n} \| L^i \|_F^2
\]  

where \(\Gamma_{\text{triplet}} = \left(\| x^{(n)}_1 - x^{(n)}_p \|_2^2 + 1 - \| x^{(n)}_1 - x^{(n)}_q \|_2^2 \right)\) indicates the triplet loss of the final output of the model (where \( \| z \|_2 = \max(0, z) \)), \(\Gamma_{\text{local}}\) denotes the local loss of the \(i\)th OML layer [i.e., (4)], and \(\| L^i \|_F^2\) represents the Frobenius norm of parameter matrix \( L^i \), i.e., the transformation matrix learnt in the \(i\)th OML layer. Moreover, \(\lambda\) is the predefined hyper-parameter. While \(w_i\), the weight of the \(i\)th metric layer can be learnt by SGD during the training phase, indicating the importance of each metric layer.
1) New Mahalanobis-Based OML: To build a metric layer of the proposed MLOML, a new OML algorithm named MOML is presented, which can act as a representative of Mahalanobis-based algorithms. Note that, in essence, MLOML can be constructed by other Mahalanobis-based algorithms. However, with MOML as a building component, MLOML enjoys better theoretical properties. The goal of MOML, learnt from triplet constraints, is to learn a Mahalanobis distance function $D$ that satisfies the following large margin constraint,

$$D_M(x, x_q) > D_M(x, x_p) + r \forall x, x_p, x_q \in \mathbb{R}^d$$

(2)

where $x$ and $x_p, x_q$ belong to the same class, while $x$ and $x_q$ come from different classes. $D_M(x_1, x_2) = (x_1 - x_2)^T M (x_1 - x_2)$, where $M \in \mathbb{R}^{d \times d}$ is a PSD parameter matrix. Also, $r$ is the margin. Naturally, the hinge loss (i.e., $\gamma = 1$), can be employed below,

$$\ell(M, (x, x_p, x_q)) = \max(0, 1 + D_M(x, x_p) - D_M(x, x_q)).$$

(3)

In a sequential manner, given a triplet $(x_i, x_p, x_q)$ at the $t$-th time step. Inspired by the PA algorithms (i.e., a family of margin-based online learning algorithms) [15], we design a convex objective function at each time step as follows:

$$\Gamma = \arg\min_{M \succcurlyeq 0} \frac{1}{2} \|M - M_{t-1}\|_F^2 + \gamma \left[ 1 + \operatorname{Tr}(M A_t) \right]_+$$

(4)

where $\| \cdot \|_F$ is Frobenius norm, $[z]_+ = \max(0, z)$ is the hinge loss, $\operatorname{Tr}(\cdot)$ denotes the trace operation and $A_t = (x_t - x_p)(x_t - x_p)^T - (x_t - x_q)(x_t - x_q)^T$. We can easily get that $\Gamma$ is a convex function for $M$, because $\operatorname{Tr}(M A_t)$ is a linear function of $M$ which is convex, the hinge loss function $[1 + z]_+$ is convex (not continuous at $z = -1$), and $\| \cdot \|_F$ and the domain $M \succcurlyeq 0$ are convex too. It can be shown that an optimal solution can be found within the domain $M \succcurlyeq 0$ by properly setting the value of $\gamma$. Thus, we can get the optimal solution of (4) by calculating the gradient $\frac{\partial \Gamma(M)}{\partial M} = 0$,

$$\frac{\partial \Gamma(M)}{\partial M} = \begin{cases} M - M_{t-1} + \gamma A_t = 0, & [z]_+ > 0 \\ M - M_{t-1} = 0, & [z]_+ = 0 \end{cases}$$

(5)

s.t. $M \succcurlyeq 0$.

According to Theorem 1 (presented below), with a proper $\gamma$, the semi-positive definitiveness of $M$ can be guaranteed. Thus, at the $t$-th time step, the parameter matrix $M_t$ can be updated as below,

$$M_t = \begin{cases} M_{t-1} - \gamma A_t, & [z]_+ > 0 \\ M_{t-1}, & [z]_+ = 0 \end{cases}$$

(6)

From (6), we can see that the time complexity of MOML is $O(d^2)$ at each time step. Using MOML as the base metric layer of MLOML has the following advantages: 1) the objective function of MOML is convex and enjoys a closed-form solution, which is beneficial to theoretical analysis and 2) without loss of generality, MOML can act as a representative of Mahalanobis-based OML algorithms.

2) Theoretical Guarantee: Several theoretical guarantees are given for the proposed algorithms. Theorem 1 is a positive-definite guarantee of the parameter matrix $M$ in MOML. Moreover, Theorem 2 presents a regret bound of MOML. Proposition 1 tries to analyze and explain the effectiveness of the proposed framework, i.e., MLOML. All the details of the proofs can be found in the appendix.

**Theorem 1:** Suppose $M_t$ is positive-definite, then $M_{t+1}$ given by the MOML update, i.e., $M_{t+1} = M_t - \gamma A_{t+1}$ is positive definite by properly setting $\gamma$.

**Theorem 2:** Let $(x_1, x_p, x_q), \ldots, (x_T, x_p, x_q)$ be a sequence of triplet constraints where each sample $x_i \in \mathbb{R}^d$ has $\|x_i\|_2 = 1$ for all $t$. Let $M_t \in \mathbb{R}^{d \times d}$ be the solution of MOL at the $t$-th time step, and $U \in \mathbb{R}^{d \times d}$ denotes an arbitrary parameter matrix. By setting $\gamma = (1/\Phi \sqrt{T})$ (where $\Phi \in \mathbb{R}^+$), the regret bound is,

$$R(U, T) = \sum_{t=1}^{T} \ell(M_t) - \sum_{t=1}^{T} \ell(U) \leq \frac{1}{2} \|I - U\|_F^2 + \frac{32}{\Phi^2}. \quad (7)$$

**Proposition 1:** Let $M_1, \ldots, M_n$ be the parameter matrices learnt by each metric layer of MLOML. The subsequent metric layer can learn a feature space, that is, at least as good as the one learnt by the former metric layer. That is, the composite feature space learnt by both $M_1$ and $M_2$ is better than the feature space learnt only by $M_1$ in most cases (i.e., the feature space is more discriminating for classification).

3) Other OML Algorithms: In addition to MOML, other OML algorithms such as LEGO [19], RDML [20] and OPML [21] etc., can also be adapted into the proposed multi-layer framework (namely LEGO-Multi, RDML-Multi and OPML-Multi).

It is worth mentioning that both LEGO and RDML learn a Mahalanobis parameter matrix $M$, while OPML just learns a transformation matrix $L$. Hence, OPML doesn’t need an additional matrix decomposition operation (i.e., $M = L^T L$). The experimental results of LEGO-Multi, RDML-Multi, and OPML-Multi will be discussed in Section IV-G.

**B. Forward and Backward Propagation**

The proposed MLOML (i.e., MLOML-r, MLOML-s, MLOML-t) is made up of a series of OML algorithms (i.e., MOML metric layer) and nonlinear functions (i.e., ReLU, Sigmoid, tanh). Then, MLOML attempts to explore a new way to train the metric layer by introducing FP updating. In fact, MLOML can not only be learnt by BP but also be learnt by BP. Moreover, these two strategies can be adopted simultaneously too. During FP, each metric layer can be learnt immediately, through this way, new feature space can be explored sequentially. When BP, the return gradients can be used to fine-tune all the metric layers, amending the feature spaces learnt by the BP.
The comparison between these variations will be shown in Fig. 3.

MLOML is trained from scratch in an end-to-end manner, which is totally different from traditional layer-by-layer training. Note that, MLOML is trained with three different propagation strategies as follows: 1) MLOML-FP, which is only trained by employing FP strategy; 2) MLOML-FBP, which utilizes forward and BP strategies simultaneously. Specifically, a loss layer is added as the last layer to calculate the final loss, where the loss function [i.e., (1)] is adopted; and 3) MLOML-BP is similar to MLOML-FBP, while MLOML-BP only utilizes the final loss to train the entire model without the local losses and without forward updating.

The flowcharts of these three variations can be seen in Fig. 3. The comparison between these variations will be shown in Section IV-D.

IV. EXPERIMENTS

To verify the effectiveness and applicability of the proposed MLOML, we conduct various experiments on the UCI datasets, which include multiple real-world machine learning tasks for which only vectorized features can be accessed, to analyze and interpret the properties of MLOML. First, we introduce the training process of MLOML.

A. Training

We will describe how to train MLOML in detail in this section. Note that, MLOML is trained from scratch in an end-to-end manner, which is totally different from traditional layer-by-layer training.

1) Initialization: Parameter matrix $M_i$ ($i = 1, 2, \ldots, n$) is initialized as an identity matrix. The hyper-parameter $\gamma$ in MOML and the $\lambda$ in the loss layer need to be chosen by cross-validation according to a specific task (3 or 5 layers are usually enough).

2) Forward Propagation: At the $t$-th time step, one triplet $(x_t^{(0)}, x_p^{(0)}, x_q^{(0)})$ is constructed. Then the triplet is fed into the first OML layer, and the current local triplet loss [i.e., (4)] is calculated by using the current metric matrix $M_1$. According to the updating strategy of MOML [i.e., (6)], the metric matrix $M_1$ is updated for the first time. Then, $M_1$ is mathematically decomposed as $L_1^+L_1$. After transformation by using $L_1$, the new triplet $(x_t^{(1)} = L_1x_t^{(0)}, x_p^{(1)} = L_1x_p^{(0)}, x_q^{(1)} = L_1x_q^{(0)})$ is fed into the next ReLU, Sigmoid or tanh layer. In a serial manner, the final output of the last layer is $(x_t^{(n)}, x_p^{(n)}, x_q^{(n)})$. Through the linear(i.e., OML layer) and nonlinear transformation (i.e., ReLU, Sigmoid, or tanh layer), new feature spaces are sequentially explored. At the same time, the metric matrix of each OML layer is also learnt.

3) Backward Propagation (Optional): The final loss is calculated according to (1) by using the output of the last OML layer. By using chain rule, SGD is adopted to update all the decomposed transformation matrix $L_i$ ($i = 1, 2, \ldots, n$). Then each metric matrix $M_i$ ($i = 1, 2, \ldots, n$) can be obtained naturally by $M_i = L_i^+L_i$. Note that all these three samples in a triplet are used to calculate the gradients. Ideally, forward updating can explore new feature spaces, while backward updating can amend the exploration. In this way, that is, exploration with amendment, a much better feature space can be found. In practice, the BP indeed can further slightly improve the feature space learnt by the FP in some cases, but this could also bring additional computation load. As a tradeoff between time and performance, if not specified, we will train the proposed MLOML only by FP, similar to the Deep forest [37]. More details can be seen in Section IV-D.

B. Datasets

We pick twelve commonly used datasets from UCI Machine Learning Repository [38], which vary in dimensionality and size. The details of these twelve datasets can be seen in Table I. The reason for choosing these datasets is that they are all vectorized data and can be representative data for real-world applications. For example, lsat is a real voice rehabilitation treatment dataset. pems contains 15 months worth of daily data that describes the occupancy rate of different car lanes of the San Francisco bay area freeways. Also, ionosphere is real radar data, which is collected by a system in Goose Bay, Labrador.

Classification tasks will be conducted on these datasets. For each dataset, 50% samples are randomly sampled as the training set, and the rest is taken as the test set. Each dataset will be resampled 30 times, and each algorithm will be tested on all these sampled datasets. When the feature dimensionality $d \geq 200$, the $d$-dimensional feature will be reduced to a 100-D feature by principal component analysis (PCA) for easier handling. All datasets are normalized by $\ell_2$ normalization. The error rate is adopted as the evaluation criterion.

C. Comparison With the State of the Art

To evaluate the effectiveness of the family of MLOML (MLOML-r, MLOML-s, and MLOML-t), six state-of-the-art...
### TABLE II
ERROR RATES (MEAN ± STD. DEVIATION) ON THE UCI DATASETS. */◦ INDICATES THAT MLOML-r, MLOML-s, MLOML-t ARE SIGNIFICANTLY BETTER/WORSE THAN THE RESPECTIVE ALGORITHM ACCORDING TO THE \(t\)-TESTS AT 95% SIGNIFICANCE LEVEL. THE STATISTICS OF WIN/TIE/LOSS BETWEEN MLOML-r AND OTHER ALGORITHMS IS ALSO COUNTED.

| Datasets | Euclidean | Batch | Online |
|----------|-----------|-------|--------|
|          |           |       |        |
| Isvt     | 0.369±0.051 | 0.387±0.057 | 0.403±0.098 |
| iris     | 0.038±0.016 | 0.040±0.018 | 0.039±0.020 |
| wine     | 0.218±0.039 | 0.170±0.044 | 0.069±0.212 |
| spect    | 0.354±0.031 | 0.357±0.032 | 0.365±0.031 |
| ionosphere | 0.180±0.017 | 0.157±0.016 | 0.156±0.023 |
| pem      | 0.498±0.033 | 0.402±0.038 | 0.188±0.288 |
| balance  | 0.108±0.013 | 0.088±0.013 | 0.101±0.016 |
| breast   | 0.106±0.012 | 0.107±0.012 | 0.106±0.014 |
| pima     | 0.324±0.018 | 0.326±0.020 | 0.333±0.021 |
| diabetic | 0.343±0.018 | 0.335±0.017 | 0.288±0.018 |
| waveform | 0.195±0.006 | 0.190±0.005 | 0.158±0.006 |
| mlprove  | 0.084±0.005 | 0.037±0.004 | 0.234±0.273 |

| win/tie/loss | 8/4/0 | 7/3/2 | 7/1/4 | 6/3/3 | 7/3/2 | 9/2/1 | 8/3/1 |
|--------------|-------|-------|-------|-------|-------|-------|-------|

| Datasets | OASIS | OPML | SLOMOL | MOMOL | MLOML-t | MLOML-s | MLOML-r |
|----------|-------|------|--------|-------|---------|---------|---------|
| Isvt     | 0.333±0.000 | 0.370±0.051 | 0.369±0.051 | 0.369±0.051 | 0.369±0.054 | 0.369±0.052 | 0.326±0.053 |
| iris     | 0.333±0.000 | 0.035±0.016 | 0.038±0.016 | 0.028±0.018 | 0.027±0.015 | 0.025±0.017 | 0.026±0.017 |
| wine     | 0.586±0.061 | 0.220±0.040 | 0.225±0.040 | 0.226±0.041 | 0.214±0.038 | 0.216±0.039 | 0.219±0.039 |
| spect    | 0.385±0.033 | 0.321±0.029 | 0.355±0.032 | 0.331±0.032 | 0.323±0.028 | 0.317±0.024 | 0.320±0.025 |
| ionosphere | 0.181±0.017 | 0.107±0.020 | 0.107±0.020 | 0.108±0.032 | 0.102±0.021 | 0.086±0.013 | 0.081±0.016 |
| pem      | 0.651±0.036 | 0.331±0.029 | 0.495±0.033 | 0.416±0.057 | 0.319±0.032 | 0.407±0.030 | 0.397±0.036 |
| balance  | 0.125±0.010 | 0.073±0.012 | 0.077±0.012 | 0.070±0.010 | 0.064±0.013 | 0.066±0.011 | 0.066±0.012 |
| breast   | 0.175±0.050 | 0.109±0.014 | 0.106±0.012 | 0.112±0.014 | 0.106±0.013 | 0.104±0.013 | 0.105±0.012 |
| pima     | 0.349±0.003 | 0.324±0.022 | 0.334±0.022 | 0.323±0.020 | 0.322±0.019 | 0.321±0.017 | 0.323±0.018 |
| diabetic | 0.451±0.021 | 0.322±0.019 | 0.348±0.016 | 0.342±0.016 | 0.341±0.015 | 0.341±0.014 | 0.342±0.017 |
| waveform | 0.298±0.049 | 0.175±0.006 | 0.161±0.005 | 0.173±0.006 | 0.168±0.006 | 0.162±0.005 | 0.187±0.006 |
| mlprove  | 0.002±0.001 | 0.006±0.002 | 0.011±0.004 | 0.001±0.001 | 0.002±0.001 | 0.002±0.001 | 0.004±0.001 |

| win/tie/loss | 10/1/1 | 6/3/3 | 9/2/1 | 6/4/2 | 2/6/4 | 2/8/2 |
|--------------|--------|-------|-------|-------|-------|-------|

OML algorithms, i.e., RDML [20], LEGO [19], OASIS [9], OPML [21], SLOMOL [33] and the new designed MOML are employed as comparisons. Note that all of these compared OML algorithms are single-layer algorithms, while the proposed family of MLOML (MLOML-r, MLOML-s, and MLOML-t) are built on MOML is a multilayer algorithm. Euclidean distance is adopted as the baseline algorithm. Besides, four batch metric learning algorithms, i.e., LMNN [2], KISSME [39], LMDML [35] and ML-CC [40] are also employed for reference. Note that these three algorithms (i.e., Euclidean, LMNN, and KISSME) are offline.

Cross-validation is used for hyper-parameter selection for all algorithms. Specifically, the regularization parameter \(\gamma\) for the family of MLOML (i.e., the \(\gamma\) in MOML metric layer, \(\gamma \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}\)), the learning rate \(\lambda\) for RDML (\(\lambda \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}\)), the regularization parameter \(\eta\) for LEGO (\(\eta \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}\)), the regularization parameter \(\gamma\) for OPML (\(\gamma \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}\)), the weighting parameter \(\mu\) for LMNN (\(\mu \in \{0.125, 0.25, 0.5\}\)), the parameters \(K\) and \(\mu\) for ML-CC (\(K \in \{2, 4, 8\}\) and \(\mu \in \{0.1, 0.3, 0.5, 0.7, 0.9\}\), and the aggressiveness parameter \(C\) for SLOMOL (\(C \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}\)) are all set up in this way.

For a fair comparison, all OML algorithms adopt the same triplet construction strategy introduced by OPML to construct the pairwise or triplet constraints. The difference is that in OPML the triplet construction strategy is one-pass, while, here, multiple-scan strategy is employed to construct more constraints for adequately training (the scanning number is set to 20). Note that, all OML algorithms are still trained in an online manner. Moreover, three metric layers MLOML is adopted in this experiment. A \(k\)-NN classifier (i.e., \(k = 5\)) is used to get the final classification results. The results are summarized in Table II. For each dataset, the mean and standard deviation of error rate are calculated, and pairwise \(t\)-tests between MLOML and other algorithms at 95% significance.
level are also performed. Then the win/tie/loss is counted according to the \( t \)-test. From this table, we can see that the family of MLOML can not only achieve superior performance compared with other state-of-the-art OML algorithms but also better than batch metric learning algorithms except KISSME algorithm. A possible reason is that KISSME learns a distance metric from equivalence constraints which is easier to specify labels. We can also see that MLOML is robust on small datasets, e.g., lsvt, iris, spect, and ionosphere, which means that MLOML can handle small-scale data very well.

The traditional OML algorithms, RDML, LEGO, OASIS, OPML, and SLMOML algorithms, like our proposed MOML algorithm, can be used as metric layers for MLOML algorithms. However, in the literature of the RDML, LEGO, OASIS and SLMOML algorithms need to first sample triplets, and then apply these triplets to the training process. Because of this, their running time is relatively long. In order to guarantee the fairness of the comparison, Fig. 4 shows the running time of OPML and MOML on the spect and waveform datasets, where OPML and MOML use the one-pass triplet construction strategy. We can see that the proposed MOML has a shorter running time than OPML.

**D. Forward and Backward Propagation**

In this section, we analyze the learning ability of MLOML by adopting different propagation strategies, i.e., MLOML-FP, MLOML-BP, and MLOML-FBP. Specifically, we conduct classification tasks on the twelve UCI datasets to compare these three variations of MLOML, each of which contains three metric layers. The results are exhibited in Table III. From the results, we can see that MLOML-FP performs better than MLOML-BP. The reason is not difficult to perceive, because BP may suffer from the vanishing gradient problem. Taking advantage of the fact that each metric layer of MLOML is a MOML algorithm, it can learn a good metric in each layer during FP. We can also observe that MLOML-FP performs similarly to MLOML-FBP. The reason may be that MLOML-FP has achieved a quite good classification performance on some datasets, so additional BP updating cannot further improve the performance. However, on other datasets, MLOML-FBP indeed achieves the best classification performance as expected, such as iris, spect and mlprove etc. It is worth mentioning that MLOML-FP is the fastest one among these three variations with time complexity of \( O(n^2) \), where \( n \) is the number of metric layers. Overall, for the proposed MLOML, the FP training strategy is the best one when considering both training performance and training efficiency. It should be noted that in the following chapters, MLOML refers specifically to the MLOML-r model, and the three-layer and five-layer MLOML networks are denoted by MLOML-3L and MLOML-5L.

**E. Progressive Feature Representation**

In this section, we will analyze the progressive feature representation ability of each metric layer in MLOML and verify that the metric space can become better and better by adding a metric layer gradually. Particularly, an MLOML-5L model is employed. To test the feature representation ability of each metric layer, we perform classification tasks on the output features of each metric layer, respectively. We choose nine UCI datasets and take Euclidean distance, MOML and LMNN as the baseline algorithms. Note that only the test sets of these datasets are used to perform this experiment. From Fig. 5, we can see that the classification performance of
MLOML-5L becomes better with the increase in the number of metric layers. Besides, in some datasets, the curve of error rate can converge smoothly. Moreover, we apply PCA to four UCI datasets to obtain the new space with 2-D features, and then visualize the feature space learnt by each metric layer for more intuition (shown in Fig. 6). The four UCI datasets are

- \textit{Iris}
- \textit{Ionosphere}
- \textit{Pima}
- \textit{Spect}

Fig. 5. Results of different metric layers of MLOML. Moreover, Euclidean, MOML, and LMNN are taken as the baseline algorithms.

Fig. 6. Feature visualization on four UCI datasets, demonstrating the feature representation learnt by each metric layer in MLOML-3L.
Fig. 7. Error rates on nine UCI datasets by changing the number of scans for MOML and MLOML.

Fig. 8. Results of different metric layers of RDML-multi, LEGO-multi, and OPML-multi (marked by hollow shapes), which are stacked by online metric algorithms RDML, LEGO, and OPML (marked by corresponding solid shapes) along with the ReLU layers, respectively.

datasets are picked and entered into one learnt MLOML-3L model. Next, all output samples of each metric layer are $\ell_2$ normalized and reduced to a 2-D space by PCA. As seen, in the original feature space, the distribution of samples is disordered. As the number of metric layers increases, the intraclass distance becomes smaller, the interclass distance
effectiveness and extendability of the proposed framework are
versions of these algorithms perform better than their corre-
see that LEGO-multi, RDML-multi, and OPML-multi have
strategy is employed for these three algorithms. Other settings
work, we take the other three OML algorithms (e.g., LEGO,
F. Learning Ability of MLOML
Since the multiple-scan strategy is performed in the training
phase, it is necessary to test the learning ability of MLOML by
setting different numbers of scans. Note that \( m \) times scanning
will scan the training data \( m \) times. Therefore, we set the
number of scans from 1 to 20, and compare the classification
performance between MLOML and MOML under different
scans. Specifically, nine datasets are picked, and Euclidean
distance is taken as the baseline. The results are presented in
Fig. 7. From the figure, we can see that as the number of
scans increases, the classification performance of MLOML is
significantly improved and then converges, which can reflect
the ability of MLOML for reusing data. Compared with
MOML, with the same amount of data (i.e., the same scan),
MLOML can learn better feature representation (i.e., lower
error rate). In other words, the learning ability of MLOML is
stronger than MOML, which means that MLOML can gain
more learning ability from the multilayer architecture.

G. Extendability of MLOML
In order to verify the extendability of the proposed frame-
work, we take the other three OML algorithms (e.g., LEGO,
ReLU, and OPML) as the base OML layer followed by
the ReLU layer and construct their corresponding multi-
layer versions, respectively (i.e., LEGO-multi, RDML-multi,
and OPML-multi). Note that these three algorithms are all
Mahalanobis-based OML algorithms. For simplicity, the FP
strategy is employed for these three algorithms. Other settings
are similar to the ones in Section IV-E. From Fig. 8, we can see
that LEGO-multi, RDML-multi, and OPML-multi have similar characteristics to MLOML. In most cases, multilayer
versions of these algorithms perform better than their corre-
spending shallow versions. Moreover, the progressive learning
ability of feature representation is demonstrated. Therefore, the
effectiveness and extendability of the proposed framework are
verified.

V. CONCLUSION
In this study, we propose a multilayer framework for OML.
Specifically, we implement MLOML by stacking a set of
OML algorithms. Extensive experiments have been conducted
to analyze and verify the properties of MLOML. For future
work, we will analyze and discuss this framework from three
aspects as follows.

1) Extendability: Although only OML-based algorithms are
implemented (e.g., MLOML), the proposed framework is
tenable, such as: a) mini-batch or batch metric
learning-based metric layer can be constructed and b) different metric learning algorithms can be combined as
different metric layers.

2) Advantages: The proposed MLOML has many nice
properties: a) it is online; b) it can be trained by either
forward or backward propagation; and c) it is quite fast
and effective, which can be trained by CPU; d) it can
progressively learn feature representation.

3) Drawbacks: Because MLOML is based on MOML, the
performance of MLOML depends on the performance of
MOML. Currently, MLOML cannot efficiently handle
high dimensional data well due to a full matrix \( M \)
learned in MOML. This problem can be tackled by
learning a diagonal matrix or employing dimensionality
reduction through online feature selection, which will be
investigated in the next work. Meanwhile, the number of
metric layers in MLOML is uniformly specified
according to the experimental results. However, the
optimal number of layers is often different for different
tasks. Another question is how many metric layers are
sufficient for a task will be studied in the future.

APPENDIX
A. Proof of Theorem 1
As \( A_{t+1} = (x_{t+1} - x_p)(x_{t+1} - x_q)^\top - (x_{t+1} - x_q)(x_{t+1} - x_p)^\top \),
whose rank is 1 or 2, it has at most 2 nonzero
eigenvalues. That is to say, \( \text{Tr}(A_{t+1}) = \lambda_1 + \lambda_2 \).
Specifically, we can also easily get that,
\[-\| x_{t+1} - x_q \|^2 \leq \lambda(A_{t+1}) \leq \| x_{t+1} - x_p \|^2 \quad (8)\]
where \( \lambda(A_{t+1}) \) means the eigenvalue of \( A_{t+1} \) (i.e., \( \lambda_1 \) or \( \lambda_2 \)).
For each sample \( x \) is \( \ell_2 \) normalized, the ranges of \( \| x_{t+1} - x_p \|^2 \) and
\( \| x_{t+1} - x_q \|^2 \) vary from \([0, 4]\). Thus,
\[\lambda_{\text{min}}(M_t) - 4\gamma \leq \lambda(M_t, \gamma) \leq \lambda_{\text{max}}(M_t) + 4\gamma. \quad (9)\]
When \( \gamma \leq (1/4)\lambda_{\text{min}}(M_t) \), it is guaranteed that the minimum
eigenvalue of \( M_t, \gamma A_{t+1} \) is greater than zero. As the initial
matrix \( M_1 = I \) is positive definite (i.e., \( \lambda_{\text{min}}(M_1) = 1 \)).
By properly setting a small \( \gamma \), the minimum eigenvalue
of \( M_t, \gamma A_{t+1} \) is generally larger than zero. Thus, the positive
definiteness of \( M_{t+1} = M_t, \gamma A_{t+1} \) can be guaranteed.
The same theoretical guarantee (i.e., the small perturbations
of positive-definite matrix) can also be found in chapter
9.6.12 of [41].

B. Proof of Theorem 2
According to the objective function of MOML, i.e.,
\[\Gamma = \arg \min_{M \geq 0} \frac{1}{2} \| M - M_{t-1} \|_F^2 + \gamma \left[ 1 + \text{Tr}(MA_t) \right] \quad (10)\]
we denote \( \ell_t \) as the instantaneous loss suffered by MOML at
each \( t \)-time step with the learnt \( M_t \in \mathbb{R}^{d \times d} \), and denote by \( \ell^e \)
the loss suffered by an arbitrary parameter matrix \( U \in \mathbb{R}^{d \times d} \),
which can be formalized as below,
\[\ell_t = \ell(M_t; (x_t, x_p, x_q)) = \left[ 1 + \text{Tr}(MA_t) \right]_+ \]
\[\ell^e_t = \ell(U; (x_t, x_p, x_q)) = \left[ 1 + \text{Tr}(UA_t) \right]_+ \quad (11)\]
where \( A_t = (x_t - x_p)(x_t - x_q)^\top - (x_t - x_q)(x_t - x_p)^\top \),
\( \text{Tr} \) denotes trace and \( [z]_+ = \max(0, z) \). As \( \text{Tr}(MA_t) \) is a
linear function, it is convex with respect to \( M_t \) by natural.
Besides, the hinge loss function \([z]_+ \) is a convex function (but
not continuous at \( z = 0 \)) with respect to \( z \). Hence, the resulting
composite function $\ell_i(M_t)$ is convex with respect to $M_t$. As $\ell$ is a convex function, we can introduce the first-order condition as follows:

$$
\ell(Y) \geq \ell(X) + \text{VEC}(\nabla \ell(X))^\top \text{VEC}(Y - X)
$$

where $X, Y \in \mathbb{R}^{d \times d}$, VEC denotes vectorization of a matrix, and $\nabla \ell(X)$ is the gradient of function $\ell$ at $X$.

Inspired by [15], we define $\Delta_t$, to be $\|M_t - U\|_F^2 - \|M_{t+1} - U\|_F^2$. Then calculating the cumulative sum of $\Delta_t$ over all $t \in \{1, 2, \ldots, T\}$, we can easily obtain $\sum_t \Delta_t$,

$$
\sum_{t=1}^T \Delta_t = \sum_{t=1}^T \left(\|M_t - U\|_F^2 - \|M_{t+1} - U\|_F^2\right)
$$

$$
= \|M_1 - U\|_F^2 - \|M_{T+1} - U\|_F^2
$$

$$
\leq \|M_1 - U\|_F^2.
$$

Thus,

$$
\sum_{t=1}^T (\ell_t - \ell_t^*) \leq \frac{1}{2\eta} \|M_1 - U\|_F^2 + \frac{\eta}{2} \sum_{t=1}^T \|\nabla \ell(M_t)\|_F^2
$$

$$
= \frac{1}{2\eta} \|M_1 - U\|_F^2 + \frac{\eta}{2} \sum_{t=1}^T \|A_{t+1}\|_F^2
$$

$$
\leq \frac{1}{2\eta} \|M_1 - U\|_F^2 + 32T\eta^2.
$$

In particular, setting $\eta = (1/\sqrt{\Phi})$ (where $\Phi > 0$ is a constant) yields the regret bound $R(U, T) \leq ((\Phi/2)\|I - U\|_F^2 + (32\eta^2/\Phi))\sqrt{T}$. In fact, in this study, as a closed-form solution is employed (i.e., $\eta = 1$), the regret bound is $R(U, T) \leq (1/2)\|I - U\|_F^2 + 32\eta^2$. By setting $\gamma$ in a decreasing way with the iteration number $T$, for example, $\gamma = (1/\sqrt{\Phi})$, we can obtain a regret bound $R(U, T) \leq (1/2)\|I - U\|_F^2 + (32/\Phi^2)$. Hence, proved.

C. Theoretical Analysis of Proposition 1

For simplicity, we just consider analyzing and proving this proposition of MLOML-FP that only uses the FP strategy. In fact, as MLOML-FP only has FP, each metric layer is a relatively independent MOML algorithm. Thus, Theorem 2 is applicable to each metric layer. In other words, each metric layer (i.e., a MOML algorithm) has its own tight regret bound. As the subsequent metric layer is learnt based on the output of the former metric layer, the metric space should not be worse according to the theoretical guarantee of regret bound. Moreover, ReLU, Sigmoid, and tanh activation functions can introduce nonlinear and sparsity into the feature mapping, which is also beneficial to the exploration of feature space. In some cases, if the latter metric layer is in the wrong direction, BP can be chosen to correct and adjust the direction to some extent.

REFERENCES

[1] E. Xing, M. Jordan, S. J. Russell, and A. Ng, “Distance metric learning with application to clustering with side-information,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), vol. 15, 2002, pp. 521–528.

[2] K. Q. Weinberger, J. Blitzer, and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” in Proc. Conf. Neural Inf. Process. Syst. (NIPS), 2005, pp. 1473–1480.

[3] B. Nguyen and B. De Baets, “Kernel-based distance metric learning for supervised K-means clustering,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 10, pp. 3084–3095, Jan. 2019.

[4] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, “Information-theoretic metric learning,” in Proc. Int. Conf. Mach. Learn. (ICML), 2007, pp. 209–216.

[5] S. Xiang, F. Nie, and C. Zhang, “Learning a Mahalanobis distance metric for data clustering and classification,” Pattern Recognit., vol. 41, no. 12, pp. 3600–3612, 2008.

[6] S. Ying, Z. Wen, J. Shi, Y. Peng, J. Peng, and H. Qiao, “Manifold preserving: An intrinsic approach for semisupervised distance metric learning,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 7, pp. 2731–2742, Jul. 2018.

[7] H.-J. Ye, D.-C. Zhan, and Y. Jiang, “Fast generalization rates for distance metric learning,” Mach. Learn., vol. 108, no. 2, pp. 267–295, Feb. 2019.
[8] Y. Liu, B. Du, W. Tu, M. Gong, Y. Guo, and D. Tao, “LogDet metric-based domain adaptation,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 11, pp. 4673–4687, Nov. 2020.

[9] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, “Large scale online learning of image similarity through ranking,” J. Mach. Learn. Res., vol. 11, pp. 1109–1135, Jan. 2010.

[10] X. Gao, S. C. H. Hoi, Y. Zhang, J. Wan, and J. Li, “SOML: Sparse online metric learning with application to image retrieval,” in Proc. AAAI Conf. Artif. Intell., vol. 28, no. 1, Jun. 2014, pp. 1206–1212.

[11] H. Xia, S. C. H. Hoi, R. Jin, and P. Zhao, “Multiple kernel similarity learning for visual search,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 3, pp. 536–549, Mar. 2014.

[12] A. Kavoussi-Fard, A. Khosravi, and S. Nahavandi, “A new fuzzy-based combined prediction interval for wind power forecasting,” IEEE Trans. Power Syst., vol. 31, no. 1, pp. 18–26, Jan. 2016.

[13] E. M. Duhon, “On-line consumer credit data reporting system,” U.S. Patent 6,311,169, Oct. 12, 2001.

[14] X. He et al., “Practical lessons from predicting clicks on Ads at Facebook,” in Proc. Int. Workshop Mining Online Advertising (ADKDD), 2014, pp. 1–9.

[15] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, “Online passive-aggressive algorithms,” J. Mach. Learn. Res., vol. 7, pp. 551–585, Dec. 2006.

[16] X. He et al., “Practical lessons from predicting clicks on Ads at Facebook,” in Proc. Int. Workshop Mining Online Advertising (ADKDD), 2014, pp. 1–9.

[17] H. Zhu, Z. Han, Y.-S. Liu, and M. Gu, “Fast low-rank metric learning for large-scale and high-dimensional data,” in Proc. Adv. Neur. Inf. Process. Syst. (NIPS), 2019, pp. 817–827.

[18] Y. Liu, B. Du, W. Tu, M. Gong, Y. Guo, and D. Tao, “LogDet metric-based domain adaptation,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 11, pp. 4673–4687, Nov. 2020.

[19] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, “Online passive-aggressive algorithms,” J. Mach. Learn. Res., vol. 7, pp. 551–585, Dec. 2006.

[20] H. Xia, S. C. H. Hoi, R. Jin, and P. Zhao, “Multiple kernel similarity learning for visual search,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 3, pp. 536–549, Mar. 2014.

[21] A. Kavoussi-Fard, A. Khosravi, and S. Nahavandi, “A new fuzzy-based combined prediction interval for wind power forecasting,” IEEE Trans. Power Syst., vol. 31, no. 1, pp. 18–26, Jan. 2016.

[22] E. M. Duhon, “On-line consumer credit data reporting system,” U.S. Patent 6,311,169, Oct. 12, 2001.

[23] X. He, J. Zhu, and P. Zhao, “Multiple kernel similarity learning for visual search,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 3, pp. 536–549, Mar. 2014.

[24] A. Kavoussi-Fard, A. Khosravi, and S. Nahavandi, “A new fuzzy-based combined prediction interval for wind power forecasting,” IEEE Trans. Power Syst., vol. 31, no. 1, pp. 18–26, Jan. 2016.

[25] E. M. Duhon, “On-line consumer credit data reporting system,” U.S. Patent 6,311,169, Oct. 12, 2001.

[26] X. He et al., “Practical lessons from predicting clicks on Ads at Facebook,” in Proc. Int. Workshop Mining Online Advertising (ADKDD), 2014, pp. 1–9.

[27] H. Zhu, Z. Han, Y.-S. Liu, and M. Gu, “Fast low-rank metric learning for large-scale and high-dimensional data,” in Proc. Adv. Neur. Inf. Process. Syst. (NIPS), 2019, pp. 817–827.
Yinghuan Shi is currently an Associated Professor with the Department of Computer Science and Technology, Nanjing University, Nanjing, China. He has published more than 40 papers in CCF-A conference and IEEE Transactions. His research interests include machine learning, computer vision, and medical image analysis.

Mr. Shi was awarded as the WuWenJun Al Excellent Young Scientist, the Young Elite Scientists Sponsorship Program by CAST, and the ACM Nanjing Rising Star.

Yang Gao (Senior Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Technology, Nanjing University, Nanjing, China, in 2000.

He is currently a Professor and the Deputy Director of the Department of Computer Science and Technology, Nanjing University, where he is also directing the Reasoning and Learning Research Group. He is currently a Professor with the School of Computing and Information Technology, University of Wollongong, Wollongong, NSW, Australia.

He has authored or coauthored more than 100 papers in top-tier conferences and journals. He has published more than 190 peer-reviewed papers, including those in highly regarded journals and conferences, such as IEEE Transactions on Pattern Analysis and Machine Intelligence, International Journal of Computer Vision, CVPR, ICCV, and ECCV. His research interests include machine learning, pattern recognition, computer vision, artificial intelligence, machine learning, multiagent systems, and intelligent system.

Lei Wang (Senior Member, IEEE) received the Ph.D. degree from Nanyang Technological University, Singapore, in 2004.

Dr. Wang was awarded Early Career Researcher Award by Australian Academy of Science and Australian Research Council. He served as the General Co-Chair of DICTA 2014 and the Area Chair of ICIP2019, and will serve as the Program Co-Chair of ACCV2022 in Macau. He is also the program chair and the area chair for many international conferences.

Jiebo Luo (Fellow, IEEE) is a Professor of Computer Science with the University of Rochester, Rochester, NY, USA, which he joined in 2011 after a prolific career of fifteen years with Kodak Research Laboratories. He has authored over 500 technical papers and holds over 90 U.S. patents. His research interests include computer vision, NLP, machine learning, data mining, computational social science, and digital health.

Mr. Luo has been involved in numerous technical conferences, including serving as the Program Co-Chair of ACM Multimedia 2010, IEEE CVPR 2012, ACM ICMR 2016, and IEEE ICIP 2017, as well as the General Co-Chair of ACM Multimedia 2018 and IEEE ICME 2024. He has served on the editorial boards for the IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Multimedia, IEEE Transactions on Circuits and Systems for Video Technology, IEEE Transactions on Big Data, ACM Transactions on Intelligent Systems and Technology, Pattern Recognition, Knowledge and Information Systems, Machine Vision and Applications, and Intelligent Medicine. He is the current Editor-in-Chief of the IEEE Transactions on Multimedia. He is a fellow of ACM, AAAI, SPIE, and IAPR.