What Makes Objects Similar: A Unified Multi-Metric Learning Approach

Han-Jia Ye, De-Chuan Zhan, Yuan Jiang, and Zhi-Hua Zhou, *Fellow, IEEE*

Abstract—Linkages are essentially determined by similarity measures that may be derived from multiple perspectives. For example, spatial linkages are usually generated based on localities of heterogeneous data. Semantic linkages, however, can come from even more properties, such as different physical meanings behind social relations. Many existing metric learning models focus on spatial linkages but leave the *rich* semantic factors unconsidered. We propose a Unified Multi-Metric Learning (UM^{2}L) framework to exploit *multiple types* of metrics with respect to overdetermined similarities between linkages. In UM^{2}L, types of combination operators are introduced for distance characterization from multiple perspectives, and thus can introduce flexibilities for representing and utilizing both spatial and semantic linkages. Besides, we propose a uniform solver for UM^{2}L, and the theoretical analysis reflects the generalization ability of UM^{2}L as well. Extensive experiments on diverse applications exhibit the superior classification performance and comprehensibility of UM^{2}L. Visualization results also validate its ability to physical meanings discovery.

Index Terms—Distance Metric Learning, Multi-Metric Learning, Similarity measures, Semantic

1 Introduction

Similarities measure the closeness of connections between objects and usually are also reflected by their distances. A single form of distance between objects, e.g., the Euclidean distance, is difficult to adapt for all scenarios. Therefore, Distance Metric Learning (DML) aims to take advantage of side information like linkages and comparisons between objects and learn appropriate metric that can figure out the underlying linkages or connections. With an adaptively learned distance measure, it thus can greatly improve the performance of similarity/distance-based approaches in various domains, such as classification [1], clustering [2], ranking [3], image retrieval [4], and fault localization [5].

DML methods often consider a deterministic metric measuring similarities between all object pairs. Under the supervision of side information, a well-trained distance metric reveals the specific correlation between features of objects and explains the reason underlying their linkages [2], [1], [6]. This single metric, i.e., the global metric, restricts there is only one type of feature correlation to measure similarities, and the same metric is used for measuring all objects no matter how diverse they are. Objects, however, could be linked with each other for different reasons. Recently, investigations on local DML have considered locality specific approaches, and consequently multiple metrics are learned. These metrics are in charge of different spatial areas [7], [8] and deal with heterogeneous data well. Thus, linkages between objects could be generated based on their localities, i.e., objects in the same cluster share a metric pulling similar ones together and pushing dissimilar ones away. It is the spatial locality that differentiates the latent generation among objects. The local consideration can also be extended to more extreme cases, where there exists a metric being responsible for each specific instance [9], [10].

Most existing global and local DML methods emphasize the linkage constraints (including must-link and cannot-link) in localities with univocal semantic meaning. In spite of learning multiple metrics in spatial local areas, there is actually a single metric, may be different in localities, and determines the property of a certain linkage between objects. Explaining similarity between objects in this way only reveals surface supervision of side information while neglecting the rich latent semantics. There can be many different reasons for two objects to be similar in real-world applications [11], [12]. Exploring *underlying reasons* for objects linkages and connections is helpful in various domains [13], [14], [15].

Linkages between objects can be with *multiple latent semantics*, i.e., the connection is determined jointly by more than one reason. As shown in Fig. 1, each of the multiple latent semantics reflects a particular view of objects and can be depicted by a metric. For example, in a social network, friendship linkages may lie on different hobbies of users. Although a user has many friends, their common hobbies could be different and as a consequence, one can be friends with others for different reasons. Thus, given two users, it is hard to justify their friendship from only one perspective. Users' similarity measured based on multiple types of hobbies should be considered together to determine whether there is a linkage between these two users. Another concrete example is, for articles focus on "A. Feature Learning" which are closely related to sub-domains like "B. Feature Selection" and "C. Subspace Models", and their connections are different in semantics. The linkage between A and B emphasizes "picking up some helpful features", while the common semantic between A and C is about "extracting subspaces" or " feature transformation". Although papers with key topic A are similar to each other, there still may exist diversities based on their sub-topics. It is useful to utilize multiple metrics discovering relatedness and difference between these underlying semantics. These phenomena clearly

• *H.-J. Ye, D.-C. Zhan, Y. Jiang and Z.-H. Zhou are with National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210023, China.*
E-mail: {yehj,zhandc,jiangy,zhouzh}@lamda.nju.edu.cn

indicate *ambiguities* rather than a single meaning in linkage generation. Hence, the distance/similarity measurements are *overdetermined* in these applications. As a consequence, a new type of multi-metric learner which can describe the ambiguous linkages is desired.

In this manuscript, we propose a Unified Multi-Metric Learning (UM^{2}L) approach which integrates the consideration of linking semantic ambiguities and localities in one framework. In the training process, more than one metric is learned to measure the distance between instances and each of them reflects a type of inherent spatial or semantic properties of objects. During the test, UM^{2}L can automatically pick up or integrate these measurements since semantically/spatially similar data points have small distances and otherwise they are pushed away from each other; such a mechanism enables the adaptation to complex environments to some degree [16].

In detail, UM^{2}L regards multiple metrics as various spatial or semantic components, and similarities over a pair of objects are generated based on different properties of them, i.e., coming from multiple perspectives. Both types of side information, the direct pairwise comparison or the triplet one, can be handled by variants of UM^{2}L. The overall similarity between two objects is integrated or selected based on these multiple latent values, which are supervised by the provided side information. Therefore, multiple distance metrics learned by UM^{2}L explain objects similar/dissimilar constraints in an overall view, while discovering latent spatial and semantic basics at the same time. Furthermore, the proposed framework can be easily adapted to different types of ambiguous circumstances: by specifying the mechanism of metric integration and selection, various types of linkages in applications can be considered; by incorporating sparse constraints, UM^{2}L also turns out good visualization results reflecting physical meanings of latent linkages between objects, e.g., discovering meaningful subspaces and decomposing combined attributes; besides, by limiting the number of metrics or specifying the regularizer, the approach can degenerate to some popular DML methods, such as MMLMNN [7]. A unified optimization strategy facilitates the learning process of UM^{2}L, and helps solve the general framework steadily and efficiently. UM^{2}L can also utilize the representation ability of deep neural network easily. In addition, a generalization theory of UM^{2}L guarantees the learning ability of the framework.

Our main contributions are three folds. **(I)** We propose a Unified Multi-Metric Learning framework UM^{2}L considering both data localities and ambiguous semantics inside linkages. **(II)** The UM^{2}L framework is flexible and is adaptable for different tasks. All the variants have a unified optimization solution. **(III)** Experimental investigations validate the superiority of UM^{2}L and its interpretability as well.

The rest of this manuscript starts with some notations in Section 2. Then the UM^{2}L framework is presented in detail, including the overall framework, the metric integration specification, and its general solution. In Section 5 and 6, there demonstrates the deep extension and the theoretical generalization analysis on UM^{2}L. Furthermore, we review related work about distance metric learning in Section 7. The last are experiments over various domains and conclusion.

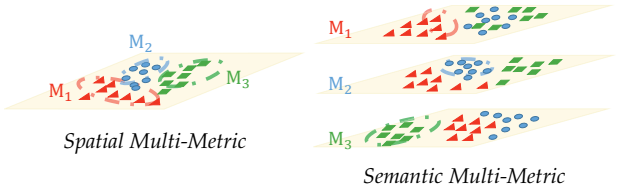


Figure 1: Illustration of the difference between *spatial* and *semantic* learned multiple metrics $\{M_1, M_2, M_3\}$. The left plot shows the multiple metrics with spatial locality, while the right plots presents the semantic multi-metric case. Different colors denote different classes. Dashed lines indicate the contours of learned metrics. Although anchored on different centers with multiple spatial metrics, there is actually only one responsible metric for a particular example in the first case, which has distinct differences from the multiple choices of metric candidates in the semantic multi-metric scenario.

2 Notations and Preliminaries

Given a dataset containing N instances $\mathcal{D} = \{\mathbf{z}_i = (\mathbf{x}_i, y_i)\}_{i=1}^N$, instance $\mathbf{x}_i \in \mathbb{R}^d$ is sampled from a d-dimensional feature space $\mathcal{X}$ while label value y_i is generated from a scalar label space $\mathcal{Y}$, and $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. Generally speaking, the supervision information for Distance Metric Learning (DML) is formed as pairwise constraints or triplet sets. For the former pairwise one, a set $\mathcal{P}$ provides indexes of several similar or dissimilar pairs. An element $\tau = (i, j) \in \mathcal{P}$ can construct a pair. We use $q_{ij} = \mathbb{I}[y_i = y_j] \in \{-1, 1\}$ to denote whether two instances are similar or not, i.e., $q_{ij} = 1$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are in the same class and $q_{ij} = -1$ otherwise. For triplet, a set $\mathcal{T}$ provides more local information. With a slight abuse of notation, each element $\tau = (i, j, l) \in \mathcal{T}$ denotes a relatively comparison between two pairs of instances. In each triplet τ, target instance $\mathbf{x}_j$ is more similar to $\mathbf{x}_i$ than the impostor $\mathbf{x}_l$. We use $(i, j) \sim \mathcal{P}$ and $(i, j, l) \sim \mathcal{T}$ to denote the enumeration over all elements in pairwise and triplet index sets. Cardinalities of $\mathcal{P}$ and $\mathcal{T}$ are P and T.

DML aims at learning a metric $M \in \mathcal{S}_d^+$ making similar instances have small distances to each other and dissimilar ones far apart. The (squared) Mahalanobis distance between a pair $(\mathbf{x}_i, \mathbf{x}_j)$ with metric M can be denoted as:

$$\mathrm{Dis}_M^2(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top M (\mathbf{x}_i - \mathbf{x}_j) = \mathrm{Tr}(M A_{ij}). \quad (1)$$

$A_{ij} = (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top \in \mathcal{S}_d^+$ is the outer product of the difference between instance $\mathbf{x}_i$ and $\mathbf{x}_j$. $\mathcal{S}_d$ and $\mathcal{S}_d^+$ are the sets of symmetric and positive semi-definite (PSD) matrix of size $d \times d$, respectively. I is the identity matrix, whose size can be determined from the context. $\mathrm{Tr}(\cdot)$ outputs the trace of a matrix, which can help present the inner product of two matrices, i.e., $\mathrm{Tr}(A^\top B) = \langle A, B \rangle$. Matrix Frobenius Norm $\|M\|_F = \sqrt{\mathrm{Tr}(M^\top M)}$. Let $\boldsymbol{m}_r$ and $\boldsymbol{m}^c$ denote the r-th row and c-th column of matrix M respectively, and define $\ell_{2,1}$-norm $\|M\|_{2,1} = \sum_{r=1}^d \|\boldsymbol{m}_r\|_2$. Operator $[\cdot]_+ = \max(\cdot, 0)$ preserves the non-negative part of the input value. $\delta[\cdot]$ is the Kronecker delta function, it outputs 1 if the input event is true and 0 otherwise.

3 The Unified Multi-Metric Framework

The distance in Eq.1 assumes that there is a *single* type of relationship between object features, which uses univocal linkages between objects. Multi-metric learning takes data

heterogeneities into consideration. However, both single metric learned by global DML and multiple metrics learned with local methods focus on exploiting locality information, i.e., constraints or metrics are closely related to the localities. In particular, local DML approaches mainly aim at learning a set of multiple metrics and one for each local area. In this manuscript, a general multi-metric configuration is investigated to deal with linkage ambiguities from both semantic and locality perspectives.

We denote the set of K multiple metrics to be learned as $\mathcal{M}_K = \{M_1, M_2, \ldots, M_K\}$ and for any $k = 1, \ldots, K$, $M_k \in \mathcal{S}_d^+$. Similarity score between a pair of instances based on M_k, w.l.o.g., can be set as the negative distance, i.e., $f_{M_k}(\mathbf{x}_i, \mathbf{x}_j) = -\mathrm{Dis}^2_{M_k}(\mathbf{x}_i, \mathbf{x}_j)$. In multi-metric scenario, consequently, there will be a set containing multiple similarity scores $f_{\mathcal{M}_K} = \{f_{M_k}\}_{k=1}^K$. Each metric/score in the set reflects a particular semantic or spatial view of data. For a similar pair $(\mathbf{x}_i, \mathbf{x}_j)$, the *overall* similarity score based on $f_{\mathcal{M}_K}$ is defined as $f_1(\mathbf{x}_i, \mathbf{x}_j) = \kappa_1(f_{\mathcal{M}_K}(\mathbf{x}_i, \mathbf{x}_j))$. $\kappa_1(\cdot)$ is a *functional operator* closely related to concrete applications, which maps the set of similarity scores w.r.t. all metrics to a single value. We use subscript 1 in $f_1(\cdot)$ and $\kappa_1(\cdot)$ to denote the specific overall similarity score and function operator over *similar* pairs. In addition, $f_{-1}(\cdot)$ and $\kappa_{-1}(\cdot)$ are used for *dissimilar* pairs with analogous meanings. Thus, the overall inter-instance similarity f_1 and f_{-1} are based on operators κ_1 and κ_{-1} respectively. Based on previous discussions, the Unified Multi-Metric Learning (UM^{2}L) framework can be formulated. The pairwise version is

$$\min_{\mathcal{M}_K} \frac{1}{P} \sum_{(i,j)\in\mathcal{P}} \ell\left(q_{ij}(f_{q_{ij}}(\mathbf{x}_i, \mathbf{x}_j) - \gamma)\right) + \lambda \sum_{k=1}^{K} \Omega_k(M_k) \ , \quad (2)$$

while the triplet variant is

$$\min_{\mathcal{M}_K} \frac{1}{T} \sum_{(i,j,l)\in\mathcal{T}} \ell\left(f_1(\mathbf{x}_i, \mathbf{x}_j) - f_{-1}(\mathbf{x}_i, \mathbf{x}_l)\right) + \lambda \sum_{k=1}^{K} \Omega_k(M_k) \ . \quad (3)$$

$\ell(\cdot)$ is a non-increasing continuous convex loss function, the larger the input value, the smaller the loss. It encourages similar pair to have larger overall similarity score while the score for dissimilar pair is small. Specifically, when dealing with pairwise constraints in Eq. 2, there is a threshold value γ. The similar pairs should have their overall similarity scores larger than γ and dissimilar pairs have small scores than it. For triplet version of UM^{2}L in Eq. 3, the relative comparisons between similar and dissimilar pairs in a triplet are placed together, i.e., the similar pair has larger overall similarity value than the dissimilar counterpart. This triplet version chooses threshold automatically [17], and incorporates more local information from data. Note that although inter-instance similarities are defined on different metrics in $\mathcal{M}_K$, the convex loss function $\ell(\cdot)$ acts as a bridge and makes the similarities measured by different metrics comparable as in [7]. Regularizer $\Omega_k(M_k)$ imposes prior or structure information on M_k. $\lambda \geq 0$ is a weighting parameter.

The fact that side information restrictions being provided without specifying concrete measurements makes it reasonable to use flexible κs. For instance, in a social network, similar nodes only share some common interests (features) rather than consistently possessing all interests. The tendency on different types of hobbies can be reflected by various metrics. Therefore, the similarity scores may be calculated with different measurements, and operator $\kappa_{\pm 1}$ is used for taking charge of "selecting" or "integrating" the right base metric for measuring similarities. The choices of loss functions and κs are substantial issues in this framework and will be described later concretely.

3.1 Choices for κ

UM^{2}L takes both *spatial* and ambiguous *semantic* linkages into account based on the configurations of κ, which *integrates* or *selects* base metrics.

A simple integrator is the summation over all components, where $\kappa_1 = \kappa_{-1} = \sum$. It treats all metrics in $\mathcal{M}_K$ equally, and this is the general case when there is no prior knowledge of applications. Since the similarities of multiple metric components are summed together, remaining metrics will degenerate to zero to obtain small regularization value if one particular component is good enough. Thus, we can treat this summation/average operation the same as global metric learning. To consider spatial influence in some applications, we can also choose κ to incorporate local impact into multiple metrics. By setting κ as RBF like functions [18], each metric anchors on a cluster center and its influence decreases as its distance to an example increases. Therefore, the locality determines the impact of each metric.

When κ acts as a selector, UM^{2}L should *automatically* assign a certain pair to one of the metrics which can explain instance's similarity/dissimilarity best. Besides, from the aspect of loss function $\ell(\cdot)$, the elected fs form a comparable set of similarity measurements [19], [7]. In this case, we may implement the operator κ by choosing the most remarkable base metric making the pair of instances $\mathbf{x}_i$ and $\mathbf{x}_j$ similar. Advantages of selection mechanism are two folds. First, it reduces the impact of initial pairs or triplets construction in localities [20]; second, it stresses the most evident semantic and reflects the consideration of ambiguous semantics in a linkage construction. Choices of κs heavily depend on concrete applications. It is actually a combiner and can get inspiration from ensemble methods [21]. In the following of the manuscript, we mainly consider 3 different types of linkages based on various sets of κs as follows. It is notable that these three types of similarities can apply to both pairwise and triplet variants of UM^{2}L.

Apical Dominance Similarity (ADS): which is named after the phenomenon in auxanology of plants, where the *most important* term dominates the evaluation. In this case, $\kappa_1 = \kappa_{-1} = \max(\cdot)$, i.e., maximum similarity among all similarities calculated with $\mathcal{M}_K$ on similar pair $(\mathbf{x}_i, \mathbf{x}_j)$ should be larger than the maximum similarity of $(\mathbf{x}_i, \mathbf{x}_l)$ in the triplet case. This corresponds to similar pairs being close to each other under *at least one* measurement, meanwhile dissimilar pairs are disconnected by *all* different measurements. This type of linkage generation often occurs in social network applications, e.g., nodes are linked together for a portion of similar orientations while nodes are unlinked because there are no common interests. By explicitly modeling each node in a social network as an instance, each of the base metrics $\{M_k\}_{k=1}^K$ can represent parts of semantics in linkages. Then the dissimilar pair in a triplet, e.g., the non-friendship relationship, should be with small similarity scores over $\mathcal{M}_K$; while for the similar pair, there should be at least one base similarity score with high value, which reflects their common

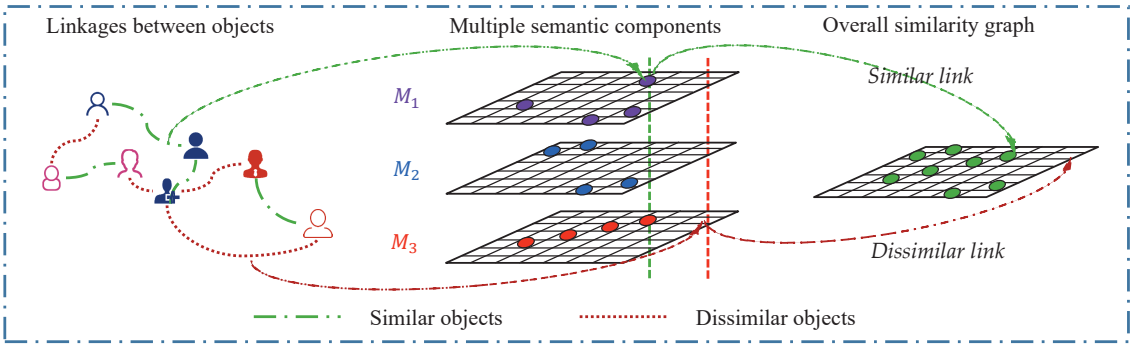


Figure 2: Illustration of UM^{2}L$_{\text{ADS}}$ in a three-metric scenario. The leftmost plot shows a social network where green dot-dashed line and red dashed line represent friends (similar) and non-friends (dissimilar) linkages between users, respectively. We can think each metric in $\mathcal{M}_K$ has a semantic similarity graph, where circles in the squares indicate friendship linkages between users, from a specific view. The rightmost plot corresponds to the overall similarity graph. To illustrate the overall graph, three sub-graphs explain from their own perspectives. They all follow the rule that two users linked in a certain semantic space results in their linkage in the overall graph, and overall dissimilar constraints restrict there are no possible linkages in any decomposed spaces.

interests [11], [13]. An illustration of learning with Apical Dominance Similarity can be found in Fig. 2.

One Vote Similarity (OVS): which indicates the *existence* of potential key metric in $\mathcal{M}_K$, i.e., either similar or dissimilar pair is judged by *at least one* key metric respectively, while remaining metrics with other semantic meanings are ignored. In this case, $\kappa_1 = \max(\cdot)$ and $\kappa_{-1} = \min(\cdot)$. This type of similarity should usually be applied as an "interpreter" in domains like image and video which are with complicated semantics. The learned metrics reveal different latent concepts in objects, and selected ones correspond to those concepts stressed by supervision information. Note that simply applying OVS in UM$_2$L with improper regularizer $\Omega(\cdot)$ will lead to a trivial solution with no generalization ability, i.e., a certain M_k outputs nonzero distance for dissimilar pairs and all other $M_{k'} = 0, k' \neq k$ to satisfy all similar pair restrictions. Therefore, we need to utilize a biased regularizer $\Omega_k(M_k) = \|M_k - I\|_F^2$ or restrict the trace of M_k equals 1.

Rank Grouping Similarity (RGS): which groups the similarity scores and makes the similar scores w.r.t. all metrics with higher ranks than those produced for dissimilar ones. This is the most rigorous similarity, and we also refer it as One-Vote Veto Similarity (OV^{2}S). In this case, $\kappa_1 = \min(\cdot)$ while $\kappa_{-1} = \max(\cdot)$, which regards a certain pair as "dissimilar" even when there is only one metric denying the linkage. This case is usually applied to applications where latent multiple views exist, and different views are measured by different metrics in $\mathcal{M}_K$. In these applications, it is obviously required that all potential views obtain consistencies, and weak conflict detected by one metric should also be punished by RGS (OV^{2}S) loss. The main idea of Rank Grouping Similarity is similar to fusing multi-modal similarities together with consistency [22], but RGS here can work even when there is only one physical modality.

The eventual choice of κs depends on both the similarity property of UM^{2}L variants discussed previously, and the possible linkage generation way in a real task. Numerical analyses and real applications on different κs can be found in the experiments. In addition to these derivatives, $\kappa_{\pm 1}$ in fact can be with richer forms, and we will leave the discussions of choosing different κs later in Section 7. Besides, in the framework, multiple choices of the regularizer $\Omega_k(\cdot)$ can be made. As most DML methods [23], $\Omega_k(M_k)$ can be set as $\|M_k\|_F^2$. Yet it also can be incorporated with more structural information, e.g., we can configure $\Omega(M_k) = \|M_k\|_{2,1}$, where the row/column sparsity filters influential features for composing linkages in a network; or $\Omega_k(M_k) = \text{Tr}(M_k)$, which guarantees the low rank property for all metrics. Due to the high applicability of the proposed framework, we name it as UM^{2}L (Unified Multi-Metric Learning).

4 Unified Solutions for UM^{2}L

In this section, we propose a unified solution for the UM^{2}L framework. UM^{2}L can be solved alternatively between metrics $\mathcal{M}_K$ and affiliation portion of each instance, when $\kappa_{\pm 1}$ are piecewise linear operators such as $\max(\cdot)$ and $\min(\cdot)$. Specifically, given current learned $\mathcal{M}_K$, the metric used to measure the similarity of pair $\tau = (\mathbf{x}_i, \mathbf{x}_j)$ can be determined directly. For example, if $\kappa = \max(\cdot)$, then $k_\tau^* = \arg\max_k f_{M_k}(\mathbf{x}_i, \mathbf{x}_j)$, which is the index of the metric in $\mathcal{M}_K$ that has the largest similarity value over the pair. Once the dominating key metric of each instance is found, the whole optimization problem considers only one active metric for each pair, which can be easily optimized. This two-step optimization is repeated to get the solution.

For facilitating the discussion and practical optimization, we implement $\ell(\cdot)$ as the smooth hinge loss, i.e.,

$$\ell(x) = \begin{cases} 0 & \text{if} \quad x \geq 1 \\ \frac{1}{2}(1-x)^2 & \text{if} \quad 0 \leq x < 1 \\ \frac{1}{2} - x & \text{if} \quad x < 0 \end{cases} .$$

For pairwise variant in Eq. 2, the gradient of loss function w.r.t. a certain $M_k, k = 1, \ldots, K$ can be computed by

$$\begin{aligned} \frac{\partial \ell(\mathcal{M}_K)}{\partial M_k} &= \frac{1}{P} \sum_{\tau=(i,j)\in\mathcal{P}} \frac{\partial \ell(q_{ij}(-\langle M_{k_\tau^*}, A_{ij}\rangle - \gamma))}{\partial M_k} \\ &= \frac{1}{P} \sum_{\tau=(i,j)\in\mathcal{P}} \frac{\partial \ell(a_\tau)}{\partial M_k} = \frac{1}{P} \sum_{\tau=(i,j)\in\mathcal{P}} \nabla_{M_k}^\tau(a_\tau) . \end{aligned} \quad (4)$$

While for triplet version in Eq. 3, we use $k_{1,\tau}^*$ and $k_{-1,\tau}^*$ to denote the affiliation of $\kappa_1(f_{\mathcal{M}_K}(\mathbf{x}_i, \mathbf{x}_j))$ and $\kappa_{-1}(f_{\mathcal{M}_K}(\mathbf{x}_i, \mathbf{x}_l))$ for similar and dissimilar component in

	Pairwise	Triplet	Condition
$\nabla^{\tau}_{M_k}(a_\tau)$	0	0	$a_\tau \geq 1$
	$(1-a_\tau)q_{ij}A_{ij}\delta[k^*_\tau = k]$	$(a_\tau - 1)\left(A_{il}\delta[k^*_{-1,\tau} = k] - A_{ij}\delta[k^*_{1,\tau} = k]\right)$	$0 < a_\tau < 1$
	$q_{ij}A_{ij}\delta[k^*_\tau = k]$	$A_{ij}\delta[k^*_{1,\tau} = k] - A_{il}\delta[k^*_{-1,\tau} = k]$	$a_\tau \leq 0$

Table 1: Gradient computation w.r.t. metric M_k for both pairwise and triplet versions of UM^{2}L. a_τ is the input value of loss function with given pair or triplet, which equals to $q_{ij}(\langle -A_{ij}, M_{k^*_\tau}\rangle - \gamma)$ and $\langle M_{k^*_{-1,\tau}}, A^t_{il}\rangle - \langle M_{k^*_{1,\tau}} A^t_{ij}\rangle$, respectively.

the triplet index $\tau = (i, j, l)$, respectively. Similarly, the gradient of loss function w.r.t. one particular metric M_k can be computed as:

$$\frac{\partial \ell(\mathcal{M}_K)}{\partial M_k} = \frac{1}{T}\sum_{\tau=(i,j,l)\in\mathcal{T}} \frac{\partial \ell(\langle M_{k^*_{-1,\tau}}, A^t_{il}\rangle - \langle M_{k^*_{1,\tau}} A^t_{ij}\rangle)}{\partial M_k} = \frac{1}{T}\sum_{\tau=(i,j,l)\in\mathcal{T}} \frac{\partial \ell(a_\tau)}{\partial M_k} = \frac{1}{T}\sum_{\tau=(i,j,l)\in\mathcal{T}} \nabla^{\tau}_{M_k}(a_\tau)\ . \tag{5}$$

Value of $\nabla^{\tau}_{M_k}(a_\tau)$ for each a_τ, for both pairwise and triplet version in Eq. 4 and Eq. 5 can be found in Table 1.

For smooth regularizer such as squared Frobenius norm $\Omega_k(M_k) = \|M_k\|^2_F$ or trace norm $\Omega_k(M_k) = \mathrm{Tr}(M_k)$, the gradient w.r.t. a particular M_k for the whole objective only has one term appended, i.e., $2\lambda M_k$ or λI. In this case, accelerated projected gradient descent method [24], [25] can solve the metric sub-problem efficiently. After each gradient descent step, a projection step is conducted to maintain the PSD property of each solution.

If structured sparsity is stressed, e.g., $\ell_{2,1}$-norm, $\Omega_k(M_k) = \|M_k\|_{2,1}$, is used as a regularizer, then FISTA [26] can be used to optimize the non-smooth regularizer efficiently. After a gradient descent with step size χ on the smooth loss to get an intermediate solution $V_k = M_k - \chi\frac{\partial \ell(\mathcal{M}_K)}{\partial M_k}$, the following proximal sub-problem is conducted to get a further update:

$$M'_k = \arg\min_{M\in\mathcal{S}_d} \frac{1}{2}\|M - V_k\|^2_F + \lambda\|M\|_{2,1}\ . \tag{6}$$

The PSD property of M_k can be ensured by a projection in each iteration, or can often be preserved by last step projection [27], [23]. Hence, in Eq. 6, only symmetric constraint of M_k is imposed. Since $\ell_{2,1}$-norm considers only one-side (row-wise) property of a matrix, [28] uses iterative symmetric projection to get a solution, which has heavy computational cost in some cases. In a reweighted way, the proximal subproblem can be tackled efficiently[1].

Lemma 1. The proximal problem in Eq. 6 can be solved by updating diagonal matrices D_1 and D_2 with diagonal elements[2]:

$$D_{1,rr} = \frac{1}{2\|\boldsymbol{m}_r\|_2},\ D_{2,cc} = \frac{1}{2\|\boldsymbol{m}^c\|_2},\ r, c = 1, \ldots, d\ ,$$

and the symmetric matrix M

$$vec(M) = (I \otimes (I + \frac{\lambda}{2}D_1) + (\frac{\lambda}{2}D_2 \otimes I))^{-1} vec(V_k)\ ,$$

alternatively, where $vec(M)$ is the vector form of M and $\otimes$ means the Kronecker product. Apart from the closed form update rule of D_1 and D_2, the *diagonal* property of each term can further accelerate the optimization.

1. Detailed derivation and efficiency comparison are in the appendix.

2. There needs a small positive perturbation added on the denominator when the values $\|\boldsymbol{m}_r\|_2$ or $\|\boldsymbol{m}^c\|_2$ approaches zero.

Algorithm 1: The general optimization strategy of UM^{2}L

Require: Training examples and side information constraints. Pre-defined functional operator $\kappa_{\pm 1}$. Parameter λ, initialized $\mathcal{M}_K$ and affiliation.

1: **while** True **do**
2: Optimize over $\mathcal{M}_K$ based on Eq. 4 or Eq. 5.;
3: Do last step PSD projection if needed..
4: **if** Change of objective value in Eq. 2 or Eq. 3 is small enough, or the maximum iteration is achieved **then**
5: Break;
6: **end if**
7: Get metric affiliation for each pair of instances
8: **end while**
9: **return** $\mathcal{M}_K$ (K metrics).

The update of M in Lemma 1 takes row-wise and column-wise ℓ_2-norm into consideration simultaneously, and usually gets converged in about $5 \sim 10$ iterations.

5 Deep Extension of UM^{2}L

The distance metric M in Eq. 1 can be decomposed as $M = LL^\top$. $L \in \mathbb{R}^{d\times d'}$ and $d' \leq d$ is the latent projection dimension. The (squared) Euclidean distance computed after projecting with L equals to the (squared) Mahalanobis distance in the original space. Thus, UM^{2}L can be regarded as learning K linear projections $\mathcal{L}_K = \{L_1, \ldots, L_K\}$.

Benefiting from the strong representation ability of deep neural networks, we can extend the learning process of multiple semantic metrics of UM^{2}L to a *nonlinear* scenario. Assume there is a neural network $h : \mathbb{R}^d \to \mathbb{R}^{d_m}$, mapping a d-dimensional object to a length d_m middle space. Based on the transformed representation, multiple linear weights $\mathcal{L}_K = \{L_k \in \mathbb{R}^{d_m\times d_l}\}_{k=1}^K$ are constructed to project the embedding to multiple different spaces. For a pair of instances $(\mathbf{x}_i, \mathbf{x}_j)$, we compute their distance in the k-th space as $\mathrm{Dis}^2_{L_k}(\mathbf{x}_i, \mathbf{x}_j) = \|L_k h(\mathbf{x}_i) - L_k h(\mathbf{x}_j)\|^2_F$. The similarity value for the pair is computed in a similar way, i.e., combining similarities for multiple semantics by κ, $f_{q_{ij}}(\mathbf{x}_i, \mathbf{x}_j) = \kappa_{q_{ij}}(-\mathrm{Dis}^2_{L_1}(\mathbf{x}_i, \mathbf{x}_j), \ldots, -\mathrm{Dis}^2_{L_K}(\mathbf{x}_i, \mathbf{x}_j))$. Then the loss function is set as that in Eq. 2. Variants of UM^{2}L can be implemented with corresponding κs. The case for triplets can be modeled in a similar way. In the implementation, we use the GoogLeNet [29] to instantiate the function $h(\cdot)$ and then fine-tune on the pre-trained weights.

6 GENERALIZATION ANALYSIS FOR UM^{2}L

In this section, we analyze the generalization ability of UM^{2}L framework theoretically. We focus on the scenario with pairwise constraints and we constraint the regularizer as (squared) Frobenius norm. Considering there are totally $P = N(N-1)$ constraints for each possible combination between different instances, the empirical objective of the UM^{2}L pairwise version in Eq. 2 can be reformulated as:

$$\begin{aligned}&\min_{\mathcal{M}_K} \frac{1}{P}\sum_{i=1}^{N}\sum_{j\neq i}\ell\left(q_{ij}(f_{q_{ij}}(\mathbf{x}_i,\mathbf{x}_j)-\gamma)\right)+\lambda\sum_{k=1}^{K}\|M_k\|_F^2\\ &=\min_{\mathcal{M}_K}\ \epsilon_N(\mathcal{M}_K,\mathcal{D})+\lambda\sum_{k=1}^{K}\|M_k\|_F^2\ .\end{aligned}\tag{7}$$

The empirical objective function above constructs an unbiased estimation of the following expected risk:

$$\begin{aligned}&\min_{\mathcal{M}_K}\mathbb{E}_{\mathbf{x}_1,\mathbf{x}_2}[\ell\left(q_{12}(f_{q_{12}}(\mathbf{x}_1,\mathbf{x}_2)-\gamma)\right)]+\lambda\sum_{k=1}^{K}\|M_k\|_F^2\ .\\ &=\min_{\mathcal{M}_K}\ \epsilon(\mathcal{M}_K,\mathcal{Z})+\lambda\sum_{k=1}^{K}\|M_k\|_F^2\ .\end{aligned}\tag{8}$$

$\epsilon_N(\mathcal{M}_K,\mathcal{D})$ and $\epsilon(\mathcal{M}_K,\mathcal{Z})$ are the empirical and expected loss function of UM^{2}L, and are based on empirical dataset $\mathcal{D}$ and true distribution $\mathcal{Z}$, respectively. We include the symbol $\mathcal{D}$ and $\mathcal{Z}$ when their dependencies are stressed. The goal of generalization analysis is to discover the relationship between $\epsilon_N(\mathcal{M}_K)$ and $\epsilon(\mathcal{M}_K)$.

First, denote $\mathcal{M}_K^*=\{M_1^*,\ldots,M_K^*\}$ as the optimal solution of Eq. 7, and we can find actually that we solve UM^{2}L in a bounded domain. Assume the constant $\ell_u=\max(\ell(\gamma),\ell(-\gamma))$, and use the optimal property of $\mathcal{M}_K^*$ compared with the zero solution, we have $\epsilon_N(\mathcal{M}_K^*)+\lambda\sum_{k=1}^{K}\|M_k^*\|_F^2\leq\epsilon_N(0)\leq\ell_u$. Thus, $\sum_{k=1}^{K}\|M_k^*\|_F^2\leq\frac{\ell_u}{\lambda}$.

Theorem 1. Given $\mathcal{M}_K\in\Gamma=\{\{M_k\in\mathcal{S}_d\}_{k=1}^K,\sum_{k=1}^K\|M_k\|_F^2\leq\frac{\ell_u}{\lambda}\}$, $\|A_{ij}\|_F\leq A$ for all possible i and j, and loss $\ell(\cdot)$ is L-Lipschitz, then with probability at least $1-\zeta$, we have[3]:

$$\epsilon(\mathcal{M}_K)\leq\epsilon_N(\mathcal{M}_K)+\frac{4L\ell_uA}{\lambda\sqrt{N}}+4L\left(\frac{A\ell_u}{\lambda}+\gamma\right)\sqrt{\frac{\log 1/\zeta}{2N}}\ .\tag{9}$$

Our analysis focuses on *multiple metrics* in a single learning objective. From Eq. 9, the difference between empirical and generalization risk has a convergence rate of $\mathcal{O}(\frac{1}{\sqrt{N}})$, which is the same order as the previous global metric learning setting [30], [31]. When the number of examples is large enough, the empirical risk will provide an accurate estimate of the true risk, and the optimal solution of empirical objective will have strong generalization ability, which guarantees the effectiveness of UM^{2}L. Although the number of metrics increases, the norm (complexity) of metrics can also be bound uniformly. It is also noteworthy that when the number of metrics is large, i.e., multiple metrics are learned simultaneously, the increase of freedom in hypothesis space gives rise to stronger representative ability of the predictor, thus gets lower empirical error $\epsilon_N(\mathcal{M}_K)$. The tighter r.h.s. of the bound corroborate UM^{2}L's superiority.

3. In the generalization discussion, we only consider the symmetric property of the given metric as in [30], [31].

Proof: define the supreme of excess loss function $g(\mathbf{z}_1,\ldots,\mathbf{z}_N)=\sup_{\mathcal{M}_K\in\Gamma}\epsilon_N(\mathcal{M}_K)-\epsilon(\mathcal{M}_K)$. Replacing one example $\mathbf{z}_o$ in $\mathcal{D}=\{\mathbf{z}_1,\ldots,\mathbf{z}_N\}$ with $\mathbf{z}_o'$, we get $\mathcal{D}'=\{\mathbf{z}_1,\ldots,\mathbf{z}_{o-1},\mathbf{z}_o',\mathbf{z}_{o+1},\ldots,\mathbf{z}_N\}$. The upper bound of the difference $\sup_{\mathcal{D},z_o'}|g(\mathcal{D})-g(\mathcal{D}')|$ is

$$\begin{aligned}&\sup_{\mathcal{D},z_o'}|g(\mathcal{D})-g(\mathcal{D}')|\\ &\leq\sup_{\mathcal{D},z_o',\mathcal{M}_K}|\epsilon_N(\mathcal{M}_K,\mathcal{D})-\epsilon_N(\mathcal{M}_K,\mathcal{D}')|&(10)\\ &=\sup_{\mathcal{D},z_o',\mathcal{M}_K}|\frac{2}{P}\sum_{i=1,i\neq o}^{N}\ell\left(\mathbf{z}_o,\mathbf{z}_i,\mathcal{M}_K\right)-\ell\left(\mathbf{z}_o',\mathbf{z}_i,\mathcal{M}_K\right)|&(11)\\ &\leq\sup_{\mathcal{D},\mathcal{M}_K}\frac{4L}{P}\sum_{i=1,i\neq o}^{N}|q_{oi}(f_{q_{oi}}(\mathbf{x}_o,\mathbf{x}_i))|+\gamma&(12)\\ &\leq\sup_{\mathcal{D},\mathcal{M}_K}\frac{4L}{P}\sum_{i=1,i\neq o}^{N}|\kappa(\{\langle A_{oi},M_k\rangle\}_{k=1}^K)|+\gamma\\ &\leq\frac{4L(N-1)}{N(N-1)}(A\sup_k\|M_k\|_F+\gamma)\leq\frac{4L}{N}(A\frac{\ell_u}{\lambda}+\gamma)\ .&(13)\end{aligned}$$

Eq. 10 comes from the basic property of $\sup(\cdot)$ operator; Since when a pair $\mathbf{z}_o$ is replaced, only pairs computed with $\mathbf{z}_o$ are affected, so the difference gives rise to Eq. 11; Eq. 12 is based on the Lipschitz property of loss function and equivalance between the original and disturbed similarity value; Since $\kappa(\cdot)$ is $\max(\cdot)$ or $\min(\cdot)$, we can bound the overall similarity value based on the largerest norm of multiple metrics as in Eq. 13. Given the bounded difference condition, together with McDiarmid Inequality [32], then with probability $1-\zeta$,

$$g(\mathcal{D})\leq\mathbb{E}_{\mathcal{D}}[g(\mathcal{D})]+4L(\frac{A\ell_u}{\lambda}+\gamma)\sqrt{\frac{\log 1/\zeta}{2N}}\ .\tag{14}$$

Thus, the remaining goal is to bound $\mathbb{E}_{\mathcal{D}}[g(\mathcal{D})]=\mathbb{E}_{\mathcal{D}}[\sup_{\mathcal{M}_K\in\Gamma}\epsilon_N(\mathcal{M}_K)-\epsilon(\mathcal{M}_K)]$. It is notable that examples appear in a pairwise manner, which is not i.i.d. any more. However, we can transform the objective in the following way. Denote $n=\lfloor\frac{N}{2}\rfloor$, we have

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}}[g(\mathcal{D})]=\mathbb{E}_{\mathcal{D}}[\sup_{\mathcal{M}_K\in\Gamma}\epsilon_N(\mathcal{M}_K)-\epsilon(\mathcal{M}_K)]\\ &\leq\mathbb{E}_{\mathcal{D}}[\sup_{\mathcal{M}_K\in\Gamma}\frac{1}{n}\sum_{i=1}^{n}\ell\left(\mathbf{z}_i,\mathbf{z}_{i+n},\mathcal{M}_K\right)-\epsilon(\mathcal{M}_K)]&(15)\\ &=\mathbb{E}_{\mathcal{D}}[\sup_{\mathcal{M}_K\in\Gamma}\frac{1}{n}\sum_{i=1}^{n}\ell\left(\mathbf{z}_i,\mathbf{z}_{i+n},\mathcal{M}_K\right)-\mathbb{E}_{\mathcal{D}'}[\ell\left(\mathbf{z}_i',\mathbf{z}_{i+n}',\mathcal{M}_K\right)]]\\ &\leq\mathbb{E}_{\mathcal{D},\mathcal{D}'}[\sup_{\mathcal{M}_K\in\Gamma}\frac{1}{n}\sum_{i=1}^{n}\ell\left(\mathbf{z}_i,\mathbf{z}_{i+n},\mathcal{M}_K\right)-\ell\left(\mathbf{z}_i',\mathbf{z}_{i+n}',\mathcal{M}_K\right)]&(16)\\ &=\mathbb{E}_{\mathcal{D},\sigma}[\sup_{\mathcal{M}_K\in\Gamma}\frac{2}{n}\sum_{i=1}^{n}\sigma_i\ell\left(\mathbf{z}_i,\mathbf{z}_{i+n},\mathcal{M}_K\right)]&(17)\\ &=\mathbb{E}_{\mathcal{D},\sigma}[\sup_{\mathcal{M}_K\in\Gamma}\frac{2L}{n}\sum_{i=1}^{n}\sigma_iq_{i,i+n}\left(f_{q_{i,i+n}}(\mathbf{x}_i,\mathbf{x}_{i+n})-\gamma\right)]&(18)\\ &=\mathbb{E}_{\mathcal{D},\sigma}[\sup_{\mathcal{M}_K\in\Gamma}\frac{2L}{n}\sum_{i=1}^{n}\sigma_i(f_{q_{i,i+n}}(\mathbf{x}_i,\mathbf{x}_{i+n}))]\ .&(19)\end{aligned}$$

Eq. 15 results from the transformation theorem based on U-Statistics [33], [31], and after which loss function focuses on i.i.d. blocks. Redefining $\mathcal{D}'=\{\mathbf{z}_i'\}_{i=1}^N$ as another random sampled dataset and extracting out the expectation on true risk term, we get an upper bound as in Eq. 16, then the symmetrization techniques [34], [35] can be utilized to simplify the whole objective in Eq. 17. $\sigma_i\in\{-1,1\}$ is a Rademacher random variable equally distributed [34], and

$q_{i,i+n}\sigma_i$ has the same distribution with σ_i. Since $\ell(\cdot)$ is L-Lipschitz, Eq. 18 is the result of comparison lemma w.r.t. Rademacher variables [35]. Since $f_{q_{i,i+n}}$ could be $\max(\cdot)$ or $\min(\cdot)$, we can use the following identities to transform maximum and minimum operators given two inputs:

$$\max(a,b)=\frac{1}{2}[a+b+|a-b|]\ ,\min(a,b)=\frac{1}{2}[a+b-|a-b|]\ . \quad (20)$$

When there are two metrics M_1 and M_2 and combined with the maximum operator, then Eq. 19 can be expanded:

$$\mathbb{E}_{\mathcal{D},\sigma}[\sup_{\mathcal{M}_2\in\Gamma}\frac{2L}{n}\sum_{i=1}^{n}\sigma_i\kappa\{\mathrm{Dis}_{M_1}(\mathbf{x}_i,\mathbf{x}_{i+n}),\mathrm{Dis}_{M_2}(\mathbf{x}_i,\mathbf{x}_{i+n})\}] \quad (21)$$

$$\leq\mathbb{E}_{\mathcal{D},\sigma}\left[\sup_{\mathcal{M}_2\in\Gamma}\frac{L}{n}\sum_{i=1}^{n}\sigma_i[\mathrm{Dis}_{M_1}(\mathbf{x}_i,\mathbf{x}_{i+n})+\mathrm{Dis}_{M_2}(\mathbf{x}_i,\mathbf{x}_{i+n})]\right]$$
$$+\mathbb{E}_{\mathcal{D},\sigma}\left[\sup_{\mathcal{M}_2\in\Gamma}\frac{L}{n}\sum_{i=1}^{n}\sigma_i|\mathrm{Dis}_{M_1}(\mathbf{x}_i,\mathbf{x}_{i+n})-\mathrm{Dis}_{M_2}(\mathbf{x}_i,\mathbf{x}_{i+n})|\right] \quad (22)$$

$$\leq\mathbb{E}_{\mathcal{D},\sigma}\left[\sup_{\mathcal{M}_2\in\Gamma}\frac{2L}{n}\sum_{i=1}^{n}\sigma_i[\langle A_{i,i+n},M_1+M_2\rangle]\right] \quad (23)$$

$$\leq\mathbb{E}_{\mathcal{D},\sigma}\left[\sup_{\mathcal{M}_2\in\Gamma}\|M_1+M_2\|_F\frac{2L}{n}\sum_{i=1}^{n}\|\sigma_iA_{i,i+n}\|_F\right] \quad (24)$$

$$\leq\frac{\ell_u}{\lambda}\frac{2L}{n}\mathbb{E}_{\mathcal{D},\sigma}\left[\sum_{i=1}^{n}\|\sigma_iA_{i,i+n}\|_F\right] \quad (25)$$

$$\leq\frac{\ell_u}{\lambda}\frac{2L}{n}\sqrt{\mathbb{E}_{\mathcal{D},\sigma}\left[\sum_{i=1}^{n}\|\sigma_iA_{i,i+n}\|_F^2\right]}\leq\frac{2L\ell_uA}{\lambda\sqrt{n}} \quad (26)$$

Since $\pm\sigma_i$ have the same distribution, bounding Eq. 19 equals to bound Eq. 21. Based on the same property, we can analyze maximum and minimum operators together after transforming Eq. 19 based on Eq. 20, then we get Eq. 22. $|\cdot|$ function is 1-Lipschitz, combined with comparison lemma, we can remove the absolute value operator and get Eq. 23. Eq. 24 comes from the Cauchy-Schwarz inequality, to get an upper bound of the inner product. In Eq. 25, we use the triangle inequality to bound the norm of metric sums. With Jensen's inequality, we can remove the Rademacher variable as in Eq. 26. In the case of multiple metrics, we can follow the same analysis and get the results in Theorem 1. ■

7 Related Work and Discussions

Global DML approaches devote to finding a single metric for all instances [36], [30]. Types of constraints are used to provide side information. For example, using label directly [37], must-link/cannot-link pairs [2], [6], triplets with randomly sampled target neighbors [38] and Euclidean nearest neighbors [1], high order supervision like quadruplets [39].

Instead of being limited to a single type of feature relationship, local DML approaches further take spatial data heterogeneities into consideration. e.g., either *assigning* cluster-specific metric to instance based on locality [7], or *constructing* local metrics generatively [40]/discriminatively [41], [8]. Furthermore, instance specific metric learning methods [10], [9] extend the locality properties of linkages to extreme and gain improved classification performance. However, these DML methods, either global or local, take univocal semantic from *label*, in other words, the side information.

Richness of semantics is noticed and exploited by machine learning researchers [13], [11]. In DML community, Boosting Framework for DML (BDM) [42] and Similarity Component Analysis (SCA) [14] are proposed. BDM focuses on the semantic information from label annotation, which is less related to UM^{2}L. SCA, a multi-metric learning method based on pairwise constraints, focuses on learning metrics under one specific type of ambiguities, i.e., linkages are with competitive semantic meanings. UM^{2}L is a more general multi-metric learning framework which considers triplet constraints and *various kinds of* ambiguous linkages from both localities and semantic views. A type of non-metric embedding of objects based on only side information has also been considered [43], [44], [45]. Both more than one embedding representation of an object and the mixture coefficients are learned simultaneously, so as to get multiple maps of objects. On the one hand, UM^{2}L does not subject to the transductive setting as [43], [44], [45] and is able to utilize feature information of objects. On the other hand, an explicit type of semantic linkage between objects can be specified based on prior knowledge or concrete application.

UM^{2}L maintains good compatibilities and can degenerate to several state-of-the-art DML methods. For example, by considering univocal semantic ($K=1$), we can get a global metric learning model used in [23]. If we further choose the hinge loss and set the regularizer $\Omega(M)=\mathrm{tr}(MB)$, with $B=\sum_{(i,j)\in\mathcal{P},y_i=y_j}(\mathbf{x}_i-\mathbf{x}_j)(\mathbf{x}_i-\mathbf{x}_j)^\top$ an intra-class similar pair covariance matrix, UM^{2}L degrades to LMNN [7]. With trace norm on M, [46] is recovered. For multi-metric approaches, when the number of local metrics is the same as the class number and κ_v is the indicator of classes for the second instance in a similar or dissimilar pair, i.e., $\kappa(f_{M_1}(\mathbf{x}_i,\mathbf{x}_j),\ldots,f_{M_K}(\mathbf{x}_i,\mathbf{x}_j))=f_{M_{y_j}}(\mathbf{x}_i,\mathbf{x}_j)$, UM^{2}L can be transformed to MMLMNN [7].

Most theoretical analyses on DML focus on the single metric case, e.g., [30], [31]. In our theorem, not only the multiple metrics but also the non-linear operators are handled simultaneously to get the multi-metric generalization result.

Deep metric learning extracts representative embeddings with the deep neural network and utilizes pairwise or triplet information to supervise the learning process. Usually, networks applied to all instances share parameters, and the objective considers the effectiveness of side information usage [47], [48] or the pair/triplet generation problem [4], [49]. The deep extension of UM^{2}L combines multiple types of embeddings together with κ in a flexible way, and the usual mechanism can improve the task performance.

Discovering rich semantics from "weakly" supervision is widely investigated, e.g., multi-label ambiguity and insufficiency modeled by MIML [50] and MDDM [56]. A more comprehensive introduction of "weakly" supervision can be found in [51]. Our UM^{2}L provides a novel perspective to explore "weakly" supervision from links.

8 Experiments on Types of Applications

Due to different choices of $\kappa_{\pm1}$ in UM^{2}L, we test the framework in diverse real applications, namely social linkages, classification, multi-view semantic detection/visualization, physical semantic meaning distinguishing, and image clustering/retrieval. To simplify the discussion, we use triplet

BER↓	KM	SP	MAC	SCA	LMNN	ITML	EGO	UM^{2}L
F_348	.669	.669	.730	.847	.586	.633	.426	**.405**
F_414	.721	.721	.699	.870	.614	.562	.449	**.420**
F_686	.637	.637	.681	.772	.589	.626	.446	**.391**
F_698	.661	.661	.640	.729	.460	.386	**.392**	.420
F_1684	.807	.807	.767	.844	.803	.727	.491	**.465**
F_3980	.708	.708	.541	.667	.454	.428	.538	**.407**

Table 2: BER (lower is better) of the linkage discovering comparisons on Facebook datasets: UM^{2}L$_{\text{ADS}}$ vs. others

	Pairwise				Triplet		
κ	ADS	OVS	RGS	SUM	ADS	OVS	RGS
Clean1	.092	.133	.106	.098	**.070**	.118	.086
German	**.277**	.326	.279	.296	.281	.281	.284
Hayes-r	.327	.307	.321	.293	**.276**	.293	.307
Heart-s	.206	.309	.196	.226	.190	**.184**	.194
Liver-d	.393	.454	.423	.440	.363	.401	**.342**
Sonar	.167	.201	.170	.148	.136	.144	**.132**

Table 3: Test error comparison between different κ implementation in the UM^{2}L framework, for pairwise and triplet variants.

variant in Eq. 3 to incorporate more local information in most cases, but also present the performance of pairwise variant in the first two subsections. In addition, we use alternative batch solver, smooth hinge loss, $\ell_{2,1}$-norm regularizer $\Omega_k(M_k) = \|M_k\|_{2,1}$ if without further statement. For each example, triplets are constructed with 3 targets and 10 impostors with its Euclidean nearest neighbors. Our experiments are performed on a cluster of 32 machines, each of which has four 6-core 2.53GHz CPUs and 48G RAM.

8.1 Social Linkage Discovering

ADS configuration is designed for social linkage and pattern discovering, where the friendship linkages between two users may be constructed from one of their common hobbies. Since pairwise linkages information are directly provided based on a social network, the pairwise variant of UM^{2}L$_{\text{ADS}}$ is used. To validate its effectiveness, we test it on social network data to show its grouping ability on linkages.

Social linkages come from 6 real-world Facebook network datasets from [13]. Given annotated friendship circles of an ego user as ground truth, the goal of ego-user linkages discovering is to utilize the *overall linkage* of multiple friendship circles and users' binary features to figure out how users are grouped. We form instances by taking absolute value of differences between features of the ego and others. After circles with < 5 nodes are removed, K is configured as the number of circles remained. Thus, users with different common hobbies are grouped together into multiple sets. MAC detects group assignments based on binary features [52]; SCA constructs user linkages in a probabilistic way, and EGO [13] can directly output user circles. KMeans (KM) and Spectral Clustering (SC) directly group users based on their features without using linkages. Two metric learning methods LMNN [7] and ITML [6] serve as baselines, where side-information helps the learning of clustering distance measure. For UM^{2}L$_{\text{ADS}}$, multiple learned metrics correspond to multiple latent semantic hobbies, and linkage between users w.r.t. a certain semantic can be determined by their distance value with the metric, thresholding by the parameter γ in Eq. 2. Therefore, with learned metric set $\mathcal{M}_K$ and γ, users in the same semantic space can be grouped together. [4]

Performance is measured as the difference between the aligned output and ground truth sets by Balanced Error Rate (BER) [13], the lower the better. Results are listed in Table 2. The superiority of DML baseline w.r.t. direct clustering shows the necessity to use side information. Since multiple semantic spaces are learned simultaneously, and the friendship linkage generation is consistent with the ADS assumption, UM^{2}L$_{\text{ADS}}$ gets better results than others.

4. The learning method for γ is in the supplemental material.

8.2 Comparisons on Classification Performance

Before comparing UM^{2}L with others, we first analyze the classification performance of UM^{2}L variants, for different selection of κ, and both the pairwise/triplet versions. The summation form $\kappa = \sum$ is listed as a baseline. For Apical Dominating Similarity (ADS), it tries to explain similarities between instances with different semantics or local reasons; For One Vote Similarity (OVS), it attributes similarities between objects to possible different reasons; For Rank Grouping Similarity (RGS), it can be regarded as rigorous restrictions on each learned metric that given constraints should be satisfied over all learned metrics. Because in UM^{2}L distance values from different metrics are comparable, so in the test phase, a variant of kNN is applied to use multiple learned metrics. With 3NN, we first compute 3 nearest neighbors for testing instance $\tilde{\mathbf{x}}$ using each base metric M_k. Then $3 \times K$ distance values are collected adaptively, and the smallest three (with the highest similarity scores) form neighbor candidates. Majority voting is used for prediction.

In the pairwise implementation, 10 targets and impostors are selected based on Euclidean nearest neighbors to generate pairs. Component number K is set as the class number. 3NN results tested on the 6 of 10 benchmark datasets in total are listed in Table 3, and evaluations are repeated for 30 times (Full tables can be found in the supp.). In each trial, 70% of instances are used for training, and the remaining part is for the test. For all methods, cross-validation is employed to choose best parameters. From the results, the triplet versions of UM^{2}L get better and stable results in most cases since more local information is utilized during training. OVS could not output comparable results in some cases, since the linkage explanation of OVS is more for a decomposition goal, not for classification. Therefore, we mainly consider ADS and RGS for classification, while using OVS for visualization tasks.

Same setting is used to test classification generalization performance with 8 state-of-the-art metric learning methods. In detail, global DML methods: ITML [6], LMNN [7] and EIG [53]; local and instance specific DML methods: PLML [41], SCML (local version) [8]; MMLMNN [7], ISD [9] and GLML [40]. Generalization errors (mean±std.) based on 3NN are listed in Table 4 where Euclidean distance results (EUCLID) are also listed as a baseline. For local DML methods except the instance specific one, the default number of metrics K is configured as the number of classes.

Table 4: Comparisons of classification performance (test errors, mean ± std.) based on 3NN. Triplet version of UM^{2}L$_{\text{ADS}}$ and UM^{2}L$_{\text{RGS}}$ are compared. The best performance on each dataset is in bold. Last two rows list the Win/Tie/Lose counts of UM^{2}L$_{\text{ADS/RGS}}$ against other methods on all datasets with t-test at significance level 95%.

	UM^{2}L$_{\text{ADS}}$	UM^{2}L$_{\text{RGS}}$	PLML	SCML	MMLMNN	ISD	GLML	ITML	LMNN	EIG	EUCLID
Autompg	**.201±.034**	.225±.031	.265±.048	.253±.026	.256±.032	.288±.033	.230±.029	.292±.032	.261±.032	.266±.031	.260±.036
Clean1	**.070±.018**	.086±.020	.098±.027	.100±.027	.097±.022	.143±.023	.111±.024	.141±.024	.084±.020	.127±.021	.139±.023
German	.281±.019	.284±.030	.280±.016	.302±.021	.289±.019	.297±.017	**.266±.017**	.288±.021	.291±.020	.284±.014	.296±.021
Glass	.312±.043	**.293±.047**	.389±.050	.328±.054	.296±.047	.334±.050	.311±.045	.311±.038	.301±.046	.314±.050	.307±.042
Hayes-r	**.276±.044**	.307±.068	.436±.201	.296±.053	.282±.062	.378±.093	.291±.058	.342±.080	.305±.065	.289±.067	.398±.046
Heart-s	.190±.035	.194±.063	.365±.127	.205±.040	.191±.037	.192±.036	.190±.035	**.186±.032**	.203±.034	.189±.034	.190±.030
House-v	.051±.015	**.048±.013**	.121±.240	.066±.019	.055±.017	.072±.024	.051±.014	.063±.023	.056±.017	.080±.024	.083±.025
Liver-d	.363±.045	**.342±.047**	.361±.055	.371±.042	.372±.045	.364±.042	.351±.039	.377±.052	.360±.046	.380±.037	.384±.040
Segment	**.023±.038**	.029±.034	.041±.031	.041±.008	.036±.006	.063±.009	.053±.008	.050±.012	.038±.007	.059±.016	.050±.007
Sonar	.136±.032	**.132±.036**	.171±.048	.193±.045	.157±.038	.182±.038	.169±.036	.174±.039	.145±.032	.159±.042	.168±.036
W / T / L	UM^{2}L$_{\text{ADS}}$ vs. others		6 / 4 / 0	7 / 3 / 0	4 / 6 / 0	7 / 3 / 0	4 / 5 / 1	6 / 4 / 0	3 / 7 / 0	6 / 4 / 0	8 / 2 / 0
W / T / L	UM^{2}L$_{\text{RGS}}$ vs. others		6 / 4 / 0	8 / 2 / 0	5 / 5 / 0	9 / 1 / 0	4 / 5 / 1	8 / 2 / 0	4 / 6 / 0	7 / 3 / 0	8 / 2 / 0

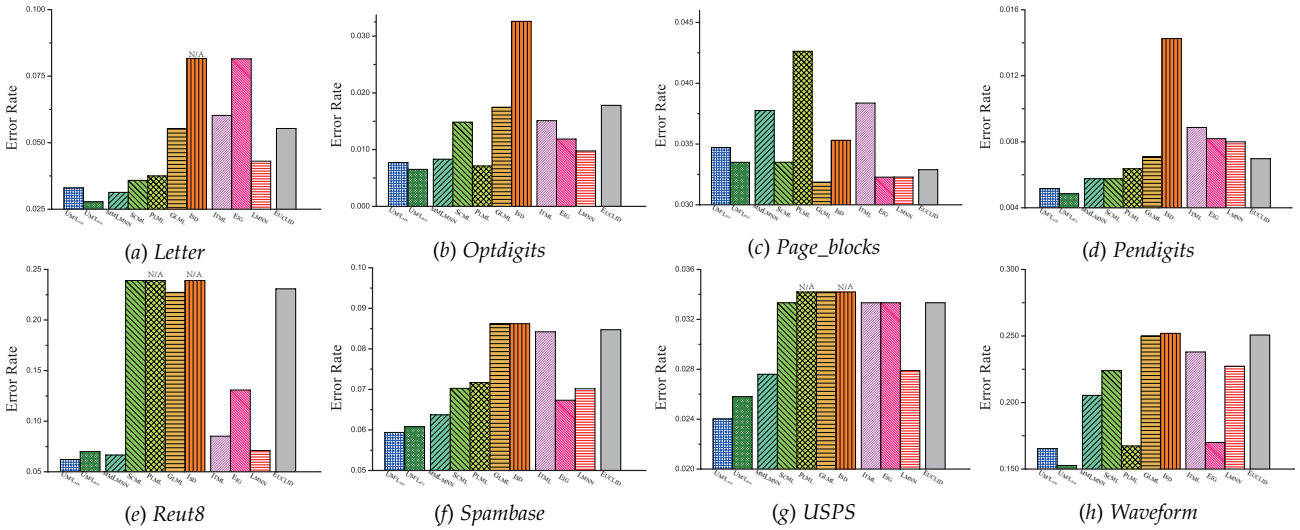


Figure 3: Classification test error of UM^{2}L with other DML methods on 8 datasets. UM^{2}L based on ADS and RGS are placed at first. Followings are local and global based methods. Last is Euclidean 3NN baseline. Four groups of methods, namely UM^{2}L, local DML, global DML and baseline kNN with Euclidean distance (denoted as EUCLID), are separated by spaces.

From comparison results, DML methods, in general, will improve the classification performance when compared with EUCLID, which indicates a learned metric is more helpful than the original Euclidean distance. When multiple metrics are learned, the local property of data can be explored, and the final performance can be further improved. For example, in "German", PLML and MMLMNN get better performance than the global LMNN and ITML. When in an extreme case like ISD, however, the model often overfits when each instance has a specific metric. Table 4 clearly shows that UM^{2}L$_{\text{ADS/RGS}}$ perform well on most datasets. Especially, UM^{2}L$_{\text{RGS}}$ achieves best on more datasets according to t-tests and this can be attributed to the rigorous restrictions of RGS.

To further test the classification ability of UM^{2}L, Fig. 3 shows the comparisons results on some larger datasets. 3NN is also used to test the performance of each method. Each dataset is randomly split into three parts, 40% for training, 30% for validation, and the rest is for test. Each method tunes parameters on the validation set and then retrains the model with the best parameters on the combination of training and validation data. If a method cannot give a result in 24 hours, we set its test error the same as the worst one among the others and denote it as "N/A". From comparison results, UM^{2}L$_{\text{ADS/RGS}}$ can achieve best results on 7/8 datasets. In general, the RGS version performs better, but it may overfit sometimes, e.g., in *Reut8*. MMLMNN is also competitive on most datasets since it considers the spatial linkages between instances well. SCML (local version) first constructs bases from data and then builds a parametric function to assign metric to each instance. The performance of SCML depends on the bases selection and optimization process. GLML performs well on small datasets, but not show obvious advantages for large problems. Its performance may depend on the scale of the data. ISD and PLML consume a lot on large datasets. In particular, ISD, which learns a metric for each instance in a transductive way, often tends to overfit.

More investigations are conducted to reveal the influences of the number of metrics on classification performance. Fig. 4 gives test errors on different datasets when UM^{2}L$_{\text{ADS/RGS}}$ are provided with different number of metrics. From the results, we can find that there are turning points on performance curves, i.e., when the number of metrics increases,

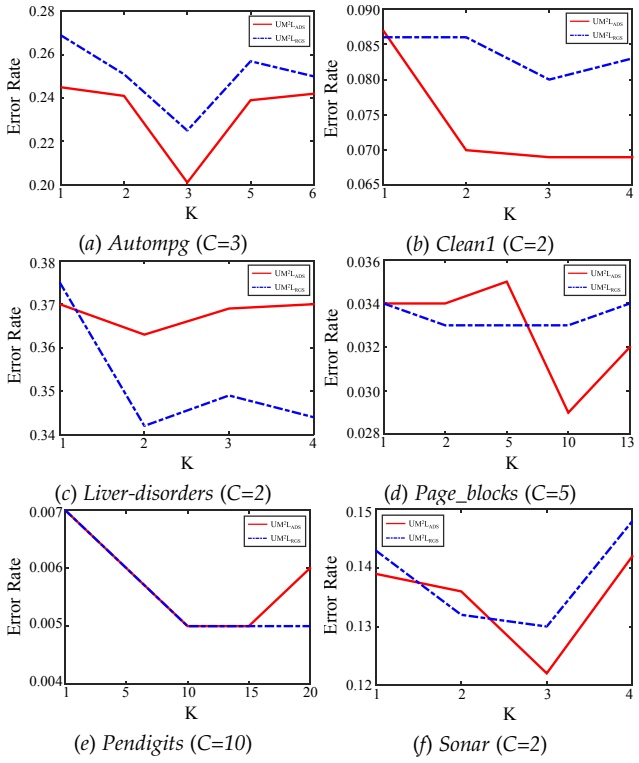


Figure 4: The change of classification performance (test error rate) when the number of metrics learned in UM^{2}L$_{\text{ADS}}$ and UM^{2}L$_{\text{RGS}}$ is changed. Bracket after the name of each dataset shows the number of classes (C).

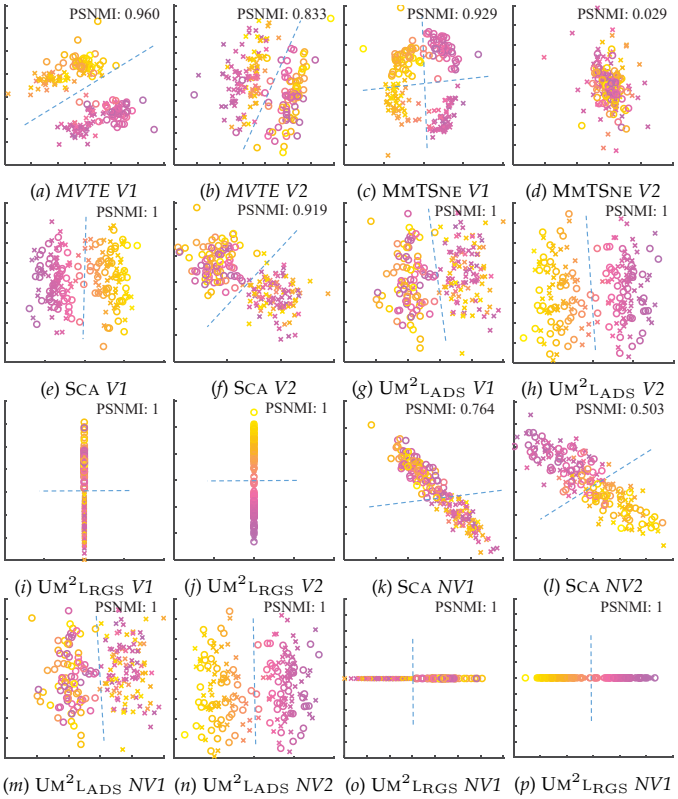


Figure 5: Subspaces discovered by MVTE, SCA, MMTSNE and UM^{2}L given instances with two semantic components, i.e., color and shape. Blue dot-lines give the possible decision boundary (best viewed in color). "V" means the discovered view, and "NV" means the noise perturbed case. Right upper corner of each plot shows the paired semantic NMI (PSNMI) value, a numerical measurement of multi-view discovery task, for a single projection, the higher the better.

the test error decreases at first and then increases. This could be attributed to the model complexity: at first the model becomes powerful and then it cannot prevent overfitting. From the empirical results, it is suggested that by configuring K (the number of metrics) close to the number of classes (when the number of latent semantics is known), the model can return satisfied results in most cases.

8.3 Investigations of Latent Multi-View Detection

Another direct application of UM^{2}L is hidden multi-view detection, where data can be described by multiple views from different channels yet feature partitions are not clearly provided [54]. Data with multi-view goes consistently with the assumption of ADS or RGS. ADS emphasizes the existence of relevant views and aims at decomposing helpful aspects or views; while RGS requires full accordance among views. Trace norm regularizes the approach here to get low dimensional projections. UM^{2}L facilitates the understanding of data by decomposing each metric to low dimensional subspace projectors, i.e., for each $M_k = L_k L_k^\top$, a $\hat{L}_k \in \mathbb{R}^{d\times 2}$ corresponds to the largest 2 eigen-values is picked.

The hidden multi-view dataset [45] contains 200 instances with two hidden views (color and shape), and groundtruth triplets are used as side-information. We perform UM^{2}L with $K = 2$. Fig. 5 (*g*) (*h*) give the 2-D visualization results by plotting the projected instances in subspaces corresponding to metric M_1 and M_2 of UM^{2}L$_{\text{ADS}}$. It clearly shows that M_1 captures the semantic view of shape, and M_2 reflects the meaning of color. While for UM^{2}L$_{\text{RGS}}$, the subspace visualization results are more discriminative, as in Fig. 5 (*i*) (*j*). It can be clearly found that both UM^{2}L$_{\text{ADS}}$ and UM^{2}L$_{\text{RGS}}$ can capture the two different semantic views hidden in data. MVTE [45] generates 2-D representation of data only based on triplets, which gives 2-view results in Fig. 5 (*a*) and (*b*). It can be found that these two representations of data correspond to color and shape, respectively. However, there are some outliers, which reduce the discriminative ability in each view. MMTSNE is the multi-map version of the popular TSNE visualization method [44], it discovers two subspaces (*c*) and (*d*) given pairwise similarity map. The phenomenon of embedding concentration in one of the two maps (*d*) is also discussed in [43]. SCA [14] is also compared. With linkages from triplets and combined original features, it can produce two low rank projections and get results in Fig. 5 (*e*) to (*f*). View1 reflects the color semantic and View2 is the shape one.

To better compare UM^{2}L and SCA, we test them under a noisy environment. By concatenating original data with 10 dimension random noise from [0,1], together with true triplet information, results of UM^{2}L$_{\text{ADS}}$ and SCA are showed in subplots (*k*) - (*p*) in Fig. 5. In this case, SCA can find the color view as well, but in color view (*l*), two colors are not easy to determine, especially for instances near the boundary. UM^{2}L$_{\text{ADS}}$ still finds color and shape views, and its projections are easy for boundary determination. Thus, UM^{2}L is robust to noise. UM^{2}L with RGS is also tested on

(*a*) LMNN *metric* (*b*) PLML *metric 1* (*c*) PLML *metric 2* (*d*) MMLMNN *1* (*e*) MMLMNN *2* (*f*) MMLMNN *3*

(*g*) SCA *metric 1* (*h*) SCA *metric 2* (*i*) SCA *metric 3* (*j*) SCA *metric 4* (*k*) SCA *metric 5* (*l*) SCA *metric 6*

(*m*) UM^{2}L$_{\text{ADS}}$ *metric 1* (*n*) UM^{2}L$_{\text{ADS}}$ *metric 2* (*o*) UM^{2}L$_{\text{ADS}}$ *metric 3* (*p*) UM^{2}L$_{\text{ADS}}$ *metric 4* (*q*) UM^{2}L$_{\text{ADS}}$ *metric 5* (*r*) UM^{2}L$_{\text{ADS}}$ *metric 6*

(*s*) UM^{2}L$_{\text{OVS}}$ *metric 1* (*t*) UM^{2}L$_{\text{OVS}}$ *metric 2* (*u*) UM^{2}L$_{\text{OVS}}$ *metric 3* (*v*) UM^{2}L$_{\text{OVS}}$ *metric 4* (*w*) UM^{2}L$_{\text{OVS}}$ *metric 5* (*x*) UM^{2}L$_{\text{OVS}}$ *metric 6*

Figure 6: Word clouds generated from the results of compared DML methods. The size of a word depends on the importance weight of each word (feature). The weight is calculated by decomposing each metric $M_k = L_k L_k^\top$, and calculate the ℓ_2-norm of each row in L_k, where each row corresponds to a specific word. Each subplot gives a word cloud for a learned base metric.

the noisy data as in Fig. 5 (*o*) (*p*). In this noisy case which is hard to classify in both views, it is notable that the results of UM^{2}L$_{\text{RGS}}$ not only reflect two views of data (color and shape) but also have high discriminative ability, resulting from both the robustness and rigorous constraints of UM^{2}L$_{\text{RGS}}$.

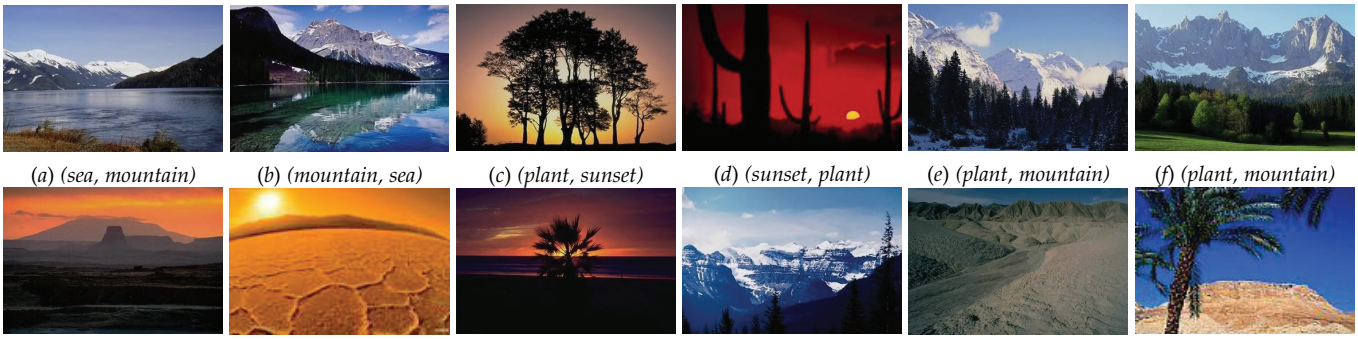
We also quantitatively measure the performance of above multi-view discovery task using a new criterion: paired semantic NMI (PSNMI). The underlying label for color and shape can be pre-collected, then with the generated semantic multi-view projection, we compute clustering NMI criterion w.r.t. both two label sets. The matching for two projections and two labels is determined by the better average NMI values between two views. The results of single view PSNMI can be found in the right upper corner of each figure, which is a comprehensive depiction of the multi-view latent space discovery ability. From the numerical results, UM^{2}L variants perform better on this specific task than others.

8.4 Comparisons of Latent Semantic Discovering

UM^{2}L is proposed for DML with both localities and semantic linkages considered. Hence, to investigate the ability of latent semantics discovering, two assessments in real applications are performed, i.e., Academic Paper Linkages Explanation (APLE) and Image Weak Label Discovering (IWLD).

In APLE, data are collected from 2012-2015 ICML papers, which can be connected with each other by more than one topic, yet only the session ID is captured to form explicit linkages. 3 main directions of sessions are picked up in this assessment, i.e., "feature learning", "online learning" and "deep learning". No sub-fields and additional labels/topics are provided. Simplest TF-IDF is used to extract features, which forms a corpus of 220 papers and 1622 words in total. Aiming at finding the hidden linkages together with their causes, both UM^{2}L$_{\text{ADS}}$ and UM^{2}L$_{\text{OVS}}$ are invoked. To avoid trivial solutions, regularizer for each metric is configured as $\Omega_k(M_k) = \|M_k - I\|_F^2$ for UM^{2}L$_{\text{OVS}}$. All feature (word) weights and correlations can be provided by learned metrics, i.e., with decomposition $M_k = L_k L_k^\top$, the ℓ_2-norm value of each row in L_k can be regarded as the weight for each feature (word). The importance of feature (word) weights is demonstrated in word clouds in Fig. 6, where the size of fonts reflects the weights of each word.

Fig. 6 shows the results of LMNN [7] (*a*), PLML [41] (*b*, *c*), MMLMNN [7] (*d*, *e*, *f*), SCA [14] (*g* $\sim$ *l*), UM^{2}L$_{\text{ADS}}$ (*m* $\sim$ *r*) and UM^{2}L$_{\text{OVS}}$ (*s* $\sim$ *x*), respectively. Global method LMNN returns one subplot. The metric learned by LMNN perhaps has discriminative ability but the weights of words cannot distinguish subfields in 3 selective domains. For multi-metric learning approaches PLML and MMLMNN, though they can provide more than one base metric and consequently have multiple word clouds, the words presented in subplots are not with legible physical semantic meanings. Especially, PLML outputs multiple metrics which are similar to each other (tends to global learner's behavior) and only focus on first part of the alphabet, while MMLMNN by default only learns multiple metrics with the number of base metrics equaling to the number of classes. The number of metrics learned by SCA and UM^{2}L is set to 6. SCA can discover some key words in "online learning" and "deep learning" fields, such as "reward", "bound" and "adversary" about online learning as well as "GPU", "layer" for deep learning. Results of UM^{2}L$_{\text{OVS}}$ clearly demonstrate all 3 fields. On session "online learning", it can discover different sub-fields such as "online convex optimization" (*s* and *t*), and "online (multi-) armed bandit problem" (*v*); for session "feature learning", it has "feature score" (*u*) and "PCA projection" (*x*); and for "deep learning", the word cloud returns popular words like "network layer", "autoencoder" and "layer" (*w*). For UM^{2}L$_{\text{ADS}}$, 3 main session semantics are also discovered, but weights of words are different from UM^{2}L$_{\text{OVS}}$. From subplot (*m*), the main topic should be domain adaptation, which can be related to transferrable feature learning in deep learning researches. Subplots (*n*), (*o*) and (*q*) are all



(*a*) (*sea, mountain*) (*b*) (*mountain, sea*) (*c*) (*plant, sunset*) (*d*) (*sunset, plant*) (*e*) (*plant, mountain*) (*f*) (*plant, mountain*)

(*g*) (*sunset, mountain*) (*h*) (*desert, sunset*) (*i*) (*plant, sunset*) (*j*) (*mountain, plant*) (*k*) (*desert, mountain*) (*l*) (*desert, plant*)

Figure 7: Results of visual semantic discovery on images. The first annotation in the bracket is the provided weak label, and the second one is one of the latent semantic labels discovered by UM^{2}L.

about feature learning but with different subfields, i.e., (*n*) should be mainly about structure feature learning in DML, and pairwise constraints between items are emphasized; (*o*) is about manifold learning in feature construction and (*q*) is about subspace learning and dimensionality reduction in feature learning, with the key word "eigenvector" being emphasized. Subplot (*p*) is related to deep learning, the word cloud clearly shows the item "network layer", "RBM", etc. (*r*) is definitely about online learning, with key words "arm", "optimal", "bandit" and "regret". It is notable that latent semantics are discovered from leaned metric, which validates the discriminative ability of UM^{2}L over semantics. It will benefit some subsequent tasks such as classification.

Besides APLE, UM^{2}L can also be applied to Image Weak Label Discovering (IWLD). For an image, there may be multiple complex semantics [55], [56], [50]. The linkage between two images may only depend on one of their shared semantic, while the disconnection between two images may also lie in a particular semantic. We use an image dataset from [55] where every image contains one or more labels from desert, mountain, sea, sunset, and plant. For each image, we select its most obvious label and transform this dataset such that each instance only has a single label. Thus, in this case, since images are with plenty of latent semantics, similarities between images are just based on one of them. UM^{2}L$_{\text{OVS}}$ can obtain multiple metrics, each of which is with a certain visual semantic. By computing similarities based on different metrics, latent semantics can be discovered, i.e., if we assume images connected with high similarities share the same label, missing labels can be completed. Results of IWLD are showed in Fig. 7, where the image annotations in brackets indicate the pair of training label and discovered label. One image may have complex semantics. For instance, image (*b*) is about a lake beside a mountain and image (*j*) is a picture of mountains and trees. They are similar since both of them have mountains (similar since they are computed with the metric for semantic "mountain"). Image (*a*) is also about mountains, but the lake is more obvious. Given the training label "sea", it is hard to link with pictures about "mountains". UM^{2}L$_{\text{OVS}}$ can learn multiple metrics, one for each semantic. If images (*a*) and (*b*) are dissimilar in the supervision triplets, UM^{2}L$_{\text{OVS}}$ finds one metric (maybe "sunset") to explain their disconnection, and does not deny their similarity over metrics about "sea" and "mountain".

	NMI	R@1	R@2	R@4	R@8
Triplet [4]	53.35	51.54	63.78	73.52	82.41
Lift Struct [47]	56.88	52.98	65.70	76.01	84.27
NPairs [49]	57.79	53.90	66.76	77.75	86.35
Clustering [48]	59.04	58.11	70.64	80.27	87.81
UM^{2}L$_{\text{ADS}}$	59.90	69.03	79.63	87.02	91.80
UM^{2}L$_{\text{OVS}}$	58.81	68.69	78.51	86.39	92.01
UM^{2}L$_{\text{RGS}}$	**61.35**	**70.61**	**80.67**	**88.21**	**93.04**

Table 5: Clustering and retrieval performance comparisons on CAR196 datasets: UM^{2}L vs. others

8.5 Comparison on Image Clustering

We investigate the ability of UM^{2}L on image clustering and retrieval using its deep extension, as introduced in Section 5, on Cars196 dataset [57]. There are totally 196 classes of different cars. Using the same test protocol as [48], we choose the first 98 classes for training, and the remaining for the test. Using GoogLeNet as feature transformation, multiple fully connected layers generate embeddings with different semantic meanings. As in [48], the embedding size d_l is set to 64, which is reported not influence the final performance a lot [47]. The number of component in UM^{2}L is set as 5 in our experiments, and we combine all learned semantic embeddings together for last clustering and retrieval task. Triplet loss is used as the objective, and semi-hard instances are selected to build triplet in each batch [4]. For pre-processing, input images are resized to 256×256, random cropped to size 227 (center crop in the test phase), and random horizontal flipped; for optimization, we set batch size to 128, use stochastic gradient descent with momentum 0.9, and basic learning rate 1e-4, where the learning rate for the last layer is 10 times faster than previous ones [47].

We use the NMI score to measure the quality of clustering, as well as Recall@K for the retrieval performance. We compare with following losses with a single embedding, i.e., the triplet semihard loss [4], the Lifted structure loss [47], the NPairs loss [49], and the clustering loss [48]. Results are shown in Table 5, where the values of compared methods come from [48] since using the same experimental setting. Due to the explicit learning of multiple semantic embeddings, UM^{2}L variants perform better than others, and the RGS case achieves the best results w.r.t. all the measurements.

9 Conclusion

Instead of learning multiple metrics spatially, we focus on the *semantic multi-metric* concept. The proposed Unified Multi-Metric Learning (UM^{2}L) framework can exploit side information from multiple aspects such as locality and semantic. It is notable that both types of the ambiguous linkages and both organized pairwise/triplet forms can be absorbed in this framework. The rich semantics underlying linkages can be modeled by various flexible function operator κs. By implementing κs in different forms, three novel similarities between objects are derived, and the learned multiple local metrics can be used for classification, latent semantic linkage discovering, etc. UM^{2}L variants have a unified solution, and the generalization ability is guaranteed theoretically. The deep extension improves the quality of embeddings of UM^{2}L a lot. Experiments show the flexibility of UM^{2}L in various applications. Automatic determination of the number of metrics is interesting future work.

Acknowledgments

This research was supported by NSFC (61673201, 61751306, 61773198) and Huawei Funding.

References

[1] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Advances in Neural Information Processing Systems 18*. MIT Press, 2006, pp. 1473–1480.

[2] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning with application to clustering with side-information," in *Advances in Neural Information Processing Systems 15*. MIT Press, 2003, pp. 505–512.

[3] B. McFee and G. R. Lanckriet, "Metric learning to rank," in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010, pp. 775–782.

[4] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA., 2015, pp. 815–823.

[5] X. Wang, S. Jiang, P. Gao, X. Ju, R. Wang, and Y. Zhang, "Cost-effective testing based fault localization with distance based test-suite reduction," *SCIENCE CHINA Information Sciences*, vol. 60, no. 9, pp. 092 112:1–092 112:15, 2017.

[6] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proceedings of the 24th International Conference on Machine Learning*, Corvalis, OR., 2007, pp. 209–216.

[7] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *Journal of Machine Learning Research*, vol. 10, pp. 207–244, 2009.

[8] Y. Shi, A. Bellet, and F. Sha, "Sparse compositional metric learning," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, Quebec, Canada, 2014, pp. 2078–2084.

[9] D.-C. Zhan, M. Li, Y.-F. Li, and Z.-H. Zhou, "Learning instance specific distances using metric propagation," in *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009, pp. 1225–1232.

[10] E. Fetaya and S. Ullman, "Learning local invariant mahalanobis distances," in *Proceedings of the 32nd International Conference on Machine Learning*, Pairs, France, 2015, pp. 162–168.

[11] D. Chakrabarti, S. Funiak, J. Chang, and S. Macskassy, "Joint inference of multiple label types in large networks," in *Proceedings of The 31st International Conference on Machine Learning*, Beijing, China, 2014, pp. 874–882.

[12] J. Hu, D.-C. Zhan, X. Wu, Y. Jiang, and Z.-H. Zhou, "Pairwised specific distance learning from physical linkages," *ACM Transactions on Knowledge Discovery from Data*, vol. 9, no. 3, p. Article 20, 2015.

[13] J. Leskovec and J. J. Mcauley, "Learning to discover social circles in ego networks," in *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012, pp. 539–547.

[14] S. Changpinyo, K. Liu, and F. Sha, "Similarity component analysis," in *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., 2013, pp. 1511–1519.

[15] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. J. Belongie, and D. Estrin, "Collaborative metric learning," in *Proceedings of the 26th International Conference on World Wide Web*, Perth, Australia, 2017, pp. 193–201.

[16] Z.-H. Zhou, "Learnware: On the future of machine learning," *Frontiers of Computer Science*, vol. 10, no. 4, pp. 589–590, 2016.

[17] H. Do, A. Kalousis, J. Wang, and A. Woznica, "A metric learning perspective of svm: on the relation of lmnn and svm," in *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, La Palma, Canary Islands, 2012, pp. 308–317.

[18] K. Q. Weinberger and G. Tesauro, "Metric learning for kernel regression," in *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, San Juan, Puerto Rico, 2007, pp. 612–619.

[19] B. Wang, J. Jiang, W. Wang, Z.-H. Zhou, and Z. Tu, "Unsupervised metric fusion by cross diffusion," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Providence, RI., 2012, pp. 2997–3004.

[20] J. Wang, A. Woznica, and A. Kalousis, "Learning neighborhoods for metric learning," in *Proceedings of the 2012 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Bristol, UK., 2012, pp. 223–236.

[21] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. Chapman & Hall/CRC, Boca Raton, FL., 2012.

[22] B. McFee and G. Lanckriet, "Learning multi-modal similarity," *Journal of machine learning research*, vol. 12, pp. 491–523, 2011.

[23] Q. Qian, R. Jin, S. Zhu, and Y. Lin, "Fine-grained visual categorization via multi-stage metric learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Boston, MA., 2015, pp. 3716–3724.

[24] Y. Nesterov, *Introductory lectures on convex optimization*. Secaucus, NJ.: Springer Science & Business Media, 2004, vol. 87.

[25] N. Li, R. Jin, and Z.-H. Zhou, "Top rank optimization in linear time," in *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 1502–1510.

[26] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[27] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *Journal of Machine Learning Research*, vol. 11, pp. 1109–1135, 2010.

[28] D. Lim, G. Lanckriet, and B. McFee, "Robust structural metric learning," in *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, GA., 2013, pp. 615–623.

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA., 2015, pp. 1–9.

[30] A. Bellet, A. Habrard, and M. Sebban, *Metric Learning*, ser. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2015.

[31] Q. Cao, Z. Guo, and Y. Ying, "Generalization bounds for metric and similarity learning," *Machine Learning*, vol. 102, no. 1, pp. 115–132, 2016.

[32] C. McDiarmid, "On the method of bounded differences," *Surveys in combinatorics*, vol. 141, no. 1, pp. 148–188, 1989.

[33] S. Clémençon, G. Lugosi, and N. Vayatis, "Ranking and empirical minimization of u-statistics," *The Annals of Statistics*, pp. 844–874, 2008.

[34] P. L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results," *Journal of Machine Learning Research*, vol. 3, pp. 463–482, 2002.

[35] V. Koltchinskii, *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems*. Springer-Verlag Berlin Heidelberg, 2011, vol. 2033.

[36] B. Kulis, "Metric learning: A survey," *Foundations and Trends in Machine Learning*, vol. 5, no. 4, pp. 287–364, 2012.

[37] H.-J. Ye, D.-C. Zhan, and Y. Jiang, "Instance specific metric subspace learning: A bayesian approach," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, Phoenix, AZ., 2016, pp. 2272–2278.

[38] M. Schultz and T. Joachims, "Learning a distance metric from relative comparisons," in *Advances in Neural Information Processing Systems 16*. MIT Press, 2004, pp. 41–48.

[39] M. T. Law, N. Thome, and M. Cord, "Learning a distance metric from relative comparisons between quadruplets of images," *International Journal of Computer Vision*, pp. 1–30, 2016.
[40] Y. Noh, B. Zhang, and D. D. Lee, "Generative local metric learning for nearest neighbor classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 1, pp. 106–118, 2018.
[41] J. Wang, A. Kalousis, and A. Woznica, "Parametric local metric learning for nearest neighbor classification," in *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012, pp. 1601–1609.
[42] L. Yang, R. Jin, L. Mummert, R. Sukthankar, A. Goode, B. Zheng, S. C. Hoi, and M. Satyanarayanan, "A boosting framework for visuality-preserving distance metric learning and its application to medical image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 30–44, 2010.
[43] J. Cook, I. Sutskever, A. Mnih, and G. E. Hinton, "Visualizing similarity data with a mixture of maps," in *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, San Juan, Puerto Rico, 2007, pp. 67–74.
[44] L. Van Der Maaten and G. Hinton, "Visualizing non-metric similarities in multiple maps," *Machine learning*, vol. 87, no. 1, pp. 33–55, 2012.
[45] E. Amid and A. Ukkonen, "Multiview triplet embedding: Learning attributes in multiple maps," in *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, 2015, pp. 1472–1480.
[46] K. Huang, Y. Ying, and C. Campbell, "Gsml: A unified framework for sparse metric learning," in *Proceedings of the 9th IEEE International Conference on Data Mining*, Miami, FL., 2009, pp. 189–198.
[47] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV., 2016, pp. 4004–4012.
[48] H. O. Song, S. Jegelka, V. Rathod, and K. Murphy, "Deep metric learning via facility location," in *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI., 2017, pp. 2206–2214.
[49] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., 2016, pp. 1849–1857.
[50] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li, "Multi-instance multi-label learning," *Artificial Intelligence*, vol. 176, no. 1, pp. 2291–2320, 2012.
[51] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National Science Review*, vol. 5, no. 1, pp. 44–53, 2018.
[52] M. Frank, A. P. Streich, D. Basin, and J. M. Buhmann, "Multi-assignment clustering for boolean data," *Journal of Machine Learning Research*, vol. 13, pp. 459–489, 2012.
[53] Y. Ying and P. Li, "Distance metric learning with eigenvalue optimization," *Journal of Machine Learning Research*, vol. 13, pp. 1–26, 2012.
[54] W. Wang and Z.-H. Zhou, "A new analysis of co-training," in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010, pp. 1135–1142.
[55] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.
[56] Y. Zhang and Z.-H. Zhou, "Multilabel dimensionality reduction via dependence maximization," *ACM Transactions on Knowledge Discovery from Data*, vol. 4, no. 3, p. 14, 2010.
[57] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *The 4th International IEEE Workshop on 3D Representation and Recognition*, Sydney, Australia, 2013.

Han-Jia Ye received his B.Sc. degree from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2013. Currently, he is working towards the PhD degree with the National Key Lab for Novel Software Technology, the Department of Computer Science & Technology in Nanjing University, China. His research interests lie primarily in machine learning, including distance metric learning, multi-modal/multi-task learning, meta-learning, and semantic mining.

De-Chuan Zhan received the Ph.D. degree in computer science, Nanjing University, China in 2010. At the same year, he became a faculty member in the Department of Computer Science and Technology at Nanjing University, China. He is currently an Associate Professor with the Department of Computer Science and Technology at Nanjing University. His research interests are mainly in machine learning, data mining and mobile intelligence. He has published over 20 papers in leading international journal/conferences. He serves as an editorial board member of IDA and IJAPR, and serves as SPC/PC in leading conferences such as IJCAI, AAAI, ICML, NIPS, etc.

Yuan Jiang received the PhD degree in computer science from Nanjing University, China, in 2004. At the same year, she became a faculty member in the Department of Computer Science & Technology at Nanjing University, China and currently is a Professor. She was selected in the Program for New Century Excellent talents in University, Ministry of Education in 2009. Her research interests are mainly in artificial intelligence, machine learning, and data mining. She has published over 50 papers in leading international/national journals and conferences.

Zhi-Hua Zhou (S'00-M'01-SM'06-F'13) received the BSc, MSc and PhD degrees in computer science from Nanjing University, China, in 1996, 1998 and 2000, respectively, all with the highest honors. He joined the Department of Computer Science & Technology at Nanjing University as an Assistant Professor in 2001, and is currently Professor and Standing Deputy Director of the National Key Laboratory for Novel Software Technology; he is also the Founding Director of the LAMDA group. His research interests are mainly in artificial intelligence, machine learning and data mining. He has authored the books *Ensemble Methods: Foundations and Algorithms* and *Machine Learning* (in Chinese), and published more than 150 papers in top-tier international journals or conference proceedings. He has received various awards/honors including the National Natural Science Award of China, the PAKDD Distinguished Contribution Award, the IEEE ICDM Outstanding Service Award, the Microsoft Professorship Award, etc. He also holds 22 patents. He is an Executive Editor-in-Chief of the *Frontiers of Computer Science*, Associate Editor-in-Chief of the *Science China Information Sciences*, Action or Associate Editor of the *Machine Learning*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* , *ACM Transactions on Knowledge Discovery from Data* , etc. He served as Associate Editor-in-Chief for *Chinese Science Bulletin* (2008-2014), Associate Editor for *IEEE Transactions on Knowledge and Data Engineering* (2008-2012), *IEEE Transactions on Neural Networks and Learning Systems* (2014-2017), *ACM Transactions on Intelligent Systems and Technology* (2009-2017), Neural Networks (2014-2016), *Knowledge and Information Systems* (2003-2008), etc. He founded ACML (Asian Conference on Machine Learning), served as Advisory Committee member for IJCAI (2015-2016), Steering Committee member for ICDM, PAKDD and PRICAI, and Chair of various conferences such as General co-chair of PAKDD 2014 and ICDM 2016, Program co-chair of SDM 2013 and IJCAI 2015 Machine Learning Track, and Area chair of NIPS, ICML, AAAI, IJCAI, KDD, etc. He is/was the Chair of the IEEE CIS Data Mining Technical Committee (2015-2016), the Chair of the CCF-AI(2012-), and the Chair of the Machine Learning Technical Committee of CAAI (2006-2015). He is a foreign member of the Academy of Europe, and a Fellow of the ACM, AAAI, AAAS, IEEE, IAPR, IET/IEE, CCF, and CAAI.