Supplemental Material for 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—This is the supplemental material for the manuscript "What makes objects similar: A unified multi-metric learning approach". We first present the detailed optimization process of UM^{2}L when we need to learn the threshold value. Then more explanations and experimental results are presented, namely, implementation details, feature pattern discovery, ablation investigation, and multi-view detection. Last, we discuss the proposed reweighted solver for the symmetric $\ell_{2,1}$-norm and its convergence.

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

1 UM^{2}L Threshold Optimization

Multiple metrics $\mathcal{M}_K = \{M_1, \ldots, M_K\}$ could be used to explain various semantic meanings between objects. For (squared) Mahalanobis distance between a pair $(\mathbf{x}_i, \mathbf{x}_j)$ with metric M_k, i.e.,

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

the overall similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ could be defined as:

$$f_{q_{ij}}(\mathbf{x}_i, \mathbf{x}_j) = \kappa_{q_{ij}}(-\mathrm{Dis}^2_{M_1}(\mathbf{x}_i, \mathbf{x}_j), \ldots, -\mathrm{Dis}^2_{M_K}(\mathbf{x}_i, \mathbf{x}_j)) \ .$$

$\kappa_{q_{ij}}$ is an operator combining/selecting K similarity values (negative distances) together. Given totally P pairwise side information $\mathcal{P} = \{(\mathbf{x}_i, \mathbf{x}_j, q_{ij})\}$, and the value $q_{ij} \in \{-1, 1\}$ reveals whether two instances are similar or not. The pairwise version of our Unified Multi-Metric Learning framework can be formalized as

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

In Eq. 2, $(i, j) \in \mathcal{P}$ enumerate all pairs in $\mathcal{P}$; $\ell(\cdot)$ is a loss function, the lower the better; $\Omega(\cdot)$ is the metric regularizer; and $\gamma > 0$ is a threshold value.

Using the objective in Eq. 2, the overall similarity values between similar objects should be larger than γ; while for a dissimilar pair, their similarity value should be small. The value of γ could be fixed in advance, or learned in the training process.

As described in the manuscript, 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. It is notable that in this sub-problem, we can optimize over threshold γ at the same time.

With the smooth hinge loss, the gradient of loss function w.r.t. threshold parameter γ can be computed by

$$\begin{aligned}\frac{\partial \ell(\mathcal{M}_K)}{\partial \gamma} &= \frac{1}{P} \sum_{\tau=(i,j)\in\mathcal{P}} \frac{\partial \ell(q_{ij}(-\langle M_{k^*_\tau}, A_{ij}\rangle - \gamma))}{\partial \gamma} \\ &= \frac{1}{T} \sum_{\tau=(i,j)\in\mathcal{P}} \frac{\partial \ell(a_\tau)}{\partial \gamma} = \frac{1}{T} \sum_{\tau=(i,j)\in\mathcal{P}} \nabla^\tau_\gamma(a_\tau) \ .\end{aligned} \quad (3)$$

The gradient value $\nabla^\tau_{M_k}(a_\tau) = 0$ when $a_\tau \geq 1$; $\nabla^\tau_{M_k}(a_\tau) = (1 - a_\tau) q_{ij} \delta[k^*_\tau = k]$ when $0 < a_\tau < 1$; and $\nabla^\tau_{M_k}(a_\tau) = q_{ij}\delta[k^*_\tau = k]$ when $0 < a_\tau \leq 0$. The gradient computation of γ can be used in the accelerated projected gradient descent or the accelerated proximal gradient descent when solving UM^{2}L variants.

The learned γ helps the determination of similar relationship between two objects, e.g., in the social linkage discovering experiment.

2 Implementation Details

When optimizing UM^{2}L in an alternative style, we initialize the metric affiliation portion for each instance first. The affiliation portion depends on which metric in $\mathcal{M}_K$ to use. GMM or KMeans is conducted with the component number the same as K set in UM^{2}L. After that, each instance can be represented as a vector of length K. Comparison between these coding values indicates which component to illustrate the similar/dissimilar relationship. For instance, in Apical Dominating Similarity (ADS) with $\kappa_1 = \kappa_2 = \max(\cdot)$, given

• *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

BER↓	KM	SP	SCA	EGO	UM^{2}L
syn1	.382	.382	.392	.467	**.355**
syn2	.564	.564	.399	.428	**.323**
ad	.670	.670	.400	.583	**.381**
ccd	.244	.244	.250	.225	**.071**
my_movie	.370	.370	.249	.347	**.155**
reuters	.704	.704	.400	.609	**.398**

Table 1: BER (the lower the better) of feature pattern discovery comparisons on synthetic datasets: UM^{2}L$_{\mathrm{ADS}}$ vs. others

similar pair $(\mathbf{x}_i, \mathbf{x}_j)$ with coding vector $(\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_j)$ in $\mathbb{R}_+^K$, the initial metric of this pair is selected by:

$$k = \arg\max_k \min(\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_j) \ .$$

The selection can be interpreted as a two-stage process. First, similarities between instances based on K components are calculated in a Histogram Intersection Kernel (HIK) style, where instances are similar based on some metric only when both of their affiliations are not small w.r.t. a certain component. Second, a particular metric is selected based on the largest pair-similarity value.

There are three types of similarity defined in the main body, namely Apical Dominating Similarity (ADS), One Vote Similarity (OVS) and Rank Grouping Similarity (RGS), which have different optimization properties. The RGS is convex, and thus alternative approach can get global optimal solutions. The OVS is a non-convex one, whose results depend on initializations. The ADS is a semi-convex problem [1]. To better utilize the property of the semi-convex part that the whole objective is convex w.r.t. dissimilar pairs, we compute only the affiliation of similar pairs in the initialization phase. Since when these affiliations are fixed, the whole problem becomes a convex one, even there still exists a $\max(\cdot)$ operator.

3 Feature Pattern Discovering

To demonstrate the ability of linkage decomposition, we test feature pattern discovering the ability of UM^{2}L$_{\mathrm{ADS}}$ on 4 transformed multi-view datasets [2]. For each dataset, we first extract principal components of each view, and construct sub-linkage candidates between instances with random thresholds on each single view. Thus, these candidates are diverse among different views. After that, the overall linkage is further generated from these candidates using an "or" operation. It is notable that the linkage generation process is the same as the property of similarity linkages for UM^{2}L$_{\mathrm{ADS}}$. With features on each view and the overall linkage, the goal of feature pattern discovering is to reveal responsible features for each sub-linkage.

With multiple learned metrics, the zero-valued rows and columns of the learned metrics indicate irrelevant features in the corresponding group. It is both the intrinsic feature sparsity in linkage generation and the sparse property of $\ell_{2,1}$-norm makes the final multiple metrics sparse over rows/columns.

Syn1 and syn2 are purely synthetic datasets with features sampled from Uniform, Beta, Binomial, Gamma and Normal distributions using two sets of different parameters. Balanced Error Rate (BER) [3] between ground truth feature indicator sets and predicted sets are listed in Table 1. UM^{2}L$_{\mathrm{ADS}}$ achieves the best on all datasets. These assessments indicate UM^{2}L$_{\mathrm{ADS}}$ can figure out reasonable linkages or patterns hidden behind observations, and even better than domain specific methods.

4 Full results of Ablation Investigation

Before comparing UM^{2}L with others, we first analyze the classification performance of UM^{2}L variants: for different selection of κ, namely, ADS/OVS/RGS, and both the pairwise/triplet versions. The summation form $\kappa = \sum$ is also listed as a baseline. 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 over them is used for prediction.

In the pairwise implementation, 10 targets and impostors are selected based on Euclidean nearest neighbors to generate pairs. While for the triplet version, we use 3 target neighbors and 10 impostors for each instance as in the usual setting [4], [5]. The component number K is set the same as the number of classes. $\ell_{2,1}$-norm is used as the regularizer. For each dataset, the evaluations are repeated for 30 times. In each trial, 70% of instances are used for training, and the remaining part is for the test. For all comparison methods, we select the parameter λ using cross-validation. Performances of different λ are computed in the training set, and the λ with the best performance is used to train the whole training set and for the test. 3NN results (error mean $\pm$ std.) are listed in Table 2.

From Table 2, the triplet versions of UM^{2}L get better and more stable results in most cases, it may be the reason that more local information is utilized during training compared with the pairwise versions. OVS could not output comparable results in some cases, and sometimes the performance is not stable, since the explain of linkages in OVS is more for a decomposition goal, not for classification. It is able to find the possible semantic component in the object as shown in other subsections.

5 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 [6]. The hidden multi-view data [7] are composed of 200 instances and each instance has two hidden views, namely the color and the shape. We perform UM^{2}L$_{\mathrm{ADS/OVS/RGS}}$ on this dataset with $K = 2$. Trace norm regularizes the approach in this part to get low dimensional projections. UM^{2}L framework facilitates the understanding of data by decomposing each base metric to low dimensional subspace projector, i.e., for each base metric M_k, two eigenvectors $L_k \in \mathbb{R}^{d\times 2}$ corresponding to the largest 2 eigenvalues are picked. The results can be found in Fig. 1. We use

	Pairwise				Triplet		
κ	ADS	OVS	RGS	SUM	ADS	OVS	RGS
Automp	.205±.064	.377±.046	.272±.033	.237±.044	**.201±.034**	.265±.071	.225±.031
Clean1	.092±.022	.133±.024	.106±.022	.098±.023	**.070±.018**	.118±.022	.086±.020
German	**.277±.016**	.326±.102	.279±.028	.296±.018	.281±.019	.281±.022	.284±.030
Glass	.356±.054	.583±.091	.381±.054	.308±.043	.312±.043	**.284±.039**	.293±.047
Hayes-r	.327±.044	.307±.074	.321±.051	.293±.052	**.276±.044**	.293±.056	.307±.068
Heart-s	.206±.038	.309±.063	.196±.039	.226±.042	.190±.035	**.184±.034**	.194±.063
House-v	**.046±.017**	.103±.037	.066±.019	.057±.019	.051±.015	.049±.015	.048±.013
Liver-d	.393±.037	.454±.077	.423±.048	.440±.049	.363±.045	.401±.042	**.342±.047**
Segment	**.022±.005**	.068±.029	.063±.061	.025±.005	.023±.038	.037±.009	.029±.034
Sonar	.167±.040	.201±.045	.170±.041	.148±.033	.136±.032	.144±.031	**.132±.036**

Table 2: Test error comparison between different κ implementations in the UM^{2}L framework, with both pairwise and triplet variants.

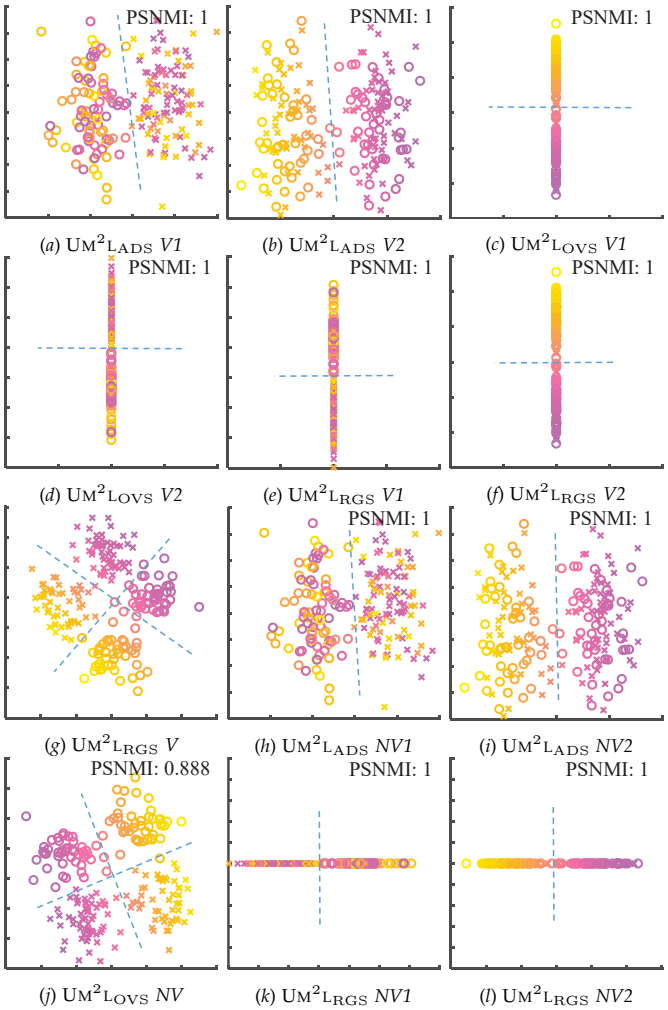


Figure 1: Subspaces discovered by UM^{2}L given instances with two semantic components, i.e., color and shape. Blue dot-lines give the possible decision boundary (best viewed in color). "V" means the discovered view, and "NV" means the noise perturbed case. Right upper corner of each plot shows the paired semantic NMI (PSNMI) value, a numerical measurement of multi-view discovery task, for a single projection, the higher the better (we do not compute PSNMI when there is only a single view projection.).

$V1$ and $V2$ to denote the projection results of a certain view. $NV1$ and $NV2$ show the case when we combine original features with random noise.

We can explain the linkages between objects among views using different UM^{2}L variants. ADS emphasizes the existence of relevant views and aims at decomposing helpful aspects or views; different from ADS, the OVS permits the reason for dissimilar objects also in a certain view; while RGS requires full accordance among views, the dissimilar and similar relationship should be kept in both views.

We set the parameter λ here as default value 1. When we increase the parameter, UM^{2}L is able to generate more sparse views (more information in a smaller number of views). For example, as in plot (g), there is only a single view for RGS output, which covers both color and shape semantics in one plot (although set to learn 2 metrics at first, the other one degenerate to zero after training).

From the results, it is also notable that ADS could discover two semantics even in noisy cases. The results of RGS are more discriminative than ADS. OVS outputs the same projections as RGS, but only one output in the noisy case. It could be the reason that OVS only requires a "weak" type of similarity generation. So when facing the noise, it concentrates the two semantics into one single view.

		MVTE	mmTSNE	SCA	ADS	OVS	RGS
No Noise	View1	.960	.929	1	1	N/A	1
	View2	.833	.029	.919	1	N/A	1
	Mean	.896	.479	.959	1	N/A	1
Noise	View1	N/A	N/A	.764	1	1	1
	View2	N/A	N/A	.503	1	1	1
	Mean	N/A	N/A	.633	1	1	1

Table 3: The Paired Semantic NMI (PSNMI) results for all comparison methods, for original and noisy cases. We use ADS/OVS/RGS to denote the case when using different κs in UM^{2}L. "N/A" in the table shows there is no applied result.

We use a novel criterion Paired Semantic NMI (PSNMI) to measure the multi-view discovery ability of the comparison methods, the higher the better. Results of all algorithms are listed in Table 3. The "N/A" shows there is no applied value. For example, MVTE and mmTSNE rely only on the linkage information, so the results will not change in the noise case. Besides, OVS outputs just one projection in the no noise case, so we do not compute the PSNMI value on it.

6 Reweighted Method on Symmetric $\ell_{2,1}$-norm

Solving $\ell_{2,1}$-norm under symmetric constrained can be transformed to the proximal sub-problem in

$$M'_k = \arg\min_{M \in \mathcal{S}_d} \frac{1}{2}\|M - V_k\|_F^2 + \lambda\|M\|_{2,1} \ . \quad (4)$$

We ignore the subscript index k in the following discussions. Since $\ell_{2,1}$-norm is the sum of ℓ_2-norm on each row of M, which violates the symmetric property, directly optimizing may be time-consuming in some cases [8]. Taking symmetry into consideration, we reformulate the problem in Eq.4 as follows:

$$M = \arg\min_{M \in \mathcal{S}_d} \frac{1}{2}\|M - V\|_F^2 + \frac{\lambda}{2}\|M\|_{2,1} + \frac{\lambda}{2}\|M^\top\|_{2,1} \ . \quad (5)$$

The impact of $\ell_{2,1}$-norm is shared on M and $M^\top$ equally. Taking derivative of Eq 5 w.r.t. M and setting it to zero get:

$$M - V + \frac{\lambda}{2}D_1 M + \frac{\lambda}{2}M D_2 = 0 \ . \quad (6)$$

Both D_1 and D_2 are diagonal matrices of size $d \times d$, and the (r,r)-th elements in D_1 and D_2 are $a_r^1 = \frac{1}{2\|\boldsymbol{m}_r\|_2}$ and $a_r^2 = \frac{1}{2\|\boldsymbol{m}^r\|_2}$, respectively. Thus, D_1 and D_2 consider ℓ_2-norm values from *both rows and columns* of M. Eq. 6 is a Sylvester equation, which has a high computational cost when solved with off-the-shelf tools. To accelerate, we consider the closed form results based on Kronecker product $\otimes$, and comes to the result of Lemma 1:

$$vec(M) = (I \otimes (I + \frac{\lambda}{2}D_1) + (\frac{\lambda}{2}D_2^\top \otimes I))^{-1} vec(V) \ . \quad (7)$$

$vec(\cdot)$ means the vectorization of a matrix. It is notable that all terms, i.e., D_1, D_2 and identity matrix I, in the inverse operation are in diagonal forms. So the inverse is not on a matrix but on a *scalar*. The (r,c)-element in M is:

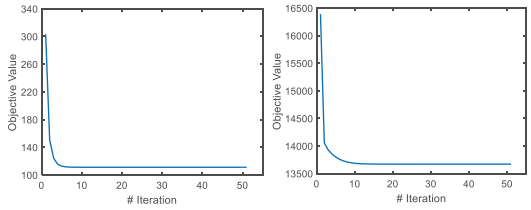
$$M_{rc} = V_{rc}/((1 + \frac{\lambda}{2}a_c^1) + \frac{\lambda}{2}a_r^2) \ . \quad (8)$$

Since metric M keeps symmetry in each iteration, this update flow is symmetric projection free. Thus, $a_r^1 = a_r^2$, which further simplifies the computation. Since D_1 and D_2 in the closed form update of M in Eq. 7 also depend on M, they should be updated alternatively. The convergence of this reweighted method can be easily proved [9] and is validated in our experiments in section 7.5.

6.1 Convergence of Solving Symmetric $\ell_{2,1}$-norm

We validate the convergence property of the reweighted method proposed in Lemma 1 when solving $\ell_{2,1}$-norm with symmetric constraint. Given different sizes of input metrics, we test the change of the objective function in Eq. 4. Two sizes of metrics are showed in Fig. 2: the first (*a*) is a 20×20 metric and the second (*b*) is of size 200.

It can be found that: 1. the reweighted method converges at last; 2. this method converges very quickly and the time to converge may depend on the size of the input. The larger the metric, the more iterations it needs to converge. When the dimension is low, it can converge in about 5 iterations. While with a larger input of size 200, the objective value converges in about 10 iterations. In addition, on the larger size input, time comparison is conducted between the reweighted method and the symmetric iterative projection one in [8]. We force the two methods to achieve nearly the same objective value in Eq. 4 at last and the time comparison is conducted 100 times in total. The average convergence time (in second) for reweighted one and symmetric projection one are 0.0093 (0.0012) and 0.0665 (0.0023) respectively. Values in brackets are standard deviations of time. This result reveals our reweighted solver can be more efficient when dealing with $\ell_{2,1}$-norm regularization on metrics, since at each step it takes norm on row and column into consideration simultaneously and thus is symmetric projection free.



(*a*) *small metric* (*b*) *large metric*

Figure 2: Convergence results of the reweighted method on symmetric $\ell_{2,1}$-norm. Changes of objective values on two different sizes of input metrics are showed.

References

[1] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[2] H.-J. Ye, D.-C. Zhan, Y. Miao, Y. Jiang, and Z.-H. Zhou, "Rank consistency based multi-view learning: A privacy-preserving approach," in *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, Melbourne, Australia, 2015, pp. 991–1000.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint l2,1-norms minimization," in *Advances in Neural Information Processing Systems 23*. Curran Associates, Inc., 2010, pp. 1813–1821.