Hierarchical Relationship Alignment Metric Learning

Lifeng Gu [1]

Abstract

Most existing metric learning methods focus on learning a similarity or distance measure relying on similar and dissimilar relations between sample pairs. However, pairs of samples cannot be simply identified as similar or dissimilar in many real-world applications, e.g., multi-label learning, label distribution learning. To this end, relation alignment metric learning (RAML) framework is proposed to handle the metric learning problem in those scenarios. But RAML learn a linear metric, which can't model complex datasets. Combining with deep learning and RAML framework, we propose a hierarchical relationship alignment metric leaning model HRAML, which uses the concept of relationship alignment to model metric learning problems under multiple learning tasks, and makes full use of the consistency between the sample pair relationship in the feature space and the sample pair relationship in the label space. Further we organize several experiment divided by learning tasks, and verified the better performance of HRAML against many popular methods and RAML framework.

1. Introduction

In many computer vision and pattern recognition tasks, e.g., face recognition (Guillaumin et al., 2009), image classification (Mensink et al., 2012), and person re-identification (Liao et al., 2015), it is crucial to learn a discriminative distance metric to measure the similarity between pairs of samples. However, for some learning tasks, e.g., multi-label learning (Zhang & Wu, 2015) and label distribution learning (Geng, 2016), relations between sample pairs cannot be simply identified as similar or dissimilar. Thus, the existing metric learning methods cannot work on the above tasks. The problem arises that it is difficult to classify two images into similar or dissimilar sample pair. Above discussions encourage us to propose a generalized metric learning method, which can be flexibly adopted to various kinds of tasks. RAML (Zhu & Yang, 2018) framework was proposed to handle the problem. Combing with deep neural networks and RAML, we propose a hierarchical relationship alignment metric leaning model HRAML, which uses the concept of relationship alignment to model metric learning problems under multiple learning tasks, and makes full use of the consistency between the sample pair relationship in the feature space and the sample pair relationship in the label space.

2. Related works

deep metric learning deep metric learning want to learn a good metric to measure the similarity of samples. Contrative loss (Hadsell et al., 2006) was used to combine with Siamese network and achieved good results. The loss of two-tuples can be used to model several problems. It is also a simple and feasible method. The sampling complexity is $o(n^2)$, easier than many subsequent variants. Wen et al. (2016) combines softmax and center loss for face recognition. Center loss is used to minimize the distance from the sample representation to the center of the category. Ge (2018) combined the triple loss and hierarchical tree to propose a new loss function. He used the hierarchical tree to encode the context of the sample, and then used the hierarchical tree to calculate the intra-class distance and the inter-class distance. Sample the more suitable triples, and finally use the dynamic interval mechanism to form the loss function. Wang et al. (2017) proposed a novel angle loss to improve the discriminative ability of sample representation. Different from the two-tuple and triple-tuple loss, the angle loss uses the angle of the sample to the center of the category to impose constraints. The angle constraint has rotation and scaling invariance. It naturally encodes the geometric relationship between the three samples. You can set the angle The size to change the constraint strength has the advantage of being very robust. The slow convergence speed of binary or ternary loss is mainly due to the fact that only one negative sample is compared in each update and other negative samples are ignored. Npair loss (Sohn, 2016) improves this point, and compares batches in each update All other negative samples in each update. Circle loss (Sun et al., 2020) was proposed, Circle loss re-weights the similarity between samples to highlight the incomplete sample similarity. Circle loss unifies two basic learning methods

Affiliations [1]Tju
Correspondence Lifeng Gu - gulifeng666@163.com

in the field of feature learning: learning based on category labels and learning based on pair label. Cakir et al. (2019) proposed FastAP, which learns features by optimizing the average accuracy of the ranked list. FastAP connects feature learning and ranking problems together, opening up new ideas.

3. Background

Zhu & Yang (2018) proposed the concept of relational alignment in the field of metric learning. We briefly describes it as follows: For metric learning, pairwise constraints are often used to describe the sample pair relationship in the decision space. Here, we introduce relation alignment learning to metric learning. $f(\mathbf{x}_i, \mathbf{x}_j, \mathbf{M}, b)$ is used to measure the sample relationship in the feature space, $g(\mathbf{y_i}, \mathbf{y_j})$ are used to measure the sample relationship in the decision space. $g(\mathbf{y_i}, \mathbf{y_j})$ is specifically designed to handle different tasks. In addition, $\mathbf{A} \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$ and $\mathbf{E} \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$ are used as The sample relationship matrix between the feature space and the decision space, the relationship alignment means The sample relationship of the feature space should be consistent with the sample relationship of the decision space. This is a wide-ranging idea,

$$\begin{bmatrix} a_{11} & ... & a_{i1} & ... & a_{n1} \\ ... & ... & ... & ... & ... \\ a_{1i} & ... & a_{ii} & ... & a_{ni} \\ ... & ... & ... & ... & ... \\ a_{1n} & ... & a_{in} & ... & a_{nn} \end{bmatrix} = \begin{bmatrix} e_{11} & ... & e_{i1} & ... & e_{n1} \\ ... & ... & ... & ... & ... \\ e_{1i} & ... & e_{ii} & ... & e_{ni} \\ ... & ... & ... & ... & ... \\ e_{1n} & ... & e_{in} & ... & e_{nn} \end{bmatrix}$$

a_{ij} and e_{ij} represent the relationship between the sample $\mathbf{x}_i$ and $\mathbf{x}_j$ in the feature space and the decision space. In order to maintain consistency, here is a requirement

$$f(\mathbf{x}_i, \mathbf{x}_j, \mathbf{M}, b) = g(\mathbf{y}_i, \mathbf{y}_j). \tag{1}$$

$g(\mathbf{y_i}, \mathbf{y_j})$ represents the different degrees of the two samples in the decision space. $g(\mathbf{y}_i, \mathbf{y}_j)$ reflects the relationship between sample pairs in the decision space and is used to guide the learning in the feature space $(\mathbf{M}, b)$.

$$\begin{aligned} f(\mathbf{x}_i, \mathbf{x}_j, \mathbf{M}, b) &= (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j) + b \\ &= \langle \mathbf{M}, \mathbf{T}_{ij} \rangle + b \end{aligned} \tag{2}$$

$\langle \cdot, \cdot \rangle$ is defined as the inner product of two Frobenius matrices, b is the offset, and $\mathbf{T}_{ij} = (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T$. (1) can be rewritten as

$$g(\mathbf{y}_i, \mathbf{y}_j) = \langle \mathbf{M}, \mathbf{T}_{ij} \rangle + b \tag{3}$$

Once the relational function $g(\mathbf{y_i}, \mathbf{y_j})$ is selected, the formula (3) can be regarded as a linear regression problem. Here, the metric learning problem is transformed into solving a sample pair regression problem, and the input is a sample pair $(\mathbf{x}_i, \mathbf{x}_j)$.

4. HRAML

Given sample pairs $\mathbf{x}_i$ and $\mathbf{x}_j$, after being modeled by a neural network of $M + 1$ layer, they can finally be formalized as $f(\mathbf{x_i}) = \mathbf{h_i^{(M)}} = \mathbf{s}(\mathbf{W^{(m)}}^\mathbf{T} \mathbf{x_i^{(m-1)}} + \mathbf{b^{(m)}})$ and $f(\mathbf{x_j}) = \mathbf{h}_j^{(M)} = s(\mathbf{W}^{(m)^T} \mathbf{x}_j^{(m-1)} + \mathbf{b}^{(m)})$, where the mapping $s : \mathbb{R} \to \mathbb{R}$ is a nonlinear activation function, and m is the number of layers. $\mathbf{x_i}^{(m)} = \mathbf{W}^{(m)^T} \mathbf{x}_i^{(m-1)} + \mathbf{b}^{(m)}$.

Given a sample pair x_i and x_j, using the idea of relationship alignment, the alignment relationship between the sample pair in the feature space and the decision space can be formalized as:

$$R(f(x_i), f(x_j)) = g(y_i, y_j) \tag{4}$$

Where R represents the sample pair relation function in the feature space, and g represents the relation function in the feature space. By specifying the relationship functions R and g, different relationships specifically used for alignment in the idea of relationship alignment can be expressed. We defines the sample pair relationship of the feature space as the distance between them: $R(f(x_i), f(x_j)) = D_{ij} = \|f(\mathbf{x_i}) - f(\mathbf{x_j})\|_2$, and then use the mean square error loss optimize formula (4) to derive the objective function of HRAML:

$$\min J = \frac{1}{4} \sum\nolimits_{(i,j)}^{n} (D_{ij}^2 - g(y_i, y_j))^2 + r(\theta) \tag{5}$$

$r(\theta) = \lambda \sum_{m=1}^{M} \left(\left\| \mathbf{W}^{(m)} \right\|_F^2 + \left\| \mathbf{b}^{(m)} \right\|_2^2 \right)$ is a regularization item, $g(y_i, y_j)$ is relation function. In order to optimize (5), we use stochastical gradient descent and we have flowing equations:
for layer $m = M$ in neural network:

$$\begin{aligned} \frac{\partial J(i,j)}{\partial \mathbf{W}^{(m)}} &= \left(D_{ij}^2 - g(y_i, y_j)\right) \Big(\frac{\partial s(\mathbf{x}_i^{(m)})}{\partial \mathbf{x}_i^{(m)}} \left(f(\mathbf{x}_i) - f(\mathbf{x}_j)\right) \\ &\qquad \mathbf{h}_i^{(m-1)^T} - \frac{\partial s(\mathbf{x}_j^{(m)})}{\partial \mathbf{x}_j^{(m)}} \left(f(\mathbf{x}_i) - f(\mathbf{x}_j)\right) \mathbf{h}_j^{(m-1)^T} \Big) \\ &= \delta_i^{(m)} \mathbf{h}_i^{(m-1)^T} - \delta_j^{(m)} \mathbf{h}_j^{(m-1)^T} \end{aligned} \tag{6}$$

$$\begin{aligned} \frac{\partial J(i,j)}{\partial \mathbf{b}^{(m)}} &= \left(D_{ij}^2 - g(y_i, y_j)\right) \Big(\frac{\partial s(\mathbf{x}_i^{(m)})}{\partial \mathbf{x}_i^{(m)}} - \frac{\partial s(\mathbf{x}_j^{(m)})}{\partial \mathbf{x}_j^{(m)}} \Big) \\ &\qquad \left(f(\mathbf{x}_i) - f(\mathbf{x}_j)\right) = \delta_i^{(m)} - \delta_j^{(m)} \end{aligned} \tag{7}$$

Algorithm 1 The algorithms of our proposed HRAML

```
Require:
    Training data $\mathbf{X} \in \mathbb{R}^{\mathbf{d}\times\mathbf{m}}$, where $d$ and $m$ are the numbers of feature dimension and samples, respectively. number of network layers $M$, learning rate $\mu$, iterative number $T$
Generate sample pairs $(\mathbf{x}_{i1}, \mathbf{x}_{i2})$, $i = 1, 2, ..., n$.
Compute sample relation $g(\mathbf{x}_{i1}, \mathbf{x}_{i2})$, $i = 1, 2, ..., n$.
repeat
    Randomly select a pair$(\mathbf{x_i}, \mathbf{x_j})$ from sample pairs.
    Set $\mathbf{h}_i^0$= $\mathbf{x}_i$, $\mathbf{h}_j^0$ = $\mathbf{x}_j$.
    for m = 1,2,...,M do
        Do forward propagation to get $\mathbf{h}_i^m$ and $\mathbf{h}_j^m$.
    end for
    for m = M,M-1,...,1 do
        Do back propagation to get $\frac{\partial J(i,j)}{\partial \mathbf{W}^{(m)}}$ and $\frac{\partial J(i,j)}{\partial \mathbf{b}^{(m)}}$ by (6),(7),(8),(9)
    end for
    update $\{\mathbf{W}^m, \mathbf{b}^m\}, m = 1, 2, \ldots, M$ by gradient descent.
until converge
Ensure:
    Weights and biases: $\{\mathbf{W}^m$, $\mathbf{b}^m\}$, m=1,2,...,M
```

for layer $m = 1, 2 \ldots M - 1$ in neural network:

$$\begin{aligned}\frac{\partial J(i,j)}{\partial \mathbf{W}^{(m)}} &= \frac{\partial s(\mathbf{x}_i^{(m)})}{\partial \mathbf{x}_i^{(m)}} \mathbf{W}^{(m+1)^T} \frac{\partial J(i,j)}{\partial \mathbf{x_i}^{(m+1)}} \mathbf{h}_i^{(m-1)^T} \\ &- \frac{\partial s(\mathbf{x}_j^{(m)})}{\partial \mathbf{x}_j^{(m)}} \mathbf{W}^{(m+1)^T} \frac{\partial J(i,j)}{\partial \mathbf{x}_j^{(m+1)}} \mathbf{h}_j^{(m-1)^T} \\ &= \delta_i^{(m)} \mathbf{h}_i^{(m-1)^T} - \delta_j^{(m)} \mathbf{h}_j^{(m-1)^T}\end{aligned} \quad (8)$$

$$\begin{aligned}\frac{\partial J(i,j)}{\partial \mathbf{b}^{(m)}} &= \frac{\partial s(\mathbf{x}_i^{(m)})}{\partial \mathbf{x}_i^{(m)}} \mathbf{W}^{(m+1)^T} \frac{\partial J(i,j)}{\partial \mathbf{x}_i^{(m+1)}} - \frac{\partial s(\mathbf{x}_j^{(m)})}{\partial \mathbf{x}_j^{(m)}} \\ &\mathbf{W}^{(m+1)^T} \frac{\partial J(i,j)}{\partial (\mathbf{x}_j^{(m+1)})} = \delta_i^{(m)} - \delta_j^{(m)}\end{aligned} \quad (9)$$

The corresponding metric learning HRAML algorithm is summarized in Algorithm 1.

5. Implementation Details

5.1. Training Details

Due to the diversification of the training data that needs to be processed,we uses general mlp as the encoder and tanh as the activation function. In addition, we normalizes the output of the network so that a better performance can be obtained (Wu et al., 2017), the normalized form is

$$f(x) = \frac{f(x)}{\|f(x)\|} \quad (10)$$

f(x) is the output of the network.
We initializes $\mathbf{b}^m, m = 1, 2, \ldots, M$ is 0, and $\mathbf{W}^m, m = 1, 2,\ dots, M$ is initialized to the distribution $U[-0.2, 0.2]$.

5.2. Relation function

In the feature space, we need to use differentiable functions as the relation function to ensure the stability and ease of learning. We uses ordinary Euclidean distance as the relation function. In the decision space, we can imitate the feature space and use the function with parameters, which will make HRAML method has a broader form and can be linked to more methods, but for the convenience of the sample selection stage, we are consistent with (Zhu & Yang, 2018), using a fixed function without parameters as the decision space Function, we consider the l_1 norm: Let $\mathbf{y}_i$ and $\mathbf{y}_j$ represent label of $\mathbf{x}_i$ and $\mathbf{x}_j$. Relation function will be :

$$g(\mathbf{y}_i, \mathbf{y}_j) = \|\mathbf{y}_i - \mathbf{y}_j\|_1 \quad (11)$$

where $\|\mathbf{a}\|_1$ is l_1-norm of item $\mathbf{a}$.

5.3. Sample selection problem

In the process of neural network learning, sample selection is very important and has a great impact on the results. We can uses difficult sample mining (Shrivastava et al., 2016).

6. Experiment

In this section, we conduct experiments to validate the performance of the proposed HRAML. We consider three applications, including single-label classification, multi-label classification, label distribution learning . The following part will be organized as the corresponding parts.

6.1. Classification

Experiment setup. Following (Zhu & Yang, 2018), we use "S/F/C" represents the number of samples, features and classes. We compare our method with popular methods, e.g., ITML (Davis et al., 2007), LMNN (Weinberger & Saul, 2009), DML (Ying & Li, 2012), DoubletSVM (DSVM) (Wang et al., 2015),GMML (Zadeh et al., 2016) and RAML(Zhu & Yang, 2018). For fair comparison, the parameters of all compared methods are set as the default setting of the original references.

Experimental analysis. Table 1 list the classification accuracy of different metric learning methods on many datasets, respectively, where the best results are marked in bold face. HRAML get the best performance, it beat others methods, not only linear methods but also kernel methods based nolinear methods. HRAML has a clear and sample objective function Under classification task, neural network can optimise easily and learn discriminate metric.

	Data	emotion	flags	corel800
MLKNN	Hamming Loss↓	0.2137	0.3099	0.0137
	Ranking Loss↓	0.1729	0.2228	0.1888
	One Error↓	0.3317	0.2154	0.6825
	Coverage↓	1.9158	3.8154	88.5100
	Average Precision↑	0.7808	0.8084	0.3276
RAML-SVR	Hamming Loss↓	**0.2054**	**0.2967**	**0.0135**
	Ranking Loss↓	**0.1577**	**0.2179**	**0.1882**
	One Error↓	**0.2376**	**0.2000**	**0.6425**
	Coverage↓	**1.8960**	**3.8115**	**88.2350**
	Average Precision↑	**0.8101**	**0.8128**	**0.3386**
RAML-KRR	Hamming Loss↓	**0.2046**	**0.2967**	**0.0134**
	Ranking Loss↓	**0.1382**	**0.2113**	**0.1888**
	One Error↓	**0.2574**	**0.2000**	**0.6550**
	Coverage↓	**1.7327**	**3.7692**	**88.5100**
	Average Precision↑	**0.8225**	**0.8112**	**0.3388**
HRAML	Hamming Loss↓	0.2060	**0.2791**	**0.0135**
	Ranking Loss↓	0.1690	**0.2021**	**0.1861**
	One Error↓	**0.2871**	**0.2154**	**0.6075**
	Coverage↓	1.8861	**3.6769**	87.6025
	Average Precision↑	**0.8052**	**0.8244**	**0.3582**

Table 2. results on muti-label datasets

.

Criterion	Chebyshev↓	Clark↓	Canberra↓	Cosine↑	Intersection↑
AAKNN	0.3261	1.8448	4.3412	0.6905	0.5506
RAML-SVR	**0.3102**	**1.6986**	**3.8576**	**0.7051**	**0.5739**
RAML-KRR	**0.3139**	**1.6865**	**3.8419**	**0.7057**	**0.5743**
HRAML	**0.2907**	**1.7664**	**4.0922**	**0.7321**	**0.5929**

Table 3. results on Nature Scene dataset.

6.2. Multi-label Learning

Dataset. Like (Zhu & Yang, 2018), we use three multi-label datasets to evaluate performance[1], emotion (Trohidis et al., 2008), flags, and corel800 dataset (Hoi et al., 2006).

Evaluation Method. We use the performance of the MLKNN algorithm to evaluate the discriminative ability of the learned metric, We use multiple popular multi-label learning indicators used in (Zhu & Yang, 2018) to evaluate the permormance of MLKNN. The up arrow represents the higher the indicator, the better, and the down arrow represents the lower the indicator, the better.

Experimental Analysis. Table 2 shows the results on multi-label datasets, Compared with the original performance of MLkNN, RAML and HRAML learn more discriminative metric in multi-learning task. Compared with RAML algorithm, HRAML has better experimental results. Thanks to the improvement of the encoder, it has better performance than RAML. Beyond linear transform or kernel methods, HRAML has more power to extract features. And it can distinguish samples under different labels well through its optimization goals, .

6.3. Label Distribution Learning

dataset. We use the Nature Scene dataset used in (Zhu & Yang, 2018).
Evaluation Method. We use the performance of AAKNN to evaluate the learned metric, and use AAKNN and RAML as comparison algorithms. We use multiple evaluation indicators used in (Zhu & Yang, 2018) as the evaluation indicators. In order to maintain consistency with other experimental parts, We use (11) as the relationship function of the decision space. Of course, some better distribution functions such as kl divergence can be used as the relationship function, which should achieve better effect. **Experimental analysis.** Table 3 shows the comparison of AAKNN, RAML and HRAML. Thanks to the HRAML's encoder and optimization goal, it's performance is better than RAML.

7. Conclution

This paper proposes a hierarchical relationship alignment model HRAML, which uses the concept of relationship alignment to model metric learning problems under multiple learning tasks, and makes full use of the consistency between the sample pair relationship in the feature space and the sample pair relationship in the label space. Finally, the performance of HRAML is verified under a variety of tasks.

References

Cakir, F., He, K., Xia, X., Kulis, B., and Sclaroff, S. Deep metric learning to rank. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1861–1870, 2019.

Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. Information-theoretic metric learning. In *ICML*, pp. 209–216, 2007.

Ge, W. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 269–285, 2018.

Geng, X. Label distribution learning. *IEEE TKDE*, 28(7): 1734–1748, 2016.

Guillaumin, M., Verbeek, J., and Schmid, C. Is that you? metric learning approaches for face identification. In *ICCV*, pp. 498–505. IEEE, 2009.

Hadsell, R., Chopra, S., and LeCun, Y. Dimensionality reduction by learning an invariant mapping. In *2006*

[1] http://mulan.sourceforge.net/datasets-mlc.html

Data	S/F/C	ITML	LDML	LMNN	DSVM	GMML	DML	RAML-SVR	RAML-KRR	HRAML
binalpha	1404/320/36	0.6303±0.0501	0.6542±0.0317	0.6112±0.0358	0.5625±0.0322	0.5338±0.1986	0.5063±0.0251	**0.7250±0.0348**	**0.6850±0.0351**	**0.7243±0.0262**
caltech101	8641/256/101	**0.5803±0.0162**	0.5528±0.0157	0.5795±0.0126	0.5584±0.0159	0.5500±0.0117	0.3936±0.0123	**0.5855±0.0095**	**0.5803±0.0147**	**0.6015±0.0161**
MnistDat	3495/784/10	0.8695±0.0142	0.8858±0.0124	0.8721±0.0255	0.8848±0.0194	0.8589±0.0171	0.8323±0.0239	**0.9019±0.0175**	**0.9087±0.0137**	**0.8983±0.0168**
Mpeg7	1400/6000/70	0.8214±0.0333	0.7971±0.0365	0.8253±0.0232	0.8271±0.0353	**0.8429±0.0228**	0.7071±0.0267	**0.8450±0.0305**	0.7936±0.0341	0.8043±0.0248
news20	3970/8014/4	0.8678±0.0200	0.8816±0.0145	0.8734±0.0290	0.8594±0.0159	0.8647±0.0143	0.8166±0.0222	**0.9025±0.0132**	**0.9217±0.0145**	**0.9436±0.0133**
TDT2_20	1938/3677/20	0.9587±0.0358	0.9531±0.0306	0.9352±0.0197	0.9499±0.0175	0.9437±0.0275	0.6333±0.0176	**0.9679±0.0244**	**0.9762±0.0164**	**0.9860±0.0154**
uspst	2007/256/10	0.8979±0.0261	0.9084±0.0243	0.9096±0.0217	0.9125±0.0172	0.8858±0.0168	0.8030±0.0330	**0.9525±0.0147**	**0.9447±0.0157**	**0.9537±0.0202**

Table 1. classification accuracy on many datasets

IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 2, pp. 1735–1742. IEEE, 2006.

Hoi, S. C., Liu, W., Lyu, M. R., and Ma, W.-Y. Learning distance metrics with contextual constraints for image retrieval. In *CVPR*, volume 2, pp. 2072–2078, 2006.

Liao, S., Hu, Y., Zhu, X., and Li, S. Z. Person re-identification by local maximal occurrence representation and metric learning. In *CVPR*, pp. 2197–2206, 2015.

Mensink, T., Verbeek, J., Perronnin, F., and Csurka, G. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. *ECCV*, pp. 488–501, 2012.

Shrivastava, A., Gupta, A., and Girshick, R. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 761–769, 2016.

Sohn, K. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29:1857–1865, 2016.

Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., and Wei, Y. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6398–6407, 2020.

Trohidis, K., Tsoumakas, G., Kalliris, G., and Vlahavas, I. P. Multi-label classification of music into emotions. In *ISMIR*, volume 8, pp. 325–330, 2008.

Wang, F., Zuo, W., Zhang, L., Meng, D., and Zhang, D. A kernel classification framework for metric learning. *IEEE TNNLS*, 26(9):1950–1962, 2015.

Wang, J., Zhou, F., Wen, S., Liu, X., and Lin, Y. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2593–2601, 2017.

Weinberger, K. Q. and Saul, L. K. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009.

Wen, Y., Zhang, K., Li, Z., and Qiao, Y. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pp. 499–515. Springer, 2016.

Wu, C.-Y., Manmatha, R., Smola, A. J., and Krahenbuhl, P. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2840–2848, 2017.

Ying, Y. and Li, P. Distance metric learning with eigenvalue optimization. *Journal of Machine Learning Research*, 13 (Jan):1–26, 2012.

Zadeh, P., Hosseini, R., and Sra, S. Geometric mean metric learning. In *ICML*, pp. 2464–2471, 2016.

Zhang, M.-L. and Wu, L. Lift: Multi-label learning with label-specific features. *IEEE TPAMI*, 37(1):107–120, 2015.

Zhu, Pengfei, R. Q. Q. H. Q. W. C. Z. and Yang, L. Beyond similar and dissimilar relations.: A kernel regression formulation for metric learning. In *IJCAI*, pp. 3242–3248, 2018.