动态生成难样本的度量学习算法

韩 露[1]
HAN Lu

摘 要 度量学习算法的性能在很大程度上受样本构建的约束影响，通常情况下由难样本构造的约束越多模型性能会越好，但目前大部分度量学习算法挖掘到的难样本非常少，从而导致学习的度量判别力不高。为了解决这一问题，文章提出了一种动态生成难样本的度量学习算法（metric learning algorithm for dynamically generating of hard sample，SGML），算法主要思想是在数据集原有的异类样本中间生成新样本，以此生成的样本更难区分，从而提升模型的判别力。在 UCI 数据集上进行准确率以及参数灵敏度分析的相关实验，结果表明 SGML 算法可以提升模型的判别力和健壮性。

关键词 难样本；马氏距离；生成样本；度量学习

doi：10.3969/j.issn.1672-9528.2025.02.029

0 引言

在机器学习领域，分类算法的性能在很大程度上受度量样本间距离函数的影响，传统的机器学习算法根据欧式距离计算样本相似度，往往会忽略不同维度之间的相关性和量纲不同带来的问题。因此，“学习”一个度量函数来更好地度量样本间的距离成为机器学习的重要研究分支，也就是度量学习。

度量学习在攻击检测 [1-2]、推荐系统 [3-4]、图像分类 [5-6] 等领域都有十分广泛的应用。目前，绝大多数的度量学习方法都是学习马氏距离，即学习一个半正定矩阵 $\boldsymbol{M}$，来解决欧氏距离缺点。度量学习的核心思想是学习一个距离函数，使得标签相同的样本间距离尽可能小，标签不同的样本间的距离尽可能大，以此提高分类算法的性能。

据了解，样本类别信息构造的约束是影响学习度量性能的指标之一。目前大多数度量学习方法都是根据数据集原有的真实样本构造约束，不论是构造二元约束还是三元约束，约束数量巨大，但能够为度量的学习提供帮助的约束数量太少，究其原因是难样本个数过少导致。为解决难样本数目过少的问题，本文提出在数据集原有异类样本中间生成新样本的方法，提高算法判别力和健壮性。

1 相关工作

Xing 等人 [7] 提出了 MMC 算法，根据构造的二元约束，学习一个度量函数，使得标签相同的样本聚集在一起，标签不同的样本距离尽可能大。以迭代投影的方式满足所有约束，利用梯度上升进行求解，这也是度量学习被首次提出，但该算法利用全部样本构造约束，使得构造的约束数量过多，求解速度过慢。Weinberger 等人 [8] 提出了 LMNN 算法，选取距离目标样本最近的 k 个同类样本及异类样本构造三元约束，在一定程度上可以减少约束的数量。算法的主要思想是最大化 $d_M^2(x_i, x_k)-d_M^2(x_i, x_j)$，$x_i$、$x_j$ 为标签相同的两个样本，x_i、x_k 为标签不同的两个样本，但是该算法构建的约束数量仍然很大，导致模型收敛速度缓慢。为了提升算法效率，Davis 等人 [9] 提出利用信息理论方法来学习马氏距离度量矩阵，利用 Bregman 优化方法进行求解，该方法相较之前的方法，缩短了运行时间。Ying 等人 [10] 提出 DML-eig 算法，该算法通过最小化对称矩阵最大特征值的方法来进行学习，对称矩阵为 $\sum_{\tau\in D}u_\tau\tilde{\boldsymbol{X}}_\tau$，其中 u_τ 为不相似约束对应的权重，$\tilde{\boldsymbol{X}}_\tau=\boldsymbol{X}_s^{-1/2}\boldsymbol{X}_\tau\boldsymbol{X}_s^{-1/2}$，$\boldsymbol{X}_s$ 为相似约束构成的距离矩阵，$\boldsymbol{X}_\tau$ 为不相似约束构成的距离矩阵，目标就是找到合适的权重 u 使得上述的对称矩阵的最大特征值最小，该方法很好地把约束的权重学习和度量学习融合到一起。但是上述算法根据真实样本构造约束，能够为度量的学习提供的信息有限。为此，Chen 等人 [11] 将对抗的思想与度量学习算法结合起来，提出 AML 算法。其主要思想是通过在真实样本附近生成新的样本，来提高算法的性能，但该算法冗余度比较高，为了解决这一问题，本文提出一种全新的生成难样本的方法。

2 动态生成难样本的度量学习算法

传统的度量学习方法依赖于固定的训练样本集，这在数据量有限的情况下容易遇到性能瓶颈。本文提出的算法通过动态生成难样本来增强训练过程，从而使模型在有限数据下

1. 运城学院数学与信息技术学院 山西运城 044000
[基金项目] 运城学院院级科研项目（XJ2023001301）

能更有效地学习相似度度量。该算法由两部分构成，首先生成难样本，然后根据由难样本构造的约束去学习度量。

2.1 符号说明

$D=\{x_i,y_i\}_{i=1}^{N}$表示模型训练过程中用到的训练集。N表示训练集中样本的数量；$\boldsymbol{x}_i\in\mathbb{R}^d$表示训练集中第$i$个样本；$\boldsymbol{y}_i$表示样本的类别；$d$表示特征维数。$\boldsymbol{C}=\{(\boldsymbol{y}_i,\boldsymbol{y}_j)_c\}_{c=1}^{K}$表示不同标签两两结合构成的集合，其中$\boldsymbol{y}_i$表示训练集样本的标签；$(\boldsymbol{y}_i,\boldsymbol{y}_j)_c$表示集合$C$中的第$c$个不同标签结合构成的自己；$K$表示集合$c$中元素的个数。

2.2 样本生成方法

分别在每类样本中随机选取m个样本（根据先验知识指定），分别计算在这两类样本中选取的m个样本的均值作为样本代表，R为生成样本的个数，也是由先验知识给定，通过多次取样实现。生成样本的主要思想是生成的新样本与样本代表之间的距离越小越好，这样可以保证生成的样本分布在两类样本之间，只有利用这样的生成样本构成的约束才可以为度量的学习提供更多信息，由此构建模型公式为：

$$\min_{p_m^c} L=\sum_{c\in C}\sum_{r=1,2,\cdots,R}\alpha d_M^2(E(X_r^{y_i}),p_r^c)+(1-\alpha)d_M^2(E(X_r^{y_j}),p_r^c) \tag{1}$$

式中：p表示生成的新样本。模型的第一部分表示生成的新样本与随机选取的m个$\boldsymbol{y}_j$类样本代表间的间隔。第二部分表示生成的新样本与随机选取的m个$\boldsymbol{y}_i$类样本代表间的间隔。$\boldsymbol{X}_r^{y_i}$表示属于$\boldsymbol{y}_i$类的样本随机获取的第r个子集；α为平衡因子，用来平衡生成的样本更靠近哪类真实样本；$E(X)$计算样本均值也就是前面的提到的样本代表。

2.3 学习度量

把生成的样本的类别算作全新的类，然后将数据集中原有的训练集与生成样本的集合合并，作为度量学习的训练集$\boldsymbol{X}=[\boldsymbol{X}_{old};\boldsymbol{\Pi}]$，然后依据该训练集进行模型的训练，学习度量的思想参照了GMML[12]算法，主要思想是在保证生成的新样本间隔尽可能小的同时，标签不同的样本间隔越大越好，与其他度量学习方法不同的是，使用$\boldsymbol{M}^{-1}$来度量异类样本间的距离，综上所述，构造目标函数公式为：

$$\begin{aligned}&\min_{\boldsymbol{M}}\sum_{(\boldsymbol{\pi}_i,\boldsymbol{\pi}_j)\in\boldsymbol{S}}d_{\boldsymbol{M}}(\boldsymbol{\pi}_i,\boldsymbol{\pi}_j)+\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in\boldsymbol{D}}d_{\boldsymbol{M}^{-1}}(\boldsymbol{x}_i,\boldsymbol{x}_j)\\&s.t.\quad\boldsymbol{M}\succeq 0\end{aligned} \tag{2}$$

2.4 优化求解

SGML算法求解分两步进行，首先利用梯度下降算法来得到生成样本，步骤公式为：

$$\begin{aligned}&\frac{\partial L}{\partial\boldsymbol{\pi}_r^c}=\sum_{(\boldsymbol{y}_i,\boldsymbol{y}_j)\in\boldsymbol{C}}-2\alpha(E(\boldsymbol{X}_r^{y_i})-\boldsymbol{\pi}_r^c)-2(1-\alpha)(E(\boldsymbol{X}_r^{y_j})-\boldsymbol{\pi}_r^c)=0\\&\Rightarrow\alpha\boldsymbol{\pi}_r^c+(1-\alpha)\boldsymbol{\pi}_r^c=\alpha E(\boldsymbol{X}_r^{y_i})+(1-\alpha)E(\boldsymbol{X}_r^{y_j})\\&\Rightarrow\boldsymbol{\pi}_r^c=\alpha E(\boldsymbol{X}_r^{y_i})+(1-\alpha)E(\boldsymbol{X}_r^{y_j})\end{aligned} \tag{3}$$

可以看到，算法不需迭代就可以得到结果，因此生成难样本这一步对模型效率的影响可以忽略不计。

然后根据生成样本构造约束学习度量，利用梯度下降算法求解，步骤为：

$$\frac{\partial J(\boldsymbol{M})}{\partial\boldsymbol{M}}=\boldsymbol{S}-\boldsymbol{M}^{-1}\boldsymbol{D}\boldsymbol{M}^{-1}=0\Rightarrow\boldsymbol{M}\boldsymbol{S}\boldsymbol{M}=\boldsymbol{D} \tag{4}$$

式中：$S=\sum_{(\boldsymbol{\pi}_i,\boldsymbol{\pi}_j)\in\boldsymbol{\Pi}}(\boldsymbol{\pi}_i-\boldsymbol{\pi}_j)(\boldsymbol{\pi}_i-\boldsymbol{\pi}_j)^{\mathrm{T}}$，$\boldsymbol{D}=\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in D}(\boldsymbol{x}_i-\boldsymbol{x}_j)\,(\boldsymbol{x}_i-\boldsymbol{x}_j)^{\mathrm{T}}$。

可得度量矩阵的最优解为：

$$\boldsymbol{M}^*=\boldsymbol{S}^{-1/2}(\boldsymbol{S}^{1/2}\boldsymbol{D}\boldsymbol{S}^{1/2})^t\boldsymbol{S}^{-1/2},\qquad t\in[0,1] \tag{5}$$

式中：t为超参数，取值范围为0到1。具体求解过程如算法1所示。

算法1 基于样本生成的度量学习算法

Input：X：$n\times d$维样本集；Y：样本标签集；α,t：超参数；R生成样本的个数

Output：M^*

For $r=1,2,\cdots,R$

（1）每一类样本随机划分为R个簇

（2）求出每一个簇的样本均值

（3）根据式（3）求解得到生成的新样本

End for

（1）根据数据集生成相似集合S和不相似集合D

（2）根据式（5）求解度量M^*

3 实验设计及结果分析

实验部分主要通过对比SGML与kNN、ITML、AML、GMML、ANML[13]以及RVML[14]算法，分别在10个数据集上的准确率进行。选取的数据集样本数目跨度较大，小至150个样本，大至9298个样本，数据集的维度和类别也不尽相同，这样可以保证实验数据更具有说服力。实验中所用数据均来自UCI数据集（下载地址：https://archive.ics.uci.edu/），表1对实验所用的数据集进行了简单的描述。

表1 数据集描述

数据集	序号	样本数	维度	类别
Iris	1	150	4	3
Wine	2	178	13	3
Heart	3	270	13	2
Breast	4	699	10	2
Pima	5	768	8	2
Cars	6	392	8	3
Solar	7	323	12	6
German	8	1000	20	2
Waveform	9	5000	21	3
Usps	10	9298	256	10

3.1 实验数据与设计

由于 Usps 和 Wilt 数据集有给定的训练集和测试集，剩余 8 个数据集均采用随机取样的方式进行划分，选取数据集全部样本的 70% 为训练集，剩余样本为测试集。SGML 算法中的第一个超参数 α 取值范围为：$[10^{-3}, 10^{-2}, \cdots, 10^{2}]$，另外一个超参数 t 的取值为 $[0.1, 0.2, \cdots, 1]$，生成难样本的数量 m 取值为 30。其他对比算法的参数与其提出论文中给定的取值范围一致，在此不逐一列举。

由于在进行实验时所有算法均采用随机抽样的方法，来划分学习度量需要用到的训练集以及用于计算分类准确率的测试集，因此仅一次实验结果得到的分类准确率并不具有代表性，所以通过计算每个算法运行 20 次的准确率平均值来代表该算法在数据集上最终的准确率；然后分别对两个超参数进行灵敏度分析，观察算法性能受超参数变化而变化的情况；最后利用 t-SNE[15] 技术展示了数据集的原始分布和生成新样本之后的样本分布情况，验证 SGML 算法生成的新样本是否分布在真实样本的密度间隙。

3.2 实验结果分析

实验第一部分进行分类准确率的对比，得到了 SGML 算法与其余 6 个度量学习算法在 UCI 数据集上的准确率，从结果可以看出，SGML 算法除了在 Iris 数据集上与 ITML 算法并列第一，在其余 9 个 UCI 数据集上的分类准确率都是最高的。由于 ANML 算法在 Usps 数据集上运行时间过长，因此从没有统计该算法在 Usps 数据集上的准确率，详细的实验结果如表 2 所示。

表 2 SGML 算法与其余度量学习算法在 UCI 数据集上的分类准确率对比

数据集	kNN	LMNN	ITML	RVML	GMML	AML	ANML	SGML
1	0.964	0.962	0.984	0.946	0.962	0.950	0.955	0.984
2	0.965	0.987	0.991	0.929	0.987	0.953	0.966	0.997
3	0.779	0.798	0.822	0.780	0.820	0.785	0.835	0.857
4	0.956	0.963	0.963	0.953	0.964	0.952	0.967	0.978
5	0.724	0.732	0.745	0.685	0.748	0.729	0.749	0.772
6	0.816	0.863	0.863	0.743	0.862	0.827	0.837	0.883
7	0.696	0.707	0.716	0.709	0.721	0.710	0.718	0.743
8	0.631	0.585	0.666	0.585	0.651	0.6	0.631	0.679
9	0.804	0.816	0.811	0.829	0.816	0.807	0.806	0.851
10	0.945	0.945	0.947	0.880	0.946	0.944	—	0.951
Mean	0.807	0.826	0.835	0.783	0.838	0.809	—	0.861

实验第二部分选取 German、Heart、Breast 三个数据集分别进行参数灵敏度分析实验，图 1 展示了 SGML 算法分类准确率随着超参数 α 变化的情况，另外一个超参数 t 的取值固定为 0.5。图 2 展示了 SGML 算法分类准确率随着超参数 t 变化的情况，另一个超参数 α 的取值固定为 0.5。

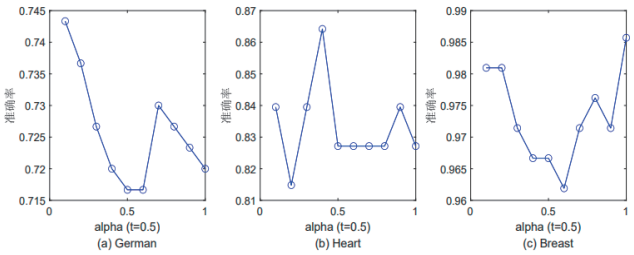


图 1 SGML 算法分类准确率随着超参数 α 变化的情况

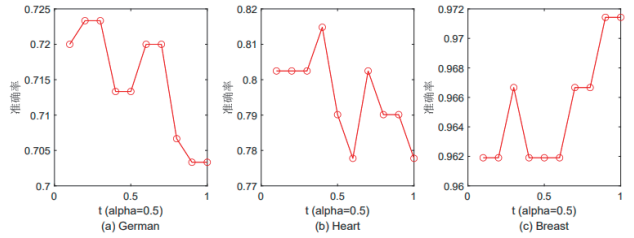
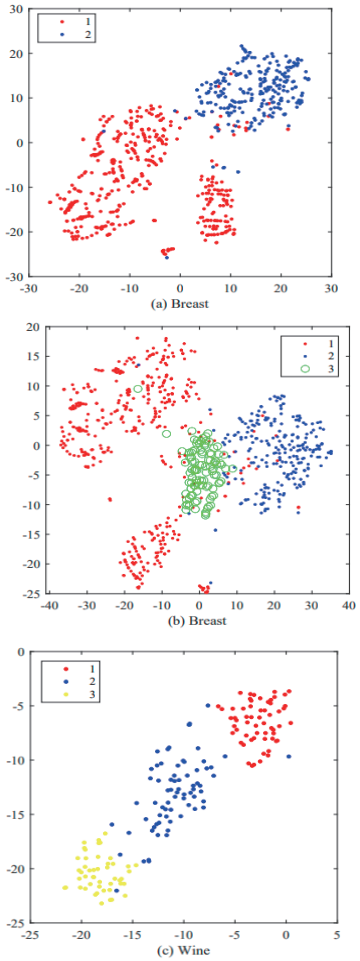


图 2 SGML 算法分类准确率随着超参数 t 变化的情况

实验第三部分分别选取 Breast 以及 Wine 两个数据集来展示生成难样本的分布情况。由于数据集的维度都比较高，因此利用 t-SNE 将算法降至二维来展示数据分布情况。图 3 展示了生成样本和原始样本的分布，图中实心圆表示数据集中原有的样本，用空心圆表示生成的难样本。



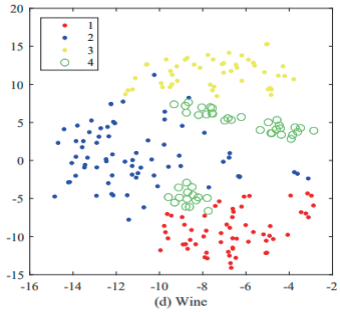


图 3 真实样本与生成样本分布示意图

从以上图表中可以看出，本文提出的利用生成的难样本以及原始样本共同作为模型的训练集来构造二元约束，能够为度量的学习提供更有意义的附属信息，使得学习的度量更具有判别力，在很大程度上提高了算法分类准确率及鲁棒性。

4 结论

为了解决传统的度量学习算法可以利用的难样本数量较少，从而导致模型判别力不足的问题，本文提出了一种新的生成难样本的办法，在数据集原有的异类样本密度间隙生成样本，通常这类样本更难区分，根据这类样本以及真实样本构造的约束，能够为度量的学习提供更多有意义的信息。同时，本文也进行了大量的实验，通过实验可以看出相较于以往提出的一些度量学习方法，SGML 算法的性能和鲁棒性都得到了一定的提升。但 SGML 算法需要通过先验知识指定生成难样本的数量，还需要进一步改进。

参考文献：

[1] WU C W, LIU X L, DING K Y, et al. Attack detection model for BCoT based on contrastive variational autoencoder and metric learning[J/OL].Journal of cloud computing,2024[2024-09-10].https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-024-00678-w.

[2]LIU C C, FERRARA M, FRANCO A, et al. Differential morphing attack detection via triplet-based metric learning and artifact extraction[C]//2024 International Conference of the Biometrics Special Interest Group (BIOSIG). Piscataway: IEEE, 2024:1-7.

[3] ALFARHOOD S, ALFARHOOD M .CAML: a context-Aware Metric Learning approach for improved recommender systems[J].Alexandria engineering journal, 2024,100(8): 53-60.

[4]ZHAO X X, HU Y P, MU Y S, et al. Graph Convolutional Metric Learning for Recommender Systems in Smart Cities[J]. IEEE transactions on consumer electronics, 2024,70(3): 5929-5941.

[5] 岳之一，钱素琴．基于多模态和度量学习的小样本图像分类方法 [J]. 东华大学学报 (自然科学版),2024,50(6):146-150.

[6]BEL K N S, SAM I S. Black hole entropic fuzzy clustering-based image indexing and tversky index-feature matching for image retrieval in cloud computing environment[J]. Information sciences, 2021, 560(27): 1-19.

[7] XING E P, NG A Y, JORDAN M I, et al. Distance metric learning with application to clustering with side-Information[C]//Proceedings of the 16th International Conference on Neural Information Processing Systems.MA: MIT Press, 2002: 521-528.

[8] WEINBERGER K Q, SAUL L K. Distance metric learning for large margin nearest neighbor classification[J]. The journal of machine learning research, 2009, 10: 207-244.

[9] DAVIS J V, KULIS B, JAIN P, et al. Information-theoretic metric learning[C]//Proceedings of the 24th International Conference on Machine Learning. NewYork: ACM, 2007: 209-216.

[10] YING Y M, LI P. Distance metric learning with eigenvalue optimization[J]. The journal of machine learning research, 2012, 13(1): 1-26.

[11] CHEN S, GONG C, YANG J, et al. Adversarial metric learning[DB/OL].(2018-02-09)[2024-05-11].https://doi.org/10.48550/arXiv.1802.03170.

[12] ZADEH P H, HOSSEINI R, SRA S. Geometric mean metric learning[DB/OL].(2016-07-18)[2024-03-19].https://doi.org/10.48550/arXiv.1607.05002.

[13] SONG K, HAN J W, CHENG G, et al. Adaptive neighborhood Metric learning[J]. IEEE transactions on pattern analysis and machine intelligence, 2021, 44(9): 4591-4604.

[14] PERROT M, HABRARD A. Regressive virtual metric learning[C]//Proceedings of the 29th International Conference on Neural Information Processing Systems.MA: MIT Press, 2015:1810-1818.

[15]CHAN D M, RAO R, HUANG F, et al. T-SNE-CUDA: gpu-accelerated t-SNE and its applications to modern data [C]//International Symposium on Computer Architecture and High Performance Computing. Piscataway: IEEE, 2018: 330-338.

【作者简介】

韩露（1997—），女，山西运城人，硕士，助教，研究方向：机器学习、度量学习。

（收稿日期：2024-10-15）