

## 基于锚图分类的在线半监督跨模态哈希

秦亮, 谢良, 陈盛双, 徐海蛟

### 引用本文

秦亮, 谢良, 陈盛双, 徐海蛟. 基于锚图分类的在线半监督跨模态哈希[J]. 计算机科学, 2023, 50(6): 183-193.

QIN Liang, XIE Liang, CHEN Shengshuang, XU Haijiao. [Online Semi-supervised Cross-modal Hashing Based on Anchor Graph Classification](#) [J]. Computer Science, 2023, 50(6): 183-193.

---

### 相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

#### Similar articles recommended (Please use Firefox or IE to view the article)

#### [基于多模态生成对抗网络的多元时序数据异常检测](#)

Multimodal Generative Adversarial Networks Based Multivariate Time Series Anomaly Detection  
计算机科学, 2023, 50(5): 355-362. <https://doi.org/10.11896/jsjcx.220400221>

#### [一种基于GRU的半监督网络流量异常检测方法](#)

Semi-supervised Network Traffic Anomaly Detection Method Based on GRU  
计算机科学, 2023, 50(3): 380-390. <https://doi.org/10.11896/jsjcx.220100032>

#### [基于层级化数据记忆池的边缘侧半监督持续学习方法](#)

Hierarchical Memory Pool Based Edge Semi-supervised Continual Learning Method  
计算机科学, 2023, 50(2): 23-31. <https://doi.org/10.11896/jsjcx.221100133>

#### [基于视频的在线学习情感识别研究](#)

Online Learning Emotion Recognition Based on Videos  
计算机科学, 2022, 49(11A): 211000049-6. <https://doi.org/10.11896/jsjcx.211000049>

#### [监督和半监督学习下的多标签分类综述](#)

Survey of Multi-label Classification Based on Supervised and Semi-supervised Learning  
计算机科学, 2022, 49(8): 12-25. <https://doi.org/10.11896/jsjcx.210700111>

# 基于锚图分类的在线半监督跨模态哈希

秦亮<sup>1</sup> 谢良<sup>1</sup> 陈盛双<sup>1</sup> 徐海蛟<sup>2</sup>

<sup>1</sup> 武汉理工大学理学院 武汉 430070

<sup>2</sup> 广东第二师范学院计算机学院 广州 510303

(839805126@qq.com)

**摘要** 近年来,哈希算法由于其存储成本小、检索速度快的特点,在大规模多媒体数据的高效跨模态检索中受到了广泛关注。现有的跨模态哈希算法大多是有监督和无监督方法,其中有监督方法通常能够获得更好的性能,但在实际应用中要求所有数据都被标记并不具有可行性。此外,这些方法大多数是离线方法,面对流数据的输入需要付出高额训练成本且十分低效。针对上述问题,提出了一种新的半监督跨模态哈希方法——在线半监督锚图跨模态哈希(Online Semi-supervised Anchor Graph Cross-modal Hashing, OSAGCH),构建了半监督锚图跨模态哈希模型,在只有部分数据有标签的情况下,利用正则化锚图预测数据标签,并通过子空间关系学习哈希函数,一步生成统一的哈希码,同时针对流数据输入的情况对该模型进行了在线化学习,使其能够处理流数据。在公共多模态数据集上进行了实验,结果表明所提方法的性能优于其他现有方法。

**关键词:** 跨模态哈希;半监督学习;锚图正则化;在线学习;子空间学习

**中图分类号** TP391.3

## Online Semi-supervised Cross-modal Hashing Based on Anchor Graph Classification

QIN Liang<sup>1</sup>, XIE Liang<sup>1</sup>, CHEN Shengshuang<sup>1</sup> and XU Haijiao<sup>2</sup>

<sup>1</sup> College of Science, Wuhan University of Technology, Wuhan 430070, China

<sup>2</sup> School of Computer Science, Guangdong University of Education, Guangzhou 510303, China

**Abstract** In recent years, hashing algorithm have been widely concerned in efficient cross-modal retrieval of large-scale multimedia data due to small storage costs and high retrieval speed. Most of the existing cross-modal hashing algorithms are supervised or unsupervised methods, and supervised methods usually achieve better performance. However, in real world applications, it is not feasible to require all data to be labeled. In addition, most of these methods are offline, which need to pay high training costs and are very inefficient when facing input of large stream data. This paper proposes a new semi-supervised cross-modal hashing method—online semi-supervised anchor graph cross-modal hashing(OSAGCH), which builds a semi-supervised anchor graph cross-modal hashing model. It uses regularized anchor graphs to predict data labels in the case where only part of the data has labels, and uses subspace relationship learning to learn hash functions, generating a unified hash code by one step. Then the model is expanded to online version for streaming data input, allowing it to process streaming data. Experiments on public multi-modal data sets indicate that the performance of proposed method is superior to other existing methods.

**Keywords** Cross-modal hashing, Semi-supervised learning, Anchor graph regularization, Online learning, Subspace learning

## 1 引言

当今信息技术迅速发展,大量文本、图像、视频、音频等多媒体数据在网络呈爆炸式增长。通常为了更生动形象地描述一个物品或事件,人们会使用多种模态的数据,如在目前热门的社交网站 Facebook, YouTube 上,用户可以提交任意的

多媒体数据进行查询。假设我们正在故宫游玩,可以通过拍的景点照片检索相关的文本、音频和视频等介绍材料作为游玩向导。

多模态数据的快速涌入迫切需求高效的跨模态检索解决方案。哈希算法由于其存储成本小、检索速度快的特点,在大规模多媒体数据的高效跨模态检索中受到了广泛关注<sup>[1]</sup>,它

到稿日期:2022-04-03 返修日期:2022-09-07

基金项目:广东省自然科学基金(2020A151501212);广州市基础研究的计划基础与应用基础研究项目(202102080353);广东省普通高校自然科学类特色创新项目(2019KTSCX117)

This work was supported by the Natural Science Foundation of Guangdong Province, China(2020A151501212), Basic and Applied Basic Research Project of Guangzhou Basic Research Teaching Program(202102080353) and Characteristic Innovation Project of Natural Science in General Colleges and Universities in Guangdong Province(2019KTSCX117).

通信作者:谢良(whutxl@hotmail.com)

通过将高维度的特征投影到短长度哈希码,有效地提高了存储和检索效率。

现有的跨模态哈希方法可以分为两类,即无监督哈希方法和监督哈希方法。在缺少语义监督信息的情况下,无监督哈希方法仅仅通过训练没有标记的样本数据得到学习模型,准确性较差;而监督哈希方法能够利用探索到的相关语义信息,如标签、标记等,这些信息可以被进一步使用到学习的二进制码中,因此相比无监督方法可以获得更好的检索结果。随着深度学习的发展,基于深度学习的哈希方法结合深度学习强大的特征提取能力表现出了优越的性能,吸引了众多国内外学者。

然而,在无人机、移动环境等实时性较高的应用场景中,由于数据标记的高额成本,通常只能获得部分标记数据和大量未标记数据,该情况下传统的无监督和监督跨模态哈希方法的性能往往无法令人满意。此外,在这些实时场景中,数据实例以流数据的形式输入,深度哈希需要付出高额的训练成本,不适用于对哈希函数和哈希码进行实时更新。

为了应对上述挑战,本文提出了一种新的在线半监督跨模态哈希方法——在线半监督锚图跨模态哈希(OSAGCH)。本文的主要贡献如下:

(1)当只有部分标记数据时,引入正则化锚图,基于图学习对锚点进行正则化;同时利用阈值设计针对多标签数据的标签预测公式,得到所有数据的预测标签。

(2)针对流数据的输入对基础模型进行在线化学习,通过在线离散优化算法求解目标函数,实时优化更新投影矩阵及哈希码,有效加快训练速度,降低对存储空间的需求。

(3)在哈希函数中使用核映射和子空间投影,保留了原始特征的非线性结构,并结合预测的类标签和子空间关系信息,共同学习统一的哈希码和更具区分性的哈希函数。

本文第2节介绍了相关工作;第3节对OSAGCH进行了详细介绍;第4节给出了在3个数据集上进行实验的结果;最后总结全文并展望未来。

## 2 相关工作

根据语义信息的使用情况,哈希方法分为无监督、监督和半监督哈希三大类。无监督哈希方法不使用来自给定数据库的语义标签信息<sup>[2-3]</sup>。迭代量化哈希(Iterative Quantization, ITQ)<sup>[4]</sup>解决了在大规模图像数据集中保留相似性二进制哈希码的学习问题,通过将实值数据向量和二进制哈希码之间的量化误差最小化,为图像找到保留相似性的二进制哈希码,并通过迭代更新旋转矩阵和哈希码来最小化ITQ的目标函数。无监督双线性局部哈希(Unsupervised Bilinear Local Hashing, UBLH)<sup>[5]</sup>通过紧凑的双线性投影矩阵将局部特征从高维特征空间投影到低维汉明空间,输入局部特征矩阵表达式,保留局部特征的特征-特征和图像-图像结构,在大规模图像相似性检索中学习局部特征上分辨率高的二进制哈希码。Heo等<sup>[6]</sup>提出了一种新的基于超球的哈希函数及二进制编码距离函数(即球面汉明距离),它通过找到超球而不是超平面来将数据集划分成不同的哈希桶,并迭代优化生成高效的哈希码。监督哈希方法要求数据库中的所有数据都被

标记,并利用这些语义标签学习哈希函数。Strecha等<sup>[7]</sup>提出了一种新颖的由SIFT描述符生成哈希码的方法,通过线性判别分析数据特征构建哈希函数。Liu等<sup>[8]</sup>提出了基于核的监督哈希模型(Kernel-based Supervised Hashing, KSH),将相似数据对的汉明距离最小化以及不同数据对的汉明距离最大化,以此训练哈希函数,生成紧凑的哈希码。Nguyen等<sup>[9]</sup>提出了一种新的基于深度学习的监督哈希方法,该方法利用深度神经网络训练哈希函数学习非线性嵌入,保留了数据的复杂语义结构并为语义相似的数据生成附近的哈希码。数据库中的语义标签提供额外的鉴别信息,具有相同类标签的数据应该具有相同或极为相似的哈希码,从而监督哈希方法,通常可以获得比无监督哈希方法更好的检索结果。

然而,现实中数据库通常是部分标记的,要求所有数据都被标记并不现实。当数据库不能提供足够的语义标签时,监督哈希方法往往会过拟合。因此,半监督哈希方法被提出,用于充分利用部分有标记的数据和大量未标记的数据<sup>[10-13]</sup>。Zhang等<sup>[14]</sup>提出了半监督师生网络哈希框架(Pairwise Teacher-student Semi-supervised Hashing, PTS<sup>3</sup>H),利用教师网络产生的成对信息指导学生网络训练。Anculef等<sup>[15]</sup>提出了自监督伯努利变分自编码器(Self-supervised Bernoulli Variational Autoencoders, SSB-VAE),通过预测标签分布使数据成对,在标签稀缺的情况下明显改善结果。受生成模型和极大极小二人博弈思想的启发,Wang等<sup>[16]</sup>提出了半监督生成对抗哈希算法(Semi-supervised Generative Adversarial Hashing, SSGAH),将生成模型、判别模型以及深度哈希模型相结合,充分利用所有有/无标签数据,并通过半监督排序损失和对手排序损失学习哈希码,捕获有/无标签数据的语义信息。Yu等<sup>[17]</sup>提出了新的半监督哈希框架学习半配对数据下的哈希函数,通过联合优化特征表示和分类器参数提高哈希码质量,利用训练数据集提供的已有标签信息满足实际应用。在只有非常有限的标记数据的情况下,Song等<sup>[18]</sup>提出了半监督流形嵌入哈希(Semi-supervised Manifold-embedded Hashing, SMH),通过一种半监督流形嵌入联合优化特征表示和分类器学习,确保学习到的哈希码最适合分类。大多数半监督学习方法无法很好地适应数据大小的变化,限制了半监督学习在大规模数据集中的应用,本文参考锚图正则化方法ARG对该问题进行处理,并对锚图进行在线更新,使OSAGCH能够应用于大规模数据集。

由于多模态数据的快速涌入,一系列跨模态哈希方法被相继提出。受潜在子空间学习的影响,Xu等<sup>[19]</sup>提出了离散跨模态哈希(Discrete Cross-modal Hashing, DCH)。DCH通过学习模态特定的哈希函数生成统一的哈希码,保留离散约束,保证哈希码的可分辨性,并通过离散优化有效减小量化误差。Eto等<sup>[20]</sup>进一步提出了哈达玛编码离散跨模态哈希(Hadamard-coded Discrete Cross-modal Hashing, HC-DCH),将哈达玛编码应用于DCH,在保证精度的情况下大幅度缩短训练时间。Yu等<sup>[21]</sup>提出了灵活的跨模态哈希方法(Flexible Cross-modal Hashing, FlexCMH),用于处理弱配对数据,该方法引入基于聚类的匹配策略探索不同模态下聚类之间的潜在对应关系,有效提高了跨模态哈希任务的灵活性。

Lin等<sup>[22]</sup>提出了基于概率的语义保留哈希方法(Semantics-preserving Hashing, SePH),该方法考虑视图间的语义一致性,生成统一哈希码,并在训练时将语义亲和度转换为概率分布,在汉明空间中保持该分布以建立不同模态的语义关联。基于深度学习强大的特征提取能力,Jiang等<sup>[23]</sup>提出了深度跨模态哈希算法(Deep Cross-modal Hashing, DCMH),将深度特征学习与哈希变换方法相结合,同时对特征和哈希码进行学习。在无人机、移动环境等实时性较高的应用场景中,数据实例通常以流数据的形式输入,上述方法均属于离线方法,处理流数据需要付出高额的训练成本。因此,在线哈希方法被提出,根据流数据的输入实时更新哈希函数。由于深度哈希通常训练时间较长,不适用于处理在线检索问题,因此目前没有基于深度学习的在线哈希方法。

Huang等<sup>[24]</sup>提出了在线哈希模型(Online Hashing, OH),通过一个新的损失函数度量汉明空间中数据对样本的相似损失,以被动攻击的方式优化导出后的结构化哈希模型。Lin等<sup>[25]</sup>提出了基于哈达玛码本的在线哈希方案(Hadamard Codebook based Online Hashing, HCOH),为每个类标号分配一个由哈达玛码本随机生成的高维哈希码,并采用局部敏感哈希根据哈希位减少哈达玛码的长度以在线适应预定义的哈希码,保证语义相似性,实现鲁棒的监督在线哈希。然而,

HCOH简单考虑线性回归拟合虚拟类别,导致松弛过程中承受了更多的量化误差,Lin等<sup>[26]</sup>进一步提出哈达玛矩阵引导哈希(HMOH),通过引入经典的核化感知器算法来很好地解决该问题。目前在线哈希方法更多地侧重于无监督及监督的单模态哈希,在半监督哈希及跨模态哈希方面的研究较少。

### 3 在线半监督锚图跨模态哈希

本节对OSAGCH的具体信息进行了详细介绍。3.1节介绍了本文算法的整体框架,3.2节对在线哈希问题进行了描述,3.3节给出了半监督锚图跨模态哈希的目标模型,3.4节给出了该模型的在线离散优化算法,以实现线性时间复杂度的在线哈希方法。

#### 3.1 OSAGCH 算法框架

图1给出了本文算法的整体框架。每当数据库中有新的未标记数据到来,通过在线离散学习对锚图进行在线更新,同时利用正则化锚图对未标记数据进行标签预测,通过子空间学习对类标向量及数据特征向量进行子空间投影。然后,通过在线离散学习对投影矩阵、关系投影矩阵和统一哈希码进行在线交替迭代优化,得到新的哈希函数。最后,输入查询图像或文本进行跨模态检索。

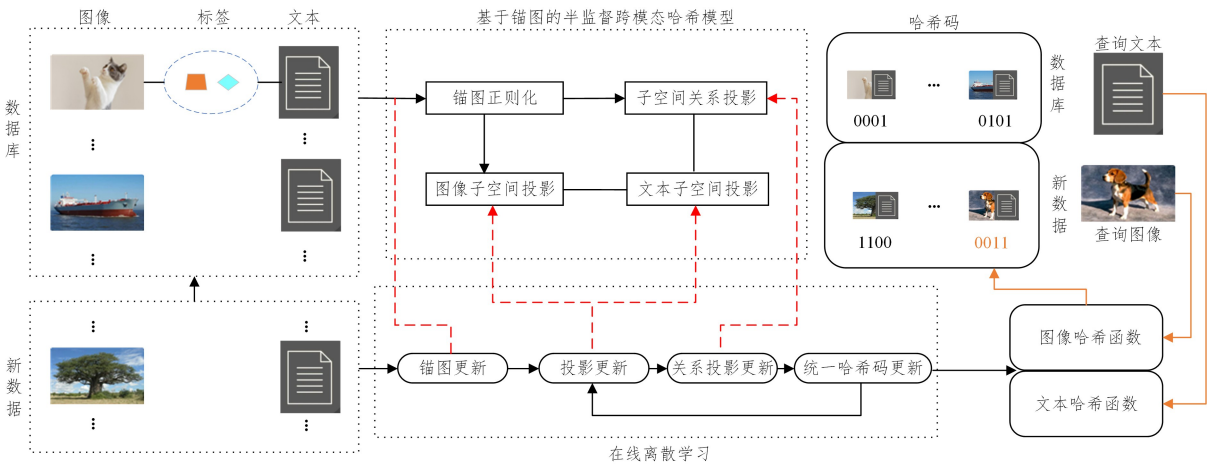


图1 OSAGCH 框架图

Fig.1 OSAGCH framework

#### 3.2 问题描述

假设数据集由一组  $n$  个图像文本对组成,  $\mathbf{X} = \{x_i\}_{i=1}^n$ , 其中标记数据为  $x_i$  ( $i = 1, 2, \dots, l$ ), 对应的标签  $y_i = [y_{i1}, y_{i2}, \dots, y_{ik}] \in \mathbb{R}^c$ , 如果该实例属于第  $j$  个类, 则  $y_{ij} = 1$ , 否则为  $y_{ij} = 0$ .  $\mathbf{Y}_l = [y_1^T, y_2^T, \dots, y_l^T]^T \in \mathbb{R}^{l \times c}$  表示标记数据点上的标签矩阵。对于每个实例  $x_i = (v_i, t_i)$  ( $i = 1, 2, \dots, n$ ),  $v_i \in \mathbb{R}^{f \times n}$  是图像特征向量,  $t_i \in \mathbb{R}^{d \times n}$  是文本特征向量。哈希学习的目的是通过  $L$  个哈希函数  $H(\mathbf{X}) = \{h_i(\mathbf{X})\}_{i=1}^L$  对数据集  $\mathbf{X}$  进行投影来学习一组哈希码  $B = \{b_i\}_{i=1}^n \in \{-1, +1\}^{L \times n}$ , 确保保留需要的邻域结构, 即:

$$\mathbf{B} = H(\mathbf{X}) = \text{sgn}(\mathbf{W}^T \mathbf{X}) \quad (1)$$

其中,  $\mathbf{W} = \{w_i\}_{i=1}^L$  是映射数据原始特征到低维潜在空间的投影矩阵。若符号函数  $\text{sgn}(\cdot)$  为正数则输出  $+1$ , 否则输出  $-1$ 。

在线设置中,  $\mathbf{X}$  以流数据形式出现, 不能一次性使用。因此, 将数据分为  $m$  个阶段输入,  $\mathbf{V}^k = \{v_i^k\}_{i=1}^{n_k} \in \mathbb{R}^{f \times n_k}$  和  $\mathbf{T}^k = \{t_i^k\}_{i=1}^{n_k} \in \mathbb{R}^{d \times n_k}$  分别表示图像和文本第  $k$  阶段的流数据输入, 相应地, 在第  $k$  阶段更新的  $\mathbf{W}_v, \mathbf{W}_t, \mathbf{B}$  表示为  $\mathbf{W}_v^k, \mathbf{W}_t^k, \mathbf{B}^k$ , 其中  $n_k$  为第  $k$  阶段流数据的数量。

#### 3.3 基于锚点图的半监督跨模态哈希模型

锚图正则化模型通过数据点和锚点探索数据集的底层结构, 构建正则化框架, 它基于一个聚类假设: 附近的点可能具有相同的类标签。本节首先通过锚图正则化对无标签数据进行标签预测。

由于图像和文本是经典的非结构化数据, 无法较好地保留原始特征向量的潜在结构, 考虑使用核技巧对数据进行处理, 通过核函数表达原始特征向量之间的非线性模态内的相关性, 因此本文将核函数表示为  $Q(\cdot)$ , 其中图像向量的核化

特征表示为  $\mathbf{Q}_v(V)$ , 文本向量的核化特征表示为  $\mathbf{Q}_t(T)$ 。对于特定的特征向量  $\mathbf{x}_i$ , 核化特征的表达式如下:

$$\mathbf{Q}_t(\mathbf{x}_i) = [k(\mathbf{x}_i, \mathbf{x}_{(1)}), \dots, k(\mathbf{x}_i, \mathbf{x}_{(e)})]^T \quad (2)$$

其中,  $\mathbf{x}_{(1)}, \dots, \mathbf{x}_{(e)}$  是随机选取的  $e$  个锚点样本,  $k(x, y) = \exp(-\|x - y\|^2 / 2\eta^2)$  表示高斯核函数,  $\eta$  是学习过程中需要调整的带宽。

另一方面, 高斯核函数  $k(x, y)$  反映了两个数据点之间的距离关系, 数据点间的距离越小,  $k(x, y)$  的值越大。本节将  $\mathbf{Q}_t(\mathbf{x}_i)$  作为数据的局部权重矩阵, 用于反映数据点  $\mathbf{x}_i$  与各锚点间的距离权重。通过对  $\mathbf{Q}_v, \mathbf{Q}_t$  加权相加, 总局部权重矩阵  $\mathbf{Q}_x$  的表达式如下:

$$\mathbf{Q}_x = \epsilon_v \mathbf{Q}_v + \epsilon_t \mathbf{Q}_t \quad (3)$$

其中,  $\epsilon_v, \epsilon_t$  是权重参数。

在锚图正则化中, 所有数据点上的正则化实际上等价于锚点上的正则化。基于总局部权重矩阵  $\mathbf{Q}_x$ , 锚点间的邻接矩阵的设计如下:

$$\mathbf{Z} = \mathbf{Q}_x \mathbf{Q}_x^T \quad (4)$$

其中,  $\mathbf{Z}_{sp} = \sum_{k=1}^n \mathbf{Q}_{xks} \mathbf{Q}_{xkp}$ ,  $i, j, g$  和  $s, p, r$  分别是数据点和锚点的索引。它的正则化拉普拉斯矩阵的表达式如下:

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{Z} \mathbf{D}^{-1/2} \quad (5)$$

其中, 对角矩阵  $\mathbf{D}$  被定义为  $\mathbf{D}_{ss} = \sum_{\rho=1}^e \mathbf{Z}_{s\rho}$ 。

为了预测与锚点相关的未标记数据的标签, 假设对于每个数据点  $\mathbf{x}_i$ , 有一个标签预测函数  $f_i = \mathbf{Q}_x(\mathbf{x}_i)^T \mathbf{A}$ , 其中  $\mathbf{A} = [a_1^T, \dots, a_m^T]^T \in \mathbb{R}^{e \times c}$  表示锚点集上的标签预测矩阵,  $a_i \in \mathbb{R}^{1 \times c}$  是锚点  $u_i$  的标签向量。本文将锚点和数据点相结合, 构造锚图正则化模型:

$$\arg \min_{\mathbf{A}} \sum_{i=1}^l \|\mathbf{Q}_x(\mathbf{x}_i)^T \mathbf{A} - y_i\|^2 + \frac{\gamma}{2} \sum_{i,j=1}^m \mathbf{Z}_{ij} \left\| \frac{a_i}{\sqrt{\mathbf{D}_{ii}}} - \frac{a_j}{\sqrt{\mathbf{D}_{jj}}} \right\|^2 \quad (6)$$

这个优化问题可以用矩阵形式来表示:

$$\arg \min_{\mathbf{A}} \|\mathbf{Q}_x^T \mathbf{A} - \mathbf{Y}_l\|_F^2 + \gamma \text{tr}(\mathbf{A}^T \mathbf{L} \mathbf{A}) \quad (7)$$

其中,  $\gamma > 0$  是权衡参数。利用式(7)得到锚点标签矩阵  $\mathbf{A}$  的全局最优解, 并基于标签预测函数  $f_i$ , 设置阈值  $\rho$ , 对未标记数据的标签进行预测, 得到所有数据的标签矩阵  $\mathbf{F} = [y_1^T, \dots, y_l^T, \hat{y}_{l+1}^T, \dots, \hat{y}_n^T] \in \mathbb{R}^{c \times n}$ 。

接下来本节通过子空间关系学习哈希函数, 目标哈希函数表示为:

$$H_v(\mathbf{V}) = \text{sgn}(\mathbf{W}_v^T \mathbf{Q}_v(\mathbf{V})) \quad (8)$$

$$H_t(\mathbf{T}) = \text{sgn}(\mathbf{W}_t^T \mathbf{Q}_t(\mathbf{T})) \quad (9)$$

其中,  $\mathbf{W}_v \in \mathbb{R}^{e \times L}$ ,  $\mathbf{W}_t \in \mathbb{R}^{e \times L}$  分别是图像和文本的投影矩阵。

为了减小异构差异并匹配同一类别的数据, 在子空间中构造一个新的投影矩阵  $\mathbf{P} \in \mathbb{R}^{c \times L}$ , 将类标向量映射到低维子空间, 将其视为子空间的关系信息。然后, 通过联合最小化二进制码和子空间关系信息的距离, 学习所需的投影矩阵及哈希函数。目标函数如下:

$$\min_{H_v, H_t} \mu_v \|\mathbf{H}_v(\mathbf{V}) - \mathbf{P}^T \mathbf{F}\|_F^2 + \mu_t \|\mathbf{H}_t(\mathbf{T}) - \mathbf{P}^T \mathbf{F}\|_F^2 \quad (10)$$

其中,  $u_v$  和  $u_t$  是平衡参数,  $\|\cdot\|_F^2$  表示 Frobenius 范数。

式(10)中的目标函数是非线性的, 无法直接进行求解。

对于特定模式的特征矩阵  $\mathbf{X}$ , 通过简单去掉式(1)中的  $\text{sgn}(\cdot)$  得到线性函数  $\hat{H}(\mathbf{X}) = \mathbf{W}^T \mathbf{X}$ , 同时引入统一哈希码  $B = B_v = B_t$ , 避免最小化  $\|B_v - B_t\|_F$  时不必要的权衡项。式(10)可以改写为:

$$\begin{aligned} \min_{\mathbf{W}_v, \mathbf{W}_t, B, \mathbf{P}} \quad & \|\mathbf{B} - \mathbf{P}^T \mathbf{F}\|_F^2 + \nu_v \|\mathbf{B} - \mathbf{W}_v^T \mathbf{Q}_v(\mathbf{V})\|_F^2 + \\ & \nu_t \|\mathbf{B} - \mathbf{W}_t^T \mathbf{Q}_t(\mathbf{T})\|_F^2 + \Omega(\mathbf{P}, \mathbf{W}_v, \mathbf{W}_t) \\ \text{s. t.} \quad & \mathbf{B} \in \{-1, 1\}^{L \times n} \end{aligned} \quad (11)$$

其中,  $\Omega(\mathbf{P}, \mathbf{W}_v, \mathbf{W}_t) = \delta \|\mathbf{P}\|_F^2 + \alpha \|\mathbf{W}_v\|_F^2 + \beta \|\mathbf{W}_t\|_F^2$  表示  $L_2$  正则化惩罚项, 用于提高模型的稳定性,  $\delta, \alpha, \beta$  是惩罚参数。  $\nu_v$  和  $\nu_t$  是两种模式的新平衡参数,  $\nu_v = \frac{u_v}{u_v + u_t}$ ,  $\nu_t = \frac{u_t}{u_v + u_t}$ 。

### 3.4 在线离散学习算法

随着互联网上可用数据量的增加, 大规模的流数据出现, 传统的离线哈希方法无法适应数据变化, 检索效率降低, 因此在线哈希被提出。本节针对流数据输入的情况, 对 3.3 节提出的模型进行在线化学习。为了实时更新哈希函数及哈希码, 目标函数设计如下:

$$\begin{aligned} \min_{\mathbf{W}_v^k, \mathbf{W}_t^k, B^k, \mathbf{P}^k} \quad & \|\mathbf{B} - (\mathbf{P}^k)^T \mathbf{F}\|_F^2 + \nu_v \|\mathbf{B} - (\mathbf{W}_v^k)^T \mathbf{Q}_v(\mathbf{V})\|_F^2 + \\ & \nu_t \|\mathbf{B} - (\mathbf{W}_t^k)^T \mathbf{Q}_t(\mathbf{T})\|_F^2 + \Omega(\mathbf{P}^k, \mathbf{W}_v^k, \mathbf{W}_t^k) \\ \text{s. t.} \quad & \mathbf{B} \in \{-1, 1\}^{L \times n}, n = \sum_{i=1}^k n_i \end{aligned} \quad (12)$$

其中,  $\mathbf{B} = [\mathbf{B}^1, \dots, \mathbf{B}^k]$ ,  $\mathbf{F} = [\mathbf{F}^1, \dots, \mathbf{F}^k]$ ,  $\mathbf{Q}_v(\mathbf{V}) = [\mathbf{Q}_v(\mathbf{V})^1, \dots, \mathbf{Q}_v(\mathbf{V})^k]$ ,  $\mathbf{Q}_t(\mathbf{T})$  与  $\mathbf{Q}_v(\mathbf{V})$  类似。

各变量的更新过程如下。

首先, 对  $\mathbf{Q}_v^k, \mathbf{Q}_t^k$  加权相加计算得到第  $k$  阶段的总局部权重矩阵  $\mathbf{Q}_x^k$ :

$$\mathbf{Q}_x^k = \epsilon_v \mathbf{Q}_v^k + \epsilon_t \mathbf{Q}_t^k \quad (13)$$

当  $k=1$  时, 设计邻接矩阵  $\mathbf{Z}^1$ :

$$\mathbf{Z}^1 = \mathbf{Q}_x^1 (\mathbf{Q}_x^1)^T \quad (14)$$

当  $k \geq 2$  时, 设  $\mathbf{C}^{k-1} = \sum_{i=1}^{k-1} \mathbf{Q}_x^i (\mathbf{Q}_x^i)^T \in \mathbb{R}^{e \times e}$ ,  $\mathbf{C}^{k-1}$  表示中间变量, 用于存储前  $k-1$  阶段邻接矩阵的计算结果。在第  $k$  阶段对邻接矩阵  $\mathbf{Z}^k$  进行在线更新:

$$\mathbf{Z}^k = \mathbf{C}^k = \mathbf{C}^{k-1} + \mathbf{Q}_x^k (\mathbf{Q}_x^k)^T \quad (15)$$

基于  $\mathbf{Z}^k$ , 正则化拉普拉斯公式及锚点标签矩阵  $\mathbf{A}^k$  可进行如下的在线更新:

$$\mathbf{L}^k = \mathbf{I} - (\mathbf{D}^k)^{-1/2} \mathbf{Z}^k (\mathbf{D}^k)^{-1/2} \quad (16)$$

$$\mathbf{A}^k = (\mathbf{Q}_{x_i}^k (\mathbf{Q}_{x_i}^k)^T + \gamma \mathbf{L}^k)^{-1} \mathbf{Q}_{x_i}^k \mathbf{Y}_i \quad (17)$$

其中, 对角矩阵  $\mathbf{D}^k$  被定义为  $\mathbf{D}_{ss}^k = \sum_{\rho=1}^e \mathbf{Z}_{s\rho}^k$ 。

最后, 利用更新后的总局部权重矩阵  $\mathbf{Q}_x^k$  及锚点标签矩阵  $\mathbf{L}^k$  对未标记数据的标签进行预测, 计算得到第  $k$  阶段的预测标签矩阵  $\mathbf{F}^k = [(\hat{y}_1^k)^T, \dots, (\hat{y}_{n_k}^k)^T] \in \mathbb{R}^{c \times n_k}$ :

$$\hat{y}_{ij}^k = \begin{cases} 1, & \text{if } \frac{(\mathbf{Q}_{x_i}^k \cdot)^T \times \mathbf{A}_{\cdot j}^k}{\lambda_j^k} \geq \rho, i=1, \dots, n_k \\ 0, & \text{if } \frac{(\mathbf{Q}_{x_i}^k \cdot)^T \times \mathbf{A}_{\cdot j}^k}{\lambda_j^k} < \rho, j=1, \dots, c \end{cases} \quad (18)$$

其中,  $(\mathbf{Q}_{x_i}^k \cdot)^T$  表示  $(\mathbf{Q}_x^k)^T$  的第  $i$  行,  $\mathbf{A}_{\cdot j}^k$  表示  $\mathbf{A}^k$  的第  $j$  列,  $\lambda_j^k = \mathbf{1}^T (\mathbf{Q}_x^k)^T \mathbf{A}_{\cdot j}^k$  是正则化因子, 用于平衡偏类态分布。当

$k=1$  时,  $y_i^1 = y_i^1 (i=1, \dots, L)$ 。

式(15)的最优化问题对于 4 个矩阵变量是非凸的,难以得到局部最优解,但固定其他 3 个变量之后,它对于剩下的变量是凸的。因此,本节通过在线交替迭代优化算法对式(12)进行求解,直到收敛。下面将详细介绍在线优化算法的求解步骤。

Step1 固定其他变量,学习投影矩阵  $\mathbf{W}_v^k, \mathbf{W}_t^k$ , 由于两个投影矩阵是独立的,式(12)的最优化问题可以改写为:

$$\min_{\mathbf{W}_v^k} v_v \|\mathbf{B} - (\mathbf{W}_v^k)^T \mathbf{Q}_v(\mathbf{V})\|_F^2 + \alpha \|\mathbf{W}_v^k\|_F^2 \quad (19)$$

$$\min_{\mathbf{W}_t^k} v_t \|\mathbf{B} - (\mathbf{W}_t^k)^T \mathbf{Q}_t(\mathbf{T})\|_F^2 + \beta \|\mathbf{W}_t^k\|_F^2$$

分别对该式求对应投影矩阵的偏导,使偏导数为 0, 计算得  $\mathbf{W}_v^k$  和  $\mathbf{W}_t^k$ :

$$\mathbf{W}_v^k = (\mathbf{Q}_v(\mathbf{V})\mathbf{Q}_v(\mathbf{V})^T + \eta_v \mathbf{I})^{-1} \mathbf{Q}_v(\mathbf{V})\mathbf{B}^T \quad (20)$$

$$\mathbf{W}_t^k = (\mathbf{Q}_t(\mathbf{T})\mathbf{Q}_t(\mathbf{T})^T + \eta_t \mathbf{I})^{-1} \mathbf{Q}_t(\mathbf{T})\mathbf{B}^T$$

其中,  $\eta_v = \frac{\alpha}{v_v}$ ,  $\eta_t = \frac{\beta}{v_t}$ , 对于特定模态, 设  $\mathbf{D}^{k-1} = \eta \mathbf{I} +$

$\sum_{i=1}^{k-1} \mathbf{Q}^i(\mathbf{Q}^i)^T \in \mathbb{R}^{e \times e}$ ,  $\mathbf{E}^{k-1} = \sum_{i=1}^{k-1} \mathbf{Q}^i(\mathbf{E}^i)^T \in \mathbb{R}^{e \times e}$ ,  $\mathbf{D}^{k-1}$  和  $\mathbf{E}^{k-1}$  表示中间变量, 分别用于存储前  $k-1$  阶段  $\mathbf{Q}^i(\mathbf{Q}^i)^T$  和  $\mathbf{Q}^i(\mathbf{E}^i)^T$  的计算结果。在第  $k$  阶段, 两个中间变量进行在线更新, 即:

$$\mathbf{D}^k = \mathbf{D}^{k-1} + \mathbf{Q}^k(\mathbf{Q}^k)^T \quad (21)$$

$$\mathbf{E}^k = \mathbf{E}^{k-1} + \mathbf{Q}^k(\mathbf{E}^k)^T$$

结合式(20)和式(21), 两个投影矩阵的更新表示如下:

$$\mathbf{W}_v^k = (\mathbf{D}_v^k)^{-1} \mathbf{E}_v^k \quad (22)$$

$$\mathbf{W}_t^k = (\mathbf{D}_t^k)^{-1} \mathbf{E}_t^k$$

Step2 固定其他变量, 学习关系投影矩阵  $\mathbf{P}^k$ , 式(12)的最优化问题可以改写为:

$$\min_{\mathbf{P}^k} \|\mathbf{B} - (\mathbf{P}^k)^T \mathbf{F}\|_F^2 + \delta \|\mathbf{P}^k\|_F^2 \quad (23)$$

根据 F 范数的特性, 该式进一步可表述为:

$$\min_{\mathbf{P}^k} \text{Tr}((\mathbf{B} - (\mathbf{P}^k)^T \mathbf{F})^T (\mathbf{B} - (\mathbf{P}^k)^T \mathbf{F})) + \delta \text{Tr}((\mathbf{P}^k)^T \mathbf{P}^k) \quad (24)$$

其中,  $\text{Tr}(\cdot)$  表示迹范数, 可以求得如下解:

$$\mathbf{P}^k = (\delta \mathbf{I} + \mathbf{F}\mathbf{F}^T)^{-1} \mathbf{F}\mathbf{B}^T \quad (25)$$

设  $\mathbf{G}^{k-1} = \delta \mathbf{I} + \sum_{i=1}^{k-1} \mathbf{F}^i (\mathbf{F}^i)^T$ ,  $\mathbf{H}^{k-1} = \sum_{i=1}^{k-1} \mathbf{F}^i (\mathbf{B}^i)^T$ ,  $\mathbf{G}^{k-1}$  和  $\mathbf{H}^{k-1}$  表示中间变量, 分别用于存储前  $k-1$  阶段  $\mathbf{F}^i (\mathbf{F}^i)^T$  和  $\mathbf{F}^i (\mathbf{B}^i)^T$  的计算结果。在第  $k$  阶段, 两个中间变量进行在线更新:

$$\mathbf{G}^k = \mathbf{G}^{k-1} + \mathbf{F}^k (\mathbf{F}^k)^T \quad (26)$$

$$\mathbf{H}^k = \mathbf{H}^{k-1} + \mathbf{F}^k (\mathbf{B}^k)^T$$

结合式(27)和式(29), 关系投影矩阵  $\mathbf{P}^k$  的更新表示如下:

$$\mathbf{P}^k = (\mathbf{G}^k)^{-1} \mathbf{H}^k \quad (27)$$

Step3 固定其他变量, 学习  $\mathbf{B}^k$ , 式(12)的最优化问题可以改写为:

$$\min_{\mathbf{B}^k} \|\mathbf{B}^k - (\mathbf{P}^k)^T \mathbf{F}^k\|_F^2 + v_v \|\mathbf{B}^k - (\mathbf{W}_v^k)^T \mathbf{Q}_v(\mathbf{V})^k\|_F^2 + v_t \|\mathbf{B}^k - (\mathbf{W}_t^k)^T \mathbf{Q}_t(\mathbf{T})^k\|_F^2 \quad (28)$$

式(28)进一步表述为:

$$\min_{\mathbf{B}^k} \text{Tr}((\mathbf{B}^k - (\mathbf{P}^k)^T \mathbf{F}^k)^T (\mathbf{B}^k - (\mathbf{P}^k)^T \mathbf{F}^k)) + v_v \text{Tr}((\mathbf{B}^k -$$

$$(\mathbf{W}_v^k)^T \mathbf{Q}_v(\mathbf{V})^k)^T (\mathbf{B}^k - (\mathbf{W}_v^k)^T \mathbf{Q}_v(\mathbf{V})^k)) + v_t \text{Tr}((\mathbf{B}^k - (\mathbf{W}_t^k)^T \mathbf{Q}_t(\mathbf{T})^k)^T (\mathbf{B}^k - (\mathbf{W}_t^k)^T \mathbf{Q}_t(\mathbf{T})^k)) \quad (29)$$

由于  $\mathbf{B}^k \in \{-1, +1\}^{L \times n_k}$ , 对于  $\mathbf{B}^k$  的封闭解, 更新如下:

$$\mathbf{B}^k = \text{sgn}(v_v (\mathbf{W}_v^k)^T \mathbf{Q}_v(\mathbf{V})^k + v_t (\mathbf{W}_t^k)^T \mathbf{Q}_t(\mathbf{T})^k + (\mathbf{P}^k)^T \mathbf{F}^k) \quad (30)$$

将本文方法称为在线半监督锚图跨模态哈希(OS-AGCH), 其中中间变量统一设为  $\theta$ , 即  $\theta(k) = \{\mathbf{C}^k, \mathbf{D}_v^k, \mathbf{D}_t^k, \mathbf{E}_v^k, \mathbf{E}_t^k, \mathbf{G}^k, \mathbf{H}^k\}$ 。OSAGCH 算法的流程如算法 1 及算法 2 所示, 其中算法 1 表示第  $k$  阶段 OSAGCH 在线更新算法, 可单独用于第  $k$  阶段在线检索, 算法 2 表示在线检索整体流程。

#### 算法 1 在线更新算法

输入: 图像和文本特征矩阵  $\mathbf{V}$  和  $\mathbf{T}$ , 类标签矩阵  $\mathbf{Y}$ , 中间变量  $\theta(k-1)$ , 参数  $\epsilon_v, \epsilon_t, v_v, v_t, \eta, \alpha, \beta, \delta, \gamma$

输出: 统一哈希码  $\mathbf{B}^k$ , 中间变量  $\theta(k)$ , 特定模态投影矩阵  $\mathbf{W}_v^k$  和  $\mathbf{W}_t^k$

1. 通过 RBF 映射得到  $\mathbf{Q}_v(\mathbf{V})^k, \mathbf{Q}_t(\mathbf{T})^k$ 。
2. 通过式(13)对  $\mathbf{Q}_v^k, \mathbf{Q}_t^k$  加权相加计算得总局部权重矩阵  $\mathbf{Q}_v^k$ 。
3. 更新中间变量  $\mathbf{C}^k$ , 根据式(15)对邻接矩阵进行在线更新, 通过式(16)~式(18)对未标记数据进行标签预测, 得到预测标签矩阵  $\mathbf{F}^k$ 。
4. 随机初始化  $\mathbf{B}^k$ 。
5. while 不收敛 do
6. 固定其他变量, 更新中间变量  $\mathbf{D}_v^k, \mathbf{D}_t^k, \mathbf{E}_v^k, \mathbf{E}_t^k$ , 根据式(22)在线更新投影矩阵  $\mathbf{W}_v^k, \mathbf{W}_t^k$ 。
7. 固定其他变量, 更新中间变量  $\mathbf{G}^k, \mathbf{H}^k$ , 根据式(27)在线更新关系投影矩阵  $\mathbf{P}^k$ 。
8. 固定其他变量, 根据式(30)在线更新哈希码  $\mathbf{B}^k$ 。
9. 回到步骤 5。

#### 算法 2 在线检索整体流程

输入: 图像和文本特征矩阵  $\mathbf{V}$  和  $\mathbf{T}$ , 类标签矩阵  $\mathbf{Y}$ , 参数  $\epsilon_v, \epsilon_t, v_v, v_t, \eta, \alpha, \beta, \delta, \gamma$

输出: 统一哈希码  $\mathbf{B}$ , 特定模态投影矩阵  $\mathbf{W}_v^m, \mathbf{W}_t^m$ 。

1. 初始化  $\mathbf{B} = \mathbf{O}, \theta(0) = \mathbf{O}$ 。
2. for  $k=1:m$  do
3. 利用算法 1 更新中间变量  $\theta(k-1)$ , 得到新的中间变量  $\theta(k)$ 。根据式(22)更新投影矩阵  $\mathbf{W}_v^k, \mathbf{W}_t^k$ 。
4. 更新统一哈希码  $\mathbf{B} = [\mathbf{B}, \mathbf{B}^k]$ 。
5. end for。

#### 3.5 时间复杂度分析

算法 1 中, 对于特定模态, 假设  $\tilde{\mathbf{B}} = [\mathbf{B}^1, \dots, \mathbf{B}^{k-1}]$ ,  $\widetilde{\mathbf{Q}(\mathbf{X})} = [\mathbf{Q}(\mathbf{X})^1, \dots, \mathbf{Q}(\mathbf{X})^{k-1}]$ 。由于  $\tilde{\mathbf{B}}$  和  $\widetilde{\mathbf{Q}(\mathbf{X})}$  与新数据点无关, 且  $\widetilde{\mathbf{Q}(\mathbf{X})}\widetilde{\mathbf{Q}(\mathbf{X})}^T$  及  $\widetilde{\mathbf{Q}(\mathbf{X})}\tilde{\mathbf{B}}^T$  的结果均已在之前的阶段计算, 在第  $k$  阶段只需要计算  $\mathbf{Q}(\mathbf{X})^k (\mathbf{Q}(\mathbf{X})^k)^T$  以及  $\mathbf{Q}(\mathbf{X})^k (\mathbf{B}^k)^T$ , 则式(22)的计算复杂度与新数据点的数量成线性关系。因此, 可以在  $O(n_k)$  的时间内计算得  $\mathbf{W}_v^k, \mathbf{W}_t^k, \mathbf{P}^k$  和  $\mathbf{B}^k$  与之类似。在算法 1 的基础上, 算法 2 整个流程的计算复杂度与所有数据点的数量  $n$  成线性关系, 即可在  $O(n)$  的时间内完成计算。

## 4 实验

本节通过对比实验来评估所提方法。为了验证所提出的

OSAGCH 的性能,本节在 3 个广泛使用的数据集(MSCOCO2017, MIRFlickr, NUS-WIDE)上与 5 种先进的方法进行了跨模态检索对比实验,其中文献[27]对数据集进行了相应的特征提取处理。

#### 4.1 数据集

MSCOCO2017<sup>[28]</sup>是一个大规模数据集,包含 115 525 个实例数据,每个实例数据包含一个图像和文本,分为 80 个类别。对于每个实例,通过基于 VGG 网络的 CNN 对图像进行特征提取,表示为 4096 维的特征向量,通过 BoW 对文本进行特征提取,表示为 2000 维的特征向量。随机选取数据集中的 10000 个数据作为测试数据进行查询,将剩下的数据用于训练哈希函数及构建检索数据库。从训练数据的每一类中随机抽取 10% 作为标记数据。其余为未标记数据。

MIRFlickr<sup>[29]</sup>是一个中规模数据集,包含从 Flickr 下载的 17772 个实例数据,每个实例数据包含一个图像和相应的文本,分为 24 个类别。对于每个实例,通过基于 VGG 网络的 CNN 对图像进行特征提取,表示为 4096 维的特征向量,通过 BoW 对文本进行特征提取,表示为 1386 维的特征向量。随机选取数据集中的 5000 个数据作为测试数据进行查询,剩下的数据用于训练哈希函数及构建检索数据库。从训练数据的每一类中随机抽取 10% 作为标记数据,其余为未标记数据。

NUS-WIDE<sup>[30]</sup>是一个大规模图像数据集,包含 193 749 个实例数据,每个实例数据包含一个图像和相应的文本,分为 21 个类别。对于每个实例,通过基于 VGG 网络的 CNN 对图像进行特征提取,表示为 4096 维的特征向量,通过 BoW 对文本进行特征提取,表示为 1000 维的特征向量。随机选取数据集中的 10000 数据作为测试数据进行查询,剩下的数据用于训练哈希函数及构建检索数据库。从训练数据的每一类中随机抽取 10% 作为标记数据,其余为未标记数据。

#### 4.2 评价指标

本实验进行两个跨模态检索任务,即通过图像查询文本以及通过文本查询图像。为了测试不同方法的性能,采用目前广泛应用的平均精度(mAP)<sup>[31]</sup>作为评价指标,其中 mAP 返回所有结果进行计算,即 mAP-all。除此以外,本实验还采用了另外两种评价指标:1)反映精度随检索实例数量变化的 top-K 精度曲线<sup>[32]</sup>;2)反映准确率与召回率关系的 P-R 曲线<sup>[33]</sup>。一般来说,这 3 个评价指标数值越大,结果就越好。

#### 4.3 对比方法

本节将提出的 OSAGCH 与以下 5 种跨模态哈希方法进行了比较。

鲁棒和灵活的离散哈希(Robust and Flexible Discrete Hashing, RFDH)<sup>[34]</sup>;通过离散矩阵分解直接学习哈希码,并引入  $l_{2,1}$  范数来提高鲁棒性。

联合和独立矩阵分解哈希(Joint and Individual Matrix Factorization Hashing, JIMFH)<sup>[35]</sup>;通过联合矩阵和单个矩阵分别学习统一哈希码和单个哈希码,最后将两者结合得到最终哈希码。

离散跨模态哈希(Discrete Cross-modal Hashing, DCH);将特定模态的哈希函数和统一哈希码转化为联合优化问题,并提出一种高效离散优化算法学习哈希码。

跨模态哈希的子空间关系学习(Subspace Relation Learning for Cross-modal Hashing, SRLCH)<sup>[36]</sup>;利用语义标签使不同模态的相似数据更加接近,保留离散约束以有效提高训练效率。

其中,RFDH 和 JIMFH 是无监督方法,DCH 和 SRLCH 是有监督方法。所有的对比方法由官方网址提供代码,根据相应文献中的参数对对比方法进行实验。这 4 种跨模态哈希方法均为离线方法,在每个阶段训练新数据时需要调用先前阶段的所有旧数据一起训练。此外,由于现有在线跨模态哈希方法较少,本节将相关工作中介绍的 HMOH 由在线单模态哈希方法简单拓展为在线跨模态哈希方法,即将文本和图像的目标函数相加得到总目标函数,并将新方法命名为哈达玛矩阵引导在线跨模态哈希(Hadamard Matrix Guided Online Cross-modal Hashing, HMOCH)。OSAGCH 和上述方法的对比实验在一台拥有 Intel 2.4 GHz 处理器、8GB 内存和 64 位 Windows 10 操作系统的个人电脑上进行。

#### 4.4 参数设置

本节介绍本文中各参数的设置,将阶段数  $k$  设置为 5,锚点数  $e$  设为 500。关于核映射,参数  $\eta$  在值不是特别小时对结果不敏感,因此在两个核函数中都将  $\eta$  设为 0.6。为了适应不同模态特征,根据经验将  $\epsilon_o, \epsilon_t, v_o, v_t$  分别设置为 0.9, 0.1,  $10^{-1}, 10^{-5}$ 。为了简化计算,将式(22)中的  $\alpha, \beta$  替换为  $\eta_o, \eta_t$ ,并根据经验将所有实验的  $\delta, \alpha, \beta$  固定为 0.1, 0.1 和 0.01。对于  $\gamma$ ,将其从 0.001 变化到 1000,得到  $\gamma$  为 1 时性能最好。

#### 4.5 实验结果

本节通过改变 3 个数据集上的哈希码长度将 OSAGCH 与对比方法做比较,范围为 16 位到 128 位,表 1—表 3 分别列出了 3 个数据集上的 mAP 结果,其中 I to T 表示图像检索文本, T to I 表示文本检索图像,加粗字体表示最优结果。哈希码长度为 64 位时的 P-R 曲线如图 2—图 4 所示, top-K 曲线如图 5—图 7 所示。

表 1 NUS-WIDE 数据集上的 mAP  
Table 1 mAP on NUS-WIDE dataset

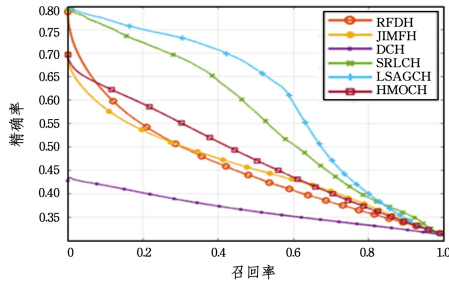
Algorithm	I to T				T to I			
	16-bit	32-bit	64-bit	128-bit	16-bit	32-bit	64-bit	128-bit
JIMFH	0.4064	0.4460	0.4846	0.4332	0.4118	0.4345	0.4714	0.4161
RFDH	0.4731	0.4615	0.4776	0.4961	0.4626	0.4538	0.4720	0.4875
DCH	0.3627	0.3680	0.3824	0.4069	0.3998	0.4068	0.4254	0.4611
SRLCH	0.5786	0.6099	0.6149	0.6322	<b>0.5658</b>	0.5961	0.6008	0.6156
HMOCH	0.4180	0.4913	0.5030	0.5083	0.4245	0.5030	0.5135	0.5204
OSAGCH	<b>0.5888</b>	<b>0.6677</b>	<b>0.6534</b>	<b>0.6750</b>	0.5405	<b>0.6140</b>	<b>0.6043</b>	<b>0.6245</b>

表2 MIRFlickr数据集上的mAP  
Table 2 mAP on MIRFlickr dataset

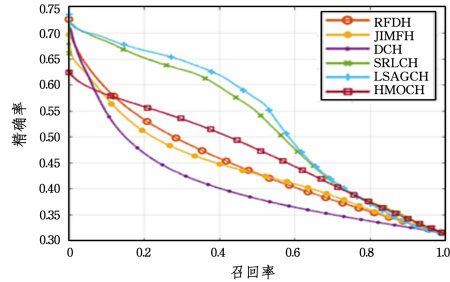
Algorithm	I to T				T to I			
	16-bit	32-bit	64-bit	128-bit	16-bit	32-bit	64-bit	128-bit
JIMFH	0.6493	0.6279	0.6291	0.6083	<b>0.6512</b>	0.6448	0.6310	0.6135
RFDH	0.6246	0.6235	0.6304	0.6344	0.6476	0.6517	0.6522	0.6526
DCH	0.6407	0.6438	0.6551	0.6683	0.6324	0.6328	0.6452	0.6553
SRLCH	0.6387	0.6496	0.6696	0.6772	0.6331	0.6198	0.6479	0.6498
HMOCH	0.6125	0.6267	0.6326	0.6333	0.6057	0.6243	0.6275	0.6287
OSAGCH	<b>0.6793</b>	<b>0.7244</b>	<b>0.7371</b>	<b>0.7434</b>	0.6390	<b>0.6739</b>	<b>0.6873</b>	<b>0.6895</b>

表3 MSCOCO2017数据集上的mAP  
Table 3 mAP on MSCOCO2017 dataset

Algorithm	I to T				T to I			
	16-bit	32-bit	64-bit	128-bit	16-bit	32-bit	64-bit	128-bit
JIMFH	0.4246	0.4070	0.4224	0.4287	0.4241	0.4339	0.4383	0.4384
RFDH	0.3952	0.3973	0.3983	0.3990	0.3978	0.3992	0.3997	0.4005
DCH	0.4361	0.4378	0.4304	0.4291	0.4227	0.4246	0.4315	0.4409
SRLCH	0.3921	0.3989	0.4029	0.4070	0.4022	0.4131	0.4189	0.4231
HMOCH	0.3451	0.3469	0.3285	0.3259	0.3961	0.3995	0.4016	0.4033
OSAGCH	<b>0.4509</b>	<b>0.4548</b>	<b>0.4584</b>	<b>0.4808</b>	<b>0.4962</b>	<b>0.5219</b>	<b>0.5185</b>	<b>0.5356</b>



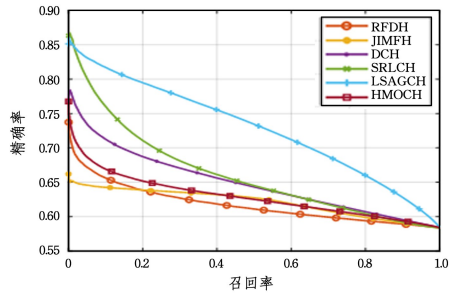
(a) Image query text



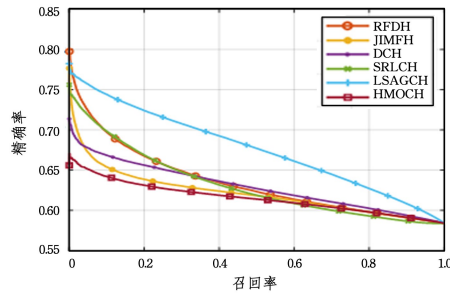
(b) Text query image

图2 NUS-WIDE数据集上的64位P-R曲线

Fig. 2 64-bit P-R curve on NUS-WIDE dataset



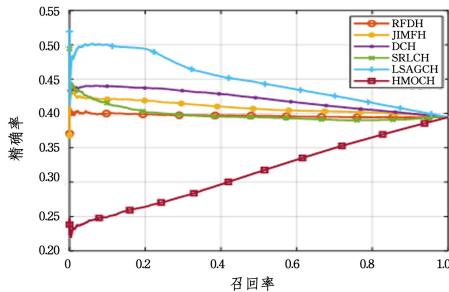
(a) Image query text



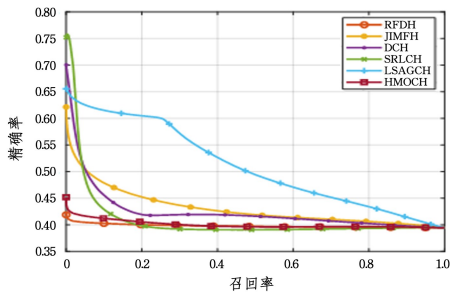
(b) Text query image

图3 MIRFlickr数据集上的64位P-R曲线

Fig. 3 64-bit P-R curve on MIRFlickr dataset



(a) Image query text



(b) Text query image

图4 MSCOCO2017数据集上的64位P-R曲线

Fig. 4 64-bit P-R curve on MSCOCO2017 dataset

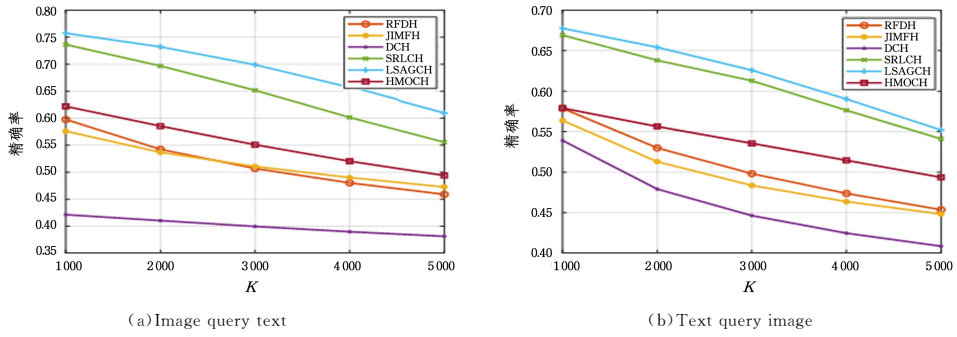


图 5 NUS-WIDE 数据集上的 64 位 top-K 曲线  
Fig. 5 64-bit top-K curve on NUS-WIDE dataset

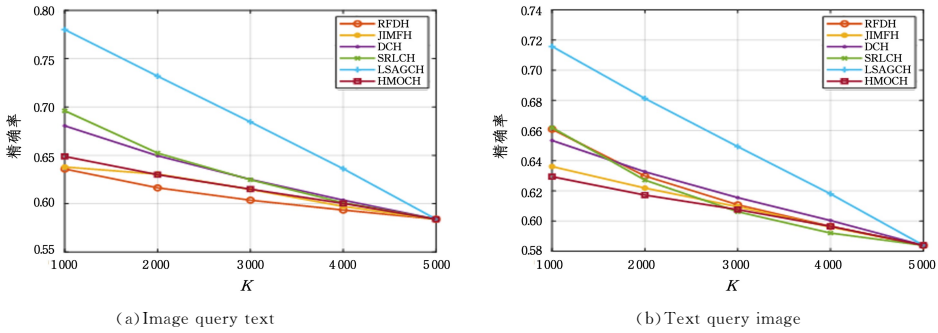


图 6 MIRFlickr 数据集上的 64 位 top-K 曲线  
Fig. 6 64-bit top-K curve on MIRFlickr dataset

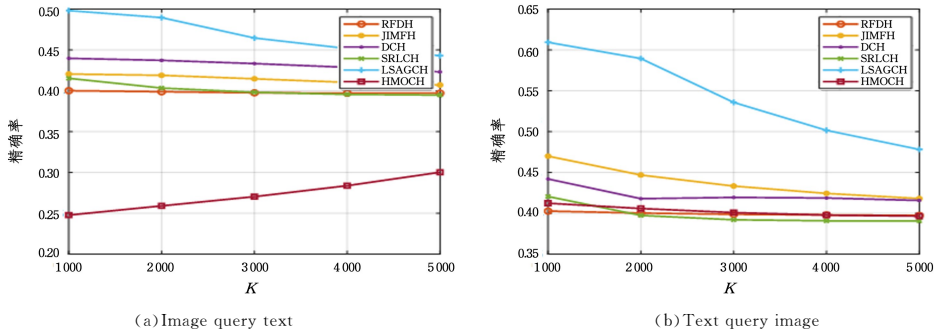


图 7 MSCOCO2017 数据集上的 64 位 top-K 曲线  
Fig. 7 64-bit top-K curve on MSCOCO2017 dataset

由表 1—表 3 可知,在 3 个数据集上,OSAGCH 对两个不同检索任务的 mAP 值大于大多数监督和非监督方法。在 NUS-WIDE 数据集上,OSAGCH 图像检索文本的 mAP 值对比的最佳方法 SRLCH 平均提高了 4%,文本检索图像的 mAP 值涨幅并不明显。同时,可以看到 JIMFH 在两个检索任务上哈希码长度为 128 位时的 mAP 值对比 64 位出现了一定程度的下降,说明该方法在 NUS-WIDE 数据集上 64 位时性能最好,而另外 3 种方法的 mAP 值均随哈希码变长而增大。在 MIRFlickr 数据集上,与对比方法相比,OSAGCH 在两个检索任务上的 mAP 值均有着明显提升,其中图像检索文本时比最佳对比方法 SRLCH 平均提高了 6%,文本检索图像时比最佳对比方法 SRLCH 平均提高了 4%,可见本文提出的 OSAGCH 能够充分利用标记数据及未标记数据,因此能够在半监督检索任务上获得比对比的有监督方法和无监督方法更好的结果。与前两个数据集相比,OSAGCH 在 MSCOCO2017 数据集上图像检索文本的 mAP 值与最佳对比方法

DCH 的提升幅度相似,但在文本检索图像时有着大幅度的提高,例如在哈希码长度为 128 位时,OSAGCH 的 mAP 值相比最佳的对比方法 DCH 提高了 11%。观察表 1—表 3 可知,在文本检索图像任务上,哈希码长度为 16 位时,OSAGCH 在 NUS-WIDE 和 MIRFlickr 数据集上未能取得最高的 mAP 值,说明在这两个数据集的文本检索图像任务上较小的哈希码长度对 OSAGCH 的性能有一定影响。然而,由于 16 位的哈希码长度较短,哈希码中蕴含着较少的编码信息,因此无法充分展示 OSAGCH 的性能优势。在哈希码长度大于 16 时,OSAGCH 具有明显优势,在实际应用中数据量更大的情况下一般会使用更长的哈希码长度。综上,在大多数情况下,OSAGCH 在 3 个数据集的两个检索任务上均保持着最佳的 mAP 值,证明了 OSAGCH 在半监督检索任务上的优越性能。另外,本节在图 2—图 4 中分别绘制了 3 个数据集上哈希码长度为 64 位时的 P-R 曲线。一般来说,精确率和召回率呈负相关关系,从而 P-R 曲线与坐标轴所围区域面积的大小

在一定程度上可以反映方法的性能,即区域面积越大,性能就越好。显然,除了极个别情况,OSAGCH在3个数据集的两个检索任务上总是保持最好的精确率-召回率指标。在同一召回率时,与无监督方法RFDH和JIMFH相比,OSAGCH有着较高的精确率,说明标记数据中的监督信息能够有效地提升方法性能;与监督方法DCH,SRLCH和HMOCH相比,OSAGCH依旧有着更好的精确率,说明通过正则化锚图进行标签预测使其能够充分利用未标记数据信息,因此能够获得比监督方法更好的检索性能。图5—图7分别显示了3个数据集上对于64位哈希码的两个检索任务在样本数从1000到5000时精确率的变化。可以从图中看到,OSAGCH在图像检索文本和文本检索图像任务的3个数据集上均表现出了很好的性能,而有些对比方法仅在特定的检索任务和数据集上具有竞争力。例如,DCH在MSCOCO2017数据集图像检索文本任务上与对比方法相比取得了最好的性能,但在NUS-WIDE数据集图像检索文本任务上性能较差,同时在MSCOCO2017数据集文本检索图像任务上性能差于JIMFH。此外,从图7(a)中可以看到,由于受数据类别数的影响,HMOCH在MSCOCO2017数据集图像检索文本任务上

性能较差,而OSAGCH依旧保持着最佳的检索性能。上述情况说明了OSAGCH在半监督检索任务中的有效性及普适性。

#### 4.6 参数敏感度分析

为研究参数对算法检索精度的影响,本节对参数敏感度进行了分析。实验设置 $e \in \{10, 100, 500, 1000, 2000\}$ ,  $v_v, v_t \in [10^{-6}, 10^{-1}]$ ,  $\delta, \alpha, \beta \in [10^{-6}, 10^{-1}]$ ,  $\epsilon_v, \epsilon_t \in [0.1, 0.9]$ ,  $\gamma \in [10^{-3}, 10^3]$ 。实验中固定其他参数不变,只改变其中一个参数值。图8给出了MIRFlickr数据集上哈希码长度为64位时各参数值的mAP变化曲线。观察图8(a)可知, $k$ 的取值对算法并不敏感,说明使用不同阶段数对本文方法影响不大,因此选取中等表现将阶段数设为5。此外,除 $e, \epsilon_v, \epsilon_t, \gamma$ 外,其余参数对本文方法也是不敏感的,分别在 $10^{-1}, 10^{-5}, 10^{-1}, 10^{-1}, 10^{-2}$ 处达到相对最佳检索性能,因此设置参数 $v_v = 10^{-1}, v_t = 10^{-5}, \delta = 10^{-1}, \alpha = 10^{-1}, \beta = 10^{-2}$ 。由图8(b)可知, $e$ 达到500后mAP值变化趋于平稳,因此设置 $e = 500$ 。由图8(h)—图8(j)可知, $\epsilon_v, \epsilon_t$ 和 $\gamma$ 的变化对检索性能影响较大,但分别于区间 $[0.4, 0.9], [0.1, 0.6], [10^{-3}, 1]$ 内变化趋于平稳且性能较好,因此设置参数 $\epsilon_v = 0.9, \epsilon_t = 0.1, \gamma = 1$ 。

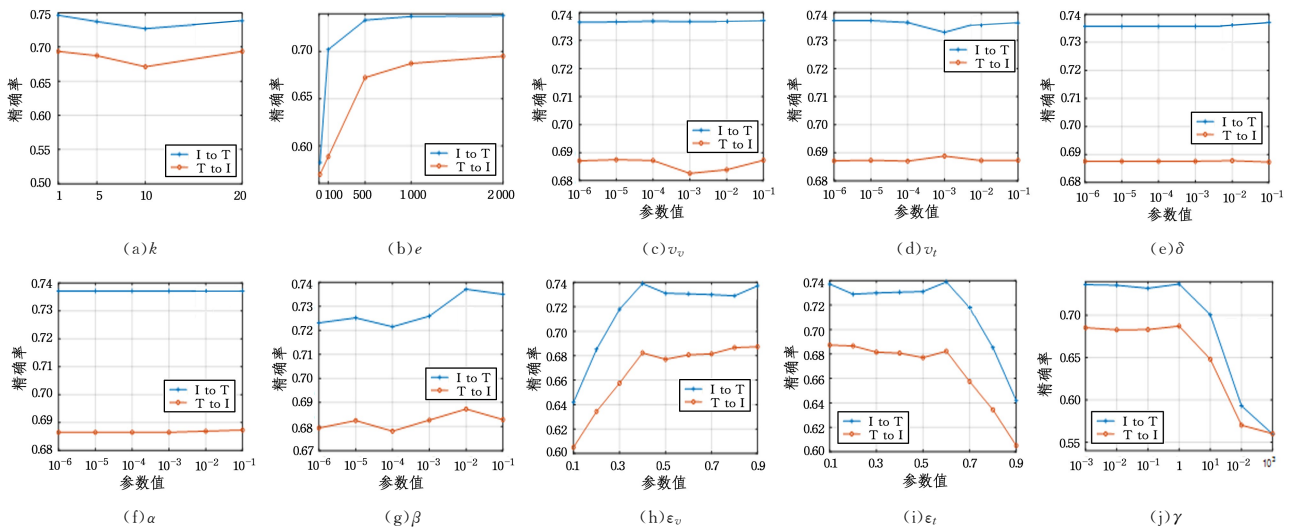


图8 各参数的敏感度分析

Fig. 8 Sensitivity analysis of each parameter

#### 4.7 训练速度

本节在MIRFlickr数据集上测试了第5阶段不同哈希码长度时各种方法的训练时间(单位为s),如表4所列。为了与在线监督跨模态哈希方法HMOCH进行对比,实验中设置HMOCH均为标记数据输入。从表中可以看到,SRLCH在各哈希码长度下都具有十分有竞争力的速度,然而其在训练时只对标记数据进行了训练,大大缩短了训练时间,DCH也是如此。OSAGCH相比SRLCH只增加了很少的时间,但在效果上有着较高提升,具有合理性。与对比的离线方法在每一阶段需要调用所有先前阶段旧数据一起训练相比,OSAGCH及HMOCH只需训练当前阶段的新数据,有效地提高了训练效率。除了SRLCH以外,OSAGCH比其他所有对比方法都有着更快的训练速度,甚至在哈希码长度为32位到128位时比最佳对比方法HMOCH快了约0.5~3倍。

表4 MIRFlickr数据集上的训练时间

Table 4 Training time on MIRFlickr dataset

(单位:s)

Algorithm	Training time			
	16-bit	32-bit	64-bit	128-bit
JIMFH	37.59	38.81	43.89	49.67
RFDH	81.55	88.98	115.89	200.09
DCH	7.68	7.92	8.01	8.19
SRLCH	0.47	0.65	0.84	1.32
HMOCH	0.70	1.75	3.85	7.60
OSAGCH	1.12	1.20	1.41	1.74

**结束语** 本文提出了一种新的在线半监督跨模态哈希方法,称为在线半监督锚图跨模态哈希(OSAGCH),用于高效的多媒体检索。具体来说,为了能充分利用所有数据,引入正则化锚图,对未标记数据进行标签预测,最终得到所有数据标签。为了解释语义标签,将各数据的类标签投影到子空间中,

通过子空间关系学习在公共汉明空间中学习两种模态的统一哈希码和各自的哈希函数。在算法的优化过程中,提出了一种在线学习算法,使哈希函数及哈希码能够实时更新,适用于流数据的输入,同时在一个步骤中生成哈希码,而不需要逐位计算,加快了训练的速度,提高了跨模态检索的效率。实验结果表明,在只有部分数据有标签的情况下,对于两个跨模态检索任务,OSAGCH 在性能上优于许多现有的跨模态哈希方法。同时,相比离线跨模态哈希方法,OSAGCH 在在线检索任务中有着更快的训练速度及更低的空间成本。

本文的研究工作虽一定程度上解决了该领域存在的一些问题,但仍有改进之处,未来将从以下几个方向进行研究。

(1)深度学习由于其丰富的表示能力及超强的计算能力,可以很好地适用于大规模数据集,在跨模态检索中展现出了巨大的潜力,但深度学习在在线检索中时间开销较大,将其应用于跨模态在线哈希具有挑战性。在以后的工作中将考虑引入深度学习方法,设计时间开销低、适合跨模态检索的学习方法,推导更具区分性的哈希函数,提升跨模态检索性能。

(2)目前跨模态检索研究大多更注重检索结果的准确性,但网上充斥着大量虚假信息,会对检索结果的可靠性造成极大影响。因此,今后将考虑如何在保证检索精度的情况下,对虚假信息进行过滤,保证检索结果真实有效。

## 参 考 文 献

- [1] WANG J, ZHANG T, SEBE N, et al. A survey on learning to hash [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, 40(4): 769-790.
- [2] GORISSE D, CORD M, PRECIOSO F. Locality-sensitive hashing for chi2 distance [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011, 34(2): 402-409.
- [3] MATSUSHITA Y, WADA T. Principal component hashing: an accelerated approximate nearest neighbor search [C] // *Advances in Image and Video Technology*. Tokyo: Springer, 2009: 374-385.
- [4] GONG Y C, LAZEBNIK S, GORDO A, et al. Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval [J]. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2013, 35(12): 2916-2929.
- [5] LIU L, YU M Y, SHAO L. Unsupervised local feature hashing for image similarity search [J]. *IEEE Trans. Cybern.*, 2015, 46(11): 2548-2558.
- [6] HEO J P, LEE Y, HE J F, et al. Spherical hashing: binary code embedding with hyperspheres [J]. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 2015, 37(11): 2304-2316.
- [7] STRECHA C, BRONSTEIN A, BRONSTEIN M, et al. LDA-Hash: improved matching with smaller descriptors [J]. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011, 34(1): 66-78.
- [8] LIU W, WANG J, JI R R, et al. Supervised Hashing with Kernels [C] // *IEEE Conference on Computer Vision and Pattern Recognition*. Providence: CVPR, 2012: 2074-2081.
- [9] NGUYEN V A, DO M N. Deep learning based supervised hashing for efficient image retrieval [C] // *IEEE International Conference on Multimedia and Expo*. Seattle: ICME, 2016: 1-6.
- [10] WANG J, KUMAR S, CHANG S F. Semi-supervised hashing for large-scale search [J]. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012, 34(12): 2393-2406.
- [11] WU C X, ZHU J K, CAI D, et al. Semi-supervised nonlinear hashing using bootstrap sequential projection learning [J]. *IEEE Trans. Knowl. Data Eng.*, 2013, 25(6): 1380-1393.
- [12] GAO L L, SONG J K, ZOU F H, et al. Scalable multimedia retrieval by deep learning hashing with relative similarity learning [C] // *Proceedings of the 23rd ACM International Conference on Multimedia*. New York: ACM, 2015: 903-906.
- [13] ZHOU X C, TIAN X, WANG X Z, et al. Bagging-boosting-based semi-supervised multi-hashing with query-adaptive re-ranking [J]. *Neurocomputing*, 2018, 275(10): 916-923.
- [14] ZHANG S F, LI J M, ZHANG B. Pairwise teacher-student network for semi-supervised hashing [C] // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. New York: IEEE, 2019: 730-737.
- [15] ANCULEF R, MENA F, MACALUSO A, et al. Self-Supervised Bernoulli Autoencoders for Semi-Supervised Hashing [C] // *Iberoamerican Congress on Pattern Recognition*. Porto: Springer, 2021: 258-268.
- [16] WANG G A, HU Q H, CHENG J, et al. Semi-supervised generative adversarial hashing for image retrieval [C] // *Proceedings of the 15th European Conference on Computer Vision*. Munich: ECCV, 2018: 469-485.
- [17] YU J, WU X J, KITTLER J. Semi-supervised hashing for semi-paired cross-view retrieval [C] // *Proceedings of the 24th International Conference on Pattern Recognition (ICPR)*. Beijing: IEEE, 2018: 958-963.
- [18] SONG T C, CAI J F, ZHANG T Q, et al. Semi-supervised manifold-embedded hashing with joint feature representation and classifier learning [J]. *Pattern Recognition*, 2017, 68(2): 99-110.
- [19] XU X, SHEN F M, YANG Y, et al. Learning discriminative binary codes for large-scale cross-modal retrieval [J]. *IEEE Transactions on Image Processing*, 2017, 26(5): 2494-2507.
- [20] ETO K, KOUTAKI G, SHIRAI K. Hadamard Coded Discrete Cross Modal Hashing [C] // *2018 25th IEEE International Conference on Image Processing (ICIP)*. Switzerland: IEEE, 2018: 2007-2011.
- [21] YU G X, LIU X W, WANG J, et al. Flexible Cross-Modal Hashing [J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2020, 1(1): 1-11.
- [22] LIN Z J, DING G G, HAN J G, et al. Cross-view retrieval via probability-based semantics-preserving hashing [J]. *IEEE Transactions on Cybernetics*, 2016, 47(12): 4342-4355.
- [23] JIANG Q Y, LI W J. Deep cross-modal hashing [C] // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Hawaii: CVPR, 2017: 3232-3240.
- [24] HUANG L K, YANG Q, ZHENG W S. Online hashing [J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2017, 29(6): 2309-2322.
- [25] LIN M B, JI R R, LIU H, et al. Supervised online hashing via hadamard code-book learning [C] // *Proceedings of the 26th*

- ACM International Conference on Multimedia, Seoul Republic: ACM, 2018: 1635-1643.
- [26] LIN M B, JI R R, LIU H, et al. Hadamard matrix guided online hashing [J]. International Journal of Computer Vision, 2020, 128(8): 2279-2306.
- [27] ZHENG C, ZHU L, ZHANG S, et al. Efficient parameter-free adaptive multi-modal hashing [J]. IEEE Signal Processing Letters, 2020, 27(9): 1270-1274.
- [28] LIN T Y, MAIRE M, BELONGIE S, et al. Microsoft coco: Common objects in context[C]//European Conference on Computer vision. Zurich: Springer, Cham, 2014: 740-755.
- [29] HUISKES M J, LEW M S. The mir flickr retrieval evaluation [C]//Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval. Canada: ACM, 2008: 39-43.
- [30] CHUA T S, TANG J, HONG R, et al. Nus-wide: a real-world web image database from national university of singapore[C]//Proceedings of the ACM International Conference on Image and Video Retrieval. Greece: CIVR, 2009: 1-9.
- [31] SONG J, YANG Y, HUANG Z, et al. Effective multiple feature hashing for large-scale near-duplicate video retrieval [J]. IEEE Transactions on Multimedia, 2013, 15(8): 1997-2008.
- [32] LU X, ZHU L, LI J, et al. Efficient supervised discrete multi-view hashing for large-scale multimedia search [J]. IEEE Transactions on Multimedia, 2019, 22(8): 2048-2060.
- [33] SCHEFFER T, WROBEL S. Active learning of partially hidden markov models[C]//Proceedings of the ECML/PKDD Workshop on Instance Selection. Freiburg: ECML, 2001: 1-15.
- [34] WANG D, WANG Q, GAO X. Robust and flexible discrete hashing for cross-modal similarity search [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2017, 28(10): 2703-2715.
- [35] WANG D, WANG Q, HE L, et al. Joint and individual matrix factorization hashing for large-scale cross-modal retrieval [J]. Pattern Recognition, 2020, 107(4): 7-19.
- [36] LIU L, YANG Y, HU M, et al. Index and retrieve multimedia data: Cross-modal hashing by learning subspace relation[C]//International Conference on Database Systems for Advanced Applications. Australia: Springer, 2018: 606-621.



**QIN Liang**, born in 1996, postgraduate. His main research interests include machine learning and cross-modal retrieval.



**XIE Liang**, born in 1987, Ph.D, associate professor. His main research interests include multimedia retrieval and machine learning.

(责任编辑:喻黎)