基于排序的监督离散跨模态哈希

李慧琼 王永欣 陈振铎 罗 昕 许信顺
(山东大学软件学院 济南 250101)

摘 要 近年来,随着信息技术的发展,图像、文本、视频、音频等多媒体数据呈现出快速增长的趋势.当处理大量数据时,某些传统检索方法的效率可能会受到影响,并且无法在可接受的时间内获得令人满意的准确性.此外,海量的数据还导致了巨大的存储消耗问题.为了解决上述问题,哈希学习被提出.现有的哈希学习方法首先为数据生成二进制哈希码,并且在学习中让原本相似的数据有相似的哈希码,让不相似的数据有不同的哈希码.然后,在学到的哈希码空间中,通过异或操作进行快速的相似性比较.通过用二进制哈希码代替数据原始的高维特征,可以达到显著降低存储成本的目的.基于哈希学习高效索引和快速查询的特点,其在跨模态检索领域受到了广泛的关注.但是目前的跨模态哈希方法面临着以下几个问题:(1)大多数方法都尝试保持样本间的成对相似性,而忽视了样本间的相对相似性,即样本的排序信息,但排序信息对检索有很重要的作用,因而导致这些方法效果并非最优;(2)许多基于成对相似性的哈希检索方法的时间复杂度为 $O(n^2)$,无法直接扩展到大规模数据集上,具有一定的局限性;(3)为了简化离散求解问题,目前很多方法采用松弛策略来学习哈希码的近似解,但这种策略会引入较大的量化误差.为了解决以上问题,我们提出了一种基于排序的监督离散跨模态哈希方法(简称为 RSDCH).该方法由排序信息学习和哈希学习两步骤组成.在排序信息学习阶段,我们通过嵌入数据的流形结构和语义标签来学习一个具有排序信息的得分矩阵.在哈希学习阶段,我们通过保持学到的排序信息来生成训练样本的哈希码并学出对应的哈希函数.为了让模型能够更好地扩展到大规模数据集,我们使用了锚点采样策略,以获得可接受的且与训练样本数成线性关系的时间复杂度.为了学到高质量的哈希码表示,我们设计了两种有效的相似性保持策略.除此之外,为了避免松弛求解策略引入的量化误差,我们设计了一种交替迭代的优化算法来离散地学习哈希码.我们在 MIRFlickr-25K 及 NUS-WIDE 这两种广泛使用的多标签数据集上进行了对比实验.结果表明,本文提出的方法在平均精确率均值(MAP)、归一化折损累计增益(NDCG)、精确率-召回率曲线(Precision-Recall Curve)等方面均优于现有的几种跨模态哈希方法.通过消融实验,我们验证了 RSDCH 模型中各个模块的必要性和有效性.此外,我们还通过额外的实验测试了模型的收敛性、参数敏感性和训练效率,进一步验证了 RSDCH 模型的有效性.

关键词 跨模态检索;哈希学习;排序哈希;离散优化;相似性保持
中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2021.01620

Ranking-Based Supervised Discrete Cross-Modal Hashing

LI Hui-Qiong WANG Yong-Xin CHEN Zhen-Duo LUO Xin XU Xin-Shun
(*School of Software, Shandong University, Jinan* 250101)

Abstract In recent years, with the development of information technology, the explosion of multimedia data such as images, texts, videos, audios, has occurred. When dealing with a huge amount of data, the efficiency of some traditional retrieval methods may be affected and cannot obtain satisfactory accuracy within an acceptable time. In addition, the massive amount of data has also caused huge storage consumption problems. In order to solve the above problems, hashing is

收稿日期:2020-10-30;在线出版日期:2021-04-07.本课题得到国家自然科学基金(61991411,61872428)、山东省重点研发项目(2019JZZY010127)、山东省自然科学基金项目(ZR2019ZD06,ZR2020QF036)、山东大学基本科研业务费专项资金(2019GN075)资助.
李慧琼,硕士研究生,主要研究方向为多媒体检索和计算机视觉.E-mail:Huiqiong.sdu@gmail.com.**王永欣**,博士研究生,主要研究方向为机器学习、哈希、跨模态检索和计算机视觉.**陈振铎**,博士研究生,主要研究方向为机器学习和信息检索.**罗 昕**(通信作者),博士,助理研究员,中国计算机学会(CCF)会员,主要研究方向为机器学习、智能媒体分析与检索.E-mail:luoxin.lxin@gmail.com.**许信顺**,博士,教授,中国计算机学会(CCF)会员,主要研究领域为机器学习、机器视觉、数据挖掘、信息检索和媒体内容分析与检索.

proposed. It first transforms data from original representations into binary codes, minimizing the Hamming distance of similar data points and maximizing that of dissimilar ones. Then, pairwise comparisons can be carried out extremely efficiently in the learned Hamming space, using XOR operations. Moreover, by representing data with binary codes rather than original high-dimensional features, the storage cost can be dramatically reduced. Due to the efficient indexing and quick query, hashing has received extensive attention in the field of cross-modal retrieval, and many cross-modal hashing methods have been proposed. However, there still exist some issues worthy of investigation for existing cross-modal hashing methods. (1) For example, most methods only consider the pairwise similarity between samples and ignore the ranking information. However, lack of ranking information may lead to sub-optimal performance since it is also important. (2) A lot of hashing methods employ a pairwise similarity matrix to preserve similarity, which makes the algorithm complexity $O(n^2)$ and cannot extend to large-scale datasets. (3) Besides, most methods relax the discrete constraint to solve the discrete optimization problem, which may introduce serious quantization error. To overcome the aforementioned issues, in this paper, we propose a new method named Ranking-based Supervised Discrete Cross-modal Hashing (RSDCH for short). RSDCH consists of ranking learning step and hashing learning step. In the first step, the proposed method learns ranking information from the manifold structure and semantic labels of data and generates a ranking score matrix. In the second step, RSDCH jointly learns hash codes and hash functions while preserving the learned ranking information. To make our method scalable to large-scale datasets, anchor sampling is leveraged and the time complexity of our method is linear to the number of training samples. To learn high-quality hash codes, two effective similarity-preserving strategies are proposed. To avoid large quantization error, an alternative optimization algorithm, which discretely solves the binary codes learning problem, is designed. We conducted comparative experiments on two widely-used multi-label datasets, i.e., MIRFlickr-25K and NUS-WIDE. To comprehensively evaluate our proposed method RSDCH, we adopted three evaluation metrics, i.e., Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG) and Precision-Recall Curve. The experimental results have shown that the proposed RSDCH is superior to several state-of-the-art methods, including both non-deep and deep cross-modal hashing methods. To further evaluate the effectiveness of our method, we also carried out ablation experiments in order to test the necessity and effectiveness of each module in the RSDCH model. Finally, the effectiveness of the model convergence, parameter sensitivity, and training efficiency were tested by additional experiments, and the results further demonstrate that the proposed method is effective.

Keywords　cross-modal retrieval; learning to hash; ranking-based hashing; discrete optimization; similarity preserving

1　引　言

随着互联网技术的飞速发展，通过网络进行社交、检索已经成为人们日常生活的一部分. 随着大数据时代的到来，管理和使用海量高维的多媒体数据的需求日渐增强. 对多媒体数据进行相似性检索是最基本的需求之一，然而，数据量大、数据维度高等特性会使得基于最近邻检索的方法面临响应时间长、反馈不及时的问题. 为了缓解检索效率低这一问题，近似最近邻检索技术通过牺牲一定程度的检索准确率从而大大提高了检索效率. 哈希检索方法作为代表性的近似最近邻检索技术之一，将数据表示成在海明空间中的低维度的二进制串，从而极大地降低了存储消耗，并且通过二进制串的异或操作来进行相似性计算，可以极大降低时间消耗，因而获得

了多媒体检索领域的广泛青睐.

只针对单一模态数据的哈希检索[1-4],比如以图搜图的哈希方法等最先受到关注.随着多媒体数据的迅速增长,用户更希望可以通过某一模态的数据查询到与之相关的另一模态的数据,比如通过几个特定的关键词可以找到与之对应的图像,即跨模态检索.由于多模态数据的异构性,比如文本、图像、音频、视频等不同模态的非结构化数据之间存在巨大的语义鸿沟,跨模态检索面临着巨大的挑战,如何挖掘和保持这些不同模态数据间的语义信息是当前跨模态检索研究的难点和热点.现有的跨模态哈希方法[5-8]依据是否采用了语义标签等监督信息可以分为两类:无监督哈希方法和有监督哈希方法.无监督哈希方法往往从数据的底层结构、特征分布中挖掘相似信息,如协同矩阵分解哈希(Collective Matrix Factorization Hashing,CMFH)[9]、综合相关量化(Composite Correlation Quantization,CCQ)[10]、融合相似哈希(Fusion Similarity Hashing,FSH)[5]和协同重构嵌入(Collective Reconstructive Embeddings,CRE)[11]等方法.有监督哈希方法[12-15]则利用语义标签作为监督信息,从而指导哈希函数和哈希码的学习,如语义保持哈希(Semantics Preserving Hashing,SePH)[6]、离散跨模态哈希(Discrete Cross-modal Hashing,DCH)[12]、基于矩阵分解的可扩展离散哈希(Scalable disCRete mATrix faCtorization Hashing,SCRATCH)[13]、标签一致矩阵分解哈希(Label Consistent Matrix Factorization Hashing,LCMFH)[14]等.由于语义标签等人工干预信息的介入,有监督哈希方法的检索效果往往要高于无监督哈希方法,但是目前大多数有监督跨模态哈希方法仍存在一些问题:(1)大多数有监督方法通过语义标签直接构建相似性矩阵,构建规则为两个样本只要共享一个标签即为相似,但是这样无法挖掘多标签数据集的相对相似性;(2)相似性矩阵的构造复杂度为 $O(n^2)$,使用全部训练数据参与模型训练会造成巨大甚至不可接受的空间和时间消耗,因此无法扩展到大规模数据集.大多数方法用随机抽取部分训练数据进行训练的方法来缓解这个问题,虽然这种方式可以让方法避免巨大的时空开销,但是会因训练数据使用不充分而导致方法的精度不足;(3)哈希码是海明空间中的二进制串,哈希码的求解问题天然具有离散的约束条件.大多数方法采用把哈希码取值松弛到连续值的策略,先学习一个实值变量,再将其二值化得到哈希码,但这样往往会导致较大的量化误差,无法学到高质量的哈希码.

针对以上这些问题,本文提出了一种新的有监督跨模态哈希方法.首先在预训练阶段,我们提出了一个可扩展的排序学习框架,将数据的低层视觉特征和高层语义标签信息联合嵌入到一个排序得分矩阵中.然后在哈希学习阶段,我们利用学到的得分矩阵去学习有效的哈希码和哈希函数.学习过程中我们设计了一个具有线性复杂度的算法,进而保证全部训练数据都可以被考虑到.此外我们还提出了一种离散的优化算法,避免了松弛方式带来的量化误差.本文主要工作总结如下:

(1)本文提出了一种全新的基于排序信息的有监督跨模态哈希方法(Ranking-based Supervised Discrete Cross-modal Hashing,RSDCH).我们将语义标签作为排序信息的度量标准,通过嵌入语义标签来学习一个具有排序信息的得分矩阵,然后将排序信息作为监督信息来训练哈希函数.

(2)为了让模型能够更好地扩展到大规模数据集,我们设计了两种基于锚点采样的相似性保持策略,在保证训练效率的同时,可以获得高质量的哈希码,进而保证模型的准确性.

(3)为了避免松弛策略带来的量化误差,我们设计了一种交替迭代的离散优化算法来学习有效的哈希码.

(4)在 MIRFlickr-25K 和 NUS-WIDE 两个通用的多模态数据集上的对比实验表明,我们的方法在表现出最好的检索效果的同时,保持着优越的训练效率,证实了本文方法的有效性和实用性.

2 相关工作

近年来,由于多模态数据的爆炸性增长,基于哈希学习的跨模态检索方法得到了广泛关注.本章首先介绍了与本文相关的基于排序的哈希方法,然后对部分非深度跨模态哈希学习方法进行简单介绍,最后简述了基于深度学习的跨模态哈希方法.

2.1 基于排序的哈希方法

对于多媒体检索而言,给定一个特定查询,考虑它 top$N$ 近邻的位置信息可以更好地优化检索效果.目前已经有很多基于排序的哈希方法被提出,比如列生成哈希(Column Generation Hashing,CGH)[16]、海明距离度量学习(Hamming Distance Metric Learning,HDML)[17]、排序保持哈希(Ranking Preserving Hashing,RPH)[18]、序数嵌入哈希(Ordinal

Embedding Hashing，OEH）[19]、序数约束哈希（Ordinal Constraint Hashing，OCH）[20]等. CGH[16]通过列生成策略来学习哈希函数，同时考虑了基于三元组的相对排序来辅助哈希函数学习；HDML[17]提出了基于哈希码的三元组排序损失函数并对其进行优化，从而得出哈希函数和对应的哈希码；RPH[18]直接将归一化折损累计增益（NDCG）作为损失函数进行优化，从而学习出高质量的哈希函数；OEH[19]通过构建一个有向无权图来获取位置信息，并借此学习哈希函数来保持排序关系；OCH[20]与 OEH[19]类似，构建张量位置图来近似位置关系，并借助这个位置图来学习哈希函数. 为了保持离散约束的同时能够优化非凸排序损失函数，上述基于排序的方法通常借助松弛技术来进行模型训练. 首先对离散约束进行松弛，然后对学到的实值变量进行二值化得到哈希码，但这会导致量化误差增大，从而导致学习到的哈希函数和哈希码质量较差. 一些基于离散技术的排序哈希也已经提出，如离散语义排序哈希（Discrete Semantic Ranking Hashing，DSeRH）[21]，但是这些方法只适用于单模态场景，很难扩展到跨模态场景下.

2.2　非深度跨模态哈希方法

跨模态哈希方法根据是否使用语义标签等监督信息，又可以划分为无监督和有监督方法. 无监督跨模态哈希方法主要倾向于只从多个模态数据本身出发，从底层结构、特征分布中挖掘相似信息，从而节省人力标注成本. Collective Matrix Factorization Hashing（CMFH）[9]通过矩阵分解学习原始数据中隐藏的语义特征，将不同模态的数据映射到一个统一的子空间；Composite Correlation Quantization（CCQ）[10]通过两个相关最大映射将图像和文本转换到同构隐藏空间中，并通过复合量化器将其量化为二值哈希码；Fusion Similarity Hashing（FSH）[5]通过一个多模态融合图来挖掘异构数据的相关性，并通过一个交替优化的图哈希框架学习哈希码和哈希函数；Collective Reconstructive Embeddings（CRE）[11]通过不同的投影嵌入模型来处理不同模态的数据，并采用集体重构嵌入的方式将其映射为统一的哈希码. 有监督跨模态哈希方法则倾向于使用人工标注的语义标签等作为监督信息，从而达到更好的检索效果. Semantics Preserving Hashing（SePH）[6]通过最小化 KL 散度使得待学习的哈希码能够保持数据语义相似性的概率分布；Discrete Cross-modal Hashing（DCH）[12]将语义标签信息直接应用到哈希学习中，通过离散循环坐标下降法，采用逐位迭代优化的方式达成离散学习哈希码的目的；Scalable disCRete mATrix factorization Hashing（SCRATCH）[13]利用矩阵分解和语义嵌入来同时进行哈希码和哈希函数的训练，同时训练过程中保持了哈希码的离散特性；Label Consistent Matrix Factorization Hashing（LCMFH）[14]为多模态数据学习一个潜在语义空间的同时保持它们的标签一致性.

2.3　深度跨模态哈希方法

近年来深度学习技术日趋成熟，效果也非常显著. 目前也已经有很多跨模态哈希方法[22-26]结合深度模型，取得了非常优越的检索效果. 如深度跨模态哈希（Deep Cross-Modal Hashing，DCMH）[22]首次将特征学习和哈希学习结合到一个统一的跨模态深度学习框架中；注意力深度对抗哈希（Attention-aware Deep Adversarial Hashing，ADAH）[23]提出了一个带有注意力机制的对抗哈希网络，有选择地关注多模态数据的部分信息来增强数据相似性的度量；半监督对抗哈希（Self Supervised Adversarial Hashing，SSAH）[24]通过一个自监督语义生成网络学习多标签数据的语义信息，并且采用对抗性学习的方式学习哈希码和哈希函数；基于排序的深度跨模态哈希（Ranking-based Deep Cross-Modal Hashing，RDCMH）[25]提出了一种基于标签和特征信息的半监督语义排序度量方法，将深度特征学习和语义排序信息相结合，在哈希学习中保持多标签数据的语义相似度.

3　基于排序的监督离散跨模态哈希

在本节，我们首先给出本文中涉及的一些符号的定义，随后我们将详细介绍本文提出的基于排序的监督离散跨模态哈希算法，并给出具体的优化过程，最后我们会分析算法的时间复杂度，证明 RSDCH 可以扩展到大规模数据集.

3.1　符号定义

假设我们有 n 个训练样本，每个样本都有 m 个模态，我们定义训练集为 $\boldsymbol{X}^t=\{\boldsymbol{x}_1^t,\boldsymbol{x}_2^t,\cdots,\boldsymbol{x}_n^t\}^\top\in\mathbb{R}^{n\times d_t}$，其中 $\boldsymbol{x}_i^t$ 是第 t 个模态的第 i 个样本，d^t 是第 t 个模态的特征维度，$t\in\{1,\cdots,m\}$. 此外我们定义 $\boldsymbol{B}\in\{-1,1\}^{n\times r}$ 为待学习的哈希码矩阵，其中 r 是

哈希码长度；$\boldsymbol{L}\in\{0,1\}^{n\times l}$ 为 n 个训练样本的标签矩阵，其中 l 是标签个数；$\boldsymbol{L}_q\in\{0,1\}^{q\times l}$ 为从训练样本中随机选取的 q 个锚点的标签矩阵；$\boldsymbol{A}^t\in\mathbb{R}^{n\times n}$ 为通过锚点构建的第 t 个模态的近似相似性矩阵；$\boldsymbol{F}\in\mathbb{R}^{n\times q}$ 为待学习的排序得分矩阵；$\boldsymbol{1}$ 表示值全为 1 的列向量；$\|\cdot\|_F$ 表示矩阵的 Frobeniu 范数；$sgn(\cdot)$ 是一个符号函数，如果 $x>0$，$sgn(x)=1$，否则 $sgn(x)=-1$.

3.2 损失函数

3.2.1 学习排序信息

近年来流形排序[27-28]在图像检索领域取得了不错的成果. 流形排序通过挖掘数据隐藏的流形结构来获取有效的排序信息，但是传统的流形排序算法采用无监督的方式去学习排序信息并且基于整个训练集做训练，无法扩展到较大规模的数据集. 为了从多模态数据中获取更有效的排序信息，我们提出了一种可扩展的监督排序学习框架，将标签信息整合到流形排序算法中，为多个模态的数据学习一个统一的排序得分矩阵. 具体来说，我们首先从 n 个训练样本中随机选取 q 个锚点作为查询样本，用第 t 个模态的所有样本和选取的锚点之间的欧氏距离，来近似计算所有样本之间的相似性矩阵 $\boldsymbol{A}^t$. 我们定义 $\boldsymbol{A}^t=\boldsymbol{Z}^t\boldsymbol{\Omega}^{-1}\boldsymbol{Z}^{t\top}$，其中 $\boldsymbol{\Omega}=diag(\boldsymbol{Z}^{t\top}\boldsymbol{1})$，$\boldsymbol{Z}^t$ 表示 n 个训练样本和 q 个锚点之间的相似性. 我们采用锚点图哈希(Anchor Graph Hashing，AGH)[29]中的锚点采样来计算 $\boldsymbol{Z}^t$，并且设计了下面的式(1)学习一个有效的排序得分矩阵. 式(1)的第一项表示在第 t 个模态中特征分布相似的样本点应该获得相似的得分，第二项表示学得的得分矩阵应该保持样本的标签相似性.

$$\min_{\boldsymbol{F}}\frac{1}{2}\sum_{t=1}^{m}\left(\sum_{i,j=1}^{n}\boldsymbol{A}_{ij}^t\left\|\frac{1}{\sqrt{\boldsymbol{D}_{ii}^t}}\boldsymbol{F}_{i*}-\frac{1}{\sqrt{\boldsymbol{D}_{jj}^t}}\boldsymbol{F}_{j*}\right\|^2+\sum_{k=1}^{q}\|\boldsymbol{F}_{*k}-\boldsymbol{Y}_{*k}\|^2\right)\tag{1}$$

其中 $\boldsymbol{A}_{ij}^t$ 是第 t 个模态的第 i 个样本和第 j 个样本的相似度；$\boldsymbol{D}_{ii}^t=\sum_{j=1}^{n}\boldsymbol{A}_{ij}^t$ 是第 t 个模态的度矩阵；$\boldsymbol{F}_{*k}$ 是当第 k 个锚点作为查询样本时 n 个样本对应的得分向量；$\boldsymbol{F}_{i*}$ 是第 i 个样本在 q 个锚点上对应的得分向量；$\boldsymbol{Y}_{*k}$ 表示 n 个样本和第 k 个锚点的标签相似性. 我们通过如下的式(2)[28]来迭代求解式(1)的损失函数，最终可以获得一个包含了排序信息的得分矩阵 $\boldsymbol{F}$.

$$\boldsymbol{F}(c_0+1)=\frac{1}{2}\sum_{t=1}^{m}(\boldsymbol{R}^t\boldsymbol{F}(c_0)+\boldsymbol{Y})\tag{2}$$

其中 c_0 为迭代次数，$\boldsymbol{R}^t=(\boldsymbol{D}^t)^{-0.5}\boldsymbol{A}^t(\boldsymbol{D}^t)^{-0.5}$. 如果直接计算 $\boldsymbol{R}^t$ 的话会造成 $O(n^2)$ 的时间复杂度和空间复杂度，不利于模型在大规模数据集上的训练，因此在这里我们利用矩阵的结合律来解决这个问题. 首先我们通过式(3)来计算度矩阵 $\boldsymbol{D}^t$ 并采用稀疏存储方式进行存储，然后我们可以通过式(4)来计算 $\boldsymbol{R}^t\boldsymbol{F}(c_0)$.

$$\begin{aligned}\boldsymbol{D}_{ii}^t&=\sum_{j=1}^{n}\boldsymbol{A}_{ij}^t\\&=\sum_{j=1}^{n}\boldsymbol{z}_{i*}^t\boldsymbol{\Omega}^{-1}\boldsymbol{z}_{j*}^{t\top}\\&=\boldsymbol{z}_{i*}^t\boldsymbol{\Omega}^{-1}\sum_{j=1}^{n}\boldsymbol{z}_{j*}^{t\top}\\&\text{s.t. }\boldsymbol{\Omega}=diag(\boldsymbol{Z}^{t\top}\boldsymbol{1})\in\mathbb{R}^{q\times q}\end{aligned}\tag{3}$$

其中 $\boldsymbol{z}_{i*}^t$ 和 $\boldsymbol{z}_{j*}^t$ 分别是 $\boldsymbol{Z}^t$ 的第 i 行和第 j 行.

$$\boldsymbol{R}^t\boldsymbol{F}(c_0)=\boldsymbol{D}^{t^{-0.5}}\boldsymbol{Z}^t\boldsymbol{\Omega}^{-1}((\boldsymbol{D}^{t^{-0.5}}\boldsymbol{Z}^t)^{\top}\boldsymbol{F}(c_0))\tag{4}$$

方便起见我们将学得的 $\boldsymbol{F}$ 归一化到[0,1]并将归一化后的得分矩阵表示为 $\bar{\boldsymbol{F}}$，利用它来学习有效的哈希码和哈希函数.

3.2.2 哈希学习

相似性保持项. 为了充分地利用 $\bar{\boldsymbol{F}}$ 中所包含的有效信息，我们设计了两种相似性保持策略来学习高质量的哈希码. 首先如式(5)所示，我们希望第 i 个样本和第 k 个锚点的海明距离应该与它们之间的排序信息保持一致. 如果第 i 个样本在第 k 个锚点上的得分越高，那么在海明空间中，它们之间的海明距离应越小，即它们的哈希码的内积应尽可能大，反之亦然.

$$\begin{aligned}&\min_{\boldsymbol{B},\boldsymbol{B}^q}\sum_{i=1}^{n}\sum_{k=1}^{q}(r\bar{\boldsymbol{F}}_{i,k}-\boldsymbol{B}_{i*}\boldsymbol{B}_{k*}^{q\top})^2\\&\text{s.t. }\boldsymbol{B}\in\{-1,1\}^{n\times r}\end{aligned}\tag{5}$$

哈希码拟合项. 其次根据 $\bar{\boldsymbol{F}}$ 中所包含的相对相似信息，我们也可以通过 q 个锚点的哈希码加权求和来拟合所有训练样本的哈希码，进一步加强相似性关系的映射. 如式(6)所示，$\boldsymbol{B}_{i,s}$ 表示第 i 个样本的第 s 位哈希位，$\bar{\boldsymbol{F}}_{i*}$ 表示 $\bar{\boldsymbol{F}}$ 的第 i 行，可以看作第 i 个样本和 q 个锚点的相似度向量，$\boldsymbol{B}_{*s}^q$ 表示 $\boldsymbol{B}^q$ 的第 s 列，代表 q 个锚点的第 s 位哈希位. $\bar{\boldsymbol{F}}$，$\boldsymbol{B}^q$ 两个矩阵相乘，可以看做是 q 个锚点对应的哈希码通过相似度加权求和，来拟合所有样本的哈希码.

$$\min_{\boldsymbol{B},\boldsymbol{B}^q}\sum_{i=1}^{n}\sum_{s=1}^{r}(\boldsymbol{B}_{i,s}-\overline{\boldsymbol{F}}_{i*}\boldsymbol{B}_{*s}^{q})^2 \tag{6}$$

$$\text{s.t. } \boldsymbol{B}\in\{-1,1\}^{n\times r}$$

哈希函数项. 为了解决跨模态检索中多模态数据的异构性问题，我们通过 m 个不同的线性映射函数将不同模态的数据映射为语义统一的哈希码. 考虑到学到的哈希码可能会有位不平衡和位冗余的问题，我们引入独立平衡约束，使得学到的哈希码的每一位都尽可能包含不同的信息.

$$\min_{\boldsymbol{B},\boldsymbol{W}^t}\sum_{t=1}^{m}\lambda_t\|\boldsymbol{B}-\boldsymbol{X}^t\boldsymbol{W}^t\|_F^2 \tag{7}$$

$$\text{s.t. } \boldsymbol{B}\in\{-1,1\}^{n\times r},\ \mathbf{1}^{\top}\boldsymbol{B}=\mathbf{0},\ \boldsymbol{B}^{\top}\boldsymbol{B}=n\boldsymbol{I}_r$$

3.2.3　哈希损失函数

将式(5)、式(6)及式(7)结合，我们可以得到完整的哈希损失函数如式(8)所示，同时学习哈希码和哈希函数. 为了避免过拟合，我们还引入了一个正则化项.

$$\min_{\boldsymbol{B},\boldsymbol{W}^t,\boldsymbol{B}^q}\alpha_1\|r\overline{\boldsymbol{F}}-\boldsymbol{B}\boldsymbol{B}^{q\top}\|_F^2+\alpha_2\|\boldsymbol{B}-\overline{\boldsymbol{F}}\boldsymbol{B}^q\|_F^2+\sum_{t=1}^{m}\lambda_t\|\boldsymbol{B}-\boldsymbol{X}^t\boldsymbol{W}^t\|_F^2+\mu\|\boldsymbol{W}^t\|_F^2 \tag{8}$$

$$\text{s.t. } \boldsymbol{B}\in\{-1,1\}^{n\times r},\mathbf{1}^{\top}\boldsymbol{B}=\mathbf{0},\boldsymbol{B}^{\top}\boldsymbol{B}=n\boldsymbol{I}_r$$

3.3　优化算法

考虑到哈希码的离散性及独立平衡约束，为了降低优化的复杂性，我们引入了两个松弛矩阵 $\boldsymbol{T}$、$\boldsymbol{T}^q$ 分别作为 $\boldsymbol{B}$、$\boldsymbol{B}^q$ 的实值近似矩阵. 最终我们可以获得以下的损失函数：

$$\min_{\boldsymbol{B},\boldsymbol{W}^t,\boldsymbol{T},\boldsymbol{T}^q}\alpha_1\|r\overline{\boldsymbol{F}}-\boldsymbol{T}\boldsymbol{T}^{q\top}\|_F^2+\alpha_2\|\boldsymbol{B}-\overline{\boldsymbol{F}}\boldsymbol{T}^q\|_F^2+\sum_{t=1}^{m}\lambda_t\|\boldsymbol{B}-\boldsymbol{X}^t\boldsymbol{W}^t\|_F^2+\|\boldsymbol{B}-\boldsymbol{T}\|_F^2+\mu\|\boldsymbol{W}^t\|_F^2 \tag{9}$$

$$\text{s.t. } \boldsymbol{B}\in\{-1,1\}^{n\times r},\mathbf{1}^{\top}\boldsymbol{T}=\mathbf{0},\boldsymbol{T}^{\top}\boldsymbol{T}=n\boldsymbol{I}_r$$

目标函数式(9)是一个多变量优化问题，如果同时优化所有变量，这个问题很显然是一个 NP 难问题. 针对这一问题，我们设计了一种交替优化的策略，迭代地更新各个变量. 具体来说，我们每次选取一个变量进行更新，固定其余的几个变量. 具体的优化算法如下所示：

(1) 固定 $\boldsymbol{B},\boldsymbol{W}^t,\boldsymbol{T}$，优化 $\boldsymbol{T}^q$. 关于 $\boldsymbol{T}^q$ 的目标函数可以表示为

$$\min_{\boldsymbol{T}^q}\alpha_1\|r\overline{\boldsymbol{F}}-\boldsymbol{T}\boldsymbol{T}^{q\top}\|_F^2+\alpha_2\|\boldsymbol{B}-\overline{\boldsymbol{F}}\boldsymbol{T}^q\|_F^2 \tag{10}$$

式(10)对 $\boldsymbol{T}^q$ 求偏导，并将偏导置零，可以得到 $\boldsymbol{T}^q$ 的闭式解如下：

$$\boldsymbol{T}^q=\left(n\boldsymbol{I}_q+\frac{\alpha_2}{\alpha_1}\overline{\boldsymbol{F}}^{\top}\overline{\boldsymbol{F}}\right)^{-1}\left(r\overline{\boldsymbol{F}}^{\top}\boldsymbol{T}+\frac{\alpha_2}{\alpha_1}\overline{\boldsymbol{F}}^{\top}\boldsymbol{B}\right) \tag{11}$$

(2) 固定 $\boldsymbol{T},\boldsymbol{W}^t,\boldsymbol{T}^q$，优化 $\boldsymbol{B}$. 关于 $\boldsymbol{B}$ 的目标函数可以表示为

$$\min_{\boldsymbol{B}}\alpha_2\|\boldsymbol{B}-\overline{\boldsymbol{F}}\boldsymbol{T}^q\|_F^2+\sum_{t=1}^{m}\lambda_t\|\boldsymbol{B}-\boldsymbol{X}^t\boldsymbol{W}^t\|_F^2+\|\boldsymbol{B}-\boldsymbol{T}\|_F^2 \tag{12}$$

$$\text{s.t. } \boldsymbol{B}\in\{-1,1\}^{n\times r}$$

去掉一些常数项和与变量 $\boldsymbol{B}$ 无关的项，以上公式可以化简为

$$\max_{\boldsymbol{B}}tr(\boldsymbol{B}^{\top}\boldsymbol{Q}) \tag{13}$$

其中 $\boldsymbol{Q}=\sum_{t=1}^{m}\lambda_t\boldsymbol{X}^t\boldsymbol{W}^t+\alpha_2\overline{\boldsymbol{F}}\boldsymbol{T}^q+\boldsymbol{T}$.

显而易见，我们可以得到 $\boldsymbol{B}$ 的闭式解为

$$\boldsymbol{B}=sgn(\boldsymbol{Q}) \tag{14}$$

(3) 固定 $\boldsymbol{T},\boldsymbol{B},\boldsymbol{T}^q$，优化 $\boldsymbol{W}^t$. 关于 $\boldsymbol{W}^t$ 的目标函数可以表示为

$$\min_{\boldsymbol{W}^t}\|\boldsymbol{B}-\boldsymbol{X}^t\boldsymbol{W}^t\|_F^2 \tag{15}$$

式(15)对 $\boldsymbol{W}^t$ 求偏导，并将偏导置零，可以得到 $\boldsymbol{W}^t$ 的闭式解如下：

$$\boldsymbol{W}^t=(\boldsymbol{X}^{t\top}\boldsymbol{X}^t)^{-1}\boldsymbol{X}^{t\top}\boldsymbol{B} \tag{16}$$

(4) 相似地我们可以得到关于 $\boldsymbol{T}$ 的损失函数为

$$\min_{\boldsymbol{T}}\alpha_1\|r\overline{\boldsymbol{F}}-\boldsymbol{T}\boldsymbol{T}^{q\top}\|_F^2+\|\boldsymbol{B}-\boldsymbol{T}\|_F^2 \quad \text{s.t. } \mathbf{1}^{\top}\boldsymbol{T}=\mathbf{0},\boldsymbol{T}^{\top}\boldsymbol{T}=n\boldsymbol{I}_r \tag{17}$$

通过化简我们可以得到以下公式：

$$\max_{\boldsymbol{T}}tr(\boldsymbol{P}^{\top}\boldsymbol{T}) \text{ s.t. } \mathbf{1}^{\top}\boldsymbol{T}=\mathbf{0},\boldsymbol{T}^{\top}\boldsymbol{T}=n\boldsymbol{I}_r \tag{18}$$

其中 $\boldsymbol{P}=r\overline{\boldsymbol{F}}\boldsymbol{T}^q+\boldsymbol{B}$.

由于 $\boldsymbol{T}$ 的约束，我们不能通过将损失函数求导置零直接获得它的最优解. 为了解决这个问题，我们首先定义一个中心化矩阵 $\boldsymbol{M}=\boldsymbol{I}_n-\frac{1}{n}\mathbf{1}\mathbf{1}^{\top}$，然后对 $\boldsymbol{M}\boldsymbol{P}$ 进行奇异值分解(SVD)，可以得到 $\boldsymbol{M}\boldsymbol{P}=\boldsymbol{C}\sum\boldsymbol{J}^{\top}=\sum_{k=1}^{r'}\sigma_k\boldsymbol{c}_k\boldsymbol{j}_k^{\top}$. 其中 $r'\leq r$ 是 $\boldsymbol{M}\boldsymbol{P}$ 的秩，$\sigma_1,\cdots,\sigma_{r'}$ 是正奇异值序列，$\boldsymbol{C}=[\boldsymbol{c}_1,\cdots,\boldsymbol{c}_{r'}]$ 和 $\boldsymbol{J}=[\boldsymbol{j}_1,\cdots,\boldsymbol{j}_{r'}]$ 分别是对应的左奇异向量和右奇异向量. 通过施密特正交化过程(Gram-Schmidt process)，我们可以获得 $\overline{\boldsymbol{C}}\in\mathbb{R}^{n\times(r-r')}$ 和 $\overline{\boldsymbol{J}}\in\mathbb{R}^{r\times(r-r')}$，并且满足 $\overline{\boldsymbol{C}}^{\top}\overline{\boldsymbol{C}}=\boldsymbol{I}_{r-r'}$，$[\boldsymbol{C}\mathbf{1}]^{\top}\overline{\boldsymbol{C}}=\mathbf{0}$，$\overline{\boldsymbol{J}}^{\top}\overline{\boldsymbol{J}}=\boldsymbol{I}_{r-r'}$，$\overline{\boldsymbol{J}}^{\top}\overline{\boldsymbol{J}}=\mathbf{0}^2$ 的条件[30]. 最终我们可以得到如下所示的闭式解：

$$\boldsymbol{T}=\sqrt{n}[\boldsymbol{C}\overline{\boldsymbol{C}}][\boldsymbol{J}\overline{\boldsymbol{J}}]^{\top} \tag{19}$$

我们在算法 1 中也给出了具体的优化流程.

算法 1. RSDCH 优化算法.

输入：训练集 $\boldsymbol{X}^t, t\in 1,\cdots,m$；得分矩阵 $\overline{\boldsymbol{F}}$；超参数 α_1，α_2，λ_t，μ，q；哈希码长度 r；迭代次数 c

输出：哈希映射矩阵 $\boldsymbol{W}^t$，哈希码矩阵 $\boldsymbol{B}$

1. 随机初始化 $\boldsymbol{W}^t, \boldsymbol{B}\in\{-1,1\}^{n\times r}, \boldsymbol{T}, \boldsymbol{T}^q$.
2. FOR $i=1$ to c DO
3. 根据式(11)更新 $\boldsymbol{T}^q$；
4. 根据式(14)更新 $\boldsymbol{B}$；
5. 根据式(16)更新 $\boldsymbol{W}^t$；
6. 根据式(19)更新 $\boldsymbol{T}$；
7. END FOR
8. RETURN $\boldsymbol{W}^t, \boldsymbol{B}$.

3.4 时间复杂度分析

我们所提出的方法总的时间复杂度为 $O((2d^{t^2}+q^2c+4qrc+md^trc+md^tc+r^2c)n+d^{t^3}+q^3c+q^2rc)$，其中 n 远远大于其他的超参数，计算 $\boldsymbol{T}^q$ 的时间复杂度为 $O((q^2+2qr)n+q^3+q^2r)$；计算 $\boldsymbol{B}$ 的时间复杂度为 $O((md^tr+qr)n)$；计算式(16)的时间复杂度为 $O((2d^{t^2}+d^tr)n+d^{t^3})$，其中 $(\boldsymbol{X}^{t\top}\boldsymbol{X}^t)^{-1}\boldsymbol{X}^{t\top}$ 可以在训练开始前计算；最后 $O(r^2n)$ 为计算式(19)时奇异值分解和施密特正交化对应的时间复杂度，$O(qrn)$ 为计算 $\boldsymbol{P}$ 所需的时间复杂度. 总体而言，我们提出方法的时间复杂度为线性，可以扩展到大规模数据集.

3.5 样本外扩展

通过以上的哈希学习算法我们可以为训练数据学习到高质量的哈希码，同时学到的哈希函数可以将训练样本外任意的查询样本映射到海明空间. 给定一个查询样本 $\boldsymbol{x}_q^t$ 我们可以通过以下公式生成它对应的哈希码：

$$\boldsymbol{b}_q=sgn(\boldsymbol{x}_q^t\boldsymbol{W}^t) \tag{20}$$

其中 $\boldsymbol{W}^t$ 是第 t 个模态的哈希映射矩阵.

4 实验与分析

在本节中，我们在 MIRFlickr-25K[31] 及 NUS-WIDE[32] 这两个通用数据集上进行对比实验来验证本文提出方法的有效性. 此外我们还通过详细的消融实验测试了损失函数中各个模块的有效性，通过参数敏感性实验测试了模型的稳定性.

4.1 数据集

MIRFlickr-25K. 该数据集包含 25 000 张图像，每张图像都有与之对应的文本信息，并且这 25 000 个样本都被 24 个语义标签中的一个或多个标签所标注. 我们遵循文献[23]中的设置，选取数据集中被不少于 20 个标签标记的样本用于实验. 经过数据预处理，数据集包含有 20 015 个图像-文本对. 对于图像，我们采用 512 维的 GIST 特征来表示，对于文本，我们采用 1386 维的 BOW 特征进行表示. 我们随机选取 2000 个样本作为测试集，剩余的 18 015 个样本作为训练集和待检索数据集.

NUS-WIDE. 该数据集包含 269 648 张图像，每张图像都有与之对应的文本信息，并且这些样本都被 81 个语义标签中的一个或多个标签所标注. 我们遵循文献[23]中的设置，选择数据集中被不少于 21 个标签标记的样本用于实验. 经过数据预处理，数据集包含有 195 834 个图像-文本对. 对于图像，我们采用 500 维的 SIFT 特征来表示，对于文本，我们采用 1000 维的 BOW 特征来进行描述. 我们随机选取 2000 个样本作为测试集，剩余的 193 834 个样本作为训练集和待检索数据集.

4.2 对比方法

在实验部分，我们与几种当前最好的跨模态哈希方法进行对比实验，包括 CCQ[10]、FSH[5]、CRE[11]、SePH-km[6]、DCH[12]、SCRATCH[13] 以及 LCMFH[14]. 其中 CCQ、FSH、CRE 为无监督跨模态哈希方法，SePH-km、DCH、SCRATCH 以及 LCMFH 为有监督的跨模态哈希方法. 除 LCMFH 为我们自己复现代码外，其余对比方法的代码均为作者公开的源代码. 由于 SePH-km 和 FSH 的计算复杂度较高，训练成本太大，我们遵循文献[13]中的策略，在 NUS-WIDE 数据集上从训练集中随机采样 5000 个样本作为它们的训练集. 实验服务器配置为 Intel XEON E5-2650 2.30 GHz CPU，128 GB RAM，64 位 Linux 操作系统.

4.3 评价标准

我们采用 3 种广泛使用的评价指标，包括平均精确率均值(MAP)、归一化折损累计增益(NDCG)、精确率-召回率曲线(Precision-Recall Curve)来评价所有方法的检索性能. MAP 往往用来评价算法整体的检索性能，MAP 值越高证明检索结果的平均精确率越高. 精确率-召回率曲线(Precision-Recall Curve)则展示了在检索到的样本列表中随着召回率的增大，精确率的变化趋势. 一般来说，召回率越大，对应的精确率越低. NDCG 更关注高相关性的样本有没有被返回，如果高相关性的样本返回的位置越靠前，则对应的 NDCG 得分越高，代表检索效果越好. 归一化折损累计增益(NDCG)的定义如下：

$$\mathrm{NDCG@}p=\frac{\mathrm{DCG@}p}{\mathrm{IDCG@}p}=\frac{\sum_{i=1}^{p}\frac{2^{rel_i}-1}{\log_2(i+1)}}{\sum_{i=1}^{|REL|}\frac{2^{rel_i}-1}{\log_2(i+1)}} \quad (21)$$

其中 DCG(折损累计增益)是样本经过哈希映射后的 DCG 结果，IDCG 是理想状态下最大的 DCG 结果，p 表示取返回的检索列表中的前 p 个结果. 在实验中，我们设置 $p=100$，即对于每一个查询样本，我们计算检索到的前 100 个样本的 NDCG. $|REL|$ 表示按照与查询样本的相关性降序排列的待检索样本的集合. rel_i 代表排在第 i 个位置的样本与查询样本的相关性. 我们通过共享的标签数量来定义两个样本之间的相关性，共享的标签越多，代表两个样本之间的相关性得分越高. 对于所有的评价标准来说，数值越高代表检索效果越好.

4.4　参数设置

公平起见，对于所有的对比方法，我们采用作者原论文中所建议的参数设置. 对于本文所提出的方法，模型各个参数的设置如下：总的迭代次数 $c=10$，平衡参数 $\alpha_1=\alpha_2=1$，两个模态的权重参数 $\lambda_1=\lambda_2=0.5$，正则化项参数 $\mu=10$. 所有的参数设置都是基于交叉验证得到的结果. 在后面的实验部分我们会进一步分析模型对于各参数变化的敏感度.

4.5　结果与分析

4.5.1　MIRFlickr-25K 数据集上的实验结果

表 1 给出了所有方法在 MIRFlickr-25K 数据集上哈希码长度为 16 位、32 位、64 位及 128 位时，图像检索文本及文本检索图像这两个跨模态检索任务的 MAP 和 NDCG@100 结果. 图 1 给出了所有方法在 MIRFlickr-25K 数据集上哈希码长度为 64 位及 128 位时的精确率-召回率曲线. 观察图表中数据，我们可以得出以下几个结论：

(1) 从表 1 结果来看，RSDCH 在两个跨模态检索任务上的效果明显好于其他对比方法. 整体来说，在图像检索文本任务上，RSDCH 的 MAP 比效果最好的对比方法 SCRATCH 平均提升了 1.8%，NDCG@100 比效果最好的对比方法 LCMFH 平均提升了 2.8%. 在文本检索图像任务上，RSDCH 的 MAP 比效果最好的对比方法 SCRATCH 平均提升了 4.0%，NDCG@100 比效果最好的对比方法 LCMFH 平均提升了 0.5%.

(2) SCRATCH 等现有方法，与本文提出的方法一样，均是基于离散优化算法学习哈希码. 通过与这类方法对比，我们可以发现本文的方法可以有更好的 MAP 和 NDCG@100 结果. 这样的实验结果说明了我们的模型可以生成更高质量的哈希码. 具体来说，本文方法在预训练阶段获得的得分矩阵包含了丰富的具有判别力的信息，并且通过精心设计的学习策略，使得哈希码能充分地保持有效信息.

(3) 观察图 1 中所有方法的精确率-召回率曲线，我们设计的方法依然取得了最优的结果，进一步验证了基于排序的哈希学习和离散优化的重要性.

(4) 整体来看，监督哈希方法要比无监督哈希方法的效果好，说明了利用监督信息的重要性.

表 1　MIRFlickr-25K 数据集 MAP 和 NDCG@100 结果对比

检索任务	对比方法	MAP				NDCG@100			
		16 位	32 位	64 位	128 位	16 位	32 位	64 位	128 位
图像检索文本	CCQ	0.5718	0.5751	0.5758	0.5763	0.3773	0.3770	0.3795	0.3876
	SePH-km	0.6817	0.6857	0.6879	0.6886	0.4166	0.4356	0.4440	0.4481
	FSH	0.5935	0.6133	0.6173	0.6247	0.3810	0.4002	0.4110	0.4172
	DCH	0.6685	0.6736	0.6924	0.7038	0.3924	0.4342	0.4462	0.4657
	SCRATCH	0.7142	0.7161	0.7250	0.7336	0.4727	0.4724	0.4939	0.5112
	CRE	0.6197	0.6243	0.6218	0.6285	0.2117	0.2151	0.2215	0.2257
	LCMFH	0.6945	0.6992	0.7070	0.7084	0.4838	0.4968	0.5178	0.5152
	RSDCH	**0.7310**	**0.7359**	**0.7449**	**0.7479**	**0.5258**	**0.5165**	**0.5375**	**0.5463**
文本检索图像	CCQ	0.5787	0.5808	0.5821	0.5820	0.3422	0.3444	0.3470	0.3454
	SePH-km	0.7245	0.7301	0.7332	0.7343	0.4757	0.4988	0.5104	0.5178
	FSH	0.5897	0.6098	0.6143	0.6240	0.3819	0.4078	0.4304	0.4467
	DCH	0.7395	0.7483	0.7695	0.7876	0.4857	0.5304	0.5398	0.5452
	SCRATCH	0.7692	0.7750	0.7884	0.8003	0.5405	0.5237	0.5495	0.5581
	CRE	0.6276	0.6305	0.6304	0.6369	0.2353	0.2453	0.2680	0.2733
	LCMFH	0.7420	0.7587	0.7737	0.7754	0.5518	0.5588	0.5719	0.5764
	RSDCH	**0.8069**	**0.8197**	**0.8317**	**0.8340**	**0.5591**	**0.5652**	**0.5740**	**0.5823**

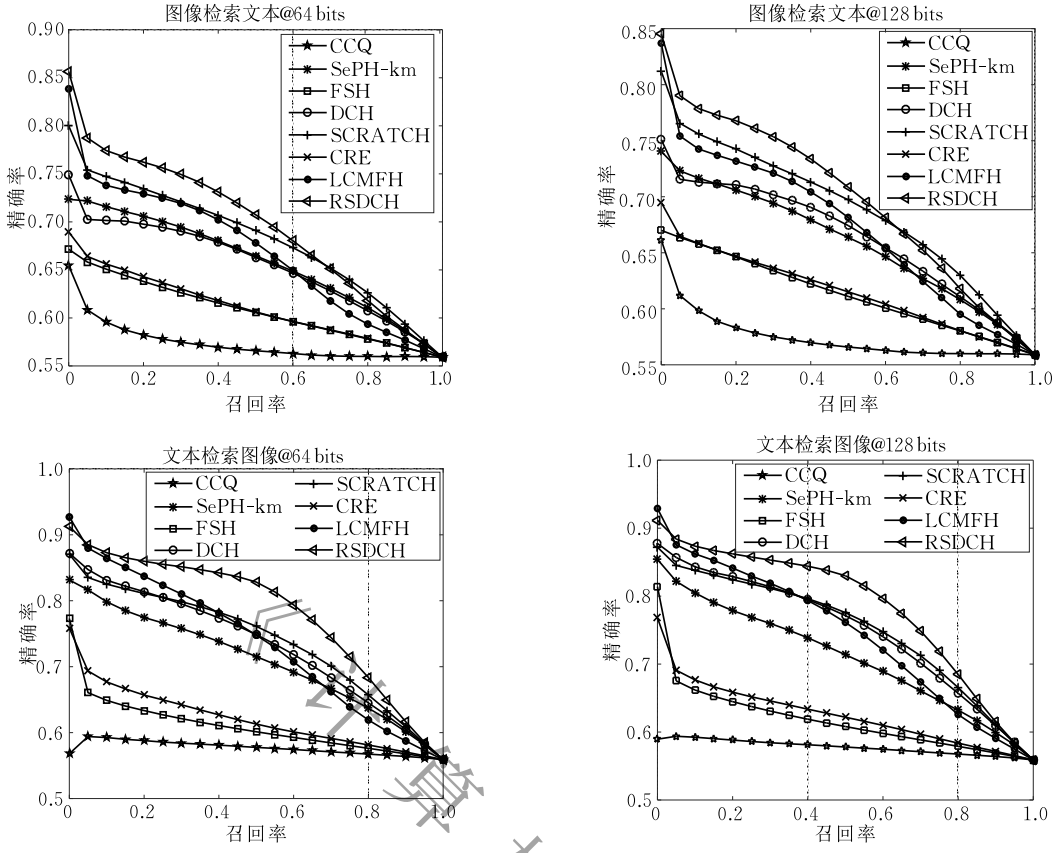

图 1 MIRFlickr-25K 数据集上不同编码长度下对应的精确率-召回率曲线

4.5.2 NUS-WIDE 数据集上的实验结果

表 2 给出了所有方法在 NUS-WIDE 数据集上不同的哈希码长度对应的 MAP 和 NDCG@100 结果. 图 2 给出了所有方法在 NUS-WIDE 数据集上的精确率-召回率曲线. 我们可以从图表中数据得出以下结论:

(1) 在不同的哈希码长度设置下,本文提出的方法 RSDCH 在 MAP 和 NDCG@100 评价指标下均能取得最好的效果. 在图像检索文本任务上,RSDCH 的 MAP 比效果最好的对比方法 SCRATCH 平均提升了 0.8%,NDCG@100 比效果最好的对比方法 SCRATCH 平均提升了 2.3%. 在文本检索图像任务上,RSDCH 的 MAP 比效果最好的对比方法 SCRATCH 平均提升了 1.6%,NDCG@100 比效果

表 2 NUS-WIDE 数据集 MAP 和 NDCG@100 结果对比

检索任务	对比方法	MAP				NDCG@100			
		16 位	32 位	64 位	128 位	16 位	32 位	64 位	128 位
图像检索文本	CCQ	0.3745	0.3792	0.3848	0.3839	0.3255	0.3427	0.3581	0.3510
	SePH-km	0.5074	0.5144	0.5160	0.5183	0.3857	0.3956	0.4035	0.4128
	FSH	0.3352	0.3426	0.3459	0.3530	0.2811	0.2943	0.3045	0.3216
	DCH	0.5463	0.5813	0.5885	0.5900	0.4122	0.4966	0.4825	0.5122
	SCRATCH	0.6020	0.6051	0.6150	0.6202	0.4765	0.5038	0.5035	0.5224
	CRE	0.4988	0.5034	0.5074	0.5123	0.4062	0.4131	0.4240	0.4323
	LCMFH	0.5670	0.5862	0.5977	0.6114	0.4633	0.4701	0.5076	0.5224
	RSDCH	**0.6126**	**0.6138**	**0.6201**	**0.6280**	**0.5070**	**0.5166**	**0.5294**	**0.5443**
文本检索图像	CCQ	0.3558	0.3557	0.3582	0.3581	0.3177	0.3291	0.3385	0.3329
	SePH-km	0.6071	0.6185	0.6231	0.6246	0.5152	0.5364	0.5491	0.5593
	FSH	0.3346	0.3428	0.3473	0.3537	0.2807	0.2919	0.3080	0.3218
	DCH	0.6978	0.7212	0.7315	0.7407	0.6028	0.6634	0.6563	0.6629
	SCRATCH	0.7291	0.7439	0.7549	0.7656	0.5903	0.6495	0.6469	0.6653
	CRE	0.5329	0.5374	0.5418	0.5476	0.4628	0.4897	0.5025	0.5189
	LCMFH	0.6815	0.6855	0.7010	0.7119	0.5965	0.6203	0.6499	0.6637
	RSDCH	**0.7468**	**0.7563**	**0.7710**	**0.7816**	**0.6523**	**0.6696**	**0.6763**	**0.6898**

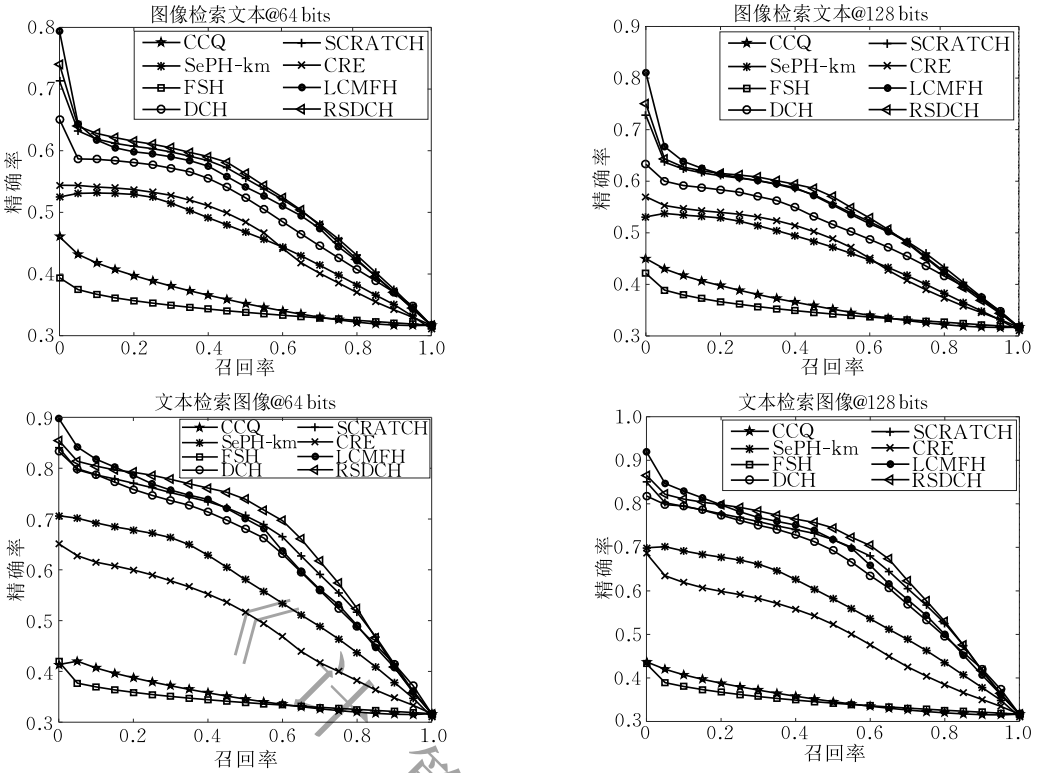

图 2　NUS-WIDE 数据集上不同编码长度下对应的精确率-召回率曲线

最好的对比方法 DCH 平均提升了 2.6%.这也证明了基于排序思想的哈希学习方法能够获取更有效的相似信息,进而在相似性检索任务中取得更好的效果.

(2)在 NUS-WIDE 数据集上,FSH 和 SePH-km 由于时间复杂度较高只能采样部分数据进行训练,检索效果明显欠佳,具有一定的局限性.相比较而言,本文所提出的方法通过引入锚点策略,可以在线性复杂度下使用全部训练数据来进行学习,在大规模数据上具有很好的可扩展性.

(3)从图 2 的精确率-召回率曲线来看,我们提出的 RSDCH 整体上要优于其他的对比方法,特别是在文本检索图像任务上效果更为明显.

4.6　消融实验

为了验证本文提出的哈希损失函数和排序损失函数各个模块的有效性,我们设计了相应的消融实验来进行测试.

4.6.1　对哈希损失函数模块消融的分析

首先我们针对哈希损失函数设计了 4 种变体,以测试其中各项对本文所提方法的影响,对应的实验结果如表 3 所示.RSDCH-1 表示参数 $\alpha_1=0$,即消去哈希损失函数中的第 1 项(基于锚点的相似性保持项);RSDCH-2 表示参数 $\alpha_2=0$,即消去哈希损失函数中的第 2 项(基于锚点的哈希码拟合项);

表 3　在 MIRFlickr-25K 和 NUS-WIDE 数据集上哈希损失函数消融实验结果对比

检索任务	对比方法	MAP				NDCG@100			
		MIRFlickr-25K		NUS-WIDE		MIRFlickr-25K		NUS-WIDE	
		32 位	64 位	32 位	64 位	32 位	64 位	32 位	64 位
图像检索文本	RSDCH-1	0.5641	0.5689	0.3372	0.3405	0.3381	0.3347	0.2918	0.3019
	RSDCH-2	0.6140	0.5736	0.4671	0.4686	0.4427	0.3622	0.4722	0.4808
	RSDCH-3	0.7289	0.7369	0.6127	0.6103	0.4822	0.4971	0.5164	0.5200
	RSDCH-4	0.7223	0.7265	0.5252	0.5245	0.5110	0.5373	0.4483	0.4554
	RSDCH	**0.7359**	**0.7449**	**0.6138**	**0.6201**	**0.5165**	**0.5375**	**0.5166**	**0.5294**
文本检索图像	RSDCH-1	0.5660	0.5704	0.3628	0.3684	0.3554	0.3666	0.3421	0.3592
	RSDCH-2	0.6415	0.5911	0.5865	0.5833	0.5039	0.4348	0.6327	0.6158
	RSDCH-3	0.8130	0.8252	0.7514	0.7588	0.5568	0.5619	0.6695	0.6761
	RSDCH-4	0.8042	0.8153	0.5983	0.6051	0.5624	0.5734	0.5908	0.6040
	RSDCH	**0.8197**	**0.8317**	**0.7563**	**0.7710**	**0.5652**	**0.5740**	**0.6696**	**0.6763**

RSDCH-3 表示参数 $\mu=0$，即消去哈希损失函数中的正则化项；RSDCH-4 表示去掉对哈希码的独立和平衡约束. 不难看出，去掉模型中的任意一项，实验结果都有不同程度的降低. 具体而言，去掉损失函数中的相似性保持项或者哈希码拟合项在两个数据集上的 MAP 和 NDCG@100 都有显著的降低，说明我们在预训练阶段学到的得分矩阵包含了丰富的语义信息，能够很好地指导哈希码的学习. 而去掉对哈希码的独立和平衡性的约束对 MIRFlickr-25K 数据集的影响较小，对 NUS-WIDE 数据集的影响较大，说明在大规模数据集上，适当地增强每一位哈希码的独立性，消除冗余，有利于学到更具有区分力的哈希码. 整体而言，损失函数中的各个模块都是不可或缺的.

4.6.2 对排序损失函数中 $\boldsymbol{Y}$ 的构建方式的分析

此外我们还对排序损失函数中采用不同的方式构建的标签相似性矩阵 $\boldsymbol{Y}$ 对 NDCG@100 的影响做了进一步的测试，结果如表 4 所示. 我们知道，每个样本都对应一个标签向量，我们通过计算两个样本的标签向量的相似性来定义标签相似性矩阵 $\boldsymbol{Y}$. RSDCH-Cx 表示采用余弦相似性对样本特征向量的相似性进行度量，是一个无监督的基础模型；RSDCH-J 表示采用 Jaccard 相似性度量两样本的标签相似性；RSDCH-L 表示直接定义标签相似性矩阵 $\boldsymbol{Y}=\boldsymbol{L}\boldsymbol{L}_q^{\top}$；RSDCH-CL 表示采用余弦相似性度量样本的标签相似性. 从表 4 中的结果来看，采用标签比不采用标签效果提升明显，而采用余弦相似性度量标签相似性在两个数据集上多数能取得最好的效果，因此，在实验中，我们采用 RSDCH-CL 的方式来构造 $\boldsymbol{Y}$.

表 4 在 MIRFlickr-25K 和 NUS-WIDE 数据集上不同 Y 的 NDCG@100 结果对比

检索任务	对比方法	NDCG@100			
		MIRFlickr-25K		NUS-WIDE	
		32 位	64 位	32 位	64 位
图像检索文本	RSDCH-Cx	0.3815	0.3863	0.3429	0.3475
	RSDCH-J	0.4967	0.5132	0.4903	0.4951
	RSDCH-L	0.5118	0.5313	0.5076	0.5164
	RSDCH-CL	**0.5165**	**0.5376**	**0.5167**	**0.5245**
文本检索图像	RSDCH-Cx	0.3921	0.4025	0.4226	0.4433
	RSDCH-J	0.5605	**0.5745**	0.6492	0.6658
	RSDCH-L	0.5594	0.5709	0.6642	0.6808
	RSDCH-CL	**0.5652**	0.5740	**0.6705**	**0.6817**

4.7 参数敏感性分析

为了进一步验证模型的稳定性，我们对模型中的各项参数做了敏感性测试. 图 3、图 4、图 5 和图 6 分别展示了随着参数取值的变化，对应的 MAP 和 NDCG@100 的变化曲线. “I-T”和“T-I”分别代表图像检索文本任务和文本检索图像任务.

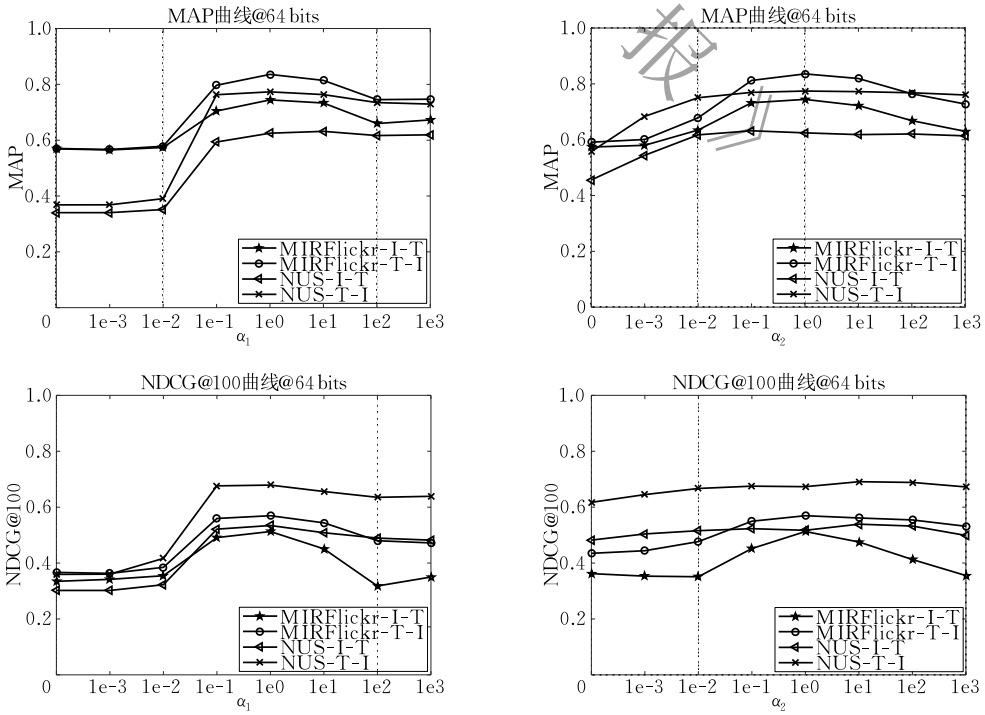

图 3 α_1 和 α_2 对 MAP 和 NDCG@100 的影响

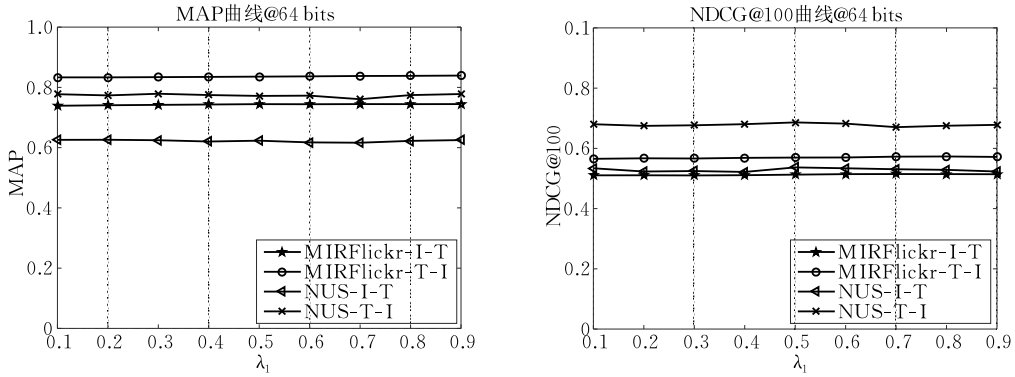

图 4　λ_1 对 MAP 和 NDCG@100 的影响

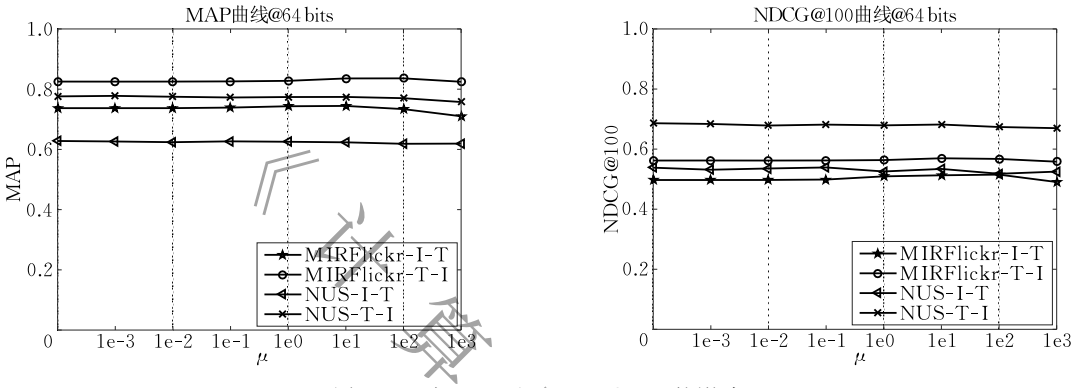

图 5　μ 对 MAP 和 NDCG@100 的影响

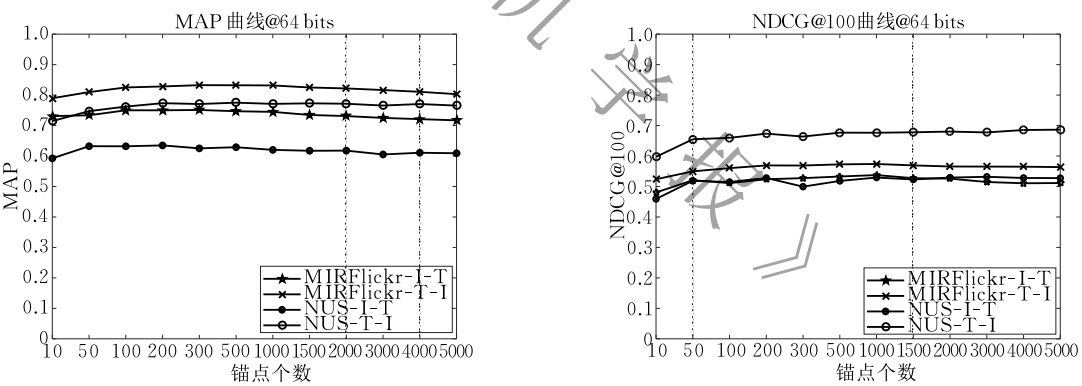

图 6　锚点个数对 MAP 和 NDCG@100 的影响

(1) α_1 代表相似性保持项的权重参数，α_2 为哈希码拟合项的权重参数. 从图 3 中结果来看，当 α_1 和 α_2 的变化范围为[1e-1，1e1]时，MAP 和 NDCG@100 结果曲线波动幅度较小. 我们可以发现这两个参数的取值同步变化时对应 MAP 曲线的变化趋势也是相同的，说明模型的相似性保持项和哈希码拟合项是相辅相成，同等重要的.

(2) λ_1 和 λ_2 分别为图像模态和文本模态的哈希函数项的权重参数. 由于我们设置 $\lambda_1+\lambda_2=1$，当 λ_1 确定时 λ_2 也随之确定，所以我们只给出了 λ_1 的取值对 MAP 和 NDCG@100 的影响. 从图 4 可以看到，当 λ_1 从 0.1 增大到 0.9 时，MAP 和 NDCG@100 的变化是极其微小的，说明我们设计的损失函数对于两个模态的权重变化是鲁棒的.

(3) 参数 μ 为模型的正则化项的权重参数. 从图 5 我们可以发现模型受参数 μ 的影响也是比较小的，不易被干扰.

(4) 除此之外，我们还测试了在实验中采样不同数目的锚点对实验结果的影响. 从图 6 中我们也可以看出，当锚点个数从 10 增加到 500 时，MAP 和 NDCG@100 整体呈现上升趋势，说明随着锚点数目的增加，模型能够获取更多的排序信息，因而实验性能有所提升. 当锚点个数从 500 增大到 5000 时，对应的 MAP 和 NDCG@100 变化是比较小的，说

明此时模型已经获取到了足够的排序信息,结果趋于稳定.在实验中我们把锚点个数设置为1000.总的来说,我们设计的模型对各项参数变化具有较强的鲁棒性,模型较为稳定.

4.8 模型收敛性分析

此外,我们还对模型收敛性进行了实验分析.图7给出了在64位哈希码长度下两个基准数据集上MAP和NDCG@100随着迭代次数的变化曲线.显而易见,在迭代两次后,模型的效果已经有显著的提升并且稳定在一定范围内,这表明我们的方法具有很好的收敛性.

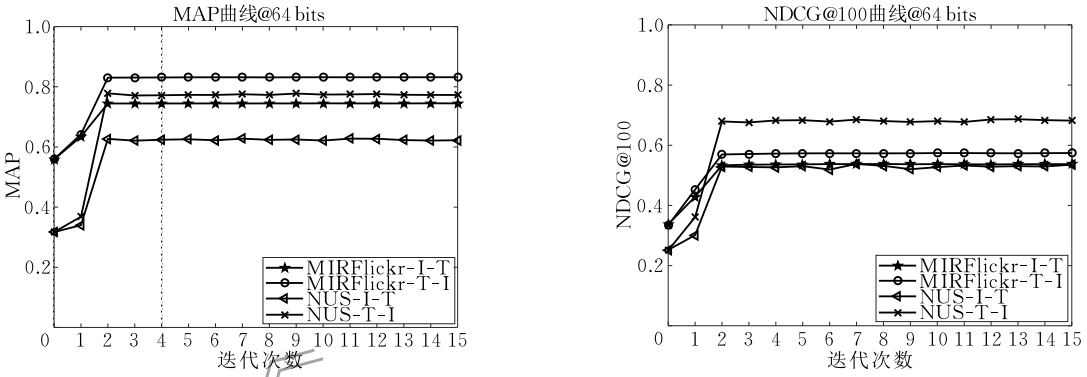

图7 模型收敛性分析

4.9 算法训练时间分析

在前文中我们已经分析过所提方法的时间复杂度,为了进一步验证其效率,本节给出关于训练时间的实验结果.表5给出了MIRFlickr-25K数据集上所有方法在不同的哈希码长度设置下所需的训练时间.其中RSDCH-pre表示RSDCH算法在预训练阶段计算排序得分矩阵所需的时间,RSDCH-hash表示RSDCH算法在哈希学习阶段所需的时间.RSDCH表示两阶段所需的总时间,即前两者时间之和.从表中我们可以看出,FSH和SePH-km的训练时间要远远地高出其他方法,这是因为这两种方法的时间复杂度较高,从而导致了较长的训练时间.与其他的哈希方法对比,我们所提出的方法RSDCH在哈希学习阶段的训练时间是较低的,而在预训练阶段学习排序得分矩阵时则需要花费较多的时间,从而导致总的训练时间较高.RSDCH的预训练时间与哈希码长度无关,换句话说,学习的一个排序得分矩阵可以去学习不同长度的哈希码,结合跨模态检索任务上良好的检索效果,我们可以在可接受的训练时间内达到最好的检索性能.

表5 在MIRFlickr-25K数据集上训练时间对比

(单位:s)

对比方法	MIRFlickr-25K				
	8位	16位	32位	64位	128位
CCQ	3.1	5.0	9.9	23.8	83.4
SePH-km	2807.7	2739.7	2802.2	2997.0	3330.4
FSH	111.0	118.5	118.0	116.7	120.1
DCH	2.1	2.4	3.0	6.7	25.8
SCRATCH	1.6	1.6	1.9	2.4	3.5
CRE	8.9	9.0	11.9	12.2	19.2
LCMFH	4.2	4.4	4.5	5.0	5.7
RSDCH-pre	54.2	—	—	—	—
RSDCH-hash	3.3	3.4	3.9	3.8	4.5
RSDCH	57.5	57.6	58.1	58.0	58.7

4.10 与深度方法的对比

深度哈希方法利用深度网络来提取数据的有效特征表示,往往能比传统的哈希方法取得更好的效果.为了更进一步验证本文所提方法检索性能的优越性,我们选取了三种新的深度哈希方法ADAH[23]、SSAH[24]以及RDCMH[25]进行对比实验,这三种深度哈希方法的MAP结果均取自它们各自的原始论文.在训练所有的非深度哈希方法时,我们采用在ImageNet数据集上预训练CNN-F[33]网络提取出的4096维的图像特征以及1386维的文本特征.表6列出了MIRFlickr-25K数据集上所有方法在哈希码为16位、32位及64位上的MAP结果.从表中结果我们可以看出,本文所提出的方法优于所有的非深度哈希方法,并且与三种深度方法对比,只在哈希码为16位时文本检索图像任务上的MAP与效果最好的对比方法RDCMH有0.02%的差距,其他位数上均具有明显优势.与同样是基于排序思想的深度哈希方法RDCMH相比,本文提出的RSDCH算法在图像检索文本任务上平均提升了约4.2%,文本检索图像任务上平均提升了约1.5%,这也更加验证了我们所设计的模型的优越性.

表 6　MIRFlickr-25K 数据集深度特征 MAP 结果对比

检索任务	对比方法	MAP		
		16 位	32 位	64 位
图像检索文本	CCQ	0.6261	0.6388	0.6423
	SePH-km	0.7796	0.7839	0.7865
	FSH	0.6038	0.6374	0.6506
	DCH	0.7395	0.7526	0.7561
	SCRATCH	0.7972	0.8090	0.8205
	CRE	0.7067	0.7120	0.7103
	LCMFH	0.7785	0.7913	0.7915
	ADAH	0.7922	0.8062	0.8074
	SSAH	0.7820	0.7900	0.8000
	RDCMH	0.7723	0.7735	0.7789
	RSDCH	**0.7977**	**0.8208**	**0.8309**
文本检索图像	CCQ	0.6232	0.6266	0.6283
	SePH-km	0.7538	0.7577	0.7601
	FSH	0.6035	0.6364	0.6477
	DCH	0.7527	0.7645	0.7646
	SCRATCH	0.7695	0.7745	0.7842
	CRE	0.6918	0.6920	0.6945
	LCMFH	0.7787	0.7903	0.7907
	ADAH	0.7563	0.7719	0.7720
	SSAH	0.7910	0.7950	0.8030
	RDCMH	**0.7931**	0.7924	0.8001
	RSDCH	0.7929	**0.8128**	**0.8236**

5　总　结

本文提出了一种新的监督跨模态哈希方法，叫做基于排序的监督离散跨模态哈希. 方法在预训练阶段为多个模态学习到一个统一的排序得分矩阵，并将其作为监督信息训练哈希模型，同时学习哈希码和哈希函数. 为了学到高质量的哈希码，我们设计了基于锚点策略的相似性保持项和哈希码拟合项，避免了过高的时间复杂度，并且提出了一种离散的交替优化策略来优化模型，使得获得的哈希码更具有判别能力. 我们在 MIRFlickr-25K 和 NUS-WIDE 两个公开数据集上与当前最好的几种无监督跨模态哈希方法和有监督跨模态哈希方法进行了对比实验，验证了本文所提方法的有效性，并且通过大量的消融实验和参数实验测试了模型的有效性和稳定性.

接下来我们考虑将目前的模型和深度网络相结合，构造一个端到端的框架，借助深度网络强大的特征表征能力来学习富含多层语义信息的排序信息，将特征学习与哈希学习相结合，训练更有效，检索性能更好的深度模型.

参　考　文　献

[1] Luo X, Zhang P, Huang Z, et al. Discrete hashing with multiple supervision. IEEE Transactions on Image Processing, 2019, 28(6): 2962-2975

[2] Li Z, Tang J, Zhang L, et al. Weakly-supervised semantic guided hashing for social image retrieval. International Journal of Computer Vision, 2020, 128(8): 2265-2278

[3] Cui H, Zhu L, Li J, et al. Scalable deep hashing for large scale social image retrieval. IEEE Transactions on Image Processing, 2020, 29: 1271-1284

[4] He S, Wang B, Wang Z, et al. Bidirectional discrete matrix factorization hashing for image search. IEEE Transactions on Cybernetics, 2020, 50(9): 4157-4168

[5] Liu H, Ji R, Wu Y, et al. Cross-modality binary code learning via fusion similarity hashing//Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition. Honolulu, USA, 2017: 6345-6353

[6] Lin Z, Ding G, Hu M, et al. Semantics-preserving hashing for cross-view retrieval//Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition. Boston, USA, 2015: 3864-3872

[7] Wang Y, Luo X, Nie L, et al. BATCH: A scalable asymmetric discrete cross-modal hashing. IEEE Transactions on Knowledge and Data Engineering, 2020, PP(99): 1-1

[8] Nie X, Liu X, Xi X, et al. Fast Unmediated hashing for cross-modal retrieval. IEEE Transactions on Circuits and Systems for Video Technology, 2020, PP(99): 1-1

[9] Ding G, Guo Y, Zhou J. Collective matrix factorization hashing for multimodal data//Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition. Columbus, USA, 2014: 2083-2090

[10] Long M, Cao Y, Wang J, et al. Composite correlation quantization for efficient multimodal retrieval//Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval. Pisa, Italy, 2016: 579-588

[11] Hu M, Yang Y, Shen F, et al. Collective reconstructive embeddings for cross-modal hashing. IEEE Transactions on Image Processing, 2019, 28(6): 2770-2784

[12] Xu X, Shen F, Yang Y, et al. Learning discriminative binary codes for large-scale cross-modal retrieval. IEEE Transactions on Image Processing, 2017, 26(5): 2494-2507

[13] Li C, Chen Z, Zhang P, et al. SCRATCH: A scalable discrete matrix factorization hashing for cross-modal retrieval// Proceedings of the ACM International Conference on Multimedia Information Retrieval. Seoul, Korea, 2018: 1-9

[14] Wang D, Gao X, Wang X, et al. Label consistent matrix factorization hashing for large-scale cross-modal similarity search. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 41(10): 2466-2479

[15] Wang Y, Luo X, Xu X. Label embedding online hashing for cross modal retrieval//Proceedings of the ACM International Conference on Multimedia Information Retrieval. Seattle, USA, 2020: 871-879

[16] Li X, Lin G, Shen C, et al. Learning hash functions using column generation//Proceedings of the International Conference on Machine Learning. Atlanta, USA, 2013: 142-150

[17] Norouzi M, Fleet D J, Salakhutdinov R R. Hamming distance metric learning//Proceedings of the Annual Conference on Neural Information Processing Systems. Lake Tahoe, USA, 2012: 1061-1069

[18] Wang Q, Zhang Z, Si L. Ranking preserving hashing for fast similarity search//Proceedings of the International Joint Conference on Artificial Intelligence. Buenos Aires, Argentina, 2015: 3911-3917

[19] Liu H, Ji R, Wu Y, et al. Towards optimal binary code learning via ordinal embedding//Proceedings of the AAAI Conference on Artificial Intelligence. Phoenix, USA, 2016: 1258-1265

[20] Liu H, Ji R, Wu Y, et al. Ordinal constrained binary code learning for nearest neighbor search//Proceedings of the AAAI Conference on Artificial Intelligence. San Francisco, USA, 2017: 2238-2244

[21] Liu L, Shao L, Shen F, et al. Discretely coding semantic rank orders for supervised image hashing//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, USA, 2017: 5140-5149

[22] Jiang Q, Li W. Deep cross-modal hashing//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, USA, 2017: 3270-3278

[23] Zhang X, Lai H, Feng J. Attention-aware deep adversarial hashing for cross-modal retrieval//Proceedings of the European Conference on Computer Vision. Munich, Germany, 2018: 614-629

[24] Li C, Deng C, Li N, et al. Self-supervised adversarial hashing networks for cross-modal retrieval//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake, USA, 2018: 4242-4251

[25] Liu X, Yu G, Domeniconi C, et al. Ranking-based deep cross-modal hashing//Proceedings of the AAAI Conference on Artificial Intelligence. Honolulu, USA, 2019: 4400-4407

[26] Zhan Y, Luo X, Wang Y, et al. Supervised hierarchical deep hashing for cross-modal retrieval//Proceedings of the ACM International Conference on Multimedia Information Retrieval. Seattle, USA, 2020: 3386-3394

[27] He J, Li M, Zhang H J, et al. Manifold-ranking based image retrieval//Proceedings of the ACM International Conference on Multimedia. New York, USA, 2004: 9-16

[28] Xu B, Bu J, Chen C, et al. Efficient manifold ranking for image retrieval//Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval. Beijing, China, 2011: 525-534

[29] Liu W, Wang J, Kumar S, et al. Hashing with graphs//Proceedings of the International Conference on Machine Learning. Bellevue, USA, 2011: 1-8

[30] Liu W, Mu C, Kumar S, et al. Discrete graph hashing//Proceedings of the Annual Conference on Neural Information Processing Systems. Montreal, Canada, 2014: 3419-3427

[31] Huiskes M J, Lew M S. The MIR Flickr retrieval evaluation//Proceedings of the ACM International Conference on Multimedia Information Retrieval. Vancouver, Canada, 2008: 39-43

[32] Chua T S, Tang J, Hong R, et al. NUS-WIDE: A real-world web image database from national university of Singapore//Proceedings of the ACM International Conference on Image and Video Retrieval. Santorini Island, Greece, 2009: 48

[33] Chatfield K, Simonyan K, Vedaldi A, et al. Return of the devil in the details: Delving deep into convolutional nets//Proceedings of the British Machine Vision Conference. Nottingham, UK, 2014: 1-5

LI Hui-Qiong, M. S. candidate. Her research interests include multimedia retrieval and computer vision.

WANG Yong-Xin, Ph. D. candidate. Her research interests include machine learning, hashing, cross-media retrieval, and computer vision.

CHEN Zhen-Duo, Ph. D. candidate. His research interests include machine learning and information retrieval.

LUO Xin, Ph. D., assistant researcher. His research interests include machine learning and media content analysis and retrieval.

XU Xin-Shun, Ph. D., professor. His research interests include machine learning, computer vision, data mining, information retrieval, and media content analysis and retrieval.

Background

Recently, multimedia data shows an explosive growth, cross-modal hashing is becoming more and more popular for multimedia retrieval due to its high efficiency and effectiveness. The idea of cross-modal hashing is to learn binary representations of data which could well preserve the similarity in the original space. As distances between two binary hash codes could be efficiently calculated by the XOR operation, cross-modal retrieval can be efficiently done. Although a variety of

cross-modal hashing methods have been proposed, there still exist some issues worthy of investigation. For example, most cross-modal hashing methods ignore the importance of the ranking orders that may indicate the top neighbors of a specific query, which makes them suboptimal. In addition, some hashing methods employ a pairwise similarity matrix to preserve similarity, which makes the algorithm complexity $O(n^2)$ and cannot extend to large-scale datasets. What's more, in order to solve the discrete optimization problem, most methods relax the discrete constraint, obtain a continuous solution and threshold the solution into binary hash codes, which may introduce serious quantization error.

To overcome the aforementioned issues, in this paper, we propose a novel supervised cross-modal hashing method, dubbed Ranking-based Supervised Discrete Cross-modal Hashing (RSDCH). RSDCH consists of ranking learning step and hashing learning step. In the first step, the proposed method learns ranking information from the manifold structure and semantic labels of data and generates a ranking score matrix. In the second step, RSDCH jointly learns hash codes and hash functions while preserving the learned ranking information. The novelty of RSDCH can be summarized as follows. (1) To make our method scalable to large-scale datasets, anchor sampling is leveraged. (2) To learn high-quality hash codes, two effective similarity-preserving strategies are proposed. (3) To avoid large quantization error, an alternative optimization algorithm, which discretely solves the binary codes learning problem, is designed. Extensive experiments on two widely-used benchmark datasets, i. e., MIRFlickr-25K and NUS-WIDE, have verified the effectiveness of our RSDCH model.

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61991411, 61872428, in part by the Shandong Provincial Key Research and Development Program under Grant No. 2019JZZY010127, in part by the Natural Science Foundation of Shandong Province under Grant Nos. ZR2019ZD06, ZR2020QF036, and in part by the Fundamental Research Funds of Shandong University under Grant No. 2019GN075.