

基于相似度驱动的线性哈希模型参数再优化方法*

聂秀山¹, 刘兴波^{2,3}, 袁肖明¹, 尹义龙³

¹(山东建筑大学 计算机科学与技术学院, 山东 济南 250101)

²(山东大学 计算机科学与技术学院, 山东 青岛 266237)

³(山东大学 软件学院, 山东 济南 250100)

通讯作者: 尹义龙, E-mail: ylyin@sdu.edu.cn



摘要: 哈希学习通过设计和优化目标函数,并结合数据分布,学习得到样本的哈希码表示.在现有哈希学习模型中,线性模型因其高效、便捷的特性得到广泛应用.针对线性模型在哈希学习中的参数优化问题,提出一种基于相似度驱动的线性哈希模型参数再优化方法.该方法可以在不改变现有模型各组成部分的前提下,实现模型参数的再优化,提升模型检索性能.该方法首先通过运行现有哈希算法多次,获得训练集的多个哈希码矩阵,然后基于相似度保持度量标准和融合准则对多个哈希码矩阵进行优化选择,获得训练集的优化哈希矩阵,最后利用该优化哈希矩阵对原模型的参数进行再优化,进而获得更优的哈希学习算法.实验结果表明,该方法对不同的哈希学习算法性能都有较为显著的提升.

关键词: 内容检索; 哈希学习; 线性模型; 参数优化; 相似度驱动

中图法分类号: TP181

中文引用格式: 聂秀山, 刘兴波, 袁肖明, 尹义龙. 基于相似度驱动的线性哈希模型参数再优化方法. 软件学报, 2020, 31(4): 1039-1050. <http://www.jos.org.cn/1000-9825/5918.htm>

英文引用格式: Nie XS, Liu XB, Xi XM, Yin YL. Model parameter re-optimization for linear hashing based on similarity drive. Ruan Jian Xue Bao/Journal of Software, 2020, 31(4): 1039-1050 (in Chinese). <http://www.jos.org.cn/1000-9825/5918.htm>

Model Parameter Re-optimization for Linear Hashing Based on Similarity Drive

NIE Xiu-Shan¹, LIU Xing-Bo^{2,3}, XI Xiao-Ming¹, YIN Yi-Long³

¹(School of Computer Science and Technology, Shandong Jianzhu University, Ji'nan 250101, China)

²(School of Computer Science and Technology, Shandong University, Qingdao 266237, China)

³(School of Software, Shandong University, Ji'nan 250100, China)

Abstract: By designing and optimizing an objective function, and combining the distribution of samples, hash learning learns the hash codes of samples. In the existing hashing models, linear model is widely used due to its conciseness and high efficiency. For the parameter optimization of linear hashing model, a model parameter re-optimization method is propose based on similarity drive, which can improve the precision of the existing linear model-based hashing algorithms. Given a hashing method, this method is firstly run for several times with obtaining several hash matrices. Then, some bits are selected for these hash matrices to obtain a new final hash matrix based on the similarity preserving degree and a fusion strategy. Finally, this new hash matrix is used to re-optimize the model parameters, and a better hash model is obtained for out-of-sample extension. Extensive experiments are performed based on three benchmark datasets and the results demonstrate the superior performance of the proposed framework.

Key words: content retrieval; hash learning; linear model; parameter optimization; similarity drive

* 基金项目: 国家自然科学基金(61671274, 61876098)

Foundation item: National Natural Science Foundation of China (61671274, 61876098)

本文由“非经典条件下的机器学习方法”专题特约编辑高新波教授、黎铭教授、李天瑞教授推荐.

收稿时间: 2019-03-09; 修改时间: 2019-07-11; 采用时间: 2019-09-20; jos 在线出版时间: 2020-01-10

CNKI 网络优先出版: 2020-01-17 14:06:23, <http://kns.cnki.net/kcms/detail/11.2560.TP.20200117.1405.001.html>

随着互联网的普及,社交平台、分享网站等网络平台丰富了图像和视频等可视媒体数据的发布、传播和获取途径.互联网和传统媒体的融合也进一步促进了图像和视频等媒体数据的爆炸性增长.数据的爆炸性增长在为提供更加丰富和多元化信息来源的同时,也使得大量内容相似或重复的图像和视频等可视媒体充斥网络,对互联网内容提供商来说,过滤此类“垃圾内容”或对搜索到的相似内容进行重排序,让用户从海量数据中快速检测到对自己有用或喜欢的媒体内容非常重要.同时,根据用户搜索或观看的媒体内容进行个性化推荐,也是互联网企业的重要业务.另外,可视媒体的广泛传播在丰富人们文化娱乐生活的同时,也为暴力恐怖、淫秽色情、谣言等有害信息的传播提供了便利,这些有害图像和视频极大地危害了社会稳定和政府公信力,淫秽图片或视频更是影响到青少年的身心健康,因此,对此类有害可视媒体内容的检测和过滤也十分必要.

针对上述问题,利用人工方式很难对海量数据进行有效审核,因此,利用计算机技术对海量可视媒体内容进行自动分析,从而高效地完成内容相似性搜索和检测是现有可行的方法.但在当前大数据环境下,内容相似性搜索检测技术面临“维数灾难”的严峻挑战,即随着内容特征维度的增加,计算量呈指数增加.同时,海量数据的内容存储和检索复杂度对现代计算机设备来说也是一个巨大的考验.虽然现有的计算机软/硬件水平和通信技术已有显著提高,但对海量数据来说,软/硬件处理能力和通信的带宽依然有限,为解决以上难题和挑战,哈希学习技术被提了出来,并成为智能信息处理领域的研究热点.

哈希学习(learning to hash)^[1]结合数据自身的特性和分布,通过机器学习方法将数据映射成二进制串的形式,同时尽可能地保持原空间中的近邻关系,哈希学习能够显著减少数据的存储和通信开销,从而有效提高学习系统的效率.在现有哈希学习算法中,无监督哈希方法和有监督哈希方法是两类主要形式.

无监督哈希学习算法在哈希函数设计或哈希码的学习过程中不需要样本标签信息,其中,谱哈希^[2,3]和迭代量化哈希^[4]及其改进版本是两类典型的无监督哈希算法.近年来,在图像检索领域,结合样本的其他属性,出现了很多无监督哈希学习算法^[5-8],例如,Zhu 等人^[5,6]提出了利用文本辅助的语义迁移构造无监督哈希算法;Liu 等人^[7]提出了基于序列紧凑码学习的无监督哈希算法;Jiang 等人^[8]提出了可扩展图哈希算法;Sheng 等人^[9]提出了基于流形的归纳式哈希算法.无监督的哈希算法大多从样本内在的特性和相互关系的角度设计哈希模型,不需要样本的标签信息,因此,在样本标记成本过高或者在无法获得大规模数据样本标签的情境下,无监督哈希算法起到了重要的作用.

与无监督哈希算法相比,有监督哈希学习在哈希学习过程中充分利用了样本标签信息.例如,Lin 等人^[10]结合标签信息,提出一种使用图割和决策树构建的快速监督哈希方法;Wang 等人^[11]通过最小化经验误差,提出一种半监督哈希学习算法;Liu 等人^[12]提出了基于核函数的监督哈希学习算法;Nie 等人^[13,14]结合样本类标记和相似性标记分别提出了针对视频和跨模态检索的哈希学习算法.另外,一些以样本顺序作为监督信息的学习算法^[15-18]也是常见的有监督哈希学习方法.近年来,随着深度学习的发展,基于深度学习的监督哈希算法也陆续被提出^[19-23].监督哈希学习算法充分利用了数据标签信息,在性能上与无监督算法相比有了很大的提升.因此,近年来越来越多的有监督哈希学习算法被研究者所关注.

在现有监督或无监督的哈希学习算法中,线性模型、核函数模型、神经网络模型等都是常用的哈希函数形式和模型.哈希学习的目标之一是实现海量数据的快速检索,而线性模型因其计算简单,运行效率高,更符合哈希学习算法快速检索的目标,所以在哈希学习中得到广泛应用.另外,一些核函数模型也是线性模型的变种,因此,线性模型在哈希学习算法中具有重要的地位.一般来说,基于线性模型的哈希学习算法通过一系列线性关系的项来构造目标函数,进而学习哈希函数.

具体来说,设有 N 个样本组成的样本特征矩阵为 \mathbf{X} ,训练样本的标签矩阵为 \mathbf{Y} ,标签矩阵 \mathbf{Y} 的元素为 1 或 0,分别表示属于或不属于对应类别.线性哈希模型的目标是学到一个可以保持样本特征空间相似性的哈希矩阵 \mathbf{B} .结合数据的分布和内在特性,线性哈希模型的形式可以设计成多种形式,同时也可以增加多种约束项和正则项.其中,哈希码和原始特征之间的线性回归项是主要部分,其刻画了特征矩阵和哈希矩阵之间的关系,通过对参数的学习和优化,可以实现样本外的哈希学习.除了特征和哈希码的线性关系项之外,在无监督哈希算法中,还通常包含样本原始特征相似关系和哈希码相似关系保持项,而在有监督哈希算法中,标签矩阵和哈希矩阵

以及标签矩阵和特征矩阵之间的线性回归项则是目标函数中常见的部分.不失一般性,式(1)给出了一类常见的有监督线性哈希算法的目标函数:

$$\begin{cases} \min_{B,P,W} \|B - P^T X\|_F^2 + \nu \|Y - BW\|_F^2 + \lambda \|W\|_F^2 \\ \text{s.t. } B \in \{-1,1\}^{n \times l} \end{cases} \quad (1)$$

其中, $\|\cdot\|_F$ 代表矩阵的 F 范数.第 1 项代表哈希矩阵和原始特征矩阵的线性关系,第 2 项代表标签矩阵和哈希矩阵的线性关系.该优化问题可以通过交替优化算法来解决.现有线性哈希模型大多是以上目标函数的变形或改进,同时再设计不同的约束项和正则项,构造不同的哈希学习算法.与现有算法不同,本文方法并不设计新的目标函数,也不增加任何约束项和正则项.对于给定的已有哈希学习算法,本文提出一个基于相似度驱动的参数再优化框架,通过该框架可以在不增加任何目标函数项和约束项的情况下,学习得到更优的目标函数参数,进一步提升现有哈希算法的性能.

哈希学习的一个重要特性就是相似性保持,即在原始特征空间相似的样本,其哈希码在汉明空间也要相似(汉明距离较小).越能够保持此相似性的哈希码,其检索性能越好.因此,如果在训练过程中能够获得训练集样本较优的哈希码,则更能够学习到哈希模型较优的参数.基于此动机,本文提出了一个基于相似度驱动的线性哈希模型参数再优化的框架.该框架的主要思路为:给定一个线性哈希模型,首先在训练集上运行该模型多次,得到训练集的多个哈希矩阵.然后基于相似度保持水平来选取较优的哈希比特作为训练集样本的哈希矩阵,利用此哈希矩阵作为哈希模型的输入,再次优化给定哈希模型,获得更优的哈希模型参数,用于生成样本外哈希码.

本文的主要贡献总结如下.

(1) 提出一个哈希学习模型参数再优化框架.该框架可以应用到现有基于线性模型的哈希学习算法中,并有效提升相应哈希学习算法的检索性能;

(2) 提出一种训练集样本优化哈希码的选择方案,并给出理论分析.该方案的主要目的是为参数再优化提供基础,通过该方案,在哈希模型再优化的过程中,可以把哈希矩阵作为已知项,减少待优化参数,提高效率;

(3) 在常用的公开数据库上,把本文提出的框架应用到不同的哈希学习算法中,实验结果表明,本文方法对各类型哈希算法的性能都可以实现有效提升,在部分情况下可以获得 20% 以上的性能提升.

本文第 1 节对基于相似度驱动的线性哈希模型参数再优化方法进行详细介绍,包括基于相似度驱动的优化哈希码选择策略和哈希模型再优化方案.第 2 节是实验结果分析,本节把本文的方法在公开数据库上进行实验,并对实验结果进行分析.第 3 节总结全文,并对未来的研究内容进行说明.

1 基于相似度驱动的线性哈希模型参数再优化方法

本文方法的框架图如图 1 所示,分为两个过程.

- (1) 基于相似度驱动的较优哈希码选择;
- (2) 线性哈希模型参数再优化.

本文方法的相关符号定义和说明见表 1.以下各小节将对该方法进行详细介绍.

Table 1 Notation of symbols

表 1 符号定义和说明

名称	描述
N	训练集样本个数
L	样本哈希码长度
T	哈希模型运行次数
H_A	给定的哈希模型
sp	相似度保持水平
b_i	哈希矩阵的第 i 行
B_i	哈希模型第 i 次运行得到的哈希矩阵
B	经过哈希比特选择后得到的较优哈希矩阵
X	样本特征矩阵
P	特征矩阵和哈希矩阵之间的映射矩阵

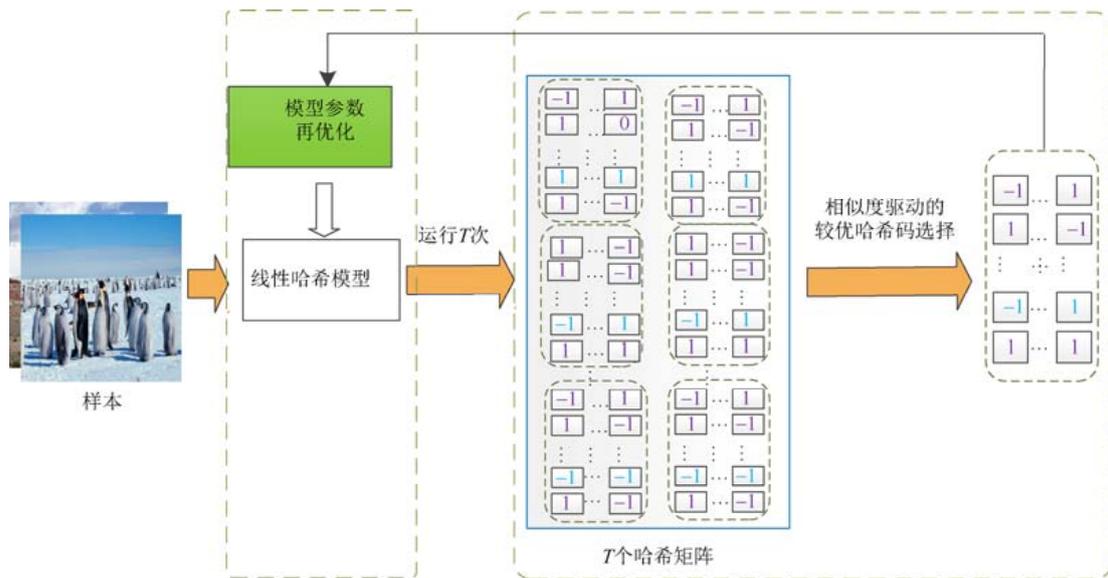


Fig.1 Framework of the proposed method

图1 本文方法的框架图

1.1 基于相似度驱动的较优哈希码选择

哈希码的选择是指如何在多个哈希码矩阵中选择较优的比特构造训练集样本的哈希码矩阵,本文提出了一个基于相似度驱动的哈希码选择策略.如上文所述,相似度保持是哈希学习算法的核心目标,所谓相似度保持,即在特征空间相似的样本,其对应的哈希码在汉明空间中的距离要尽可能地小.为了利用相似度保持来选择较优哈希码,本文首先定义相似度保持水平,然后给出一个哈希码乱序相似度保持定理,最后基于相似度保持水平和相似度保持定理介绍一个较优的哈希码选择方案.

对于给定的样本哈希矩阵 $B \in \{-1, 1\}^{L \times N}$,该矩阵第 i 行记为 b_i ,而 b_i 也是所有样本的第 i 比特组成的向量.设矩阵 $S = \{s_{ij}\}$ 为样本的相似度矩阵,对于有监督哈希学习算法,相似度矩阵可以利用样本标签计算,若第 i 个样本和第 j 个样本至少有一个相同标签,则 $s_{ij}=1$,否则, $s_{ij}=-1$.对于无监督哈希学习算法,相似度矩阵可以通过聚类算法计算,若第 i 个样本和第 j 个属于同一个类,则 $s_{ij}=1$,否则, $s_{ij}=-1$.

样本的第 i 位比特的相似度保持水平 sp 定义如下:

$$sp = \|b_i^T b_i - S\|^2 \tag{2}$$

相似度保持水平 sp 反映了样本哈希码第 i 位的相似度保持的程度,该位的 sp 值越小,说明在此位上相似度保持得越好.

为了介绍较优哈希码选择策略,本文给出如下哈希矩阵的乱序相似度保持定理.

定理 1. 在通过线性哈希模型得到的样本哈希矩阵中,矩阵行的顺序对样本语义相似保持没有影响.

证明:本文从两个角度对该定理进行证明.首先,从相似度保持的角度,设样本哈希矩阵为 B ,对哈希矩阵行进行乱序后得到的矩阵为 \bar{B} ,可以推导出, $\bar{B}^T \bar{B} = B^T B$,因此,经过行乱序后,样本哈希矩阵的相似度不变.

其次,从哈希表示的信息损失的角度进行证明.本文用一个简单的线性哈希模型来证明经过行乱序,样本的哈希表示和原始特征之间的损失不变.不失一般性,设乱序前的线性哈希模型的优化问题表示如下:

$$\min_P \|B - P^T X\|_F^2 + \lambda \|P\|_F^2 \quad \text{s.t. } B \in \{-1, 1\}^{L \times N} \tag{3}$$

为优化映射矩阵 P ,目标函数对矩阵 P 求导数,并让导数等于 0,则得到矩阵 P 的优化解:

$$P = (XX^T + \lambda I)^{-1} XB^T \tag{4}$$

乱序后的优化问题,只需要用乱序后的矩阵 \bar{B} 代替优化问题(3)中的矩阵 B ,同样的求解过程,可以得到乱序

后哈希矩阵和原始特征的映射矩阵 \bar{P} :

$$\bar{P} = (XX^T + \lambda I)^{-1} X\bar{B}^T \tag{5}$$

设 $Q = (XX^T + \lambda I)^{-1} X$, 则:

$$\|B - P^T X\|^2 = \|B - BQ^T X\|^2 = \|B(I - Q^T X)\|^2 = \|BM\|^2 = Tr(M^T B^T BM) = Tr(B^T BMM^T) \tag{6}$$

其中, $Tr()$ 表示迹运算, $M = I - Q^T X$.

又因为 $\bar{B}^T \bar{B} = B^T B$, 则可以得到 $\|B - P^T X\|^2 = \|\bar{B} - \bar{P}^T X\|^2$, 由此可以看出, 行乱序后, 样本哈希矩阵和原始特征之间的信息损失不变.

综上所述可以得到, 哈希矩阵行的顺序对样本语义保持没有影响. □

对于一种哈希学习算法来说, “好” 的样本特征有利于模型参数的优化和学习. 如果把样本的哈希码作为样本特征的一种表示形式, 那么哈希码的每一位比特就是特征表示的每一个维度, 其对样本的表示越 “好”, 则对模型参数的训练和优化就越有利, 而对哈希码每一位比特来说, “好” 指的是能够充分地保留样本原始空间的相似性. 因此, 为了学习得到更好的模型参数, 需要得到训练集样本较 “好” 的哈希码表示. 基于以上定义和定理, 本文提出一个较优哈希码选择方案, 具体过程如下.

对于给定的一种线性哈希算法, 首先在训练集上运行 T 次, 得到训练集的 T 个哈希矩阵 $\{B_i\}_{i=1}^T$, 然后在列方向上, 把 T 个哈希矩阵连接成一个较长的矩阵 H . 其次, 计算哈希矩阵每一行的相似度保持水平 sp 值, 并对 sp 值进行排序, 若用户最后期望得到的数据哈希码长度为 L , 则在矩阵 H 中, 选取 sp 值最小的前 L 行构成训练集样本的较优哈希矩阵 B , 因为矩阵 B 中每一行都有较好的相似度保持水平. 为了更好地理解该过程, 图 2 给出了一个简单的示例, 在图 2 中, $T=3, L=2, N=6$, 在对给定的哈希模型运行 3 次后, 得到 3 个哈希矩阵 B_1, B_2 和 B_3 , 在列方向上把 3 个矩阵连接成矩阵 H , 通过计算 sp 值和排序, 得到训练集样本的较优哈希矩阵 B .

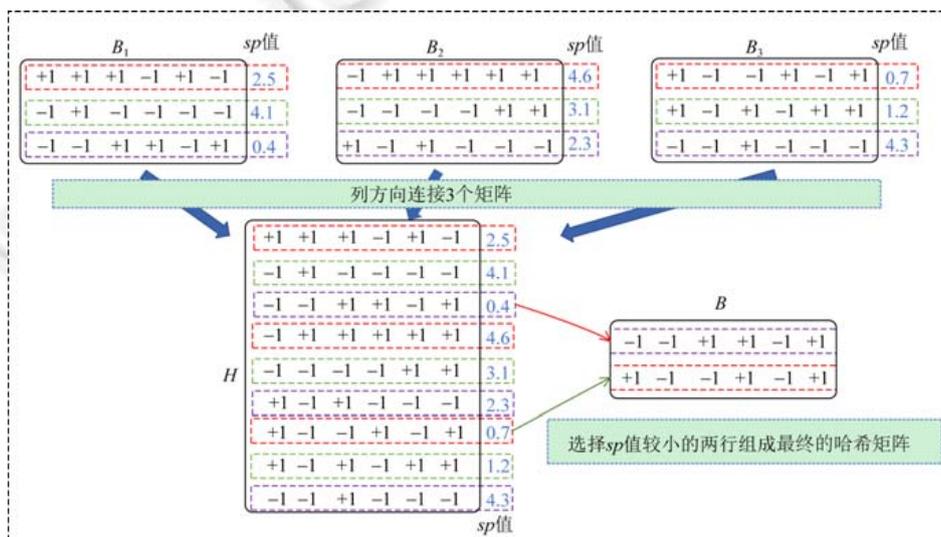


Fig.2 Illustration of hash bit selection

图 2 较优哈希比特选择示意图

1.2 模型参数再优化

利用以上选择策略得到的训练集哈希矩阵, 对原始哈希模型进行重新优化, 不失一般性, 本文用下式表示一个常见的线性哈希模型, 对无监督或有监督的哈希学习算法, 则再需要加上相关的项和约束即可.

$$\min_P \|B - P^T X\|_F^2 + \lambda \|P\|_F^2 \quad \text{s.t. } B \in \{-1, 1\}^{L \times N} \tag{7}$$

在此优化问题中, 哈希矩阵 B 由第 1.1 节得到的最优哈希码矩阵代替, 因此, 该优化问题只需要优化映射矩阵 P 即可. 为得到优化映射矩阵, 目标函数对矩阵 P 求导数, 并令导数等于 0, 则可以得到 P 的优化解:

$$\mathbf{P} = (\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})^{-1} \mathbf{X}\mathbf{B}^T \quad (8)$$

综上所述,本文提出的基于相似度驱动的线性哈希模型参数再优化框架如算法 1 所示.

算法 1. 基于相似度驱动的线性哈希模型参数再优化方法.

输入:训练集特征矩阵 \mathbf{X} ,期望哈希码长度 L ;给定的哈希算法 HA;哈希模型运行次数 T ;

1. 初始化哈希码和特征映射矩阵 \mathbf{P}
2. 运行给定哈希算法 T 次,得到 T 个样本哈希矩阵
3. 利用哈希码选择策略得到训练集样本的较优哈希矩阵
4. 利用式(7)和式(8)重新优化模型参数

输出:模型参数 \mathbf{P}

1.3 复杂度分析

假设原哈希模型算法复杂度为 C ,训练集样本数目为 N ,样本哈希码长度为 L ,则运行 T 次得到 T 个哈希矩阵的复杂度近似为 TC .相似度保持水平值 sp 计算的复杂度为 $O(NL)$,相似度保持水平值 sp 排序的复杂度为 $O(NL \lg L)$,模型再优化过程的计算复杂度为 $O(NdL + Nd^2)$,其中, d 为样本特征维度.因此,本文方法的总复杂度为 $O(NdL + Nd^2 + TL \lg L + NTL) + TC$.而通常情况下哈希码长度 L 和原始模型运行次数 T 远小于样本特征维度 d 和样本数目 N ,因此本文方法总复杂度可以近似为 $O(Nd^2) + TC$.进一步地,线性哈希模型再优化映射矩阵 \mathbf{P} 的复杂度往往远小于原始哈希算法复杂度,因此,近似来说,本文提出的方案的算法复杂度是原始哈希模型复杂度的 T 倍.直观来看, T 值越大,本文方法再优化参数的性能就越好,检索精度就越高,因为较大的 T 值可以提供较多哈希矩阵以供选择.但当 T 值增大时,本文方法的时间和空间复杂度也会变大.本文在实验部分针对 T 的不同取值做了相关实验发现,当 T 较大时,检索精度的提升很缓慢,一个可能的原因就是,当原始哈希模型运行的次数较大时,得到的哈希矩阵的变化就很小了,哈希矩阵之间的 sp 值变化也就很小.因此,为了降低时间和空间复杂度,在几乎不损失精度的前提下,可以使用较小的 T 值(本文实验中 $T=3$).总体来说,本文提出的方法与原始模型相比,复杂度增加较少,但是哈希算法检索性能的提升却很大,下面第 2 节中的实验结果也说明了这一点.

2 实验

2.1 实验设置

本文方法主要在 3 个公开的图像数据库 CIFAR-10^[24]、MS-COCO^[25]和 NUS-WIDE^[26]进行了实验,3 个数据库的相关信息介绍如下.

CIFAR-10:该数据库为单标签数据库,包含 60 000 幅图像,其中有 50 000 个训练和 10 000 个测试图像.图像由 10 个类组成,每个类有 6 000 个图像.本文实验随机地从每个类中选取 5 000 个图像作为训练集,1 000 幅图像作为测试集.

MS-COCO:该数据库为多标签数据库,包含 82 783 幅图像,共 91 个类.根据现有哈希算法的通用方式,在本文实验中,如果两幅图片中有至少一个相同的标签,就认为这两幅图片是相似的.实验中,本文方法随机地选择 10 000 幅图像作为训练集,5 000 幅图像作为测试集.

NUS-WIDE:该数据库包含了 269 648 幅图像,对应于 1 000 个类别标签,本文实验中选取 21 个包含图片较多的类别,共约 195 834 幅图片.同样地,与 MS-COCO 数据库的处理方式类似,在本文实验中,如果两幅图片有至少一个相同的标签,就认为这两幅图片是相似的.同时,本文实验在每类中随机选取 10 500 幅图像作为训练集,2 100 幅图像作为测试集.

为证明本文方法的有效性,将本文方法分别应用到不同的哈希算法:(1) 快速监督离散哈希(FSDH)^[27];(2) 快速扩展哈希(FSSH)^[28];(3) 基于列采样的离散监督哈希(COSDISH)^[29];(4) 扩展离散哈希(SSDH)^[30].实验中,对给定哈希模型运行 3 次(即 $T=3$).

本文采用平均精度均值(mean average precision,简称 mAP)作为性能指标来展示本文方法在不同哈希算法

上应用的性能,这也是多媒体检索中常用的评价准则,mAP 是指检索到的全部样本的精度均值(AP).

$$mAP = \frac{1}{Q} \sum_{i=1}^Q AP(i) \tag{9}$$

其中, Q 是查询图像的数量, $AP(i)$ 表示第 i 个样本的精度均值.AP 定义为

$$AP = \frac{1}{R} \sum_{r=1}^G precision(r)\sigma(r) \tag{10}$$

其中, R 表示检索样本 G 中相关实例的数量.如果第 r 个实例与查询相关,则 $\sigma(r)=1$;否则为 0. $precision(r)$ 表示第 r 个实例的检索精度,定义如下:

$$precision = \frac{Num(\text{True Positive})}{Num(\text{True Positive})+Num(\text{False Positive})} \tag{11}$$

其中, $Num()$ 表示相关类别的数目.

2.2 实验结果与分析

表 2~表 4 展示了在 3 个公开数据库上本文方法应用在不同的哈希算法上的 mAP 的值.其中,样本哈希码长度分别取 24、48 和 64.“方法名称- i ”表示相关方法第 i 次运行后的结果,“方法名称- s ”表示把本文方法应用到相关哈希算法后的结果.从表 2 所示结果可以看出,本文提出的再优化方法对现有哈希算法性能的提升有显著效果.特别地,在 COSDISH 算法上的性能更是提高了 20% 以上,在其他算法上对原算法性能的提升也在 3%~8% 左右.

Table 2 Performance in terms of mAP score on Cifar-10

表 2 在数据库 Cifar-10 上的 mAP 性能结果

算法名称	24 bit	48 bit	64 bit
FSDH-1	0.648 1	0.685 5	0.6880
FSDH-2	0.634 7	0.674 6	0.689 2
FSDH-3	0.653 1	0.679 2	0.682 7
FSDH- s	0.670 3	0.698 5	0.700 8
FSSH-1	0.653 9	0.672 2	0.677 8
FSSH-2	0.664 9	0.668 1	0.677 4
FSSH-3	0.641 2	0.663 9	0.678 5
FSSH- s	0.650 4	0.673 7	0.680 3
COSDISH-1	0.436 1	0.475 9	0.502 0
COSDISH-2	0.419 3	0.504 9	0.484 7
COSDISH-3	0.471 0	0.521 9	0.488 2
COSDISH- s	0.559 9	0.620 7	0.605 9
SSDH-1	0.668 9	0.668 5	0.682 4
SSDH-2	0.610 0	0.682 4	0.675 9
SSDH-3	0.621 0	0.677 0	0.675 8
SSDH- s	0.659 9	0.679 5	0.676 0

Table 3 Performance in terms of mAP score on MS-COCO

表 3 在数据库 MS-COCO 上的 mAP 性能结果

算法名称	24 bit	48 bit	64 bit
FSDH-1	0.666 3	0.804 5	0.789 1
FSDH-2	0.706 5	0.747 5	0.775 3
FSDH-3	0.723 6	0.772 0	0.751 5
FSDH- s	0.783 9	0.827 1	0.829 1
FSSH-1	0.836 4	0.857 5	0.870 1
FSSH-2	0.826 9	0.859 9	0.866 4
FSSH-3	0.827 9	0.858 7	0.872 1
FSSH- s	0.850 6	0.862 2	0.877 6
COSDISH-1	0.584 7	0.580 4	0.562 9
COSDISH-2	0.565 8	0.665 6	0.642 5
COSDISH-3	0.505 4	0.593 8	0.666 8
COSDISH- s	0.714 0	0.733 5	0.741 7
SSDH-1	0.708 4	0.694 2	0.712 3
SSDH-2	0.694 2	0.698 3	0.712 4
SSDH-3	0.673 0	0.667 8	0.697 0
SSDH- s	0.717 6	0.780 6	0.789 6

Table 4 Performance in terms of mAP score on NUS-WIDE

表 4 在数据库 NUS-WIDE 上的 mAP 性能结果

算法名称	24 bit	48 bit	64 bit
FSDH-1	0.757 5	0.781 4	0.784 8
FSDH-2	0.770 3	0.784 9	0.782 0
FSDH-3	0.772 9	0.785 0	0.780 8
FSDH-s	0.777 7	0.795 1	0.798 1
FSSH-1	0.790 3	0.809 2	0.818 2
FSSH-2	0.782 3	0.809 2	0.811 2
FSSH-3	0.792 1	0.815 4	0.811 5
FSSH-s	0.775 9	0.802 2	0.826 8
COSDISH-1	0.536 3	0.554 5	0.612 1
COSDISH-2	0.516 7	0.579 9	0.632 0
COSDISH-3	0.496 6	0.623 3	0.549 9
COSDISH-s	0.623 1	0.693 8	0.707 6
SSDH-1	0.636 5	0.624 6	0.660 7
SSDH-2	0.645 7	0.657 8	0.662 8
SSDH-3	0.636 4	0.647 1	0.667 1
SSDH-s	0.658 4	0.641 8	0.675 2

不可否认,在 NUS-WIDE 数据库上,某些情况下,本文方法应用在 FSSH 和 SSDH 两种算法上并没有得到性能的提升,其中一个可能的原因是,这两种算法在原始模型的构造中对哈希码相似度的特性已经进行了充分利用,因此其运行得到的哈希码的相似度保持水平已经较高,所以本文方法在应用到此类算法时,在哈希码的优化选择上并没有体现出显著优势.但本文方法应用到这两种算法时仍然取得了与原算法相似的性能.

图 3~图 5 分别展示了在 3 个数据库上本文方法应用到不同哈希算法后的检索精度(precision)和哈希码长度的关系图,其中,哈希码长度从 12 比特到 128 比特之间变化.在各图中,“方法名称_{*i*}”表示相关方法第 *i* 次运行后的结果,“方法名称-s”表示把本文方法应用到相关方法后的结果,“Precision@5000”表示仅考虑检索到的最相关的前 5 000 个结果的精度.由图 3~图 5 同样可以看出,本文方法对哈希算法的性能提升具有显著的作用.

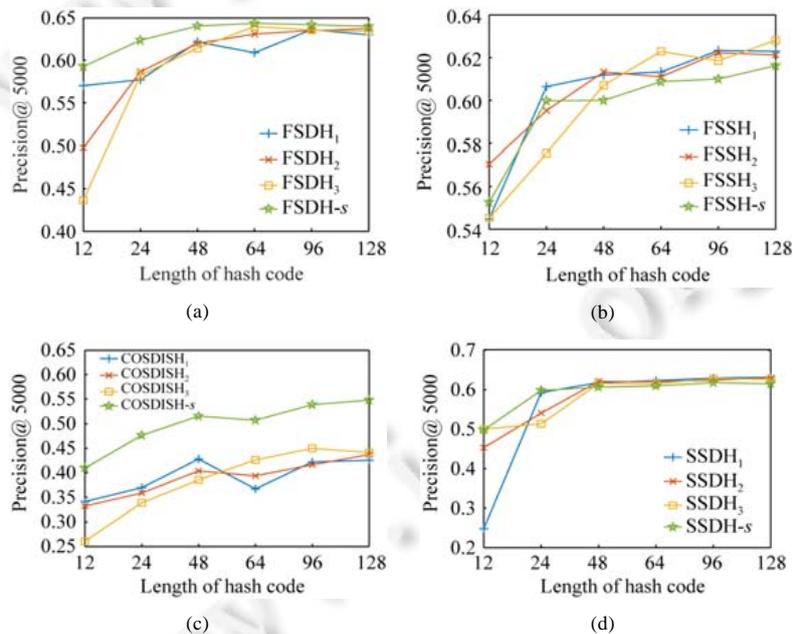


Fig.3 Performance of applying the proposed method to different hashing methods (Precision v.s. hash length. Dataset-Cifar-10)

图 3 本文方法在不同哈希算法的应用性能图(精度 v.s.哈希码长度,数据库 Cifar-10)

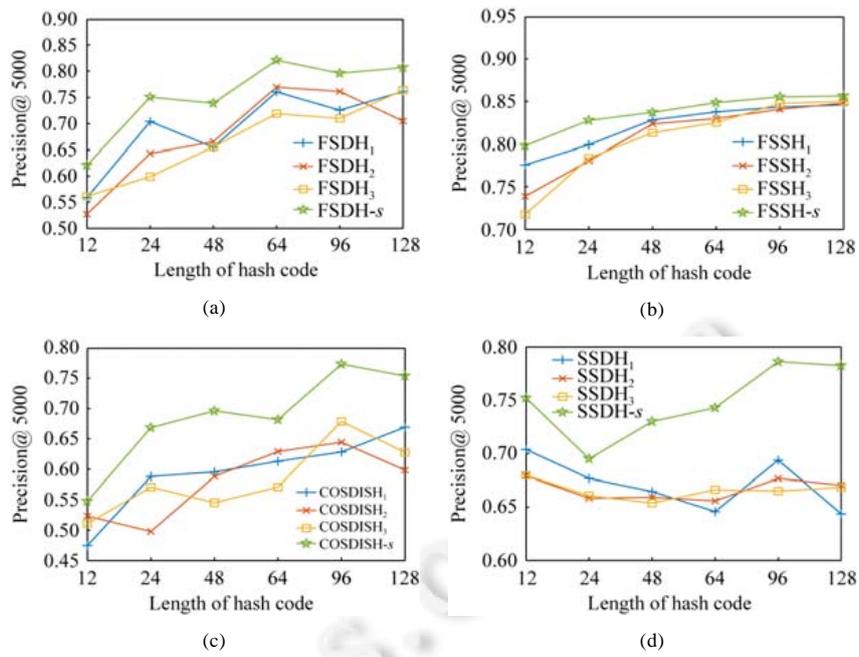


Fig.4 Performance of applying the proposed method to different hashing methods (Precision v.s. hash length. Dataset-MS-COCO)

图 4 本文方法在不同哈希算法的应用性能图(精度 v.s.哈希码长度,数据库 MS-COCO)

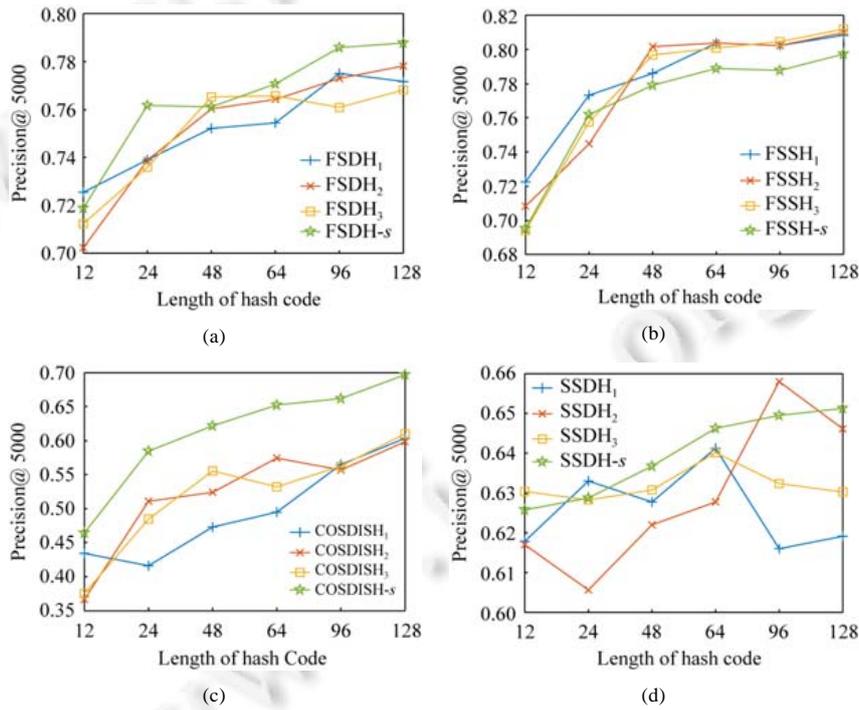


Fig.5 Performance of applying the proposed method to different hashing methods (Precision v.s. hash length. Dataset-NUS-WIDE)

图 5 本文方法在不同哈希算法的应用性能图(精度 v.s.哈希码长度,数据库 NUS-WIDE)

图6和图7展示了本文方法应用到不同算法后的精度性能与原始算法运行次数以及哈希码长度的关系图,由图可以看出,本文方法应用后的精度性能随着运行次数的增加,性能的变化很小,这说明,在本文方法中,原始模型运行次数 T 取一个较小的值即可.同时也说明,本文方法在不同哈希算法应用后取得了较为显著的检索性能上的提升,但时间和空间复杂度上仅有少许增加.

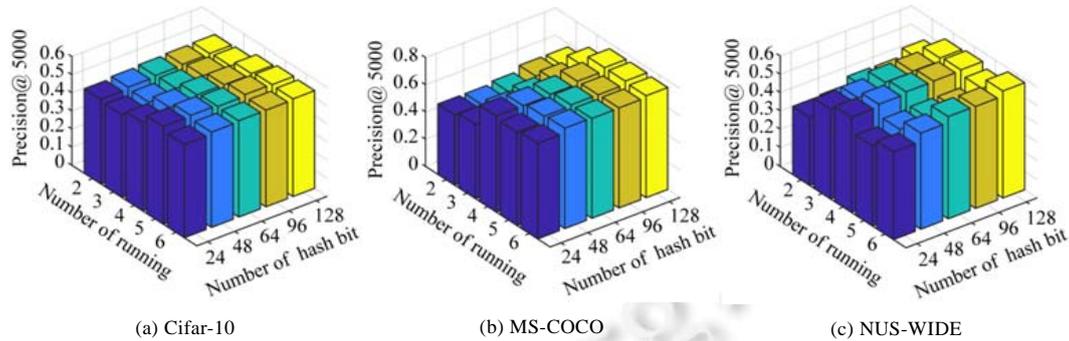


Fig.6 Performance of applying the proposed method to the hashing method COSDISH with different datasets (Precision v.s. number of running v.s. hash length)

图6 在不同数据库上本文方法应用在哈希算法 COSDISH 上的性能图(精度 v.s.运行次数 v.s.哈希码长度)

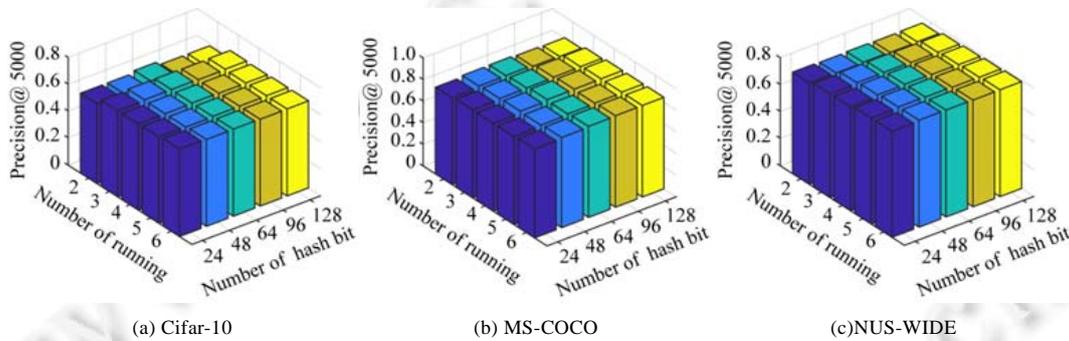


Fig.7 Performance of applying the proposed method to the hashing method FSDH with different datasets (Precision v.s. number of running v.s. hash length)

图7 在不同数据库上本文方法应用在哈希算法 FSDH 上的性能图(精度 v.s.运行次数 v.s.哈希码长度)

3 结论

本文提出了一种基于相似度驱动的线性哈希模型参数再优化方法,该方法对于给定的线性哈希模型,通过两步策略,获得原模型的更优参数,从而提升了原始算法的性能.在这两步策略中,第1步通过对原始模型的多次运行获得多个训练集哈希矩阵,然后根据本文提出的相似度保持水平度量和融合选择策略,优选得到训练集样本较优的哈希矩阵;在第2步中,将得到的较优哈希矩阵作为引导,再次优化模型参数,获得原模型更优的参数表示.实验结果表明,本文提出的方法对原始哈希模型的检索性能有显著提升.

线性模型因其简洁和高效在哈希模型领域取得广泛应用,因此,本文针对线性哈希模型的再优化问题进行了研究,未来的工作将在非线性哈希模型上进一步加以拓展和研究.

References:

- [1] Li WJ, Zhou ZH. Learning to hash for big data: Current status and future trends. Chinese Science Bulletin, 2015,60:485-490. [doi: 10.1360/N972014-00841]
- [2] Weiss Y, Torralba A, Fergus R. Spectral hashing. In: Advances in Neural Information Processing Systems. 2009. 1753-1760.

- [3] Liu Q, Liu G, Li L, Yuan XT, Wang M, Liu W. Reversed spectral hashing. *IEEE Trans. on Neural Networks and Learning Systems*, 2018,29(6):2441–2449.
- [4] Gong Y, Lazebnik S, Gordo A, Perronnin F. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2013,35(12):2916–2929.
- [5] Zhu L, Shen J, Xie L, Cheng Z. Unsupervised visual hashing with semantic assistant for content-based image retrieval. *IEEE Trans. on Knowledge and Data Engineering*, 2017,29(2):472–486.
- [6] Zhu L, Huang Z, Li Z, Xie L, Shen HT. Exploring auxiliary context: Discrete semantic transfer hashing for scalable image retrieval. *IEEE Trans. on Neural Networks and Learning Systems*, 2018,29(11):5264–5276.
- [7] Liu L, Shao L. Sequential compact code learning for unsupervised image hashing. *IEEE Trans. on Neural Networks and Learning Systems*, 2016,27(12):2526–2536.
- [8] Jiang QY, Li WJ. Scalable graph hashing with feature transformation. In: *Proc. of the 24th Int'l Joint Conf. on Artificial Intelligence*. Buenos Aires, 2015. 248–2254.
- [9] Shen FM, Shen CH, Shi QF, *et al.* Inductive hashing on manifolds. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*. Portland: IEEE Computer Society, 2013. 1562–1569.
- [10] Lin GS, Shen CH, Van DH. Supervised hashing using graph cuts and boosted decision trees. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2015,37(11):2317–2331.
- [11] Wang J, Kumar S, Chang SF. Semi-supervised hashing for large-scale search. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2012,34(12):2393–2406.
- [12] Liu W, Wang J, Ji R, Jiang YG. Supervised hashing with kernels. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*. Providence: IEEE Computer Society, 2012. 2074–2081.
- [13] Nie XS, Lin PG, Yang MZ, Yin YL. Hierarchical feature fusion hashing for near-duplicate video retrieval. *SCIENTIA SINICA Information Science*, 2018,48(12):1697–1708.
- [14] Liu XB, Nie XS, Zeng WJ, Cui CR, Zhu L, Yin YL. Fast discrete cross-modal hashing with regressing from semantic labels. In: *Proc. of the 26th ACM Int'l Conf. on Multimedia*. Seoul: Association for Computing Machinery Inc., 2018. 1662–1669.
- [15] Wang Q, Zhang Z, Si L. Ranking preserving hashing for fast similarity search. In: *Proc. of the 24th Int'l Joint Conf. on Artificial Intelligence*. Buenos Aires, 2015. 3911–3917.
- [16] Wang J, Liu W, Sun AX, Jiang YG. Learning hash codes with listwise supervision. In: *Proc. of the IEEE Int'l Conf. on Computer Vision*. Sydney: Institute of Electrical and Electronics Engineers Inc., 2013. 3032–3039.
- [17] Gui J, Liu T, Sun Z, Tao D, Tan T. Supervised discrete hashing with relaxation. *IEEE Trans. on Neural Networks and Learning Systems*, 2016,29(3):608–617.
- [18] Xia R, Pan Y, Lai H, Liu C, Yan S. Supervised hashing for image retrieval via image representation learning. In: *Proc. of the 28th AAAI Conf. on Artificial Intelligence*. Quebec City: AI Access Foundation, 2014. 2156–2162.
- [19] Lai H, Pan Y, Liu Y, Yan S. Simultaneous feature learning and hash coding with deep neural networks. In: *Proc. of the 2015 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*. Boston: IEEE Computer Society, 2015. 3270–3278.
- [20] Li WJ, Wang S, Kang WC. Feature learning based deep supervised hashing with pairwise labels. In: *Proc. of the 25th Int'l Joint Conf. on Artificial Intelligence*. New York, 2016. 1711–1717.
- [21] Zhu H, Long M, Wang J, Cao Y. Deep hashing network for efficient similarity retrieval. In: *Proc. of the 30th AAAI Conf. on Artificial Intelligence*. Phoenix: AAAI Press, 2016. 2415–2421.
- [22] Jiang QY, Li WJ. Asymmetric deep supervised hashing. In: *Proc. of the 32nd AAAI Conf. on Artificial Intelligence*. New Orleans: AAAI Press. 2018. 3342–3349.
- [23] Shen F, Gao X, Liu L, Yang Y, Shen HT. Deep asymmetric pairwise hashing. In: *Proc. of the 2017 ACM on Multimedia Conf*. Mountain View: Association for Computing Machinery Inc., 2017. 1522–1530.
- [24] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. Technical Report, 2009. 54–60.
- [25] Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollar P, Zitnick CL. Microsoft coco: Common objects in context. In: *Proc. of the 2014 European Conf. on Computer Vision*. Zurich: Springer-Verlag, 2014. 740–755.

- [26] Chua TS, Tang J, Hong R, Li H, Luo Z, Zheng Y. NUS-wide: A real-world Web image database from national university of Singapore. In: Proc. of the ACM Int'l Conf. on Image and Video Retrieval. WeKnow: Association for Computing Machinery, 2009. 368–375.
- [27] Gui J, Liu TL, Sun ZN, Tao DC, Tan TN. Fast supervised discrete hashing. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2018,40(2):490–496.
- [28] Luo X, Nie LQ, He XN, Wu Y, Chen ZH, Xu XS. Fast scalable supervised hashing. In: Proc. of the 41st Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. Ann Arbor: Association for Computing Machinery Inc., 2018. 735–744.
- [29] Kang WC, Li WJ, Zhou ZH. Column sampling based discrete supervised hashing. In: Proc. of the 30th AAAI Conf. on Artificial Intelligence. Phoenix: AAAI Press, 2018. 1230–1236.
- [30] Luo X, Wu Y, Xu XS. Scalable supervised discrete hashing for large-scale search. In: Proc. of the 2018 World Wide Web Conf. on World Wide Web. Lyon: Int'l World Wide Web Confs. Steering Committee, 2018. 1603–1612.



聂秀山(1981—),男,江苏丰县人,博士,教授,博士生导师,CCF 高级会员,主要研究领域为机器学习,智能媒体处理.



秦肖明(1987—),男,博士,讲师,主要研究领域为图像处理,机器学习.



刘兴波(1995—)男,学士,主要研究领域为智能媒体处理.



尹义龙(1972—),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为机器学习,数据挖掘.