# 基于变长哈希编码的跨模态检索

摘要：

针对丰富的大规模多模态数据（文本、图像、音频等），基于哈希编码的快速跨模态检索技术已成为研究热点。大多数现存的方法仅仅是把多模态数据映射到共同的哈希编码空间、并且松弛对哈希编码的二值约束求解，因此学习到的多模态哈希编码不够有效地表示原始的多模态数据以及哈希编码类别区分性较弱。针对这些问题，本文提出了基于语义保持的方式把各模态数据分别映射到各自的理想长度的哈希编码空间，再通过二值约束的离散跨模态哈希算法求解各模态数据的哈希编码，最终在潜在的空间内比较多模态数据的相似性。并在 WIKI 数据集、NUS-WIDE 数据集和 MIRFlickr 数据集中的实验效果总体上优于相关的对比方法，从而验证了本文方法的有效性。

For rich large-scale multimodal data (text, images, audio, etc.), fast cross-modal retrieval techniques based on hash coding have become a research hotspot. Most of the existing methods merely map multimodal data to a common hash code space and loosely solve the binary constraint of the hash code, so the learned multi-modal hash code is not efficient enough to represent the original multiple Modal data and hash coding categories are less distinguishable. To solve these problems, this paper proposes a semantic-preserving way to map each modal data to its own ideal hash-coded space, and then solves each modal data by the binary constraint-based discrete cross-modality hash algorithm. The Greek code eventually compares the similarities of multimodal data in the potential space. The experimental results of the WIKI dataset, the NUS-WIDE dataset, and the MIRFlickr dataset are generally superior to the related comparison methods, thus validating the effectiveness of the proposed method.

## 0 引言

随着进入大数据时代，各行各业的数据资料疯狂倍增，尤其是互联网上丰富的大规模多模态数据（文本博客、图像、音频等），为跨模态检索技术研究提供了数据基础。此外，传统的单模态检索方式已经不够满足如今的检索需求，人们更希望输入一个语义主题可以检索出相关语义的多模态数据的结果，使得检索信息丰富、多元化。然而单模态的检索模型算法不能直接用于跨模态的检索，因为不同的模态数据都嵌入于各自的异构特征空间之中、存在着语义鸿沟。如今国内外还没有非常成熟的跨模态检索技术，仍在探索研究。

近几年在跨模态检索领域【1】引起了很多专家学者强烈的关注并取得了阶段性进展，在这些研究的方法中主要有两大类。一类是以潜在子空间学习为基础的方法【2-5】，其中典型性相关分析 CCA【2】是最常用的模型。CCA 通过把两个模态的数据映射到潜在的子空间中并使得相关联的数据对相关性最大化，然后在子空间中直接进行相似度查询检索。基于相关数据在子空间中相关性最大化的思想，一些专家提出了类似 CCA 的其他变形模型算法。论文【3】通过加入标签信息使得多模态数据在子空间相关性最大化的同时也使得具有类别判别性（类间距离大，类内距离小），从而有利于进一步提高跨模态检索的准确率。文章【4】中先把每个模态的原始特征数据特征投影到各自语义特征空间，再用 CCA 或 KCCA 的方法把多模态的语义特征映射到统一的子空间。该模型利用了数据的标签信息来提高类别区分析，同时避免了直接把多模态原始特征映射到统一子空间中，因此改善了跨模态检索的效果。而论文【5】采用稀疏编码方法来学习统一的稀疏编码子空间，并且取得不错的效果。虽然基于统一子空间学习的跨模态检索算法取得一定的进展，但也存在着计算代价大、数据存储消耗高以及无法保证大规模多模态数据跨模态检索的稳定性。因此另一类基于哈希编码的跨模态检索算法越来越受到大家的关注和研究。

为了保持多模态数据之间的联系，论文【6-11】通过线性映射的方式将多模态数据投影到低维度的汉明空间，再异或操作来进行检索查询。因此基于哈希的跨模态检索方法有效地解决了大规模数据的存储问题以及提高了检索的速度和效率。但现有的方法基本上大多只适合单标签且成对训练数据的场景，论文【12】首次提出适应多个训练场景的哈希跨模态检索模型。但该模型和论文【6-11】类似，都是把多模态数据映射成等长的哈希编码，可能无法很好的表示各模态数据。此外求解二值哈希编码是个 NP 问题，现有的论文中基本上松弛对哈希编码的二值约束来求解，故学习到的哈希编码不够准确。针对以上问题，本文在论文【11、12】的研究基础上首次提出了基于变长哈希编码的跨模态检索模型，并且在求解哈希编码的过程

中加入了二值约束条件。因而学习到的变长哈希编码更能表达原始多模态的数据，以及更加准确。

本文主要工作如下：

（1） 首次提出了基于变长哈希编码的跨模态检索模型。本文把各模态数据投影到各自理想长度的哈希编码空间，因此比论文【6-12】把多模态数据投影到同一个长度的哈希编码空间的方法更加能够表示原始的多模态数据，并且本文模型在调试实验时更加灵活。

（2） 提出了比论文【12】更广义化的多场景跨模态检索。现有的跨模态检索模型大多基于单标签且成对的多模态数据集场景，不能用于多标签、非成对的多模态数据集场景。本文模型对单标签或多标签、成对或非成对的多模态数据集场景都具有很好的适应性。

（3） 在监督离散哈希论文【15】的基础上，提出了变长离散哈希编码跨模态检索的求解算法，并且在多个公开数据集上验证了该算法的有效性。

## 1 相关方法介绍

本小节中主要介绍几种基于哈希的跨模态检索模型。有关于其他方法的跨模态检索模型，有兴趣的读者可以参考跨模态检索方法的综述论文【1】。

Kumar 等人【】在单模态谱哈希方法的基础上首次提出了低纬度的多模态检索哈希编码算法，文中在每个模态的数据集上学习一个哈希函数，然后通过各个模态的哈希函数把对应的模态数据映射到哈希空间。文章目的就是通过哈希编码来计算多模态数据之间的相似性，希望哈希函数在所有模态数据上投影相似的数据到哈希空间后仍然相似。文中先计算原始数据之间的相似性矩阵，再通过优化带权重的哈希编码汉明距离函数，从而最终求解出理想的哈希编码。基本上这篇基于哈希函数学习的跨模态检索文章可以看做是单模态谱哈希文章【】的扩展。对于带有标签的数据，为了更好地提高检索的准确率，文章【】提出了基于语义相关最大化的哈希跨模态检索模型，同时比其他的监督哈希跨模态检索模型【】具有训练时间复杂度低的优点并且对于大于大规模的数据具有更好地适应性和稳定性。另外在语义保持哈希论文中【】首次提出了把数据的相似关联信息转换成概率分布的形式，然后通过最小化 KL 散度距离的方式近似求解哈希编码，整个目标函数模型在数学理论上得到了完美解释和有效保证。计算出统一哈希编码之后，下一步对哈希编码的每位属性学习核逻辑斯回归函数，从而完成多模态数据原始特征到哈希编码空间的非线性映射。

## 2 变长哈希编码跨模态检索

本节给出了变长哈希编码跨模态检索模型算法，并分析了该算法目标函数优化求解过程。为了方便描述算法模型和降低实验操作性，本文与其他文章【6-12】一样，以下主要研究两种模态数据的情形，同时 2.5 小节给出了扩展到三种或三种以上模态数据的算法模型。

### 2.1 算法模型

对于文中出现的变量定义如下，$X \in R^{d_1 \times n_1}, Y \in R^{d_2 \times n_2}$ 分别是两个模态的原始特征数据集以及 $B_X \in R^{q_1 \times n_1}$ 和 $B_Y \in R^{q_2 \times n_2}$ 是各自对应的变长哈希编码，其中每列表示一个样本、每行表示属性特征。此外 $P_X$、$P_Y$ 是投影矩阵，$W$ 是两个模态的关联矩阵，多模态数据之间的相似矩阵 $S \in R^{n_1 \times n_2}$ 主要有如下 3 种构造方式：

$$S_{ij} = \begin{cases} < l_x^i, l_y^j > & I \\ e^{-\|l_x^i - l_y^j\|^2 / \delta} & II \\ 1, if \ \ l_x^i = l_y^j; 0, if \ \ l_x^i \neq l_y^j & III \end{cases} \tag{1}$$

其中 $l$ 是样本的标签向量，相似矩阵的每个元素 $S_{ij}$ 表示 $X$ 模态的第 i 个数据与 $Y$ 模态的第 j 个数据的

相似度。接下来本文的目标就是学习各模态理想长度的紧凑哈希编码，使得这些哈希编码能够很好的表示原多模态的数据，而且能够保持多模态数据集的语义相似性。为了计算不同模态数据间的相似性，本文与文章【4】一样假设多模态数据之间存在共同的潜在抽象语义空间$V$，在这空间内多模态数据可以直接进行查询、检索。各模态哈希编码投影到潜在抽象语义空间的形式如下：

$$M_1 : B_X \xrightarrow{W_1} V_X \qquad M_2 : B_Y \xrightarrow{W_2} V_Y \tag{2}$$

则在 V 空间可根据内积关系计算数据间的相似性，故有：

$$\tilde{S} = V_X^T V_Y = (W_1 B_X)^T (W_2 B_Y) = B_X^T W_1^T W_2 B_Y \tag{3}$$

记$W = W_1^T W_2$，本文不需要显式地求解各模态数据在潜在抽象语义空间$V$空间存在形式，只需计算出$W$就可确定各模态变长的哈希编码之间的相似性。具体变长哈希编码跨模态检索的目标函数如公式（4）所示：

$$\begin{aligned} &\min_{B_X, B_Y, W, P_X, P_Y} \| B_X - P_X X \|_F^2 + \| B_Y - P_Y Y \|_F^2 + \| S - B_X^T W B_Y \|_F^2 \\ &s.t.\ \ B_X \in [-1,+1]^{q_1 \times n_1}, B_Y \in [-1,+1]^{q_2 \times n_2} \end{aligned} \tag{4}$$

目标函数前两项是把两个模态数据分别投影到各自理想长度的哈希编码空间，最后一项是在变长的哈希编码空间仍然保持原始多模态数据的语义相似度关系。通过最优化求解公式（4），相应的投影矩阵$P_X$、$P_Y$，哈希编码$B_X$、$B_Y$以及关联矩阵$W$都会被同时求解出。

**2.2 目标函数求解过程**

为了简化求解哈希编码的难度，文章【6-10】把哈希编码的二值约束条件转换成了求解连续的实值问题，然后通过符号函数获得了近似的哈希编码。但是这些求解出的哈希编码有着本质上缺陷以及不能充分有效地表示原始多模态数据，在本小节的求解过程中始终保持着哈希编码的二值约束条件。求解目标函数（4）时，对于同时求解$B_X$、$B_Y$,$W$,$P_X$,$P_Y$变量是非凸函数并且难以求解。因此本文通过先求解其中一个变量以及固定剩余的变量，然后如此求解其他变量、不断依次迭代所有变量，直至目标函数趋于收敛。

1）固定其他变量，求解$P_X$、$P_Y$。目标函数可简化成以下形式：

$$\begin{aligned} &\min_{P_X} \| B_X - P_X X \|_F^2 \\ &\min_{P_Y} \| B_Y - P_Y Y \|_F^2 \end{aligned} \tag{5}$$

因此$P_X$、$P_Y$可通过回归公式分别计算出：

$$\begin{aligned} P_X &= B_X X^T (X X^T)^{-1} \\ P_Y &= B_Y Y^T (Y Y^T)^{-1} \end{aligned} \tag{6}$$

2）固定其他变量，求解$W$。目标函数可简化成以下形式：

$$\min_W \| S - B_X^T W B_Y \|_F^2 \tag{7}$$

显然公式（6）类似双线性回归模型，计算公式如下：

$$W = (B_X B_X^T)^{-1} B_X S B_Y^T (B_Y B_Y^T)^{-1} \tag{8}$$

3) 固定其他变量，求解 $B_X$ 。目标函数可简化成以下形式:

$$\begin{aligned}&\min_{B_X} \| B_X - P_X X \|_F^2 + \| S - B_X^T W B_Y \|_F^2 \\ &s.t.\ B_X \in [-1,+1]^{q_1 \times n_1}\end{aligned} \tag{9}$$

因为有兼顾二值约束条件，显然直接求解起来非常复杂。因此本文对 $B_X$ 变量进行一行一行的求解，即求解 $B_X$ 中的某一行向量时先固定剩余的行向量，然后依次迭代求解其他行向量，对公式（9）进行展开可以变形为公式（10）的形式。

$$\begin{aligned}&\min_{B_X} \| B_X \|_F^2 - 2Tr(B_X^T P_X X) + \| P_X X \|_F^2 + \| S \|_F^2 \\ &\quad - 2Tr(B_X^T W B_Y S^T) + \| B_X^T W B_Y \|_F^2 \\ &s.t.\ B_X \in [-1,+1]^{q_1 \times n_1}\end{aligned} \tag{10}$$

因为有二值约束条件，显然第一项是个常数，即 $\| B_X \|_F^2 = q_1 * n_1$ 。除去其中的常数项及无关 $B_X$ 变量的项，则公式（10）可改写为更简洁的形式:

$$\begin{aligned}&\min_{B_X} \| DB_X \|_F^2 - 2Tr(B_X^T Q) \\ &s.t.\ B_X \in [-1,+1]^{q_1 \times n_1}\end{aligned} \tag{11}$$

其中 $D = B_Y^T W^T$ ， $Q = (WB_Y S^T + P_X X)$ ， $Tr(...)$ 为求解矩阵的迹。通过变形后，公式（11）的求解问题与文章【15】目标函数求解基本一样，因此本文参考其求解过程。当求解 $B_X$ 第 $i$ 行向量 $z^T$ 时，令 $B_X^{'}$ 为 $B_X$ 删除行向量 $z^T$ 后的矩阵， $p^T$ 为 $Q$ 的 $i$ 行向量以及 $Q'$ 为 $Q$ 删除行向量 $p^T$ 后的矩阵， $d$ 为 $D$ 的 $i$ 列向量以及 $D'$ 为 $D$ 删除列向量 $d$ 后的矩阵，则参照文章【15】的求解结果有:

$$z = sign(p - B_X^{'} D'^T d) \tag{12}$$

按照公式可求解 $B_X$ 的 $i$ 行向量，然后通过类似的步骤剩余的其他行向量。

4）固定其他变量，求解 $B_Y$ 。

在求解 $B_Y$ 的过程中基本上与求解 $B_X$ 类似，因此可以参考 $B_X$ 求解方式。

上述几个步骤分别给出了各个变量的求解表达式，然后通过循环迭代的方式使得目标函数趋于收敛，算法 1 描述了整个模型具体优化求解的详细过程。

算法 1：变长哈希编码跨模态检索

输入 ： 数据集 $X$、$Y$ 以及标签矩阵 $L_X$、 $L_Y$

初始化关联矩阵 $W$

初始化变长哈希编码 $B_X$、 $B_Y$

初始化迭代控制参数 $T$

0：利用标签矩阵 $L_X$、 $L_Y$ 和公式（1）来构造语义相似度矩阵 $S$

1：令 $iter = 0$;

2: *while iter* $< T$ *do*

3: 根据公式（6），更新字典投影矩阵 $P_X$、$P_Y$

4: 根据公式（8），更新关联矩阵 $W$

5: 根据公式（12）以及文章【15】的详细求解过程，对变长哈希编码进行每次只更新一行的方式，最终整体更新 $B_X$、$B_Y$

6: 若目标函数公式（4）趋于收敛则停止迭代，否则跳转到步骤（2）

7: *End while*

输出：变量 $B_X$、$P_X$、$B_Y$、$P_Y$、$W$

**2.3 生成查询样本的哈希编码**

按照 2.2 小节求解过程可以分别计算出每个模态的投影矩阵，然后通过符号函数即可求解出相应的哈希编码。对于查询样本 $x'$ 或 $y'$，则相应的哈希编码生成方法为：$b' = sign(P_X x')$ 或 $b' = sign(P_Y y')$。如果同时存在查询样本对 $(x', y')$ 并且为了提高生成对应哈希编码的准确性，可同时利用这两个模态的查询样本信息来生成哈希编码。若希望最终哈希编码存在于 $X$ 模态的哈希编码空间，则有 $b' = sign(P_X x' + \theta W P_Y y')$；若希望最终哈希编码存在于 $Y$ 模态的哈希编码空间，则有 $b' = sign(P_Y y' + \theta W^T P_X x')$，其中 $\theta$ 为非负的平衡参数。

**2.4 多个模态的情景**

本文提出的跨模态检索模型可以很方便的拓展到 3 种或 3 种以上的模态数据的情形，假设有 m（m>=3）个模态数据，则 m 个模态数据的变长哈希编码跨模态检索的模型如下：

$$\min_{B_i, W^{(i,j)}, P_i} \sum_{i=1}^{m} \| B_i - P_i X_i \|_F^2 + \sum_{i,j}^{m} \| S^{(i,j)} - B_i^T W^{(i,j)} B_j \|_F^2$$
$$s.t.\ B_i \in [-1, +1]^{q_i \times n_i} \tag{13}$$

公式中的首项表示把所有模态数据的映射成理想长度的哈希编码，第二项表示每个模态的哈希编码与其他模态哈希编码的语义关系保持。对于模型优化及生成查询样本哈希编码的过程可以按照 2 个模态数据情景的方式。

**3 实验与结果**

**3.1 数据集和评价标准**

为了验证本文模型的有效性，本文选择了跨模态检索常用的 WIKI 数据集、NUS-WIDE 数据集和 MIRFlickr 这 3 个公开的数据集。此外同论文【7、9、11】一样，用准确率召回率曲线图（precision-recall）、MAP（mean average precision）指标来衡量模型性能。

WIKI 数据集【4】是从维基百科网页整理而来，并且每幅图像都有与之对应的描述文本，其中每一篇文本不少于 70 个单词。数据集属于单标签数据，总共有 10 个类别，每个图像或文本只属于其中的某一类别，并且认为属于同一类的图像或文本具有相似的语义信息。2866 个样本（训练集 2173 个，测试集 693 个），其中图像由 128 维的 SIFT 特征表示、文本数据由 10 维的 LDA(latent Dirichlet allocation)特征表示。

NUS-WIDE 数据集【16】是由新加坡国立大学从互联网收集整理，其中包含 269648 张图像以及 5000 多人员进行解释性的标注。每个样本属于多标签数据，最终被分成 81 个类别。由于有些类别的样本数差异较大，本文同文章【6、7、8】一样，先筛选出样本较多的前 10 个类，最终总有 186577 个文本图像对。如果文本与图像至少有一个相同的类别属性，则两者认为相似。随后本文再随机选择 1%的数据（约 1866

个）作为测试集，5000 个样本作为训练集。NUS-WIDE 数据集的图像由 500 维的 SIFT 特征表示，文本由 1000 维的词频表示。

MIRFlickr 数据集来自 Flickr 网站，包含 25000 张图像和对应人工标注的文本信息。随后本文与文章【7】一样，删去了一些没有标签或标注中词汇少于 20 次的数据，最终有 16738 条样本并且被分成 24 个类别。每条图像文本对属于多类别数据，至少包含一个类别标签。本文选自 5%的数据作为测试集，5000 条样本作为训练集。数据集中的图像由 150 维度的边缘直方图表示，文本由 500 维的向量表示。

评判标准定义如下：

$$\text{准确率：}\ P(N)=\frac{n}{N}\times 100\% \qquad \text{召回率：}\ R(N)=\frac{n}{N_r}\times 100\%$$

其中 $n$ 是检索返回 $N$ 个结果中的相关样本个数，$N_r$ 是整个数据库中与查询样本相关的样本个数。

AP 指标计算：给定一个查询样本以及前 $R$ 个返回结果，则这个样本的 AP(average precision)计算公式如下，

$$AP=\frac{1}{K}\sum_{r=1}^{R}P(r)\delta(r)$$

其中 $K$ 是检索返回结果中与查询样本相关的个数，$P(r)$表示返回前 $r$ 个检索结果时的准确率。如果第 $r$ 个检索结果与查询样本相关，则 $\delta(r)$ 为 1，否则为 0。最后再求解全部查询样本的 AP 平均值，即得到评判整体搜索性能的 MAP 指标。

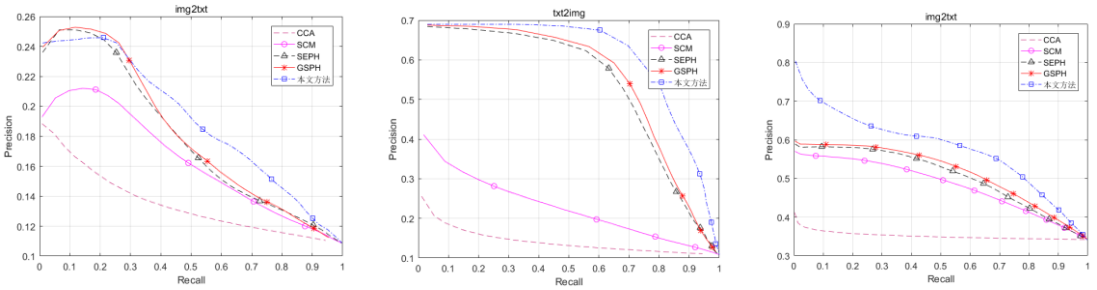
### 3.2 实验设置与对比方法分析

首先根据文章【11】的方式对各个多模态数据进行预处理，即计算样本点与随机选取参考点距离。本文随后用离散监督哈希模型【15】分别初始化各模态的哈希编码。在优化求解的过程中为了突出标签矩阵的重要性，因此对所有数据的标签矩阵放大了 10 倍。另外本文选取了在跨模态检索领域较为经典的典型性相关分析方法 CCA【2】以及近几年的基于语义相关哈希编码跨模态检索的算法作为对比试验。这些哈希跨模态检索模型分别是 SEPH【7】、SCM【8】、GSPH【12】，并且下文显示的对比实验都是按照原文设置的参数基础上运行其作者公开的 matlab 代码。在 SEPH 和 GSPH 模型中学习哈希函数有两种方式：（1）基于随机选取样本来训练哈希函数 SEPH_rnd、GSPH_rnd；（2）基于通过聚类选取样本来训练函数 SEPH_knn、GSPH_knn。原文实验部分显示出这两种训练方式得到的哈希函数在性能上基本相同。因此下文的对比试验选取了第一种方式来训练 SEPH、GSPH 模型的哈希函数。在 SCM 中有两种不同的求解方式得到的模型分别为 SCM_seq、SCM_orth，并且原文显示前者总体上比后者的性能优越，因此下文用前者作为对比实验。本文的所有实验都在个人电脑上完成，主要配置参数为：。

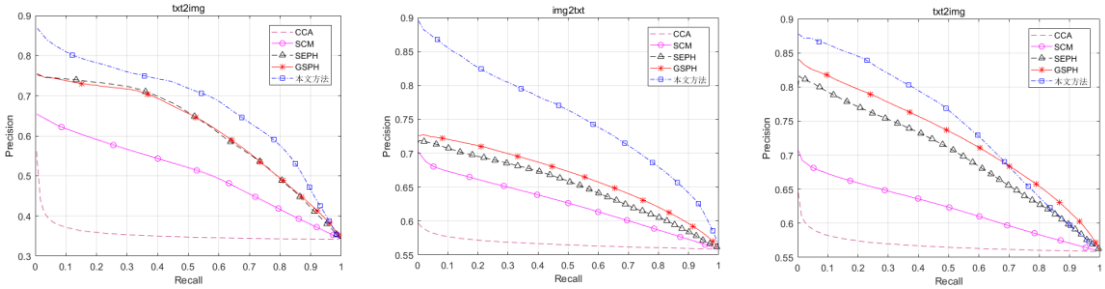
### 3.3 实验结果

本小节中展现了在 WIKI 数据集、NUS-WIDE 数据集以及 MIRFlickr 数据集上的跨模态检索实验结果。在下文的跨模态检索任务中包括图像检索文本和文本检索图像，并对这两种检索任务进行了详细的分析。

在图 2 中显示了三种数据集上的检索准确率与召回率曲线。为了方便与参照方法进行比较，本文与对比方法把图像和文本都投影到等长的哈希编码空间（64bits）。从图 2 中可以看出本文方法的性能总体上优于对比方法，虽然在 WIKI 数据集上图像检索文本任务中的曲线前部分(图 2 的子图(1))略微低于 SEPH、GSPH 方法，但是从图 3 的子图（1）可以看出本文最优哈希编码组合长度的效果略微高于 SEPH、GSPH 方法的效果。从图 2 还可以看出在另外两组多标签的数据上，本文效果比对照方法有了更大的提升，可能是因为本文模型比对照方法更适合多标签数据集的情景。



(1) (2) (3)




(4) (5) (6)

图 2　本文方法和对比方法的图像检索文本与文本检索图像的准确率与召回率曲线图，图中显示的 64bits 哈希长度的效果。(1)、(2)对应 WIKI 数据集，(3)、(4)对应 NUS-WIDE 数据集，(5)、(6)对应 MIRFlickr 数据集。

| 表 1　MAP：图像检索文本 img2txt | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | wiki 数据集 | | | | nus_wide 数据集 | | | | mirflickr25k 数据集 | | | |
| | 16 | 32 | 64 | 128 | 16 | 32 | 64 | 128 | 16 | 32 | 64 | 128 |
| CCA | 0.184 | 0.170 | 0.150 | 0.140 | 0.373 | 0.366 | 0.361 | 0.358 | 0.579 | 0.574 | 0.571 | 0.568 |
| SCM | 0.234 | 0.241 | 0.246 | 0.257 | | | 0.553 | | | | 0.647 | |
| SEPH | **0.276** | 0.296 | 0.300 | 0.313 | | | 0.582 | | | | 0.681 | |
| GSPH | 0.272 | 0.290 | 0.305 | 0.307 | 0.571 | 0.582 | 0.585 | 0.593 | 0.665 | 0.676 | 0.687 | 0.692 |
| 本文 | 0.271 | **0.368** | **0.351** | **0.369** | **0.627** | **0.632** | **0.644** | **0.656** | **0.766** | **0.772** | **0.778** | **0.779** |

| 表 2　MAP：文本检索图像 txt2img | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | wiki 数据集 | | | | nus_wide 数据集 | | | | mirflickr25k 数据集 | | | |
| | 16 | 32 | 64 | 128 | 16 | 32 | 64 | 128 | 16 | 32 | 64 | 128 |
| CCA | 0.168 | 0.159 | 0.154 | 0.150 | 0.371 | 0.365 | 0.362 | 0.360 | 0.579 | 0.574 | 0.572 | 0.570 |
| SCM | 0.226 | 0.246 | 0.249 | 0.253 | | | 0.542 | | | | 0.628 | |
| SEPH | 0.631 | 0.658 | 0.659 | 0.669 | | | 0.693 | | | | 0.727 | |
| GSPH | **0.645** | 0.663 | 0.671 | 0.674 | 0.681 | 0.697 | 0.686 | 0.714 | 0.726 | 0.742 | 0.748 | 0.764 |
| 本文 | 0.487 | **0.748** | **0.751** | **0.757** | **0.686** | **0.715** | **0.761** | **0.776** | **0.766** | **0.780** | **0787** | **0.791** |

在表 1、表 2 中分别给出了各个方法的图像检索文本和文本检索图像的 MAP 指标，并对每列最高的 MAP 值进行了标黑。为了方便比较 CCA 和其他方法效果，本文把数据投影到不同维度的子空间来观察对 CCA 方法的影响。表 1、表 2 显示了本文方法和其他哈希编码方法随着哈希编码长度增加，其 MAP 值总体上呈现略微的提升。此外从表中标黑的数值部分可以看出无论是在图像检索文本任务中还是文本图像的任务中，本文方法的 MAP 值整体上高于近几年的相关方法。当哈希编码为 64bits 时，本文比 GSPH 方法在 wiki、nus_wide 和 mirflickr25k 数据集上的图像检索文本任务中分别提高了约 15%、10%、13%，在文本检索图像任

务中分别提高了约 12%、11%、5%。

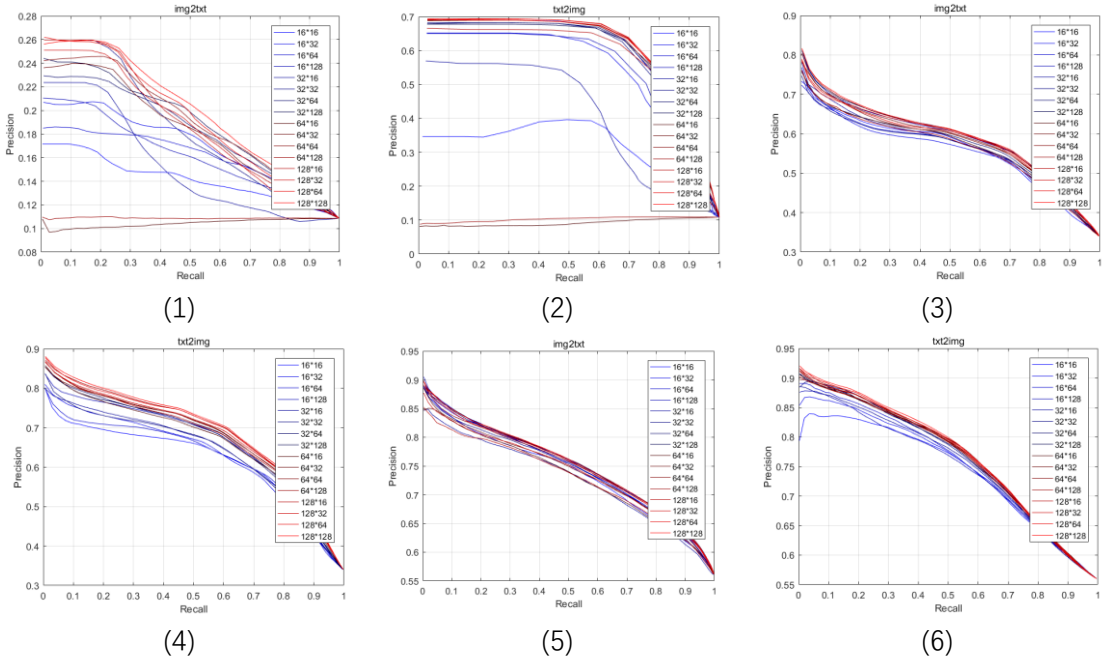
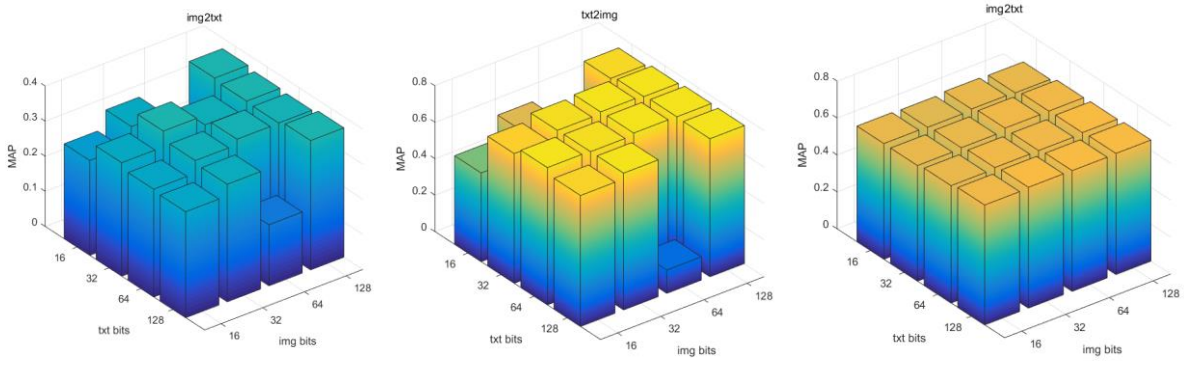



图 3　本文不同哈希编码长度组合的图像检索文本与文本检索图像的准确率与召回率曲线图，(1)、(2)对应 WIKI 数据集，(3)、(4)对应 NUS-WIDE 数据集，(5)、(6)对应 MIRFlickr 数据集。

在图 3 中显示了本文哈希编码不同长度组合(图像哈希编码长度*文本哈希编码长度)的实验效果。为了显示不同哈希长度组合的实验效果变化趋势，本文在图三中从 16*16 到 128*128 哈希编码长度组合的曲线颜色由深蓝、浅蓝、浅红再到深红逐渐变化。总体上随着图像哈希编码的增长，其跨模态检索效果也变好，特别是图 3 的子图(4)、(6)效果变好的趋势更加明显，另外图三也说明了本文的变长哈希编码跨模态检索模型对于 WIKI 数据集的影响更加显著。在图四中给出了不同哈希编码长度组合的 MAP 指标三维柱状图，子图的柱状图高低变化趋势总体与图 3 中对应的子图变化趋势保持一致，如图 3 子图(1)、(2)显示了 WIKI 数据集对不同哈希长度组合比较敏感，则在图 4 子图(1)、(2)柱状图也表现了对不同哈希长度组合的敏感性。



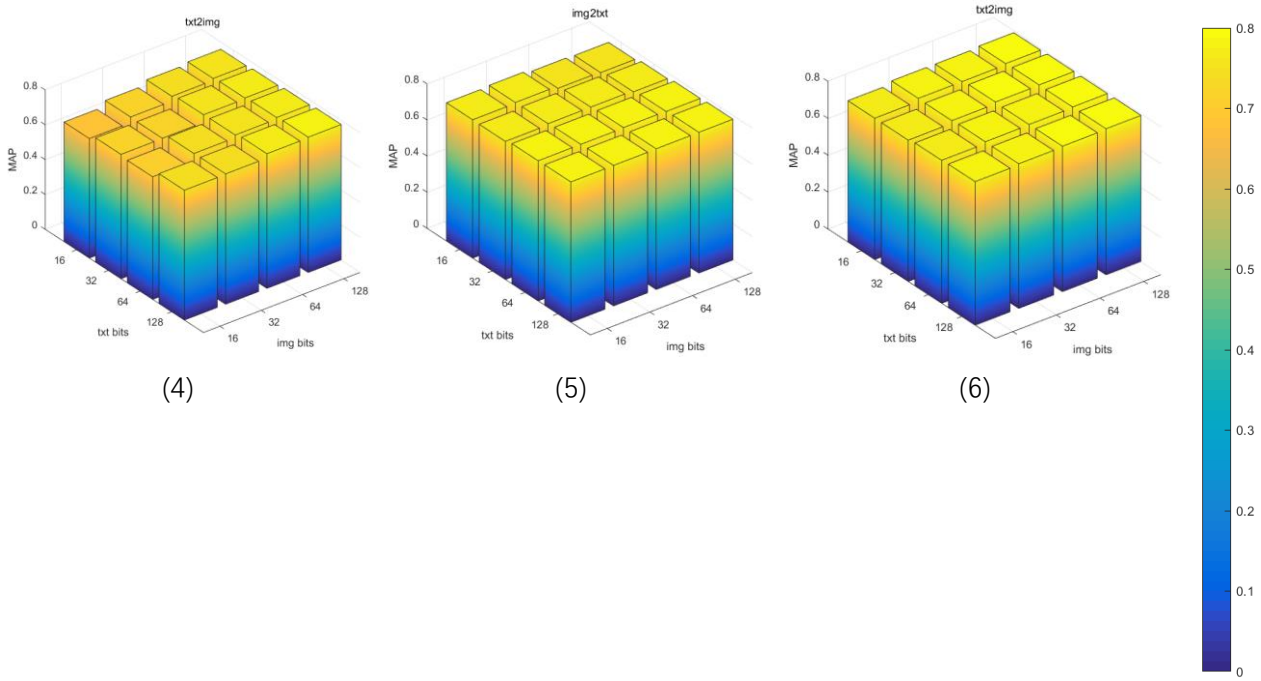



图 4　本文不同哈希编码长度组合的图像检索文本与文本检索图像的 MAP 指标三维柱状图，(1)、(2)对应 WIKI 数据集，(3)、(4)对应 NUS-WIDE 数据集，(5)、(6)对应 MIRFlickr 数据集。

## 4 结束语

本文首次提出了把多模态数据投影到各自模态数据的理想哈希长度空间的变长哈希编码跨模态检索算法。根据各模态的标签矩阵来构造多模态数据间的相似度矩阵，并保证把多模态哈希编码投影到潜在的抽象语义空间后仍拥有原始数据的语义相似度关系。随后在优化求解模型过程中始终保持对哈希编码的二值约束条件，使得学习到的多模态哈希编码更能表示原始的多模态数据。通过在 WIKI 数据集、NUS-WIDE 数据集以及 MIRFlickr 数据集上大量的实验，表明了本文方法的性能总体上优于近几年的相关对比方法。

参考文献:

[1] Wang K, Yin Q, Wang W, et al. A Comprehensive Survey on Cross-modal Retrieval[CP/OL]. 2017-09-08. https://arxiv.org/abs/1607.06215

[2] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, “Canonical correlation analysis: An overview with application to learning methods,” Neural Comput., vol. 16, no. 12, pp. 2639–2664, 2004.

[3] Sharma A, Kumar A, Daume H, et al. Generalized Multiview Analysis: A discriminative latent space[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2012:2160-2167.

[4] Pereira J C, Coviello E, Doyle G, et al. On the role of correlation and abstraction in cross-modal multimedia retrieval[J]. IEEE transactions on pattern analysis and machine intelligence, 2014, 36(3): 521-535.

[5] D. Mandal and S. Biswas, “Generalized coupled dictionary learning approach with applications to cross-modal matching,” IEEE Trans. Image Process., vol. 25, no. 8, pp. 3826–3837, Aug. 2016.

[6] J. Zhou, G. Ding, and Y. Guo, “Latent semantic sparse hashing for cross-modal similarity search,” in Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., 2014, pp. 415–424.

[7] Z. Lin, G. Ding, M. Hu, and J. Wang, “Semantics-preserving hashing for cross-view retrieval,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2015, pp. 3864–3872.

[8] D. Zhang and W.-J. Li, “Large-scale supervised multimodal hashing with semantic correlation maximization,” in Proc. AAAI Conf. Artif. Intell., 2014, pp. 2177–2183.

[9] Wang D, Gao X, Wang X, et al. Multimodal Discriminative Binary Embedding for Large-Scale Cross-Modal Retrieval[J]. IEEE Transactions on Image Processing, 2016, 25(10):4540.

[10] Irie G, Arai H, Taniguchi Y. Alternating Co-Quantization for Cross-Modal Hashing[C]// IEEE International Conference on Computer Vision. IEEE, 2015:1886-1894.
[11] Xu X, Shen F, Shen H T, et al. Learning Discriminative Binary Codes for Large-scale Cross-modal Retrieval[J]. IEEE Transactions on Image Processing, 2017, PP(99):1-1.
[12] Mandal D, Chaudhury K N, Biswas S. Generalized Semantic Preserving Hashing for N-Label Cross-Modal Retrieval[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2017:2633-2641.
[13] Shen F, Zhou X, Yang Y, et al. A Fast Optimization Method for General Binary Code Learning[J]. IEEE Transactions on Image Processing, 2016, 25(12):5610-5621.
[14] Jie G, Liu T, Sun Z, et al. Fast supervised discrete hashing[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, PP(99):1-1.
[15] F. Shen, C. Shen, W. Liu, and H. T. Shen, “Supervised discrete hashing,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2015, pp. 37–45.
[16] Chua T S, Tang J, Hong R, et al. NUS-WIDE: a real-world web image database from National University of Singapore[C]// ACM International Conference on Image and Video Retrieval. ACM, 2009:48.