

基于矩阵分解和相似性保持的跨模态检索研究

张心文

广东工业大学自动化学院, 广东 广州

收稿日期: 2023年5月22日; 录用日期: 2023年6月21日; 发布日期: 2023年6月28日

摘要

早先的基于哈希的跨模态检索方法因为语义提取以及运行速度慢不适合于大数据场景。因此提出一种新的框架叫做独特相似哈希(Unique Similar Hashing, USH)。USH是一个两步学习的哈希方法, 先学习哈希码再学习哈希函数。第一阶段, 用核函数将数据非线性地投影到核空间, 然后使用矩阵分解学习潜在空间。哈希码从潜在空间中学习而来, 为了避免量化误差并不放松哈希码的离散约束, 而是直接计算它的封闭解。在学习一个优质的哈希码之后, 再学习一个哈希函数将原始样本映射到低维的汉明空间。在Wiki数据集上与最先进的方法进行验证, USH在mAP上取得较好结果, 证明了该方法的有效性。

关键词

哈希方法, 跨模态检索, 矩阵分解

Research on Cross-Modal Retrieval Based on Matrix Factorization and Similarity Preservation

Xinwen Zhang

School of Automation, Guangdong University of Technology, Guangzhou Guangdong

Received: May 22nd, 2023; accepted: Jun. 21st, 2023; published: Jun. 28th, 2023

Abstract

Earlier hash-based cross-modal retrieval methods were not suitable for big data scenarios due to problems with semantic extraction and slow running speed. Therefore, a new framework called Unique Similar Hashing (USH) was proposed. USH is a two-stage learning-based hashing method that learns hash codes first and then hash functions. In the first stage, data is nonlinearly projected to a kernel space using kernel functions, followed by learning a latent space using matrix factorization. Hash codes are then learned from the latent space by computing their closed-form solution

directly instead of relaxing the discrete constraints to avoid quantization errors. After learning high-quality hash codes, a hash function is learned to map original samples to a low-dimensional Hamming space. It's validated on the Wiki dataset against state-of-the-art methods, USH achieved good results in mAP, demonstrating the effectiveness of this approach.

Keywords

Hashing Method, Cross-Modal Retrieval, Matrix Factorization

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着互联网的发展与电子设备的普及,数据信息成几倍的速度增加。在信息爆炸的同时,信息的种类也越来越多越样。单模态数据被定义为以单一的某种格式表示的数据,比如图像、视频、3D模型,多模态数据则是多种数据结构的汇总。单模态间的检索方式逐渐发展开来,人们可以用以图搜图等一些方式来进行检索。而在多种数据类型之间的检索,也称为跨模态检索。

跨模态检索的发展需解决以下问题。首先是模态的异构性,不同模态对于同种东西的语义传达方式是有很大差异的,例如图片的特征主要表示为尺度不变特征转换特征或者局部二值模式特征。文字的特征主要表现为线性判别分析特征或者词频-逆向文件频率特征。即便描述同一个东西,不同模态的维度依旧有大幅度的不同,无法简单直接地构建传达某一东西的不同数据的桥梁。其次是语义鸿沟,计算机通过数据的潜在表示来捕获信息,接着计算不同特征的距离以判断是否相似,而人类是通过经历与学习的高级语义来判断两种东西是否相似。最后是性能,数据量大种类多,如何快速对不同模态进行检索,以及检索的精确度也是我们需要考虑的。

而基于哈希的跨模态检索方法在近年来得到很大发展。哈希方法将数据压缩成一个二进制哈希码,通过简单的异或运算,其运行速度很快适用于大数据背景。根据语义标签的嵌入情况,我们将现有的跨模态哈希方法分为三个大类:无监督、有监督以及半监督模型。

2. 相关理论

2.1. 无监督哈希方法

集体矩阵分解哈希(Collective Matrix Factorization Hashing, CMFH) [1]设计了矩阵分解的框架来提取不同模态的共同子空间,同时把高纬度的数据压缩到低维度的特征,最后从子空间中学习到离散的二进制码。但是该方法并没有考虑样本与样本的相似性。隐藏语义稀疏哈希(Latent Semantic Sparse Hashing, LSSH) [2]联合了矩阵分解以及稀疏学习来降低原始样本的维度,并学习到不同模态的不同潜在空间,通过对不同的空间进行量化可以获取统一的哈希码,但LSSH并不是一个高效的算法,其原因在于稀疏学习的计算量相比与线性的模型来说是很大的,所以训练的成本很高。

2.2. 监督哈希方法

语义关联最大哈希(Semantic Correlation Maximization, SCM) [3]保证同成分的样本之间的关联性尽可能地高,并且将其嵌入到哈希码的学习中。但该模型由于松弛了约束条件,因此避免不了量化误差,同时训练成本比较沉重。

2.3. 半监督哈希方法

判别半监督哈希(semisupervised cross modal hashing approach by generative adversarial network, SCH-GAN) [4]利用了 GAN 技术来生成哈希码, 同时采纳一些非中央样本以提升模型能力。半监督方法没有要求全部样本具有标签, 并使用部分标签与模态内部的特征相结合, 因此能获取高质量的哈希码。所以半监督办法在表现性能跟标注时间上取了较棒的调节。

3. 方法提出

大多数方法学习一个多种模态学习一个低维的潜在空间, 尽管可以提取部分的共同语义, 但会损失每种模态的独特性。其次, 一些方法会先松弛离散的约束再去优化它, 这样会引入较大的量化误差, 从而导致精度的缺失。为了解决这些问题, 提出一种新的框架叫做独特相似哈希(Unique Similar Hashing, USH)。USH 是一个两步学习的哈希方法, 其首先学习哈希码然后学习哈希函数。第一阶段, 先使用核函数将数据非线性地投影到核空间, 然后使用矩阵分解学习潜在空间。然后为不同的模态设计了各自的潜在空间以保留它们的独特性。此外, 两个潜在空间被对齐通过局部相似性保留, 这样可以提取不同模态的高级语义。潜在空间中学习哈希码, 且不放松哈希码的离散约束来降低量化误差, 并直接计算它的封闭解。在学习一个优质的哈希码之后, 再学习一个哈希函数将原始样本映射到低维的汉明空间。为了平衡精度与速度, 一个线性的分类器被选择作为哈希函数, USH 的框架如图 1 所示。

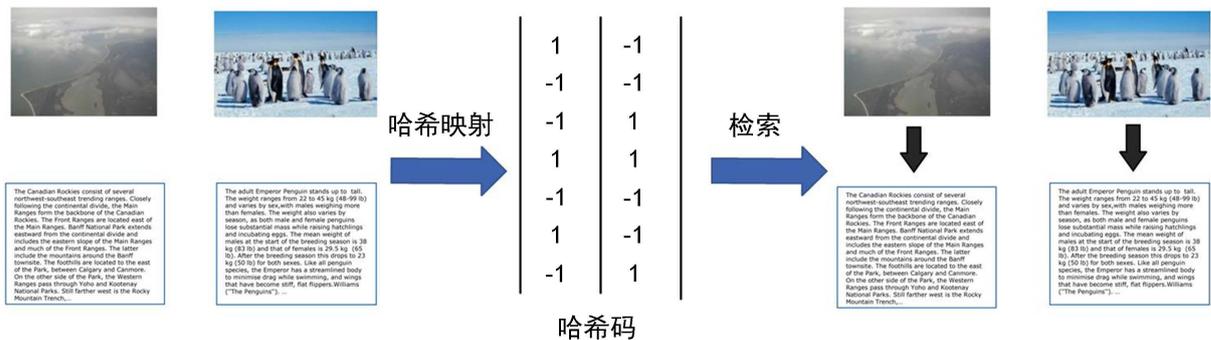


Figure 1. Schematic diagram of cross-modal hashing
图 1. 跨模态哈希示意图

3.1. 符号表示

首先用 X 表示一个多模态数据集。第 t 个数据表示成 $X_t \in \mathbb{R}^{m_t \times n}$ 。当 m 为 1 代表了图像, 当 m 为 2 代表文本, n 代表样本数量。为了增加模型的非线性, 使用核函数把 X_t 映射到核空间并获得 $\tilde{X}_t \in \mathbb{R}^{k_t \times n}$ 。由于这是一个监督算法, 因此会使用标签信息 $L \in \{0,1\}^{c \times n}$, 其中 c 代表了标签的总类别。目标是产生一个离散的哈希码, 将它表示成 $B \in \{-1,1\}^{r \times n}$, r 代表了哈希码长度, 即储存成本。最后给定一个矩阵 Z , 它的 Frobenius 范数及迹函数被分别表示为 $\|Z\|_F = \left(\sum_{ij} |z_{ij}|^2\right)^{1/2}$ 和 $Tr(Z) = \sum_i z_{ii}$ 。

3.2. 哈希码学习

原始数据可能包含一定的噪声, 因此需要学习一个样本的高级表示, 即潜在空间。集体矩阵分解时可以去掉数据的噪声, 可以利用它来学习一个共同的潜在空间以储存不同模态的高级语义。但是这样是不合理的, 因为每种模态的分布跟维度是相当不同的。直接设计一个共同空间将会散失每种模态的独特性。使用矩阵分解为每种模态学习一个潜在空间, 其被描述如下:

$$\min_{U_1, V_1, U_2, V_2} \|\tilde{X}_1 - U_1 V_1\|_F^2 + \alpha \|\tilde{X}_2 - U_2 V_2\|_F^2, \quad (1)$$

其中 α 是一个平衡参数, $U_1 \in \mathbb{R}^{k_1 \times r}$ 跟 $U_2 \in \mathbb{R}^{k_2 \times r}$ 为不同模态的投影矩阵, 而 $V_1 \in \mathbb{R}^{r \times n}$ 跟 $V_2 \in \mathbb{R}^{r \times n}$ 代表不同模态的潜在空间, 这四个变量都需要学习。潜在空间的维度被设置为哈希码的长度可以让哈希码的学习更加简单灵活。

在学习不同的潜在子空间之后, 需要探索它们共同的高级语义。通过全局相似直接令它们对齐虽然是简单的, 但是没有考虑样本与样本之间的相关性。为解决此问题, 提出局部相似性保留来对齐两个潜在空间。具体而言, 首先根据标签计算一个距离矩阵 H , 每个元素被定义如下:

$$h_{ij} = l_i^T l_j. \quad (2)$$

根据 H 定义一个相似性矩阵 S , 如果两个样本共享同一标签, 那它们相似性大的, 相反, 如果它们标签不相同, 则没有相似性。相似性矩阵 S 被计算如下:

$$S_{ij} = \begin{cases} 1 & h_{ij} \geq 1 \\ 0 & h_{ij} < 1 \end{cases}, \quad (3)$$

接着, 考虑潜在空间之间的局部相似性。具体而言, 如果两个样本是相似的, 那么它们的潜在在样本也应该是近似的。如果相似性很小, 那么潜在在样本则不要求近似, 这也可以看作它们每个空间的独特性。局部相似性保留通过下面的式子实现:

$$\min \beta \sum_{i=1}^n \sum_{j=1}^n \|(V_1)_i - (V_2)_j\|_2^2 S_{ij}, \quad (4)$$

其中 β 是一个平衡参数。通过优化公式(4)可以提取不同模态的共同高级语义并且保留它们各自的独特性。

当前目标是生成一个离散的二值哈希码 $B \in \{-1, 1\}^{r \times n}$, 以便于使用异或操作进行快速地检索。哈希码需要从样本的潜在空间进行学习因为潜在空间包含了高级语义。为了增加哈希码跟潜在空间的语义联系, 提出下面的损失项来学习哈希码:

$$\min \theta \left(\|B - P_1 V_1\|_F^2 + \|B - P_2 V_2\|_F^2 \right), \text{ s.t. } B \in \{-1, 1\}^{r \times n}. \quad (5)$$

其中 θ 是另一个平衡参数。 $P_1 \in \mathbb{R}^{r \times r}$ 跟 $P_2 \in \mathbb{R}^{r \times r}$ 是需要学习的映射矩阵。

最后, 把公式(1), 公式(4), 公式(5)结合起来, 便可以得到第一阶段的目标函数:

$$\begin{aligned} & \min_{U_1, V_1, U_2, V_2, P_1, P_2, B} \|\tilde{X}_1 - U_1 V_1\|_F^2 + \alpha \|\tilde{X}_2 - U_2 V_2\|_F^2 + \beta \sum_{i=1}^n \sum_{j=1}^n \|(V_1)_i - (V_2)_j\|_2^2 S_{ij} \\ & + \theta \left(\|B - P_1 V_1\|_F^2 + \|B - P_2 V_2\|_F^2 \right) \\ & \text{s.t. } B \in \{-1, 1\}^{r \times n}. \end{aligned} \quad (6)$$

3.3. 优化策略

目标函数公式(6)是一个非凸的函数所以难以直接优化。但如果只优化其中一个变量而固定其他的, 那么公式(6)将会变成凸的, 所以把公式(6)转化为矩阵迹的形式:

$$\begin{aligned} L = & \text{Tr} \left(\tilde{X}_1 \tilde{X}_1^T - 2 \tilde{X}_1 V_1^T U_1^T + U_1 V_1 V_1^T U_1^T \right) + \alpha \text{Tr} \left(\tilde{X}_2 \tilde{X}_2^T - 2 \tilde{X}_2 V_2^T U_2^T + U_2 V_2 V_2^T U_2^T \right) \\ & + \beta \text{Tr} \left(V_1 D V_1^T - 2 V_1 S V_2^T + V_2 D V_2^T \right) n + \theta \text{Tr} \left(B B^T - 2 B V_1^T P_1^T + P_1 V_1 V_1^T P_1^T \right) n \\ & + \theta \text{Tr} \left(B B^T - 2 B V_2^T P_2^T + P_2 V_2 V_2^T P_2^T \right) \\ & \text{s.t. } B \in \{-1, 1\}^{r \times n}. \end{aligned} \quad (7)$$

1) 更新 U 。

在固定 $V_1, U_2, V_2, P_1, P_2, B$ 之后, 关于 U_1 的目标函数被简化如下:

$$L(U_1) = \text{Tr}(-2\tilde{X}_1 V_1^T U_1^T + U_1 V_1 V_1^T U_1^T). \quad (8)$$

接着计算关于 U_1 的导数, 让其等于 0 可以得到 U_1 的封闭解, 同理 U_2 也可以得到封闭解。

$$U_1 = \tilde{X}_1 V_1^T (V_1 V_1^T)^{-1}. \quad (9)$$

$$U_2 = \tilde{X}_2 V_2^T (V_2 V_2^T)^{-1}. \quad (10)$$

2) 更新 V 。

固定 $U_1, U_2, V_2, P_1, P_2, B$ 之后, 公式(7)被转化如下:

$$\begin{aligned} L(V_1) = & \text{Tr}(-2\tilde{X}_1 V_1^T U_1^T + U_1 V_1 V_1^T U_1^T) + \beta \text{Tr}(-2V_1 S V_2^T + V_2 D V_2^T) \\ & + \theta \text{Tr}(-2B V_1^T P_1^T + P_1 V_1 V_1^T P_1^T). \end{aligned} \quad (11)$$

计算公式(7)对于 V_2 与 V_1 的导数令其等于 0 可得 V_1 与 V_2 的封闭解:

$$V_1 = \text{lyap}(E_1, F_1, G_1). \quad (12)$$

$$V_2 = \text{lyap}(E_2, F_2, G_2). \quad (13)$$

3) 更新 P 。

为了更新 P_1 令其他参数保持不变得到如下目标函数:

$$L(P_1) = \text{Tr}(-2B V_1^T P_1^T + P_1 V_1 V_1^T P_1^T). \quad (14)$$

计算公式(14)的导数并且将其设置为 0, 将得到 P_1 的最优解, 同理可得 P_2 最优解:

$$P_1 = B V_1^T (V_1 V_1^T)^{-1}. \quad (15)$$

$$P_2 = B V_2^T (V_2 V_2^T)^{-1}. \quad (16)$$

4) 更新 B 。

固定 $U_1, V_1, U_2, V_2, P_1, P_2$ 以更新 B , 考虑到 $\text{Tr}(BB^T) = rm$ 是一个常数, 因此简化目标函数如下:

$$\begin{aligned} L(B) = & \text{Tr}(BB^T - 2B V_1^T P_1^T + BB^T - 2B V_2^T P_2^T) \\ = & \text{Tr}(-2B(V_1^T P_1^T + V_2^T P_2^T)) \\ \text{s.t. } & B \in \{-1, 1\}^{r \times n}. \end{aligned} \quad (17)$$

由此可得 B 的最优解:

$$B = \text{sign}(P_1 V_1 + P_2 V_2) \quad (18)$$

3.4. 哈希函数学

USH 是一个两阶段的哈希方法, 将哈希码跟哈希函数的学习分离开。所以在学习完一个优质的哈希码之后, 需要设计一个哈希函数以将原始样本映射到低维的汉明空间。为了平衡速度跟精度, 采用核函数加上线性分类器的方式作为哈希函数。线性分类器的描述如下:

$$\min_{Q_t} \|B - Q_t \tilde{X}_t\|_F^2 + \lambda \|Q_t\|^2, \quad (19)$$

其中 Q_t 为第 t 个模态的分类器，而 $\lambda = 0.01$ 是根据经验所得的参数。优化公式(19)之后，给定一个新的检索样本，可以使用 Q_t 得到哈希码，然后通过异或操作比对不同的哈希码从而快速实现检索。为了优化公式(19)，计算其导数并令其为 0 可得 Q_t 的最优解：

$$Q_t = B \tilde{X}_t^T (X_t X_t^T + \lambda I_{k_t})^{-1}. \quad (20)$$

最后提供整个优化的步骤在 [算法 1](#) 中：

Algorithm 1. Unique similarity hashing (USH)

算法 1. 独特相似哈希(USH)

Algorithm 1 独特相似哈希(USH)

输入: 数据集 $\{X_t\}_{t=1}^2$ ，标签信息 L ，哈希码长度 r 以及三个参数 α, β, θ 。

输出: 哈希码 B 跟哈希函数 Q_t 。

1. 根据公式(9)将 X_t 非线性地映射到核空间 \tilde{X}_t ；

2. 随机初始化 $U_1, V_1, U_2, V_2, P_1, P_2, B$ ；

重复

//第一阶段：学习哈希码

1. 根据公式(9)更新 U_1 ；//学习图像映射矩阵 U_1 最优解

2. 根据公式(10)更新 U_2 ；//学习映射矩阵 U_2 最优解

3. 根据公式(12)更新 V_1 ；//由 lyap 函数学习图像潜在空间 V_1 最优解

4. 根据公式(13)更新 V_2 ；//由 lyap 函数学习文本潜在空间 V_2 最优解

5. 根据公式(15)更新 P_1 ；//学习图像映射矩阵 P_1 最优解

6. 根据公式(16)更新 P_2 ；//学习映射矩阵 P_2 最优解

7. 根据公式(18)更新 B ；//学习统一哈希码最优解

直到最大迭代次数；

//第二阶段：学习哈希函数

根据公式(x)学习哈希函数 Q_t ；

返回哈希码 B 跟哈希函数 Q_t ；

4. 实验论证

4.1. 数据集

Wiki [5]收集并分类了维基百科中的数千篇文章及其附带的图片，由于有些类别非常稀少，因此只保留了 10 个类别的样本。Wiki 提供 2866 个图像 - 文本对，它们被随机分成 2173 个文档作为训练集，693 个文档用作测试集。此外，在该数据集中，图像被表示为 128 维尺度不变特征变换(SIFT)描述符，而文本是从潜在狄利克雷分配(LDA)中提取的 10 维频率特征。

4.2. 实验设置

将所提出的 USH 与其他先进方法进行对比。他们分为两种类型，首先是无监督的哈希方法包括 LSSH

[2]以及 CMFH [1]。还有一些有监督的哈希方法, 包括 SMFH [6], SCM-seq [3], SePH [7], DCH [8], GSPH [9], MSLF [10]。所有 mAP 结果来源于原论文, 如果没有在某个数据集上验证, 那使用别人复现的结果, PR 曲线同理。因为 USH 快速收敛, 所以训练迭代次数设为 10。最后, 实验都是在一台个人计算机上实施的, 该计算机包括了 16 核 CPU 与 64 GB 内存。

4.3. 实验结果

实验结果分两方面, 首先是 mAP 结果, 在完成了两种任务在检索上, 分别是图像检索文本以及文本检索图像。并测试不同哈希码长度下各个方法的表现。mAP 的结果展示在表 1 中, 最好的表现会被标出。绘制 PR 曲线, 这些曲线是在哈希码长度为 32 位跟 64 位下的结果, 但对于其他长度情况是近似的。展现的是 PR 曲线如图 2 所示。

Table 1. mAP results on Wiki database

表 1. Wiki 数据集上的 mAP 结果

任务	方法	16 位	32 位	64 位	128 位
图像 检索 文本	LSSH	0.2330	0.2340	0.2387	0.2340
	CMFH	0.2538	0.2582	0.2619	0.2648
	SMFH	0.2276	0.2516	0.2581	0.2496
	SCM-seq	0.2210	0.2337	0.2442	0.2596
	SePH	0.2787	0.2965	0.3064	0.3134
	DCH	0.3317	0.3686	0.3762	0.3748
	GSPH	0.2900	0.3100	0.3200	0.3260
	MSLF	0.3324	0.3639	0.3751	0.3995
	USH	0.3636	0.3730	0.3833	0.3934
文本 检索 图像	LSSH	0.5571	0.5743	0.5710	0.5577
	CMFH	0.6116	0.6298	0.6398	0.6477
	SMFH	0.5242	0.6039	0.6602	0.6658
	SCM-seq	0.2134	0.2366	0.2479	0.2573
	SePH	0.6318	0.6577	0.6646	0.6709
	DCH	0.7006	0.7087	0.7241	0.7093
	GSPH	0.6460	0.6680	0.6770	0.6810
	MSLF	0.7211	0.7468	0.7467	0.7595
	USH	0.7202	0.7547	0.7640	0.7564

对结果进行观察, 发现在 16 位哈希码情况下, USH 在图像检索文本上取得最佳的精度, 在文本检索图像稍微落后于 MSLF 方法。而在 32 位哈希码以及 64 哈希码长度下, USH 无论在哪种情况下都超越了其他方法。128 位长度的哈希码下 USH 比 MSLF 稍低一些, 但仍然比其他方法要好一些。

观察图 2 的 PR 曲线, 发现无论哪种哈希码长度或者哪种任务表现上, USH 的曲线都在其他曲线之上。说明 USH 都超越了其他方法。这都与表 1 的结果一致, 以上评估标准都表明了 USH 是有效的。

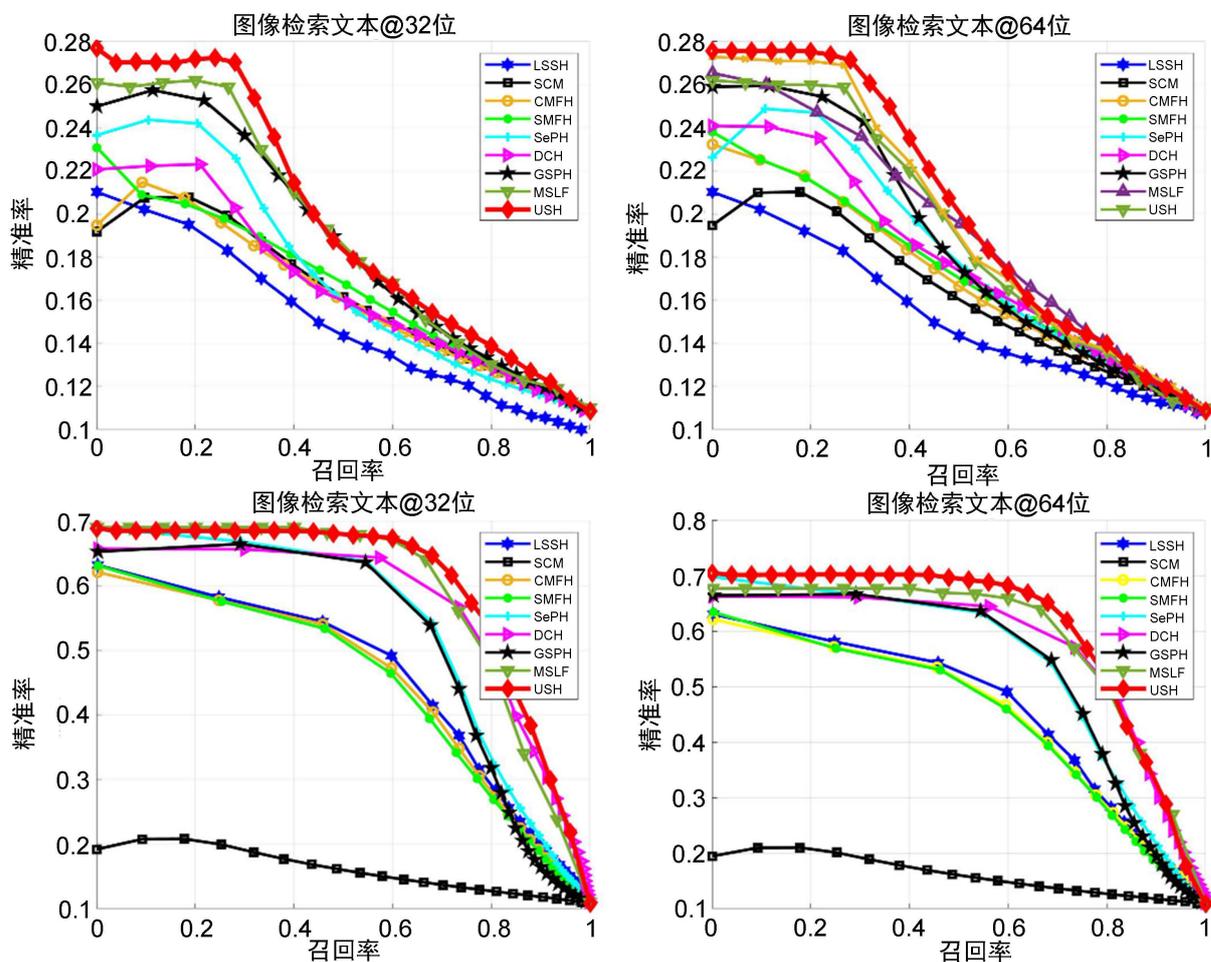


Figure 2. Precision-Recall curves on the Wiki
图 2. Wiki 上的 PR 曲线

5. 总结

本文提出独特相似哈希(USH)的新框架, 来快速实现多模态检索。USH 是一个两阶段学习框架, 它先学习哈希码再学习哈希函数。在第一阶段中, 使用矩阵分解为不同模态学习一个独特的潜在空间, 这可以保存不同模态的独特性。接着不同模态的潜在空间通过局部相似性保留对齐, 这样可以提取共同的高级语义的同时不抛弃模态独特性, 最后从潜在空间中学习哈希码。学习完一个优质的哈希码, 使用一个线性的分类器作为哈希函数, 其可以将新的样本映射到低维的汉明空间。为了证明 USH 方法是行之有效的, 在 Wiki 数据集上进行实验对比, 最后与其他方法比证明了 USH 的方法是有优势的。

参考文献

- [1] Ding, G., Guo, Y. and Zhou, J. (2014) Collective Matrix Factorization Hashing for Multimodal Data. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 23-28 June 2014, 2075-2082. <https://doi.org/10.1109/CVPR.2014.267>
- [2] Zhou, J., Ding, G. and Guo, Y. (2014) Latent Semantic Sparse Hashing for Cross-Modal Similarity Search. *Proceedings of the 37th International ACM SIGIR Conference on Research*, Gold Coast, 6-11 July 2014, 415-424.
- [3] Zhang, D. and Li, W.J. (2014) Large-Scale Supervised Multimodal Hashing with Semantic Correlation Maximization. *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, Quebec, 27-31 July 2014, 2177-2183.

-
- [4] Zhang, J., Peng, Y. and Yuan, M. (2018) SCH-GAN: Semi-Supervised Cross-Modal Hashing by Generative Adversarial Network. *IEEE Transactions on Cybernetics*, **50**, 489-502. <https://doi.org/10.1109/TCYB.2018.2868826>
 - [5] Rasiwasia, N., Costa Pereira, J., Coviello, E., *et al.* (2010) A New Approach to Cross-Modal Multimedia Retrieval. *Proceedings of the 18th ACM International Conference on Multimedia*, Firenze, 25-29 October 2010, 251-260. <https://doi.org/10.1145/1873951.1873987>
 - [6] Liu, H., Ji, R., Wu, Y., *et al.* (2016) Supervised Matrix Factorization for Cross-Modality Hashing. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, New York, 9-15 July 2016, 1767-1773.
 - [7] Lin, Z., Ding, G., Hu, M., *et al.* (2015) Semantics-Preserving Hashing for Cross-View Retrieval. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 7-12 June 2015, 3864-3872. <https://doi.org/10.1109/CVPR.2015.7299011>
 - [8] Xu, X., Shen, F., Yang, Y., *et al.* (2017) Learning Discriminative Binary Codes for Large-Scale Cross-Modal Retrieval. *IEEE Transactions on Image Processing*, **26**, 2494-2507. <https://doi.org/10.1109/TIP.2017.2676345>
 - [9] Mandal, D., Chaudhury, K.N. and Biswas, S. (2018) Generalized Semantic Preserving Hashing for Cross-Modal Retrieval. *IEEE Transactions on Image Processing*, **28**, 102-112. <https://doi.org/10.1109/TIP.2018.2863040>
 - [10] Wang, S., Zhao, H. and Nai, K. (2021) Learning a Maximized Shared Latent Factor for Cross-Modal Hashing. *Knowledge-Based Systems*, **228**, Article ID: 107252. <https://doi.org/10.1016/j.knosys.2021.107252>