

# 基于标签一致性哈希的跨模态检索算法

刘志虎

广东工业大学计算机学院, 广东 广州  
Email: lzh9656@163.com

收稿日期: 2021年3月28日; 录用日期: 2021年4月21日; 发布日期: 2021年4月28日

---

## 摘要

针对跨模态检索任务中, 不同数据之间存在异构性以及语义鸿沟等特点, 本文提出了一种新的监督哈希方法。该方法利用矩阵分解学习训练数据集在低维潜在语义空间表示, 同时本文将标签信息也视为一个单独的模态, 也利用矩阵分解将其映射到低维潜在语义子空间中; 然后, 在子空间中最大化它们之间的相关性, 从而得到相应的低维潜在语义代表; 之后, 本文利用正交旋转矩阵学习性能更好的哈希函数得到相应的哈希码。在三个常用的数据集Wiki, MIRFlick和NUS-WIDE进行了大量的实验, 并与一些常用的跨模态哈希方法进行了比较, 结果证明了该算法的优越性。

## 关键词

跨模态检索, 哈希, 矩阵分解

---

# Label Consistency Hashing for Cross-Modal Retrieval

Zhihu Liu

School of Computers, Guangdong University of Technology, Guangzhou Guangdong  
Email: lzh9656@163.com

Received: Mar. 28<sup>th</sup>, 2021; accepted: Apr. 21<sup>st</sup>, 2021; published: Apr. 28<sup>th</sup>, 2021

---

## Abstract

In view of the heterogeneity and semantic gap between different data in cross-modal retrieval, a new supervised hash method is proposed. This method uses matrix factorization technique learning training data set to represent the low-dimensional potential semantic space. At the same time, this paper considers the semantic features as a separate mode, and maps them to the low-dimensional

latent semantic subspace by using matrix factorization, then maximizes the correlation among them in the subspace, and obtains the corresponding low-dimensional potential semantic representation. After that, the hash codes are obtained by using the hash function with better learning performance of orthogonal rotation matrix. A lot of experiments have been carried out in three commonly used data sets Wiki, MIRFlick and NUS-WIDE, and compared with some common cross-modal hashing methods, the results show the superior of this algorithm.

## Keywords

Cross-Modal Retrieval, Hashing, Matrix Factorization

Copyright © 2021 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

随着信息科技的迅速发展,人们不仅接收到各种信息也创造出各种信息,这些信息的表现方式多样,如图片、文本、视频和音频等等。其中,以任意一个单独形式表现出来的数据被称为单模态数据,多个单模态数据组合的方式表现出来的数据被称为多模态数据。如今,人们接触到的信息往往是多模态数据,例如,人们在微博上发布消息时,不仅可以上传图片或者视频,也可以加上相应的文章描述。在实际的生活和应用中,人们往往需要通过一种模态的数据去检索另外一种模态数据,例如利用图像去检索相对应的文本,这种检索方式被称为跨模态检索。然而不同模态对数据的表达不同会导致异构鸿沟,同时不同模态数据在语义描述上存在差异会产生语义鸿沟,这些是跨模态检索的难点。如何在海量高维数据中快速准确搜索到需要的内容成为一个急需解决的问题。

为了更好的解决海量数据信息检索问题,研究学者提出最近邻搜索问题,最近邻检索的核心思想是,给定一个待检测样本,返回数据库中与待检测样本距离最为接近的样本信息作为检索结果。实现最近邻检索的是线性查找,即计算待检测样本与数据集中所有样本之间的距离,然后返回满足检索样本信息作为检测结果。该方法在数据量不大的时候,具有很强的实用性,可以很高效的在数据集汇总找到符合要求的样本信息。然而面对海量的数据时,计算量迅速增加,对计算机的计算能力要求越来越高,因此该方法很难得到广泛的应用。为了有效降低计算量,研究学者提出了近似最近邻域检索方法,并且由于其高效性,在实际应用中获得广泛的应用。

在近似最近邻检索算法中,其中最有效的方法之一是哈希算法。哈希算法通过机器学习算法或随机的方法把数据映射到相应的潜在语义空间,再通过哈希函数将原始数据表示成二进制编码,利用二进制编码的位运算来进行检索,不仅可以降低存储开销,同时降低了计算复杂度,提高检索效率。因此,基于哈希的跨模态检索受到越来越多研究人员的关注,逐渐成为一个研究热点。

## 2. 相关工作

跨模态哈希检索是在建立两个模态语义关联的过程中学习哈希码,并将哈希检索的优点运用到跨模态检索问题中,根据是否利用数据本身的标签信息,跨模态哈希方法可以分为两种类型:无监督跨模态哈希和有监督跨模态哈希。

无监督跨模态哈希算法是通过多模态数据的模态内和模态间的关系来学习哈希码。一般的学习过程

是将原始数据投影到低维的汉明空间，然后在汉明空间中生成相应的哈希码。例如，协同矩阵分解哈希(Collective Matrix Factorization Hashing, CMFH) [1]，该算法利用协同矩阵分解学习多模态数据共同的潜在语义表达，然后再生成统一的哈希码。潜在语义稀疏哈希(Latent Semantic Sparse Hashing, LSSH) [2]，该算法利用稀疏编码学习图像的潜在语义表达，同时利用矩阵分解得到文本的潜在语义表达，然后再将学习到的潜在语义表达映射到一个联合的抽象空间中得到相应的哈希码。中间模态哈希(Inter-Media Hashing, IMH) [3]，该算法同时考虑两中模态数据中模态内和模态间的相关性，提出中间模态的哈希变换。

有监督跨模态哈希算法利用数据的类标签信息进行学习，往往能得到比无监督方法更好的结果。最大语义相关哈希算法(Semantic Correlation Maximization, SCM) [4]，该算法利用标签信息重构多模态数据的相关性矩阵，有监督矩阵分解哈希(Supervised Matrix Factorization Hashing for Cross-modal Retrieval, SMFH) [5]，该算法利用协同矩阵分解的方法得到多模态数据的潜在语义表示，然后再利用标签信息构造图约束，加强生成哈希码的鉴别能力。语义保持哈希(Semantics-Preserving Hashing, SePH) [6]，该算法利用标签信息构造一个亲和矩阵，并通过最小化该亲和矩阵和对应哈希编码之间的 KL 散度来学习哈希函数，从而使得学习到的哈希编码和原始数据之间的相似性保持一致。

### 3. 基于标签一致性哈希的跨模态检索算法

这一章节，将详细介绍基于标签一致性哈希的跨模态检索算法(LCH)，第一小节介绍它的目标函数。第二小节介绍它的优化方法和主要步骤。

#### 3.1. 目标函数

假设本文有一个数据集，该数据集由  $n$  个对象组成，每个对象有图片和文本两种表示模态，本文定义  $X^1 = \{x_1^1, x_2^1, \dots, x_n^1\}$  代表图像模态， $X^2 = \{x_1^2, x_2^2, \dots, x_n^2\}$  表示文本模态。对于第  $i$  个对象， $x_i^1 \in R^{d_1}$  代表  $d_1$  维度的图像特征向量， $x_i^2 \in R^{d_2}$  代表  $d_2$  维度的文本特征向量(一般  $d_1 \neq d_2$ )。每个模态的数据都可以通过下面的公式进行矩阵分解：

$$X^1 = U_1 V_1, X^2 = U_2 V_2 \tag{1}$$

其中， $U_1 \in R^{d_1 \times q}$ ， $U_2 \in R^{d_2 \times q}$ ， $V_1 \in R^{q \times n}$ ， $V_2 \in R^{q \times n}$ ， $q$  代表潜在语义代表的长度， $U_1, U_2$  分别代表图像和文本的投影矩阵， $V_1, V_2$  分别代表图像和文本的低维潜在语义表示。为了方便不同模态的潜在语义表示在这个低维子空间中进行耦合，本文将标签信息也视为一个单独的模态，定义为  $X_L \in R^{d_l \times n}$ ， $d_l$  表示类别的数量。同样利用矩阵分解的得到相应的低维潜在语义代表，即  $X_L = U_L V_L$ ， $U_L \in R^{d_l \times q}$ ， $V_L \in R^{q \times n}$ ， $U_L, V_L$  分别为标签信息的投影矩阵和潜在语义代表。本文假设图像和文本的语义代表通过线性转换之后能和标签信息的潜在语义代表进行耦合，用公式表示如下：

$$W_1 V_1 = V_L, W_2 V_2 = V_L \tag{2}$$

$W_1, W_2$  分别是图像和文本的线性转换。在得到潜在语义代表之后，大部分的哈希算法是利用符号函数  $\text{sgn}(\cdot)$  直接得到相应的哈希码，然而，这种生成哈希码的方法没有考虑量化损失带来的影响。本文通过最小化量化误差学习一个旋转矩阵，得到相应的哈希函数，从而产生性能更好的哈希码。

$$\min_{B, R} \|B - R V_L\|_F^2 \quad \text{s.t.} \quad B \in \{-1, 1\}^{q \times n}, R R^T = I \tag{3}$$

其中， $R \in R^{q \times q}$  是一个正交旋转矩阵， $B$  是所有训练数据的哈希码， $q$  代表哈希码的长度。

因此整个目标函数为：

$$\begin{aligned}
 \min_{U_1, U_2, W_1, W_2, V_1, V_2, V_L} & \lambda_1 \|X_1 - U_1 V_1\|_F^2 + \lambda_2 \|X_2 - U_2 V_2\|_F^2 + \lambda_3 \|X_L - U_L V_L\|_F^2 \\
 & + \alpha_1 \|V_L - W_1 V_1\|_F^2 + \alpha_2 \|V_L - W_2 V_2\|_F^2 + \|B - R V_L\|_F^2 \\
 & + \gamma \text{reg}(U_1, U_2, V_1, V_2, V_L, R) \\
 \text{s.t.} & B \in \{-1, 1\}^{r \times n}, R R^T = I
 \end{aligned} \tag{4}$$

其中,  $\|\cdot\|_F$  表示矩阵的 Frobenius 范数, 值为矩阵中每个元素的平方和再开平方的值。  $\lambda_1, \lambda_2, \lambda_3, \alpha_1, \alpha_2, \gamma$  分别为对应的权重因子,  $\text{reg}$  是正则化函数。

### 3.2. 优化方式

由于目标函数含有多个变量, 因此它是非凸, 值得注意, 对于其中任意一个参数目标函数是凸的, 因此本文使用交替迭代最小化方案来解决这个问题, 即固定所有的变量, 而只更新其中一个变量, 然后采用相应的方式迭代求解所有变量, 具体步骤如下:

步骤 1: 固定其它变量, 求解  $U_1, U_2, U_L$ , 目标函数可以分别重写为:

$$\begin{cases} \lambda_1 \|X^1 - U_1 V_1\|_F^2 + \gamma \|U_1\|_F^2 \\ \lambda_2 \|X^2 - U_2 V_2\|_F^2 + \gamma \|U_2\|_F^2 \\ \lambda_3 \|X_L - U_L V_L\|_F^2 + \gamma \|U_L\|_F^2 \end{cases} \tag{5}$$

通过对上面的式子分别求偏导, 求解得到:

$$\begin{cases} U_1 = X^1 V_1^T (V_1 V_1^T + (\gamma/\lambda_1) I)^{-1} \\ U_2 = X^2 V_2^T (V_2 V_2^T + (\gamma/\lambda_2) I)^{-1} \\ U_L = X_L V_L^T (V_L V_L^T + (\gamma/\lambda_3) I)^{-1} \end{cases} \tag{6}$$

步骤 2: 固定其它变量, 求解  $W_1, W_2$ , 目标函数重写为:

$$\begin{cases} \alpha_1 \|V_L - W_1 V_1\|_F^2 + \gamma \|W_1\|_F^2 \\ \alpha_2 \|V_L - W_2 V_2\|_F^2 + \gamma \|W_2\|_F^2 \end{cases} \tag{7}$$

通过对上面的式子分别求偏导, 求解得到:

$$\begin{cases} W_1 = V_L V_1^T ((V_1 V_1^T + \gamma/\alpha_1) I)^{-1} \\ W_2 = V_L V_2^T ((V_2 V_2^T + \gamma/\alpha_2) I)^{-1} \end{cases} \tag{8}$$

步骤 3: 固定其它变量, 求解  $V_1, V_2, V_L$ , 目标函数重写为:

$$\begin{cases} \lambda_1 \|X^1 - U_1 V_1\|_F^2 + \alpha_1 \|V_L - W_1 V_1\|_F^2 + \gamma \|V_1\|_F^2 \\ \lambda_2 \|X^2 - U_2 V_2\|_F^2 + \alpha_2 \|V_L - W_2 V_2\|_F^2 + \gamma \|V_2\|_F^2 \\ \lambda_3 \|X_L - U_L V_L\|_F^2 + \alpha_1 \|V_L - W_1 V_1\|_F^2 + \alpha_2 \|V_L - W_2 V_2\|_F^2 + \|B - R V_L\|_F^2 + \gamma \|V_L\|_F^2 \end{cases} \tag{9}$$

通过对上面的式子分别求偏导, 求解得到:

$$\begin{cases} V_1 = (\lambda_1 U_1^T U_1 + \alpha_1 W_1^T W_1 + \gamma I)^{-1} (\lambda_1 U_1^T X^1 + \alpha_1 W_1^T V_L) \\ V_2 = (\lambda_2 U_2^T U_2 + \alpha_2 W_2^T W_2 + \gamma I)^{-1} (\lambda_2 U_2^T X^2 + \alpha_2 W_2^T V_L) \\ V_L = (\lambda_3 U_L^T U_L + R^T R + (\alpha_1 + \alpha_2 + \gamma) I)^{-1} (\lambda_3 U_L^T X_L + \alpha_1 W_1 V_1 + \alpha_2 W_2 V_2 + R^T B) \end{cases} \quad (10)$$

步骤 4: 固定其它变量, 求解  $R$ , 目标函数重写为:

$$\min_R \|B - RV_L\|_F^2, \quad s.t. \quad RR^T = I \quad (11)$$

这是一个经典的 Orthogonal Procrustes Problem [7], 可以通过奇异值分解的方法求解。在进行奇异值分解后, 可以得到  $BV_L^T = S\Omega\tilde{S}^T$ , 然后求解  $R$ , 得到:

$$R = S\tilde{S}^T \quad (12)$$

步骤 5: 固定其它变量, 求解  $B$ , 目标函数重写为:

$$\min_B \|B - RV_L\|_F^2, \quad s.t. \quad B \in \{-1, 1\}^{q \times n} \quad (13)$$

对上式求解, 很容易就可以得到:

$$B = \text{sgn}(RV_L) \quad (14)$$

算法 1 总结了 LCH 的优化框架如下:

---

算法 1: 基于标签一致性哈希的跨模态检索(LCH)

---

输入: 训练样本的图像数据矩阵  $X^1$ , 文本数据  $X^2$ , 样本标签  $X_L$ , 参数  $\lambda_1, \lambda_2, \lambda_3, \alpha_1, \alpha_2, \gamma$ 。

输出: 哈希码  $B$

迭代更新:

1. 通过式(6)更新  $U_1, U_2, U_L$ ,
  2. 通过式(8)更新  $W_1, W_2$ ,
  3. 通过式(10)更新  $V_1, V_2, V_L$ ,
  4. 通过式(12)更新  $R$ ,
  5. 通过式(14)更新  $B$ ,
  6. 直至收敛。
- 

## 4. 实验

本章详细介绍本方法在 Wiki [8], MIRFick [9], NUS-WIDE [10]三个基准数据集上的实验结果及相应的分析。

### 4.1. 数据集介绍

**Wiki:** 该数据集从维基百科中搜集而来, 包含了 2866 对图像-文本数据对, 共分成 10 个语义类别。其中, 该数据集用 128 维的 SIFT 特征表示图片数据, 用 10 维 LDA 特征表示文本数据。在实验中, 我们随机将选择 75% 作为训练集, 25% 作为测试集。

**MIRFlick:** 该数据集是从 Flickr 网站汇总下载的图片及相应的文本, 共包含了 25,000 幅图像, 有 24 个语义类别, 其中每幅图像至少属于一个语义类别, 且一幅图像对应多个文本, 该数据集是一个多标签数据集。该数据集用 125 维的边缘特征表示图片数据, 用 500 维的 PCA 特征表示文本数据。在实验中,

将该数据集中没有文本标记的数据以及标签出现次数少于 20 次的数据剔除,然后将剩余数据中的 95% 作为训练集, 5% 作为测试集。

**NUS-WIDE:** 该数据集是新加坡国立大学公开的数据集, 共包含 269,648 张从 Flickr 网站上收集的图片及相应的文本, 每张图片平均有 6 个标注, 这些图像-文本对可以被分为 81 个类。该数据集用 500 维的 SIFT 表示图片数据, 1000 维的词向量特征表示文本数据。在实验中, 为了保证每类有足够多的训练样本, 选取数据中数量最多的 10 个类的数据进行实验, 从这些数据中选择 95% 作为训练集, 5% 作为测试集。

## 4.2. 评估度量及参数设置

本文选跨模态检索平均查准率均值(mAP)作为主要评价算法的整体性能, 其公式为:

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^N \text{AP}(q_i) \quad (14)$$

其中,  $q_i$  为一个查询样本,  $N$  为查询样本量,  $\text{AP}$  为平均查准率, 它的计算公式为:

$$\text{AP} = \frac{1}{T} \sum_{r=1}^R P_q(r) \delta(r) \quad (15)$$

其中,  $T$  是检索集中所有相关的实体个数,  $P_q(r)$  是按照相关度排名的前  $r$  个实体的查准率,  $\delta(r)$  是一个指示函数, 当第  $r$  个被检索到的实体与检索内容相关是, 其值为 1, 反之为 0。本文主要验证两种跨模态检索任务, 一种是利用图像去检索相关文本, 用 **Img2-Text** 表示, 另一种是利用文本来检索相关图像, 用 **Text2-Img** 表示。

本文针对三个数据集设置不同的参数进行实验, 对于 Wiki 数据集, 参数  $\{\lambda_1, \lambda_2, \lambda_3, \alpha_1, \alpha_2, \gamma\}$  的值分别为  $\{1, 1, 1, 0.1, 0.1, 0.1\}$ ; 对于 MIRFlick 数据集, 参数  $\{\lambda_1, \lambda_2, \lambda_3, \alpha_1, \alpha_2, \gamma\}$  的值分别为  $\{0.1, 100, 1, 0.1, 1, 1\}$ ; 对于 NUS-WIDE 参数  $\{\lambda_1, \lambda_2, \lambda_3, \alpha_1, \alpha_2, \gamma\}$  的值分别为  $\{0.1, 1, 100, 0.1, 1, 1\}$ 。为了全面评估本文算法, 哈希的长度分别设置为 16 bits, 32 bits, 64 bits 以及 128 bits。本文的实验在 Matlab2019b, Intel(R) Core(TM) i7-6700 CPU @3.40GHz 环境下进行。

## 4.3. 实验结果和分析

为了验证 LCH 算法的有效性, 我们选取了几个跨模态检索算法进行对比, 它们分别是 IMH, SCM, CMFH, SMFH, LSSH, 以及 SePH, 这些算法在前面章节已经作了简单的介绍。实验过程中, 对比算法的实验参数都是依据相关文件建议中的参数进行设置的。本文取 10 次相应的实验结果的评价值作为最终的实验结果。

### 4.3.1. 实验结果

表 1 展示了 LCH 算法和对比算法在 Wiki 数据集上的结果。表 2 展示了 LCH 算法和对比算法在 MIRFlick 数据集上的结果。表 3 展示了 LCH 算法和对比算法在 NUS-WIDE 数据集上的结果。

**Table 1.** Experimental results on the Wiki dataset

**表 1.** 在 Wiki 数据集上的实验结果

任务	方法	16	32	64	128
Img2-Text	IMH	0.1952	0.2003	0.2084	0.2097
	SCM	0.1611	0.1434	0.1366	0.1356
	CMFH	0.2132	0.2259	0.2632	0.2419

Continued

	SMFH	0.2699	0.2835	0.2920	0.2981
	LSSH	0.2141	0.22216	0.2218	0.2211
	SePH	0.2787	0.2956	0.3064	0.3134
	LCH	0.2681	0.3005	0.3118	0.3233
	IMH	0.1508	0.1581	0.1636	0.1668
	SCM	0.1542	0.1373	0.1294	0.1239
	CMFH	0.4884	0.5132	0.5269	0.5379
Text2-Img	SMFH	0.6051	0.6257	0.6357	0.6428
	LSSH	0.5031	0.5224	0.5293	0.5346
	SePH	0.6318	0.6577	0.6646	0.6609
	LCH	0.6801	0.6984	0.7203	0.7345

**Table 2.** Experimental results on the MIRFlickr dataset

**表 2.** 在 MIRFlickr 数据集上的实验结果

任务	方法	16	32	64	128
	IMH	0.5671	0.5654	0.5655	0.5654
	SCM	0.5876	0.5757	0.5704	0.5627
	CMFH	0.5861	0.5835	0.5844	0.5849
Img2-Text	SMFH	0.6237	0.6343	0.6448	0.6489
	LSSH	0.5784	0.5804	0.5797	0.5816
	SePH	0.6732	0.6771	0.6783	0.6817
	LCH	0.6708	0.6892	0.6955	0.7103
	IMH	0.5672	0.5653	0.5647	0.5647
	SCM	0.5862	0.5745	0.5657	0.5590
	CMFH	0.5937	0.5919	0.5931	0.5919
Text2-Img	SMFH	0.6133	0.6209	0.6295	0.7354
	LSSH	0.5998	0.5927	0.5932	0.5932
	SePH	0.7197	0.7271	0.7309	0.7387
	LCH	0.6996	0.7485	0.7364	0.7564

**Table 3.** Experimental results on the NUS-WIDE dataset

**表 3.** 在 NUS-WIDE 数据集上的实验结果

任务	方法	16	32	64	128
	IMH	0.4029	0.4090	0.4268	0.4248
	SCM	0.3907	0.3758	0.3646	0.3542
	CMFH	0.4267	0.4229	0.4207	0.4182
Img2-Text	SMFH	0.4553	0.4623	0.4658	0.4680
	LSSH	0.3900	0.3924	0.3962	0.3966
	SePH	0.5421	0.5499	0.5537	0.5601
	LCH	0.5841	0.5991	0.6053	0.6061

Continued

	IMH	0.4002	0.4091	0.4315	0.4317
	SCM	0.3875	0.3693	0.3608	0.3538
	CMFH	0.4627	0.4556	0.4518	0.4478
Text2-Img	SMFH	0.5033	0.5056	0.5065	0.5079
	LSSH	0.4286	0.4248	0.4228	0.4175
	SePH	0.6302	0.6425	0.6425	0.6584
	LCH	0.7382	0.7541	0.7671	0.7678

### 4.3.2. 实验结果分析

表 1, 表 2 和表 3 分别给出了 LCH 与对比算法在 Wiki, MIRFlick 和 NUS-WIDE 这三个数据集上的两种跨模态任务的 mAP 数值, 哈希码的长度分别为 16 bit 32 bit 64 bit 和 128 bit。

对于 Wiki 数据集, 从表 1 的数据可以看出, LCH 在不同哈希码长度下的 mAP 值优于所对比的算法, 验证了 LCH 在跨模态检索任务中的有效性。同时, 观察表 1, 可以发现大部分有监督跨模态哈希方法比无监督跨模态哈希方法检索效果更好, 这是因为有监督的方法通过嵌入真实标签信息到哈希码中, 可以大大增加哈希码的判别力。文本检索图片任务的 mAP 值比图片检索文本的 mAP 值普遍要高, 这是因为文本所包含的信息比图片信息要直观, 能更好表达数据的核心语义。此外, 通过表 1 还可以观察到, 哈希码越长, 哈希码所能保存的信息越多, 因此检索效果越好。

对于 MIRFlick 数据集和 NUS-WIDE 数据集, 从表 2 和表 3 中的 mAP 数值对比可以观察到, LCH 优于其它方法, 这与在 Wiki 数据集集中的观察一致, 再次验证了本文方法在跨模态检索任务中的有效性。此外, 我们可以观察到, 这两个数据集中, mAP 值都比 Wiki 数据集中要高, 这是因为这两个数据集都是多标签数据集, 图像和文本的语义关系更加紧密, 这也表明利用标签信息能更好的指导跨模态检索任务。

### 4.3.3. 模型收敛性分析

图 1 为 LCH 在三个数据集中迭代收敛过程, 从图 1 的实验结果中可以证明, LCH 算法不仅是有效收敛的, 而且收敛速度很快, 适合在大规模数据中进行跨模态检索任务。

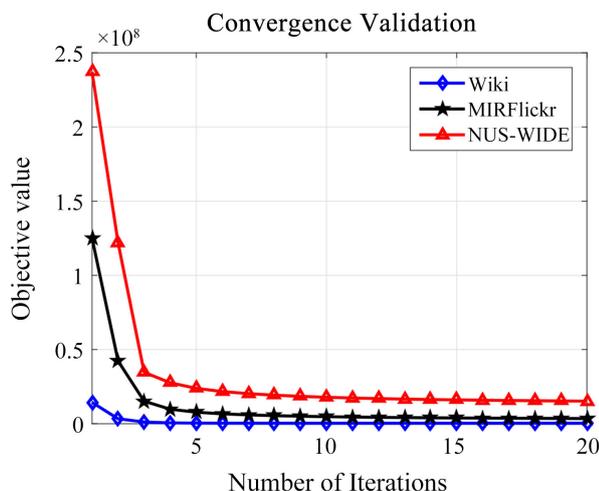


Figure 1. Curve: Model convergence analysis

图 1. 模型收敛性分析

## 5. 结论

本文提出了一种新的跨模态检索方法，即基于一致性哈希的跨模态检索算法。该算法不仅利用原始数据的信息，还将原始数据中标签信息加入到哈希码的学习过程中，同时，本文利用正交旋转矩阵来学习哈希函数，从而降低产生哈希码时带来的量化误差。本文在三种常用的数据集上进行了大量的实验，并与相关的跨模态哈希算法相比，该方法能够更好的提出检索性能。

## 参考文献

- [1] Ding, G., Guo, Y. and Zhou, J. (2014) Collective Matrix Factorization Hashing Formultimodal Data. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, 23-28 June 2014, 2083-2090. <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. *ACM SIGIR International Conference Research Development in Information Retrieval*, Queensland, July 2014, 415-424. <https://doi.org/10.1145/2600428.2609610>
- [3] Song, J.K., Yang, Y., Yang, Y., et al. (2013) Inter-Media Hashing for Large-Scale Retrieval from Heterogeneous Data Sources. *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, New York, June 2013, 785-796. <https://doi.org/10.1145/2463676.2465274>
- [4] Zhang, D. and Li, W.J. (2014) Large-Scale Supervised Multimodal Hashing with Semantic Correlation Maximization. *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, **28**, 2177-2183.
- [5] Tang, J., Wang, K. and Shao, L. (2016) Supervised Matrix Factorization Hashing for Cross-Modal Retrieval. *IEEE Transactions on Image Processing*, **25**, 3157-3166. <https://doi.org/10.1109/TIP.2016.2564638>
- [6] Lin, Z., Ding, G., Hu, M., et al. (2015) Semantics-Preserving Hashing for Cross-View Retrieval. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7-12 June 2015, 3864-3872. <https://doi.org/10.1109/CVPR.2015.7299011>
- [7] Schonemann, P.H. (1966) A Generalized solution of the Orthogonal Procrustes Problem. *Psychometrika*, **31**, 1-10. <https://doi.org/10.1007/BF02289451>
- [8] Lu, X., Zhang, H., Sun, J., et al. (2018) Discriminative Correlation Hashing for Supervised Cross-Modal Retrieval. *Signal Processing: Image Communication*, **65**, 221-230. <https://doi.org/10.1016/j.image.2018.04.009>
- [9] Chua, T.S., Tang, J., Hong, R., et al. (2009) Nus-Wide: A Real-World Web Image Database from National University of Singapore. *ACM International Conference on Image and Video Retrieval*, Article No. 48, 1-9. <https://doi.org/10.1145/1646396.1646452>
- [10] Huiskes, M.J. and Lew, M.S. (2008) The MIR Flickr Retrieval Evaluation. *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, New York, October 2008, 39-43. <https://doi.org/10.1145/1460096.1460104>