

# A Survey of Supervised Learning to Hash

Xin-Yue Feng

October 12, 2020

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Nearest Neighbor Search . . . . .	3
1.2	Search with Hashing . . . . .	3
1.3	Supervised Learning to Hash . . . . .	3
<b>2</b>	<b>有效利用监督信息</b>	<b>4</b>
2.1	Pointwise similarity . . . . .	4
2.2	Pariwise similarity . . . . .	5
2.3	Multiwise similarity . . . . .	5
<b>3</b>	<b>离散优化问题</b>	<b>6</b>
3.1	逼近方法 . . . . .	6
3.2	two-step 方法 . . . . .	7
<b>4</b>	<b>生成更有效的 code</b>	<b>10</b>
4.1	Data dependent hashing . . . . .	10
4.2	Deep hashing . . . . .	10
<b>5</b>	<b>其他问题</b>	<b>10</b>

# 1 Introduction

## 1.1 Nearest Neighbor Search

给定 query  $q$ , 数据集  $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ , 定义样本间距离  $dist(q, \mathbf{x}_i)$ , 最近邻检索 (Nearest Neighbor Search) 指从  $\mathcal{D}$  中寻找样本  $\mathbf{x}_{nn}$  使得其和  $q$  的距离最近, 即  $\mathbf{x}_{nn} = \operatorname{argmin}_{\mathbf{x}_i} dist(q, \mathbf{x}_i)$ . 最近邻检索的应用非常广泛, 以图搜图的搜索引擎就是其中的典例, 用户输入的对图像即为 query  $q$ , 庞大的网络图像库即为数据集  $\mathcal{D}$ , 该搜索引擎的目的就是根据 query 图像, 返回数据集中和其最相似的  $n$  个图像。

最近邻检索中通常使用欧式距离来定义样本间距离, 这里不妨设  $q, \mathbf{x}_i \in \mathbb{R}^d$ , 则  $dist(q, \mathbf{x}_i) = \|q - \mathbf{x}_i\|_2$ . 但通常特征维度  $d$  很高, 数据集中的样本量  $N$  更是达到千万以上的数量级, 给定一个 query 在数据集中进行检索的计算复杂度高达  $O(Nd)$ , 同时实值特征的存储也会消耗大量的空间。因此随着数据集的飞速增大, 这种时间和空间的消耗逐渐变的难以承受。

## 1.2 Search with Hashing

为了解决最近邻搜索中存在的问题, 人们提出结合 hashing 的搜索算法。详细来讲, hashing 算法将特征从原空间映射到 Hamming 空间, 相比起实值特征, hamming 空间的 hashing code 长度更短且仅由 0/1 构成, 大大减小了计算量和存储开销。但长度变短、实值变二值不可避免的会造成信息的损失, 因此 hashing 算法的重点就是如何生成尽可能保证原样本空间信息量的 code。以图像检索为例, 对于相似的图像, 我们希望其映射为 hashing code 后也相似, 对于不相似的图像反之亦然。

最初的 hashing 方法是独立于数据的 (Data independent), 经典的算法为 LSH [1]。该类方法通常使用随机映射来构造 hashing code, 虽然速度较快, 但忽略了数据集本身的信息, 导致生成的 hashing code 效果并不如人意。随后基于数据 (Data dependent) 的 hashing 方法开始发展, 包括无监督的方法 SH [2]、ITQ [3]、AGH [4]、DGH [5]、SGH [6] 等, 有监督的方法又可分为基于 class 或 pair-wise 信息的 MLH [7]、KSH [8]、TSH [9]、LFH [10]、FastH [11]、SDH [12]、COSDISH [13]、AffHash [14] 等, 以及基于 ranking 信息的 HDML [15]、OPH [16]、RSH [17]、CGH [18]、RPH [19]、Top-RSBC [20]、DSeRH [21] 等。相比于独立于数据的方法, 基于数据的方法能学习更短、且检索精度更高的 hashing code。

但前面提到的 hashing 方法通常存在两大问题: 一是其精度受限于提取的特征的准确度, 二是通常难以优化较为复杂的目标函数。而深度神经网络 (Deep Neural Network) 的出现一定程度上能够帮助解决上述的问题, 因此在 hashing 中引入 DNN 成为了目前主流的 hashing 研究方法。Deep hashing 方法也可分为有监督和无监督两大类。

本文主要从有监督方法入手, 讨论 Data dependent 和深度的 hashing 方法, 下文统称为 Supervised Learning to Hash 方法。

## 1.3 Supervised Learning to Hash

Hashing 方法的主要研究目标是如何保持样本映射前后的相似性, 为了达到这个目标, 通常有三个问题需要解决: (1) 如何有效利用监督信息; (2) 离散优化问题; (3) 如何生成更有效的 hashing code。

**如何有效利用监督信息** 如何利用监督信息不仅是 Hashing 方法需要研究的问题, 也是所有最近邻搜索方法研究的核心, 因为只有效挖掘监督信息, 才能够保证样本映射前后的相似性。现有方法主要通过设计不同的 loss

来将监督信息嵌入到网络学习过程中，根据利用监督信息的方式不同，目前的方法可以分为 pointwise similarity preserving、pairwise similarity preserving 以及 multiwise similarity preserving 三类。pointwise similarity preserving 方法只用到了每个样本的类别信息。pairwise similarity preserving 方法用到了样本间的关系信息，例如两个样本相似或不相似。multiwise similarity preserving 考虑到了多个样本间的关系例如 triplet loss 考虑了一个样本和其相似样本以及不相似样本之间的关系。

**离散优化问题** 相比于输出实值特征的普通方法，hashing 方法的不同之处是其优化的目标为离散的 binary code，这导致其在优化过程中无法使用常见的连续优化算法。

**如何生成更有效的 hashing code** hashing code 相比实值特征位数更少，且每一位只有 0/1 二值，这就要求 code 要更有效的携带信息，才能尽可能逼近实值特征的检索精度。因此如何让图像特征信息有效的嵌入 code 中，尽量减少 code 间的冗余，也是研究的一个重点。

**其他** 除了针对保相似性研究产生的以上问题，hashing 中还有一些其他方面的问题需要解决：例如如何进行 online hashing、或是分析 hashing 模型的鲁棒性。但因为针对这些问题的研究目前还比较少，所以整体放在一章进行讨论。

因此，下文将会分为四个部分，前三个部分详细介绍针对上面三个主要问题的研究，分析其发展历程和最新研究现状，最后一个部分讨论其他问题的研究情况。

## 2 有效利用监督信息

监督信息可分为 Pointwise similarity、Pairwise similarity 和 Multiwise similarity 三类，从左往右看这三类监督信息，通常来讲其架构的 loss 得到的结果是越来越好的，但同时耗时也是逐渐增加的。因此一般会根据实际情况来进行选择，并没有绝对的好坏，因此下面我们将主要进行每一类监督信息的类内方法的比较。

### 2.1 Pointwise similarity

Pointwise similarity 指的是样本的类别标签，其优点是计算效率高，对于大小为  $N$  的数据集其计算复杂度为  $O(N)$ 。对于 Data dependent hashing，SDH15 [12] 中使用了 pointwise loss，FDSH [22] 对其进行了改进。

Deep hashing 中不常见单独使用 pointwise loss 的情况，因为仅使用类别信息在一些需要细致区分（例如人脸数据集）的数据集上有时难以提取到很具有区分度的特质，大多数情况下其都是和其他 loss 联合使用，深入挖掘样本之间的关系。例如 DSDH [23] 和 DHAG [24] 都联用了 Pointwise similarity 和 Pairwise similarity。不过因为在实际应用中，不一定能得到每个样本准确的 pointwise 信息，因此更多的方法倾向于只使用 Pairwise 或 Multiwise 信息。

### 2.2 Pairwise similarity

Pairwise similarity 指的是样本对之间相似/不相似的信息，对于只有类别标签的数据集，一般认为含有相同 label 的样本对为相似对，label 完全不同的样本对为不相似对，以此转换成 Pairwise similarity。相比 Pointwise

similarity 仅仅是判断样本是否属于某个类别，Pairwise similarity 通过拉近相似对之间、推远不相似对来进一步挖掘样本间的相似和不同之处。不过由于要两两比较样本，Pairwise similarity 的花费的时间要比 Pointwise similarity 多，计算复杂度为  $O(N^2)$ 。

之前的 Survey [25] 中已经非常详细对不同构造的 pairwise loss 进行了介绍和分类，且说明了虽然不同方法对 pairwise loss 的构造有区别，但这些区别并不会对结果起决定性的影响，更重要的还是 loss 中加入的其他的限制，例如离散优化等，所以下文不会对不同的 pairwise loss 进行详细比较，只是陈列出一些目前常用的 loss。

### 2.2.1 Data dependent hashing

KSH [8] 首次提出用内积计算 hamming distance，即令  $dist_h = (qS - \mathbf{B}^T \mathbf{B})/2$ ，并提出了后来广泛使用的基于 pairwise similarity 的目标函数

$$\min_{\mathbf{B}} \|\mathbf{B}^T \mathbf{B} - qS\|^2 \quad (1)$$

相对于之前的方法，KSH 提出的目标函数简单明了，很大程度上减小了优化难度。

在 KSH 之后，LFH [10] 从概率分布的角度提出了一种基于 latent factor model 的 pairwise loss:

$$\begin{aligned} \max_{\mathbf{U}} \mathcal{L}(\mathbf{U}, \mathcal{S}) &= \log p(\mathbf{U}|\mathcal{S}) = \log p(\mathcal{S}|\mathbf{U})p(\mathbf{U}) \\ &= \sum_{s_{ij} \in \mathcal{S}} [s_{ij} \log(\sigma(\Theta_{ij})) + (1 - s_{ij}) \log(1 - \sigma(\Theta_{ij}))] \\ &\text{where } \Theta_{ij} = \mathbf{u}_i^T \mathbf{u}_j / 2 \end{aligned} \quad (2)$$

其本质上是在最大化给定 label 的情况下 binary code 的后验概率分布。相比 KSH 设计的 loss，LFH 提出的基于概率模型的 loss 有两个优点：一是灵活性，因为其可以通过设计不同的先验分布达到不同的目的；二是结果更好，因为先验分布可以带来更多的信息。但其问题在于 loss 中将距离归一化时，使用的是 sigmoid 的函数，这导致两个样本之间距离还不够大/小的时候，sigmoid 的函数就已经达到饱和了，造成的后果就是相似样本对不能足够接近，而不相似样本对也无法拉的足够开。虽然理论分析其存在这样的缺点，但在实验中并没有明确体现，原因有待探究

### 2.2.2 Deep hashing

BDNN [26]、ADSH [27] 等方法中 loss 的设计主要参考了 KSH [8]，DPSH [28]、DHN [29]、DSEH [30] 等方法中 loss 的设计都主要参考了非深度方法 LFH [10]，即基于后验概率的思想设计了 pairwise loss。DSH [31] 则基于经典的 contrastive loss 构建了 hashing loss，本质上只是将前者中的欧式距离换为 hamming 距离。

## 2.3 Multiwise similarity

Multiwise similarity 指的是多个样本之间的关系，例如样本 A、B 的距离比样本 A、C 的距离更近。Multiwise similarity 由于用到了多个样本间的交互信息，因此相比 Pairwise similarity 能更好的挖掘样本间关系。相对的，其计算复杂度也更高。

在 Multiwise similarity 中，最基础的是 triplet loss，其只讨论了 anchor 样本和其相似/不相似样本组成的三元组之间的关系，之后提出的 ranking loss 扩大了讨论范围到 anchor 样本和所有其他样本之间的关系，除此之外，

目前最新的方法着手于直接优化评价指标 MAP、NDCG 等，相当于讨论了样本间更加复杂的 ranking 关系。

### 2.3.1 Data dependent hashing

HDM [15] 基于 triplet loss 进行优化，OPH [16]、RSH [17]、Top-RSBC [20]、DSeRH [21] 都是基于 ranking loss 进行优化的。

### 2.3.2 Deep hashing

Bit-Scalable Hashing15 [32] 是第一个在 Deep hashing 中应用 triplet loss 的方法。DSRH15 [33] 是第一个结合 DNN 和 ranking 信息进行 hashing learning 的方法，而且是第一个 end-to-end 的 deep hashing 方法。除此之外，[20] 也是同时期提出的 ranking-based hashing 方法。

由于 DNN 使得学习复杂的目标函数成为可能，因此近年一些 Deep hashing 方法尝试直接去优化评价指标。TALR18 [34] 提出了 hashing 应用在检索中的一个问题：由于 hashing code 位数少，因此检索结果中会出现很多样本和 query 的距离是相同的，对这些样本进行不同的排序，会导致现有的评价指标达到的结果产生较大的波动。因此其提出了 tie-aware 的 AP 和 NDCG 指标，使得其对于上面这种情况能保持一致的结果，重要的是 TALR 做到了直接优化这两个指标，在当前数据集上达到了 SOTA 的结果。

除此之外，MinHash [35] 发现好的输出结果，其相似/不相似样本对的距离分布相差越大越好，因此使用 mutual information 来衡量两个分布间的差异，并直接去优化此指标，其设计的 loss 的优点是不包含任何参数，可以在各类数据集上进行自适应并取得不错的结果。

## 3 离散优化问题

离散优化是 hashing 方法中最核心的问题，因为 hashing 方法和直接输出实值特征的方法，其最重要的区别就在于前者输出的为 binary code，如果进行 relaxation 来适应常见的连续优化算法则会造成一定误差，而研究的目的是想要尽可能降低这种误差。

现有的解决离散优化的方法主要可以分为两大类，一是通过逼近的方法，二是 two-step 的方法。其中每一类中的方法又可以细分，下面将进行详细的介绍。

### 3.1 逼近方法

逼近方法首先会选择对 binary code 进行 relaxation，即将需要优化的 code 从  $\{-1, 1\}$  放松到实值空间，使其能够进行连续的 back propagation。之后再加入一系列限制来减少 relaxation 造成的误差。

使用 sigmoid/tanh 最常用的限制即为将 sign 函数替换为 sigmoid/tanh 函数 [36] [32]，令输出的结果控制在  $[-1, 1]$  之间。

使用 Hypercubic Quantization Loss [25] 但上面使用 sigmoid/tanh 的做法会导致优化速度较慢，而且可能造成一定的梯度消失，因此 DSH [31] 去掉了输出层的非线性变换，在 loss 中加入 HQ loss 项  $\|x - \text{sign}(x)\|^2$  来

控制  $x$  的输出范围, 由实验证明, 相比使用只 sigmoid/tanh 函数来控制  $x$  输出范围的方法, 此方法很大程度上提升了训练结果。同样使用了该种思想的方法还包括 DPSH [28]、DHN [29]、DSEH [30] 等。

**逼近 sign 函数** 虽然加入 HQ loss 项可以一定程度上让输出结果逼近  $\{-1, 1\}$ , 但实验中发现量化损失通常不会降的很低, 这说明仍然存在有较大的误差。因此 HashNet [37] 提出一种用 tanh 函数直接逼近 sign 函数的优化方法, 其主要包括两个步骤: (1) 增大 tanh 函数中的参数, 微调其他参数; (2) 训练到网络收敛。重复上面两步直到增大参数对网络训练结果不再产生较大影响。经试验证明其结果是优于 DHN [29] 的。但 HashNet 的缺点是需要多次迭代, 计算效率较低。

**泰勒展开** 基于泰勒展开的逼近原理, DMDH [38] 对离散的  $\text{loss} \mathcal{L}(\mathbf{B}) = \text{tr}(\mathbf{B}^T \mathbf{D} \mathbf{B})$  进行泰勒展开, 使得其变为连续的可优化的问题。值得注意的是, 论文中发现一次展开项很接近常见的量化损失项, 这从侧面支持论文中的做法是有效的。文中最终给基于二次展开进行优化, 得到了优于前面提到的方法的结果。论文中还用到了离散化的 back propagation 的技巧, 同期 GreedyHash [39] 也是基于贪心算法的思想, 直接将 back-propagation 过程离散化, 使得 sign 函数可以被离散的优化。

**使用 Product Quantization Loss [25]** DQN [40] 是首个提出结合 Product Quantization Loss 和 Deep hashing 的方法, PQ loss 比上文提到的 HQ loss 能够更多的降低量化损失, 因此能到的更好的结果。但 DQN 中, 学习 codebook 和学习网络参数的过程是分离的, 因此 DEPQ [41] 提出一种 end-to-end 的方法, 能学习到更合适的 codebook。同时期 DSQ [42] 也提出了 end-to-end 的方法, 且解决了 codebook 学习过程中存在的 norm variance 问题。除此之外, SPDAQ [43] 从另一个方面改进了 DQN 的方法, 其在每一次 mini-batch 的训练过程加入全部样本的影响, 以达到全局最优解。

## 3.2 two-step 方法

two-step 的方法是将离散优化问题转变成两个子问题来解决的。其最先由无监督的非深度 hashing 方法 [44] 提出, 接着在有监督的非深度 hashing 方法中被广泛利用并进一步挖掘, 之后深度 hashing 方法中也推广了该类算法来解决离散优化 binary code 的问题。下面将详细介绍其在非深度方法和深度方法中的发展过程:

### 3.2.1 Data dependent hashing

**TSH13 [9]** TSH 是第一个采用 two-step 结构的有监督 hashing 方法, 其本质上是基于 [44] 的无监督 two-step 结构进行改进的。在 Step1 中, TSH 直接利用监督信息离散优化 binary code, 得到 code 后, Step2 中训练  $q$  个分类器 ( $q$  为 code 的位数) 来得到 feature 和 code 之间的 hash function。

TSH 的贡献在于提出了一个可以适用所有 pairwise loss 的离散优化方法, 在同样的目标函数下, 该优化方法比之前的方法能得到更好的结果。例如 TSH-KSH 相比 KSH [8], 在 CIFAR10 数据集上的 map 提高了近 4%。但 TSH 的缺点是其 code 优化过程只受 label 指导, 和原数据无关, 这导致生成的 code 可能会不符合原数据的一些分布特点, 而原数据直接基于这些 code 进行分类训练也可能导致过拟合。因此 FastH14 [11] (此为 TSH 作者另一篇文章) 对 TSH 的优化步骤进行了改进: 每优化一位 code 后就会训练一次分类模型, 通过分类模型更新这一位的 code 后, 再继续下一位 code 的优化, 以此达到将原数据和 code 优化过程连接起来。除此之外, FastH 还使用了更快的优化算法 GraphCut。

**DGH14 [5]** DGH 应是第一个将 two-step 方法融合为一体的无监督算法, 其首先固定 hashing function, 优



化  $B$ ，之后固定  $B$ ，再优化 hashing function，迭代进行此过程直到收敛。除此之外，其还提出了 Anchor Graph 算法，有效减小了计算图的边权矩阵的时间复杂度。

**SDH15 [12]** SDH 是第一个将 two-step 方法进行合并的有监督算法，不过由于其是基于 point-wise 信息进行训练的，目标函数和 TSH 不同，其不能化为 BQP 形式快速优化，因此其采用的是 cyclic coordinate descent (DCC) 的方法。FSDH [22] 对 SDH 方法的速度进行了优化，对 SDH 利用 point-wise 信息的方法也进行了改进，进一步提升了结果。除此之外，AffHash16 [14] 也将融合了 two-step 方法，不过是从另一个角度考虑的：其主要基于 method of auxiliary coordinates (MAC) 算法，将 two-step 优化问题看作一个嵌套优化问题 (nested functions) 来解决。

**COSDISH16 [13]** 但 TSH 和 FastH 的缺点都是不能处理大数据集，因为其优化 code 的时候是基于整个数据集的。因此 COSDISH 又设计了 column sampling 的方法，其将 step1 中优化 code 的过程又拆分成了迭代的两步：

- 首先随机采样得到数据集的子集  $B^\Omega$ ，剩余补集为  $B^\Gamma$ ，注意  $B^\Omega \ll B^\Gamma$ 。那么原本的目标函数 (3) 就变成了 (4)，对于  $B^\Omega$ ，我们采用和 TSH 类似的方法进行优化 (即转化为 BQP 形式)
- 得到了  $B^\Omega$  后，使用  $\text{sign}(S^\Gamma B^\Omega)$  即可得到  $B^\Gamma$ ，该步计算花费很低，因此哪怕  $B^\Gamma$  很大极损速度也比较可观

在 COSDISH 的实验中，为了结果稳定性，对于一个子集  $B^\Omega$ ，通常会重复上述两个步骤 3 次，之后再重新采样新的子集。重采样 10 次左右就可以得到很好的结果，不需要遍历所有的数据。得到优化的 code 之后，基于数据训练  $q$  个二分类器即可。

疑问：COSDISH 还是类似 TSH 一样 two-step 是完全分离进行训练的，并没有像 FastH 一样融合了两个过程，理论上相比 FastH 其是 suboptimal 的，但为什么其结果会比 FastH 好很多呢？

$$\min_{B \in \{-1, 1\}^{n \times q}} \|qS - BB^T\|_F^2 \quad (3)$$

$$\min_{B^\Omega, B^\Gamma} \|qS^\Gamma - B^\Gamma [B^\Omega]^T\|_F^2 + \|qS^\Omega - B^\Omega [B^\Omega]^T\|_F^2 \quad (4)$$

**FTH16 [45]** two-step 的方法一开始都是基于 pair-wise loss，因为 pair-wise loss 由于其对称性很容易化为 BQP 形式，而 BQP 形式易于进行离散优化。FTH 提出了一种新颖的将 triplet loss 化为 BQP 形式的方法，由于 triplet loss 相比 pair-wise loss 的优越性，该方法进一步提高了 two-step 算法的精度。

除此之外，还有结合 pair-wise 和 point-wise 信息进行优化的 twp-step 方法 DSDH [23]，因为这些改进不涉及离散优化的本质，这里不再赘述。

### 3.2.2 Deep hashing

**CNNH14 [46]** CNNH 是第一个 deep hashing 方法，也是第一个将 two-step 结构用在 deep hashing 中的方法，其将 TSH [9] 的 step2 中的传统分类器 (e.g.SVM) 改为 deep convolutional neural networks，增强了分类效果。在其之后 FTDSH [45] 提出了类似的 two-step 深度方法，但其构建了适合 BQP 优化的 triplet loss。

**DSDH17** [23] 相比 CNNH, DSDH 使用了 end-to-end 网络结构, 能提出更有效的特征。其整体算法和优化步骤与 SDH 很相似, 都是合并了的 two-step 方法, 只是多加入了 pair-wise loss 以及将模型改为了深度神经网络, 这里不再赘述。

**DDSH18** [47] 先前的 two-step 方法存在的问题是: label 信息要通过 code 才能传递给 hash function, 这降低了训练效率。因此 DDSH 通过将 code、hash function 以及 label 信息放在一个式子里进行优化来解决该问题。DDSH 主要是基于 COSDISH 的模型进行改进的, 目标函数如下

$$\min_{X^\Omega, X^\Gamma} \|qS^\Gamma - X^\Gamma[X^\Omega]^T\|_F^2 + \|qS^\Omega - X^\Omega[X^\Omega]^T\|_F^2 \quad (5)$$

接着引入 auxiliary variables  $B^\Omega$ , 上式变为

$$\min_{B^\Omega, X^\Gamma} \|qS^\Gamma - X^\Gamma[B^\Omega]^T\|_F^2 + \|qS^\Omega - B^\Omega[B^\Omega]^T\|_F^2 \quad s.t. \quad b_i \in \{-1, 1\}^q, \forall i \in B^\Omega \quad (6)$$

之后类似 AffHash 中迭代优化的思想, 首先固定  $X^\Gamma$ , 对  $B^\Omega$  进行离散优化, 之后固定  $B^\Omega$ , 使用 back propogation 优化  $X^\Gamma$ .

ADSH [27] 方法对 DDSH 进行了改进, 进一步优化了算法速度。其目标函数和 DDSH 有区别, 一开始就设置为不对称:

$$\min_{B^\Omega, B^\Gamma} \|qS^\Gamma - B^\Gamma[X^\Omega]^T\|_F^2 + \|qS^\Omega - B^\Omega[X^\Omega]^T\|_F^2 + \lambda \|B^\Omega - X^\Omega\|_F^2 \quad (7)$$

每次迭代中, DDSH 需要用 back propogation 更新  $X^\Gamma$ , ADSH 需要更新  $X^\Omega$ , 由于  $|\Omega| \ll |\Gamma|$ , 因此 ADSH 的速度相比 DDSH 要快很多。

**HBMP18** [48] 之前的 two-step 方法只是分析了误差下届和方法的收敛性, HBMP 则提出一种指标来评价 two-step 方法生成的 code 的质量, 之后基于此提出了一种 residual learning scheme, 其生成的 code 能够以任意精度逼近任意 neighborhood structure。同时文章证明了 two-step 方法中的迭代过程是不必要的, 并提出了一种更快速的训练方法。

## 4 生成更有效的 code

### 4.1 Data dependent hashing

在非深度又或者无监督的方法中, 通常使用  $B^T B = nI$  来保持 bit 之间的独立性, 减少信息冗余, 用  $\mathbf{1}B = \mathbf{0}$  来使得每位 bit 有相同的可能性产生  $\{-1, 1\}$ , 以此来保证生成 code 的有效性。



## 4.2 Deep hashing

但在 Deep hashing 方法中，由于每次只能处理一个 mini-batch，而上面两个约束在 mini-batch 的作用变弱 [49]，因此一些新的方法提出来解决该问题。

NINH [36] 设置了一种专门针对 hashing 方法的网络结构，对于从倒数第二层 FC 层输出的 feature，其首先将其平均分成  $q$  段 ( $q$  为 bit 数)，之后再分别映射到一维，经过 sigmoid 变换和二值化即可得到  $q$  位的 code。相比传统直接使用两层 FC 的方法，其这么做的优点是可以减少 bit 之间信息冗余，使得不同 label 样本之间的 code 更具有区分度。

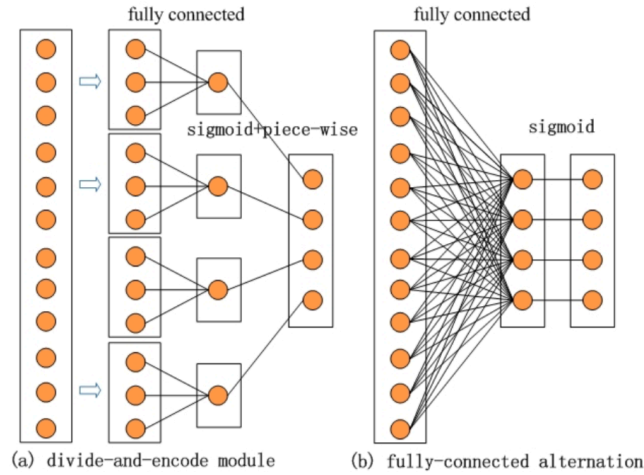


Figure 1: (a) 为 NINH 方法设计的模型结构，(b) 为传统的 FC 结构

Bit-Scalable Hashing [32] 给 code 的每一位设置权值，通过去除权值较小的位来达到自由控制 code 长度的效果，使得 code 能更有效的携带信息。

THDSH [49] 发现目前 hashing 方法生成的 code 存在 bit 之间相关性过高的问题，因此其将数据分为  $m$  组，每组训练一个模型并输出  $k/m$  位 code，之后合成  $k$  位的 code。注意为了减少计算量，这些子网络共享底层特征层。

## 5 其他问题

### 5.1 Online Hashing

Online hashing 中存在的问题是：每当有新类别的数据加入时，模型就需要重新训练，而一旦网络更新了，数据库中的所有 hashing code 就要随之更新，导致时间花费很大。

因此 MinHash [35] 通过 mutual information 指标来决定何时更新数据库中的 hashing code，以此减少不比要的更新和时间花费。针对此问题，DIHN [50] 训练了一个针对 query 的网络，其将数据分为旧数据，新增数据和 query 数据三个部分，其中旧数据的 code 是一直不变的，之后网络的训练由两部分构成：首先固定网络参数，输出新增数据的 code，之后固定新数据的 code，同时用新/旧数据的 code 通过 query 来对网络进行更新。其能保证不变动旧数据的情况下，逐渐调整新数据的方向，在不降低精度的同时大大提高了 online hashing 的效率。

## 5.2 相似/不相似对之间距离相差不够大

DCH [51] 指出：当 hashing 方法应用于工业检索时，一般会构造半径为  $r$  的 hamming ball，基于此建立 hashing table，使得给定 query，其半径  $r$  内的样本都是和其相似的样本， $r$  外都是不相似的，这样就能在  $O(1)$  内返回 hamming ball 内的所有样本（一般  $r$  设置的比较小，例如为 2）。但目前一些 hashing loss 中采用 sigmoid/tanh 函数的饱和区过长，导致相似样本之间无法足够接近。以 16bit 的 code 来举例，两个样本之间的距离降到 5 以内，loss 就已经很小了，导致模型会偏向于优化那些较难优化的样本，而不是继续拉近这两个样本的距离。

因此 DCH 将 sigmoid/tanh 函数替换成 Cauchy 函数，该函数在样本距离很小的时候仍有较大的优化空间，使得相似样本距离可以有效的拉近。但 DCH 存在的问题是生成的相似/不相似样本对之间距离不够大，举例来讲对于一个 query，相似样本和其距离分布在 0/1/2 附近，不相似样本和其距离分布在 2/3/4 附近，这会导致 hamming ball 内返回的样本不够准确。因此 MMHH [?] 设计了基于 t 分布的带 margin 的 loss，通过 margin 构造了相似/不相似样本对之间的 discrepancy.

## References

- [1] A. Gionis, P. Indyk, and R. Motwani, “Similarity search in high dimensions via hashing,” in *VLDB’99, Proceedings of 25th International Conference on Very Large Data Bases, September 7-10, 1999, Edinburgh, Scotland, UK* (M. P. Atkinson, M. E. Orlowska, P. Valduriez, S. B. Zdonik, and M. L. Brodie, eds.), pp. 518–529, Morgan Kaufmann, 1999. 2
- [2] Y. Weiss, A. Torralba, and R. Fergus, “Spectral hashing,” in *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008* (D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds.), pp. 1753–1760, Curran Associates, Inc., 2008. 2
- [3] Y. Gong and S. Lazebnik, “Iterative quantization: A procrustean approach to learning binary codes,” in *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pp. 817–824, IEEE Computer Society, 2011. 2
- [4] W. Liu, J. Wang, S. Kumar, and S. Chang, “Hashing with graphs,” in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011* (L. Getoor and T. Scheffer, eds.), pp. 1–8, Omnipress, 2011. 2
- [5] W. Liu, C. Mu, S. Kumar, and S. Chang, “Discrete graph hashing,” in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 3419–3427, 2014. 2, 6

- [6] Q. Jiang and W. Li, “Scalable graph hashing with feature transformation,” in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015* (Q. Yang and M. J. Wooldridge, eds.), pp. 2248–2254, AAAI Press, 2015. 2
- [7] M. Norouzi and D. J. Fleet, “Minimal loss hashing for compact binary codes,” in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011* (L. Getoor and T. Scheffer, eds.), pp. 353–360, Omnipress, 2011. 2
- [8] W. Liu, J. Wang, R. Ji, Y. Jiang, and S. Chang, “Supervised hashing with kernels,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pp. 2074–2081, IEEE Computer Society, 2012. 2, 4, 6
- [9] G. Lin, C. Shen, D. Suter, and A. van den Hengel, “A general two-step approach to learning-based hashing,” in *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pp. 2552–2559, IEEE Computer Society, 2013. 2, 6, 7
- [10] P. Zhang, W. Zhang, W. Li, and M. Guo, “Supervised hashing with latent factor models,” in *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’14, Gold Coast , QLD, Australia - July 06 - 11, 2014* (S. Geva, A. Trotman, P. Bruza, C. L. A. Clarke, and K. Järvelin, eds.), pp. 173–182, ACM, 2014. 2, 4
- [11] G. Lin, C. Shen, Q. Shi, A. van den Hengel, and D. Suter, “Fast supervised hashing with decision trees for high-dimensional data,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pp. 1971–1978, IEEE Computer Society, 2014. 2, 6
- [12] F. Shen, C. Shen, W. Liu, and H. T. Shen, “Supervised discrete hashing,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 37–45, IEEE Computer Society, 2015. 2, 3, 7
- [13] W. Kang, W. Li, and Z. Zhou, “Column sampling based discrete supervised hashing,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA* (D. Schuurmans and M. P. Wellman, eds.), pp. 1230–1236, AAAI Press, 2016. 2, 7
- [14] R. Raziperchikolaei and M. Á. Carreira-Perpiñán, “Optimizing affinity-based binary hashing using auxiliary coordinates,” in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain* (D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, eds.), pp. 640–648, 2016. 2, 7
- [15] M. Norouzi, D. J. Fleet, and R. Salakhutdinov, “Hamming distance metric learning,” in *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States* (P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1070–1078, 2012. 2, 5

- [16] J. Wang, J. Wang, N. Yu, and S. Li, “Order preserving hashing for approximate nearest neighbor search,” in *ACM Multimedia Conference, MM '13, Barcelona, Spain, October 21-25, 2013* (A. Jaimes, N. Sebe, N. Bouje-maa, D. Gatica-Perez, D. A. Shamma, M. Worring, and R. Zimmermann, eds.), pp. 133–142, ACM, 2013. [2](#), [5](#)
- [17] J. Wang, W. Liu, A. X. Sun, and Y. Jiang, “Learning hash codes with listwise supervision,” in *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pp. 3032–3039, IEEE Computer Society, 2013. [2](#), [5](#)
- [18] X. Li, G. Lin, C. Shen, A. van den Hengel, and A. R. Dick, “Learning hash functions using column generation,” in *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, vol. 28 of *JMLR Workshop and Conference Proceedings*, pp. 142–150, JMLR.org, 2013. [2](#)
- [19] Q. Wang, Z. Zhang, and L. Si, “Ranking preserving hashing for fast similarity search,” in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015* (Q. Yang and M. J. Wooldridge, eds.), pp. 3911–3917, AAAI Press, 2015. [2](#)
- [20] D. Song, W. Liu, R. Ji, D. A. Meyer, and J. R. Smith, “Top rank supervised binary coding for visual search,” in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pp. 1922–1930, IEEE Computer Society, 2015. [2](#), [5](#)
- [21] L. Liu, L. Shao, F. Shen, and M. Yu, “Discretely coding semantic rank orders for supervised image hashing,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 5140–5149, IEEE Computer Society, 2017. [2](#), [5](#)
- [22] J. Gui, T. Liu, Z. Sun, D. Tao, and T. Tan, “Fast supervised discrete hashing,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 490–496, 2018. [3](#), [7](#)
- [23] Q. Li, Z. Sun, R. He, and T. Tan, “Deep supervised discrete hashing,” in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA* (I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, eds.), pp. 2482–2491, 2017. [3](#), [7](#), [8](#)
- [24] Y. Chen, Z. Lai, Y. Ding, K. Lin, and W. K. Wong, “Deep supervised hashing with anchor graph,” in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 9795–9803, IEEE, 2019. [3](#)
- [25] J. Wang, T. Zhang, J. Song, N. Sebe, and H. T. Shen, “A survey on learning to hash,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 769–790, 2018. [4](#), [5](#), [6](#)
- [26] T. Do, A. Doan, and N. Cheung, “Learning to hash with binary deep neural network,” in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), vol. 9909 of *Lecture Notes in Computer Science*, pp. 219–234, Springer, 2016. [4](#)

- [27] Q. Jiang and W. Li, “Asymmetric deep supervised hashing,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018* (S. A. McIlraith and K. Q. Weinberger, eds.), pp. 3342–3349, AAAI Press, 2018. 4, 8
- [28] W. Li, S. Wang, and W. Kang, “Feature learning based deep supervised hashing with pairwise labels,” *CoRR*, vol. abs/1511.03855, 2015. 4, 6
- [29] H. Zhu, M. Long, J. Wang, and Y. Cao, “Deep hashing network for efficient similarity retrieval,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA* (D. Schuurmans and M. P. Wellman, eds.), pp. 2415–2421, AAAI Press, 2016. 4, 6
- [30] N. Li, C. Li, C. Deng, X. Liu, and X. Gao, “Deep joint semantic-embedding hashing,” in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden* (J. Lang, ed.), pp. 2397–2403, ijcai.org, 2018. 4, 6
- [31] H. Liu, R. Wang, S. Shan, and X. Chen, “Deep supervised hashing for fast image retrieval,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 2064–2072, IEEE Computer Society, 2016. 4, 5
- [32] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, “Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4766–4779, 2015. 5, 9
- [33] F. Zhao, Y. Huang, L. Wang, and T. Tan, “Deep semantic ranking based hashing for multi-label image retrieval,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 1556–1564, IEEE Computer Society, 2015. 5
- [34] K. He, F. Çakir, S. A. Bargal, and S. Sclaroff, “Hashing as tie-aware learning to rank,” in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 4023–4032, IEEE Computer Society, 2018. 5
- [35] F. Çakir, K. He, S. A. Bargal, and S. Sclaroff, “Mihash: Online hashing with mutual information,” in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 437–445, IEEE Computer Society, 2017. 5, 9
- [36] H. Lai, Y. Pan, Y. Liu, and S. Yan, “Simultaneous feature learning and hash coding with deep neural networks,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 3270–3278, IEEE Computer Society, 2015. 5, 9
- [37] Z. Cao, M. Long, J. Wang, and P. S. Yu, “Hashnet: Deep learning to hash by continuation,” in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 5609–5618, IEEE Computer Society, 2017. 6

- [38] Z. Chen, X. Yuan, J. Lu, Q. Tian, and J. Zhou, “Deep hashing via discrepancy minimization,” in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 6838–6847, IEEE Computer Society, 2018. 6
- [39] S. Su, C. Zhang, K. Han, and Y. Tian, “Greedy hash: Towards fast optimization for accurate hash coding in CNN,” in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada* (S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), pp. 806–815, 2018. 6
- [40] Y. Cao, M. Long, J. Wang, H. Zhu, and Q. Wen, “Deep quantization network for efficient image retrieval,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA* (D. Schuurmans and M. P. Wellman, eds.), pp. 3457–3463, AAAI Press, 2016. 6
- [41] B. Klein and L. Wolf, “End-to-end supervised product quantization for image search and retrieval,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 5041–5050, Computer Vision Foundation / IEEE, 2019. 6
- [42] S. Eghbali and L. Tahvildari, “Deep spherical quantization for image search,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 11690–11699, Computer Vision Foundation / IEEE, 2019. 6
- [43] J. Chen and W. K. Cheung, “Similarity preserving deep asymmetric quantization for image retrieval,” in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 8183–8190, AAAI Press, 2019. 6
- [44] D. Zhang, J. Wang, D. Cai, and J. Lu, “Self-taught hashing for fast similarity search,” in *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010* (F. Crestani, S. Marchand-Maillet, H. Chen, E. N. Efthimiadis, and J. Savoy, eds.), pp. 18–25, ACM, 2010. 6
- [45] B. Zhuang, G. Lin, C. Shen, and I. D. Reid, “Fast training of triplet-based deep binary embedding networks,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 5955–5964, IEEE Computer Society, 2016. 7
- [46] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, “Supervised hashing for image retrieval via image representation learning,” in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada* (C. E. Brodley and P. Stone, eds.), pp. 2156–2162, AAAI Press, 2014. 7
- [47] Q. Jiang, X. Cui, and W. Li, “Deep discrete supervised hashing,” *IEEE Trans. Image Process.*, vol. 27, no. 12, pp. 5996–6009, 2018. 8
- [48] F. Çakir, K. He, and S. Sclaroff, “Hashing with binary matrix pursuit,” in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part V* (V. Ferrari, M. Hebert,



C. Sminchisescu, and Y. Weiss, eds.), vol. 11209 of *Lecture Notes in Computer Science*, pp. 344–361, Springer, 2018. 8

- [49] Y. Guo, X. Zhao, G. Ding, and J. Han, “On trivial solution and high correlation problems in deep supervised hashing,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018* (S. A. McIlraith and K. Q. Weinberger, eds.), pp. 2240–2247, AAAI Press, 2018. 9
- [50] D. Wu, Q. Dai, J. Liu, B. Li, and W. Wang, “Deep incremental hashing network for efficient image retrieval,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 9069–9077, Computer Vision Foundation / IEEE, 2019. 9
- [51] Y. Cao, M. Long, B. Liu, and J. Wang, “Deep cauchy hashing for hamming space retrieval,” in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 1229–1237, IEEE Computer Society, 2018. 10