

应用集成理论的非监督哈希学习

李运强¹, 查宇飞¹, 杨 源², 张胜杰¹, 李 野²

(1.空军工程大学航空航天工程学院,西安,710038;2.空军工程大学空管领航学院,西安,710051)

摘要 针对以往算法在学习哈希函数过程引入随机性,导致得到的哈希码存在很大的差异性。该文以集成学习理论和并行计算方法为支撑,提出了一种非监督集成哈希学习(UEH)算法。首先,对于一些经典哈希算法,例如 SKLSH 和 ITQ,并不能获得唯一的汉明排序,因为在不同时刻学习得到的哈希函数并不唯一,亦即存在差异性;然后,运用集成学习算法去平衡哈希码之间的差异性,达到减少量化误差的效果;特别地,当基学习器满足高精度和较大差异性条件时,集成性能越高,因此,我们采用自举法,通过随机产生多组训练子集来增大差异性,从而进一步提高算法的泛化能力。在图像库 CIFAR-10 和 MINIST 上,运用该文算法进行图像检索,实验结果表明,该文算法的性能高于其他相关算法 6%~15%。

关键词 非监督;集成学习;并行计算;差异性;图像检索

DOI 10.3969/j.issn.1009-3516.2016.04.018

中图分类号 TN911.73 **文献标志码** A **文章编号** 1009-3516(2016)04-0095-06

Unsupervised Ensemble Hashing Learning

LI Yunqiang¹, ZHA Yufei¹, YANG Yuan², ZHANG Shengjie¹, LI Ye²

(1. Aeronautics and Astronautics Engineering College, Air Force Engineering University, Xi'an 710038, China; 2. Air Control and Navigation College, Air Force Engineering University, Xi'an 710051, China)

Abstract: The diversity existed in most recently hashing methods leads to the binary codes cannot efficiently preserve the data similarity. This paper, taking the ensemble learning theory and the parallel algorithm as a support, proposes a novel hashing method, i. e. Unsupervised Ensemble Hashing Learning (UEH). Firstly, the ensemble method is utilized to balance the diversity so as to reduce the quantization error. Specially, the higher accuracy and the larger diversity the base learner has, the more effective the ensemble method is. Then the bootstrap aggregating (bagging) method is used to increase the diversity. Finally, the paper uses iterative quantization to guarantee equivalent information of each hashing bits to effectively enhance the generalization ability. The paper validates the method on two large scale datasets CIFAR-10 and MINIST for image retrieval, and the experimental results show that the performance gains of the proposed method is improved by 6%~15% compared with the state-of-the-art methods. In addition, an important benefit of bagging scheme for hashing is inherently favorable to parallel computing.

Key words: unsupervised; ensemble learning; parallel computing; diversity; image retrieval

收稿日期:2016-01-28

基金项目:国家自然科学基金(61472442; 61473309); 陕西省科技新星基金(2015kjxx-46)

作者简介:李运强(1993-),男,河南商丘人,硕士生,主要从事模式识别研究.E-mail:Mryunqiang@126.com

通信作者:查宇飞(1979-),男,陕西西安人,副教授,博士,主要从事模式识别研究.E-mail:Zhayufei@126.com

引用格式:李运强,查宇飞,杨源,等.应用集成理论的非监督哈希学习[J].空军工程大学学报:自然科学版,2016,17(4):95-100. LI Yunqiang ZHA Yufei YANG Yuan, et al. Unsupervised Ensemble Hashing Learning[J]. Journal of Air Force Engineering University: Natural Science Edition, 2016, 17(4): 95-100.

由于哈希的低存储耗费和高查询速度,哈希学习近年来被广泛应用于大数据的近似最近邻搜索^[1-3]。哈希学习的目的是学到数据的二进制哈希码表示,使得哈希码尽可能地保持原空间中的近邻关系,即保相似性。从哈希技术的角度来看,现有的哈希学习模型可分为数据独立方法和数据依赖方法。

数据独立方法学习得到的哈希函数与训练样本无关,具有代表性的数据独立方法包括局部敏感哈希(Locality Sensitive Hashing, LSH)^[4]算法和它的拓展算法核化局部敏感哈希(Kernelized Locality-Sensitive Hashing, KLSH)^[5]及平移不变核化哈希(Locality-Sensitive Binary Codes From Shift-Invariant Kernels Hashing, SKLSH)^[6]。LSH通过随机生成一个投影矩阵,作为哈希函数,将高维数据映射为汉明空间中的低维数据;KLSH和SKLSH是局部敏感哈希的核扩展算法。数据依赖方法只需用较短的哈希编码位数就能取得理想的精度,从而进一步提高检索和学习效率,降低存储和通信开销。语义哈希(Semantic Hashing)^[7]运用原始样本的标签信息,通过训练一个多层受限波兹曼机将原始样本从高维空间映射到低维哈希空间。谱哈希(SH)^[8]挖掘数据内部的谱性质,通过约束哈希编码中某一位(bit)结果为-1或1的概率相等以及哈希码的每一位不相关,然后运用谱方法得到哈希码。

根据哈希函数生成过程是否存在随机变化,现有非监督哈希学习算法可分为随机性哈希算法和非随机性哈希算法。随机性哈希算法LSH、迭代量化方法(ITQ)^[9]和SKLSH等,它们在学习哈希函数的过程中都存在一定的随机性,如一些算法通过随机生成投影函数作为哈希投影函数,另外一些运用随机初始化的方式生成一系列哈希函数;非随机性哈希算法表示任意2次学习得到的哈希函数相同,不存在差异性,如主成分分析哈希(PCAH)^[10]算法等。本文主要关注于非监督随机性哈希算法。

由于哈希学习过程本身存在的随机性,导致得到的哈希码不能很好地保持原始数据的相似性。本文同时考虑哈希码的随机性和投影数据的各维方差,提出了一种非监督集成哈希学习(Unsupervised Ensemble Hashing Learning, UEH)算法。

1 集成哈希学习(Ensemble Hashing)

在这一小节中,我们首先对以往算法进行差异性分析,然后,引入集成学习^[11]理论。

1.1 差异性分析

现有大多数非监督哈希算法通过平衡每一维单

位编码信息来提高检索性能。然而,由于哈希学习过程本身存在的随机性,导致得到的哈希码不能很好地保持原始数据的相似性。

局部敏感哈希(LSH)随机生成投影函数 $U = \{u_i\}_{i=1}^k$,然后通过 $h(x) = \text{sgn}(xu)$ 将样本从实值空间投影到哈希空间;迭代量化哈希(ITQ)首先由主成分分析(PCA)^[12]得到投影数据各主成分方向组成投影矩阵 $V = \{v_i\}_{i=1}^k$,初始化随机生成旋转矩阵 R_0 ,然后通过迭代量化方式寻找最优的正交旋转矩阵 R ,使得量化误差最小,接着通过 $Y = \text{sgn}(XVR)$ 进行单位编码得到哈希码。以上提到的2种哈希算法,在学习哈希函数的过程中,都存在一定的随机性。因此,任意2次学习得到的哈希函数都有差异。

我们选取一种随机性哈希算法作为基准算子,进行多次学习生成 K 组不同的哈希投影函数 $W = [W^{(1)}, W^{(2)}, \dots, W^{(K)}]$,其中 $W^{(p)} = \{w_i\}_{i=1}^k$ 。采用符号函数 $g(u) = \text{sgn}(u)$ 进行编码得 $H^p(X) = \text{sgn}(XW^{(p)})$ 。任意2组哈希码的差异性大小可以由定理1^[13]来衡量:

定理1 任取 $u, v \in [-1, +1]$,则可得到: $\{g(u) \neq g(v)\} = |u - v|/2$ 。

由定理1,我们可以计算上述学习得到的 K 组哈希码之间的平均差异性大小为:

$$\{H^p(X) \neq H_1(X)\} = \frac{1}{2Klk} \sum_{p=2}^K |XW^{(p)} - XW^{(1)}| = \frac{1}{2Klk} \sum_{p=2}^K |X[W^{(p)} - W^{(1)}]| \tag{1}$$

式中: l 为测试样本个数, k 为每一组哈希函数的个数。可以看出,哈希码的平均差异性大小取决于哈希函数的差异性大小。

图1表示LSH、PCA-RR和ITQ3种随机性哈希算法通过多次学习生成的哈希码之间的平均差异性大小示意图。

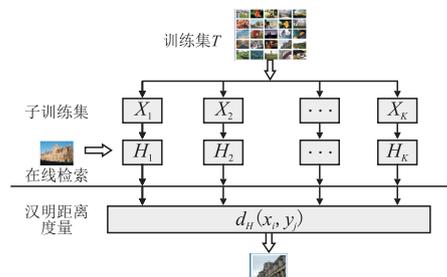


图1 不同哈希算法生成的哈希码之间平均差异性大小

Fig.1 Average diversity among Hash of different methods

从图1可以看出,局部敏感哈希(LSH)的差异性大于PCA-RR,而PCA-RR的差异性又略大于迭代量化哈希(ITQ)。

1.2 集成哈希学习

集成学习通过训练多组有差异的学习器,并将它们的预测结果进行合成。相对于单个学习器,集成学习算法在大多数情况下可以显著提高学习系统的泛化能力,因此对集成学习理论和算法的研究逐渐成为了机器学习领域的一个热点,同时,集成学习也从传统的单任务学习领域扩展到多任务学习。

图 2 表示在 CIFAR-10^[14] 和 MINIST^[15] 2 个数据库上,当哈希码为 32 维时,对图 2 中提到的 3 种随机性算法进行集成哈希学习,查准率随集成学习器个数 K 变化示意图;可以看到,当集成学习器个数 $K \geq 8$ 时,集成性能趋于平稳,该过程不会出现过拟合,基于集成学习支持并行计算的优点,为了方便多核计算,本文选取 $K = 8$;对于不同数据库而言,其性能变化又有一些差别。在 CIFAR-10 数据库上,虽然迭代量化哈希(ITQ)算法的单体性能略高于其他 2 种算法,但随着集成个数的增加,被其他 2 种算法超越,见图 2(a);在 MINIST 数据库上,由于迭代量化哈希(ITQ)算法的单体性能远高于其他 2 种算法,随着集成次数的增加,其他 2 种算法并不能超过它,见图 2(b)。

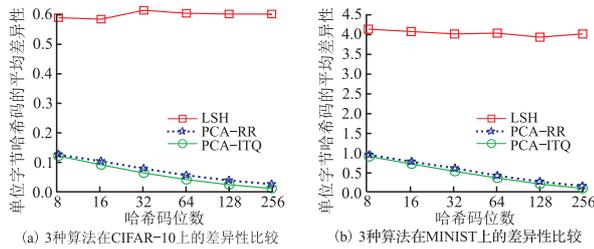


图 2 查准率随集成学习器个数变化示意图
Fig.2 Precision with the number of ensemble learning machines

总结图 1 和图 2,我们可以得到集成性能提升的原则:一是保证单个学习器的精度;二是保证各个学习器之间的差异性,差异性越大则性能提升越明显。图 3 为影响集成性能的原因示意图。

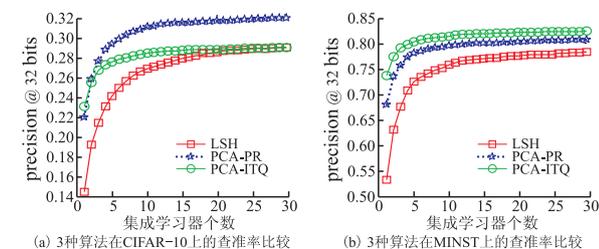


图 3 影响集成性能的原因示意图

Fig.3 Map of reasons affecting the performance of assembling

下文将介绍利用自举法(bootstrap)^[16-17]生成多组训练子集来增大集成的各组哈希函数之间的差异性方法,这种方法称为 Bootstrap Aggregating,简称

Bagging。

2 基于集成理论的非监督哈希学习

UEH 算法运用随机抽样的方式(Bagging)^[18]增加学习器间的差异性,然后通过集成学习理论平衡算法的差异性,最后运用迭代优化的方式,保证每一维的单位编码信息相同,从而提高算法的能力。图 4 为该算法流程示意图。

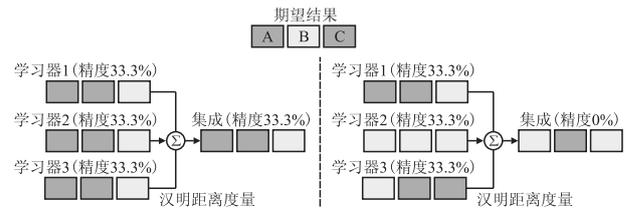


图 4 算法流程

Fig.4 The flow of our algorithm

算法的具体过程如下:套袋(Bagging)算法是集成学习的主要技术之一。该算法引入随机性,具有更好的抗噪声能力,不容易出现过拟合;同时,套袋(Bagging)算法支持并行计算,运行速度比较快。首先介绍一下符号系统,设有 n 原始样本数据 $\mathbf{X} \in \mathbb{R}^{n \times m}$, m 为样本的维度。我们通过随机抽样(Bootstrap Sampling)来获得训练集的不同子集 $\{X_t \in \mathbb{R}^{T \times m}\}$, $t = 1, 2, \dots, K$, 其中 T 表示样本子集的个数。对应训练样本子集上学习多组哈希函数 $\Phi(x) = \{\mathbf{H}_t(x)\}$, $t = 1, 2, \dots, K$, 其中 $\mathbf{H}_t(x) = [h_1^{(t)}(x), h_2^{(t)}(x), \dots, h_k^{(t)}(x)]^T \in \{-1, 1\}^k$, k 为哈希函数的个数。

2.1 模型构建

哈希学习的目的是为了得到哈希码阵列保持数据在原始空间的相似性,对于样本 x_i 和 x_j ,如果它们在欧式空间中相似,我们希望其投影到汉明空间后仍然保持相似性,即汉明距离应保持最小:

$$\begin{aligned} \text{minimize } D &= \sum_{t=1}^K w_t \cdot d_H^{(t)}(\mathbf{H}_t(x_i), \mathbf{H}_t(x_j)) \\ \sum_t w_t &= 1, w_t > 0 \end{aligned} \quad (2)$$

式中: $d_H^{(t)}(\cdot)$ 表示第 t 组哈希函数生成的哈希码之间的汉明距离, $\mathbf{H}_t(x_i) = [h_1^{(t)}(x_i), h_2^{(t)}(x_i), \dots, h_k^{(t)}(x_i)]$ 和 $\mathbf{H}_t(x_j) = [h_1^{(t)}(x_j), h_2^{(t)}(x_j), \dots, h_k^{(t)}(x_j)]$, 在 bagging 学习中,所有基学习器(哈希函数组)被赋予相同权值,本文选取权重值 $w_t = 1/K$ 。

基于 bagging 算法构建哈希函数的过程不适用于数据独立方法,因为这些算法不涉及样本的随机抽样过程,本文选取具有随机性的数据依赖方法作

为基本算子,构建优化模型:

$$(\mathbf{H}_t, \mathbf{Y}_t) = F_M(\mathbf{X}_t), t = 1, 2, \dots, K \quad (3)$$

式中: \mathbf{H}_t 为第 t 组哈希函数; \mathbf{Y}_t 为第 t 组二进制哈希码; M 表示选取的基准哈希算子。

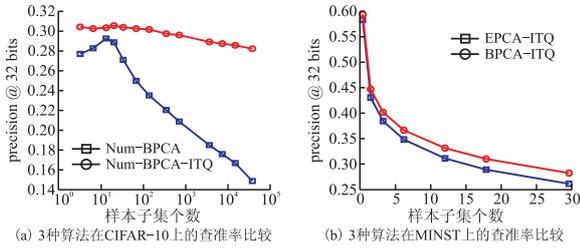


图5 在数据库 CIFAR-10 和 MNIST 上,集成性能随每次选取的样本子集个数 T 变化示意图

Fig.5 Precision with the number of subsets T on two datasets

图 5 的实验结果是在数据库 CIFAR-10 和 MNIST 上进行的,哈希码长度选为 32。图 5(a) 表示选取 PCAH 和迭代量化哈希(ITQ)作为基准算子,集成性能随着随机抽样个数 T 变化示意图。可以看出,对于迭代量化哈希(ITQ)当 T 等于 30 时,检索性能最佳;图 5(b) 表示采用 bagging 算法随机选取样本子集的算法(BPCA-ITQ)相对于固定训练样本对应算法(BPCA-ITQ)性能提升示意图,其中选取 $T = 30$ 。

2.2 模型求解

根据上面讲到的集成学习思想,基本哈希算子的选取原应满足单体精度高以及差异性尽量大的条件。本文选取迭代量化哈希作为模型(3)的基算子,在任意子训练集 \mathbf{X}_t 进行哈希函数求解:

$$\begin{aligned} \max_{\mathbf{H}} & \text{tr}(\mathbf{W}_t^T \mathbf{X}_t^T \mathbf{X}_t \mathbf{W}_t) \\ \text{s.t.} & \mathbf{W}_t^T \mathbf{W}_t = \mathbf{I} \end{aligned} \quad (4)$$

和以往很多算法一样,我们可以得到投影数据各主成分方向组成投影函数 \mathbf{W}_t , 然而,这样得到的各主成分方向的方差(或信息量)通常各不相等。

迭代量化算法通过旋转主成分方向使得旋转后各方向的方差尽量保持平衡。该方法的目标是通过寻求最优正交矩阵 \mathbf{R} , 最小化量化误差:

$$\begin{aligned} \min_{\mathbf{R}} & Q(\mathbf{Y}, \mathbf{R}) = \|\mathbf{Y}_t - \mathbf{VR}\|_F^2 \\ \text{s.t.} & \mathbf{R}^T \mathbf{R} = \mathbf{I} \end{aligned} \quad (5)$$

式中: \mathbf{Y}_t 为单位编码得到的哈希码 $\mathbf{Y}_t = \text{sgn}(\mathbf{VR})$ 。

采用迭代量化方法去逼近量化误差最小值。其交替进行的步骤将在下面阐述。

Step1 初始化 \mathbf{R} 为随机正交矩阵。

Step2 固定 \mathbf{R} , 更新 $\mathbf{Y}_t = \text{sgn}(\mathbf{X}_t \mathbf{W}_t^* \mathbf{R})$ 。

Step3 固定 \mathbf{Y}_t , 更新 \mathbf{R} 。目标函数变为:

$$\min_{\mathbf{R}^T \mathbf{R} = \mathbf{I}} \|\mathbf{Y}_t - \mathbf{VR}\|_F^2 \quad (6)$$

式(6)属于正交普克拉提斯问题(Orthogonal procrustes problem, OP)^[3], 当 \mathbf{Y}_t 固定时,求解目标函数(11)的步骤:首先对 $k \times k$ 矩阵 $\mathbf{Y}_t^T \mathbf{V}$ 进行奇异值分解 $\text{SVD}(\mathbf{Y}_t^T \mathbf{V}) = \mathbf{S} \mathbf{\Omega} \bar{\mathbf{S}}^T$, 其中 $\text{SVD}[\cdot]$ 表示奇异值分解,接着得到 $\mathbf{R} = \bar{\mathbf{S}} \mathbf{S}^T$ 。

Step4 当量化误差足够小时,所得到的正交矩阵 \mathbf{R} 和哈希码 \mathbf{Y}_t 即为所求,实验表明当迭代次数在 40~60 时,算法趋于平稳,本文选择迭代 50 次。

我们可以求得最优正交旋转矩阵 \mathbf{R} 及 $\mathbf{W}_t = \mathbf{W}^* \mathbf{R}$, 最终进行哈希编码 $\mathbf{Y}_t = \text{sgn}(\mathbf{X}_t \mathbf{W}_t)$ 。采用这种方式在每一个样本子集进行迭代量化最终求解出 K 组哈希函数。

3 实验结果及分析

3.1 数据集与实验设置

在本文实验中,选用比较常用的 CIFAR-10^[14] 和 MNIST^[15] 数据库来验证本文算法的有效性。

CIFAR-10 含有 60k 个样本,每个样本维度为 320 维,该数据库包含飞机、汽车、鸟、猫、鹿、狗、青蛙、马、轮船和卡车 10 个分类,每类数据集有 6 000 个样本。MNIST 是数字手写识别数据库,该数据库中包含 70k 个维度为 784 的数据样本。我们在不同的数据库上,设置不同的训练子集样本个数 T , CIFAR-10^[17] 随机选取 $T = 30$, MNIST 随机选取 $T = 10k$ 。数据库具体信息见表 1。

表 1 实验所用数据库具体信息

Tab.1 Specific information of the experimental database

数据库	测试集	分类	维度
CIFAR-10	1k	10	320
MNIST	1k	10	784

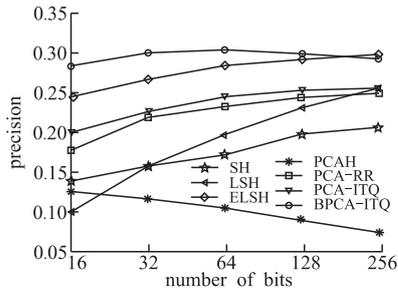
本文所有实验在 3.06 GHz, 6 G 内存的 64 位计算环境下通过 MATLAB 2013 软件平台仿真实现。

3.2 对比算法及评价标准

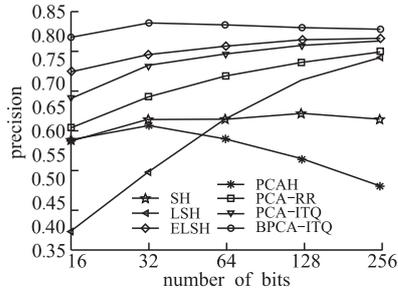
为验证本文算法的有效性,试验中选用非监督哈希算法 LSH^[4]、ITQ^[9]、谱哈希(SH)^[8] 以及我们上文运用到的集成哈希算法 ELSH。实验采用的评价指标为查准率(Precision)和查准率-召回率(Precision-recall)。考虑多核并行计算,本文选取集成次数 $K = 8$ 。

3.3 实验结果分析

图 6 表示在数据库 CIFAR-10 和 MNIST 上,当返回 500 幅检索图像时,查准率(Precision)随哈希码维度变化示意图。



(a) 查准率在CIFAR-10上的变化



(b) 查准率在MNIST上的变化

图 6 数据库 CIFAR-10 和 MNIST 上,查准率随哈希码维度变化示意图

Fig.6 Precision of top 500 returned of different methods on two datasets

由图可知,在哈希码不超过 64 维时,算法 BPCA-ITQ 的查准率随着哈希码维度的增加呈不断上升趋势,相对于基准算子 PCA-ITQ 在性能上得到了大幅度的提升;而当哈希码维度大于 64 时,查准率却有所下降,结合图 2 可知,随着哈希码维度的增加,虽然单体精度呈上升趋势,但学习器之间的差异性不断下降,集成精度也呈下降趋势;对于 ELSH 算法而言,由图 2 可知,其差异性在所有维度上都保持较大,当单体精度不断增加时,集成性能在所有维度上呈现增长趋势。

图 7 表示在数据库 CIFAR-10 上,哈希码分别为 16、32、64 和 128 维时,查准率与返回检索图像关系示意图。当哈希码为 16 维时,算法 BPCA-ITQ 相对于基准算子 PCA-ITQ 得到了 8% 的性能提升, ELSH 算法相对于基准算子 LSH 有 15% 的性能提升,且远高于 PCA-ITQ;在 128 维时, BPCA-ITQ 的性能呈现下降趋势。

图 8 是在数据库 MNIST 上,哈希码分别为 16、32、64 和 128 维时,查准率和查全率关系示意图,它表明集成算法的性能得到了大幅度提升。

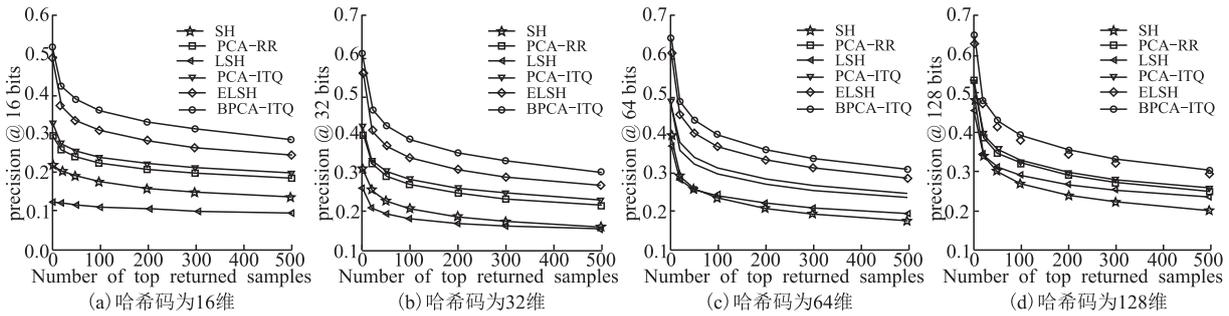


图 7 数据库 CIFAR-10 上查准率与返回检索图像关系示意图

Fig.7 Precision with the number of top returned samples on CIFAR-10

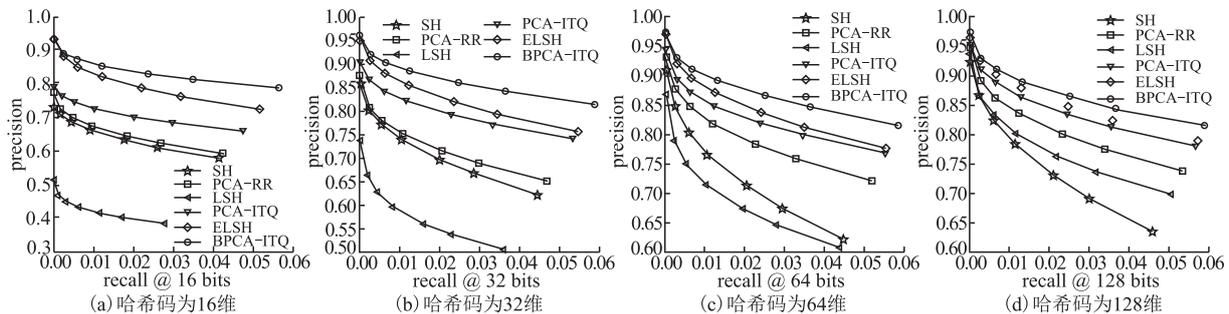


图 8 数据库 MNIST 上查准率和查全率关系示意图

Fig.8 Precision-recall curves on two datasets with bits of 16、32、64、128

以上都是在哈希码维数相同时,对所有相关算法进行的横向比较;下面采用一种纵向比较方式,即固定总的哈希码长度,对其检索性能及时间复杂度进行分析,对于本文采用 8 个集成器而言,当单个学习器的维度为 32 时,总的哈希码长度为 256,与相关算法 PCA-ITQ 在 256 维进行比较,其性能在 2

个数据库上都得到了 5% 左右的提升;而根据上述关于时间复杂度分析,当采用多核并行计算时,我们的算法相对于 ITQ 在时间上快了接近 8 倍,运行速度比较快。综合以上分析,无论从横向还是纵向进行分析,我们的算法在性能和时间上都得到了很大的提升。

4 结语

本文以集成学习理论和并行计算方法为支撑,提出了一种非监督集成哈希学习(UEH)算法。在文中,我们提炼了集成学习性能提升的2条原则,即当基学习器满足高精度和较大差异性时,集成性能提升越明显,另外,我们采用自举法,通过随机产生多组训练子集来增大差异性,从而进一步提高算法的泛化能力。实验结果表明,该文算法的性能高于其他相关算法。接下来,我们将继续深入分析哈希函数之间的差异性,从而设计一种单精度高,差异性大的哈希算法

参考文献(References):

- [1] LI Wujun, KONG Weihao, ZHANG Dongqing, et al. Learning to Hash with Its Application to Big Data Retrieval and Mining [EB/OL]. [2013-12-21] cs.nju.edu.cn/lwj/slides/hash2.pdf.
- [2] GE Tiezheng, HE Kaiming, KE Qifa, et al. Optimized Product Quantization for Approximate Nearest Neighbor Search [C]// CVPR, Portland, OR: IEEE, 2013: 2946-2953.
- [3] KALANTIDIS Yannis, AVRITHIS Yannis. Locally Optimized Product Quantization for Approximate Nearest Neighbor Search [C]// CVPR, Columbus, OH: IEEE, 2014: 2329 - 2336.
- [4] GIONIS A, INDYK P, MOTWANI R. Similarity Search in High Dimensions via Hashing [C]// International Conference on Very Large Data Bases, San Francisco, CA: Morgan Kaufmann Publishers Inc, 1999: 518-529.
- [5] KULIS B, GRAUMAN K. Kernelized Locality-Sensitive Hashing [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2012, 34(6): 1092-1104.
- [6] RAGINSKY M, LAZEBNIK S. Locality-Sensitive Binary Codes from Shift-Invariant Kernels [C]// 23rd Annual Conference on Neural Information Processing Systems, Vancouver, Canada, 2009: 1509-1517.
- [7] SALAKHUTDINOV R, HINTON G E. Semantic Hashing [J]. International Journal of Approximate Reasoning, 2009, 50(7): 969-978.
- [8] WEISS Y, TORRALBA A, FERGUS R. Spectral hashing [C]// Conference on Advances in Neural Information Processing Systems, Vancouver, Canada, 2008: 1753-1760.
- [9] GONG Yunchao, LAZEBNIK S, GORDO A. Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large Scale Image Retrieval [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(12): 2916-2929.
- [10] WANG J, KUMAR S, CHANG S F. Semi-Supervised Hashing for Large-Scale Search [J]. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2012, 34(12): 2393-2406.
- [11] OPITZ D, MACLIN R. Popular Ensemble Methods: An Empirical Study [J]. Journal of Artificial Intelligence Research, 1999, 11: 169 - 198.
- [12] JOLLIFFE I T. Principal Component Analysis [M]. 2nd Ed. New York: Springer-Verlag, 1986.
- [13] RAGINSKY M, LAZEBNIK S. Locality Sensitive Binary Codes from Shift-Invariant Kernels [C]// Neural Information Processing Systems, Vancouver, Canada, 2009: 1509-1517.
- [14] KRIZHEVSKY A. Learning Multiple Layers of Features from Tiny Images [D]. Toronto: University of Toronto, 2009.
- [15] BENGIO Y, BOTTOU L, HAFFNER P, et al. Gradient Based Learning Applied to Document Recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
- [16] EFRON B. Bootstrap Methods: Another Look at the Jackknife [J]. The Annals of Statistics, 1979, 7(1): 1-26.
- [17] EFRON B, TIBSHIRANI R. An Introduction to the Bootstrap [J]. Teaching Statistics, 2001, 23(2): 49-54.
- [18] BREIMAN L. Bagging Predictors [J]. Machine Learning, 1996, 24(2): 123 - 140.

(编辑:徐楠楠)