

文章编号:1007-5321(2019)02-0036-06

DOI:10.13190/j.jbupt.2018-116

基于主成分分析的图像哈希算法

赵 珊, 李永思

(河南理工大学 计算机科学与技术学院, 河南 焦作 454003)

摘要: 提出了一种基于主成分分析的图像哈希算法. 采用主成分分析对样本进行降维, 取位于变换矩阵顶端最具有识别信息的少量特征向量构造投影矩阵, 再对降维后样本进行局部保持映射, 同时, 对主成分分析投影矩阵进行随机旋转, 形成多个小投影矩阵, 采用矩阵拼接方法将小投影矩阵合并构造编码投影矩阵; 最后, 将训练样本投影到编码投影矩阵, 得到降维样本, 并对其进行哈希编码, 得到最终的二进制编码. 实验结果证明, 同其他经典算法相比, 该算法具有较好的稳定性, 可降低内存消耗, 并提高效率.

关键词: 哈希; 主成分分析; 局部保持投影; 图像处理

中图分类号: TP391.41

文献标志码: A

Imaging Hashing Based on Principal Component Analysis

ZHAO Shan, LI Yong-si

(College of Computer Science and Technology, Henan Polytechnic University, Henan Jiaozuo 454003, China)

Abstract: A novel image Hashing based on principal component analysis (PCA) was proposed. PCA was introduced to reduce dimension of samples, and the projection matrix was achieved by choosing several eigenvectors which have higher recognition ability. Based on which, the reduced-sample was mapped with locality preserving projection (LPP). Meanwhile, the projection matrix of principal component analysis was randomly rotated to form a series of transformational matrixes. The matrix stitching was adopted to construct the final code projection matrix. Finally, the original samples were projected into the code projection matrix to get a reduced dimensional sample, and the Hashing code was used to achieve the final binary encoding. Experiments show that the proposed method has better stability, lower memory consumption and higher efficiency compared with other traditional methods.

Key words: Hashing; principal component analysis; locality preserving projection; image processing

哈希技术能够将原特征编码成紧致的二值哈希码, 大幅降低内存消耗, 同时可以缩减查询所需时间, 近年来备受关注. 基于数据独立的局部敏感哈希 (LSH, locality sensitive Hashing)^[1] 及其改进算法被认为是在高维空间中快速搜索的重要突破之一, 但是随着编码位数的增加会降低检索效率, 同时, 多

个哈希表还会增大 CPU 的存储空间和查询响应时间. 数据依赖哈希技术^[2-9] 主要依赖原始数据的内部结构把高维数据投影到低维空间中, 通常被证明比数据独立的 LSH 更有效.

提出了一种基于主成分分析 (PCA, principal component analysis)^[10] 的局部保持哈希算法 RPLPH

收稿日期: 2018-06-20

基金项目: 河南省高等学校重点科研项目 (18B520017); 河南理工大学博士基金项目 (B2014-043)

作者简介: 赵珊 (1975—), 女, 副教授.

通信作者: 李永思 (1994—), 女, 硕士生, E-mail: yongsili888@126.com.

(rotation PCA locality preserving Hashing). 首先采用 PCA 对样本进行降维, 取位于变换矩阵顶端最具有识别信息的少量特征向量构造 PCA 投影矩阵, 在此基础上对 PCA 降维样本进行局部保持投影(LPP, locality preserving projection)^[11], 同时利用随机正交矩阵对 PCA 降维矩阵进行多次随机旋转, 构造多个 LPP 投影矩阵, 然后融合二进制编码分段思想^[12], 把这些小矩阵拼接构造最终的投影矩阵. 最后将原始样本投影到该矩阵上进行哈希编码. 算法中, 把 PCA 和 LPP 有效结合, 保留了原始数据局部和全局的相似性结构, 而且把随机旋转应用于 PCA 数据的顶部主要投影中, 减少了编码之间的量化误差, 使得投影矩阵更加多样化. 同时, 流形子空间特征提取与随机旋转的巧妙融合在很大程度上提高了处理效率. 大量实验结果证明了该方法的可行性和高效性.

1 相关理论

1.1 PCA

PCA^[10]是一种经典的无监督线性子空间特征提取方法. 优化目标函数可写成:

$$\operatorname{argmax}_W (\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W}) \text{ s. t. } \mathbf{W}^T \mathbf{W} = \mathbf{I} \quad (1)$$

其中: $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \in \mathbf{R}^{d \times n}$ 为样本矩阵, n 和 d 分别表示样本点的数目和样本点的维数; \mathbf{W}^T 为 \mathbf{W} 的转置矩阵, 在约束条件 $\mathbf{W}^T \mathbf{W} = \mathbf{I}$ 下, 满足最大化目标函数的矩阵 \mathbf{W} 就是经过 PCA 降维后得到的投影矩阵. 利用拉格朗日乘子法可得到如下的广义特征值求解问题:

$$\mathbf{X} \mathbf{X}^T \mathbf{w}_i = \lambda_i \mathbf{w}_i \quad (2)$$

假定 $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_t$ 为式(2)最大的 t 个特征值对应的特征向量, 则 PCA 投影矩阵表示为 $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_t] \in \mathbf{R}^{d \times t}$. 样本集 \mathbf{X} 的低维特征表示为 $\mathbf{y}_i = \mathbf{W}^T \mathbf{x}_i, i = 1, 2, \dots, n$.

1.2 LPP

LPP^[11]作为一种基于流形的子空间提取方法, 其基本思想就是寻找一个投影矩阵 \mathbf{V} , 将高维空间 \mathbf{R}^d 中的样本集 $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ 映射为低维空间 $\mathbf{R}^t (t < d)$ 中的样本集 $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n]$, 即 $\mathbf{y}_i = \mathbf{V}^T \mathbf{x}_i, i = 1, 2, \dots, n$, 而且在 \mathbf{R}^d 空间内互为近邻的两点经 \mathbf{V} 映射后在 \mathbf{R}^t 空间中仍互为近邻. 其目标函数为

$$\sum_j (\mathbf{y}_i - \mathbf{y}_j)^2 \mathbf{M}_{ij} \quad (3)$$

通过简单的代数运算, 目标函数(式(3))可以简

化为

$$\begin{aligned} & \frac{1}{2} \sum_{ij} (\mathbf{y}_i - \mathbf{y}_j)^2 \mathbf{M}_{ij} = \\ & \frac{1}{2} \sum_{ij} (\mathbf{V}^T \mathbf{x}_i - \mathbf{V}^T \mathbf{x}_j)^2 \mathbf{M}_{ij} = \\ & \mathbf{V}^T \mathbf{X} (\mathbf{D} - \mathbf{M}) \mathbf{X}^T = \\ & \mathbf{V}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{V} \end{aligned} \quad (4)$$

其中: \mathbf{M}_{ij} 为近邻图的权值矩阵, 如果 \mathbf{x}_i 与 \mathbf{x}_j 之间有一条边相连, $\mathbf{M}_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\delta^2)$; 否则 $\mathbf{M}_{ij} = 0$; 矩阵 \mathbf{D} 是一个对角矩阵, 其值为权值矩阵 \mathbf{M} 每一行或每一列的数据元素之和, 即 $\mathbf{D}_{ii} = \sum_j \mathbf{M}_{ij}$; $\mathbf{L} = \mathbf{D} - \mathbf{M}$ 为近邻图的拉普拉斯矩阵. 在约束条件 $\mathbf{V}^T \mathbf{X} \mathbf{D} \mathbf{X}^T \mathbf{V} = 1$ 的情况下, 最小化目标函数(式(4)), 得到最终目标函数为

$$\operatorname{argmin}_V \mathbf{V}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{V} \text{ s. t. } \mathbf{V}^T \mathbf{X} \mathbf{D} \mathbf{X}^T \mathbf{V} = 1 \quad (5)$$

经 LPP 投影后的变换矩阵就是式(5)求解后最小的 t 个特征值对应特征向量构成的 $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_t\}$.

2 基于 PCA 的图像哈希

假设给定一个数据 $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \in \mathbf{R}^{d \times n}$, 构造基于 PCA 的图像哈希具体步骤主要包括 3 部分.

2.1 PCA 降维

对样本的协方差矩阵 $\mathbf{X} \mathbf{X}^T$ 进行 PCA 降维处理, 在 PCA 中, 数据的大部分信息是由少量较大的特征向量捕获的, 降维后的较小特征值对应的特征向量往往是噪音. 因此算法中只保留 \mathbf{W} 顶端 $r (r < t)$ 个特征向量, 得到正交矩阵 \mathbf{A} . 将该样本投影到正交矩阵 \mathbf{A} 上得到 PCA 降维后的样本 \mathbf{Y} , 即

$$\mathbf{Y} = \mathbf{A}^T \mathbf{X} \quad (6)$$

其中: $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r]$, $\mathbf{A} \in \mathbf{R}^{d \times r}$. 然后利用随机正交矩阵 $\mathbf{R}^{(i)} \in \mathbf{R}^{r \times r} (i = 1, 2, \dots, K)$ 对 \mathbf{A} 进行旋转. 其中随机旋转一方面使得哈希编码间的量化误差最小; 另一方面增加了 \mathbf{A} 顶端特征向量的多样性. 此时把得到的正交矩阵记为

$$\mathbf{B} = \mathbf{A} \mathbf{R}^{(i)}, i = 1, 2, \dots, K \quad (7)$$

其中 $K = t/r$ 表示进行 K 次旋转.

2.2 LPP 投影

算法中将式(6)中的投影矩阵 $\mathbf{Y} \in \mathbf{R}^{r \times n}$ 进行 LPP 变换, 构造最终的变换矩阵, 其中 \mathbf{Y} 是经过 PCA 投影之后得到的降维样本. 把 \mathbf{Y} 代入式(5)中得出

新的目标函数如下:

$$\operatorname{argmin}_C \mathbf{C}^T \mathbf{Y} \mathbf{L} \mathbf{Y}^T \mathbf{C} \text{ s. t. } \mathbf{C}^T \mathbf{Y} \mathbf{D} \mathbf{Y}^T \mathbf{C} = 1 \quad (8)$$

对式(8)求解后, \mathbf{C} 表示最小的 r 个特征值对应的特征向量 $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_r\}$. 利用拉格朗日乘子法可将式(8)化为

$$L(c, \lambda) = \mathbf{C}^T \mathbf{Y} \mathbf{L} \mathbf{Y}^T \mathbf{C} - \lambda (\mathbf{C}^T \mathbf{Y} \mathbf{D} \mathbf{Y}^T \mathbf{C} - 1) \quad (9)$$

接着,对 $L(c, \lambda)$ 中 c 和 λ 进行求导:

$$\frac{\partial L}{\partial c} = \mathbf{Y} \mathbf{L} \mathbf{Y}^T \mathbf{C} - \lambda \mathbf{Y} \mathbf{D} \mathbf{Y}^T \mathbf{C} = 0 \quad (10)$$

$$\frac{\partial L}{\partial \lambda} = \mathbf{C} \mathbf{Y} \mathbf{L} \mathbf{Y}^T \mathbf{C} - 1 = 0 \quad (11)$$

最终结果为

$$\mathbf{Y} \mathbf{L} \mathbf{Y}^T \mathbf{C} = \lambda \mathbf{Y} \mathbf{D} \mathbf{Y}^T \mathbf{C} \quad (12)$$

式(6)和式(12)联立求得新投影矩阵 $\mathbf{Q} \in \mathbf{R}^{r \times r}$:

$$\mathbf{Q} = \mathbf{A} \mathbf{C} \quad (13)$$

用 \mathbf{B} 代替式(13)中的 \mathbf{A} . 式(13)便化为

$$\mathbf{Q}^{(i)} = \mathbf{A} \mathbf{R}^{(i)} \mathbf{C}, i = 1, 2, \dots, K \quad (14)$$

得到含有 r 个特征向量的 K 个投影矩阵块. 把 K 个投影矩阵块 $\mathbf{Q}^{(i)}$ 拼接成最终的投影矩阵 \mathbf{V} .

2.3 哈希编码

得到最终的投影矩阵后,将原始样本 \mathbf{X} 投影到 \mathbf{V} 上,得到低维样本数据. 利用式(15)所示的哈希函数对低维样本进行 0 或 1 的二值编码,得到最终的二进制码.

$$\mathbf{H} = \operatorname{sign}(\mathbf{V}^T \mathbf{X}) \quad (15)$$

3 实验

为验证本文算法的性能,在 Win8 系统 Matlab R2016a 环境下,采用 3 个国际上通用的人脸图像数据库 AR、ORL 和 Yale 进行了几组实验.

3.1 r 参数选取实验

实验中,通过改变特征向量的数目 r 来测试它对算法模型的影响. 图 1 给出了 3 个数据集下 r 从 2 ~ 32 情况下 RPLPH 的平均检索精度 (MAP, mean average precision) 值. 可以看出,RPLPH 的检索性能因 r 的不同会发生改变. 当 r 的取值较大时,降维矩阵引入了更多的噪音,降低了性能. 当 r 的取值较小时,降维矩阵块中只提取了较少的特征向量,捕获 PCA 投影数据的结构信息远远不够,因而性能并不能改善. 实验结果显示,当特征向量个数取中间值时,RPLPH 方法性能较好.

3.2 与经典 LPP 哈希算法性能比较

为验证算法的性能,将原始样本与采用本文算

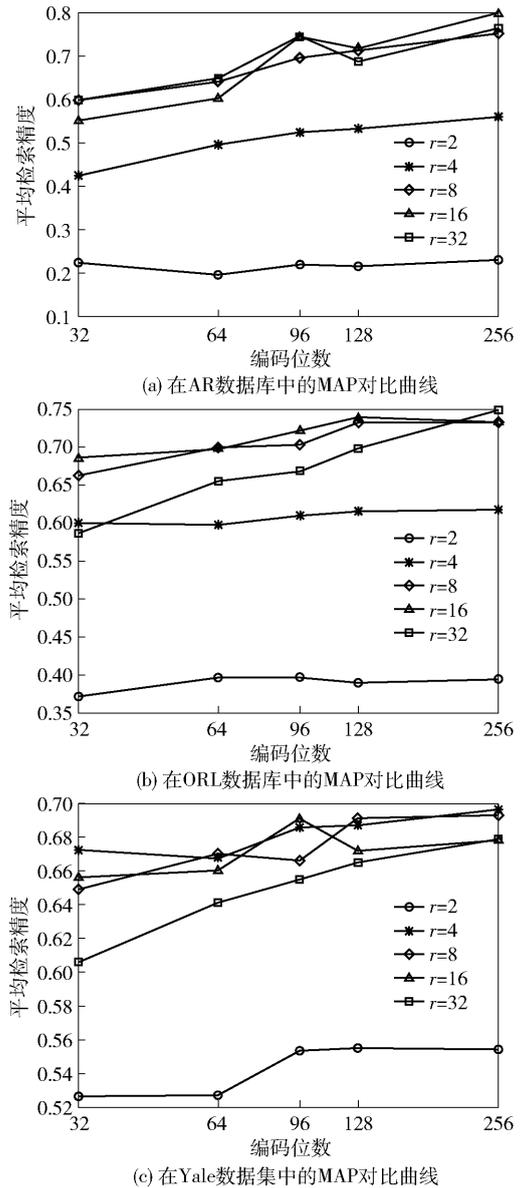


图 1 3 个数据集中文本算法的性能随 r 取值不同的 MAP 曲线

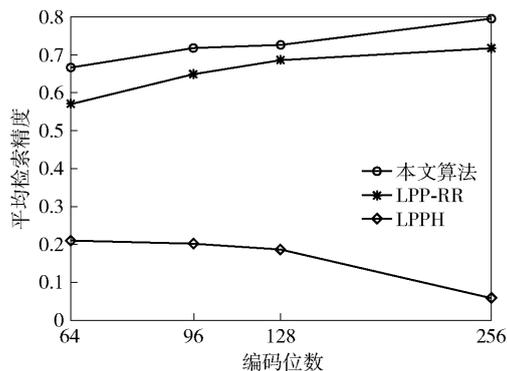
法,仅在 PCA 降维后的子空间中进行 LPP 映射算法 (LPPH, LPP Hashing) 以及仅基于 PCA 随机旋转 (LPP-RR, LPP random rotation) 3 种方法进行了对比实验,实验的 MAP 曲线如图 2 所示. 可以看出,由于本文算法在 PCA 降维时只选取少量的特征向量,减少了投影过程中不必要的噪音. 随机旋转减少了编码之间的量化误差,因此,本文算法比 LPP-RR 和 LPPH 要占优势. 同时,在 ORL 数据库中,将本文算法与 LPPH 和 LPP-RR 2 种算法的训练时间进行了对比实验. 结果如表 1 所示. 可以看出,本文算法的训练时间较少,尤其在 32 bit 和 64 bit 时效果更明显.

表 1 3 种算法下 ORL 的时间复杂度对比结果 s

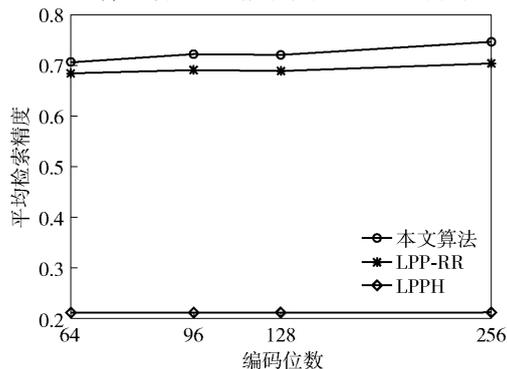
方法	32 bit	64 bit	96 bit	128 bit	256 bit
LPPH	0.454 0	0.444 3	0.477 5	0.396 6	0.457 9
LPP-RR	0.573 3	0.639 2	0.608 8	0.528 6	0.575 7
RPLPH	0.229 5	0.269 1	0.310 9	0.302 3	0.448 7

3.3 与其他哈希算法的对比实验

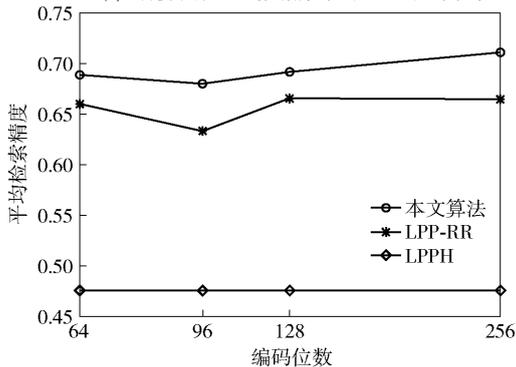
将本文算法与 7 种经典哈希方法 LSH^[1]、PCA 哈希 (PCAH^[2], PCA Hashing)、球哈希 (SpH^[3], spherical Hashing)、谱哈希 (SH^[4], spectral Hashing)、迭代量化 (ITQ^[5], iterative quantization)、随机旋转 PCA (PCA-RR^[6], PCA random rotation)、两层随机主映射哈希 (R²PCAH^[7], two-fold randomness principal projection Hashing) 进行了对比. 表 2 给出了 8 种方法在 3 个数据集下分别进行 96、128 和 256 位编码的 MAP 数值. 可以看出, 本文算法的 MAP 数值最高. 对 PCAH 而言, 由于在进行 PCA 降维时的噪音信息使得其随着编码位数的增加, 其 MAP 值逐渐减少, 而 RPLPH 在 PCA 投影时仅提取了少量有用的信息从而成功地避免了这个现象. PCA-RR 采用随机正交矩阵对 PCA 进行旋转, 使得编码间的量化误差最小, 其检索性能提升. 而本文算法同样引入随机正交矩阵对 PCA 顶部特征向量进行随机旋转, 可以获取多种投影矩阵, 减少了编码间的量化误差, 具有较高的检索性能. 图 3 给出了 8 种方法在 ORL 数据集上的 P-R 曲线. 可以看出, 本文算法同样具有较好的性能. PCAH 方法随着编码长度的增加曲线迅速地下降, 其重要的原因是在 PCA 进行特征值分解的时候引入了一些噪音, 编码位数越大, 其引入的噪音越多, 性能



(a) 3种算法在AR数据集中的MAP对比曲线



(b) 3种算法在ORL数据集中的MAP对比曲线



(c) 3种算法在Yale数据集中的MAP对比曲线

图 2 3 种算法在 3 个数据集上的 MAP 对比曲线

表 2 3 个数据集下 8 种方法不同编码长度的 MAP 对比值

方法	AR			ORL			Yale		
	96 bit	128 bit	256 bit	96 bit	128 bit	256 bit	96 bit	128 bit	256 bit
SpH	0.424 0	0.442 1	0.493 4	0.520 1	0.528 1	0.565 6	0.614 8	0.611 9	0.631 0
PCA-RR	0.651 8	0.693 7	0.676 9	0.671 0	0.703 6	0.708 8	0.675 2	0.674 1	0.698 3
SH	0.432 9	0.382 1	0.315 2	0.390 3	0.391 6	0.335 0	0.636 6	0.607 1	0.561 7
PCAH	0.194 7	0.177 1	0.140 3	0.278 0	0.274 6	0.192 9	0.466 3	0.443 6	0.425 2
LSH	0.580 8	0.613 0	0.637 4	0.576 7	0.630 8	0.647 1	0.627 5	0.658 9	0.692 4
ITQ	0.743 7	0.718 0	0.744 4	0.707 4	0.733 0	0.720 1	0.685 2	0.688 6	0.706 6
R ² PCAH	0.679 2	0.707 7	0.718 3	0.700 2	0.716 1	0.731 9	0.693 4	0.696 6	0.706 6
本文算法	0.752 1	0.766 2	0.789 4	0.739 3	0.751 7	0.757 3	0.712 5	0.715 6	0.735 6

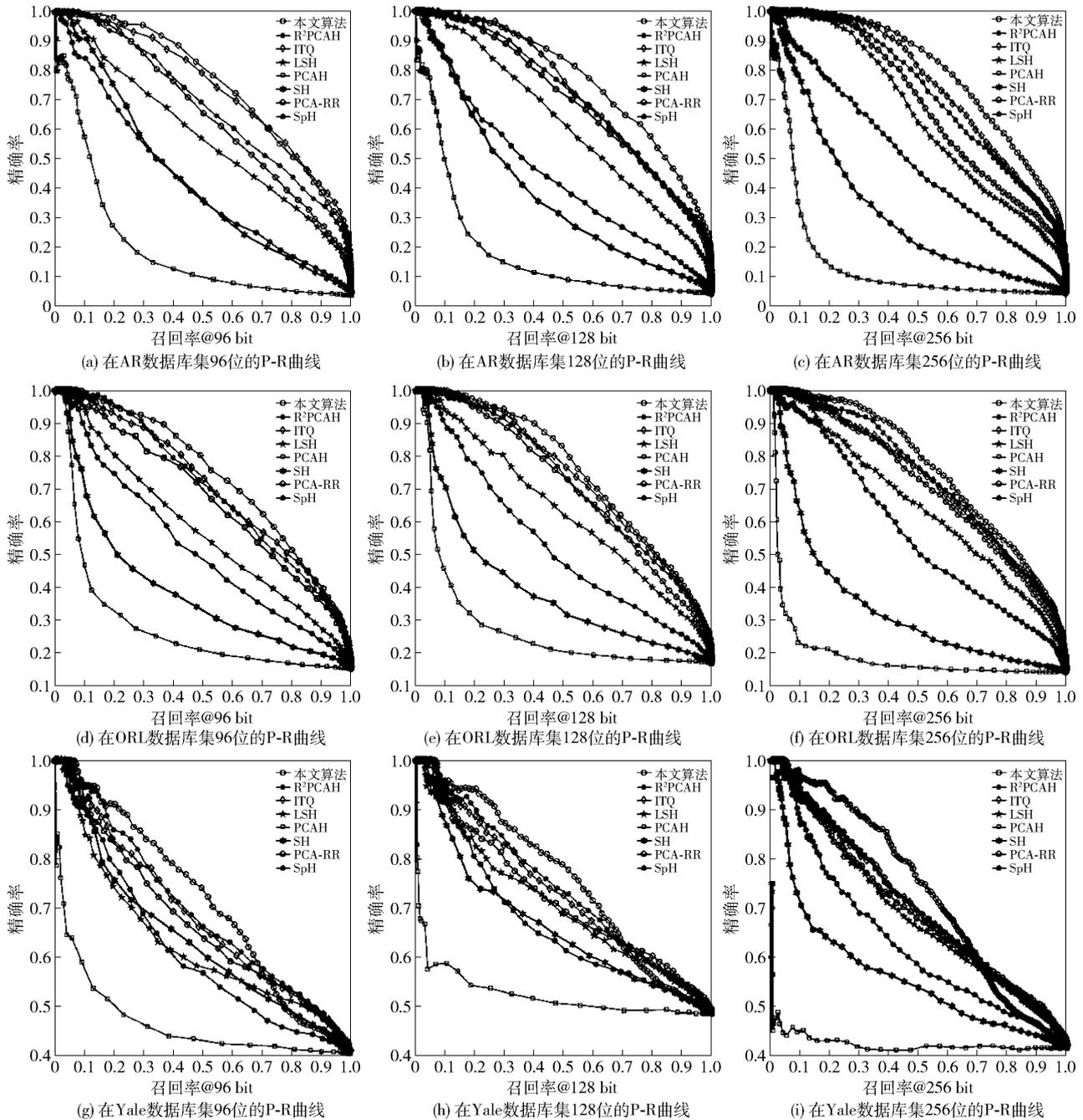


图3 3个数据集下8种方法不同编码长度时的对比曲线

越差. 本文算法充分地利用随机旋转而且只取 PCA 处理后的最顶部具有辨识度的特征向量, 有效地解决了传统 LPP 在基于哈希图像处理中出现的低效、耗时等缺点.

4 结束语

提出了一种图像哈希算法, 首先对样本进行 PCA 变换, 再对降维样本进行 LPP 变换, 并利用随

机正交矩阵对 PCA 变换后最高位的部分特征向量进行随机旋转, 以减少编码间的量化误差. 最后把得到的多个小投影矩阵拼接, 构造最终的投影矩阵, 训练样本经过该投影矩阵降维后利用哈希函数得到哈希编码. 该方法能够在投影过程中保持数据的全局结构和局部信息, 有效解决 LPP 直接用于哈希图像处理中存在的问题. 实验结果证明, 该算法取得了较好的效果.

参考文献:

- [1] Gionis A, Indyk P, Motwani R. Similarity search in high dimensions via Hashing [C] // VLDB'99. Edinburgh: IEEE, 1999: 518-529.
- [2] Wang J, Kumar S, Chang S. Semi-supervised Hashing for scalable image retrieval[C] // CVPR'10. San Francisco: IEEE, 2010: 3424-3431.
- [3] Heo J, Lee Y, He J, et al. Spherical Hashing [C] // CVPR'12. Rhode: IEEE, 2012: 2957-2964.
- [4] Weiss Y, Torralba A, Fergus R. Spectral Hashing [C] // NIPS'08. Vancouver: Elsevier, 2008: 1753-1760.
- [5] Gong Y, Lazebnik S. Iterative quantization: a procrustean approach to learning binary codes [C] // CVPR'11. Colorado Springs: IEEE, 2011: 817-824.
- [6] Jegou H, Douze M, Schmid C, et al. Aggregating local descriptors into a compact image representation [C] // CVPR'10. San Francisco: IEEE, 2010: 3304-3311.
- [7] Li P, Ren P. R²PCAH: Hashing with two-fold randomness on principal projections[J]. Neurocomputing, 2017 (235): 236-244.
- [8] 李金凤, 吴涛, 王宏霞. 基于 MFCC 相关系数的语音感知哈希认证算法 [J], 北京邮电大学学报, 2015, 38 (2): 89-93.
- Li Jinfeng, Wu Tao, Wang Hongxia. Perceptual Hashing based on correlation coefficient of MFCC for speech authentication [J]. Journal of Beijing University of Posts and Telecommunications, 2015, 38(2): 89-93.
- [9] 张秋余, 省鹏飞, 黄羿博, 等. 多格式音频感知哈希算法 [J]. 北京邮电大学学报, 2016, 39(4): 77-82.
- Zhang Qiuyu, Xing Pengfei, Huang Yibo, et al. Perceptual Hashing algorithm for multi-format audio [J]. Journal of Beijing University of Posts and Telecommunications, 2016, 39(4): 77-82.
- [10] Abdi H, Williams L J. Principal component analysis [J]. Wiley Interdisciplinary Reviews: Computational Statistics, 2010, 2(4): 433-459.
- [11] He X, Niyogi P. Locality preserving projections [C] // NIPS'04. Vancouver: ACM, 2003: 153-160.
- [12] Leng C, Cheng J, Yuan X, et al. Learning binary codes with bagging PCA [C] // ECML'14. Nacy: Springer, 2014: 177-192.