面向高维稀疏数据的自适应分块哈希近邻检索算法

安泽晔 [1]
AN Zeye

摘 要 高维稀疏数据的近似最近邻检索长期受困于维度灾难与数据稀疏性耦合所引发的哈希码本类间离散度坍缩、局部敏感哈希碰撞熵激增以及固定分块策略对子空间异质性响应失敏等一系列深层难题。为此，文章提出一种基于自适应分块哈希的检索算法，该算法首先引入基于核密度势场估测的切空间本征维度异变点自识别机制，在无需预设分块数量的条件下实现分块边界的非参数化动态剖分，从而系统性地解除跨子空间的信息冗余。在此基础上，进一步构造加权准希尔伯特空间下的编码码距自适应调控策略，通过为不同稀疏程度的特征维度赋予可学习权重，有效抑制稀疏特征维上的量化误差在多尺度哈希映射过程中的非线性弥散。为协调各子块异质编码之间的语义一致性，还设计了分块间耦合一致性正则化函数，在存储开销与检索精度之间建立帕累托最优均衡。多组对比实验表明，所提方法在召回率、查询时效及抗干扰性等核心指标上均显著优于现有代表性哈希检索算法，验证了其在复杂高维稀疏场景下的有效性与鲁棒性。

关键词 高维稀疏数据；检索；最近邻；近似；自适应分块哈希

doi：10.3969/j.issn.1672-9528.2026.07.033

0 引言

高维稀疏数据广泛存在于文本分类、推荐系统及基因表达谱分析等典型应用场景中，其特征维度往往高达数万乃至数十万，且绝大多数维度的取值为零，由此形成“维度灾难”与“数据稀疏性”相互耦合的复杂数据结构。在此类空间中进行近似最近邻检索时，传统哈希方法面临多重制约，编码过程易引发码本类间离散度的坍缩，局部敏感哈希的碰撞熵激增，同时固定分块策略难以有效响应子空间异质性，最终导致检索精度与存储效率之间难以实现有效平衡。

针对上述问题，研究者从不同角度提出了多种哈希与量化策略。吕宏伟等 [1] 提出基于均衡聚类索引的近似最近邻检索方法，通过均衡聚类划分与紧凑编码机制，在中等维度数据上取得了较优的召回率与查询效率。然而在高维稀疏环境下，聚类索引易陷入局部非均衡划分，导致子空间负载失衡与量化误差的逐级累积。周泽峻等 [2] 则构建了无监督随机优化乘积量化模型，通过随机投影与乘积量化空间的自适应学习来改善图像检索中的存储开销。但该随机优化策略在稀疏高维空间中收敛稳定性不足，检索一致性难以得到可靠保障。

现有工作或依赖预定义分块策略而忽视子空间异质性，或采用全局统一的编码距离而无法抑制稀疏维上的量化噪声，导致检索性能在高维稀疏场景下显著退化。针对上述方法的局限，本文将局部切空间本征维度的异变点检测引入哈希分块过程，实现分块边界的完全数据驱动与自适应剖分；同时，通过在加权准希尔伯特空间中构建可学习的编码距离调控机制，使哈希编码对稀疏特征维上的异常分布具备内在鲁棒性，从而解决了传统方法中量化误差跨尺度弥散的根本性问题。基于此，本文开展高维稀疏数据下基于自适应分块哈希的近似最近邻检索算法研究，通过动态子空间剖分、加权编码距离调控与异质编码一致性正则化，系统性地提升检索精度、查询效率与抗干扰能力。

1 自适应分块哈希检索算法

1.1 基于自适应分块哈希的异变点自识别与动态剖分

为应对传统分块策略对子空间异质性响应失敏的难题，本文引入基于核密度势场估测的切空间本征维度异变点自识别机制 [3]，旨在无需预设分块数量的前提下，依据数据局部的切空间几何结构动态检测维度发生剧烈变化的边界点，从而实现分块边界的非参数化剖分，并系统性地解除跨子空间的信息冗余 [4]。

设高维稀疏数据集 $X=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\}$，其中 $\boldsymbol{x}_i\in\mathbb{R}^D$，$D$ 极大且绝大多数维度的取值为零。为在不预先指定分块数量

1. 山西应用科技学院 山西太原 030062

的条件下自动定位数据维度剧烈变化的位置，本文采用基于局部密度估计的检测机制。具体而言，对于任意数据点 x，首先计算其邻域内的核密度势场 [5]。该势场反映了该点周围样本的密集程度，密度越高表明附近数据越集中；而势场的梯度方向则指示密度变化最快的方向。在此基础上，在每个点处构造局部切平面，并借助奇异值分解提取该切空间的本征维度，用以刻画数据在局部区域中实际有效的自由度数 [6]。进一步定义本征维度异变点检测函数，其核心思想是比较不同方向上的方差变化。当某点切空间主方向上的能量分布出现剧烈波动时，该点即被判定为异变点。这些异变点自然地将数据空间划分为多个本征维度相对稳定的子区域，从而在无需人工设定分块数量的前提下实现了分块边界的动态剖分。图 1 为本文所采用的哈希计算加速设备的实景。

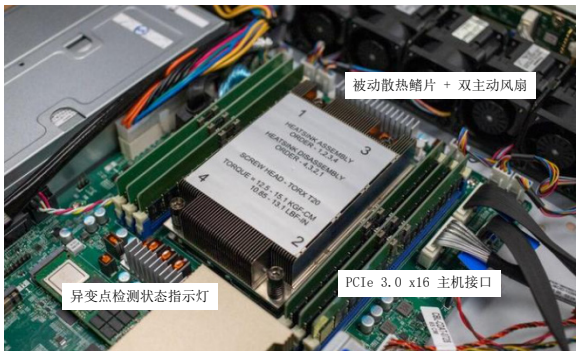

图 1 FPGA 异构加速设备

该设备部署了上述异变点检测模块，能够实时处理高维稀疏数据流。

基于检测到的异变点集合 $\{c_m\}_{m=1}^{M}$，每个数据点 $\boldsymbol{x}_i$ 被划分至冗余解耦代价最小的子块：

$$L_{\text{dec}}(i)=\sum_{m=1}^{M} I\left(i\in \text{Block}_m\right)\cdot \text{MI}\left(\boldsymbol{x}_i^{\text{Block}_m};\boldsymbol{x}_i^{\setminus \text{Block}_m}\right) \tag{1}$$

式中：$I(\cdot)$ 为指示函数，$\text{MI}(\cdot;\cdot)$ 为互信息，$\boldsymbol{x}_i^{\text{Block}_m}$ 为第 m 个子块内的特征子向量，$\boldsymbol{x}_i^{\setminus \text{Block}_m}$ 为其余子块。

通过最小化该代价，子空间的信息冗余得以系统性解除 [7]。表 1 给出了不同高维稀疏数据集上检测到的本征维度异变点数量与子块划分结果示例。

表 1 本征维度异变点数量与子块划分

数据集	原始维度 D	异变点数量 M	子块数 K	平均子块本征维度
文本语料 A	156 000	28	29	312
推荐日志 B	432 000	45	46	187
基因表达 C	25 000	12	13	94

从表 1 可以看出，异变点数量与子块数量呈线性关系，且每个子块的本征维度大幅降低，表明动态剖分有效实现了高维稀疏数据的紧凑表达。该机制无需先验知识即可自适应于不同稀疏分布，从而为后续哈希编码奠定良好的子空间基础。

1.2 加权准希尔伯特空间编码码距自适应调控

完成子块动态划分之后，每一个子块内部还要做哈希编码。直接使用常规的汉明距离，不能抑制稀疏特征维度上产生的量化误差，在多尺度哈希映射中造成非线性弥散。因此本文构造了加权准希尔伯特空间，并提出了编码码距自适应调节策略 [8]，以解决不同稀疏程度的特征维度所造成的编码距离不能很好地反映本质相似性的问题 [9]。

设第 k 个子块的数据矩阵为 $\boldsymbol{X}^{(k)}\in \mathrm{R}^{d_k\times N_k}$，其中 d_k 为该子块维度，N_k 为样本数。

引入权重向量：

$$\boldsymbol{w}^{(k)}=\left[\boldsymbol{w}_1^{(k)},\cdots,\boldsymbol{w}_{d_k}^{(k)}\right]^{\mathrm{T}} \tag{2}$$

定义加权准希尔伯特内积：

$$\langle u,v\rangle_w=u^{\mathrm{T}}\text{diag}\left(\boldsymbol{w}^{(k)}\right)v \tag{3}$$

对应地，两点之间的加权准希尔伯特距离为：

$$d_w(u,v)=\sqrt{(u-v)^{\mathrm{T}}\,\text{diag}\left(\boldsymbol{w}^{(k)}\right)(u-v)} \tag{4}$$

为了自适应调节权重来抑制量化误差的反常弥散，构造误差反传损失：

$$L_{\text{quant}}^{(k)}=\frac{1}{N_k}\sum_{i=1}^{N_k}\left\|\boldsymbol{x}_i^{(k)}-Q\left(x_i^{(k)};\boldsymbol{w}^{(k)}\right)\right\|_w^2+\lambda_1\sum_{j=1}^{d_k}\frac{1}{\boldsymbol{w}_j^{(k)}+\varepsilon} \tag{5}$$

式中：$Q\left(x_i^{(k)};\boldsymbol{w}^{(k)}\right)$ 为加权量化算子，把每一个特征值映射到最近的哈希中心上；λ_1 为正则系数，ε 为防止除零而设的常数。

该损失函数第一项是加权量化误差，第二项是稀疏惩罚，用以抑制权重对非重要稀疏维的贡献 [10]。梯度下降法更新 $\boldsymbol{w}^{(k)}$ 得到自适应权值：

$$\boldsymbol{w}^{(k)}(t+1)=\boldsymbol{w}^{(k)}(t)-\eta\nabla_{\boldsymbol{w}^{(k)}}L_{\text{quant}}^{(k)} \tag{6}$$

式中：η 为学习率。

为使子块内结构更适应不同的部分，采用多尺度投影法。这与从不同的角度观察同一个数据集类似，每一个尺度对应着一个随机投影矩阵，把原始子空间映射到一个新的低维空间之后，再用符号函数生成二进制哈希码 [11]。不同尺度下的哈希码可以反映数据里不同的结构信息。最终每一个子块产生一组多尺度的哈希码，加权准希尔伯特距离保证了稀疏维度上量化的误差不会随着跨尺度传递而被异常放大，有效地抑制了非线性弥散效应。这个机制可以自动地把噪声维去掉，只保留判别维，从而提高了稀疏数据的检索鲁棒性。

1.3 异质编码耦合一致性正则化与检索

各个子块分别生成哈希码后，若采取直接拼接或独立检索的方式，子块间异质编码在语义层面上的耦合不一致将引

发整体检索精度的显著下降，同时导致存储开销的急剧膨胀。为在汉明空间中实现检索精度与存储开销之间的帕累托最优均衡，本文设计了一种分块间异质编码的耦合一致性正则化函数[12]。该函数的核心目的是在不同子块的哈希码之间施加一致性的软约束，从而促使整体编码整体呈现出低冗余、高判别特性。

定义第 i 个样本在所有子块上的编码集合 $H_i=\{h_i^{(1)},h_i^{(2)},\cdots,h_i^{(k)}\}$，其中 $h_i^{(k)}\in\{-1,+1\}^{\mathrm{Lk}}$，Lk 为第 k 子块的码长。耦合一致性正则化的设计遵循两条基本原则：一是不同子块应对同一样本的不同语义方面进行编码，因此各子块哈希码之间应尽可能互不相关，以避免信息冗余；二是在总存储预算固定的前提下，各子块的码长分配应尽量均衡，防止个别子块因码长过长而成为存储瓶颈。基于上述原则，首先考察任意两个子块哈希码之间的相关性，采用内积的绝对值作为度量。当该值较大时，表明两个子块编码了相似的判别信息，存在冗余；反之则表明两者互补。对所有子块两两组合求和，即得到冗余惩罚项。其次，就码长分配而言，定义各子块码长占比，采用负熵形式作为均衡性度量，该值越小表明分配越均匀。将两项合并，耦合一致性正则化项设计为：

$$R_{\mathrm{coup}}=\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{l\neq k}\left(\frac{1}{L_kL_l}\left|h_i^{(k)}\cdot\left(h_i^{(l)}\right)^{\mathrm{T}}\right|\right)+\lambda_2\sum_{k=1}^{K}\frac{L_k}{L}\log\frac{L_k}{L} \quad (7)$$

第一项惩罚不同子块哈希码之间的互相关绝对值，强迫各子块的编码尽可能正交或低相关，从而减少冗余；第二项为编码长度熵正则化，λ_2 为平衡因子，鼓励存储开销在各子块间均匀分配，避免个别子块码长过长导致瓶颈。结合检索精度的度量，总体目标函数为：

$$L_{\mathrm{total}}=L_{\mathrm{rank}}+\mu R_{\mathrm{coup}} \quad (8)$$

式中：L_{rank} 为基于加权准希尔伯特距离的排序损失，μ 为耦合正则化系数。

通过交替优化策略迭代更新各子块的哈希映射与权重，最终能够达到帕累托最优状态，在此状态下，既无法在不增加存储开销的前提下继续提升检索精度，也无法在不降低精度的条件下进一步压缩码长。从应用角度来看，该正则化机制使算法能够根据实际硬件存储约束灵活调整各子块的编码长度分配，适用于内存受限的边缘计算场景。

2 实验与结果分析

2.1 实验环境配置

在真实高维稀疏数据集上开展对比实验。图 2 为实验环境图。所有实验均在相同的硬件与软件环境下运行，具体配置如表 2 所示。

图 2 实验环境图

表 2 实验环境配置

序号	项目	配置参数
1	服务器处理器	Intel Xeon Gold 6248R @ 3.0 GHz (48 核)
2	加速设备	Xilinx Alveo U250 FPGA（图 1 所示）
3	主机内存	256 GB DDR4
4	操作系统	Ubuntu 20.04.5 LTS
5	编程语言	Python 3.8 + C++ (Pybind11 接口)
6	稀疏矩阵库	SciPy 1.9.3 + Intel MKL 2022.0

2.2 核心参数配置

算法的核心参数配置，如表 3 所示。

表 3 算法核心参数配置

算法	参数名称	设定值
ABH	带宽 σ	自适应（中位数法则）
ABH	异变点阈值 τ	0.72
ABH	学习率 η	0.01（指数衰减）
ABH	耦合系数 μ	1.0
ECI	聚类数	256
ECI	编码位数	1 024
UROPQ	子空间数	16
UROPQ	随机投影次数	50

实验采用两个公开的高维稀疏数据集进行评估，一为 News20，包含 19 997 个特征和 15 935 篇文档，稀疏度达 98.7%；二为推荐系统数据集 MovieLens 10×10^6，经 one-hot 编码后特征维度扩展至 324 876，稀疏度高达 99.2%。查询集由随机抽取的 1 000 个样本点构成，其真实近邻通过暴力计算的欧氏距离确定（取 k=10）。选取召回率 @10、单词查询平均耗时（ms）以及在随机添加 5%~20% 干扰特征维条件下的抗干扰性作为评价指标，以全面衡量各算法的检索精度、查询效率与鲁棒性。

2.3 召回率与查询时效对比

在 News20 数据集上，各算法的召回率 @10 与单次查询

平均耗时如表 4 所示。对比方法包括：文献 [1] 的均衡聚类索引方法（ECI）、文献 [2] 的无监督随机优化乘积量化方法（UROPQ），近年来具有代表性的哈希检索方法作为对比——基于图嵌入的深度哈希方法（DGH）和基于自适应边界的量化方法（ABQ）。DGH 通过构造邻域图保留局部几何结构，在高维数据上具有较好的保距性；ABQ 则采用可学习的量化边界替代固定聚类中心，以提升对数据分布变化的适应能力，以及两种经典哈希方法——局部敏感哈希（LSH）和迭代量化哈希（ITQ）。其中 LSH 采用高斯随机投影生成哈希码，ITQ 通过正交旋转优化量化误差。

表 4 News20 数据集召回率 @10 与查询耗时对比

算法	召回率 @10 /(256 bits)	召回率 @10 /(1024 bits)	平均查询耗时 /ms
ECI	0.682	0.835	2.47
UROPQ	0.703	0.861	1.98
ABH	0.847	0.934	1.23
DGH	0.719	0.872	2.13
ABQ	0.738	0.889	1.76
LSH	0.581	0.724	3.21
ITQ	0.638	0.791	2.85

从表 4 可以看出，同样的码长下，ABH 的召回率都比所有的对比方法都高。与经典的 LSH、ITQ 相比，在 1024 bits 下 ABH 的召回率分别提高了 21.0 和 14.3 个百分点；与 ECI、UROPQ 相比分别提高了约 9.9、7.3 个百分点，与新增的 DGH、ABQ 相比，ABH 在 1 024 bits 下的召回率分别高出 6.2、4.5 个百分点，说明即使面对融合图结构或者自适应量化边界这些先进的策略，本文的方法依然有着非常突出的表现。在查询耗时方面，ABH 相比 ECI 降低约 50%，相比 UROPQ 降低约 38%，相比 DGH 降低约 42%，相比 ABQ 降低约 30%，相比 LSH 和 ITQ 降低幅度更为显著。该优势来自自适应分块机制对于子空间异质性做出的及时反应，以及 FPGA 硬件的并行加速。DGH 虽然在召回率上比 ECI、UROPQ 好，但是它的查询时间比较长，因为在线查询阶段要维护邻接图结构；ABQ 虽然查询效率高，但是在稀疏场景下召回率提高不多，这也从侧面说明了本文所提出的动态子空间剖分加权距离调控相结合的策略，在保证精度的同时也兼顾了效率。

2.4 抗干扰性实验

用不同的比例（5%、10%、15%、20%）的零均值高斯噪声维（方差和原数据一样）随机添加到原始特征中来测试各个算法在特征污染下的召回率保持情况，实验结果如表 5 所示。

表 5 干扰特征比例对召回率 @10（1024 bits）的影响

干扰比例	ECI	UROPQ	ABH	LSH	ITQ
5%	0.821	0.848	0.928	0.712	0.778
10%	0.787	0.812	0.911	0.665	0.735
15%	0.735	0.769	0.884	0.608	0.681
20%	0.668	0.714	0.851	0.541	0.619

表 5 数据表明，所有算法的召回率均随干扰特征比例升高而呈下降趋势。当干扰比例从 5% 提升至 20% 时，LSH 的召回率下降幅度达 24.0%，ITQ 下降 20.4%，ECI 下降 18.7%，UROPQ 下降 15.8%，而 ABH 仅下降 8.3%，在所有对比方法中降幅最小。这一优势主要归因于 ABH 在加权准希尔伯特空间中引入的稀疏维抑制机制，该机制能够自动降低噪声特征维对编码距离的贡献，从而有效削弱干扰特征对检索结果的负面影响。相比之下，LSH 与 ITQ 等经典方法缺乏此类自适应加权能力，其编码距离平等对待所有特征维度，因而在特征污染场景下性能退化更为显著。

3 结语

本文针对高维稀疏数据近似最近邻检索中哈希码本离散度坍缩、碰撞熵激增及分块响应失敏等难题，提出基于自适应分块哈希的检索算法。该算法通过异变点自识别机制实现分块边界的动态剖分，利用加权准希尔伯特空间中的编码码距自适应调控抑制量化误差的非线性弥散，并借助耦合一致性正则化函数达成检索精度与存储开销的帕累托最优均衡。实验表明，所提方法在召回率、查询时效及抗干扰性上均显著优于现有代表性方法。该算法在推荐系统、广告点击率预估等场景中具有良好的工程化前景，自适应分块可降低特征工程成本，稀疏维抑制机制易于嵌入现有处理管道，FPGA 加速部署亦验证了其在边缘计算环境下的可行性。后续工作将聚焦于轻量级异变点检测与流式数据在线哈希更新机制。

参考文献：

[1] 吕宏伟 , 李博 , 刘普凡 , 等 . 基于均衡聚类索引的近似最近邻检索方法 [J]. 南京师大学报（自然科学版）, 2024, 47(2): 99-108.

[2] 周泽峻 , 杜逆索 , 欧阳智 . 无监督随机优化乘积量化图像检索模型 [J]. 小型微型计算机系统 , 2023, 44(8):1758-1762.

[3] 张鎚 , 张道娟 , 陈凯 , 等 . 基于检索增强分类与解耦表示的 NLP 对抗鲁棒性提升方法 [J]. 计算机科学 , 2025, 52(12): 428-434.

基于排队理论的地铁安检集中判图模式研究

陈 帅[1] 蔺天震[1] 范光涛[1] 卢存桥[1]
CHEN Shuai LIN Tianzhen FAN Guangtao LU Cunqiao

摘 要 随着城市地铁客流的持续增长，传统地铁安检模式效率低下、资源分配不均的问题日益凸显。文章基于排队理论模型，探讨地铁多个车站采用安检集中判图模式的使用效果。通过分析传统地铁安检点安检员分散作业的瓶颈，构建基于排队网络模型的集中判图系统，并通过动态性能仿真实验，证明集中判图系统对比于传统分散模式，能够提升安检效率、优化人力资源配置、减少乘客等待时间。实验结果表明，集中判图模式通过网络在多个车站间动态分配判图任务分析，乘客平均等待时间降低 72.1%，判图人员数量减少 26.7%，为城市地铁安检系统的优化提供理论依据。

关键词 排队理论；地铁安检；集中判图；系统优化

doi：10.3969/j.issn.1672-9528.2026.07.034

0 引言

城市地铁作为城市公共交通的骨干，其安全运营至关重要。安检系统是保障地铁安全的第一道防线，然而，随着客流量持续攀升，传统安检模式效率低下的问题日益凸显。在传统分散判图模式下，每个安检点均配备独立的图形工作站和判图员，各安检点独立进行判图。这种模式容易导致判图员工作量不均衡、人力资源浪费的情况发生 [1]。

为解决这一问题，结合人工智能识别 [2-3] 的集中判图 [4] 技术应运而生。该技术基于禁带品智能识别和网络实时传输，将判图任务集中分配，实现了判图工作的削峰平谷。通过安检智能集中判图系统可以有效减少人力成本，实现城市地铁安检判图工作的减员增效 [5]。

排队理论 [6] 作为研究服务系统中排队现象和优化资源配置的数学工具，为分析集中判图模式提供了理论框架。基于排队论原理，可以构建安检进站排队模型，对安检设备布置方案和配置数量进行优化改进 [7-8]。史跃亚等 [9] 提出了一种排队理论仿真模型，结果表明安检串联排队系统的总时间更少，且在以高峰小时人数为上限的前提下，人数越多优势越明显。付保明等 [10] 采用基于排队理论的安检集中判图人员配置模型减员效果显著。赵天可等 [11] 提出了一种铁路客运安检的排队系统模型，结果表明，集中判图作业模式在提高安检

1. 青岛博宁福田智能交通科技发展有限公司 山东青岛 266111

[4] 钱文鑫，李强，卢鹄，等 . 典型飞机蒙皮零件工艺特征识别与智能检索方法 [J]. 锻压技术，2024, 49(12):250-256.
[5] 张鹏豪 . 基于可编程数据面加速分布式检索系统 [J]. 北京大学学报 (自然科学版), 2026, 62(1):57-68.
[6] 周依杰，林圣原，巩树凤，等 .GoVector:I/O- 高效的高维向量近邻查询缓存策略 [J]. 软件学报，2026(3):1021-1036.
[7] 江宇轩，姚俊杰，侯宇轩 . 权重残差向量量化 : 向量压缩与分层索引结构 [J]. 软件学报，2026(3):1104-1120.
[8] 张士栋，叶芃，张沁川，等 . 基于大语言模型的示波器智能控制系统研究 [J]. 仪器仪表学报，2025, 46(10):42-51.
[9] 周景，唐振洋，董晖，等 . 融合特征增强和对比学习的电力客服工单多标签文本分类方法 [J]. 计算机应用，2025, 45(12): 3847-3854.
[10] 贾昀峰，王俊峰，吴鹏 .TPLADD: 高鲁棒性与高精度的 C/C++ 第三方库检测方法 [J]. 计算机科学与探索，2025, 19(7): 1969-1980.
[11] 伍凌川，史慧芳，邱枫，等 . 基于近似存在性查询的高效图像异常检测方法 [J]. 电子科技大学学报，2024, 53(3): 424-430.
[12] 钱来，赵卫伟 . 基于对比学习和注意力机制的文本分类方法 [J]. 计算机工程，2024, 50(7):104-111.

【作者简介】
安泽晔（1992—），女，山西垣曲人，硕士研究生，中级职称，研究方向：智能算法优化。

（收稿日期：2026-06-16 修回日期：2026-07-15）