# Learned Index for Spatial Queries(ZM index)

随着基于位置的服务（LBS）的普及，空间数据处理在数据库系统管理的研究中受到了相当大的关注。在各种空间查询技术中，索引结构在数据访问和查询处理中发挥着关键作用。然而，现有的空间索引结构（例如R树）主要关注对数据空间或数据对象的划分。在本文中，我们探讨了通过学习数据分布来构建空间索引结构的潜力。我们设计了一种新的数据驱动空间索引结构，即学习Z-order模型（ZM）索引，它结合了Z-order空间填充曲线和分阶段学习模型。在真实和合成数据集上的实验结果表明，我们的学习索引在大多数情况下显著减少了内存成本，并且比R树更高效。

**介绍**

基于位置的服务无处不在。 智能手机和可穿戴设备生成大量地理空间数据，因此需要有效访问和处理此类数据[23]。 在文献中，已经设计了各种索引结构并应用于不同的应用[12]。 例如，Rtree是B+树的扩展，用于索引地理信息、多媒体对象等多维数据； R+-tree允许一个对象在多个子树中建立索引，并避免内部节点的重叠； kd树在k维空间中递归地划分数据对象，对于涉及多维搜索键的信息搜索是高效的。

然而，这些传统的空间索引结构没有利用现实世界数据中存在的常见模式。例如，R-tree中的搜索总是需要遍历树结构来找到想要的数据，即使是均匀分布的数据集。另一方面，如果索引能够捕捉到确切的数据分布，搜索时间和索引需要的内存大小都会很明显地减少。为了利用数据模式和分布，Kraska 等人提出了学习型索引的概念，其利用机器学习的方法来学习一个能够反映数据的模式和相关性的模型，因此实现了简洁结构的高效搜索

但是，对于学习型索引的研究仍然局限于一维数据。在这篇论文中，我们关注多维空间数据，并调查了如何学习空间数据的高效索引。不同于一维数据可以在属性值上排序，多维数据不能简单的排序。为了解决这个挑战，我们提出了一个新的叫做学习型ZM索引索引结构，来探索空间数据模式和如何高效的处理空间查询。学习型ZM-index利用了Z-order曲线来将多维数据映射到一维数据空间，然后构建一个多维索引模型来学习数据分布并预测数据项的位置。此外，我们开发了一种基于预测数据位置处理空间范围查询的算法。我们使用真实数据集和合成数据集进行的实验表明，学习到的 ZM 索引显着降低了内存成本，并且在大多数情况下比 R 树执行得更快。

论文的剩余部分组织如下。在第二节中，我们调查了现有的空间索引结构、范围查询和人工神经网络的研究。在第三节，我们提出了学习型ZM索引以及相应的查询算法。所提出的索引和算法在第四节进行了实验评估。最后，我们在第五节中总结了本文的未来工作的可能方向。

**相关工作**

1. 空间索引和范围查询

空间索引是为了高效的数据访问而设计的数据索引。大多数空间索引要么是空间驱动，或是数据驱动的结构[18].空间驱动的结构(例如，固定网格索引[15])将空间分解为单元，并基于几何标准映射数据对象。相反，数据驱动结构（例如，R 树 [4]）将数据对象划分为簇，并通过其最小边界矩形（MBR）分割空间。另外，UB tree，Z顺序曲线[17]和B+树的组合，是一种按Z顺序存储数据对象的平衡树。

范围查询是使用空间索引的最通用的查询。范围查询可以通过访问索引的根节点和递归检索与查询区域相交的孩子节点来处理。Markl [13] 通过检索 UB 树中与查询区域正确相交的所有 Z 区域来处理范围查询。文献[16]提出了一种递归分解算法，它形成了Z区域的最小集合。在文献[20]中，Skopal等人为UB tree提出了一个由下至上的算法基于两种类型的叶节点优化。在文献[3]中,Ahmed等人建议结合R树和B树来提高空间查询的性能。然而，这些方法没有利用数据对象的分布模式来获得更密集的索引表示。

1. 神经网络

人工神经网络(ANNs)是一种由人脑激发的编程范式，使计算机能够通过观察学习数据找出解决方案。它们已经在许多困难的任务中显示出了极大的成功例如自然语言处理，图像识别和语音识别。人脑可以被看做一个非线性和并行信息处理的系统。在这个系统中，许多神经元通过突触连接，形成神经网络。神经网络中的神经元在整个网络中同时存储和处理信息。受到生物神经网络工作机制的启发，人工神经网络包含大量基本的处理元叫做人工神经元，或简称为神经元或结点。这些结点彼此高度互连，类似于人脑中的生物神经元。与神经网络类似，人工神经元通过与数值权重相关的链接进行连接。在一个ANN中，人工神经元从链接获取输入，对其进行计算，然后将输出传输到其他神经元。通过重复调整这些权重，ANN“学会”了解决问题。

随着可用训练数据的增加，包含多层的深度神经网络（DNN）在解决复杂任务方面表现出显着的改进[6]。在DNN中，神经元被放置在不同的层中，执行不同的信息传输功能

在本节中，我们提出了一个新的叫做学习型ZM index的索引结构，并且开发了基于learned ZM index的空间查询技术。首先，我们使用Z-order曲线并给每个点一个z-address.然后，我们构建将 Z 地址与多阶段模型索引相结合的学习 ZM 索引。第三，我们提出了一种算法来处理学习到的 ZM 索引上的空间范围查询。

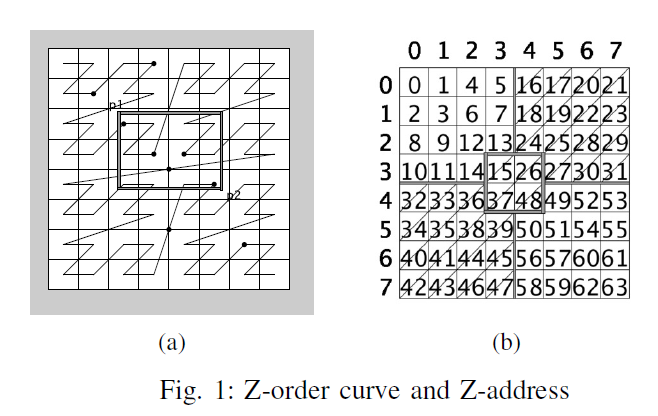
A.Z-address计算

将学习模型应用到空间索引的一个挑战就是在多维数据中没有序列化的顺序。在一维数据空间，所有的数据点都天然有一个线性的顺序。因此，如果数据点排序过，通过检索上下界之间的数据来处理范围查询是很容易的。例如，为了找到键值在α和β之间的数据项，我们只需要找到两个边界键的位置，然后返回它们之间的所有项。然而，对于多维空间的范围查询并没有这样的顺序。每个多维向量都可以被转化为一个唯一的z-address.图1a说明了Z-order曲线的一个例子。我们利用z-order曲线作为多维空间的映射的原因是它提供了一个重要的称为单调排序的几何属性[9]。在展示单调性之前，我们首先介绍dominance概念。

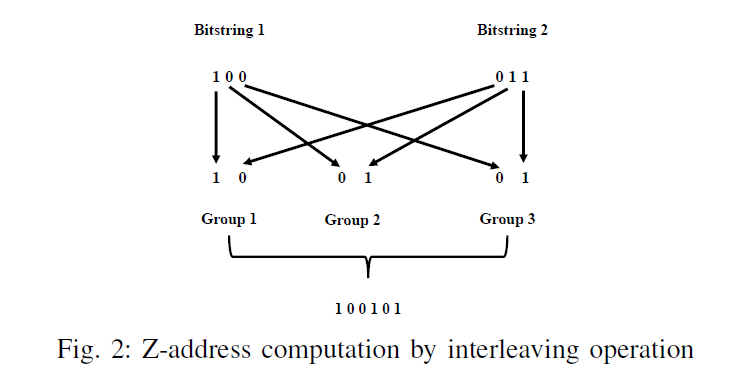
定义1(Dominance):给定两个点p和p’,如果p在任何维度都不差于p’,那么我们称p优于p’

性质1（单调排序）：按非递减的Z地址排序的数据点在某种程度上是单调的，即支配点位于其被支配点之前。

根据单调排序性质，如图1所示，对于点q，被q支配的任何点的Z地址都大于q，而支配q的任何点的Z地址都小于q。因此，对于给定的范围查询（用左上角为p1、右下角为p2的矩形指定），检索所有在p1和p2之间的点将确保检索到查询范围内的所有点。



Z 地址可以通过交织向量的每个标量值的位来计算。特别地，在n维空间，假设所有的数据点的值范围在[0,2m-1];那么每个标量值都可以用一个m位的bit string来表示。对于一个数据点，它的z-address包括m组，每组有n个bits;总共有m.n个bits。Z 地址中的第 i 个 (i ∈ [0,m]) 组由每个标量值的第 i 位贡献。图1b显示了图1a中Z顺序曲线对应的Z地址。 例如，p(4, 3) 的 Z 地址，其二进制形式为 (100, 011)，由三组 10-01-01（十进制为 37）组成。 图2显示了交织操作的过程。



B.多阶段索引模型的建立

学习型ZM索引的关键思想是构建一个能够学习查找键数据分布并有效预测记录位置或存在的模型。在所有现有索引结构都可以被视为某种模型的前提下，机器学习提供了一种用较低工程成本逼近数据分布的机会。例如，典型的B树查找时间和索引内存大小的成本为O(logn)和O(n)，而经过高度调优的学得模型有潜力达到O(1)的时间复杂度和O(1)的索引内存大小。

正如在第III-A节中介绍的，数据对象根据它们的Z地址进行排序，以便可以有效地检索查询范围内的数据。此外，对于已排序的数据集，能够有效预测关键位置的模型对累积分布函数（CDF）有很好的近似。给定一个查找键，可以如下预测相应对象的位置：

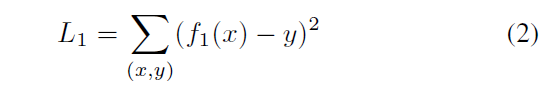
p = F(key) × N (1)

其中F(key)是学习的CDF用来评估一个键不大于查询键地可能性，N代表了数据项的总数。

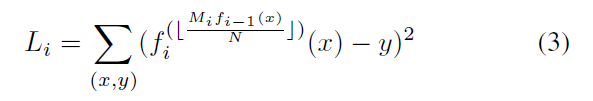
由于大多数机器学习模型在拟合数据方面不如 B 树那么好，因此很难学习精确的 CDF。幸运的是，最近的研究表明，ANN 模型可以非常有效地近似 CDF 的整体形状 [11]。

基于此，我们应用了一个多阶段模型索引(MMI),它能递归地将数据空间划分为子区域，从而减少估计误差[7].

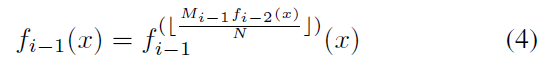
正如图3所示，我们构建了一个分阶段的模型，由多个阶段组成，每个阶段包含一个或多个较小的模型。每个较小的模型可以是一个人工神经网络（ANN）或其他学习模型（例如，线性模型）。在每个阶段，模型根据输入的查询键选择下一个阶段的另一个模型，直到在最后一个阶段估计出位置。具体而言，假设x是一个查询键，y是x的相应位置（y ∈ [0, N)）。我们将第1阶段的模型称为f1，其损失函数L1如下:



假设在阶段i的模型数量是Mi，阶段i的第j个模型表示为fi(j),模型i的损失函数是:



可以通过递归应用以下方程来构建完整的模型:



阶段模型中的每个模型以一定的误差估计搜索关键字的位置。 该估计用于在下一阶段选择模型，该模型负责整个数据集的特定子集，以便可以做出误差较低的预测。

请注意，由最终阶段估计的位置可能不是查询键的确切位置。然而，它很可能非常接近数据对象的实际位置。通过对每个键执行模型并记录最小误差（最差过度预测）和最大误差（最差低估），我们可以提供保证：如果存在某个给定的查找键，则该键必须在范围 [位置 - 最小误差，位置 + 最大误差] 内。然后，可以应用二分搜索或其变体来找到查询键的真实位置。

C ZM index的查询过程

*构建学得的ZM索引：*在学得的ZM索引中，每个点的Z地址是通过交织二进制位来计算的。所有的Z地址都被一个MMI索引。在这里，MMI索引有两个阶段，因为在我们的实验证明中，这样的设置能够达到最佳性能。所使用的模型是一个具有一个全连接隐藏层的前馈神经网络。我们选择了修正线性单元（ReLU）激活函数，因为它能够学习各种复杂的数据分布并具有较快的收敛速度。

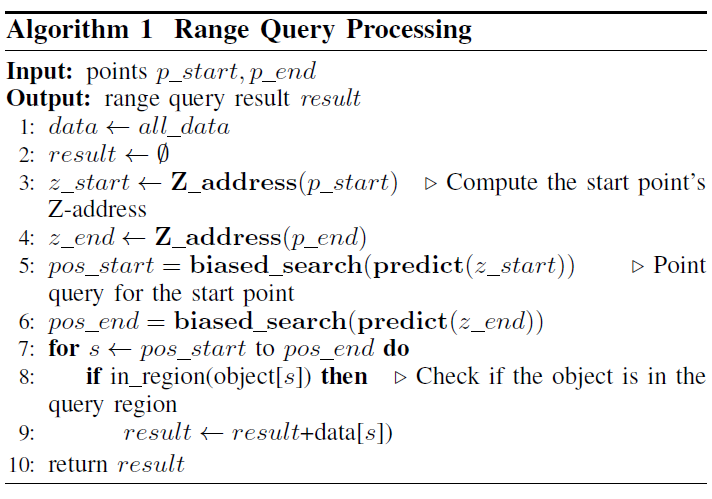
*范围查询处理：*如前所述，MMI索引根据查询键估计位置。估计的位置并不总是键的确切位置。我们现在提出了一种搜索策略来找到确切的位置。与通常出于效率原因返回页面指针的B树不同[7]，学得的模型提供了对象位置的预测。由于预测的位置预计接近确切位置，因此利用估计来找到数据对象预计会比传统的二分搜索更快。在本文中，我们应用了模型偏置搜索（MBS）[7]来在MMI估计位置后找到数据对象的精确位置。MBS是二分搜索的变体，其中第一个中间元素是估计的位置。

利用学得的ZM索引，我们可以设计有效的算法来处理空间查询。在本文中，我们以范围查询为例，因为范围查询是最流行和实用的空间查询之一。值得注意的是，学得的ZM索引还可以支持其他空间查询，如点查询和kNN查询。

给定一个查询区域（由两个空间点p\_start和p\_end表示），范围查询旨在检索驻留在该区域内的所有数据对象。范围查询的形式定义如下：

定义2（范围查询）：对于一个n维空间，给定由p\_start = (a0, a1, , an−1)和p\_end = (b0, b1, , bn−1)定义的查询区域，其中对于0 ≤ i ≤ n−1，都有ai ≤ bi，范围查询返回所有的空间对象q = (q0, q1, , qn−1)，使得对于0 ≤ i ≤ n − 1，都有ai ≤ qi ≤ bi。

我们的模型根据单调排序性质（性质1）处理范围查询。如图1a所示，给定由左上角p1和右下角p2定义的查询区域，区域内的任何对象q都满足条件Z地址(p1) < Z地址(q) < Z地址(p2)。如果我们检索所有Z地址在范围[Z地址(p1), Z地址(p2)]内的对象，就可以保证找到区域内的所有对象。算法1详细描述了处理范围查询的过程。该算法将表示查询范围的点p\_start和p\_end作为输入。首先计算p\_start和p\_end的Z地址（第3-4行）。函数predict返回点的估计位置。然后，算法使用MBS搜索（第5-6行）找到两个点的确切位置。算法继续检查pos\_start和pos\_end之间的每个对象，并将在查询范围内的点添加到结果集（第7-9行）。请注意，这里扫描并不总是高效的。例如，在图1b中，即使查询区域非常小，我们仍然需要为范围查询检索阴影部分中的所有对象。然而，通过将查询范围优化为多个连续段，我们需要检索的数据对象数量可以显著减少。



3) 理论分析： 学得的ZM索引的优势在于为所有空间对象构建线性顺序并学习Z地址的分布模式。与传统的R树相比，R树在非叶子节点中需要搜索以找到适当的子节点，而学得的ZM索引在各阶段之间没有搜索过程。当前阶段的结果用于选择下一个阶段的模型。因此，搜索时间复杂度降低。此外，搜索时间复杂度和索引内存大小与数据集的大小无关。考虑一个包含N个对象的数据集。R树需要O(logMN)的搜索时间和O(N)的索引内存大小，其中M是页面中的数据对象的最大数量。对于学得的ZM索引，N不会影响搜索时间和索引内存大小。假设学得的ZM索引是通过一个阶段进行训练的，这个阶段的模型是一个包含h和w个神经元的前馈神经网络。该模型的规模为O(hw)的乘法和加法以及O(hw)的索引内存大小。具有更多阶段和层的模型将导致更高的工程成本，但我们可以手动管理阶段和神经元的数量，而不受N的影响。换句话说，如果我们的模型能够学习一个非常简单但精确的Z地址分布，我们可以实现常数搜索时间和索引内存大小。例如，如果Z地址的分布是线性的，我们的模型将变成一个简单的线性模型，其搜索时间和内存大小是常数。