Research Proposal

Group 4

Zhu Yuzhang 20881929

Shu Yi 20928806

Ma Xiaolei 20887791

DENG Nan 20881163

Zhang Hexiao 20932780

Introduction

Learn-to-index (LTI) is an approach to build data indexing structures with machine learning algorithms to learn efficient mappings between the search keys and data locations. The index structure is learned from data and can adapt to specific data distribution which is different from traditional indexing structures such as B-trees or hash tables.

The basic idea of LTI is training a machine learning model to predict the location of data item in the dataset given a seach key. Then the trained model can be used to build an indexing structure for key-value mapping and optimizing the index for specific dataset. The resulting data index structure is faster and more efficient than traditional indexing structures with less space requirement.

However, when dealing with multi-dimensional data, indexing becomes much more difficult especially in traditional indexing structures such as B-trees and R-trees resulting in a phenomenon called the "curse of dimensionality", where the number of points required to cover a high-dimensional space grows exponentially, making indexing impractical. In high-dimensional spaces, traditional indexing structures are limited by poor data sparsity, low query selectivity, poor query performance and fragmentation.

LTI algorithms are capable of handling high-dimensional data and overcoming the curse of dimensionality by leveraging machine learning techniques to learn the best mapping method that fits the distribution of data well. It can automatically adjust the index as the data changes over time and handle sparse data as well as missing values, which are common in high-dimensional datasets.

The earliest LTI algorithms were primarily based on neural networks. Since then, different LTI algorithms has been proposed based on different machine learning techniques including decision tree and probabilistic graphical models. The efficiency of indexing and querying high-dimensional data is the primary target for LTI, and it als focus on the efficiency, saclability, robustness and flexibiliy of indexing.

In our project, we aim to study the LTI to efficiently index muti-dimensional data while minimizing memory usage and querying response time with Learn-to-index algorithms. And we have chosen three related papers in recent years to study.

Chosen Paper and Explanations

1. Effectively learning spatial indices. Qi, et al. PVLDB (2020)

The paper proposes a learned spatial index method based on ordering the data points by a rank space-based transformation. This index, called the Recursive Spatial Model Index (RSMI), focuses on point data and is designed to handle large datasets using a multi-dimensional data partitioning technique.

The RSMI overcomes challenges faced by traditional Z-order approaches, which struggle with grid resolution selection. Specifically speaking, traditional Z-order approaches face difficulties with large cells, which result in more false positives due to many points per cell, and with small cells, which make it hard to learn due to uneven gaps in the Cumulative Distribution Function (CDF).

The RSMI utilizes a Multi-Layer Perceptron (MLP) to fit the CDF with the transformed z-value as input and a recursive partitioning strategy based on data distribution. It can efficiently manage point, window, and k-nearest neighbor (KNN) searches. It also supports dynamic updates without compromising accuracy. Tests reveal that RSMI outperforms traditional spatial indices and other learned index techniques, improving query performance for a dataset of over 100 million points.

The RSMI approach focuses on point data and supports various kinds of queries and updates. The method is versatile and very related to the Z-order and R-tree we learned in class. That's why we choose this article.

2. LISA: Learned Index Structure for Spatial Data. Li, et al. SIGMOD(2020)

This paper introduces a novel learned index structure for spatial data called LISA (Learned Index Structure for Spatial Data). LISA first redefine a presentation of the gird cell to provide a more reasonable partition of the data distribution. Then, LISA uses mapping function to map spatial keys to 1-dimensional values. A learned shard prediction function is try to directly allocate the data layout in the disk pages by getting a predicted shard id for the mapped value and partitions the mapped space into shards. And the local models assign pages for all shards and perform intra-shard operations. This is the overall structure of LISA.

Meanwhile, LISA supports range query using lattice regression model based on KNN query. Specifically, if the query results is less than the setting number(K nearest neighbors), the query would expanded with another query to satisfy the setting number. Based on LISA structure, clients could also operate update operations on the data efficiently such as insertion and deletion. Finally, the paper summarizes the performance of LISA including storage consumption and I/O cost for range and KNN queries and so on, which meets the request of our topic.

Although learned index used in the model could provide improved query performance, we would have different methods considering different data target. The recursive model index (RMI) is not an efficient indexing in the context of spatial data. Even if considering multi-dimensional CDFs, RMI will have to access many pages, incurring considerable IO cost. In this case, multi-dimensional index is important. LISA is to Build a disk-based learned multi-dimensional index for spatial queries. It uses the learned model to generate the data layout, which is the shards in the paper. However, LISA did actually provide a novel indexing for us, how to use it wider or more efficiently

is worth considering. In our topic, to query a large amount of spatial data, the different resources consumption of different query method and query type should be taken into consideration such as spatial joins, closet pairs and so on.

3. Learning Multi-dimensional Indexes. Nathan, et al. SIGMOD(2020)

This paper proposes Flood, the first learned multi-dimensional in-memory index, which is read-optimized and can adapt itself to a particular dataset and workload by jointly optimizing the index structure and data storage layout using machine learning techniques.

The paper first explains how the multi-dimensional clustered index works in Flood. Flood is a variant of a basic grid index that divides d-dimensional data space into a d-dimensional grid of contiguous cells, so that data in each cell is stored together. For an index with d dimensions, Flood first chooses one dimension as the sort dimension for sorting points in the same cell, then chooses a ranking of the remaining d-1 dimensions to overlay a (d-1)-dimensional grid on the data. To answer a query, Flood first performs projection on the (d-1)-dimensional grid to obtain a range of cells that may satisfy the query, then by utilizing the sort dimension refines the range to a smaller size, finally scans the records in range and process the result.

According to the time cost of the query process, the paper proposes a query time equation as the cost model to optimize. Considering the varying properties of different datasets, the paper proposes three learnable weights in the equation and uses a random forest regression model to learn them based on sample data and queries. To optimize the data layout, Flood trains RMIs on each dimension to estimate the Cumulative Distribution Function (CDF) so that it can create cells in the grid containing similar numbers of points when possible while choosing an appropriate sort dimension. As for the sort dimension, Flood tries to accelerate the refinement process by using a piecewise linear model (PLM) on each cell to create slices.

The effectiveness of the Flood model was evaluated on a variety of datasets and query workloads. The authors stated that its performance was faster than or on par with every other index on the tested workloads. Also, there was another paper (The Case for Learned Spatial Indexes [AIDB@VLDB' 20]) which implemented the Flood model and performed extensive evaluation specifically on spatial indexes which showed significant impact on the performance of low selectivity queries while being less effective under higher selectivities.

Purpose and Outcome

Purpose:

1. Enhance our understanding of the importance of index structures for efficient query processing in spatial databases.
2. Explore how to apply machine learning techniques to learn index structures for spatial databases and how the techniques improve the efficiency of spatial query processing.
3. Learn how to assess the performance of proposed structures with different evaluation methods.
4. Identify potential research directions for further practice.

Expected outcome:

1. A comprehensive introduction to the multi-dimensional data indexing problems in large-scale

spatial databases, including the key challenges and motivations for finding improved index structures
2. Summarize proposed index structures, key concepts and techniques involved and evaluation methods from each selected papers, and analyze respective application scenarios and limitations.
3. Achieve some insights into the Learn-to-index algorithms based on the materials we have referenced.
4. Analyze the strengths and weaknesses of existing index structures and provide recommendations for future work to efficiently index muti-dimensional data .

Initial Allocation of Work

Name	Duty
Deng Nan	Read all the papers, Collect related papers , Summerize them into catogories and background information, Gather the related new solutions, Analyse the potential issues
Ma Xiaolei	Read all the papers, Collect related papers, Summerize them into catogories and background information, Gather the related new solutions, Analyse the potential issues
Shu Yi	Read all the papers, Gather the related new solutions, Find methods to evaluating the new solutions and experimental results, Analyse the potential issues
Zhang Hexiao	Read all the papers, Analyse the relationship of the new solutions, Find methods to evaluating the new solutions and experimental results
Zhu Yuzhang	Read all the papers, Analyse the relationship of the new solutions, Find methods to evaluating the new solutions and experimental results