A Tutorial on Learned Multi-dimensional Indexes

Abdullah-Al-Mamun
Purdue University
mamuna@purdue.edu

Hao Wu
Purdue University
wu1112@purdue.edu

Walid G. Aref
Purdue University
aref@purdue.edu

ABSTRACT

Recently, Machine Learning (ML, for short) has been successfully applied to database indexing. Initial experimentation on *Learned Indexes* has demonstrated better search performance and lower space requirements than their traditional database counterparts. Numerous attempts have been explored to extend learned indexes to the multi-dimensional space. This makes learned indexes potentially suitable for spatial databases. The goal of this tutorial is to provide up-to-date coverage of learned indexes both in the single and multi-dimensional spaces. The tutorial covers over 25 learned indexes. The tutorial navigates through the space of learned indexes through a taxonomy that helps classify the covered learned indexes both in the single and multi-dimensional spaces.

CCS CONCEPTS

• **Database Systems** → **Indexing**; • **Machine Learning** → *ML for Systems*.

KEYWORDS

Learned Indexes, Spatial, Multi-dimensional

ACM Reference Format:
Abdullah-Al-Mamun, Hao Wu, and Walid G. Aref. 2020. A Tutorial on Learned Multi-dimensional Indexes. In *28th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '20), November 3–6, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397536.3426358

1 INTRODUCTION

Due to recent successes in the field of Machine Learning (ML, for short), two trends of research have emerged in the systems community: *Systems for ML* and *ML for Systems*. Systems for ML aims at building large-scale systems for efficient ML workloads. In contrast, ML for Systems aims at using ML-based approaches to replace core components of systems for better performance and less space requirement. This tutorial falls under the broad category of ML for System. More specifically, this tutorial addresses the following question: Can one use ML techniques to guide data indexing? Can ML techniques replace and act in place of a multi-dimensional index?

Database Management Systems (DBMS) are designed to be general purpose. This general purpose nature of a modern DBMS does not consider the specifics of a particular application and data of the user [21]. In most DBMSs, for efficient data access, an index structure, e.g., a B+-Tree is used. As a result, these index structures are highly optimized but are general-purpose data structures. In other words, they do not utilize knowledge of the underlying data distribution in the optimization process of an index. To illustrate, assume that we have 1 to 5M continuous integer keys. Now, in order to search a particular key, we can use the key itself (instead of a B+Tree) as an offset. As a result, the logarithmic complexity of search operation can be reduced to $O(1)$.

By addressing this issue, the first work on "Learned Index" [22] has changed the perception of DBMS indexing. The key idea behind the above mentioned work is that "Indexes are models" of the data. Given a key, say k, an index simply predicts the position of k in the dataset. As a result, indexes can be learned. Surprisingly, learned indexes have demonstrated better search performance and lower space requirements.

Although the term "Learned Indexes" has been popular very recently, the idea of using a learning mechanism in data indexing is not completely new. An example of an earlier index that uses ML techniques is the handwritten trie that uses hidden Markov Models on a trie structure to index the learned models [1]. However, this earlier work focuses on *indexing the learned models* in contrast to the more recent trend of *learning the index*. In this tutorial, we will cover both trends: (1) Indexing the learned models, and (2) Learning the index or what is termed the learned indexes.

These initial works have been focused on read-only workloads. To deal with updates, a new class of updatable adaptive learned indexes has been proposed, e.g., [4]. It has been demonstrated that a careful space-time trade-off can lead to an updatable data structure.

In the area of Spatial Database Indexing, support for multi-dimensional data is required. The R-Tree and its variants, e.g., [2, 12], and the quadtree and its variants, e.g., [9, 36, 37], are widely studied and are used extensively in practice. Initial attempts have been made to replace the R-Tree with a learned counterpart, e.g., [21]. A series of followup works have followed to build learned multi-dimensional indexes.

- Part 1: Learned Index Structures (20 minutes)
  - Introduction and Background (5 minutes)
  - Learned Single Dimensional Indexes (15 minutes)
- Part 2: Learned Multidimensional Indexes (25 minutes)
  - Motivation and Challenges (5 minutes)
  - State-of-the-art Learned Multidimensional Indexes (20 minutes)
- Part 3: Open Problems for Future Research (5 minutes)

Figure 1: The outline of the tutorial (50 minutes).

This tutorial will provide up-to-date coverage of learned multi-dimensional indexes. The target audience for the tutorial is students,

SIGSPATIAL '20, November 3–6, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-8019-5/20/11.
https://doi.org/10.1145/3397536.3426358

academics, researchers and practitioners with basic knowledge on data structures and algorithms. We assume basic understanding of fundamental data indexing structures e.g., the B-tree, the R-Tree, the quadtree, space-filling curves, and the Bloom Filter. The tutorial is designed to be self-contained in providing all the necessary background on the concepts related to the "Learned" part of the Index Structures. The target outcomes of this tutorial are as follows:

- Understanding the limitations of traditional multi-dimensional indexes.
- Understanding the motivation behind developing learned multi-dimensional indexes.
- Familiarity and up-to-date coverage of the state-of-the-art learned multi-dimensional index structures.
- Highlighting the research challenges and new opportunities in the area of learned multi-dimensional indexes.

2 OUTLINE OF THE TUTORIAL

This tutorial consists of two main parts as illustrated in Figure-1. The first part will contain the overall problem setting and the learned indexes introduced for the one-dimensional case. The second part of the tutorial will cover the existing work on learned multi-dimensional indexes. We will navigate through the space of learned indexes using a simple taxonomy that we develop over the literature on the existing learned indexes. A sample snapshot of the taxonomy is given below.

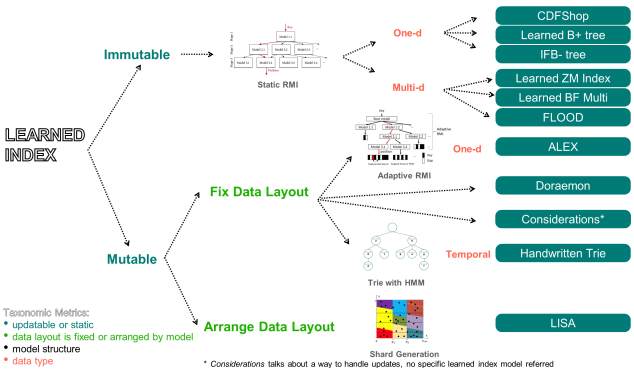

Figure 2: A sample taxonomy of Learned Index Structures

2.1 Part 1: Learned Index Structures

The chain of work in this particular area has started in 2018 with the paper titled "The Case for Learned Index Structures" [22]. In this paper, the key idea is that a one-dimensional index, e.g., the B+-Tree, can be treated as a learned model. For searching a key, a B+-Tree simply finds (predicts) the position of the key within a logical sorted array at the leaf level. If we follow this assumption, by learning the Cumulative Distribution Function (CDF) of the input data, the mapping function of an index can be learned. Due to the complexity of the CDF, a single ML model learned over the complete data cannot provide the desired accuracy [4]. To address this issue, a Recursive Model Index (RMI, for short) has been introduced. Several learned indexes utilize RMI, e.g., AIDEL [23], ASLM [25] and Hybrid-O [34]. A demo on tuning the learned indexes can be found in [29].

Handling dynamic data sets in the context of learned indexes is challenging. The reason is due to the following. Given a data set, it takes significant time to train an ML-based model to capture the CDF of the underlying data set. Given a new insert or update to the underlying dataset, this in a sense may change the CDF or at least perturb the distribution of the data and the learned model. Thus, upon multiple inserts, deletes, and updates, we need to retrain the model that, in term, reduces the utility of the learned index until it is retrained again. Updates and inserts in learned indexes have been addressed in several newly proposed indexes, e.g., [4, 13].

A series of followup works for addressing various aspects of learned indexes can be found in [7, 8, 17, 27, 30, 38, 40, 42, 43, 43, 44]. In [10], a data-aware index structure is proposed using interpolation. Hybrid approaches (with helper models) have been introduced [14, 15, 26]. Reinforcement Learning has been used for routing query and data in learned indexes [45]. Other related recent papers that we cover in the tutorial are: [19, 20, 28, 39] as well as a survey on learned data structures [6]. These as well as other learned indexes will be covered in Part 1 of this tutorial, and will serve as the foundation for Part 2 of the tutorial.

2.2 Part 2: Learned Multi-dimensional Indexes

Naturally, researchers have explored how to extend the concept of learned indexes into the multi-dimensional space. Several works have explored projecting the multi-dimensional data into the one-dimensional space as a preprocessing step, and then a learned index is built over the one-dimensional projection of the data (e.g., as in [21]. Flood [31] is another learned in-memory read-optimized index that automatically adapts itself to a particular multi-dimensional data set and workload. In [16], an interpolation-friendly multi-dimensional index has been proposed. LISA [24] is a disk-based learned multi-dimensional index. In [41], the Z-order space filling curve has been incorporated with the staged learning model to build a multi-dimensional index. Other recent works are: [3, 5, 11, 32, 33]. The tutorial will cover these multi-dimensional learned indexes, and demonstrate how they work and the challenges they face. Finally, the tutorial will conclude by listing several open problems for future research.

3 RELATED TUTORIALS

One related tutorial titled "From Auto-tuning One Size Fits All to Self-designed and Learned Data-intensive Systems (Tutorial)" [18] has been offered in SIGMOD 2019. Another very closely related tutorial offered in ICDE 2020 is titled "Machine Learning Meets Big Spatial Data" [35]. These tutorials address how ML approaches can be used in place of the various systems components. While these tutorials are complementary and are related, they are not directly focused on the recent hot topic of learned indexes that is the topic of this tutorial.

4 PRIOR TUTORIALS

As it stands, this tutorial has not been offered by the authors in any other venue, and SIGSPATIAL 2020 will be the first venue where this tutorial will be offered.

One of the authors, Walid G. Aref, has offered several tutorials on different yet related subjects in the past. These are listed below:

(1) Ahmed R. Mahmood and Walid G. Aref, "Query Processing Techniques for Big Spatial-Keyword Data", International Conference on Management of Data (SIGMOD): 1777-1782, 2017.
(2) Mohamed F. Mokbel and, Walid G. Aref, "Location-aware Query Processing and Optimization". In the IEEE International Conference on Mobile Data Management (MDM), Mannheim, Germany May 2007.
(3) Mohamed F. Mokbel and Walid G. Aref, "Location-aware Query Processing", In the International Conference on Extending Database Technology (EDBT), Munich, Germany, March 2006.
(4) Ihab F. Ilyas and Walid G. Aref, "Rank-aware Query Processing Tutorial", In the IEEE International Conference on Data Engineering, Japan, April 2005.
(5) Ihab F. Ilyas and Walid G. Aref. Rank-aware Query Processing Tutorial, the 9th International Conference on Extending Database Technology (EDBT), Heraklion - Crete, Greece, Mar. 2004.

5 BIOGRAPHIES

Abdullah-Al-Mamun is a Ph.D. student at the Department of Computer Science (CS), Purdue University. His research interest is in the area of Database Systems (DB) + Machine Learning (ML): "ML for DB" and "DB for ML". Particularly, he is interested in the area of Learned Multi-dimensional and Spatial Indexes. Previously, he completed his M.Sc. in CS from Memorial University of Newfoundland where he was a Fellow of the School of Graduate Studies.

Hao Wu is a senior undergraduate student at Purdue University with majors in Data Science, Statistics-Math, Aviation Management. He is interested in ML-oriented research as well as its application in data-driven multi-disciplinary projects.

Walid G. Aref is a professor of computer science at Purdue. His research interests are in extending the functionality of database systems in support of emerging applications, e.g., spatial, spatio-temporal, graph, biological, and sensor databases. He is also interested in query processing, indexing, data streaming, and geographic information systems (GIS). Walid's research has been supported by the National Science Foundation, the National Institute of Health, Purdue Research Foundation, CERIAS, Panasonic, and Microsoft Corp. In 2001, he received the CAREER Award from the National Science Foundation and in 2004, he received a Purdue University Faculty Scholar award. Walid is a member of Purdue's CERIAS. He is the Editor-in-Chief of the ACM Transactions of Spatial Algorithms and Systems (ACM TSAS), an editorial board member of the Journal of Spatial Information Science (JOSIS), and has served as an editor of the VLDB Journal and the ACM Transactions of Database Systems (ACM TODS). Walid has won several best paper awards including the 2016 VLDB ten-year best paper award. He is a Fellow of the IEEE, and a member of the ACM. Between 2011 and 2014, Walid has served as the chair of the ACM Special Interest Group on Spatial Information (SIGSPATIAL).

6 ACKNOWLEDGEMENTS

Walid G. Aref acknowledges the support of the National Science Foundation under Grant Numbers III-1815796 and IIS-1910216.

REFERENCES

[1] Walid Aref, Daniel Barbará, and Padmavathi Vallabhaneni. 1995. The Handwritten Trie: Indexing Electronic Ink. *SIGMOD Rec.* 24, 2 (May 1995), 151–162. https://doi.org/10.1145/568271.223811
[2] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. 1990. The R*-tree: an efficient and robust access method for points and rectangles. In *Proceedings of the 1990 ACM SIGMOD international conference on Management of data*. 322–331.
[3] Angjela Davitkova, Evica Milchevski, and Sebastian Michel. 2020. The ML-Index: A Multidimensional, Learned Index for Point, Range, and Nearest-Neighbor Queries.. In *EDBT*. 407–410.
[4] Jialin Ding, Umar Farooq Minhas, Hantian Zhang, Yinan Li, Chi Wang, Badrish Chandramouli, Johannes Gehrke, Donald Kossmann, and David Lomet. 2019. ALEX: An Updatable Adaptive Learned Index. *arXiv preprint arXiv:1905.08898* (2019).
[5] Jialin Ding, Vikram Nathan, Mohammad Alizadeh, and Tim Kraska. 2020. Tsunami: A Learned Multi-dimensional Index for Correlated Data and Skewed Workloads. *arXiv preprint arXiv:2006.13282* (2020).
[6] Paolo Ferragina and Giorgio Vinciguerra. 2020. Learned Data Structures. In *Recent Trends in Learning From Data*, Luca Oneto, Nicolò Navarin, Alessandro Sperduti, and Davide Anguita (Eds.). Springer International Publishing, 5–41. https://doi.org/10.1007/978-3-030-43883-8_2
[7] Paolo Ferragina and Giorgio Vinciguerra. 2020. The PGM-index. *Proceedings of the VLDB Endowment* 13, 8 (Apr 2020), 1162–1175. https://doi.org/10.14778/3389133.3389135
[8] Paolo Ferragina, Giorgio Vinciguerra, and Michele Miccinesi. 2019. Superseding traditional indexes by orchestrating learning and geometry. *arXiv preprint arXiv:1903.00507* (2019).
[9] Raphael A. Finkel and Jon Louis Bentley. 1974. Quad trees a data structure for retrieval on composite keys. *Acta informatica* 4, 1 (1974), 1–9.
[10] Alex Galakatos, Michael Markovitch, Carsten Binnig, Rodrigo Fonseca, and Tim Kraska. 2019. Fiting-tree: A data-aware index structure. In *Proceedings of the 2019 International Conference on Management of Data*. 1189–1206.
[11] Behzad Ghaffari, Ali Hadian, and Thomas Heinis. 2020. Leveraging Soft Functional Dependencies for Indexing Multi-dimensional Data. *arXiv preprint arXiv:2006.16393* (2020).
[12] Antonin Guttman. 1984. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*. 47–57.
[13] Ali Hadian and Thomas Heinis. 2019. Considerations for handling updates in learned index structures. In *Proceedings of the Second International Workshop on Exploiting Artificial Intelligence Techniques for Data Management*. ACM, 3.
[14] Ali Hadian and Thomas Heinis. 2019. Interpolation-friendly B-trees: Bridging the gap between algorithmic and learned indexes. In *22nd International Conference on Extending Database Technology (EDBT 2019)*. https://doi.org/10.5441/002/edbt.2019.93
[15] Ali Hadian and Thomas Heinis. 2020. MADEX: Learning-augmented Algorithmic Index Structures. In *Proceedings of the 2nd International Workshop on Applied AI for Database Systems and Applications*.
[16] Ali Hadian, Ankit Kumar, and Thomas Heinis. 2020. Hands-off Model Integration in Spatial Index Structures. In *Proceedings of the 2nd International Workshop on Applied AI for Database Systems and Applications*.
[17] Benjamin Hilprecht, Andreas Schmidt, Moritz Kulessa, Alejandro Molina, Kristian Kersting, and Carsten Binnig. 2020. DeepDB: Learn from Data, Not from Queries! 13, 7 (2020).
[18] Stratos Idreos and Tim Kraska. 2019. From Auto-tuning One Size Fits All to Self-designed and Learned Data-intensive Systems (Tutorial). In *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*. 2054–2059. http://people.csail.mit.edu/kraska/pub/sigmod19tutorialpart2.pdf
[19] Andreas Kipf, Ryan Marcus, Alexander van Renen, Mihail Stoian, Alfons Kemper, Tim Kraska, and Thomas Neumann. 2019. SOSD: A Benchmark for Learned Indexes. *ArXiv* abs/1911.13014 (2019).
[20] Andreas Kipf, Ryan Marcus, Alexander van Renen, Mihail Stoian, Alfons Kemper, Tim Kraska, and Thomas Neumann. 2020. RadixSpline: A Single-Pass Learned Index. *ArXiv* abs/2004.14541 (2020).
[21] Tim Kraska, Mohammad Alizadeh, Alex Beutel, Ed H Chi, Jialin Ding, Ani Kristo, Guillaume Leclerc, Samuel Madden, Hongzi Mao, and Vikram Nathan. 2019. Sagedb: A learned database system. (2019).
[22] Tim Kraska, Alex Beutel, Ed H Chi, Jeffrey Dean, and Neoklis Polyzotis. 2018. The case for learned index structures. In *Proceedings of the 2018 International Conference on Management of Data*. ACM, 489–504.
[23] Pengfei Li, Yu Hua, Pengfei Zuo, and Jingnan Jia. 2019. A Scalable Learned Index Scheme in Storage Systems. *CoRR* abs/1905.06256 (2019). arXiv:1905.06256 http://arxiv.org/abs/1905.06256
[24] Pengfei Li, Hua Lu, Qian Zheng, Long Yang, and Gang Pan. 2020. LISA: A Learned Index Structure for Spatial Data. *SIGMOD* (2020).

[25] Xin Li, Jingdong Li, and Xiaoling Wang. 2019. ASLM: Adaptive single layer model for learned index. In *International Conference on Database Systems for Advanced Applications*. Springer, 80–95.

[26] A Llavesh, Utku Sirin, R West, and A Ailamaki. 2019. Accelerating b+ tree search by using simple machine learning techniques. In *Proceedings of the 1st International Workshop on Applied AI for Database Systems and Applications*.

[27] Stephen Macke, Alex Beutel, Tim Kraska, Maheswaran Sathiamoorthy, Derek Zhiyuan Cheng, and EH Chi. 2018. Lifting the curse of multidimensional data with learned existence indexes. In *Workshop on ML for Systems at NeurIPS*.

[28] Ryan Marcus, Andreas Kipf, Alexander van Renen, Mihail Stoian, Sanchit Misra, Alfons Kemper, Thomas Neumann, and Tim Kraska. 2020. Benchmarking Learned Indexes. *arXiv preprint arXiv:2006.12804* (2020).

[29] Ryan Marcus, Emily Zhang, and Tim Kraska. 2020. CDFShop: Exploring and Optimizing Learned Index Structures. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 2789–2792.

[30] Michael Mitzenmacher. 2018. A model for learned bloom filters and optimizing by sandwiching. In *Advances in Neural Information Processing Systems*. 464–473.

[31] Vikram Nathan, Jialin Ding, Mohammad Alizadeh, and Tim Kraska. 2020. Learning Multi-dimensional Indexes. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 985–1000.

[32] Varun Pandey, Alexander van Renen, Andreas Kipf, Ibrahim Sabek, Jialin Ding, and Alfons Kemper. 2020. The Case for Learned Spatial Indexes. *arXiv preprint arXiv:2008.10349* (2020).

[33] Jianzhong Qi, Guanli Liu, Christian S Jensen, and Lars Kulik. 2020. Effectively learning spatial indices. *Proceedings of the VLDB Endowment* 13, 12 (2020), 2341–2354.

[34] Wenwen Qu, Xiaoling Wang, Jingdong Li, and Xin Li. 2019. Hybrid indexes by exploring traditional B-tree and linear regression. In *International Conference on Web Information Systems and Applications*. Springer, 601–613.

[35] Ibrahim Sabek and Mohamed F Mokbel. 2020. Machine learning meets big spatial data. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 1782–1785.

[36] Hanan Samet. 1984. The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)* 16, 2 (1984), 187–260.

[37] Hanan Samet. 2006. *Foundations of multidimensional and metric data structures*. Morgan Kaufmann.

[38] Chuzhe Tang, Zhiyuan Dong, Minjie Wang, Zhaoguo Wang, and Haibo Chen. 2019. Learned Indexes for Dynamic Workloads. *arXiv preprint arXiv:1902.00655* (2019).

[39] Chuzhe Tang, Youyun Wang, Zhiyuan Dong, Gansen Hu, Zhaoguo Wang, Minjie Wang, and Haibo Chen. 2020. XIndex: a scalable learned index for multicore data storage. *Proceedings of the 25th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (2020).

[40] Peter Van Sandt, Yannis Chronis, and Jignesh M Patel. 2019. Efficiently Searching In-Memory Sorted Arrays: Revenge of the Interpolation Search?. In *Proceedings of the 2019 International Conference on Management of Data*. ACM, 36–53.

[41] Haixin Wang, Xiaoyi Fu, Jianliang Xu, and Hua Lu. 2019. Learned Index for Spatial Queries. In *2019 20th IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 569–574.

[42] Youyun Wang, Chuzhe Tang, Zhaoguo Wang, and Haibo Chen. 2020. SIndex: a scalable learned index for string keys. In *Proceedings of the 11th ACM SIGOPS Asia-Pacific Workshop on Systems*. 17–24.

[43] Yingjun Wu, Jia Yu, Yuanyuan Tian, Richard Sidle, and Ronald Barber. 2019. Designing Succinct Secondary Indexing Mechanism by Exploiting Column Correlations. *arXiv preprint arXiv:1903.11203* (2019).

[44] Wenkun Xiang, Hao Zhang, Rui Cui, Xing Chu, Keqin Li, and Wei Zhou. 2018. Pavo: A RNN-Based Learned Inverted Index, Supervised or Unsupervised? *IEEE Access* 7 (2018), 293–303.

[45] Zongheng Yang, Badrish Chandramouli, Chi Wang, Johannes Gehrke, Yinan Li, Umar Farooq Minhas, Per-Åke Larson, Donald Kossmann, and Rajeev Acharya. 2020. Qd-tree: Learning Data Layouts for Big Data Analytics. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 193–208.