Indexability-Based Dataset Partitioning

Hoyos, Angello; Ruiz, Ubaldo; Marchand-Maillet, Stéphane; Chávez, Edgar

How to cite

HOYOS, Angello et al. Indexability-Based Dataset Partitioning. In: 12th International Conference on Similarity Search and Applications, SISAP 2019. Newark, NJ (USA). Cham : Springer International Publishing, 2019. p. 143–150. doi: 10.1007/978-3-030-32047-8_13

This publication URL: https://archive-ouverte.unige.ch/unige:128619
Publication DOI: 10.1007/978-3-030-32047-8_13

Indexability-based dataset partitioning

Angello Hoyos[1], Ubaldo Ruiz[1],
Stephane Marchand-Maillet[2], and Edgar Chávez[1]

[1] Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE), México
[2] Viper group,
Department of Computer Science,
CUI - University of Geneva
ahoyos@cicese.edu.mx
uruiz@cicese.mx
Stephane.Marchand-Maillet@unige.ch
elchavez@cicese.mx

Abstract. Indexing exploits assumptions on the inner structures of a dataset to make the nearest neighbor queries cheaper to resolve. Datasets are generally indexed at once into a unique index for similarity search. By indexing a given dataset as a whole, one faces the parameters of its global structure, which may be adverse. A typical well-studied example is a high global dimensionality of the dataset, making any indexing strategy inefficient due to the curse of dimensionality.
We conjecture that a dataset may be partitioned into subsets of variable indexability. The strategy is, therefore, to define a procedure to extract parts of the dataset with predictable indexability and to adapt the index structure to this parameter.
In this paper, we define and discuss indexability related to the curse of dimensionality and propose a related heuristic to partition the dataset into low-dimensional parts. Each data object is ranked according to its degree centrality, under a connected sparse graph, the Half-Space Proximal Graph (HSP). We postulate centrality measures are good predictors of dimensionality and indexability.
In view of validation, we conducted an experiment using the degree centrality of the HSP graph as unique dimensionality/indexability measure. We ranked the data objects by their respective centrality degree under the HSP graph, then extracted the lower dimensional subsets, recomputed the HSP and repeated. Subsets were then indexed with an exact method in increasing, decreasing and random order. We measured the complexity of a fixed set of queries for each of the three arrangements. For each set we used a fixed dataset with 250 queries.
The above single experiment demonstrated that the heuristic can extract low dimensional subsets, and also that those subsets are easier to index. This initial results demonstrate the validity of our conjecture and motivate the need for exploring further the notion of indexability and related dataset partitioning strategies.

Keywords: Indexability · Dataset partitioning · Spanning graph · Centrality measure · Curse of dimensionality

1 Introduction

The nearest neighbor search in a dataset is at the core of data analysis because it is via neighborhoods that the data makes sense, as opposed to being a set of arbitrary unrelated items. Resolving effectively range queries or the k-nearest neighbor problem has countless applications in machine learning, data mining and many other fields of data processing. It is therefore critical to make this step both effective and accurate. It is well-known that the effectiveness of indexing structures is reduced as a function of the dimensionality of the dataset. In this paper we present a study that takes an alternative approach to the general index structure improvement proposed in most of the literature. Under the assumption that effective index structures exist for "well-behaved" datasets, we propose to attack the dataset rather than the index structure and make it suitable to be indexed by state-of-the-art structures (eg [14]).

In section 2, we briefly review related work and introduce the notion *dataset indexability*, that will be our criterion for adapting the dataset to index structures. In section 3, we present our strategy to boost indexability, resulting into our main conjecture that is initially tested in section 4 and discussed in section 5.

2 Indexability

Measuring the *performance of an index structure* generally means evaluating the performance of an indexing strategy over standard benchmarks (datasets, queries and measures). Measuring the *indexability of a dataset* takes the problem upside down and looks at whether or not a given dataset can benefit from an index structure to answer nearest neighbor queries. Intuitively, a dataset is said to be indexable if one can build an exact index able to answer reasonably selective queries in time that is not proportional to the size of the dataset. A trivial example of an indexable dataset is a set of points on a line, the plane or with "small" dimension in general. In this example case, a classical data structure like the *kd*-tree [3], can handle the indexing task.

There are several dimensions to index fitness. Any index computes an index distance that approximates the true metric while being cheaper to compute. The effectiveness of the index measures how good the index distance bounds the original distance. The efficiency measures how fast the index distance can be computed for the entire dataset. These two measures are complemented with the memory usage and the speed at which the index can be constructed. It is usually the case that the effectiveness of an index can be boosted at the expense of its efficiency and memory usage or construction speed.

2.1 The importance of local dimensionality

Indexability is therefore related to the deep foundations of distance-based indexing, essentially related to distance computation. From this perspective, indexability has been studied in relation to the curse of dimensionality and much

has been discussed around this concept. Essentially, the main result of [4] and subsequent papers (eg, [13, 17, 19]) is that, as defined in [20] (definition 2.2), a *workload* $W = (S, F, n, d)$ consisting of a dataset S of n objects drawn iid from a distribution F and measured via distance function $d(.,.)$ can be made into a series W_i which will be said to have *vanishing variance* if there exists $\alpha > 0$ such that

$$\lim_{m\to\infty} \text{var}\left(\frac{D_m^\alpha}{\mathbb{E}[D_m^\alpha]}\right) = 0,$$

where D_m is the distance distribution of W_m (ie the distribution of distances between points in S_m). In that case, ([4], Theorem 1), for every $\varepsilon > 0$

$$\lim_{m\to\infty} P[D_m^{\max} \leq (1+\varepsilon) D_m^{\min}] = 1.$$

Simply said, all distance values become indistinguishable as m increases. This is even more true in a fixed precision environment. A typical example of such a workload is a dataset with coordinates *iid* distributed in all m dimensions. As a result, the use of sum-based distance functions (such as Minkowski metrics) for high-dimensional datasets impedes their indexability.

Directly considering the global dimensionality of the dataset therefore appears as a crude approximation for indexability. Rather, provided one knows how to exploit local structures from within the data, the effective indexability should be boosted. The workload may have high representational dimension but an intrinsic low dimension, and be indexable using a classic metric indexing method like the BK-tree [6]. For other cases of intrinsic high dimension, the dataset would not be indexable, even in the approximate sense, as stated in a recent theoretical result on the conditional hardness of nearest neighbor search using polynomial preprocessing time [19]. In that paper the authors prove that computing a $(1+\epsilon)$-approximation to the nearest neighbor requires $\Omega(N - \delta)$ time, with N the size of the dataset.

It is therefore critical to obtain a proper understanding of what dimensionality means locally. There are several proposals to measure local intrinsic dimension. An excellent review is provided by Michael Houle [10], who also proposed the *expansion dimension* for that purpose. The idea is to measure locally how many points are contained in a ball as its radius increases. Since in Euclidean spaces the volume of a ball of radius r is about r^m with m the number of dimensions, fitting the increase of the number of points contained in a ball of increasing radius allows for the estimation of the local intrinsic dimension. In this line of work, authors [11, 2] advocate for feature selection for removing "spurious" dimensions while preserving original distances. The aim is to provide an equivalent but better indexable dataset. Alternatively, ranks may also be used as a robust replacement to distance values [12, 7].

2.2 Dataset shattering

An interesting alternative avenue for investigation is that of the VC dimension [22]. The relationship between the VC dimension and indexing has already been

put forward by Pestov in [18]. Although the VC dimension is related to measuring the complexity of a class of functions, the notion of *shattering* is easily related to that of indexing. If a dataset is shattered, any of its elements can be particularized as a result of such shattering. Indexing has a similar objective. For example, the capabilities of permutation-based indexing schemes to shatter a dataset are explored in [15, 1].

3 Boosting dataset indexability

In this exploratory work, we propose an alternative approach to combat the curse of dimensionality. Rather than considering the dataset as an integral entity, we seek a decomposition that will extract parts with higher indexability than the whole. Indexes can then be built over these parts individually and a query sent to the multiple index structures and recomposed globally.

3.1 Dataset layering

We assume we are given a non-indexable dataset. Our aim is to decompose it into easily indexable parts. From the above discussion, non-indexability allows us to model the dataset as a blob of high dimension, which we will partition into fragments of low dimension.

Hence, we construct a partition by iteratively *peeling* the dataset (blob) into layers corresponding to surfaces of points equidistant from the blob center. We therefore inherit from the notion of centrality measures to define the layers which will be indexable. Centrality is classically defined in relation to a spanning graph. Various definitions of centrality exist [5, 9], from the simplest based on node degree, to those exploiting a spectral decomposition of the graph (such as PageRank and others [21]).

We initially base our study on a degree-based centrality measure applied over the Half Space Proximal (HSP) graph constructed over the dataset, as detailed next.

3.2 The Half Space Proximal graph

The Half Space Proximal is a local test for building a directed graph, which is a bounded dilation spanner over a set of objects in a metric space. Without needing synchronization, each node can compute its neighbors using the simple rule described below.

Let S a finite subset of a metric space. Let $u \in S$, we take its nearest element $v \in S$ and add an edge from u to v. We remove all the elements that are closer to v than to u. The region of objects closer to v than to u is called the *forbidden region* from the point u with respect to v. From the remaining points we take the nearest point to u and repeat until we have removed all points in S. We do this process for every point in S. In the end, we will have a directed graph

with vertex set S and the edges found with the previous mechanism. The HSP, presented in [8], has maximum out-degree of six for points in the plane.

We conjecture that the out-degree of each node in the HSP depends only on the local intrinsic dimension of the node. Hence, in particular, it can be used as an estimator of the indexability of a point collection. The rationale behind this conjecture is related to the test conducted at each step of the construction. Every edge from the node is associated to an hyperplane, and the out-degree will be related to the number of hyperplanes needed to isolate the node.

Please notice that the HSP test in each node requires searching for the nearest neighbor of the node, then splitting the set into two parts and repeat until the set is empty. A careful implementation will require a quadratic number of distance computations. This imposes a severe limitation in practice, because interesting datasets are quite large.

4 Layered Indexing with the HSP

As an empirical validation of the above stated conjecture, we conducted an experiment using a set of 100'000 deep feature vectors of 4'096 real values. For this set we computed the HSP and ranked the nodes according to their degree. After this, we removed the 1'000 nodes with the smallest degree in the graph, recomputed the HSP in the remaining objects, and repeated. Note that the nodes linked to the removed objects are the most likely to have its out-degree modified. We only recomputed the edges of those nodes in the next iteration. Figure 1 shows the evolution of the average degree centrality when adding different layers in the dataset.

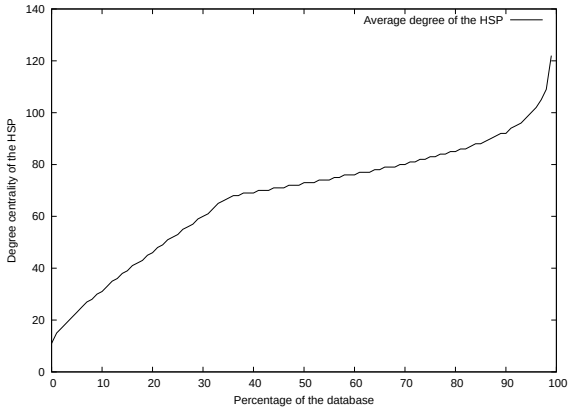

Fig. 1. Average degree centrality of different layers of dataset objects.

In this experiment, we noted that the number of changes in the out-degree of the touched nodes was slowly decreasing, and after some 40% of the dataset, stopped changing. This supports the existence of a hard kernel in the dataset.

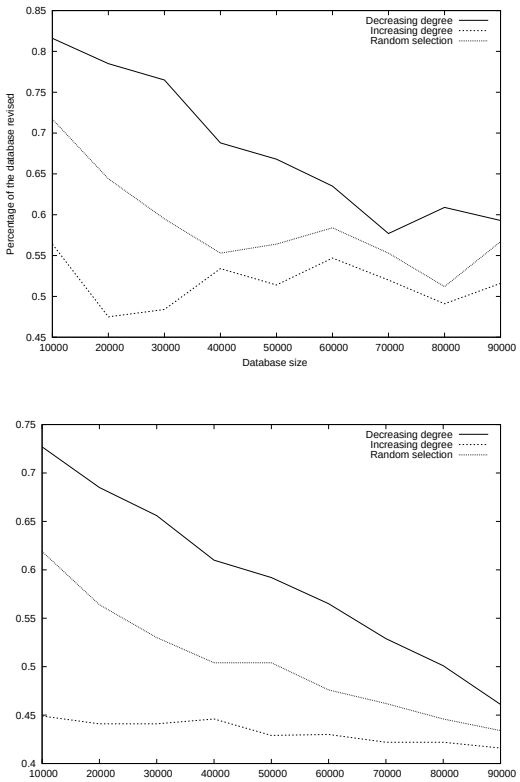

Fig. 2. Layered indexing of a dataset of 100,000 deep feature vectors of dimension 4096 with two exact indices, with the SAT (top) and with the VP-tree (bottom). Note the large difference for the low and large degree nodes. See text for more details.

The result of this layered indexing experiment is summarized in figure 2. In the plot, chunks of increasing size (horizontal axis) are indexed independently with an exact indexing method (SAT [16] and VP-Tree [23] respectively). Increasing the size of the dataset from 10'000 to 90'000 objects was first done by adding nodes of decreasing degree. According to our conjecture, this corresponds to going from least to most indexable subsets (least favorable indexing setup). We compare to the case of increasing degree, again varying the size from 10'000 to 90'000 objects, hoping to create the most favorable setup. We also included a control plot with a random selection of the dataset of the same size. For the rest, we kept the same index and the same set of 250 queries not included at indexing time. The results plotted correspond to the average over 250 queries in the index.

The differences become apparent, in some places it was almost twice the number of distance computations (indicated by the percentage of the dataset visited on vertical axis). This difference becomes smaller when the subset is almost the entire dataset.

Notice that the difference in indexability persists across different indexes. The SAT is more sensitive the centrality of the collection, while the VP-tree is almost not affected when the dataset excludes its 10% least indexable part.

5 Discussion

The preliminary experimental results discussed in this communication are encouraging. They are an empirical corroboration of the intuition that indexability, local intrinsic dimensionality and centrality are related. This paper certainly does not propose a new indexing method, mainly because of the large cost of computing the degree centrality of the HSP graph. It rather motivates the quest for a faster-to-compute latent graph of the dataset and gives some hope in dealing with the curse of dimensionality.

Some open questions remain. What type of guarantees is it possible to give in a layered index? In other words, assume each part, from the most to least indexable, is indexed independently using a mixture of exact and approximate methods, and then queried at once, there will be an answer from each one of the indexes, some from the exact and some from the approximate methods. Even if the nearest neighbor belongs to an exact index, it is not sure that it is the true global nearest neighbor. What is then a good heuristic to assign a probability to the global answer?

It is also interesting to explore additional properties of the most or least indexable parts of the dataset. In the case of such deep features of images, what are the most representative objects of a class? Is it the most central, i.e. the least indexable? Or is it the opposite? Is it possible to build a classifier based only on the centrality of the objects in a class?

References

1. Amato, G., Falchi, F., Rabitti, F., Vadicamo, L.: Some theoretical and experimental observations on permutation spaces and similarity search. In: Similarity Search and Applications - 7th International Conference, SISAP 2014, Los Cabos, Mexico, October 29-31, 2014. Proceedings. pp. 37–49 (2014)
2. Amsaleg, L., Chelly, O., Furon, T., Girard, S., Houle, M.E., Kawarabayashi, K., Nett, M.: Extreme-value-theoretic estimation of local intrinsic dimensionality. Data Mining and Knowledge Discovery **32**(6), 1768–1805 (2018)
3. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Communications of the ACM **18**(9), 509–517 (1975)
4. Beyer, K.S., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is "nearest neighbor" meaningful? In: Proceedings of the 7th International Conference on Database Theory. pp. 217–235. ICDT '99, Springer-Verlag, London, UK, UK (1999)

5. Boldi, P., Vigna, S.: Axioms for centrality. Internet Mathematics **10**(3-4), 222–262 (2014)
6. Burkhard, W.A., Keller, R.M.: Some approaches to best-match file searching. Communications of the ACM **16**(4), 230–236 (1973)
7. Chavez, E., Figueroa, K., Navarro, G.: Effective proximity retrieval by ordering permutations. Pattern Analysis and Machine Intelligence, IEEE Transactions on **30**(9), 1647–1658 (Sept 2008)
8. Chavez, E., Dobrev, S., Kranakis, E., Opatrny, J., Stacho, L., Tejeda, H., Urrutia, J.: Half-space proximal: A new local test for extracting a bounded dilation spanner of a unit disk graph. In: International Conference On Principles Of Distributed Systems. pp. 235–245. Springer (2005)
9. Grando, F., Granville, L.Z., Lamb, L.C.: Machine learning in network centrality measures: Tutorial and outlook. ACM Comput. Surv. **51**(5), 102:1–102:32 (Oct 2018)
10. Houle, M.E.: Local intrinsic dimensionality I: an extreme-value-theoretic foundation for similarity applications. In: Similarity Search and Applications - 10th International Conference, SISAP 2017, Munich, Germany, October 4-6, 2017, Proceedings. pp. 64–79 (2017)
11. Houle, M.E., Ma, X., Oria, V., Sun, J.: Efficient algorithms for similarity search in axis-aligned subspaces. In: Similarity Search and Applications - 7th International Conference, SISAP 2014, Los Cabos, Mexico, October 29-31, 2014. Proceedings. pp. 1–12 (2014)
12. Houle, M.E., Nett, M.: Rank-based similarity search: Reducing the dimensional dependence. IEEE Trans. Pattern Anal. Mach. Intell. **37**(1), 136–150 (2015)
13. Hsu, C.M., Chen, M.S.: On the Necessary and Sufficient Conditions of a Meaningful Distance Function for High Dimensional Data Space, pp. 12–23 (2006)
14. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. CoRR **abs/1702.08734** (2017)
15. Marchand-Maillet, S., Roman-Rangel, E., Mohamed, H., Nielsen, F.: Quantifying the invariance and robustness of permutation-based indexing schemes. In: Similarity Search and Applications - 9th International Conference, SISAP 2016, Tokyo, Japan, October 24-26, 2016. Proceedings. pp. 79–92 (2016)
16. Navarro, G.: Searching in metric spaces by spatial approximation. The VLDB Journal **11**(1), 28–46 (2002)
17. Pestov, V.: On the geometry of similarity search: Dimensionality curse and concentration of measure. Information Processing Letters **73**(1-2) (Jan 2000)
18. Pestov, V.: Indexability, concentration, and VC theory. Journal of Discrete Algorithms **13**, 2–18 (2012)
19. Rubinstein, A.: Hardness of approximate nearest neighbor search. In: Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing. pp. 1260–1268. STOC 2018, ACM, New York, NY, USA (2018)
20. Shaft, U., Ramakrishnan, R.: Theory of nearest neighbors indexability. ACM Trans. Database Syst. **31**(3), 814–838 (Sep 2006)
21. Sun, K., Morrison, D., Bruno, E., Marchand-Maillet, S.: Learning representative nodes in social networks. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) Advances in Knowledge Discovery and Data Mining. pp. 25–36. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
22. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer-Verlag, Berlin, Heidelberg, 2nd edn. (2000)
23. Yianilos, P.N.: Data structures and algorithms for nearest neighbor search in general metric spaces. In: SODA. vol. 93, pp. 311–21 (1993)