On Applications of Parameterized Hyperplane Partitioning

Jakub Lokoč, Tomáš Skopal
SIRET Research Group
Department of Software Engineering, Faculty of Mathematics and Physics, Charles University in Prague
Malostranské nám. 25, 118 00 Prague, Czech Republic
http://siret.ms.mff.cuni.cz

ABSTRACT

The efficient similarity search in metric spaces is usually based on several low-level partitioning principles, which allow filtering of non-relevant objects during the search. In this paper, we propose a parameterizable partitioning method based on the generalized hyperplane partitioning (GHP), which utilizes a parameter to adjust "borders" of the partitions. The new partitioning method could be employed in the existing metric indexes that are based on GHP (e.g., GNAT, M-index). Moreover, we could employ the parameterizable GHP in the role of a new multi-example query type, defined as a partition determined by an available query object and several "anti-example" objects. We believe that both applications of parameterizable GHP can soon take place in metric access methods and new query models.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: [Retrieval models]

1. INTRODUCTION

The similarity search using the metric space model proved to be a general approach applicable in various domains. A database $\mathbb{S}$ is supposed to consist of unstructured raw data, while the only available information is a metric distance δ defined for each pair of objects from $\mathbb{S}$. From the database management point of view, the search performance is crucial, hence indexes allowing efficient filtering are necessary. During the last two decades, there emerged many metric access methods (MAMs) [1], [3] that utilize metric postulates to filter non-relevant (sets of) objects during query processing. One of the most fundamental rules used in the design of efficient MAMs is *metric partitioning*, which divides the database into separate classes of similar objects. The general metric space model offers two basic types of partitioning – the *ball partitioning* and the *generalized hyperplane partitioning* (GHP). The ball partitioning employs a selected object (so-called *pivot*) and a radius, dividing the space in two partitions (e.g., in M-tree) or more, when a combination of balls is used (e.g., in MVP-tree or PM-tree). The GHP uses up to n pivots to divide the space into n voronoi-like partitions, where the i^{th} partition consists of objects that are closer to the i^{th} pivot than to any other pivot (e.g., in GNAT or M-index). In this paper, we propose a parameterized extension of the GHP, which employs a parameter to adjust "borders" of the partitions. We also prove several lemmas necessary for correct filtering rules.

SISAP 2010, September 18-19, 2010, Istanbul, Turkey.

2. PARAMETERIZED GENERALIZED HYPERPLANE PARTITIONING

For the lack of the space, we will define partitioning just for two pivots, but the definition and all the presented lemmas can be simply extended for an arbitrary number of pivots.

DEFINITION 1. *The parameterized generalized hyperplane partitioning (pGHP) using two pivots a, b and a parameter c divides a database $\mathbb{S}$ into two subsets A, B as follows:*

$$A = \{x | x \in \mathbb{S} \wedge \delta(a,x) < \delta(b,x) + c\}$$
$$B = \{x | x \in \mathbb{S} \wedge \delta(a,x) \geq \delta(b,x) + c\}$$

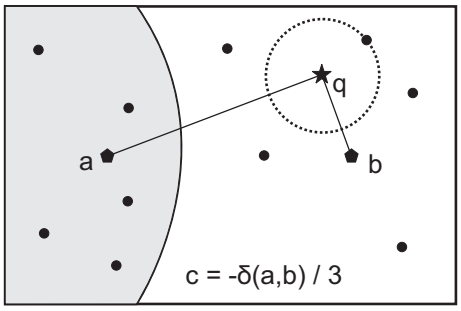

Figure 1: Parameterized generalized hyperplane partitioning.

Since the ball-region is the most frequent shape of a query, we also propose a lemma giving the relation between pGHP and the ball region (see also Figure 1).

LEMMA 1. *Let (q, r) be a range query and a, b be pivots used for the pGHP with parameter c, then (a) if $\delta(a,q) - r \geq \delta(b,q) + r + c$ then $\forall x$ where $\delta(x,q) \leq r$ holds $\delta(a,x) \geq \delta(b,x) + c$ and (b) if $\delta(a,q) + r < \delta(b,q) - r + c$ then $\forall x$ where $\delta(x,q) \leq r$ holds $\delta(a,x) < \delta(b,x) + c$.*

PROOF. Let us start with variant (a). By combination of assumptions $\delta(x,q) \leq r$ and $\delta(a,q) - r \geq \delta(b,q) + r + c$ we obtain $\delta(a,q) - \delta(x,q) \geq \delta(b,q) + \delta(x,q) + c$. From the triangle inequality we also have $\delta(a,x) \geq \delta(a,q) - \delta(x,q)$ and $\delta(b,q) + \delta(x,q) + c \geq \delta(b,x) + c$. Thus, by combination of these formulas we obtain $\delta(a,x) \geq \delta(a,q) - \delta(x,q) \geq \delta(b,q) + \delta(x,q) + c \geq \delta(b,x) + c$ which implies $\delta(a,x) \geq \delta(b,x) + c$. The proof of the variant (b) is similar. □

As a direct consequence of Lemma 1, we obtain a filtering rule for searching by ball-shaped queries. The filtering behavior of pGHP is similar as for the original GHP. In Figure 2, the two dotted curves depict the centers of query balls with fixed radius that have to visit both partitions (two such query balls are depicted). Note that pruning near the line connecting a and b is more effective (the border thickness is here close to $2r$), however this property is not specific to pGHP. Even in the original GHP the pruning near to the "connecting line" is more effective due to tighter lower- and upper-bound distances to the points in the query ball.

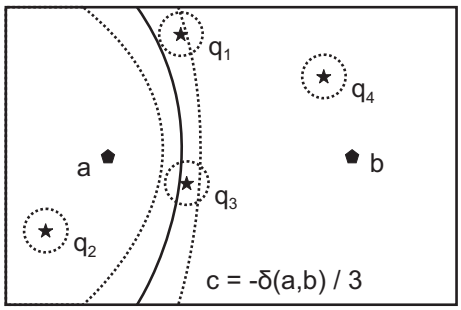

Figure 2: Filtering ability of pGHP

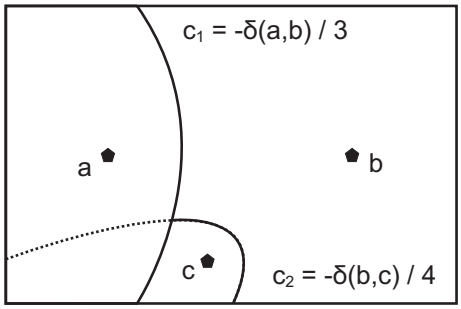

Figure 3: pGHP using multiple pivots

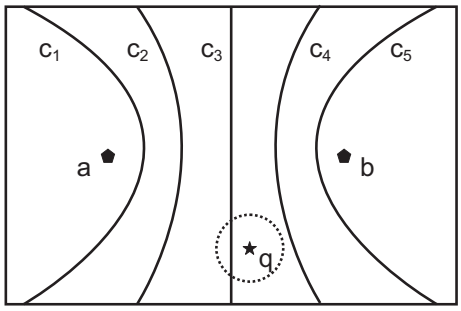

Figure 4: pGHP using multiple parameters

Unlike the original GHP, the pGHP utilization could be used for more flexible data partitioning.

1. We can dynamically adjust the parameter c in order to establish a balanced partitioning of data. The balancing is even more important if more than two pivots are used, e.g., in a future variant of GNAT based on pGHP. In Figure 3, the pGHP partitioning using pivots a, b, c was two-step, first the space was partitioned between a and b, while the partition b was further divided into partitions b and c.
2. We can employ more parameters c_i to define more partitions (as depicted in Figure 4). By the use of multiple parameters we can define multiple partitions, however, the interval of useful values c is bounded, as shown in the following lemma.

LEMMA 2. *Let a, b be pivots used for the pGHP with parameter c, then (a) if $c \leq -\delta(a,b)$ then partition A is always empty and (b) if $c > \delta(a,b)$ then partition B is always empty.*

PROOF. We just substitute a border value for c in the partitioning rule and we immediately obtain the statement. □

Besides indexing, a completely new family of multi-example query types based on pGHP could be defined within the framework proposed in [2]. An example of such a query is depicted, again, in Figure 3. The query region/partition c could be interpreted such that we search for objects similar to c (the example) and not very similar to a, b (anti-examples). Note that efficient processing of the new query is easy when using MAMs based on ball-partitioning (e.g., M-tree). Instead of having a query ball and a pGHP-defined partition in index (e.g., a sort of pGHP-based GNAT), we just interchange the roles of query and index regions, using the same pGHP filtering.

Acknowledgments

This research has been supported in part by Czech Science Foundation project Nr. 201/09/0683 and by the grant SVV-2010-261312.

3. CONCLUSIONS

We have introduced pGHP, a parameterized version of GHP that can be utilized for definition of flexible hyperplane-based metric regions. The pGHP can be used to modify the existing MAMs based on the GHP. We can also define specific query regions using more pivots in the role of examples and anti-examples. All the presented ideas concerning indexing and querying are the topic of our future work.

4. REFERENCES

[1] E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín. Searching in Metric Spaces. *ACM Computing Surveys*, 33(3):273–321, 2001.

[2] P. Ciaccia, M. Patella, and P. Zezula. Processing complex similarity queries with distance-based access methods. In *Proceedings of the 6th International Conference on Extending Database Technology (EDBT 98)*, pages 9–23, 1998.

[3] P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity Search: The Metric Space Approach (Advances in Database Systems)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.