Proximity Graphs for Similarity Searches: Experimental Survey and the New Connected-Partition Approach HGraph

Larissa C. Shimomura[1]*, **Daniel S. Kaster**[1] **(supervisor)**

[1]Graduate Program in Computer Science
Department of Computer Science
University of Londrina – Londrina, PR – Brazil

l.capobianco.shimomura@tue.nl, dskaster@uel.br

Abstract. *Similarity searching is a widely used approach to retrieve complex data (images, videos, time series, etc.). Similarity searches aim at retrieving similar data according to the intrinsic characteristics of the data. Recently, graph-based methods have emerged as a very efficient alternative for similarity retrieval, with reports indicating they have outperformed methods of other categories in several situations. This work presents two main contributions to graph-based methods for similarity searches. The first contribution is a survey on the main graph types currently employed for similarity searches and an experimental evaluation of the most representative graphs in a common platform for exact and approximate search algorithms. The second contribution is a new graph-based method called HGraph, which is a connected-partition approach to build a proximity graph and answer similarity searches. Both of our contributions and results were published and received awards in international conferences.*

1. Introduction

Similarity searches are based on retrieving similar data of one or more data used as reference according to an intrinsic characteristic of the data. The similarity between two complex data is measured by applying a (dis)similarity function (commonly, a distance function) to the feature vectors describing the compared data [Barioni et al. 2011]. Similarity retrieval of data is employed on a wide range of modern applications, for example, content-based image retrieval, pattern recognition, and recommender systems, to name a few [Skopal and Bustos 2011].

Recently, graph-based methods have emerged as a very efficient option to execute similarity queries in metric and non-metric spaces [Naidan et al. 2015, Malkov et al. 2014]. Graphs model the interconnectivity among data, enabling us to explore relationships and neighbors in an agile way. To model the similarity space as a graph, a common approach is to use vertices to represent complex data and edges connecting two vertices as the similarity relationship between the pair of complex data [Malkov et al. 2014, Paredes and Chávez 2005, Ocsa et al. 2007]. Pairs of vertices can be connected by edges according to special conditions. For instance, in *k-NN* graphs, each vertex has an edge connecting it to each of its k-Nearest Neighbors. Notice that in this approach, the type of graph is defined by how the vertices are connected.

*Larissa is currently a Ph.D. candidate at the Database Group of the Eindhoven University of Technology, Eindhoven - Netherlands

Some graph-based methods proposed in the literature have already demonstrated superior efficiency when compared to other types of exact and approximate similarity search methods [Paredes and Chávez 2005, Naidan et al. 2015]. However, to the best of our knowledge, until our work there were no survey articles that focused on comparing the main graph-based methods of the literature. Moreover, some properties of the types of graphs used for similarity search seem to be very effective in enhancing the precision of approximate queries, such as *long-range edges*. Therefore, another open problem is to address whether and how these properties can be successfully integrated into other proximity graphs to improve them. In this context, this master's thesis presented two main contributions:

1. A survey of graph-based methods for similarity searching – This contribution presents a survey of the main proximity graph types, their properties and limitations to support similarity searching, the main search algorithms for exact and approximate retrieval, and a comprehensive performance analysis of the graph-based methods proposed in the literature.
2. The *HGraph* method – *HGraph* is a connected-partition approach to build different types of graphs used for similarity searches. The approach's objective is to accelerate the construction of the graph-based methods using a divide and conquer approach and increase the similarity search quality (query time and recall) by adding edges to selected vertices.

The survey of graph-based methods provides a quantitative view of the exact search compared to accurate setups for approximate search. These results reinforce the tradeoff between graph construction cost and search performance according to the construction and search parameters. These results can be used as a baseline for further research; thus, our results could also contribute to the development of other graph-based methods. A short version of our survey was published in a paper in the 2018 International Conference on Similarity Search and Applications (SISAP) [Shimomura et al. 2018]. This paper was one of the five papers invited for publication of an extended version in a special issue of the Information Systems journal [Shimomura et al. 2021].

A paper describing our proposed graph-based method *HGraph* was published at the 2019 International Conference on Database and Expert Systems Applications (DEXA) [Shimomura and Kaster 2019]. Experimental results showed that when comparing the *HGraph* to other graph-based methods given a recall rate $= 1$ (exact answer), the *HGraph* was able to outperform or have approximate query time compared to other graph-based methods. This paper won the DEXA's Gabriela e Roland Wagner Award, which is given to one of the distinguished papers in the conference. In the next section, we give more details about the contributions and their results.

2. Contributions and main results

In this section, we give a summary of our contributions and main results. For more details and results, see our publications [Shimomura et al. 2018, Shimomura and Kaster 2019, Shimomura et al. 2021].

Our first contribution is *a comprehensive survey on the main graph-based methods for similarity searches* [Shimomura et al. 2018, Shimomura et al. 2021]. This contribution includes the three following items.

1. *Literature review on graph-based methods.* We reviewed the literature and organized the proposed methods according to their graph type, applicable search algorithms, and construction algorithms proposed for each graph type. We created a taxonomy for the existing graph-based methods. From this literature review, we could conclude that the types of graphs used for similarity searches are a class of graphs called proximity graphs in which an edge connects a pair of vertices if and only if these vertices (complex data) follows a proximity property, for example, the k-NN graph. Thus, proposed search algorithms are mainly based on the spatial approximation introduced by [Navarro 2002]. To the best of our knowledge, before our work, there were no surveys that focused on graph-based methods or any other work which categorized the existing graph methods according to the same criteria as we did. This review is a good starting reference point for researchers who are interested in the area and need an overview of the existing methods.

2. *Search algorithms applicability in different types of graphs*. Given the search methods reviewed in our survey, we provided a discussion on the applicability of these algorithms in each graph type. The discussion is carried out by showing counter-examples on how the spatial approximation property does not hold for the main graph types and in which situations the graphs do not perform as expected. We also discuss how the exact search algorithm proposed by [Paredes and Chávez 2005], originally for k-NN graphs, can be applied to RNG (Relative Neighborhood Graphs) [Ocsa et al. 2007]. This discussion is an important contribution as it shows when the graphs do not perform as expected and how the properties of the discussed methods should be taken into consideration when proposing a new graph type, search algorithm, or even when analyzing the experimental performance of a tested method. This contribution is included in the thesis and was published in [Shimomura et al. 2021].

3. *Performance analysis of the graph-based methods*. We made an experimental comparison of the main graph-based methods and evaluated their performance according to varying construction and search parameters. To provide a fair comparison, we extended the Non-metric space lib (NMSLib) [Boytsov and Naidan 2013] to evaluate all the methods in the same setup and the same platform using real-world datasets. From this evaluation, we could observe the general behavior of each type of graph and observe significant tradeoffs according to the main construction and search parameters. Some of the results were: there is a tradeoff between construction and query time (as the number of neighbors per-vertex increases, query time and the parameter value for the search algorithm to return answers close to the exact answers decrease); when comparing graph settings for a given recall rate, we were not able to point out a winner method for every condition tested; long-range edges present in the Navigable Small World Graph (NSW [Malkov et al. 2014]) can improve the search quality (result recall rate and query time). Some of the challenges of this contribution were: (1) understanding the proposed methods of the literature and finding the similar and dissimilar points between these methods to categorize the methods; (2) implementing some of the graph-based methods and search algorithms in a common platform; and (3) running a wide number of experiments on several datasets with different combinations of construction and search parameters and analyze the results considering each parameter behavior for each method. The result analysis was a big challenge given the different parameters depending on the graph type and its construction method.

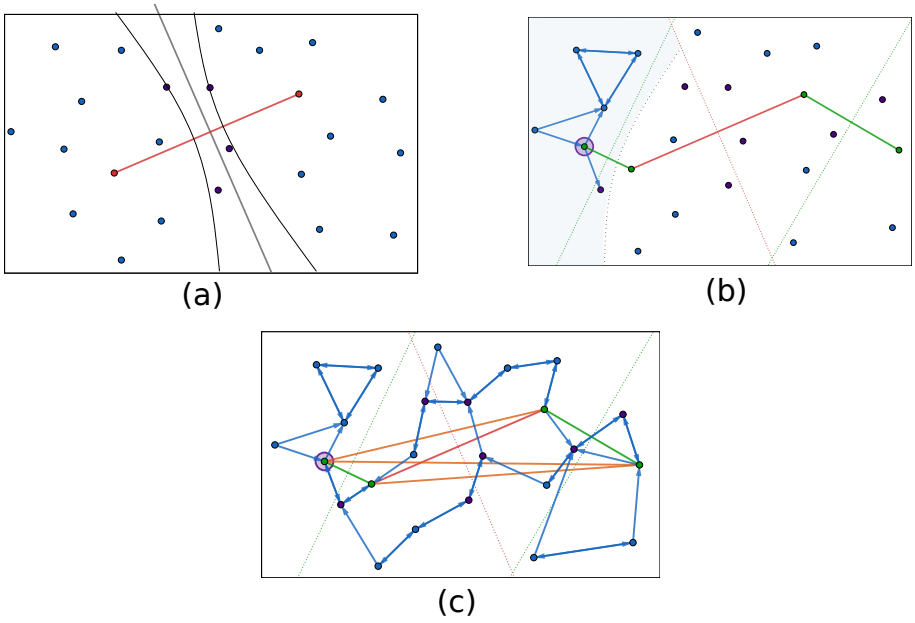

Figura 1. HGraph Construction

Our second contribution is the *HGraph* method [Shimomura and Kaster 2019]. From our experimental analysis, we conjectured that partitioning the data space could speed up building the graphs, and adding edges to connect vertices from different graph regions (inspired by the long-range edges from NSW) could increase the recall for different types of graphs. From this hypothesis, we proposed the *HGraph* method, a connected-partition approach to build different types of proximity graphs for similarity searching.

HGraph speeds up the graph construction time and increases the approximate search result quality by adding long-range edges connecting different graph regions via the pivots used in the partitioning process. Figure 1 shows the general process of building *HGraph*. The *HGraph*'s foundation is twofold. (1) Divide and conquer strategy to build different types of graph. The method divides the dataset into smaller overlapping subsets according to elements selected as pivots, speeding up the graph construction and allowing parallel execution as presented in Figure 1(a). This process adjusts the recursive partitioning to yield a "balanced" amount of elements per partition until reaching a limit. Then, independent calls build a subgraph following the desired graph type for each partition (Figure 1(b)). (2) Interconnect subgraphs from adjacent partitions using an overlapping strategy applicable to general spaces and subgraphs from apart partitions through long-range edges connecting pivots.

When designing the *HGraph* method, we faced several challenges. Our first challenge was how to select vertices to act as pivots and how to divide the elements. The pivots are important because they are responsible for the dataset division and work as representatives of each graph region connected using long-range edges. To select the pivots, we applied three different methods: random selection, the Hull of Foci(HF) [Traina et al. 2007] method, and the clustering algorithm k-Medoids in which the pivots are the final medoids. Given the selected pivots, we used the generalized hyperplane partition approach [Uhlmann 1991] to divide the dataset into subsets. Our second challenge was how to

keep the partitioned subsets connected since disconnected regions could affect the query answer. To solve this challenge, we introduced a parameter o (overlap rate), which defines the number of elements that are duplicated in adjacent dataset partition. We selected the overlap elements according to the element distance to the "hyperplane" used in the dataset partition, which applies to general spaces. Observe that in Figure 1(a), the elements $\{$C,N,G,T$\}$ are the closest to the "hyperplane", this means that C and N will be duplicated in the left side partition, and G and T will be duplicated in the right side partition.

We stop the division process when the cardinality (number of elements) of a partition becomes smaller than a user-defined parameter m, triggering an independent construction of the parameter-defined graph for the partition elements, as showed in Figure 1(b). After that, the duplicated vertices are merged, connecting adjacent partitions, and the graph is enhanced by connecting pivots using long-range edges (observe the orange and green edges connecting the pivots in Figure 1(c)).

We analyzed the behavior of the *HGraph* main parameters and suggested default parameters according to experimental evaluation. From these experiments, we concluded: (1) long-range edges increased the answer quality compared to the "base" graph type; (2) *HGraph* can build a graph that is a suitable approximation of the actual graph even with a small overlap rate of 0.1; (3) the HF pivot selection algorithm generated the best graphs while the costly k-Medoids algorithm did not show advantages over the random pivot selection. We also evaluated the performance of using *HGraph* to build the k-NNG (*HGraph-k-NNG*) and the NSW (*HGraph-NSW*) regarding both construction and search performance compared to the base methods and the tree-based method Spatial Approximation Tree (SAT) [Navarro 2002]. From our experiments, we concluded that the *HGraph-k-NNG* was able to improve the k-NNG method in construction time, query time, and recall. Thus, we showed cases in which *HGraph-k-NNG* and *HGraph-NSW* outperformed NSW and SAT when Recall$= 1$.

Given the promising experimental results and the *HGraph* construction characteristic of building graphs using independent subsets, the *HGraph* method implementation can be further extended to run the graph-based methods in parallel. This is a valuable contribution as most proposed graph-based methods rely on centralized algorithms.

3. Conclusion

This article summarizes the contributions and results from the master's thesis entitled "*Proximity Graphs for Similarity Searches: Experimental Survey and the New Connected-Partition Approach HGraph*"[Shimomura 2019]. The thesis presented an experimental survey on graph-based methods and search algorithms for exact and approximate searches on metric spaces and proposed the connected-partition method *HGraph* to build proposed graph types in the literature. The survey can be used as a baseline for researchers interested in graph-based methods. It also highlights open challenges in graph-based methods in similarity searches, such as automatic parameter configuration, addressed in a subsequent work [Oyamada et al. 2020]. Likewise, *HGraph* showed to be able to speed up the graph construction and improve searching (recall rate and query time). The method opens research directions such as running graph-based methods in parallel, improving partitioning and connectivity strategies, and supporting similarity searches in large datasets.

4. Acknowledgements

This work has been supported by CAPES and CNPq.

Referências

Barioni, M. C. N., Kaster, D. d. S., Razente, H. L., Traina, A. J., and Júnior, C. T. (2011). *Advanced Database Query Systems*. IGI Global.

Boytsov, L. and Naidan, B. (2013). Engineering efficient and effective non-metric space library. In *Similarity Search and Applications*, pages 280–293. Springer Berlin Heidelberg.

Malkov, Y., Ponomarenko, A., Logvinov, A., and Krylov, V. (2014). Approximate nearest neighbor algorithm based on navigable small world graphs. *Inf. Syst.*, 45:61–68.

Naidan, B., Boytsov, L., and Nyberg, E. (2015). Permutation search methods are efficient, yet faster search is possible. *Proc. VLDB Endow.*, 8(12):1618–1629.

Navarro, G. (2002). Searching in metric spaces by spatial approximation. *The VLDB Journal The Int'l Journal on Very Large Data Bases*, 11(1):28–46.

Ocsa, A., Bedregal, C., and Cuadros-Vargas, E. (2007). A new approach for similarity queries using neighborhood graphs. In *Brazilian Symp. on Databases*, pages 131–142.

Oyamada, R. S., Shimomura, L. C., Junior, S. B., and Kaster, D. S. (2020). Towards proximity graph auto-configuration: An approach based on meta-learning. In *Advances in Databases and Information Systems*, pages 93–107. Springer International Publishing.

Paredes, R. and Chávez, E. (2005). *Using the k-Nearest Neighbor Graph for Proximity Searching in Metric Spaces*, pages 127–138. Springer Berlin Heidelberg.

Shimomura, L. C. (2019). Proximity graphs for similarity searches: Experimental survey and the new connected-partition approach *HGraph*. Master's thesis, Universidade Estadual de Londrina, Londrina-PR, Brazil.

Shimomura, L. C. and Kaster, D. S. (2019). Hgraph: A connected-partition approach to proximity graphs for similarity search. In *Database and Expert Systems Applications*, pages 106–121. Springer International Publishing.

Shimomura, L. C., Oyamada, R. S., Vieira, M. R., and Kaster, D. S. (2021). A survey on graph-based methods for similarity searches in metric spaces. *Information Systems*, 95:101507.

Shimomura, L. C., Vieira, M. R., and Kaster, D. S. (2018). Performance analysis of graph-based methods for exact and approximate similarity search in metric spaces. In *Similarity Search and Applications*, pages 18–32. Springer International Publishing.

Skopal, T. and Bustos, B. (2011). On nonmetric similarity search problems in complex domains. *ACM Comput. Surv.*, 43(4):1–50.

Traina, Jr., C., Filho, R. F., Traina, A. J., Vieira, M. R., and Faloutsos, C. (2007). The omni-family of all-purpose access methods: A simple and effective way to make similarity search more efficient. *The VLDB Journal*, 16(4):483–505.

Uhlmann, J. K. (1991). Satisfying general proximity / similarity queries with metric trees. *Information Processing Letters*, 40(4):175 – 179.