Metrized Small World Approach for Nearest Neighbor Search

Andrey Logvinov, Alexander Ponomarenko, Vladimir Krylov, Yury Malkov
MeraLabs, Nizhny Novgorod, Russia
alogvinov@meralabs.com, aponom@meralabs.com, vkrylov@meralabs.com,
ymalkov@meralbs.com

Abstract

In different areas attempts are made to organize data into multi-linked structures which are well suited for information search, in particular the nearest neighbor search where the result data items are metrically close to a given data item. These structures often take the form of trees (M-Tree, cover tree, KD-tree, GNAT) or networks (M-Chord, VoroNet, RayNet) built over a set of data items.

In this paper we give the regular approach to the construction of links between data items which provides logarithmical time complexity of the nearest neighbor search in the structure. According to this approach, data items are organized into an undirected graph with Small World properties, which ensure the existence of a short path between any two data items regardless of the graph size.

We propose different construction and search algorithms depending on the properties of the metric which determines the proximity of data items. The types of metric we consider are abstract metric and ordered metric. Further we extend the ordered metric approach to compound data items in the form of attribute-value pair sets to enable inclusion search by an arbitrary subset of attribute-value pairs.

Finally we provide simulation results for the structure with compound data items.

1. Introduction

The nearest neighbor search problem is defined as follows: given a set S of n points in some metric space (X, d), build a data structure on S so that for a given query point $p \in X$ one can efficiently find a point $q \in S$ which minimizes $d(p, q)$.

Different approaches exist for building such a structure. The works [4, 5, 11] suggest hierarchical tree structures constructed using information about metric proximity of the elements. One notable shortcoming of this approach is the presence of the mandatory root node in tree-like structures which makes building totally distributed implementations problematic.

There are also ways to build a distributed structure over the set S. The works [12] suggest distributed hash table as the data structure using the pivot-based metric space indexing approach.

The work [6] discusses the VoroNet distributed data structure. The elements of S are two-dimensional Euclidian space points. Each point from S is linked to all of its neighbor points on Voronoi diagram (Delaunay graph) plus additional distant points to give the structure Small World properties. Greedy search algorithm is used.

The following work [7] by the same authors considers the structure where the elements are points in a n-dimensional Euclidean space. The main difference from the previous work is that every point is connected with only a subset of the Voronoi neighbors to avoid exponential dependence of complexity on the number of dimensions. But this link set reduction leads to inexact search results, i.e. the result point is not always the nearest neighbor of the query point although number of such result can be made insignificant. Another drawback of this approach is that it can only be applied to the points of Euclidian space with a fixed number of dimensions.

In this paper we propose a regular approach to the construction of links between data elements in the form of an undirected graph with Small World properties [9, 10] to provide logarithmical complexity of the nearest neighbor search. We called the resulting structure Metrized Small World [1] (MSW).

We propose different construction and search algorithms depending on the properties of the metric which determines the proximity of data items.

The rest of the paper is structured as follows. Section 2 describes the construction of MSW structure based on abstract semi-metric. Section 3 describes MSW structure construction algorithms for ordered metrics. In the section 4 we extend the ordered metric approach to compound data items in the form of attribute-value pair sets to enable inclusion search by an arbitrary subset of attribute-value pairs. Finally we

provide simulation results for the structure with compound data items in the section 5.

2. Metrized Small World data structure

Metrized Small World data structure on the set of data items S is expressed by the graph $G(V, E)$. Each vertex $v \in V$ corresponds to a single element of the set *S.* Each edge $e \in E$ is associated with a link between two data items from the set *S*. Assume that $d(v, p)$ equivalent to $d(s, p)$ where s is the data item which corresponds to the vertex v. Then the search of the nearest neighbor of the query point $p \in X$ comes to finding the vertex $v \in V$ with the minimal distance to p.

In the work [1] we gave the construction and search algorithms for that structure. In the paper [2] we also suggested a distributed storage architecture based on the proposed structure. Here we re-cite those algorithm according to the notation assumed for this paper.

We provide the algorithm which adds v_{new} vertex to the graph $G(V, E)$, where V is the set of previously added vertices. Thus the parameters of the algorithm are V — the set of previously added vertices, v_{new} the vertex being added, $v_{start} \in V$ – an arbitrarily selected vertex from V (the starting point of the search) and two integer numbers m and n.

Algorithm: $add\_metric(V, v_{new}, v_{start}, n, m)$

1. Arbitrarily select an element $v_{curr} \in V$
2. Let *VisitedList* be the set of visited elements.
3. Let *CandidateList* be the set of candidate elements for link establishment sorted by value of semi-metric to v_{new} in ascending order.
4. Assume that *CandidateLists* initially contains only v_{start}.
5. For i =1 to n do
   - 5.1. Sort *CandidateList* by value of semi-metric to v_{new} in ascending order.
   - 5.2. Select the first element *p* from *CandidateList* not contained in *VisitedList.* If no such element exists then break.
   - 5.3. Add *p* to *VisitedList.*
   - 5.4. Add the set of *p* neighbor elements to *CandidateList.*
6. Mutually connect the element with *m* arbitrary elements from *VisitedList.*

We shown that the structure constructed using this algorithm provides the necessary condition for the existence of effective search algorithm, because the Small World properties of the graph $G(V, E)$ ensure the existence of a short path between any two vertices. But this structure requires search algorithms which are more complex than the greedy algorithm due to the existence of metric local minimums.

An advantage of this approach is that the proximity measure M can be any function which is a general metric or even semi-metric defined over the set S.

3. Single-attribute Distributed Metrized Small World Data Structure

In the paper [3] we gave the algorithm for constructing the similar structure for a narrower class of metrics, i.e. for the metrics for which the order between data items is defined. If any data item will be linked with its direct predecessor and successor with regard to the metric, there will be no local minimums. The condition of the data item being linked to its direct successor and predecessor ensures the existence of the Delaunay graph which in its turn provides for correctness of the greedy search algorithm which attempts to minimize the distance from the query on each step.

Algorithm:

$add\_ordered\_metric\ (V, v_{new}, v_{start}, m)$

1. Let $v_{cur} = v_{start}$.
2. For each neighbor v_i of v_{cur} calculate $d_i = d(v_i, v_{new})$.
3. If $\min(v_i) < d(v_{cur}, v_{new})$ let $v_{cur} = v_i$ for which $d_i = \min(d_i)$ and go to step 2.
4. If $v_{cur} < v_{new}$ let $v_{pre} = v_{cur}$ and let v_{succ} be the direct successor of v_{new} chosen from the neighbors of v_{cur}.
5. If If $v_{cur} > v_{new}$ let let $v_{succ} = v_{cur}$ and let v_{pre} be the direct predecessor of v_{new} chosen from the neighbors of v_{cur}.
6. Mutually connect v_{new} with v_{pre} and v_{succ} if they exist.
7. Repeat m times:
   - 7.1. If v_{pre} exists, let v'_{pre} be the direct predecessor of v_{pre} chosen from its neighbors.
   - 7.2. If v_{succ} exists, let v'_{succ} be the direct successor of v_{succ} chosen from its neighbors.
   - 7.3. If none of v'_{pre} and v'_{succ} exist then break.
   - 7.4. If only v'_{pre} exists or $d(v'_{pre}, v_{new}) < d(v'_{succ}, v_{new})$ mutually connect v_{new} and v'_{pre} and let $v_{pre} = v'_{pre}$.
   - 7.5. If only v'_{succ} exists or $d(v'_{succ}, v_{new}) < d(v'_{pre}, v_{new})$ mutually connect v_{new} and v'_{succ} and let $v_{succ} = v'_{succ}$.

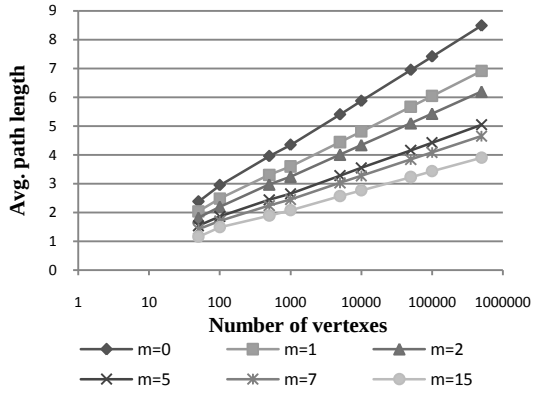

Figure 1. Average shortest path length between two vertexes

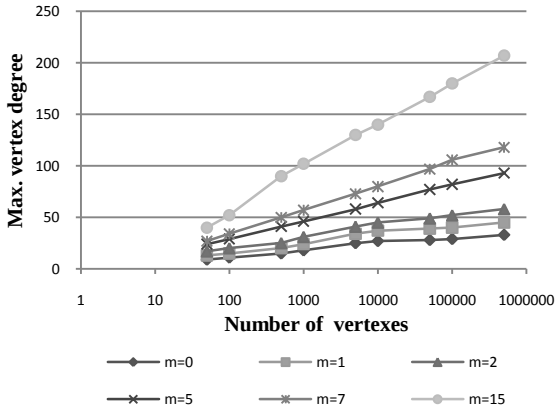

Figure 2. Maximum vertex degree

The nearest neighbor search is performed by following links from one element to another in the direction of the minimal metric.

The Small World properties of the graph ensure the logarithmical search complexity for a random data set. The absence of the root element and the construction of the structure on the data item level provides for creating a completely distributed implementation of the structure. As can be seen on Fig. 1 and 2, both average shortest path length and maximum vertex degree scale logarithmically with the number of vertexes. Therefore the structure is suitable for storing very large amounts of data.

The nearest neighbor search is reduced to finding the minimum of the metric from the query to a data item. If the distance between the query and the found data item is lower than the query radius than the fond data item is the result, otherwise there is no result. If we must find all data items inside the query radius, we perform a sequential search in both directions from the first found data item.

The proposed data addition algorithm is incremental, i.e. the addition of a new data item affects only a small number of existing data items.

4. Multi-attribute Distributed Metrized Small World Data Structure

In the two previous sections we considered the elements as atomic entities relative to the metric. Now we want to extend our approach to composite data items. We will consider the composite objects which are represented by an unordered set of atomic objects for all of which one common ordered metric is defined.

Then we define the search problem as the search of at least one of all of the composite objects which include the given set of atomic objects. This data model is often used for describing application domain entities with a set of tags or keywords, e.g. images, hyperlinks, musical tracks, blog posts etc. This model can also represent objects consisting of non-fixed set of attribute-value pairs.

Therefore for convenience we will consider arbitrary strings (or tags) as atomic objects. Hence the composite objects will be represented as unordered sets of tags.

Our main idea was to construct the graph $G(V,E)$ in a way that objects with any matching subset of atomic objects T_{fix} would constitute the sub graph (layer) $L_{T_{fix}} \in G(V,E)$ consisting of a single connected component which in its turn would form the MSW structure described in the previous section. Then the search for an element containing the given set of tags $T_q = \{t_1, t_2, \dots, t_m\}$ would be performed by first finding object from sub graph (layer) $L_{T_{t1}}$ consisting of objects containing the tag t_1. After that, inside this subgraph-layer $L_{T_{t_1}}$ another element from the subgraph-layer $L_{T_{t_1,t_2}} \subset L_{T_{t_1}}$ is recursively searched for. The subgraph-layer $L_{T_{t_1,t_2}}$ consists of objects containing both tags t1 and t2. The process continues until an object form the subgraph-layer $L_{T_{t_1,t_2,\dots,t_m}}$ is found which consists of objects containing all the given tags $\{t_1, t_2, \dots, t_m\}$.

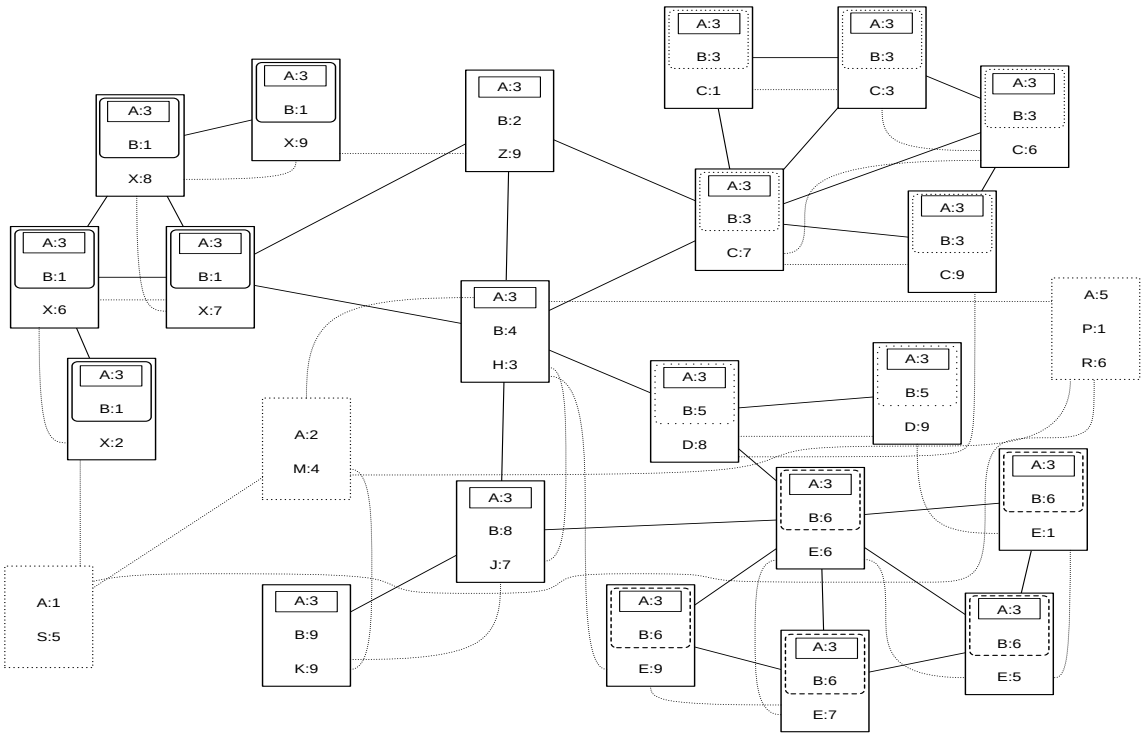

Figure 3. Example Multi-attribute Distributed Metrized Small World Data Structure. The dashed lines represent the edges in the $L_{\emptyset}$ layer. Solid straight lines show the links between objects having a common subset of tags.

For demonstration purposes we provide the example of the network of objects almost all of which contain three tags. Dashed curved lines show the links between objects which contain tags which are neighbors in lexicographical order. Solid straight lines show the links between objects having a common subset of tags.

Further we give a more formal description of the construction and search algorithms for this structure Let $T = \{t\}$ be the set of all possible tags which are distinct string values.

For each data element a let there be the unordered set $T_a \subset T$ of tags associated with the object. Given a query set $T_q \subset T, T_q \neq \emptyset$ we must find the set A_r of resulting data elements such that $\forall a \in A_r \ T_q \subset T_a$, i.e. all data elements which have all of the tags specified in the query.

Let the set $MSW_x = \left\{\left(t_{a_i}^k, t_{a_j}^l\right)\right\};\ a_i, a_j \in X, t_{a_i,a_j}^{k,l} \in T_{a_i,a_j}$ be the MSW structure built over a set of elements X. Every element of MSW_X represents a link between pair of tags in data elements (it can be the same element). If there is no element corresponding to a pair tags, there is no link between them. Two identical tags on the different items cannot have links simultaneously in one MSW_X. We consider a tag t being a member of the MSW_X if $\exists t_j, (t, t_j) \in MSW_X$. We can use our algorithm described in the section 3 of this paper to search for given tag in MSW.

Let $L_{T_{fix}} = \left(T_{fix}, MSW_X\right)$; be the MSW layer built over a set of tags T_{fix}. For every tag that is a member of $L_{T_{fix}}$, $t \in T_a, T_{fix} \subset T_a$.

Let the $a = search\_single(L_{T_{fix}}, t_{start}, t);\ a \in X$ be the operation of searching for a single element, member of $L_{T_{fix}}$ for which $t \in T_a$. The tag t_{start} (member of $L_{T_{fix}}$) is the entry point of the algorithm described in the second section of this paper.

Let $add\_partial(L_{T_{fix}}, t_{start}, a, t_a)$ be the operation of addition of the tag t_a of the element a to the MSW layer $L_{T_{fix}}$. The tag t_{start} is used as the entry point. The time complexity of the $add\_partial$ operation is logarithmic to the number of tags in $L_{T_{fix}}$. We consider an element a being a member of the MSW layer $L_{T_{fix}}$ if it has been partially added to $L_{T_{fix}}$ at least once.

Let $add\_complete(L_{T_{fix}}, t_p, a)$ be the operation of complete addition of the element a to the MSW layer $L_{T_{fix}}$. The $add\_complete$ operation is performed using the following algorithm:

Algorithm: $add\_complete(L_{T_{fix}}, t_p, a)$

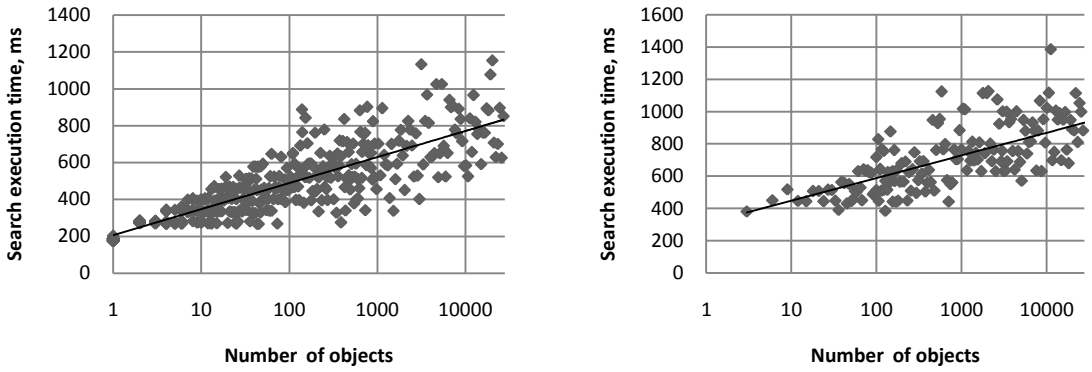

Figure 4. Experimental results. Left: two common tags. Right: three common tags.

1. Let $T_{free} = T_a \setminus T_{fix}$ be the set of all tags associated with the element a but not contained in T_{fix}.
2. For each $t_a \in T_{free}$ do $add\_partial(L_{T_{fix}}, t_p, a, t_a)$.

Let $S = \left\{L_{T_{fix}}\right\}$ be the set of all MSW layers (the structure being described). An arbitrary member t_{global_start} of the MSW layer $L_\emptyset$ can serve as a global entry point for addition process.

Let $add\_recursive(S, L_{T_{fix}}, t_{start}, a)$ be the operation of addition of the element a to the structure S.

The $add\_recursive$ operation is performed using the following algorithm, assuming that the initial values are $T_{fix} = \emptyset$ and $t_{start} = t_{global\_start}$.

Algorithm: $add\_recursive(S, L_{T_{fix}}, t_{start}, a)$

For each $t \in T_a \setminus T_{fix}$

1. Find $a_{next} = search\_single(L_{T_{fix}}, t_{start}, t)$
2. **If** p_{next} exist, perform $add\_recursive(S, L_{\{T_{fix},t\}}, t_{a_{next}}, a)$, where $t_{a_{next}}$ is a random tag of a_{next}, $t_{a_{next}} \notin \{T_{fix}, t\}$ **else** $add\_partial(L_{T_{fix}}, t_{start}, a, t)$

Let $A = search\_recursive(S, L_{T_{fix}}, t_{start}, T_{query})$ be the operation of searching all elements A for which $T_{query} \subset T_a$.

The $search\_recursive$ operation is performed using the following algorithm, assuming that the initial values are $T_{fix} = \emptyset$ and $t_{start} = t_{global\_start}$.

Algorithm: $search\_recursive(S, L_{T_{fix}}, t_{start}, T_{query})$

1. If $T_{query} = \emptyset$ then return all elements in layer $L_{T_{fix}}$
2. for random $t_q \in T_{query}$ find.
3. $a_{next} = search\_single(L_{T_{fix}}, t_{start}, t)$
4. Remove t_q from T_{query}
5. $search\_recursive(S, L_{\{T_{fix},t\}}, t_{a_{next}}, a)$ where $t_{a_{next}}$ is a random tag of a_{next} $t_{a_{next}} \notin \{T_{fix}, t\}$

Constructing link using the above approach is to a certain degree equivalent to indexing by all possible combinations of columns in a relational database. The main advantage of this approach is the possibility to quickly find an object or a set of objects with any given set of tags without regard to the quantity of objects with a certain subset of tags (atomary objects).
Further we give the experimental data obtained on the structure prototype to confirm the theoretical assumptions regarding the advantages of our approach.

5. Experimental data

The experiments were set up as follows.

In the first experiment a set of N objects was generated half of which contained the single common tag “X”, other half contained the single common tag

"Y" and a single object with both "X" and "Y" tags. The objects were added to the structure in random order. We measured the time of search for the object containing "X" and "Y" tags. The measurement was repeated many times for different values of N, the set of random objects was regenerated each time. See the left graph.

In the second experiment the test set contained N random objects containing equal amounts of object containing two common tags "X", "Y"; "Y", "Z"; "X", "Z" and the single object containing all three tags "X","Y","Z". See the right graph.

The results are shown on Figure 4. The graphs show that in both cases the object search time depends logarithmically on the number the objects in the structure which confirms our theoretical assumptions.

6. Conclusion and future work

We believe that the key to the building of search-oriented distributed systems is the construction of multilinked structures similar to social networks. But the metric distance between data items must be correlated to the number of links which separate them. In this paper we described the methods of construction of such structures for certain data types. The necessary and sufficient condition of correctness of the greedy search algorithm is the inclusion of Delaunay graph into the structure graph. Failure to satisfy this particular condition was the obstacle for using the greedy search algorithm with the structure described in the section II. The condition of existence of Delaunay subgraph has been satisfied in the structures described in sections III an IV. But supporting the correct Voronoi tessellation as in [6] or in section IV requires large overhead with the number of dimensions greater than two. For this reason we intend to focus our further research on finding the compromise between search accuracy and calculation overhead.

7. References

[1] V. Krylov, A. Logvinov, A. Ponomarenko, D.Ponomarev "Metrized Small World Properties Data Structure", Proc. Software Engineering and Data Engineering (SEDE 2008).

[2] V. Krylov, A. Logvinov, A. Ponomarenko, D.Ponomarev "Active Database Architecture for XML Documents", Proc. Computer applications in Industry and Engineering (CAINE 2008).

[3] V. Krylov, A. Logvinov, A. Ponomarenko, D.Ponomarev, "Single-attribute Distributed Metrized Small World Data Structure", Proc. IEEE International Conference on Intelligent Computing and Intelligent Systems 2009 (CAS)

[4] CIACCIA, P., PATELLA,M., AND ZEZULA, P. 1998. A cost model for similarity queries in metric paces. In Proc. 17th ACMSymp. on Principles of Database Systems (Seattle), 59–67.

[5] BRIN, S. 1995. Near neighbor search in large metric spaces. In Proceedings of the 21st conference on Very Large Databases (VLDB'95), 574–584.

[6] Beaumont, O. and Kermarrec, A.M. and Marchal, L. and Riviere, E., VoroNet: A scalable object network based on Voronoi tessellations, in IEEE IPDPS, 2007

[7] O. Beaumont, A.-M. Kermarrec, and E. Rivire. Peer to peer multidimensional overlays: Approximating complex structures. In OPODIS,11th International conference on principles of distributed systems, 2007.

[8] J.-D. Boissonnat and M. Yvinec. Algorithmic Geometry. Cambridge University Press, 1998.

[9] D.J. Watts "Small Worlds", Princeton, New Jersey: Princeton University Press, 1999.

[10] R. Albert and A.-L. Barabasi "Statistical mechanics of complex networks." Rev. Mod. Phys., 74(1): pp. 47-97, January 2002.

[11] A. Beygelzimer, S. Kakade, and J. Langford. "Cover trees for nearest neighbor". Proceedings of the 23rd International Conference on Machine Learning, pages 97–104, 2006

[12] D. Novak and P. Zezula. M-Chord: A scalable distributed similarity search structure. In Proceedings of First International Conference on Scalable Information Systems (INFOSCALE 2006), Hong Kong, May 30 June 1 . IEEE Computer Society, 2006.