Proximate Objects Probabilistic Searching Method

Andrey Chukhray and Olena Havrylenko(✉)

National Aerospace University “Kharkiv Aviation Institute”, Kharkiv, Ukraine
achukhray@gmail.com, o.havrylenko@khai.edu

Abstract. The method of objects probabilistic search is developed based on the necessary proximity conditions in a Euclidean space, which were previously proved for the Levenshtein’s metric. The method is based on a random selection of k pivots in Euclidean space among the original objects, projecting all source objects in a k-dimensional Euclidean space, filling special hash data structures, and fast search facilities, similar to the desired, based on proven necessary conditions for the objects proximity in Euclidean space. Experimental studies of the proposed method show the higher speed in comparison with the known method.

Keywords: Probabilistic search · Euclidean space · Necessary proximity conditions

1 Problem Statement

Recent decades in the artificial intelligence sphere there are developed and used a set of “nearest neighbor” search methods for objects arrangement and clustering [1–3]. Nevertheless, the problem of the high-performance methods development in circumstances when calculation of distances between objects still take the certain search time is still actual. The method based upon the necessary proximity conditions, which is a generalization of the conditions previously proved for Levenshtein’s metric, is developed, experimentally tested and described in [4, 5].

In common, the problem statement is the following. Assume that an edit distance δ between objects of a certain class Cl satisfies the conditions:

$$\begin{cases} \delta(X,Y) \geq 0; \\ \delta(X,X) = 0; \\ \delta(X,Y) = \delta(Y,X); \\ \delta(X,Z) \leq \delta(X,Y) + \delta(Y,Z). \end{cases} \tag{1}$$

A certain object rt of a class Cl and a set of objects $ET = (et_1, et_2, \ldots, et_n)$ of the same class are given. It is required to find all et_i of the set ET, such that the distance δ between et_i and rt is not greater than a given positive integer λ. Formally required to find $ET_s = \{et_{s1}, et_{s2}, \ldots, et_{sl}\}$, such that $\forall et_{si} \in ET_s \subseteq ET : \delta(et_{si}, rt) \leq \lambda,\ \lambda \in N,\ l \leq n$.

The proposed method consists of two steps. 1st step. k elements $o_1, o_2, \ldots, o_k,\ k \leq n$ are randomly selected from the ET set. These elements are

M. Nechyporuk et al. (Eds.): *Integrated Computer Technologies in Mechanical Engineering*,
AISC 1113, pp. 219–227, 2020.
https://doi.org/10.1007/978-3-030-37618-5_20

considered as k pivots in a k-dimensional Euclidean space E^k. After that each element et_i of the ET set is associated with E^k point coordinates which are equal to the distances to the pivots, i.e. $P(et_i)_j = \delta(et_i, o_j)$, $i = \overline{1,n}$, $j = \overline{1,k}$. 2nd step. The rt object is associated in space E^k with the point coordinates $P(rt)_j = \delta(rt, o_j)$, $j = \overline{1,k}$. Distances are calculated only between the object rt and those objects, whose corresponding points in space E^k are located closest to the point $P(rt)$.

To determine points closeness in Euclidean space it is necessary introduce the proximity conditions for a given objects X, Y and Z of class *Cl*.

Proposition 1. For a given object X, Y and Z of a class *Cl* the distance δ between them satisfies the conditions (1) and the following inequality $\forall X, Y, Z\ \ \delta(X,Y) \geq |\delta(X,Z) - \delta(Z,Y)|$ is true.

Proof. Consider two of the triangle inequality: $\delta(X,Z) \leq \delta(X,Y) + \delta(Y,Z)$; $\delta(Y,Z) \leq \delta(Y,X) + \delta(X,Z)$. From the first inequality it is followed $\delta(X,Z) - \delta(Y,Z) \leq \delta(X,Y)$, from the second - $\delta(Y,Z) - \delta(X,Z) \leq \delta(Y,X)$. By combining both expressions and using the symmetry property, we obtain the system of inequalities: $\begin{cases} \delta(X,Y) \geq \delta(X,Z) - \delta(Z,Y); \\ \delta(X,Y) \geq \delta(Z,Y) - \delta(X,Z), \end{cases}$ or, as follows, $\delta(X,Y) \geq |\delta(X,Z) - \delta(Z,Y)|$ Q.E.D.

Proposition 2. For a given objects et_i and et_j of a class *Cl*, distance δ between which satisfies conditions (1) and does not exceed a threshold λ, corresponding points $P(et_i)$ and $P(et_j)$ it space E^k are situated on a distance of no more than $\lambda\sqrt{k}$, i.e. $\forall i \forall j \neq i\ \delta\ (et_i, et_j) \leq \lambda$: $\rho(P(et_i), P(et_j)) \leq \lambda\sqrt{k}$.

Proof. From the definition of the metric E^k follows: $\rho(P(et_i), P(et_j)) = \sqrt{(P(et_i)_1 - P(et_j)_1)^2 + (P(et_i)_2 - P(et_j)_2)^2 + \ldots + (P(et_i)_k - P(et_j)_k)^2}$. According to Proposition 1: $|\delta(et_i, o_1) - \delta(et_j, o_1)| \leq \delta(et_i, et_j)\ldots, |\delta(et_i, o_k) - \delta(et_j, o_k)| \leq \delta(et_i, et_j)$. Hence, transitively: $|P(et_i)_1 - P(et_j)_1| \leq \lambda\ldots, |P(et_i)_k - P(et_j)_k| \leq \lambda$ and, therefore: $\sqrt{(P(et_i)_1 - P(et_j)_1)^2 + \ldots + (P(et_i)_k - P(et_j)_k)^2} \leq \sqrt{\lambda^2 k} = \lambda\sqrt{k}$, Q.E.D.

Proposition 3. For a given objects et_i and et_j of a class *Cl* the distance δ between which satisfies conditions (1) and does not exceed a threshold λ, point $P(et_j)$ located in the space E^k within a hypercube centered at $P(et_i)$ with the side equal to 2λ.

Proof. According to Proposition 1 the following systems of inequalities can be consequently obtained:

$$\begin{cases} |\delta(et_i, o_1) - \delta(et_j, o_1)| \leq \delta(et_i, et_j); \\ \\ |\delta(et_i, o_2) - \delta(et_j, o_2)| \leq \delta(et_i, et_j); \\ \\ \qquad \cdots \\ |\delta(et_i, o_k) - \delta(et_j, o_k)| \leq \delta(et_i, et_j), \\ \end{cases} = \begin{cases} P(et_j)_1 \geq P(et_i)_1 - \lambda; \\ P(et_j)_1 \leq P(et_i)_1 + \lambda; \\ P(et_j)_2 \geq P(et_i)_2 - \lambda; \\ P(et_j)_2 \leq P(et_i)_2 + \lambda; \\ \qquad \cdots \\ P(et_j)_k \geq P(et_i)_k - \lambda; \\ P(et_j)_k \leq P(et_i)_k + \lambda. \end{cases} \qquad (2)$$

The geometric meaning of the system of inequalities (2) is a hypercube centered at $P(et_i) = (\delta(et_i, o_1), \delta(et_i, o_2), \ldots, \delta(et_i, o_k))$ with the side length 2λ, Q.E.D.

Proposition 4. For a given objects et_i and et_j of a class Cl the distance between which δ satisfies conditions (1) and does not exceed a threshold λ, absolute value of the difference of the distances from points $P(et_i)$ and $P(et_j)$ to the origin in the space E^k does not exceed $\lambda\sqrt{k}$, i.e. $|\rho(P(et_i), 0) - \rho(P(et_j), 0)| \leq \lambda\sqrt{k}$.

Proof. According to the property of the Euclidean metric space (triangle inequality) $|\rho(P(et_i), 0) - \rho(P(et_j), 0)| \leq \rho(P(et_i), P(et_j))$. On the other hand, according to Proposition 2: $\rho(P(et_i), P(et_j)) \leq \lambda\sqrt{k}$. From the above it can be obtained: $|\rho(P(et_i), 0) - \rho(P(et_j), 0)| \leq \rho(P(et_i), P(et_j)) \leq \lambda\sqrt{k}$ and hence $|\rho(P(et_i), 0) - \rho(P(et_j), 0)| \leq \lambda\sqrt{k}$, Q.E.D.

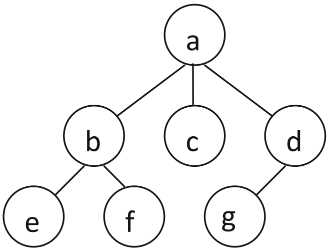

Fig. 1. An example of a tree as an object

Proposition 5. Assume that $u, w \in R$ and $u, w > 0$. Then from $[u] \leq w$ it is followed: $[u] \leq [w]$, where $[u]$, $[w]$ are the whole parts of the numbers u and w respectively.

Proof. There are two cases when $[u] \leq w$ is true: $[u] = [w]$ and $[u] < [w]$. Generalizing both of them, we can obtained condition $[u] \leq [w]$. It is also obvious that assumption $[u] > [w]$ could not be true for $[u] \leq w$, therefore what was required to prove $[u] \leq [w]$ is true.

Proposition 6. Assume that $u, v, w \in R$ and $u, v, w > 0$. Then from $|u - v| \leq w$ it is followed: $|[u] - [v]| \leq [w] + 1$, where $[u]$, $[v]$, $[w]$ - the whole parts of the numbers u, v, w respectively.

Proof. First consider the case, when $u \geq v$. Then from $u \leq w + v$ it could be obtained $[u] \leq w + v$, considering $u \geq [u]$, furthermore $[u] \leq w + [v] + 1$, considering $v < [v] + 1$. Further, according to the Proposition 5 and the fact that any positive number greater than any negative one, if $[u] - [v] \leq w + 1$, then $[u] - [v] \leq [w] + 1$. Second case $v > u$ could be considered the same way and obtained the next inequity:$[v] - [u] \leq [w] + 1$. Summarizing both cases: $|[u] - [v]| \leq [w] + 1$, Q.E.D.

Definition 1. A size of an object et_i of a class Cl is the number of its elements and marked as $\overline{et_i}$. For example, if et_i is the string “home”, then $\overline{et_i} = 4$ (line length); if et_i is the tree shown on Fig. 1, then $\overline{et_i} = 7$ (the number of vertices).

Proposition 7. If the absolute value of the difference between the sizes of objects et_i and et_j of a class Cl, distance δ between which satisfies conditions (1), is greater than λ, then the distance between these objects is also greater than λ, i.e. $\left(\left|\overline{et_i} - \overline{et_j}\right| > \lambda\right) \Rightarrow (\delta(et_i, et_j) > \lambda)$.

Proof. This proposition is obvious and follows from the fact that if the objects sizes differ on λ, then to convert an object et_i into object et_j or vice versa, it is should be completed at least λ element deletions in the best case and more – in the other cases.

After necessary objects proximity conditions have been proved, the essence of the proposed method could be described more detailed.

On the first step, after the random k pivots from a set ET selection and the points coordinates $P(et_i)$ calculation, distance by the points $P(et_i)$ to the origin in space E^k could be obtained as: $\rho(P(et_i), 0) = \sqrt{P(et_i)_1^2 + P(et_i)_2^2 + \ldots + P(et_i)_k^2}$.

In addition, we form a matrix D, which is distribution of distances in space E^k by points $P(et_i)$ to the origin. To do this, the set $\Psi = \{[\rho(P(et_1), 0)], [\rho(P(et_2), 0)], \ldots, [\rho(P(et_n), 0)]\} = \{\psi_1, \psi_2, \ldots, \psi_z\}$, $z \leq n$ is introduced, where $[\rho(P(et_i), 0)]$ means the integer part of $\rho(P(et_i), 0)$.

Each index $\psi_i \in \Psi$ deals with a set of integers $IND_i = \{ind_{i1}, ind_{i2}, \ldots, ind_{iw}\}$, which are the objects indexes with the distance from the origin equal to ψ_i. In this case, the following condition is true: $\forall q \in \{1, \ldots, w\}, ind_{iq} \in \{1, \ldots, n\}$, $\exists[\rho(P(et_{ind_{iq}}), 0)] = \psi_i$. Considered matrix D has dimension $(\max(\Psi) - \min(\Psi) + 1) \times (\max\{|IND_1|, \ldots, |IND_z|\})$. Using auxiliary set IND_i, there is assigned: $D_{\psi_i q} = ind_{iq}$, what means, that raw ψ_i of the matrix D contains the indices of the objects, for which an integer part of the distance in space E^k to the origin is equal to ψ_i.

For the target object rt it is required to find row with index $[\rho(P(rt), 0)]$ in the matrix D. After that, according to Propositions 4 and 6, it is need to review the neighbor rows of the matrix D with the indices from the set $\Psi 1 = \{[\rho(P(rt), 0)] - [\lambda\sqrt{k}] - 1, [\rho(P(rt), 0)] - [\lambda\sqrt{k}], \ldots, [\rho(P(rt), 0)] - 1, [\rho(P(rt), 0)], [\rho(P(rt), 0)] + 1, \ldots, [\rho(P(rt), 0)] + [\lambda\sqrt{k}], [\rho(P(rt), 0)] + [\lambda\sqrt{k}] + 1\} = \{\psi 1_1, \psi 1_2, \ldots, \psi 1_v\}$, and $\Psi 1 \subset \Psi$. $v \leq z$.

Then, when viewing the element rows of the matrix D with index $\psi 1_t$, i.e. $D_{\psi 1_{tq}}$, further screening “candidates” can be find out among the proximate objects: first, by checking the conditions from the Proposition 7: $\left|\overline{et_i} - \overline{et_j}\right| > \lambda$, and second, by checking the condition of the Proposition 3: does the $P(et_{D_{\psi 1_{tq}}})$ lie in a hypercube centered at the point $P(rt)$ and side of 2λ. Finally, if $P(et_{D_{\psi 1_{tq}}})$ is within the hypercube, then the distance $\delta(rt, et_{D_{\psi 1_{tq}}})$ between the objects rt and $et_{D_{\psi 1_{tq}}}$ is calculated.

2 Method Instantiation

There are considered two cases: (1) Class *Cl* objects are ordered m-ary trees, $m \in N$; (2) class *Cl* objects are strings.

1st Case. Objects of class *Cl* are ordered m-ary trees. Then the statement of the problem is as follows. A tree *rt* and a set of trees $ET = (et_1, et_2, \ldots, et_n)$, $i = \overline{1, n}$ are given. It is required to find all the trees et_i, such that the distance $\delta(rt, et_i)$ is not greater than a given positive integer λ.

In this case, one of the metrics for δ can be chosen. The metric used in [7] and the metric from [234, 235] can be considered as reasonable alternatives. Differences between these two metrics are in the set of valid tree editing operations: each metric permits rename operation, removal and insertion of tree nodes, but the first one consider the last two operations as applicable to the tree leaves only, i.e. to the vertices with no descendants, whereas the second one applies them to any tree node.

Further the essence of the insertion and deletion of tree nodes in a second metric will be review in details. As a result of the insert operation, some or all descendants of the parent node for the inserted node transform into inserted node descendants. After the delete operation all deleted node descendants transform into the descendants of its parent node. Consider concrete examples.

Example 1. Two ordered binary trees X and Y are represented on Figs. 2 and 3. The edit distance between X and Y in the Selkow metric [7] is 6.

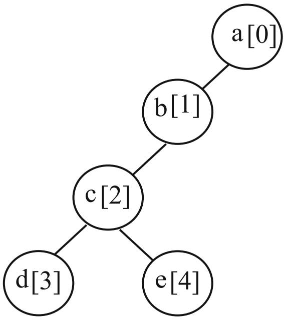

Fig. 2. Ordered binary tree X

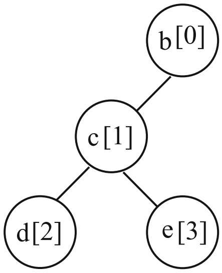

Fig. 3. Ordered binary tree Y

The minimal set of editing operations which convert X to Y includes operations:

(1) replace the node “a” with index 0 with the name «b»;
(2) replace the node “b” with index 1 with the name “c”;
(3) delete the leaf node “e” with index 4;

(4) delete the leaf node “d” with index 3;
(5) replace the name node “c” with index 2 with the name “d”;
(6) insert a right child “e” to the node “b” with index 1;

For metric used in [7, 8], the edit distance between X and Y will be equal to 1, as for converting Y to X it is necessary to delete node “a” with index 0.

Example 2. Two ordered ternary tree X1 and Y1 are represented on Figs. 4 and 5.

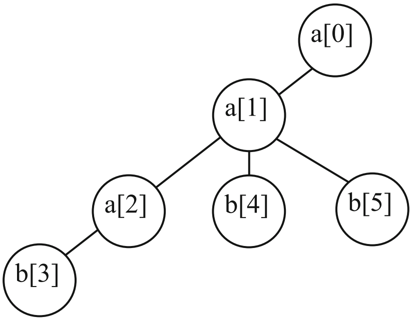

Fig. 4. Ordered ternary tree X1

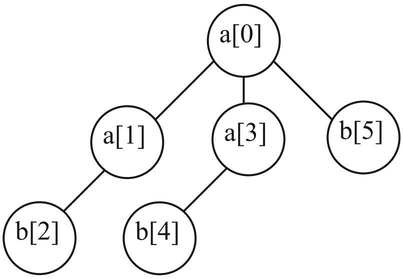

Fig. 5. Ordered ternary trees Y1

According to Selkow metric [7] edit distance between X1 and Y1 is equal to 7. The minimal set of editing operations which convert X to Y includes operations:

(1) insert a child “a” to node “a” with index 0;
(2) insert a child “b” to node “a” with index 0;
(3) insert a child “b” to the newly inserted node «a»;
(4) delete the leaf “b” with index 3;
(5) delete the leaf “b” with index 4;
(6) delete the leaf “b” with index 5;
(7) replace the name node “a” with index 2 with the name “b”.

For the second metric the distance between X1 and Y1 is equal to 3, as to convert X1 to Y1 it is necessary to perform operations:

(1) remove the node “a” with index 0;
(2) insert a child “b” to the node “b” with the index 4;
(3) replace the name node “b” with index 4 with the name “a”.

As follow from the above examples, trees are more proximate if the second metric is used. In common, the metric selection criteria should be determined depending on the specific practical problem.

2nd Case. Class *Cl* objects of are the strings. For this case using the Levenshtein distance, which is a minimal number of string edit operations for its conversion into another, is the best choice. That was proved theoretically and empirically in [4, 5], where the problem of finding similar strings is considered.

3 Experimental Research of the Method

Experimental studies of this method were carried out for the metric Zhang and Shasha [8]. To compare the results method described in [1] was chosen, where the condition formulated in Proposition 7 was also embedded to exclude a series of extra "expensive" tree edit distance calculations.

For randomly generated trees in every method time spent on the edit distance calculation was excluded from the overall time of the second search stage. The results of experimental research of the lists of 10, 100 and 1000 trees are shown on Figs. 6, 7 and 8.

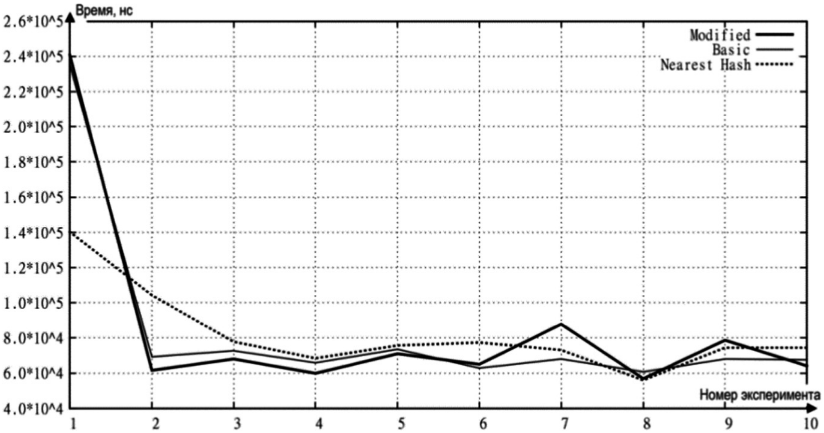

Fig. 6. The experimental results for the list of 10 trees

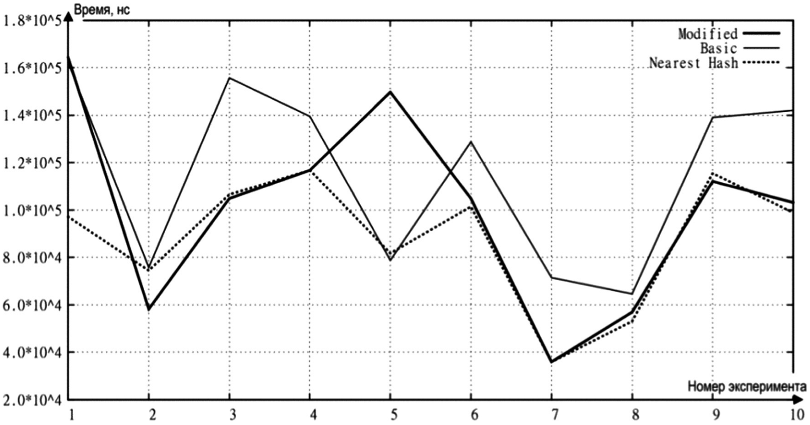

Fig. 7. The experimental results for the list of 100 trees

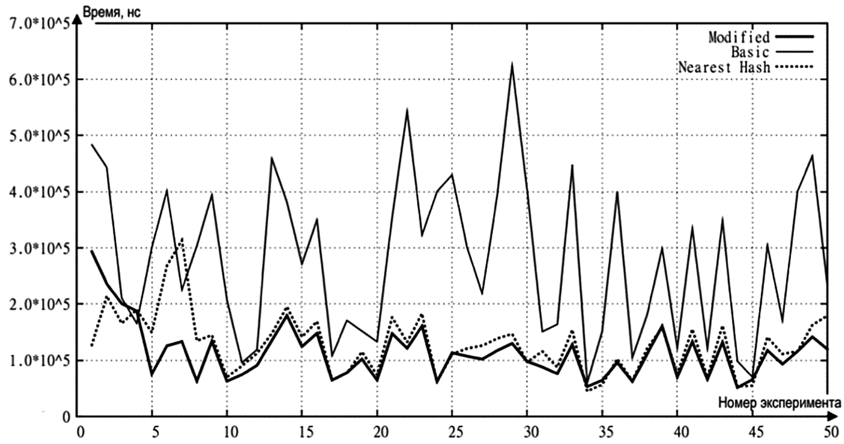

Fig. 8. The experimental results for the list of 1,000 trees

In the experiments shown on Figs. 6 and 7, the first 10 trees from the original list were target, and in experiment on Fig. 8 search target were first 50 trees, that are noted on the horizontal axis. As following from the figures, the proposed search method (Nearest Hash) is better than known ones (Basic and Modified) by performance 1.71 and 1.67 times – in the first experiment, 1.66 and 1.69 times – in the second experiment and 2.30 and 3.80 times – in the third experiment. Thus theoretical performance increase due to exclude extra objects on the first step and hash matrix constructing on the second is proved empirically.

4 Summary

Concluded, the improved method of proximity objects probabilistic search is represented with the introduction of the necessary theoretically proved the proximity objects conditions, which allow improving search performance. Proposed method can be used for the fast and precise search in Intelligent Search Systems, Big Data technology as well as in Intelligent Tutoring Systems to solve the problems of knowledge clustering and trainee answers analysis for effective pedagogical feedback.

Further research will focus on new experimental research obtained by the method: comparison with other known methods, the choice of the pivots number, the rational selection of specific pivots, as well as various applications of the method in practical tasks.

References

1. Bustos, B., Navarro, G., Chavez, E.: Pivot selection techniques for proximity searching in metric spaces. Pattern Recogn. Lett. **24**(14), 2357–2366 (2003)
2. Batko, M., Falchi, F., Lucchese, C., et al.: Building a web-scale image similarity search system. Multimed. Tools Appl. **3**(47), 599–629 (2010)
3. Zezula, P., Amato, G., Dohnal, V., Batko, M.: Similarity Search: The Metric Space Approach. Springer, New York (2006)
4. Chukhray, A.G.: Quick search method "similar" relational tuples relations. Radioelectron. Comput. Syst. **2**(2), 64–69 (2003)
5. Kulik, A., Chukhray, A., Zavgorodniy, A.: Similar strings detecting methods. In: Proceedings of the East-West Fuzzy Colloquium, pp. 38–47. IPM, Zittau (2005)
6. Selkow, S.M.: The tree-to-tree editing problem. Inf. Process. Lett. **6**(6), 184–186 (1977)
7. Tai, K.C.: The tree-to-tree correction problem. J. ACM **26**(3), 422–433 (1979)
8. Zhang, K., Shasha, D.: Simple fast algorithms for the editing distance between trees and related problems. Soc. Ind. Appl. Math. J. Comput. **18**(6), 1245–1262 (1989)