Optik 127 (2016) 10838–10843

Contents lists available at ScienceDirect

Optik

journal homepage: www.elsevier.de/ijleo

Original research article

An improved location difference of multiple distances based nearest neighbors searching algorithm

Liu Yang[a,*], Limei Dong[b], Xiaoru Bi[a]

[a] Chongqing Nanfang Translators College of SISU, Chongqing 401120, China
[b] Upper Changjiang River Bureau of Hydrological and Water Resources Survey, Chongqing 400014, China

ARTICLE INFO

Article history:
Received 12 April 2016
Received in revised form 2 August 2016
Accepted 27 August 2016

Keywords:
Location difference of multiple distances
k-Nearest neighbors
Tree structure

ABSTRACT

The location difference of multiple distances based nearest neighbors search algorithm (LDMDBA) has a good performance in efficiency compared with other kNN algorithm. The major advantage of it is its precision is litter lower than the full search algorithm (FSA) algorithm. In this paper, we proposed an improved LDMDBA algorithm (ILDMDBA) by increasing the number of the reference points from log(d) to d, where the d is the dimensionality of data set. By this way, the prediction of ILDMDBA is improved. Our analysis results show that the time complexity of the proposed algorithm is not increased. The effectiveness and efficiency of the proposed algorithm are demonstrated in experiments involving public and artificial datasets.

© 2016 Elsevier GmbH. All rights reserved.

1. Introduction

kNN algorithms are used to find k nearest neighbors of data points in a dataset. These algorithms are used in many fields including feature selection [1], pattern recognition [2,3], clustering [4], classification noise detection [5], and classification [6–10]. The basic method of finding k nearest neighbors of a point is to compute all Euclidean distances from the query point to all other data points. This method is known as the full search algorithm (FSA). The FSA has a complexity of O(n^2) so that it is very time consuming. To reduce the computation complexity, two classes of algorithms [11–27] were proposed.

The first class of algorithms creates a search tree to store data points. In these algorithms, the search strategy is bounded by branches of the search tree. Fukunaga and Narendra [11] used the hierarchical clustering technique to decompose data points and represented results using Ball tree, which is highly influenced by clustering algorithms [12]. To decrease the influence of clustering algorithms, fiveBall tree construction methods were introduced by Omohundro and Friedman [13,14]. A refined version of k-d tree method was introduced by Sproull [15]. Kim and Park [16] used the ordered partition method to create a multiple branch tree. Mico et al. [17] used a pre-stored distance table to eliminate more impossible nodes. McNames [18] proposed a method based on a principal axis search tree (PAT). Wang and Gan [19] combined projected clusters and the PAT algorithm to reduce the computation time. Chen et al. [20] used winner update search method and a lower-bound tree (LB tree) to speed up the algorithm. The performance of those algorithms deteriorates with the increase in dimensions, which is shown in [21] and our experiments. The reason is that higher dimensions lead to higher complexity of tree structures. Thus, as our experiments show, the performance of various kNN methods using various tree structures reduces significantly. In

* Corresponding author.
E-mail address: 23989060@qq.com (L. Yang).

http://dx.doi.org/10.1016/j.ijleo.2016.08.091
0030-4026/© 2016 Elsevier GmbH. All rights reserved.

addition, as the complexity of the tree structure corresponding to various datasets is different, the stability of such methods is poor as well.

The other classes of algorithms uses different method without create a tree structure. Bei and Gray [22] introduced the method of partial distortion to reduce the time of distance calculations. Ra and Kim [23] utilized the difference between mean values of the query point and other data points to eliminate impossible data points. Tai et al. [24] eliminated impossible data points using the projection values of data points. Nene and Nayar used a projection value to limit the distance from a query point [25]. Lu et al. [26] used the norm, mean value and variance to eliminate impossible data points. Lai et al. [27] utilized triangle inequality and projection values to accelerate the algorithm, which is now referred to as FkNNUPTI. These methods can speed up the process of finding nearest neighbors to some extent, but their time complexities have not been reduced any more so that they are still not enough efficient for different datasets. Xia et al. [28–30] research the affection of dimensionality in nearest neighbors searching algorithms and proposed location differences and location difference based algorithm (LDMDBA).

The LDMDBA has a time complexity of O(log*dn*log*n*) that is far less than FSA and most of other algorithms. The algorithm does not rely on any tree structure so that it can run efficiently on datasets of high dimensionality and has very good stability in various datasets. Furthermore, the algorithm has a time complexity of O(log*d*log*n*) for predicting a data point outside datasets with different dimensionality. However, a small loss in prediction accuracy of the LDMDBA compared with FSA still exists on some datasets [28]. In this paper, by increasing the number of reference points, we further improve the prediction precision of LDMDBA. At the same time, the time complexity is not increased.

2. Improved location difference of multiple distances based nearest neighbors searching algorithm

2.1. Location difference of multiple distances based factor (LDMDBF)

LDMDBF is a factor computed by a queried point to a reference point and used to measure the location difference instead of Euclidean distance between them. By avoiding the direct calculations between different points, the time complexity of location difference of multiple distances based algorithm (LDMABA) is decreased to O(nlogn). The definition of LDMDBF is described as the following:

Definition 1. LDMDBF.

Given a database D, a point A $\in D$, and the norm denoted as $||.||:R_d \rightarrow R$, the distance from O_1 to A is denoted by dis (O_1A). The neighbors of point A found using the i-th reference point are denoted by neighbors$_i$ (A), and label (D') represents labels of all the points in D'. The label of A is determined by the sum of label (neighbors$_i$ (A)). Thus, the Location Difference of Multiple Distances Based Factor denotes the label of A that is computed using its neighbors found by the proposed method. LDMDBF (A) is equal to the sign of the sum of $LDMDBF_i$(A):

$$LDMDBF(A) = sign(\sum_{i=1}^{\log_2 d} LDMDBF_i(A)) = sign(\sum_{i=1}^{\log_2 d} \sum label(neighbors_i(A))) \quad (1)$$

where $LDMDBF_i(A) = \sum \text{label}(neighbors_i(A))$.

The distance can be denoted as:

$$Dis_i(A) = ||A - O_i||$$

Function sign is expressed as the following:

$$\text{sign}(x) = \begin{cases} 1, x >= 0 \\ -1, x < 0 \end{cases}$$

2.2. Improved location difference of multiple distances based nearest neighbors searching algorithm

In our proposed algorithm, the number of reference points is increased to be d. Refer to the description in [28], considering the mth point A as an example and to compute neighbors$_i$ (A), all data points are, first, sorted by the values of the LDMDBF$_i$ and a sorted sequence is obtained. The data points near A in the sorted sequence are approximate neighbors of A. In other words, the true *k-nearest* neighbors of point A are mostly located in a subsequence with the center point A in the sequence. The nearest neighbors will be more exact with a larger subsequence. The length of the subsequence varies with the number of neighbors, and can be denoted as $2k^*\varepsilon$, where k is the number of neighbors in the neighbor-searching algorithm andε is a positive value. All exact Euclidean distances between the data points in the subsequence are computed. The number of values of the distances in all subsequences corresponding to all reference points is equal to be k^*d. Those points corresponding to the k smallest distances in the distances can be considered as the *k-nearest* neighbors of A.

On the basis of the design of the LDMDBA, the Improved LDMDBA algorithm is described as follows:

Algorithm 1. Improved LDMDBA (ILDMDBA).

Download English Version:

https://daneshyari.com/en/article/5026440

Download Persian Version:

https://daneshyari.com/article/5026440

Daneshyari.com