Pattern Recognition Letters 80 (2016) 30–36

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Natural neighbor: A self-adaptive neighborhood method without parameter K☆

Qingsheng Zhu*, Ji Feng, Jinlong Huang

Chongqing Key Lab. of Software Theory and Technology, College of Computer Science, Chongqing University, Chongqing 400044, China

ARTICLE INFO

Article history:
Received 27 November 2015
Available online 25 May 2016

Keywords:
Nearest neighbor
Natural neighbor method
Classification
Outlier detection

ABSTRACT

K-nearest neighbor (KNN) and reverse k-nearest neighbor (RkNN) are two bases of many well-established and high-performance pattern-recognition techniques, but both of them are vulnerable to their parameter choice. Essentially, the challenge is to detect the neighborhood of various data sets, while utterly ignorant of the data characteristic. In this paper, a novel concept in terms of nearest neighbor is proposed and named natural neighbor (NaN). In contrast to KNN and RkNN, it is a scale-free neighbor, and it can reflect a better data characteristics. This article discusses the theoretical model and applications of natural neighbor in a different field, and we demonstrate the improvement of the proposed neighborhood on both synthetic and real-world data sets.

1. Introduction

Over the last decade, nearest neighbor method has received considerable attention from the community of data mining and pattern recognition [1–3]. Traditionally, the definition of neighborhood plays an important role, a reasonable definition of neighborhood can effectually improve the performance. In spite of their simplicity, the k-nearest neighbors (KNN) and reverse k-nearest neighbor (RkNN) are demonstrated themselves to be the most useful and effective algorithms. And then a fundamental task that arises in KNN and RkNN is to determinate the optimum value of the parameter *k*.

The problem of choosing the optimum value of *k* is one of the best studied problem in the area of nearest neighbor method since its birth. The best choice of *k* depends upon the data. Generally, larger values of *k* reduce the effect of noises on the classification, but make boundaries between classes less distinct. If the neighborhood is too large with respect to folds in manifold on which the data points lie, large values of *k* may cause the short-circuit errors. Alternatively, small values of *k* reduce the correlation of neighborhood, or separate the data points from the same class.

In fact, determination of parameter *k* is dependent on knowledge of researchers experience and lots of experiments. In order to solve this problem, a new term called Natural Neighbor (NaN) is presented in this paper. NaN method is inspired by the friendship of human society and could be regarded as belonging to the category of scale free nearest neighbor method. The proposed method makes three key contributions to the current state:

1. Natural neighbor method can create an applicable neighborhood graph based on the local characteristics of various data sets. This neighborhood graph can identify the basic clusters in the data set, especially manifold clusters and noises.
2. This method can provide a numeric result named Natural Neighbor Eigenvalue(NaNE) to replace the parameter *k* in traditional KNN method, and the number of NaNE is dynamically chosen for different data sets.
3. The natural neighbor number of each point is flexible, and this value is a dynamic number ranging from 0 to NaNE. The center point of the cluster has more neighbors, and the neighbor number of noise is equal to 0.

2. Related work

2.1. K-nearest neighbor (KNN) method

The concept of k-nearest neighbor (KNN) is a foundation scientific issue in various fields of application study. Its colorful history begins in 1951 with the pioneering work of Stevens [4], who point out that one point and its nearest neighbor can be considered as a subset, and gave an efficient algorithm for the general version of the problem.

Definition 1 (Nearest Neighbor Search)**.** Give a set *X* of points and a query point *q*, the Nearest Neighbor Search problem is to find a

☆ This paper has been recommended for acceptance by Prof. F. Tortorella.

* Corresponding author. Tel.: +86 023 65105660; fax: +86 023 65104570.
E-mail address: qszhu@cqu.edu.cn (Q. Zhu).

http://dx.doi.org/10.1016/j.patrec.2016.05.007
0167-8655/

subset $NN_s(q)$ of X defined as follows

$$NN_X(q) = \{r \in X \mid \forall p \in X : D(q, r) \leq D(q, p)\} \quad (1)$$

Nowadays the improved algorithms based on the KNN method are widely used in a lot of research fields [5–7]. They increasing the KNN performance is obtained by the estimation of optimal k parameter [8,9] or the forms of distance metrics [10,11].

2.2. Reverse k-nearest neighbor (RkNN) method

One type of neighborhood that received attention is the concept of reverse k-nearest neighbor, and RkNN method appear in many practical situations such as decision support and resource management.

Definition 2 (Reverse Nearest Neighbor Search)**.** Give a set X of points and a query point q, the Reverse Nearest Neighbor Search problem is to find a subset $RNN_s(q)$ of X defined as follows

$$RNN_X(q) = \{r \in S \mid \forall p \in X : D(q, r) \leq D(r, p)\} \quad (2)$$

Korn and Muthukrishnan [12] firstly do the fundamental research in 2000, and then the concept of reverse k-nearest neighbors (RkNN) search is purposed [13]. Recently, research achievements aim to find the reverse k-nearest neighbors quickly and exactly [14].

Additionally, analogous to KNN and RkNN, the mutual k-nearest neighbor (MkNN) capture the inter-connectivity of adjacent regions. Brito et al. firstly use the connectivity properties of mutual nearest neighborhood graphs [15], and recently it is effectively used in classification [16,17] and clustering [18]. MkNN method reduces the computational complexity for large data sets. However, parameter k still exists, a bad k may led to unsatisfactory results.

3. Natural neighbor

Neither KNN nor RkNN, the problem of parameter selection is not in a position to avoid. As several studies have revealed, the solution of this problem often relies on the estimation of datas characteristics, by means of determining local decision boundaries in which the shape of the neighborhood can be modified to be more elongated. In this literature, we present a new term, natural neighbor, not only to estimate the parameter k, but also to explore a new way of nearest neighbor method without the parameter of k.

The natural neighbor method is inspired by the friendship of human society.

3.1. Friendship of human society

Friendship is a relationship of mutual affection between two or more people. It is a stronger form of interpersonal bond than an association. Friendship has been investigated in academic fields such as sociology, social psychology, anthropology, and philosophy.

As we know, friendship is the most elementary relationships in our life. As human, everyone must have one or more friends. If you wish to get along well with others, you are required to be friendly, and it would be used to make more friends.

Now we get two concepts, friendship and friendly. Case 1: considering a sample relationship in three people, named A, B and C. If A is friendly to B, B is friendly to A, C is friendly to A, we can only say that A and B are friends, A is a unilateral friend of C, but not a real friend of C. Thus the first friendship comes into being (A and B), and in contrast, we named the relationship between A and C unilateral friendship.

Case 2: then we can take such problem bigger. There are many people here in the virtual society, and everyone owns some, maybe three, unilateral friends. Everyone is friendly to three people in this case, and two people are friends if and only if each of them is mutually friendly to the other one. Therefore, some people may have friends, and some of them may not. It is dependent on the number of unilateral friends we choose.

Case 3: thinking about a city in the real world, we can ride the problem more complex. In our daily life, it is impossible to calculate the number to distinguish between friendly and unfriendly clearly. Then, someone who is more friendly may has more friends. So in this case, everyone here has a ranking list of the others, in which the higher level means friendlier. With the help of the principles below, we can find the friendship of this city. Firstly, everyone must have one or more friends. Secondly, two people are friends if and only if each of them is mutually friendly to the other one. Thirdly, the friendship is found by searching the lists of all people from beginning to the end. Absolutely, if we search everyones whole lists, everyone will have at least one friend. But it is necessary only if there is one person whose name lies on the end of everyones lists.

Case 4: the stranger. When there is a stranger in the city, his name must lie on the end of everyones lists because no one knows him. Thus we cannot find the real friendship of the city if we use the principles above. So the first principle must be modified. Let the city has the population of n, and then one stranger comes into the city. It is known that without the stranger, the friendship is found when the list number increases to r in case 3. It is clear that no one is friendly to the stranger until the list number increases to n, and we only want the number increases to r and stop, $r = n$ is a wrong result. Thus, the new principle is: anyone who belongs to the city must have one or more friends; the stranger(s) can have no friend only if the others have one or more friends, and when the list number continue increase, the people who have no friends still have no friends.

3.2. Natural neighbor method

As mentioned, the natural neighbor method is inspired by the friendship of human society to find the optimal path to overcome the disadvantages in KNN and RkNN. Particularly, the natural neighbor method can effectively determinate the neighborhood in a data set without the given parameter k and, meanwhile, calculate an approximate k.

The natural spirit of our method mainly manifests in three aspects: the neighborhood, the searching algorithm and the number of neighbors. Firstly, this neighborhood is inspired by the friendship of human society. Secondly, the searching algorithm can independently find the neighbor without human intervention. Thirdly, the process of determining the natural neighbor is a passive process, the number of all points neighbors is mutually independent, and it embodies the thought of nature.

3.2.1. Concepts of natural neighbor method

Given a set of data points $x_1, x_2, x_3, \ldots, x_n$ and some notion of similarity s_{ij} between all pairs of data points x_i and x_j, the goal is to find the natural neighbors of the points in the data set. Particularly, the notion of similarity can be given as prior knowledge, or be calculated and stored in a distance matrix. One of the most popular choices to measure this distance is known as Euclidean.

In what follows, we assume that X is a set of points, s_{ij} is the similarity between two points x_i and x_j. With the help of comparing the similarity, let $findKNN(x_i, r)$ denote the function of KNN searching which return the rth nearest neighbor of point x_i, $KNN_r(x_i)$ is a subset of X, and it is defined as follow

$$KNN_r(x_i) = \bigcup_{n=1}^{r} \{findKNN(x_i, n)\} \quad (3)$$

Definition 3 (Stable Searching State)**.** The natural neighbor searching process reach Stable Searching State only if:

$$(\forall x_i)(\exists x_j)(r \in N) \wedge (x_i \neq x_j) \\ \rightarrow (x_i \in KNN_r(x_j)) \wedge (x_j \in KNN_r(x_i)) \quad (4)$$

when the searching round r increases from 1 to λ.

Here, RkNN method is used to restrict the KNN searching progress, and value of the searching round is close to the optimum value of KNN method. As same as the case of stranger in the previous subsection, the noise can significantly enlarge the value of searching round. Thus Stable Searching State is focus on the normal points, and the definition of noise is presented after the definition of Natural Neighbor.

Definition 4 (Natural Neighbor Eigenvalue)**.** When the algorithm reaching the Stable Searching State, Natural Neighbor Eigenvalue(NaNE) λ is equal to the searching round r.

$$\lambda \triangleq r_{r\in N}\{r|(\forall x_i)(\exists x_j)(r \in N) \wedge (x_i \neq x_j) \\ \rightarrow (x_i \in KNN_r(x_j)) \wedge (x_j \in KNN_r(x_i))\} \quad (5)$$

Definition 5 (Natural neighbor)**.** Natural neighbor of x_i is defined as follow

$$x_j \in NN(x_i) \Leftrightarrow (x_i \in KNN_\lambda(x_j)) \wedge (x_j \in KNN_\lambda(x_i)) \quad (6)$$

Here, a significant difference between traditional neighbor and natural neighbor is the number of the neighbor. Thus, we may extract more interesting information from a data set by this natural way.

Definition 6 (Noise)**.** A data point belongs to noises only if all points except noises reach Stable Searching State and it cannot have only natural neighbor after $\sqrt{\lambda}$ more rounds.

Definition 7 (Natural Neighborhood Graph)**.** When the algorithm reaching the Stable Searching State, the neighborhood structure of the data set constitutes Natural Neighborhood Graph (NaNG). Each vertex v_i in this graph represents a data point x_i. Two vertexes x_i and x_j are connected if the data point x_i is a natural neighbor of x_j.

The Natural Neighborhood Graph is a nice way of representing the relationship of the data set. In addition, the graph can have different forms depending on its applications, e.g., Weighted Natural Neighborhood Graph (WNaNG) in NaNE calculation and Maximum Neighborhood Graph (MNG) in outlier detection.

The following is the description of natural neighbor algorithm.

Here λ is the value of NaNE, each element of set *NaN_Edge* is a set of edge which consists with two vertex(or one more vertex denote the weight of edge when WNaNG is used), $NaN\_Num(x_i)$ is the record of neighbor number of each data x_i. Recall that, *findKNN*(x_i, r, T)denote the function of KNN searching which return the $r-th$ nearest neighbor of point i, by using k-d tree T. In Algorithm 1, function $count(NaN\_Num(x_i) == 0)$ calculate the number of zero-natural-neighbor points, function $repeat(cnt)$ calculate the repetition of variable *cnt*. Here the Stable Searching State is determined by step 16. It is a slack rule, in order to avoid the destructive effect of noises in the data set. By the help of the functions $count(NaN\_Num(x_i) == 0)$ and $repeat(cnt)$, regardless of whether the noises exist, the algorithm can return the right result without the pre-processing step to denoise the data set.

The time complexity of this algorithm is $O(n * \log n)$. Algorithm 1 first creates a k-d tree of the data set, and the time complexity of this step is $O(n * \log n)$. After that, for each r, the complexity of natural neighbor searching is $O(n * \log n)$, so the complexity of the rest steps is $O(\lambda * n * \log n)$. The value range of λ must $2 \leq \lambda < n$, it is generally 6 or 7, and for high-dimensional or irregular data sets, the λ will be more than 20 but less than 30.

Algorithm 1 Natural neighbor algorithm.

```
1: r=1,flag=0, NaN_Edge = ∅
2: Create a k-d tree T from data set X
3: ∀x_i ∈ X,NaN_Num(x_i)=0
4: while flag==0 do
5:     for all x_i ∈ X do
6:         knn_r(x_i)=findKNN(x_i, r, T)
7:         KNN_r(x_i)=KNN_r(x_i) ∪ {knn_r(x_i)}
8:         if x_i ∈ KNN_r(knn_r(x_i)) && {knn_r(x_i), x_i} ∉ NaN_Edge then
9:             NaN_Edge=NaN_Edge ∪ {x_i, knn_r(x_i)}
10:            NaN_Num(x_i)=NaN_Num(x_i)+1
11:            NaN_Num(knn_r(x_i))=NaN_Num(knn_r(x_i)) + 1
12:        end if
13:    end for
14:    cnt=count(NaN_Num(x_i)==0)
15:    rep=repeat(cnt)
16:    if all(NaN_Num(x_i)) ≠ 0 || rep ≥ √(r − rep) then
17:        flag = 1
18:    end if
19:    r = r + 1
20: end while
21: λ = r − 1
22: Return: λ, NaN_Edge, NaN_Num(x_i)
```

3.2.2. Characteristics analysis of NaN method

Theorem 1 (The invariant property of the NaN)**.** *If x_i is a natural neighbor of x_j when the algorithm doesn't reach the Searing Stable State, it is true that x_i will be a natural neighbor of x_j when the algorithm reaching the Searing Stable.*

$$(\forall r \leq \lambda)(x_i \in KNN_r(x_j) \wedge x_j \in KNN_r(x_i)) \\ \rightarrow x_i \in NN(x_j) \wedge x_j \in NN(x_i) \quad (7)$$

Proof.

1. $KNN_r(x_i) = \bigcup_{s=1}^{r}\{findKNN(x_i, s)\} \Rightarrow KNN_r(x_i) \subseteq KNN_\lambda(x_i), r \leq \lambda$
2. $x_i \in KNN_r(x_j) \wedge x_j \in KNN_r(x_i) \Rightarrow x_i \in KNN_\lambda(x_i) \wedge x_j \in KNN_\lambda(x_i)$
3. $x_i \in NN(x_j) \wedge x_j \in NN(x_i)$ □

Theorem 2 (The stability of Natural Neighbor Eigenvalue)**.** *Natural neighbor eigenvalue maintain its stability after repeated trials of the same data set.*

Proof.

1. Assume that NaNE doesn't maintain its stability, means that there exist at least two tests whose results are unequal. Let $X = \{x_1, x_2, x_3, \ldots, x_n\} \cup \{noises\}$ is the data set, the two eigenvalues are λ and λ' in test T and test T', and $\lambda < \lambda'$.
2. From algorithm 1 we can know that, during repeated trials of the same data set, the distance matrix maintain its stability.
3. Let $t = \lambda' - 1$, so $\lambda \leq t < \lambda'$. There is a point $x_i \in X$ and $numNaN_r(x_i)' = 0$ when $r = t$ because λ' is a NaNE in test T'.
4. Also $numNaN_r(x_i) \neq 0$ when $r = t$ because λ is a NaNE in test T and $\lambda \leq t$.
5. It follows that there is a point x_j in test T, and $(x_i \in KNN_r(x_j)) \wedge (x_j \in KNN_r(x_i))$.
6. By step 2 we know that $\forall x_i \in X$, $\forall r \in N$, $KNN_r(x_i)$ is unchanged.
7. $(x_i \in KNN_r(x_j)) \wedge (x_j \in KNN_r(x_i))$in test T' is also true, and thus $numNaN_r(x_i)' \geq 1$.
8. By step 3 $numNaN_r(x_i)' = 0$ is true, which contradicts that $numNaN_r(x_i)' \geq 1$ in step 7 when $r = t$.
9. assumption is invalid. □

Theorem 3 (The stability of Natural Neighbor Method)**.** *Natural neighbor method maintains its stability after repeated trials of the same data set.*

Proof. We improve the argument from the previous proof, refining the induction step. $\forall x_i \in X$, $\forall r \in N$, $KNN_r(x_i)$ is unchanged, and also λ is unchanged. When $r = \lambda$, $\forall x_i \in X$, $KNN_r(x_i)$ is unchanged. It follows that $\forall x_i \in X$, $NN_r(x_i)$ is unchanged. Hence, the natural neighborhood of each point, NaNE and NaNG maintains its stability. □

3.3. A fast Natural Neighbor Eigenvalue estimating algorithm

From the beginning, we know that the choice of k is essential in KNN method. In fact, k can be regarded as one of the most important factors that can strongly influence the quality of the result. Natural Neighbor Eigenvalue can be used to help the traditional KNN method avoiding the problem of parameter selection. Thus a new algorithm is proposed to estimate the optimal k effectively and exactly.

Definition 8 (Key vertex of Natural Neighborhood Graph)**.** Let $G = (V, E)$ denote NaNG, $V_{zero}(r)$ is a subset of V and $V_{zero}(r) = \{x|nb(x) == 0\}$, means the set of vertexes which have no neighbor when the searching depth is r. The vertexes set $V_{key} = \{v_{key}|v_{key} \in V_{zero}(\lambda - 1) - V_{zero}(\lambda)\}$ is the set of all key vertexes.

Note that the number of V_{key}: $|V_{key}| \geq 1$.

Recall that Weighted Natural Neighborhood Graph (WNaNG) is a special form of NaNG which we mentioned previously.

Definition 9 (Weighted Natural Neighborhood Graph)**.** Weighted Natural Neighborhood Graph associates a weight with every edge in Natural Neighborhood Graph. the weight of each edge identified the searching round of two vertexes becomes natural neighbors.

Lema 1. Let $G_{weight} = (V, E)$ is WNaNG, $neighborSort(x, y) = k$ denote x is the kth nearest neighbor of y, the weight of an edge (x, y) is defined as follows:

$$weight(x, y) = \max[neighborSort(x, y), neighborSort(y, x)]$$

Here weight(e) results in λ if one of its vertex is a key vertex.

Our goal is to design an effective algorithm to calculate NaNE of the data set, and the key vertex searching is an essential part of that. As we know, the weight of each key edge is equal to NaNE in WNaNG, and thus it is easy to calculate the eigenvalue if a point in the data set is found as a key vertex in WNaNG.

The time complexity of this algorithm is $O(n*\log n)$. Algorithm 2 firstly creates a k-d tree of the data set, and the time complexity of this step is $O(n*\log n)$. The complexity of step 3–11 in each circle is $O(zeroNum*\log n)$, and $zeroNum$ begins with n and drops exponentially, so the complexity of all circles similar to $O(n*\log n)$. The last steps aims at the vertexes without neighbor after steps 11, the number of them are too small after the exponentially drops, so their complexity can be ignored.

4. Experimental evaluation

In general, the proposed NaN algorithm is not a stand-alone algorithm; it is always integrated with other application algorithm for different areas. So in this section, firstly we just show NaNG in some characteristic data sets. Secondly, we experimentally analyze the stability of NaN method. Thirdly, we performed a comparative study of NaN and other competing neighborhood construction algorithms for clustering and outlier detection. All algorithms and experiments run on Matlab R2010a.

Algorithm 2 Natural neighbor eigenvalue estimating algorithm.

1: Create a k-d tree T from data set X
2: Initial: $r = 1$, $zeroNum = N$, $vertexWithoutNeighbor = X$
3: Find rth neighbor y of each point x in $vertexWithoutNeighbor$ by using k-d tree T
4: **if** x belongs to y's first r neighbors **then**
5: $zeroNum = zeroNum - 1$
6: Remove x and y from $vertexWithoutNeighbor$
7: **end if**
8: **if** $zeroNum$ is changed **then**
9: $r = r + 1$
10: Return to step 3
11: **end if**
12: **if** $zeroNum \neq 0$ **then**
13: Find first r neighbors $\{y_1, y_2, \ldots, y_r\}$ of each point x in $vertexWithoutNeighbor$
14: $weight(x) = min(weight(y_i, x))$
15: Sort all unequal weights and let r=weight(1), $weight(1) < weight(2) < \cdots < weight(k)$ $(k \leq zeroNum + 1)$
16: $\lambda = weight(i)$ if $weight(i + 1) - weight(i) \leq \sqrt{weight(i)}$
17: **else** $\lambda = r$
18: **end if**
19: Return: λ

4.1. Natural Neighborhood Graph

In order to show the characteristics of NaN method intuitively, the neighborhood graphs of NaN and KNN are presented. The graphs of four synthetic data sets (see in Fig. 1) are 2-dimensional, which makes the validation easier from just visualization. Train [19] distributes as four clusters of different size and various sharps with outliers. Flame [20] is generated as a large arrow and two noise. Compound [21] is composed of six different structures of clusters, where there are connected clusters, noise and embedded-cluster. Spiral and 3D-spiral [22] consists of three spiral clusters. Fig. 1 shows that the neighborhood graph of our method wins a higher performance on representing the relationship of the data set.

It can be seen from Fig. 1 that NaNG can maintain the basic shape of the data, and present a reasonable natural neighbor eigenvalue and cluster result. Besides, the outliers in train are significantly identified, and there is no short-circuit error in spirals. The experimental results show that NaN method can provide a perfect neighborhood graph without any priori information about the data set. Traditional neighborhood graphs(KNN in Fig. 1(b) and (c)) can reach a same effect if they choose a good parameter under some circumstances, and sometimes their shortage is unavoidable.

4.2. The stability of Natural Neighbor Eigenvalue

Firstly, to perform the statistical test for the stability of natural neighbor eigenvalue, 200 uniform data sets with points number N ranging from 500 to 1000 are generated. Fig. 2.a shows that the eigenvalue can maintain its stability in uniform distribution, the value is nearly equal to six. And then, to perform the statistical test for the stability of natural neighbor eigenvalue, 200 Gaussian data sets with N ranging from 500 to 1000 are generated. Fig. 2.b shows that the eigenvalue can maintain its stability in Gaussian distribution, the value is nearly equal to eight.

The result in Fig. 2 shows that:

1. Whether or not the data scale is equal, the eigenvalue in uniform distribution has a great difference of which in Gaussian distribution.

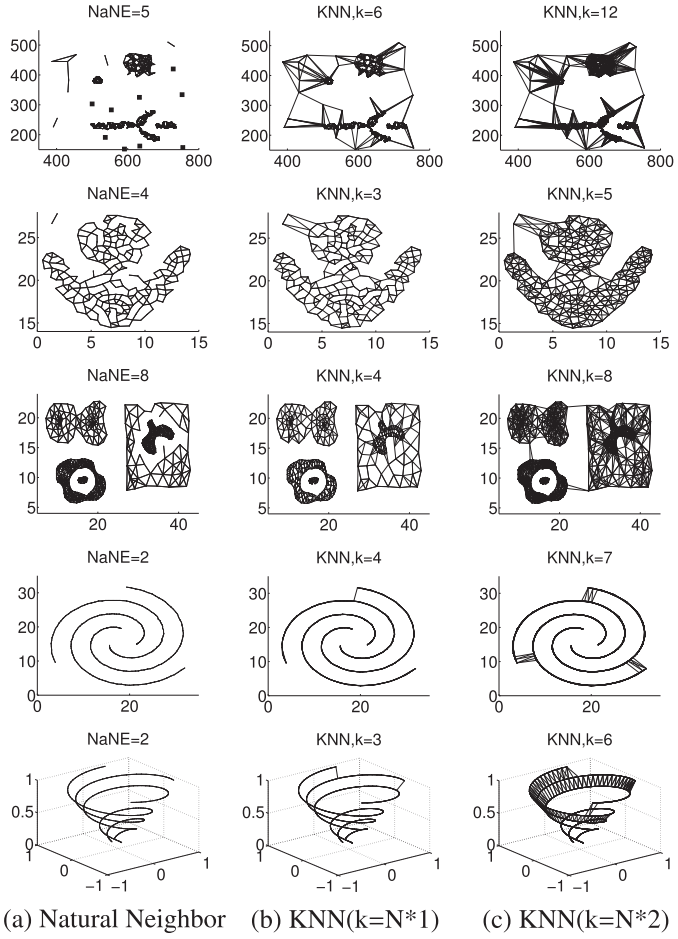

Fig. 1. The neighborhood graph of NaN(no parameter) and KNN method. (a) The result of NaN method (include the eigenvalue and the neighborhood graph); (b) and (c): The results of KNN method for different numbers of parameter k. From top to bottom the data sets are: Train, Flame, Compound Spiral and 3D-Spiral.

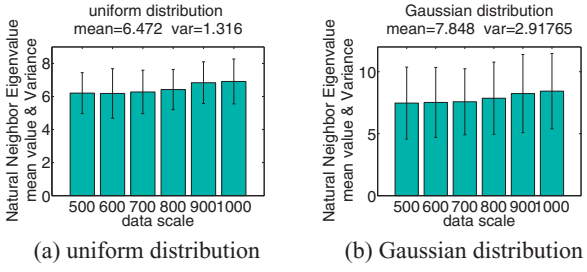

Fig. 2. Natural Neighbor Eigenvalue(mean value & Variance in 200 times), N ranging from 500 to 1000. (a) Uniform (Mean = 6.472, Var = 1.316). (b) Gaussian (Mean = 7.848, Var = 2.91765).

2. Natural neighbor eigenvalue maintain its stability after repeated trials of the data set of different scale.

These experimental results demonstrate the robustness of the proposed approach.

4.3. KNN method based on Natural Neighbor Eigenvalue

In this section, Natural Neighbor Eigenvalue(NaNE) is used to be the parameter of k in traditional KNN method. To demonstrate the generality of our method, two basic application fields, namely, outlier detection and classification are used for performance comparisons.

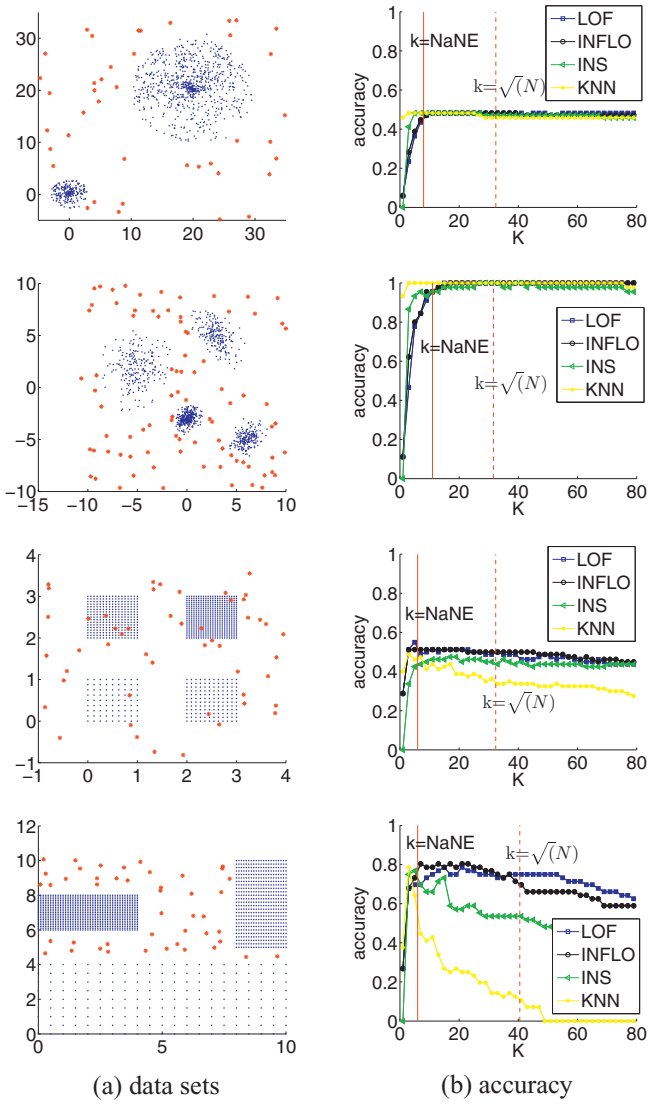

Fig. 3. Detection accuracies of the four detection methods over a range of k values (0,80). The solid line indicates the accuracies of four methods when we use NaNE to be the parameter k, the broken line indicates $\sqrt{N}$.

Outlier detection: a comparative study based on four synthetic data sets [23] was conducted in order to show the effectiveness of the proposed method. The data sets were designed to consider the various cluster patterns, different degrees of cluster density, and different cluster sizes in order to evaluate different methods in a harsh testing environment. Fig. 3 shows the test data sets and the outlier detection results of four methods: KNN, LOF, INFLO and INS. Our method is significantly better than the choice of $\sqrt{N}$ in the four data sets. And more importantly, our method can offer a flexible parameter which is suited for every data set to achieve a high outlier detection accuracy.

Classification: we have experimented with five data sets from the UCI machine learning data repository for the evaluation of the proposed method using 10-fold partitioning. Classification accuracy of the KNN method of our eigenvalue(NaNE) is compared with that of the KNN algorithm with $k = 1,3,5$ [24] and $k = \sqrt{n}$ [25]. In this paper, the Euclidean distance is used for performance comparisons. Classification accuracies and precisions are shown in Table 1.

From Table 1, overall NaNE method achieves higher accuracy and precision when the Euclidean distance is considered. Our method wins in 3 out of 5 cases over the choice of $k = \sqrt{N}$.

Table 1
Comparison of accuracy & precision of KNN parament selection with Euclidean distance.

	Data sets	$k=1$	$k=3$	$k=5$	$k=\sqrt{n}$	$k=NaNE$
Accuracy	CANC.	91.633	92.476	92.970	93.145	93.656
	VEHI.	64.935	62.151	56.850	65.891	63.676
	SONA.	81.997	71.676	70.969	79.891	82.202
	IONO.	86.639	84.815	88.311	84.853	87.793
	IRIS	95.733	96.533	97.061	97.201	96.288
Precision	CANC.	91.505	92.307	92.888	93.139	93.664
	VEHI.	64.197	64.237	60.324	59.008	64.696
	SONA.	82.140	82.364	72.161	70.786	80.077
	IONO.	88.868	89.128	87.620	81.649	87.316
	IRIS	96.266	95.544	96.842	96.681	97.438
Overall		84.392	83.124	81.600	82.225	84.681

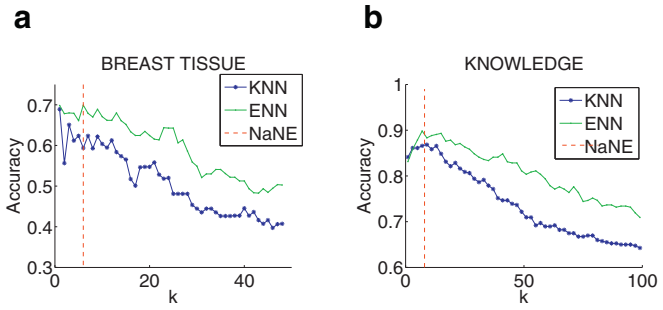

Fig. 4. Classification accuracies of KNN and ENN methods over a range of k. The solid line indicates the accuracies of two method when we use NaNE to be the parameter k.

Particularly, the classification precision of the proposed method wins in 5 out of 5 cases over the choice of $k=\sqrt{N}$.

4.4. ENN method based on Natural Neighbor Eigenvalue

Recently, based on MkNN method, Tang and He propose an Extended nearest neighbor (ENN) Method for classification which makes use of the two-way communication style [16]. Unlike the classic KNN rule which only considers the nearest neighbors of a test sample to make a classification decision, ENN method considers not only who are the nearest neighbors of the test sample, but also who consider the test sample as their nearest neighbors. We experiment with 15 data sets from the UCI machine learning data repository for the evaluation of the proposed method using 10-fold partitioning. We set a continuously changing k in first two data sets, and choose the parameter k similar to KNN classification experiment in the last data sets.

Fig. 4 firstly demonstrates the effectiveness of the ENN method. Furthermore, it is clear that the parameter choice problem still exists in ENN, and our NaNE provides a good approach. Because of the inner relation between KNN and MkNN, both of them have the similar parameter selection problem, and NaNE can help MkNN based method choose the parameter k without any priori knowledge.

From Table 2, our method wins in 2 out of 13 cases over all other choices, and it is very close to the best one in most situations. Overall, NaNE method achieves the highest accuracy. This result demonstrates the universality of NaNE, and this characteristic exists in most nearest neighbor based methods. Therefore, we have the chance to innovate and solve those problems in KNN, RkNN and MkNN with our NaNE, or more NaN based method.

Table 2
Accuracy comparison of ENN parament selection.

Data sets	$k=1$	$k=3$	$k=5$	$k=\sqrt{n}$	$k=NaNE$
IRIS	96.000	95.333	94.000	96.000	95.333
DIAB.	100.000	100.000	100.000	100.000	100.000
ECOL.	73.770	74.091	77.344	79.768	77.995
HABE.	69.806	60.677	68.215	73.763	71.806
IONO.	80.325	84.024	86.016	86.000	80.000
SONA.	43.214	45.095	40.690	43.643	42.262
VEHI.	62.294	63.464	64.525	60.745	63.700
WINE	69.444	67.320	68.366	64.477	68.399
SEGE.	66.667	65.238	66.667	65.238	68.095
CANC.	90.695	91.563	91.917	92.801	92.976
LETT.	29.975	30.305	30.195	22.050	28.580
LIBR.	33.333	33.111	30.444	22.222	33.111
PAGE.	79.787	73.758	78.910	88.411	87.114
Overall	68.870	67.998	69.022	68.855	69.952

4.5. An outlier detection algorithm based on natural neighbor

Recently, a density-based algorithm for outlier detection based on natural neighbor is proposed [26]. It is a no-parametric algorithm with well-deserved adaptation for both synthetic data sets and real data sets. In that paper, Maximum Neighborhood Graph (MNG), a new form of NaNG is constructed for outlier mining and gains a good result.

5. Conclusions

Inspired by the friendship of human society, in this work we present the natural neighbor method, NaN. In contrast to the traditional neighbor methods, NaN is parameter-free, and it finds multiple-valued neighbors of each data point by considering the characteristics of the data set. Our method can improve the performance in handling noise and manifold data. We present the effectivity of NaN method on the outlier detection and classification. It can also be used in other areas such as clustering, image segmentation, and face recognition, insteading of KNN method and win a better performance.

Natural neighbor method is a new thinking about nearest neighbor method, further work should address the reduction of the complexity and incorporation of better application areas.

Acknowledgments

This work was supported by the National Nature Science Foundation of China (No. 61272194 and No. 61073058) Natural Science Foundation Project of CQ CSTC (cstc2013jcyjA40049) and the Fundamental Research Funds for the Central Universities(106112013CDJZR180015).

References

[1] K.H. Ambert, A.M. Cohen, k-information gain scaled nearest neighbors: A novel approach to classifying protein-protein interaction-related documents., IEEE/ACM Trans. Comput. Biol. Bioinf. 9 (1) (2011) 305–310.
[2] B. McFee, C. Galleguillos, G. Lanckriet, Contextual object localization with multiple kernel nearest neighbor., IEEE Trans. Image Process. 20 (2) (2011) 570–585.
[3] G. Bhattacharya, K. Ghosh, A.S. Chowdhury, Outlier detection using neighborhood rank difference, Pattern Recognit. Lett. 60 (2015) 24–31.
[4] S.S. Stevens, S.S. Stevens, Mathematics, measurement, and psychophysics., S.S.Stevens Handbook of Experimental Psychology, 1951, pp. 1–49.
[5] J. Wang, P. Neskovic, L.N. Cooper, Improving nearest neighbor rule with a simple adaptive distance measure., Pattern Recognit. Lett. 28 (2) (2006) 43–46.
[6] S. Garcia, J. Derrac, J.R. Cano, F. Herrera, Prototype selection for nearest neighbor classification: taxonomy and empirical study, IEEE Trans. Pattern Anal. Mach. Intell. 34 (3) (2011) 417–435.
[7] F. Qian, K. Chiew, Q. He, H. Huang, Mining regional co-location patterns with knng, J. Intell. Inf. Syst. 42 (3) (2013) 485–505, doi:10.1007/s10844-013-0280-5.

[8] A.K. Ghosh, On optimum choice of k in nearest neighbor classification, Comput. Stat. Data Anal. 50 (11) (2006) 3113–3123.
[9] A.K. Ghosh, On nearest neighbor classification using adaptive choice of k, J. Comput. Graph. Stat. 16 (2) (2007) 482–502.
[10] C. Domeniconi, J. Peng, D. Gunopulos, Locally adaptive metric nearest neighbor classification, in: Proceedings of the IEEE Transaction on Pattern Analysis and Machine Intelligence, 2002, pp. 1281–1285.
[11] G. Bhattacharya, K. Ghosh, A.S. Chowdhury, An affinity-based new local distance function and similarity measure for KNN algorithm., Pattern Recognit. Lett. 33 (3) (2012) 356–363.
[12] F. Korn, S. Muthukrishnan, Influence sets based on reverse nearest neighbor queries., Sigmod Record 29 (2) (2000) 201–212.
[13] M.L. Yiu, N. Mamoulis, Reverse nearest neighbors search in ad hoc subspaces, IEEE Trans. Knowl. Data Eng. 19 (3) (2007) 412–426.
[14] WANG, CHAI, A pruning based continuous RKNN query algorithm for large k, Chin. J. Electr. 3 (2012) 523–527.
[15] M. Brito, E. Chvez, A. Quiroz, J. Yukich, Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection, Stat. Probab. Lett. 35 (1) (1997) 33–42.
[16] B. Tang, H. He, Enn: Extended nearest neighbor method for pattern recognition [research frontier], IEEE Comput. Intell. Mag. 10 (3) (2015) 52–60.
[17] P. Shivakumara, A. Dutta, T.Q. Phan, C.L. Tan, U. Pal, A novel mutual nearest neighbor based symmetry for text frame classification in video, Pattern Recognit. 44 (8) (2011) 1671–1683.
[18] H. Huang, Y. Gao, K. Chiew, L. Chen, Q. He, Towards effective and efficient mining of arbitrary shaped clusters, in: ICDE, 2014, pp. 28–39.
[19] T. Inkaya, S. Kayalgil, N.E. zdemirel, An adaptive neighbourhood construction algorithm based on density and connectivity, Pattern Recognit. Lett. 52 (2015) 17–24.
[20] L. Fu, E. Medico, Flame, a novel fuzzy clustering method for the analysis of dna microarray data, Bmc Bioinf. 8 (1) (2007) 1–15.
[21] C.T. Zahn, Graph-theoretical methods for detecting and describing gestalt clusters, IEEE Trans. Comput. 20 (1) (1971) 68–86.
[22] H. Chang, D.-Y. Yeung, Robust path-based spectral clustering, Pattern Recognit. 41 (1) (2008) 191–203.
[23] J. Ha, S. Seok, J.S. Lee, Robust outlier detection using the instability factor, Knowl. Based Syst. 63 (3) (2014) 15–23.
[24] K. Kozak, M. Kozak, K. Stapor, Weighted k-nearest-neighbor techniques for high throughput screening data, Int. J. Biomed. Sci 1 (2006) 155–160.
[25] P. Mitra, C.A. Murthy, S.K. Pal, Unsupervised feature selection using feature similarity, IEEE Trans. Pattern Anal. Mach. Intell. 24 (3) (2002) 301–312.
[26] J. Huang, Q. Zhu, L. Yang, J. Feng, A non-parameter outlier detection algorithm based on natural neighbor, Knowl. Based Syst. 92 (2016) 71–77.