A Fast Directed Tree Based Neighborhood Clustering Algorithm for Image Segmentation

Jundi Ding[1], SongCan Chen[1], RuNing Ma[2],and Bo Wang[1]

[1] Nanjing University of Aeronautics and Astronautics,
College of Information Science & Technology, 210016, P.R.China,
s.chen@nuaa.edu.cn,
WWW home page: http://parnec.nuaa.edu.cn
[2] Nanjing University of Aeronautics and Astronautics,
College of Science, 210016, P.R.China

Abstract. First, a modified Neighborhood-Based Clustering (MNBC) algorithm using the directed tree for data clustering is presented. It represents a dataset as some directed trees corresponding to meaningful clusters. Governed by Neighborhood-based Density Factor (NDF), it also can discover clusters of arbitrary shape and different densities like NBC. Moreover, it greatly simplify NBC. However, a failure applying in image segmentation is due to an unsuitable use of Euclidean distance between image pixels. Second, Gray NDF (GNDF) is introduced to make MNBC suitable for image segmentation. The dataset to be segmented is all grays and thus MNBC has the constant computational complexity O(256). The experiments on synthetic datasets and real-world images shows that MNBC outperforms some existing graph-theoretical approaches in terms of computation time as well as segmentation effect.

1 Introduction

Neighborhood-Based Clustering (NBC) algorithm [1] proposed by Zhou S. G. etc is a good data clustering algorithm and can discover clusters of arbitrary shape and different densities using the neighborhood relationship among data points. Experiments in [1] show that NBC is advantageous over DBSCAN [2] in both clustering effectiveness and efficiency. However, in order to develop the algorithm the authors introduced thirteen pre-requisite definitions including the neighborhood based density factor (NDF). Besides, they incorporated the cell-based structure and VA file [3] for clustering very large and high dimensional databases. The two aspects mentioned above make NBC conceptually and structurally complex. In addition, NBC fails in segmenting an image due to an unsuitable use of Euclidean distance between image pixels.

In fact, we just require three key definitions from the thirteen basic ones in NBC, i.e. k-neighborhood, reverse k-neighborhood and NDF, and additionally borrow the idea of directed tree to develop a modified NBC (MNBC) for data clustering, which not only simplifies NBC but also can discover clusters of arbitrary shape and different densities like NBC. It represents a dataset as some

directed trees corresponding to meaningful clusters. So, its goal is to find the numbers of directed trees constructed in a top-down strategy. On the other hand, we introduce Grayscale k-neighborhood, Grayscale reverse k-neighborhood and Grayscale NDF (GNDF) and apply MNBC to image segmentation. GNDF characterizes the local density of a gray scale's neighborhood in a relative sense. And MNBC governed by GNDF takes the 256 intensities in a common gray image $I_{m\times n}$ (encoded with 8-bit resolution, m and n are the numbers of rows and columns respectively) as the dataset to be segmented and accomplishes segmentation fast and efficiently. Its computational complexity is O(256), which is independent of the size of the image $m \times n$.

There is some of the related work to our approach: early graph-based methods (EGA) [6], spectral clustering algorithms (SCA) [4], [5], minimum spanning trees (MST) based clustering algorithm [7], [8]. EGA is to generate directed trees for data clustering with a bottom-up process and also guided by a single-scalar control variable but the user must specify it by cross-validation. Its computational complexity is O(N^2). SCA cluster points using eigenvectors of affinity matrices derived from the data set. While powerful, computational cost remains a major obstacle for real-time applications. Its computational complexities is O(N^3). MST based clustering algorithm is a greedy one for segmenting images based on intensity differences between neighboring pixels and requires O($MlogM$),where M is the number of edges in the graph.

The remainder of this paper is organized as follows: Section 2 gives an overview of NBC and refines its three key definitions. The three key definitions avail to design MNBC. Section 3 describes MNBC in detail and presents the evaluation results on some synthetic toy datasets with (EGA) [6], (SCA) [4], [5] and MST [7], [8] to show the good performance of MNBC. Section 4 introduces GNDF and details MNBC for image segmentation, while Sect. 5 delivers comparisons on real world images with MST [7], [8], and Sect. 6 concludes the whole paper.

2 Review of NBC

NBC algorithm [1] uses the neighborhood relationship among data points to build a neighborhood based clustering model with goal to discover clusters of arbitrary shape and different densities. In the description of NBC algorithm, the authors had to introduce thirteen pre-requisite definitions including the neighborhood based density factor (NDF). Here we refine its thirteen basic concepts into just three ones: k-neighborhood, reverse k-neighborhood and NDF. The three key definitions facilitate to design MNBC based on the directed tree. Given a dataset, $X = \{x_1, x_2, \cdots, x_N\}$, N is the size of the d-dimension data set. Euclidean distance between x and y is denoted by dist(x, y).

Definition 1. *(k-Neighborhood) The k-nearest neighbors set of x ($kNN(x)$) is a set of k nearest neighbors of x($k > 0$), then the x's k-neighborhood ($kNB(x)$) is the set of objects that lie within the circle region with x as the center and r*

as the radius, where r is the maximal distance of between x and kNN(x), i.e. $\exists z \in kNN(x)$, $r = dist\ (x, z)$, *s.t.* $\forall y \in kNN(x)$, $dist(x, y) \leq r$.

Definition 2. *(Reverse k-Neighborhood) The reverse k-neighborhood of x (R-kNB(x)) is the set of objects whose k neighborhood contain x, which can be formally represented as* $R\text{-}kNB(x) = \{y \in X : x \in kNB(y)\}$

Definition 3. *(Neighbor-based Density Factor) The neighbor-based density factor of data point x, denoted by NDF(x),is evaluated as follows:*

$$NDF(x) = \frac{|R\text{-}kNB(x)|}{|kNB(x)|} \tag{1}$$

In practice, $|k\text{NN}(x)|$ is around k for a given single-scalar control variable k. According to Definition 1, it may be a little greater but not less than k. $|\text{R-}k\text{NB}(x)|$ is quite discrepant for different data points. As a result, there are three situations for $\text{NDF}(x)$: larger than 1 (dense point), equal to 1 (even point) and less than 1 (sparse point) [1]. In MNBC, the data points with $\text{NDF}(x) \geq 1$ are seed nodes, which could be taken as a root node while the others with $\text{NDF}(x) < 1$ can only be taken as leaf nodes or outlier nodes appearing in no directed trees.

3 MNBC

In this section, we describe MNBC. EGA [6] generates directed trees in a bottom-up process and while MNBC adopts a top-down process to construct the directed trees. We will begin by discussing the concepts of graph theory (see [9] and [6]) which are pertinent to MNBC in Sect.3.1 and then proceed to construct the direct trees in Sect.3.2. The evaluation results on some synthetic toy datasets are presented in Sect.3.3.

3.1 The concepts of graph theory

Definition 4. *(Directed Graph and Directed Path) A directed graph is a set of nodes and arcs, each arc leading from an initial node A to a final node A'. A set of arcs* $e_1, e_2, \cdots, e_n$ *is said to be a directed path from A to A', if A is the initial node of* e_1*, A' is the final node of* e_n*, and the final node of* e_k *is the initial node of* e_{k+1} *for* $k = 1, 2, \cdots, n-1$.

Definition 5. *(Directed Tree) A directed tree is a directed graph satisfying 1) Every node* $A \neq R$ *is the final node of exactly one arc; 2) L is the initial node of no arc; 3) R is the final node of no arc; 4) There is no directed path from a node A to itself (i.e. no cycles).*

The nodes R and L are called the root and leaf of the directed tree respectively. The final node of the arc whose initial node is A is called the child node of A, denoted C(A). Notice that the root of a directed tree must be unique but the leaf of a directed tree can be more than one, and a path from the root to one of the leaves in a directed tree is unique and consists of the arcs from R to one C(R), C(R) to one C(C(R)), etc.

3.2 Construction of the Directed Trees based on NDF

For a given k, MNBC is made up of three phases:
(P1) Computing all kNB(x), R-kNB(x) and NDF (x) according to (1);
(P2) Constructing all directed trees based on NDF evaluated in P1;
(P3) Nodes exist in no directed tree are called outliers.
Obviously, one or more directed trees can be constructed in P2, dependent of the single variable k. The following algorithmic steps summarize P2:

1. *Initially,* $numT = 0$, $V = \{x : NDF(x) \geq 1, x \in X\}$;
2. *While* $V \neq \emptyset$, *an arbitrary* $x \in V$ *is taken as a root node to construct* T_x *(the directed tree of* x*):*
   $T_x = \emptyset$; $C(x) = kNB(x)$; $T_x = \{x\} \cup C(x)$;
   $Y = \{y : y \in C(x), NDF(y) \geq 1\}$;
   While $Y \neq \emptyset$,
   For each seed node $y \in Y$, $C(y) = \{z : z \in kNB(y), z \notin T_x\}$
   If $C(y) = \emptyset$, y *becomes a leaf node of* T_x;
   Else y *is a root node of the subtree* $T_y, T_y = \{y\} \cup C(y)$;
   End
   End
   $C(C(x)) = \bigcup_{y \in Y} C(y), T_x = T_x \cup \bigcup_{y \in Y} T_y$;
   $C(x) = C(C(x))$; $Y = \{y : y \in C(x), NDF(y) \geq 1\}$;
   End
   $numT = numT + 1; X = X \backslash T_x; V = \{x : NDF(x) \geq 1, x \in X\}$;
   End

Complexity. The time complexity of P1 is O(N^2) because the most time-consuming work in P1 is the evaluation of kNB queries, which takes O(N^2). The recursive procedure of constructing the directed trees to discover clusters takes O(N) with only three key definitions, i.e. the time complexity of P2 is O(N). Therefore, the total computational complexity of MNBC is O(N^2). MNBC, robust to the order of the initial node selection, has the outstanding capability of discovering all clusters of arbitrary shape and recognizing the outlier points as well as NBC [1].

3.3 Synthetic Datasets and Experimental Results

To evaluate the performance of MNBC for data clustering, we compare it with EGA [6], SCA [4], [5] and MST [7], [8] on three synthetic datasets: three concentric circles (out-circle: 300 points; mediate-circle: 200 points; inner-circle: 100 points), two half circles (each has 500 points) and three spirals (each has 400 points). The NDF values of all data points in the respective dataset and the cluster results of MNBC identified by label are put in Fig.1, which shows that MNBC does not cluster data wrongly. The experimental results are illustrated in Fig.2. For each method, its parameters are tuned over a range in which their clustering results for the three toy data sets are different. To make a fair comparison, we carefully choose those parameters for each dataset which make each method

work best. From Fig.2, EGA and SCA perform poorly for the three synthetic toy data sets; whereas MNBC and MST have a good structural representation, especially the clustering results by MNBC are identical to the original synthetic data sets as show in the first row of Fig.2. Therefore, MNBC outperforms the others.

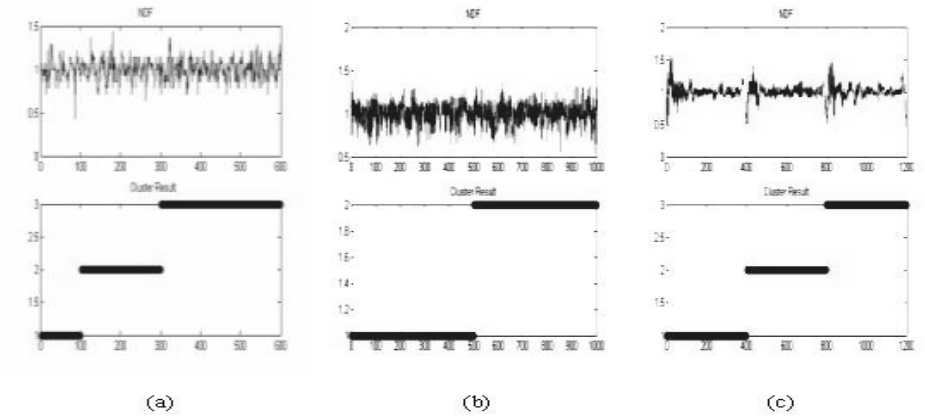

Fig. 1. The NDF curves and cluster labels in MNBC for (a) Three concentric circles (N=600, k=14); (b) Two half circles (N=1000, k=50); (c) Three spirals (N=1200, k=25)

4 GNDF and MNBC for image segmentation

Image segmentation is the most essential and important step of any low-level vision system. In general, a common gray image $I_{m\times n}$ is encoded with 8-bit resolution and has at most 256 grays (m and n are the number of rows and columns respectively). To apply MNBC to segment an image fast and efficiently, we introduce Grayscale k-neighborhood, Grayscale reverse k-neighborhood and Grayscale NDF (GNDF), which characterizes the local density of a gray's neighborhood in a relative sense. MNBC governed by GNDF takes the 256 intensities as the dataset to be segmented and has the computational complexity O(256), which is independent of the size of the image $m \times n$.

4.1 Grayscale Neighborhood-based Density Factor (GNDF)

Suppose $I = \{0, 1, \cdots, N\}, 0 \leq N \leq 255$, then GNDF is given in Definition 6.

Definition 6. *(Grayscale Neighborhood-based Density Factor)*

$$GNDF(q) = \frac{|R\text{-}kNB(q)|}{|kNB(q)|}, q = 0, 1, \cdots, 255 \tag{2}$$

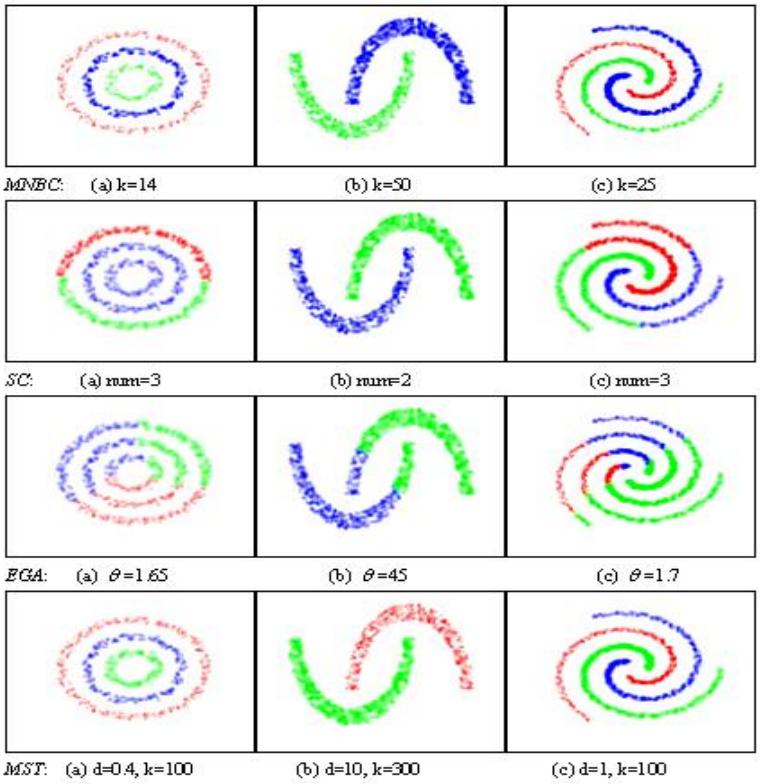

Fig. 2. Clustering results: (a) Original data set and MNBC (1st row); (b) SC (2nd row); (c) EGA (3rd row); (d) MST (4th row)

Denote $num(q)$ as the number of pixels whose intensity is q and $l(q)$ as a natural number satisfying $(num(q) \neq 0)$

$$l(q) = \min \left\{ l(q) \geq 0; \sum_{x \in I, |x-q| \leq l(q)} num(x) \geq k \right\} \tag{3}$$

Then

$$kNB(q) = I \cap \{q - l(q), q - l(q) + 1, \cdots, q + l(q) - 1, q + l(q)\}, \tag{4}$$

$$R\text{-}kNB(q) = \{y \in I; y \in kNB(q)\}, \tag{5}$$

where $kNB(q)$ and $R\text{-}kNB(q)$ are k-neighborhood and reverse k-neighborhood of the grayscale q respectively.

Because the grays of an image are consecutive natural numbers, both k-neighborhood and reverse k-neighborhood of arbitrary grays q are the intersection between the truncation of several consecutive natural numbers and X.

Further, it is easy to draw a conclusion in Proposition 1 but its proof is left out due to space of limitation:

Proposition 1. *If* $q_1 \leq q_2$*, then* $q_1 - l(q_1) \leq q_2 - l(q_2), q_1 + l(q_1) \leq q_2 + l(q_2)$.

Proposition 1 indicates that the left and right endpoint values of k-neighborhood of gray q both are monotonically increasing with respect to q, which implies that there is some expanded direction of k-neighborhood of the q. Since MNBC is robust to the initial node selection analyzed above, we can select the minimal gray q^* as the initial node to construct a directed tree.

Like NDF, GNDF of a gray will also probably be larger than 1 or equal to 1 or smaller than 1. The grays with GNDF$(q) \geq 1$ are seed nodes, which could be taken as root nodes while the others with GNDF$(q) < 1$ can only be taken as leaf nodes or outlier nodes not residing on any directed trees. Figure 3 illustrates a simple schematic diagram ($k = 200$), e.g. $num(28) = 50 < 200$, then according to (3) and (4), $l(28) = 2$, kNB$(28) = \{26, 27, 28, 29, 30\}$ and $|k\text{NB}(28)| = 282$ because $\sum_{q=27}^{29} num(q) = 134 < 200$ $\sum_{q=26}^{30} num(q) = 282 > 200$; According to (5), R-$kNB(28) = \{27, 28\}$, $|\text{R-}k\text{NB}(28)| = 104$, then GNDF$(28) = 104/282 < 1$ according to (2).

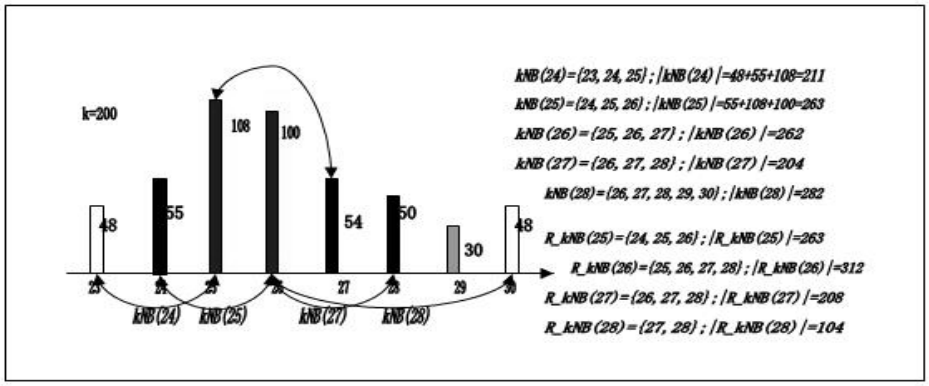

Fig. 3. A simple schematic diagram of kNB(q), R-kNB(q)

4.2 MNBC for Image Segmentation with GNDF

Similarly, MNBC based on GNDF is also made up of three phases:
(P1') Input k and compute GNDF (q) according to (2), $q = 1, \cdots, N$;
(P2') Construct the directed trees based on GNDF evaluated in P1';
(P3') Assign pixels to a corresponding directed tree constructed in P2' and the pixels with its gray not in any directed trees are designated as outliers.

However, P2' is different from P2. First, the directed trees are constructed with a non-decreasing order, namely, the root of each directed tree T_q is minimal gray q instead of arbitrary q, $q \in I, GNDF(q) \geq 1$. Second, according to Proposition 1, only the maximal seed node $p, p \in C(q), GNDF(p) \geq 1$ is qualified as the root of a subtree of T_q to expand T_q, denoted by T_p, where $C(q)$ is the children nodes of the q, while in P2 all seed nodes must be traversed over x's children nodes to expand T_x. Such a one-direction search makes it for MNBC to segment image easily and fast. P2' is summarized in the following:

1. *Initially,* $numT = 0$, $V = \{q : GNDF(q) \geq 1, q \in I\}$;
2. *While* $V \neq \emptyset$, $q^* = minV, q = q^*$, *then* q *is taken as a root node to construct* T_q *(the directed tree of* q*):*
   $T_q = \emptyset$; $C(q) = kNB(q) \backslash q$, $T_q = T_q \cup kNB(q)$;
   $P = \{p : p \in C(q), GNDF(p) \geq 1\}$;
   While $P \neq \emptyset$, $p^* = maxP$; $p = p^*$;
   $C(p) = \{o : o \in kNB(p) \backslash p, o \notin T_q\}$;
   If $C(p) = \emptyset$, p *becomes a leaf node of* $T_q, T_q = T_q$; *break*
   Else p *is a root node of* T_p *(a subtree of* T_q*):*
   $T_p = kNB(p); T_q = T_q \cup T_p$;
   $C(q) = C(p); P = \{p : p \in C(q), GNDF(p) \geq 1\}$;
   End
   End
   $numT = numT + 1; I = I \backslash T_q; V = \{q : GNDF(q) \geq 1, q \in I\}$;
   End

I) Selection of k. The single input parameter k determines the number of regions and the relative size of each region. It can be selected flexibly and purposefully. Let $n_{min} = min_{q \in I} num(q)$, $n_{tot} = \sum_{q \in I} num(q)$. When $k \leq n_{min}$, each gray itself becomes a single cluster. Hence, the number of so-formed regions will be close to 256, which is an over-segmentation problem. In contrast, when $k \geq n_{tot}$, all grays are grouped together to form a single cluster. Thus the number of regions formed is only 1, meaning an under-segmentation. To avoid these two unacceptable extreme cases, we should select k satisfying $n_{min} < k < n_{tot}$. Once k is appropriately selected, the number of regions to be formed is determined automatically.

II) Complexity. The total time complexity of MNBC based on GNDF is O(256), which is independent of the size $m \times n$ of the image $I_{m \times n}$. Because the most time-consuming part of the whole algorithm is the evaluation of kNB queries, which takes only O(256) according to (3) and (4).

4.3 Real Images and Segmented Results

In this subsection, we present three real image experiments to show that MNBC based on GNDF outperforms MST in terms of segmented quality as well as computation time.

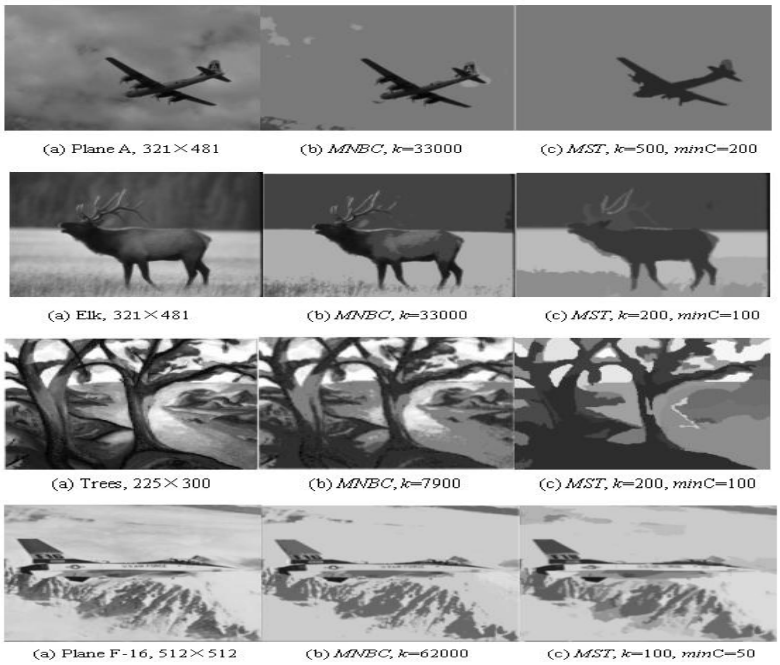

Fig. 4. Segmented Results

Figure 4 shows the segmentation results for three real world images, namely, "Plane A", "Elk", "Trees" and "Plane F-16". Each of them presents different level of difficulties in image segmentation. From left to right, the three columns correspond to respectively the original images, segmented images based on MNBC and MST. The segmentation results are shown with different gray levels representing different segmenting regions. It can be seen that MNBC visually outperforms MST. On one hand, MNBC is capable of preserving details well, such as (1) the letter "A" in the image "Plane A"; (2) "F-16' mark, the entrance with shape "□", the star signature, the text "US.ATR.FORCE" and ID # "01568" in the image "Plane F-16", whereas MST completely fails. On the other hand, although MST segments the sky correctly as a whole for the image "Plane A", the plane A is under-segmented. For the image "Elk", MST succeeds in segmenting the body of elk correctly as a whole except for the antler, but the background is over-segmented. For the image "Trees", MST has the branches of the trees merged with the riverbank.

5 Conclusion

This paper first presents a MNBC algorithm using the directed tree, which not only can discover clusters of arbitrary shape and different densities like NBC [1]

but also simplify NBC greatly. MNBC represents a dataset as some directed trees corresponding to meaningful clusters with just three key definitions refined from NBC. Second, GNDF is defined to make MNBC suitable for image segmentation. Taking all grays in an image as the dataset to be segmented, MNBC has the computational complexity O(256), which is independent of the size of the image. The experiments on synthetic datasets and real-world images shows that MNBC outperforms some existing graph-theoretical approaches in terms of computation time as well as segmentation effect. Our future work will include incorporating the spatial information to MNBC for more effective image segmentation and exploring various applications to which MNBC can be applied.

References

1. Zhou, S., Zhao, Y., Guan, J. and Huang, J.: A Neighborhood-Based Clustering Algorithm. PAKDD 2005, LNAI 3518 (1982) 361-371
2. Ester M., Kriegel H., Sander J. and Xu X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In Proc. KDD96 (1996), Portland, Oregon, 226-231 .
3. Weber, R., Schek, H. and Blott,S.: A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In Proc. of VLDB98 (1998) Aug. New York City, NY, 194-205.
4. Shi, J. and Malik, J.: Normalized Cuts and Image Segmentation. Proc. of IEEE Conf. on CVPR (1997) 731-737.
5. Shi, J. and Malik, J.: Normalized Cuts and Image Segmentation. IEEE Trans. on PAMI **22(8)** (2000) 888-905.
6. Koontz, W., Narendra, P. and Fukunaga, K.: A Graph-Theoretic Approach to Nonparametric Cluster Analysis. IEEE Trans. on Comp. **C-25(9)** (1976) Sep. 936-944.
7. Felzenszwalb, P. and Huttenlocher, D.: Image segmentation using local variation. Proc. of IEEE Conf. on CVPR (1998) 98-104.
8. Felzenszwalb, P. and Huttenlocher, D.: Efficient Graph-Based Image Segmentation. International Journal of Computer Vision **59(2)** (2004) Sep.
9. Reinhard, D.: Graph Theory. Electronic Edition (2005) Springer-Verlag Heidelberg, New York.