Katrutsa A.M., Kuznetsov M.P., Rudakov K.V., Strijov V.V. Metric concentration search procedure using reduced matrix of pairwise distances // Intelligent Data Analysis, 2015, 19(5) : 1091-1108.

Metric concentration search procedure using reduced matrix of pairwise distances

A. M. Katrutsa[1], M. P. Kuznetsov[1], V. V. Strijov[1,2,3], and K. V. Rudakov[1,2]

[1] *Moscow Institute of Physics and Technology, Institutskiy lane 9, Dolgoprudny city, Moscow region, 141700, Russia*

[2] *Dorodnicyn Computing Center of Russian Academy of Sciences, Vavilov st. 40, 119333 Moscow, Russia*

[3] *Corresponding author. Tel: +7 (499) 135 41 63, fax: +7 (499) 783 33 27, e-mail: strijov@ccas.ru*

Abstract

This paper presents a new fast clustering algorithm RhoNet, based on the metric concenration location procedure. To locate the metric concentration, the algorithm uses a reduced matrix of pairwise ranks distances. The key feature of the proposed algorithm is that it doesn't need the exhaustive matrix of pairwise distances. This feature reduces computational complexity. It is designed to solve the protein secondary structure recognition problem. The computational experiment collects tests and to hold performance analysis and analysis of dependency for the algorithm quality and structure parameters. The algorithm is compared with k-modes and tested on different metrics and data sets.

Keywords: categorical data, ρ-net, metric concentration, pairwise distances matrix, fast clustering.

1. Introduction

This paper investigates the problem of the large data sets clustering in metric spaces. We propose a clustering method based on the *metric concentration* search procedure. A metric concentration is a dense subset of the given data set in metric space. This subset must satisfy the following condition: it consists of as many elements as possible while the inner cluster distance is relatively small.

Currently the most common clustering algorithm based on the notion of density is DBSCAN (Density-based spatial clustering of applications with noise) [10] and its extensions [11, 6]. The DBSCAN cluster consists of the elements from the neighborhood of the centers of the density-connected metric balls. The important problem is to find a metric ball containing the maximum elements inside. The required dense subset is the maximum union of the such balls. The CURD algorithm [11] develops the idea of DBSCAN and constructs a graph with the cores in its vertices. If the distance between two cores is less than some distance threshold then the vertices corresponding to these cores are linked by an edge. After that, the graph is partitioned on the set of disconnected subgraphs forming the clusters. The paper [6] presents an agglomerative hierarchical clustering algorithm ROCK, based on the notions of neighbors and links. Desirable clustering structure is obtained by merging clusters with the common links or neighbors.

As an application of the metric concentration search method we consider the problem of the frequency dictionary construction of the amino acid residues of the protein primary structure. The chains of the amino acid residues database consists of the 11 million records, the length of every record is 20–33000 symbols [3, 4]. That size of the given data leads to the restrictions on the algorithm complexity. In particular, it becomes impossible to compute all distances between every pair of elements from the data set.

To reduce the computational and memory complexity we use a vantage points idea [14].

According to this idea, the distance function is computed between all elements in data set and elements of a small subset further called *ρ-net*. To estimate the distance between any pair of elements of the data set we use a distance between this pair of elements and elements of the ρ-net. For a good approximation of the required distance the distance function should satisfy triangle inequality. This approach allows to reduce computational complexity of the proposal algorithm. Therefore, the proposed metric concentration search algorithm consists of the following steps:

1) define a metric function for elements in a data set;

2) define the ρ-net subset;

3) compute the distances between elements of the ρ-net and elements of the data set;

4) find metric concentration as the maximum intersection of the relative neighbors.

The fourth step is carried through the search of the joint neighbors of ρ-net elements. To search the nearest neighbors the distances to an element of the ρ-net are sorted.

The proposed algorithm is compared with k-means [7, 9] algorithms, adapted for solving fast clusterization problem in linear metric spaces. Two stages k-means algorithm [13, 12] composed of the fast and slow stages. The fast stage computes expected clusters centers from the subset of the given data set. The size of the subset is much less than the total size of the data set. The slow stage computes the distances between all elements and all centers obtained on the fast stage. The k-modes algorithm [8] is a modification of the k-means algorithm for the categorical data. The modification of the k-means algorithm uses different dissimilarity measures and replaces means with mode values.

The computational experiment shows the performance of the considered algorithms on the different quality measures and the dependence quality of the proposal algorithm on the structure parameters. Experiments are carried out on the synthetic data sets and the real data

set from the repository UCI [5].

2. The metric concentration search problem

Consider a set of elements $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ in a metric space with given metric ρ. Suppose the set $X(\mathbf{c}|r)$ contains an element $\mathbf{x}_i \in X$ such that $\mathbf{x}_i$ belongs to a metric ball of radius r centered at $\mathbf{c} \in X$,

$$X(\mathbf{c}|r) = \{\mathbf{x}_i \in X |\ \rho(\mathbf{x}_i, \mathbf{c}) \leq r\}.$$

Definition 1. *The metric concentration is a set $C(r) = X(\hat{\mathbf{c}}|r)$ such that the metric ball of radius r centered at the element $\hat{\mathbf{c}} \in X$ contains maximum number of elements in the set X,*

$$\hat{\mathbf{c}}(r) = \arg\max_{\mathbf{c} \in X} |X(\mathbf{c}|r)|, \tag{1}$$

where $\hat{\mathbf{c}}(r)$ is called the center of the metric concentration.

Note that the metric concentration location problem differs from the standart clustering problem. The standart clustering problem is to find a cluster with the minimum distance between elements, but the problem (1) is to find a set of some fixed radius with maximum cardinality.

The triangle inequality and the metric concentration search problem solving. To reduce the computational complexity and the required memory we propose to compute a rectangular matrix of distances between the elemenets from X and the elements from a small subset $R \subset X$ instead of computing the exhaustive pairwise distances matrix.

Definition 2. *Consider an element $\mathbf{z} \in X$. The $\mathbf{z}$-relative distance between elements $\mathbf{x}, \mathbf{y} \in X$ is callled the distance $\rho_{\mathbf{z}}(\mathbf{x}, \mathbf{y})$ such that*

$$\rho_{\mathbf{z}}(\mathbf{x}, \mathbf{y}) = |\rho(\mathbf{z}, \mathbf{x}) - \rho(\mathbf{z}, \mathbf{y})|. \tag{2}$$

From the triangle inequality it follows the important fact such that:

$$\rho_{\mathbf{z}}(\mathbf{x},\mathbf{y}) \le \rho(\mathbf{x},\mathbf{y}) \quad \forall \mathbf{z} \in X. \tag{3}$$

The inequality (3) means that for any element $\mathbf{z} \in X$ the $\mathbf{z}$-relative distance $\rho_{\mathbf{z}}(\mathbf{x},\mathbf{y})$ is the lower bound for a given distance $\rho(\mathbf{x},\mathbf{y})$. Introduce ρ-net, a set of elements $R \subset X$ further called *ρ-net*. Denote by $\rho_R(\mathbf{x},\mathbf{y})$ an R-relative distance between elements $\mathbf{x},\mathbf{y} \in X$ such that $\rho_R(\mathbf{x},\mathbf{y})$ is the maximum $\mathbf{z}$-relative distance over all elements from the set R:

$$\rho_R(\mathbf{x},\mathbf{y}) = \max_{\mathbf{z}\in R} \rho_{\mathbf{z}}(\mathbf{x},\mathbf{y}).$$

Note that the X-relative distance equals some given distance $\rho(\mathbf{x},\mathbf{y})$:

$$\rho_X(\mathbf{x},\mathbf{y}) = \rho(\mathbf{x},\mathbf{y}).$$

To construct a fast clustering algorithm we must evaluate these pairwise distances and construct a sparse set

$$R \subset X, \text{where } |R| \ll |X|,$$

such that an R-relative distance for every pair of elements is an upper bound of given distance $\rho(\mathbf{x},\mathbf{y})$ with the constant c_0,

$$\rho_R(\mathbf{x},\mathbf{y}) \ge c_0\rho(\mathbf{x},\mathbf{y}), \quad \forall \mathbf{x},\mathbf{y} \in X. \tag{4}$$

The constant c_0 and the inequality (4) are needed to prove the feasibility of the approach proposed below. After here we omit this constant. From the inequality (4) it follows that a sparse set $R \subset X$ is sufficient to approximate distances between all elements in X with some accurancy. Note that if (4) is satisfied, then it is possible to find approximate solution of the problem (1) in the form

$$\hat{\mathbf{c}}(r) = \arg\max_{\mathbf{c}\in X} \left| \bigcap_{\mathbf{z}\in R} X(\mathbf{z},\mathbf{c}|r) \right|, \tag{5}$$

where

$$X(\mathbf{z},\mathbf{c}|r) = \{\mathbf{x}_i \in X| \quad \rho_{\mathbf{z}}(\mathbf{x}_i,\mathbf{c}) \le r\} \tag{6}$$

is the $\mathbf{z}$-relative metric ball for the element $\mathbf{z} \in R$ of radius r centered at the element $\mathbf{c}$.

Problem statement for the metric concentration location search problem. Reformulate the problem (5) using given number of nearest neigbours of the center of a metric concentration instead of given radius r. This formulation allows to develop the constructive algorithm to solve the problem (1). Let the parameter $w \in \mathbb{N}$ be the number of the considering nearest neighbours. Denote by $X(\mathbf{z}, \mathbf{c}|w)$ a set containing first w elements in the ascending ordered $\mathbf{z}$-relative distance array, where $\mathbf{z} \in R$. Call an estimation of the metric concentration the intersection of the sets $X(\mathbf{z}, \mathbf{c}|w)$ for all elements $\mathbf{z}$ of the set R:

$$\hat{C}(w) = \bigcap_{\mathbf{z} \in R} X(\mathbf{z}, \hat{\mathbf{c}}|w), \qquad \text{where } \hat{\mathbf{c}}(w) = \arg\max_{\mathbf{c}} \left| \bigcap_{\mathbf{z} \in R} X(\mathbf{z}, \mathbf{c}|w) \right|. \tag{7}$$

Efficiency of this estimation follows from next claim. If two elements are $\mathbf{z}$-relative adjacent, then they are adjacent in the metric ρ for all $\mathbf{z} \in R$. Note that using distance ranks instead of linear distance values makes this approach more stable.

3. The metric concentrations location procedure

To solve the problem (7) we propose the following procedure to locate the metric concentration.

Construct the ρ-net R as a subset of X. Suppose the ρ-net

$$R = \{\mathbf{x}_j | j \in I\}$$

satisfies inequality (4). The ρ-net R is an inner subset of X, $R \subset X$ with some fixed cardinality n. Construct the set R such that the distance between its nearest elements is maximum. By assumption, number of points $N \gg n$. To construct the set R use the following iterative procedure.

1. Let the initial set be empty, $R = \emptyset$.

2. For some given element $\mathbf{y} \in X$ select

$$\mathbf{x}' = \arg\max_{\mathbf{x} \in X} \rho(\mathbf{x}, \mathbf{y});$$

then add the element $\mathbf{x}'$ to the set $R := R \cup \mathbf{x}'$.

3. While $|R| < n$ compute

$$\mathbf{x}' = \arg\max_{\mathbf{x}\in X} \min_{\mathbf{z}\in R} \rho(\mathbf{x}, \mathbf{z}),$$

add element $\mathbf{x}'$ to the set $R := R \cup \mathbf{x}'$.

The complexity of this procedure is $O(n^2 N)$, where $n^2 \ll N$, it is linear to the number of objects.

Estimate the metric concentration. To find the set $\hat{C}(w)$ from (7), construct the reduced $(n \times N)$-matrix $\mathbf{D}$ of pairwise distances between elements $\mathbf{z}_j$ of the ρ-net R and all elements $\mathbf{x}_i$ of the set X:

$$\mathbf{D}(j, i) = \rho(\mathbf{z}_j, \mathbf{x}_i), \qquad \mathbf{z}_j \in R, \qquad \mathbf{x}_i \in X,$$

where j is the index of an element from the ρ-net R, i is the index of an element from the set X. Denote by $\mathbf{D}'$ a matrix such that j-th row of the matrix $\mathbf{D}'$ consists of the indexes of the elements belonging to the set X and not belonging to the set R. These row elements of $\mathbf{D}'$ are sorted in increasing order. For the sorted matrix $\mathbf{D}'$ locate the metric concentration:

$$\hat{C}(w) = \bigcap_{\mathbf{z}_j \in R} X(\mathbf{z}_j, \hat{\mathbf{c}}|w). \tag{8}$$

We propose the following three-step procedure to locate the metric concentration (8).

1. For any element $\mathbf{x}_i \in X$ find a subset $X(\mathbf{z}_j, \mathbf{x}_i|w)$ from (6) as a $\mathbf{z}_j$-relative set consisting of w nearest neighbours for all elements $\mathbf{z}_j \in R$. For fast nearest neighbours search the j-th row of the matrix $\mathbf{D}'$ is used.

2. For any element $\mathbf{x}_i \in X$ find the set

$$\bigcap_{\mathbf{z}_j \in R} X(\mathbf{z}_j, \mathbf{x}_i|w)$$

as the intersection of the $\mathbf{z}_j$-relative nearest neighbour sets for all $\mathbf{z}_j \in R$.

3. Find the center $\hat{\mathbf{c}}(w)$ of the metric concentration (8) as the center of the intersection with maximum cardinality,

$$\hat{\mathbf{c}}(w) = \arg\max_{\mathbf{c}\in X} \left| \bigcap_{\mathbf{z}_j \in R} X(\mathbf{z}_j, \mathbf{c}|w) \right|.$$

Clustering procedure using metric concentration In this section we describe the consecutive metric concentration estimation procedure. This procedure removes elements $\mathbf{x}_i \in \hat{C}(w)$ from the set X, where the estimation $\hat{C}(w)$ is obtained iteratively and repeates the metric concentration search. This procedure obtaines estimations of the metric concentrations $\hat{C}_j(w)$. Partition the set X

$$X = \bigsqcup_{k=0}^{K} \hat{C}_k(w),$$

where K is some given number of concentrations and $\hat{C}_0(w)$ a set of non-clustered elements $\mathbf{x}_i \in X$. Therefore, the set X is partitioned into K clusters. The clusters are the estimations of the metric concentrations $\hat{C}_1(w), ..., \hat{C}_K(w)$.

Illustrate the proposed algorithm on the Fig. 1. Here the elements of the set X are the points in the 2-dimensional euclidean space. By A, B, C denote the points of the ρ-net. The cluster is the set of points belonging to the intersection of the rings formed by the circles centered at the points of the ρ-net. The radius of the smaller circle, centered at the point A, equals 0. The circles corresponding to the point C haven't been shown because the ring they formed fully include concentration shown by cross. The dash circle is centered at the point A. The solid circles are centered at the point B. The cluster, emphasized by crosses, is formed by the intersection of 3 rings, centered at the points of the ρ-net.

Optimization of parameters. The algorithm includes two parameters: w is the number of neighbours and n is the number of the elements of the ρ-net. The parameter w is chosen according to the following assumptions. Introduce the *concentration density* κ as the ratio of

the number of neighbours w to the size of the data set N:

$$\kappa = w/N.$$

The concentration density approximately equals the ratio of the metric concentration radius r to the diameter of the data set, $\max\limits_{\mathbf{x},\mathbf{y}\in X} \rho(\mathbf{x},\mathbf{y})$:

$$\kappa \approx \frac{r}{\max\limits_{\mathbf{x},\mathbf{y}\in X} \rho(\mathbf{x},\mathbf{y})}.$$

That is, the relation between the number of the neighbours w and the metric concentration radius r is proportional to the concentration density κ.

4. Algorithm k-modes

The basic algorithm k-modes to compare with is presented in [8]. This algorithm extends the k-means principle to categorical data using some metric for categorical data instead of euclidean distance. In [8] the metric is similar to the metric Overlap (11) with the weights $p_s = 1$ and $s = 1,\ldots,d$. Further, we will use this metric, except for some cases.

In this case and further the set of elements $X = \{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$ is the set of words $\mathbf{x}_i,\ i = 1,\ldots,N$, each word $\mathbf{x}_i$ is the set of letters. Denote by x_{is} the s-th letter in the word $\mathbf{x}_i$. The set X and the number of clusters K are given. One must partition the set X: $X = Y_1 \sqcup \ldots \sqcup Y_K$, where Y_k consists of the elements from k-th cluster.

Denote by Y the set of centers, $Y = \{\mathbf{y}_k,\ k = 1,\ldots,K\}$. The center of the cluster Y_k is the word $\mathbf{y}_k$ such that the every letter y_{ks} is the most frequent s-th letter among all words belonging to the set Y_k. The clustering procedure starts with the initialization of the cluster centers $\mathbf{y}_k,\ k = 1,\ldots,K$.

In our implemetation of the k-modes algorithm the first K distinct words are selected as the initial K centers of the clusters. Suppose that the order of the words $\mathbf{x}_i$ is random and

the first K words belong to the different clusters. This initializaition method is proposed in [8] and used in the performance analysis.

The alternative method of initializaition of the centers $\mathbf{y}_k$ selects the most distant words from X.

The sets $Y_1^t \dots Y_K^t$ partition the set X at every iteration t. Suppose the clusters remain unchanged if for $k = 1, \dots, K$ there exists the bijection

$$g : Y_k^t \to Y_k^{t+1}, \tag{9}$$

where Y_k^t is the k-th cluster at t-th iteration. Execute the following steps, while this mapping doesn't exist for all clusters:

1. At t-th iteration allocate every word $\mathbf{x}_i$ to the cluster Y_k^t if the distance between this word $\mathbf{x}_i$ and the center $\mathbf{y}_k$ is the smallest, i.e.

$$\mathbf{x}_i \in Y_k^t, \quad k = \arg \min_{k=1,\dots,K} \rho(\mathbf{x}_i, \mathbf{y}_k).$$

2. Update the centers $\mathbf{y}_k^t$ of the clusters Y_k^t according to the current partition of the set X in the following way. The letter y_{ks}^t equals the most frequent s-th letter among all elements belonging to the cluster Y_k^t:

$$y_{ks}^t = \arg \max_{l \in Q_s} \sum_{\mathbf{x} \in Y_k^t} [x_s = l],$$

where Q_s is the set of possible s-th letters, x_s is the s-th letter of the word $\mathbf{x}$. The indicator function $[x_s = l]$ equals one iff the equality $x_s = l$ is true and equals zero otherwise.

The centers $\mathbf{y}_k$ may not belong to the set X after update. As soon as the clusters are unchanged, i.e. there exists the mapping (9), the clustering procedure is finished.

5. Distance functions for categorical data

Consider two words $\mathbf{x} = [x_1, ..., x_d]^\intercal$ and $\mathbf{y} = [y_1, ..., y_d]^\intercal$. A distance function $\rho(\mathbf{x}, \mathbf{y})$ between words $\mathbf{x}, \mathbf{y}$ must be a metric. Therefore, the following statements must be satisfied:

1) identity: $\rho(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$;

2) symmetry: $\rho(\mathbf{x}, \mathbf{y}) = \rho(\mathbf{y}, \mathbf{x})$;

3) triangle inequalitiy: $\rho(\mathbf{x}, \mathbf{z}) \leqslant \rho(\mathbf{x}, \mathbf{y}) + \rho(\mathbf{y}, \mathbf{z})$.

Define some metrics using in proposal approach.

Ordered (SO) and unordered (SU) symmetric defference of two sets Define this distance function between words $\mathbf{x}$ and $\mathbf{y}$ as

$$\rho(\mathbf{x}, \mathbf{y}) = \frac{|\mathbf{x}| + |\mathbf{y}| - 2S(\mathbf{x}, \mathbf{y})}{|\mathbf{x}| + |\mathbf{y}| - S(\mathbf{x}, \mathbf{y})}. \tag{10}$$

In case of the unordered symmetric defference the function $S(\mathbf{x}, \mathbf{y})$ is the cardinality of the intersection of the words $\mathbf{x}$ and $\mathbf{y}$. By $|\cdot|$ denote the set cardinality, in this case it equals the size of the word.

In case of the ordered symmetric defference the function $S(\mathbf{x}, \mathbf{y})$ is the size of the longest common subsequence in words $\mathbf{x}$ and $\mathbf{y}$. The cardinality of the longest common subsequence equals the length of *the cheapest diagonal way* defining in the next paragraph. The matrices of the pairwise distances for these metrics are shown in Fig. 2, 3.

Optimal alignment (OA) To compute this function find the optimal alignment between the ordered elements of two words. The distance between two letters is the boolean function ω:

$$\omega(x_i, x_j) = \begin{cases} 1, & \text{if } x_i \neq y_j, \\ 0, & \text{otherwise.} \end{cases}$$

To compute the distance between words find the *cost matrix* $P(d+1 \times d+1)$. Denote the index of the first row by $i = 0$, and the index of the first column by $j = 0$. Assign $P(0, 0) = 0$. For all $i = 1, \ldots, d$ and $j = 1, \ldots, d$ assign $P(0, j) = P(i, 0) = \infty$. For all $i = 1, \ldots, d$ and $j = 1, \ldots, d$ we sequentially compute all elements of the matrix P:

$$P(i, j) = \omega(x_i, y_j) + \min\big(P(i-1, j-1), P(i-1, j), P(i, j-1)\big).$$

The distance between words is the element in d-th row and d-th column of the matrix P:

$$\rho(\mathbf{x}, \mathbf{y}) = P(d, d).$$

Note that this distance function is a metric and the case of the Levenstein's distance.

The matrix of pairwise distances for $\rho(\mathbf{x}, \mathbf{y})$, is shown in the Fig. 4. The cost matrix P for the optimal alignment is shown in the Fig. 5 with $d = 8$. The cheapest way is shown by dots. The start and the end of this way is fixed in elements with indices $(0, 0)$ and $(7, 7)$.

Weighted distance between words Define the similarity measure between two words $\mathbf{x}, \mathbf{y}$ as the weighted sum of the similarities between their letters:

$$\text{sim}(\mathbf{x}, \mathbf{y}) = \sum_{s=1}^{d} h_s \text{sim}_s(x_s, y_s),$$

where h_s is the weigth assigned the s-th letter, x_s, y_s is s-th letter in words $\mathbf{x}, \mathbf{y}$, d is the words length. Define the distance function as:

$$\rho(\mathbf{x}, \mathbf{y}) = 1 - \frac{1}{\text{sim}(\mathbf{x}, \mathbf{y})}.$$

To find the distance $\rho(\mathbf{x}, \mathbf{y})$: 1) find the similarities $\text{sim}_s(x_s, y_s)$ between all letters in words $\mathbf{x}, \mathbf{y}$; 2) define the weigths assisgned every letters.

Consider the matrix M of the pairwise distances between all possible values of the s-th letter. The 0-th row and column consist of the possible values taken the s-th letter. The element $M(i, j)$ equals the similarity between the i-th and the j-th possible values taken the s-th letter, $i, j = 1, \ldots, |Q_s|$, where $|Q_s|$ is the number of the possible values taken the s-th letter.

Test the proposal algorithm with three types of similarity functions, according to usage of elements of the matrix M.

1. Similarity function uses only diagonal elements of the matrix M:

$$\mathrm{sim}_s(x_s, y_s) = \begin{cases} p, & \text{if } x_s = y_s, \\ 0, & \text{otherwise,} \end{cases} \tag{11}$$

where p shows the dependence between the similarity s-th letters in the words $\mathbf{x}, \mathbf{y}$ and similarity corresponding words. The weights $h_s = \frac{1}{d}$ for all letters.

For distance function Overlap the parameter $p = 1$ and the weights $h_s = \frac{1}{d},\ s = 1, \ldots, d$.

2. Similarity function uses only non-diagonal elements of the matrix M:

$$\mathrm{sim}_s(x_s, y_s) = \begin{cases} 1, & \text{if } x_s = y_s, \\ q, & \text{otherwise,} \end{cases}$$

where q shows the dissimilarity between the s-th letter in the words $\mathbf{x}, \mathbf{y}$ and dissimilarity of the corresponding words. The weights $h_s = \frac{1}{d}$ are similar to the previous case.

In particular, for the distance function Goodall1

$$q = 1 - \sum_{l \in Q_s} \tilde{p}_s(l), \ \ \tilde{p}_s(x) = \frac{f_s(x)(f_s(x) - 1)}{N(N-1)},$$

where N is the size of the data set, Q_s is the set of the possible values of s-th letter and $f_s(x)$ is the number of times s-th letter takes a value x in the data set. If s-th letter doesn't take the value x, then $f_s(x) = 0$. The weights $h_s = \frac{1}{d},\ s = 1, \ldots, d$.

3. Similarity function uses both diagonal and non-diagonal elements of the matrix M:

$$\mathrm{sim}_s(x_s, y_s) = \begin{cases} p, & \text{if } x_s = y_s, \\ q, & \text{otherwise.} \end{cases}$$

Equalities for p, q and weights based on information-theoretic framework for similarities.

In particular, for the distance function Lin1:

$$p = \sum_{l \in Q} \log \hat{p}_s(l),$$

$$q = 2 \sum_{l \in Q} \log \hat{p}_s(l),$$

where $\hat{p}_s(x) = f_s(x)N^{-1}$, other notations are similar to the previous case. The weights $h_s = \left(\sum_{s=1}^{d} \sum_{l \in Q} \log \hat{p}_s(l) \right)^{-1}$.

6. Analysis of the complexity and required memory

The proposed RhoNet algorithm is compared with the algorithms from [6, 11, 10, 8] based on two criteria: complexity and required memory. Table 1 shows the algorithm complexity and the required memory dependence on the size of a data set, the size of a ρ-net and on the parameters m_a, m_m, m, i, k of the algorithms from [6, 11, 10, 8]. Note that the required memory for RhoNet algorithm increases sublinearly with increasing size of the data set in contrast to the other algorithms.

7. Computational experiments and performance analysis

The proposed RhoNet algorithm is tested on the synthetic data sets and the data set Mushroom [2] from the repository UCI [5]. The synthetic data set is generated by the procedure describing below.

Real data set description. The mushroom data set from the repository UCI consists of 8124 objects, described by 22 letters. Every object can be represented as a word. One has to a cluster the data set for poisonous and edible mushroom. The cluster for object is known, so it is possible to use clustering error E_k for cluster k.

Synthetic data sets description. The synthetic data sets were generated according to the following parameters. An $m \times u$ matrix A consists of every possible letters, where m is the number of letters, u is the number of possible values taken by every letter, the size of the

data set N, the number of the clusters K, the distance function ρ and the maximum distance between two generated objects

$$\max_{\text{dist}} = \max_{\mathbf{x}_i, \mathbf{x}_j \in X} \rho(\mathbf{x}_i, \mathbf{x}_j);$$

the portion $\text{var}_{\text{cent}} \in (0; 1)$ of the maximum distance $\max_{\text{dist}}$ equals the minimum distance between the centers $\mathbf{y}_k, k = 1, \ldots, K$ of the generated clusters

$$\rho(\mathbf{y}_i, \mathbf{y}_j) \geq \text{var}_{\text{cent}} \cdot \max_{\text{dist}}, \ i, j = 1, \ldots, K;$$

the distance variation var_{obj} between every next word from the current

$$\rho(\mathbf{x}_i, \mathbf{x}_{i-1}) < \text{var}_{\text{obj}} \cdot \max_{\text{dist}}, \ i = 1, \ldots, N.$$

The data generation procedure *DataGen* executes two stages: the first stage is generation of centers $\mathbf{y}_k, k = 1, \ldots, K$ for every cluster, the second stage is generation of the words for every clusters. To create the centers the values from the matrix A are selected randomly. The centers are created if the distance between every pair of centers is more than $\text{var}_{\text{cent}} \cdot \max_{\text{dist}}$:

$$\rho(\mathbf{y}_j, \mathbf{y}_i) > \text{var}_{\text{cent}} \cdot \max_{\text{dist}}, \ i \neq j; \ i, j = 1, \ldots, K.$$

Denote the number of words in every cluster by $N_k = \lfloor N/K \rfloor, \ k = 1, \ldots, K$. Suppose the difference in the number of words between clusters is no more than 1. After that, the $N_k \times m$ submatrix X_k consists of the words belonging to the k-th cluster is created. Every matrix is initialized with N_k copies of the cluster center. Further, the number of the letters in every copy, except the first, is changed at random. Suppose the word $\mathbf{x}_i$, $i = 2, \ldots, N_k$ from the k-th cluster is created if it satisfies all following conditions:

it doesn't equal the center of the k-th cluster;

it doesn't equal the word $\mathbf{x}_{i-1}$;

the distance between the word $\mathbf{x}_i$ and $\mathbf{x}_{i-1}$ is less than $\text{var}_{\text{obj}} \cdot \max_{\text{dist}}$, i.e. $\rho(\mathbf{x}_i, \mathbf{x}_{i-1}) < \text{var}_{\text{obj}} \cdot \max_{\text{dist}}$.

This procedure executes for every cluster. After that, the rows of the matrix X concatenates the rows of the matrices $X_k,\ k = 1, \ldots, K$, Finally, these rows are randomly permutated.

Quality measures. We use the following quality measures to compare the algorithms. By E_k denote the number of clustering error for k-th cluster, if the cluster for every element is given:

$$E_k = \frac{\sum\limits_{i=1}^{N_k}[k_i \neq a_i]}{\sum\limits_{i=1}^{N}[k = a_i]},$$

where k_i is given cluster for the i-th element, a_i is a cluster for the i-th element defining by algorithm, N_k is the size of the k-th cluster defining by algorithm. By F_1 denote the cluster mean external distance:

$$F_1 = \frac{\sum\limits_{i<j}[a_i \neq a_j]\rho(\mathbf{x}_i, \mathbf{x}_j)}{\sum\limits_{i<j}[a_i \neq a_j]},$$

where $\rho(\mathbf{x}_i, \mathbf{x}_j)$ is given distance function. By F_0 denote the cluster mean inner distance:

$$F_0 = \frac{\sum\limits_{i<j}[a_i = a_j]\rho(\mathbf{x}_i, \mathbf{x}_j)}{\sum\limits_{i<j}[a_i = a_j]},$$

where $\rho(\mathbf{x}_i, \mathbf{x}_j)$ is given distance function.

Comparison of the algorithms. Table 2 shows the value E_k for poisonous (Po) and edible (Ed) mushroom. The values of quality measures F_0 and F_1 are shown in Table 3, which is similar to the Table 2.

The synthetic data set was generated for every distance function with the following parameters: $N = 200$, $m = 6$, $K = 4$, $\mathrm{var}_{\mathrm{cent}} = 1$, $\mathrm{var}_{\mathrm{obj}} = 0.4$ and the maximum distance $\max_{\mathrm{dist}}$, corresponding given distance function. To generate these data sets the algorithm describing early is used. The quality measures using for compare are the mean inner and external cluster distances $F_0,\ F_1$. For the RhoNet algorithm the part of non-clustered elements is computed additionally. The RhoNet and k-modes algorithms are run on the same data sets corresponding every given distance function. Next in this section we use the distance function Overlap (11).

The results of applying k-modes algorithm to the generated data sets are shown in Table 4, the number of clusters K used in the algorithm equals the number of generated clusters. There are using metrics in the columns. The quality measures are F_0 and F_1.

Because of the strong dependence the RhoNet algorithm on the parameters, the results of applying it to the generated sets can't be shown in the table like the results of the k-modes algorithm. To demostrate the results applying the RhoNet algorithm to the data sets the iterative procedure is designed. By i denote the number of iteration corresponding to the fixed pair of the parameters: the number of the elements in the ρ-net and the concentration desity κ. The number of the elements in the ρ-net is changed from 2 to 9 with step equals 1. The concentration density is changed from 0.1 to 0.9 with step equals 0.05 ($b = 17$). The results are shown in Fig. 6, 7, 8, 9. The equalities for defining parameters n and κ from the iteration number i are follows: $n = \left\lfloor \frac{i}{b} \right\rfloor + 2$, $\kappa = 0.1 + 0.05 \cdot (\mathrm{mod}(i, (b+1)) - 1)$.

7.1. Properties the RhoNet algorithm

To test RhoNet algorithm on the synthetic data sets for every distance function, seven data sets are generated with the following parameters: $m = 6$, $N = 200$, $K = 4$, $\mathrm{var}_{\mathrm{cent}} = 1$, $\mathrm{var}_{\mathrm{obj}} = 0.4$ and maximal distance $\mathrm{max}_{\mathrm{dist}}$, corresponding given distance function. All data sets are generated using procedure *DataGen*. For all generated data sets we obtain the dependence mean inner and external cluster distances on the size of the ρ-net n and the concentration density κ.

The forms of the dependence the mean external and inner cluster distances on the concentration density for 3 points in ρ-net and distance function Overlap are shown in the Fig. 10 and 11.

The forms of the dependence the mean external and inner cluster distances on the concentration density for 3 points in ρ-net and distance function SO are shown in the Fig. 12

and 13. Every line on the graphs corresponds to one generated data set.

The forms of the dependence the mean external and inner cluster distances on the number points in the ρ-net for concentration density equals 0.5 and distance function Overlap are shown in the Fig. 14 and 15.

The forms of the dependence the mean external and inner cluster distances on the concentration density for 3 points in ρ-net and distance function SO are shown in the Fig. 16 and 17

8. Conclusion

In this paper we propose the metric concentration search algorithm RhoNet. The key feature of the proposal algorithm is using reduced matrix of pairwise distances between the elements of the ρ-net and all elements of the data set. Consequently the required memory is $O(nN)$ rather than $O(N^2)$, where N is the size of the data set, n is the size of the ρ-net and $n \ll N$. The experiments show that quality of the proposal algorithm isn't lower than quality of the k-modes algorithm. The quality measures are mean cluster inner and external distances. The dependence from using distance function is weak. The experiments can be reproduced using the code and the dataset from [1].

References

[1] Source code for RhoNet algorithm. `http://svn.code.sf.net/p/mlalgorithms/code/RhoNetClustering/mcode`.

[2] UCI Machine Learning Repository, Mushroom Data Set. `http://archive.ics.uci.edu/ml/datasets/Mushroom`.

[3] (2011), Fasta sequence database. `http://bit.ly/sP2A26`.

[4] (2011), Fasta sequence database, example of a record. http://www.uniprot.org/uniprot/Q08753.

[5] Bache, K. and Lichman, M. (2013), UCI machine learning repository. http://archive.ics.uci.edu/ml.

[6] Guha, S., Rastogi, R., and Shim, K. (2000) ROCK: A Robust Clustering Algorithm for Categorical Attributes. *Information Systems*, **25**, 345–366.

[7] Hartigan, J. A. (1975) *Clustering Algorithms*. Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons.

[8] Huang, Z. (1997) A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining. *In Research Issues on Data Mining and Knowledge Discovery*.

[9] Jain, A. and Dubes, R. (1988) *Algorithms for clustering data*. Prentice Hall advanced reference series, Prentice Hall.

[10] Kriegel H.-P., S. J., Kröger P. and A., Z. (2011) Density-based clustering. *WIREs Data Mining and Knowledge Discovery*, **1**, 231–240.

[11] Ma, S., Wang, T., Tang, S., Yang, D., and Gao, J. (2003) A New Fast Clustering Algorithm Based on Reference and Density. *Advances in Web-Age Information Management*, pp. 214–225.

[12] Salman, R., Kecman, V., Li, Q., and Strack, R. (2011) Fast k-means algorithm clustering. *International Journal of Computer Networks & Communications (IJCNC)*, **3**.

[13] Tang, C. and Zhao, W. (2004) A New Clustering Algorithm for Categorical Attributes. *International Conference on Electronic Business*, pp. 1065–1069.

[14] Yianilos, P. N. (1993) Data structures and algorithms for nearest neighbor search in general metric spaces. *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete*

Algorithms, pp. 311–321, SODA '93, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Table 1: Algorithm comparison. There are the names of comparison algorithms in the first column. Complexity and required memory of the comparison algorithms are in the second and third columns. The comments are in the last column

Algorithm	Complexity	Required memory	Assumptions
RhoNet	$O(Kn\kappa N^2)$	$O(nN)$	$n \ll N$, $\kappa < 1$
ROCK [6]	$O(Nm_m m_a + N^2 \log N)$.	$O(\min\big(N^2, Nm_m m_a\big))$	$m_a \ll N$, $m_m \lesssim N$
CURD [11]	$O(N(m + ik))$	$O(N) + O(K)$	$k, i, m \ll N$
DBSCAN [10]	$O(N \log N)$	$O(N^2)$	
k-modes [8]	$O(NKi)$	$O(N^2)$	

Table 2: Algorithms compare through the clustering error E_k using mushroom data set from the UCI repository. There are the distance functions using in the algorithms in the columns. Comparison algorithms with their parameters between parentheses are in the rows

		Overlap	Godall1	SymmOrd	SymmUnord	Lin1	OA
RhoNet	Po	0.399	0.405	0.396	0.426	0.391	0.419
($n = 2$, $\kappa = 0.8$)	Ed	0.255	0.273	0.236	0.434	0.252	0.262
RhoNet	Po	0.518	0.479	0.528	0.541	0.549	0.402
($n = 3$, $\kappa = 0.8$)	Ed	0.482	0.435	0.491	0.503	0.508	0.342
k-modes	Po	0.435	0.437	0.223	0.324	0.434	0.381
($K = 2$)	Ed	0.419	0.427	0.202	0.346	0.427	0.393

Table 3: Algorithms compare through the quality measures F_0 and F_1 using mushroom data set from the UCI repository. There are the distance functions using in the algorithms in the columns. Comparison algorithms with their parameters between parentheses are in the rows

		Overlap	Godall1	SymmOrd	SymmUnord
RhoNet	F_1	11.52	1.53	0.897	0.53
($n = 2$,	F_0	10.53	1.45	0.863	0.48
$\kappa = 0.8$)					
RhoNet	F_1	11.50	1.52	0.896	0.52
($n = 3$,	F_0	10.46	1.42	0.861	0.49
$\kappa = 0.8$)					
k-modes	F_1	12.14	6.93	0.911	0.55
($K = 2$)	F_0	9.86	4.62	0.846	0.46

Table 4: The values of the quality measures F_0 and F_1 after test the k-modes algorithm on the generated data sets. The distance functions using in the algorithm are in the columns

		Overlap	SO	SU	OA
k-modes ($K = 4$)	F_1	0.85	0.92	0.83	2.98
	F_0	0.75	0.41	0.73	2.52

Fig. 1 illustrates the applying of the proposal algorithm to the data set in the 2-dimensional euclidean space.

Fig. 2 shows the matrix of pairwise distances for the SO metric

Fig. 3 shows the matrix of pairwise distances for the SU metric

Fig. 4 shows the matrix of pairwise distances for the DTW metric

Fig. 5 shows the cost matrix for the DTW metric. The cheapest way is shown by dots.

Fig. 6 shows possible values of considering quality measures after applying RhoNet algorithm with the Overlap distance function to the generated data set.

Fig. 7 shows possible values of considering quality measures after applying RhoNet algorithm with the SO distance function to the generated data set.

Fig. 8 shows possible values of considering quality measures after applying RhoNet algorithm with the SU distance function to the generated data set.

Fig. 9 shows possible values of considering quality measures after applying RhoNet algorithm with the DTW distance function to the generated data set.

Fig. 10 shows the form of the dependence the mean external cluster distance of the concentration density for 7 synthetic data sets with 3 points in ρ-net, distance function is Overlap.

Fig. 11 shows the form of the dependence the mean inner cluster distance of the concentration density for 7 synthetic data sets with 3 points in ρ-net, distance function is Overlap.

Fig. 12 shows the form of the dependence the mean external cluster distance of the concentration density for 7 synthetic data sets with 3 points in ρ-net, distance function is SO.

Fig. 13 shows the form of the dependence the mean inner cluster distance of the concentration density for 7 synthetic data sets with 3 points in ρ-net, distance function is SO.

Fig. 14 shows the form of the dependence the mean external cluster distance of the number points in ρ-net for 7 synthetic data sets with concentration density equals 0.5, distance function is Overlap.

Fig. 15 shows the form of the dependence the mean inner cluster distance of the number points in ρ-net for 7 synthetic data sets with concentration density equals 0.5, distance function is Overlap.

Fig. 16 shows the form of the dependence the mean external cluster distance of the number points in ρ-net for 7 synthetic data sets with concentration density equals 0.5, distance function is SO.

Fig. 17 shows the form of the dependence the mean inner cluster distance of the number points in ρ-net for 7 synthetic data sets with concentration density equals 0.5, distance function is SO.

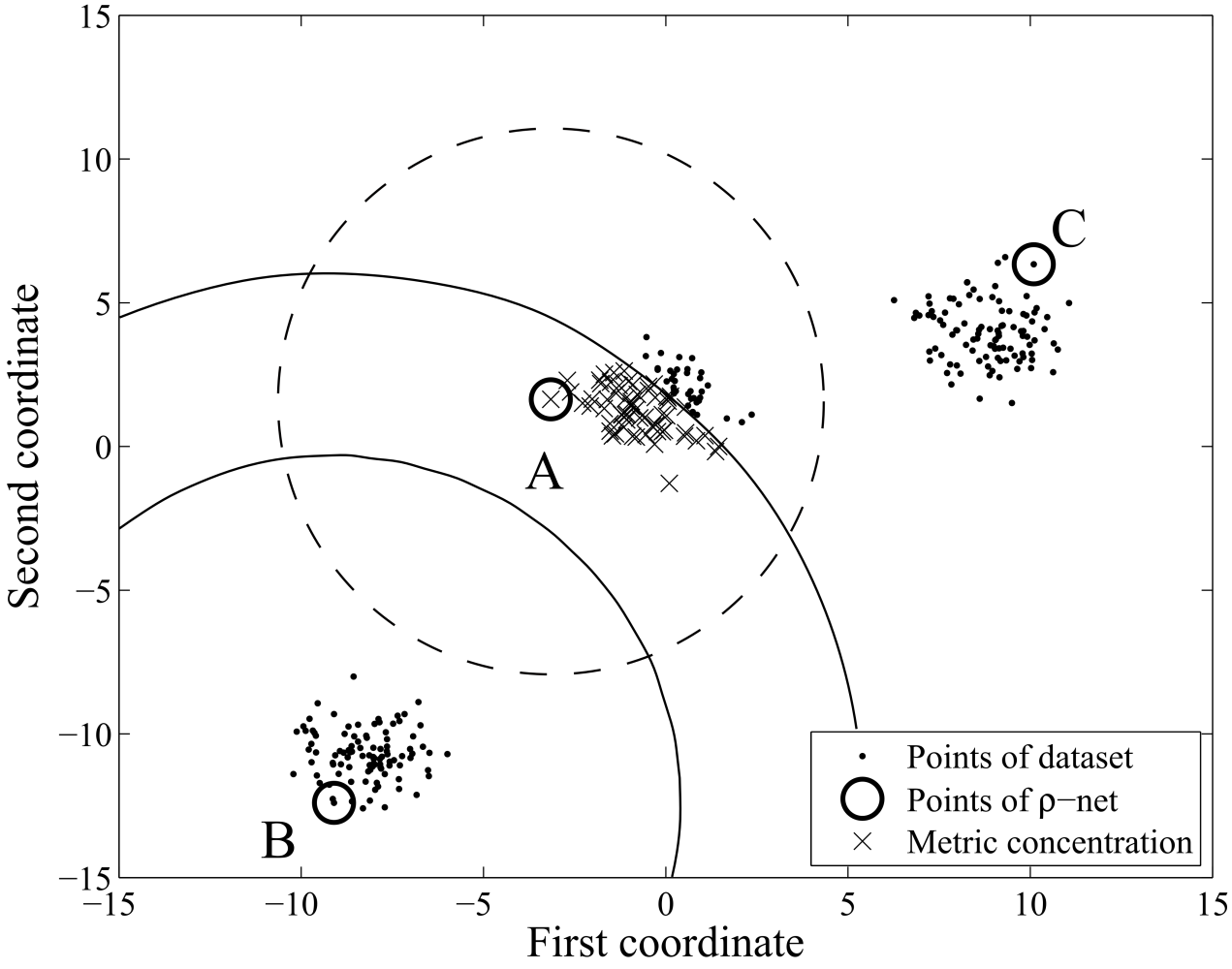
Second coordinate
First coordinate
A
B
C
Points of dataset
Points of ρ−net
Metric concentration
15
10
5
0
−5
−10
−15
−15
−10
−5
0
5
10
15

Figure 1

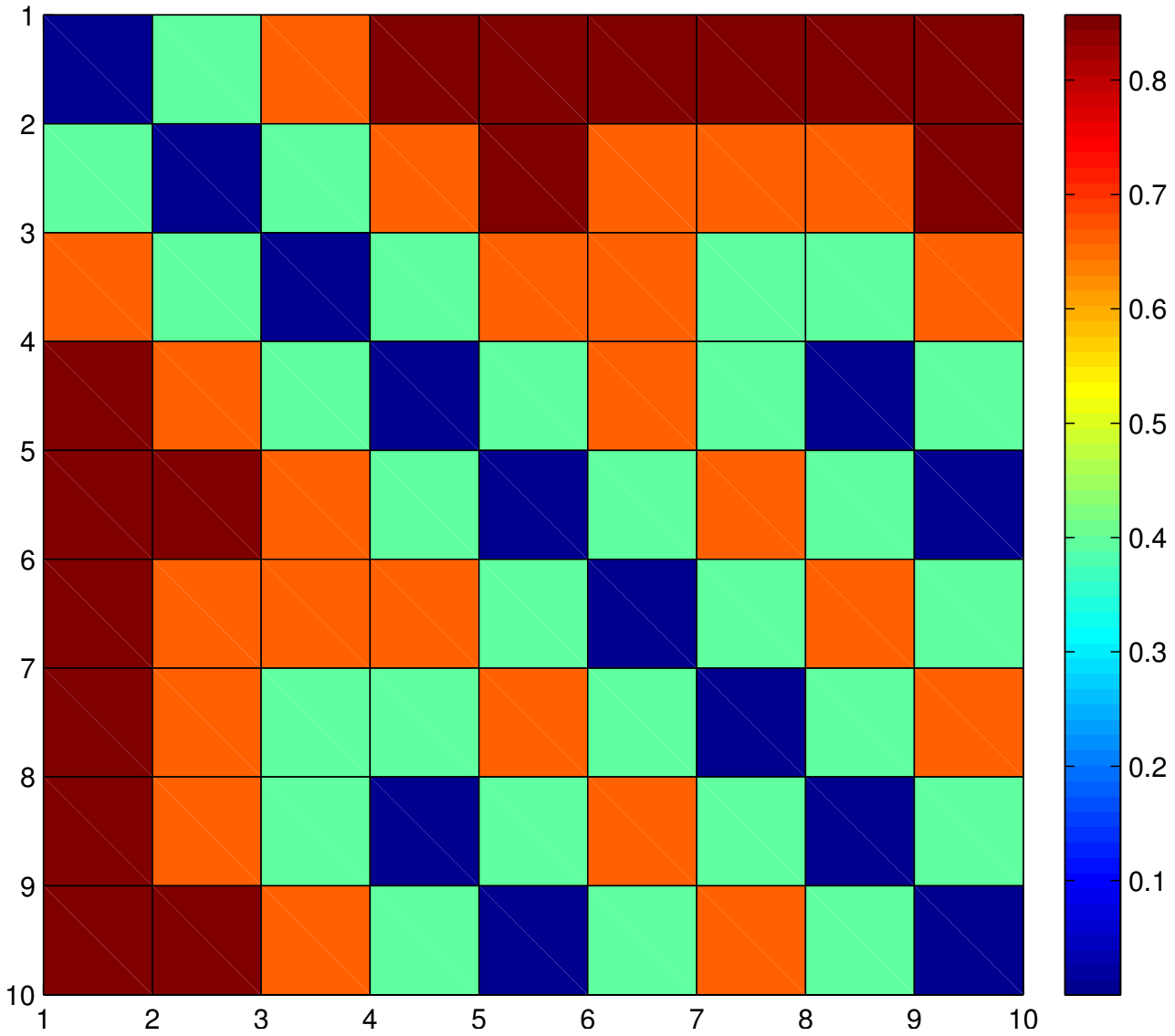

Figure 2

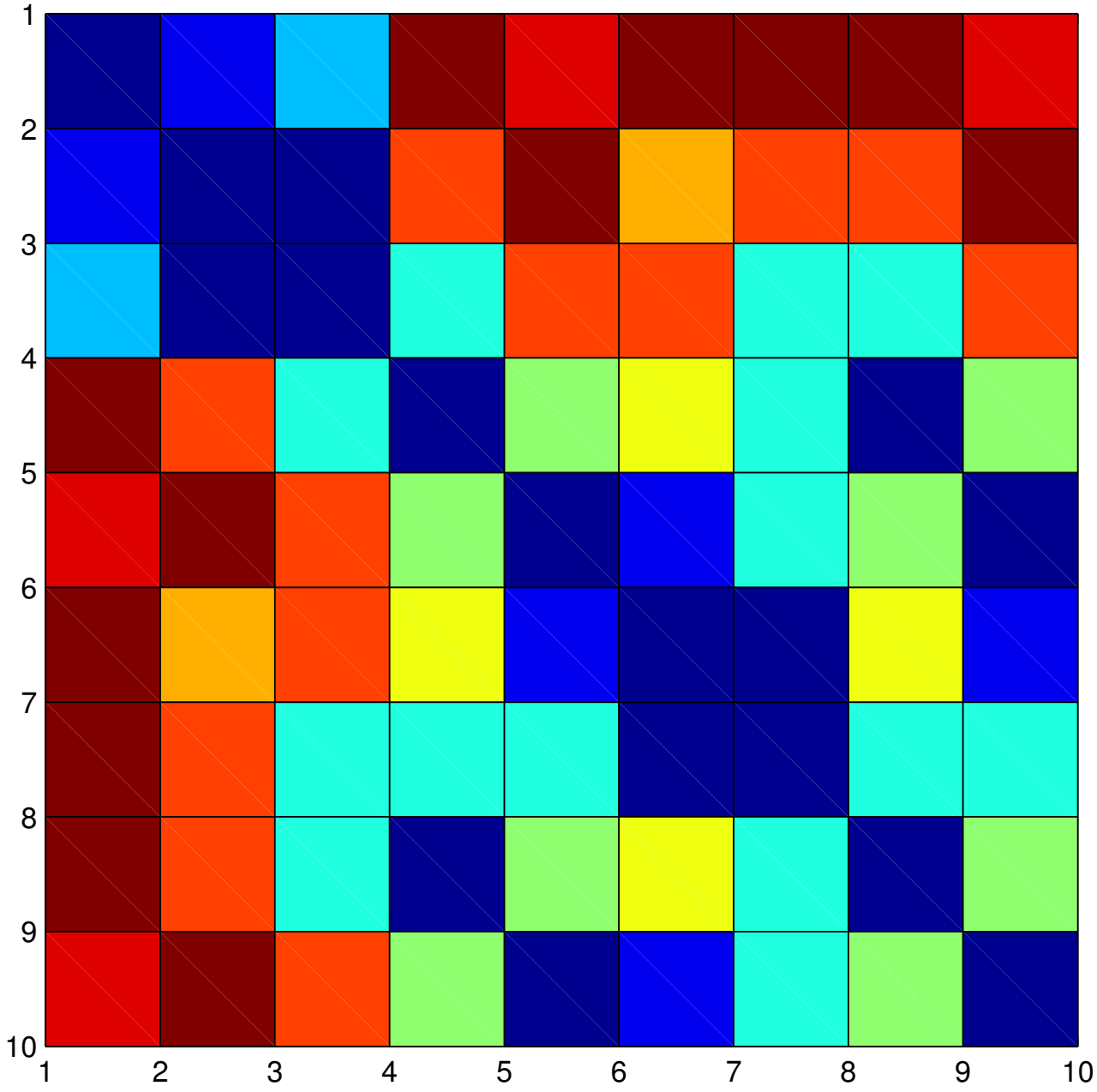

Figure 3

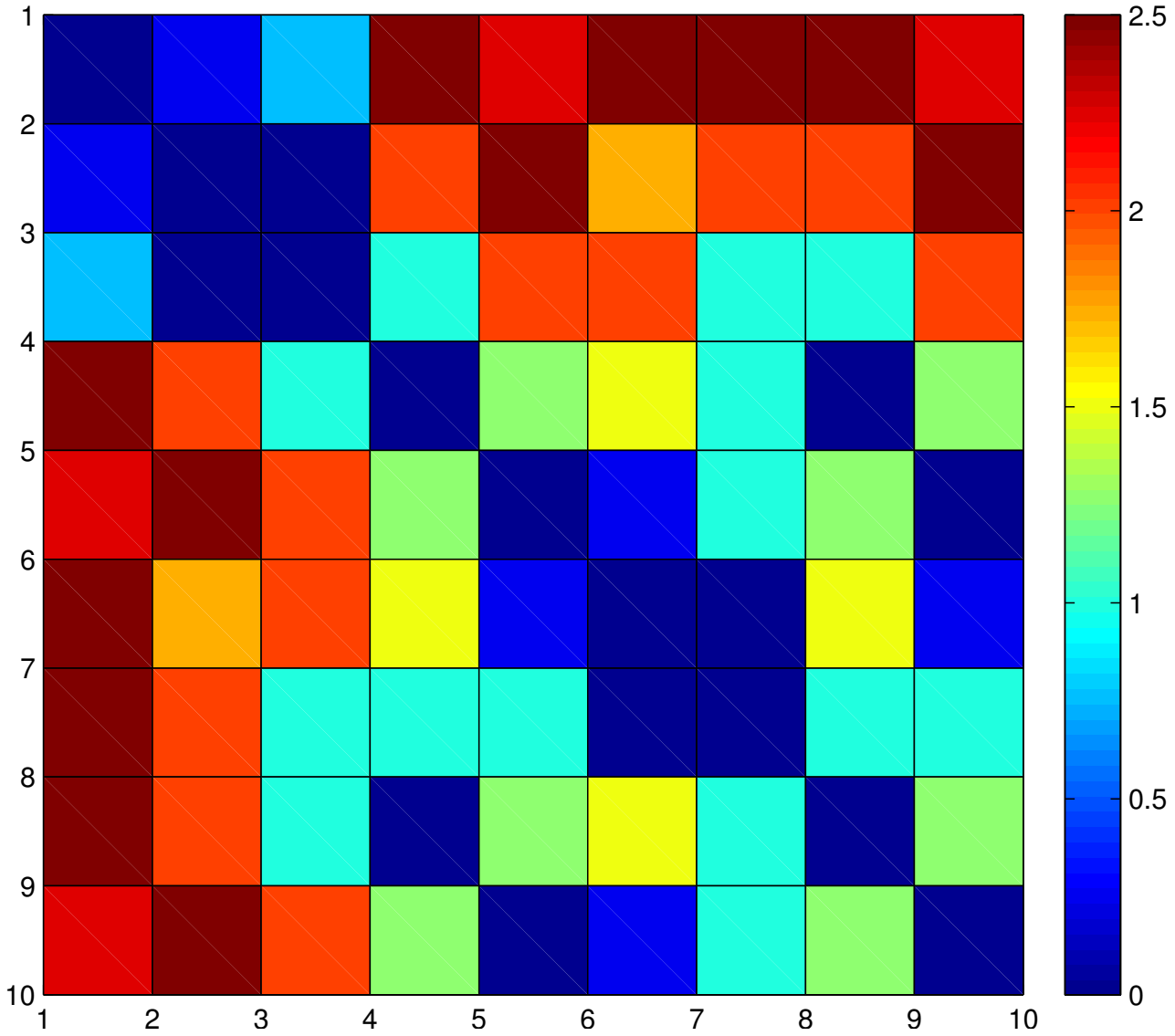

Figure 4

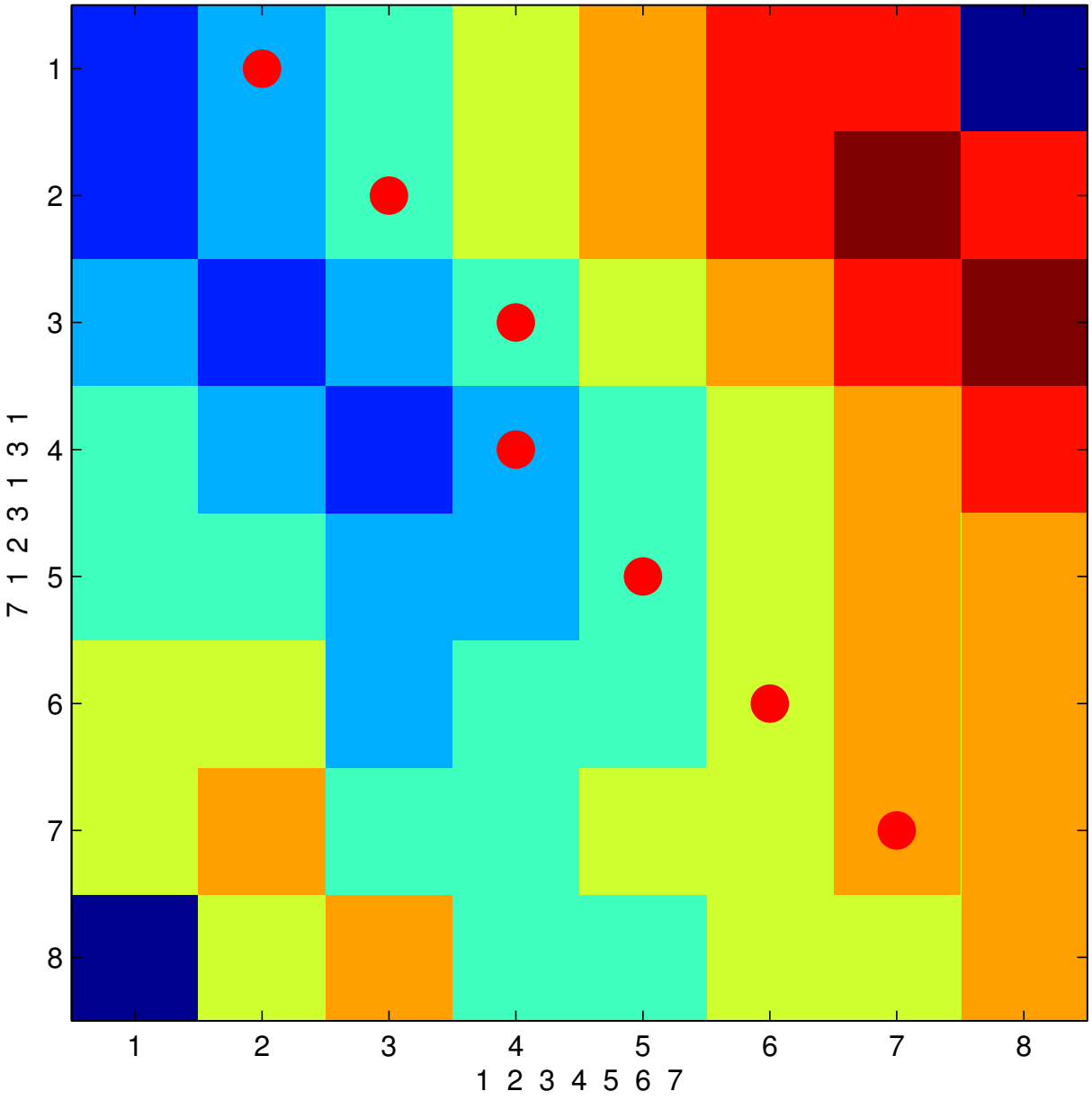

Figure 5

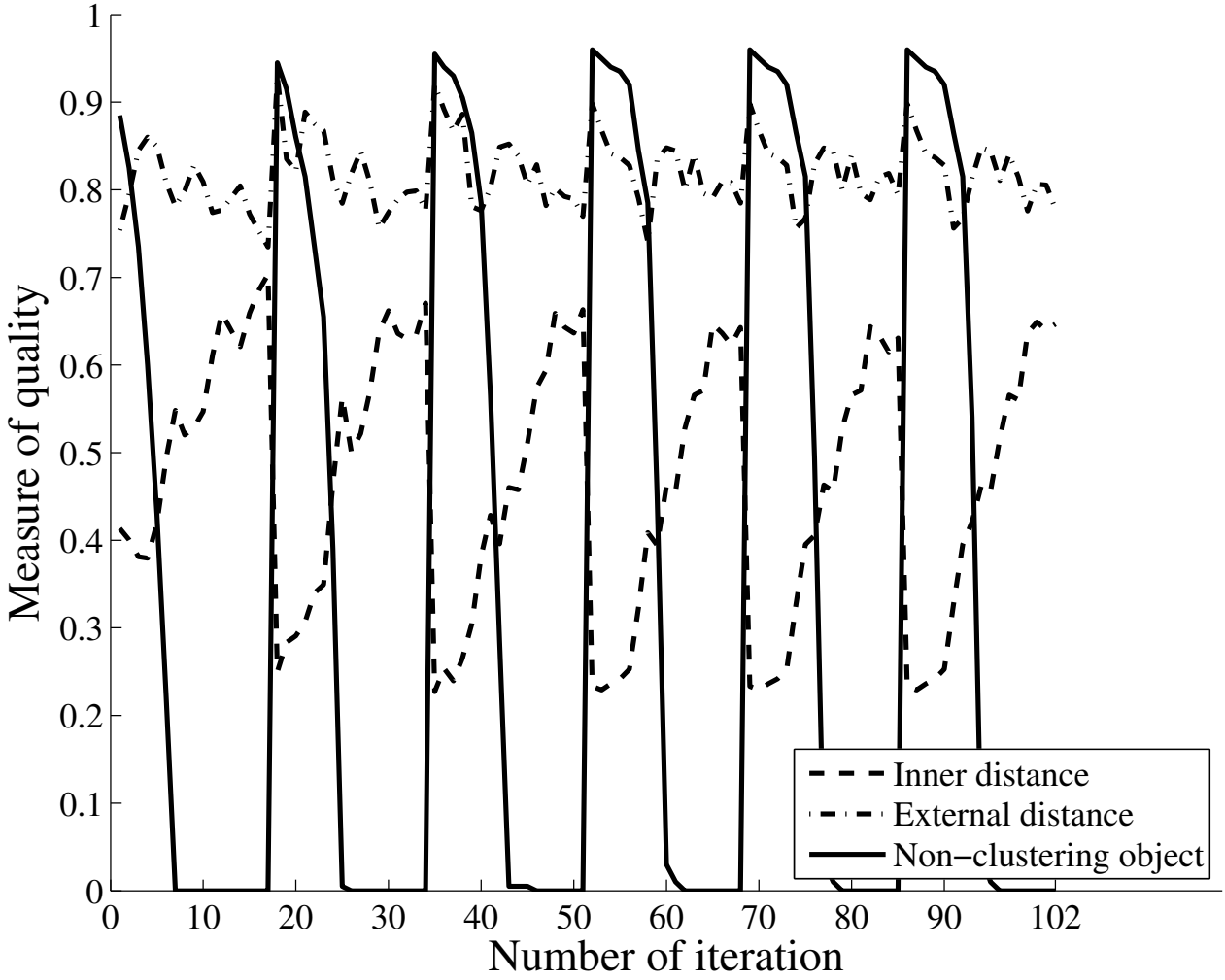
Measure of quality
Number of iteration
Inner distance
External distance
Non−clustering object
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
10
20
30
40
50
60
70
80
90
102

Figure 6

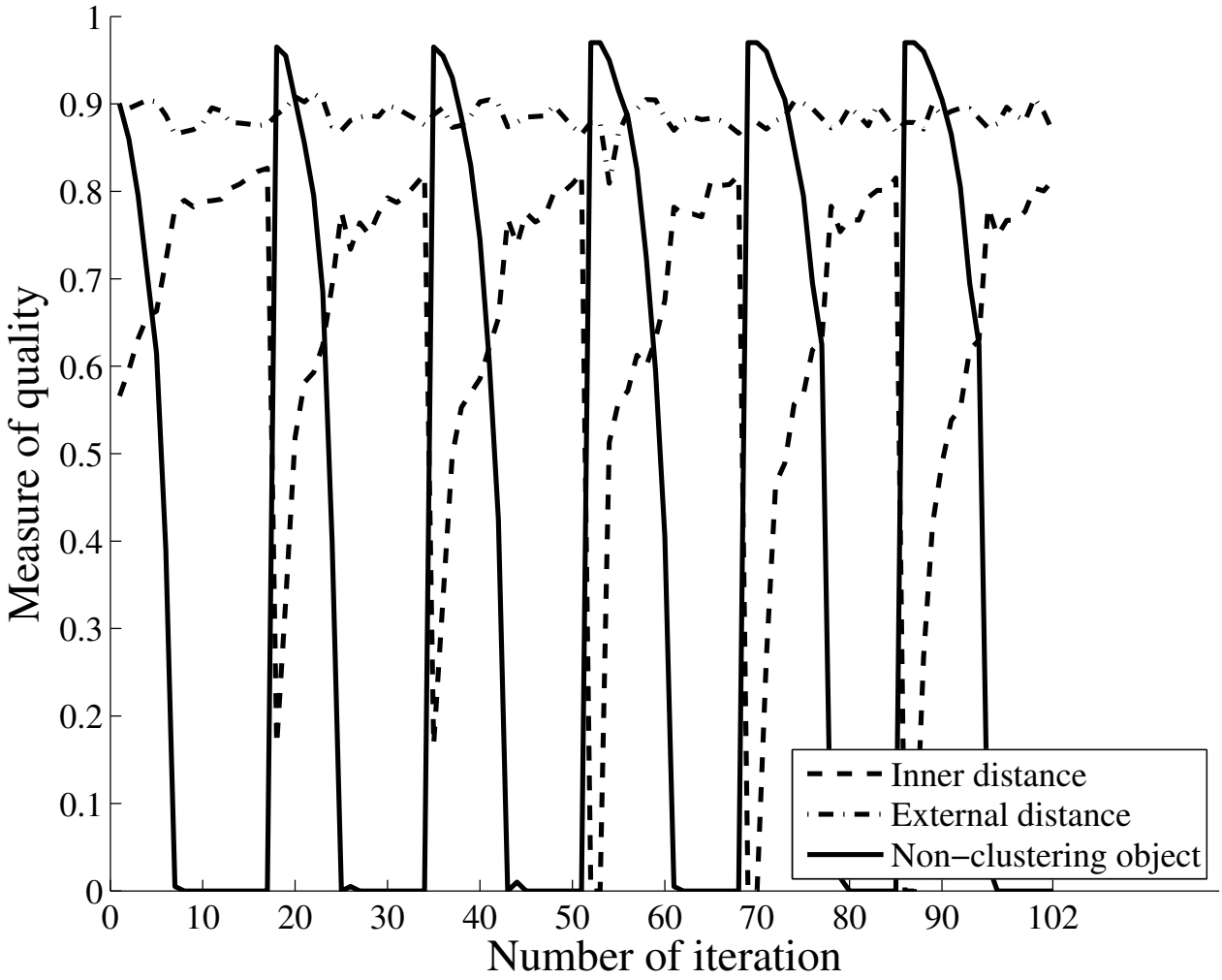
Measure of quality
Number of iteration
Inner distance
External distance
Non−clustering object
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
10
20
30
40
50
60
70
80
90
102

Figure 7

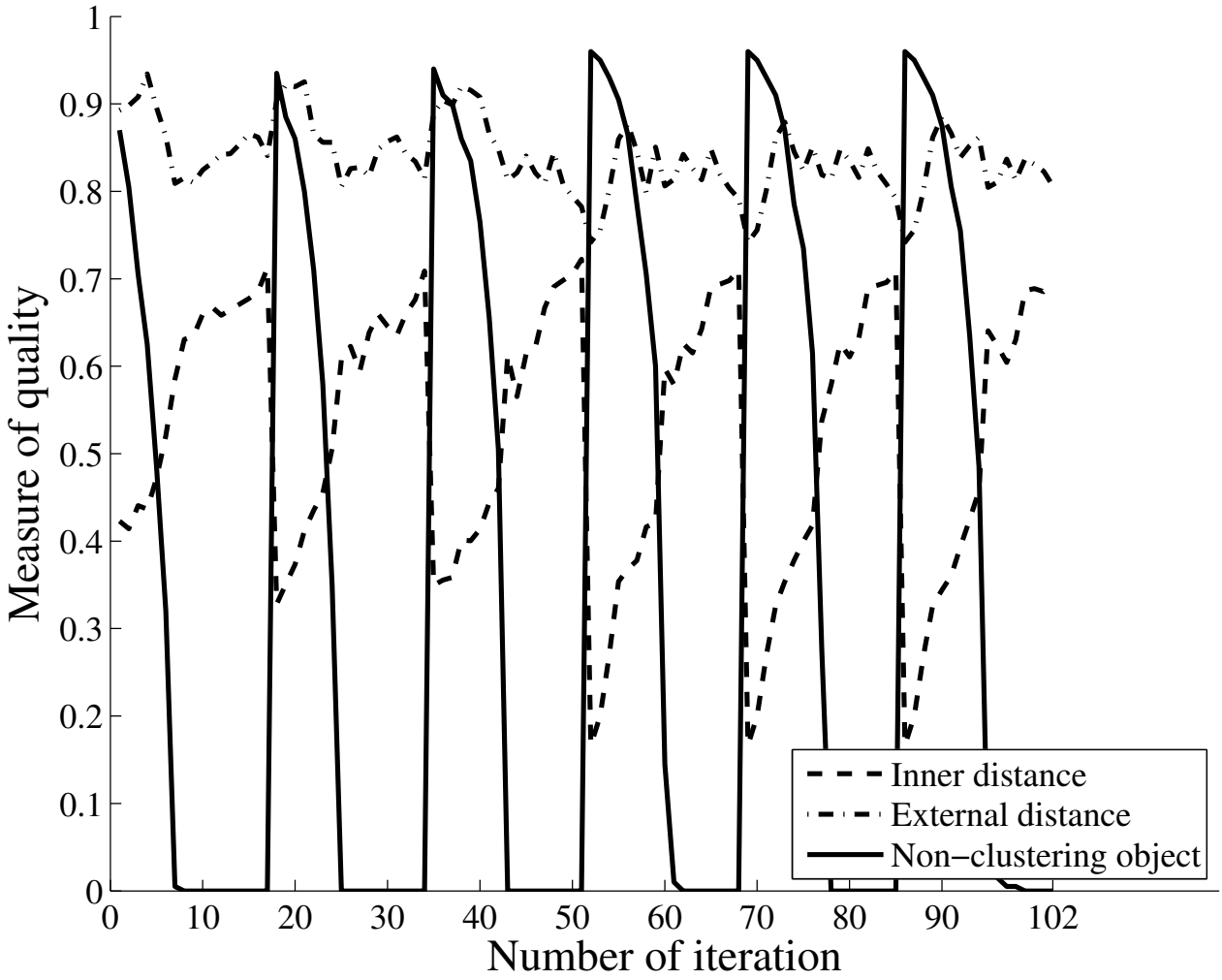
Measure of quality
Number of iteration
Inner distance
External distance
Non−clustering object
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
10
20
30
40
50
60
70
80
90
102

Figure 8

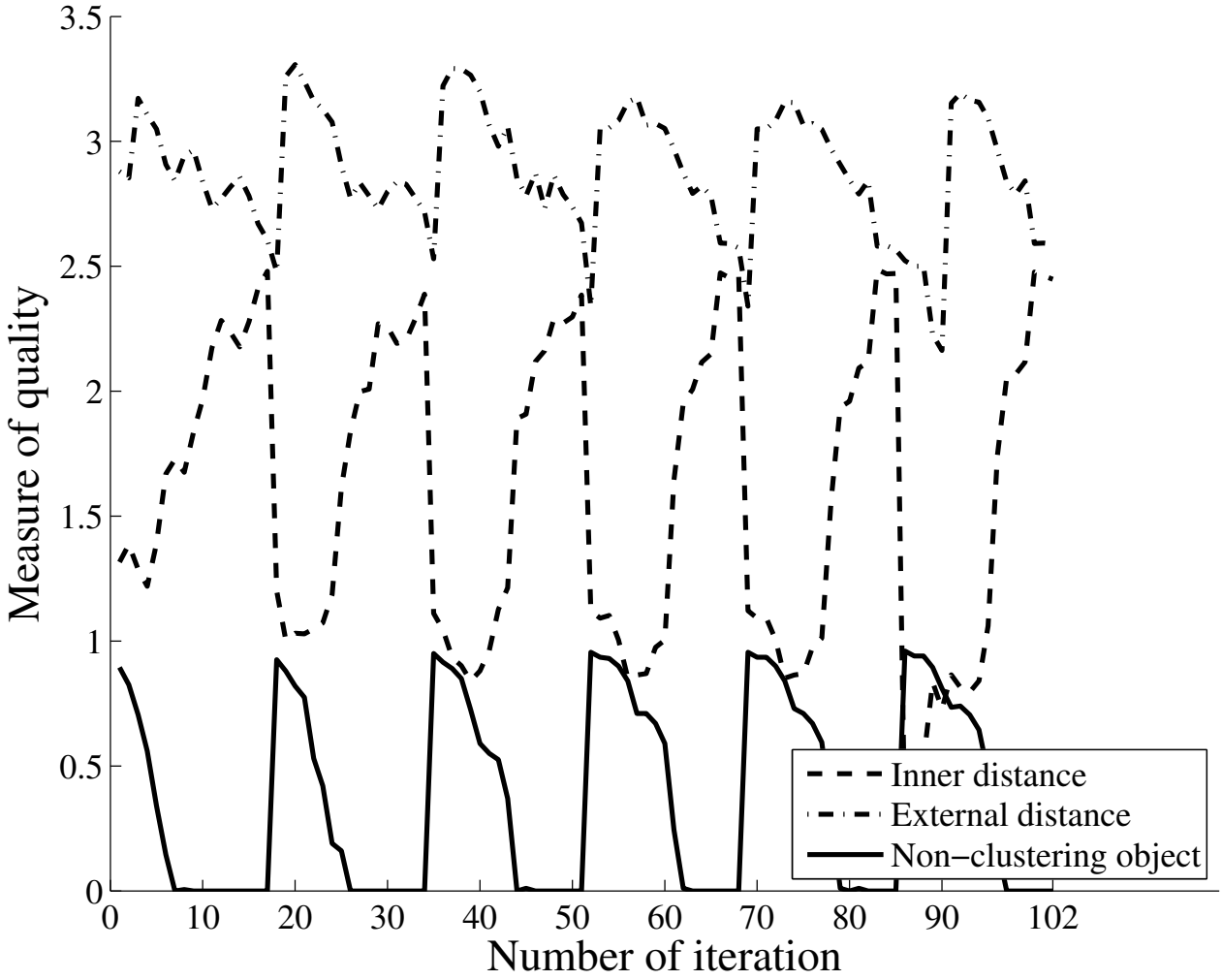
3.5
3
2.5
2
1.5
1
0.5
0
Measure of quality
0
10
20
30
40
50
60
70
80
90
102
Number of iteration
Inner distance
External distance
Non−clustering object

Figure 9

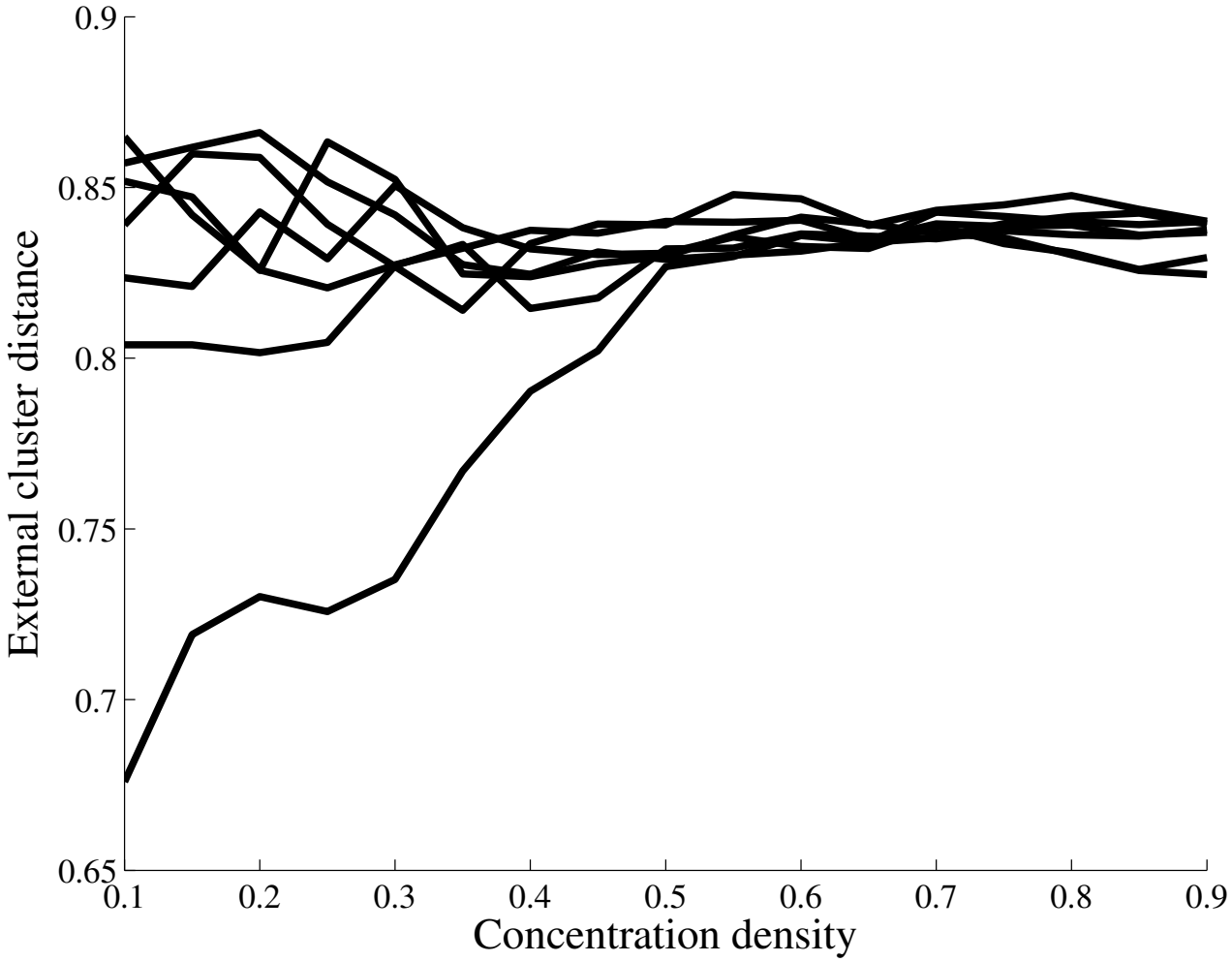
0.9
0.85
0.8
0.75
0.7
0.65
External cluster distance
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Concentration density

Figure 10

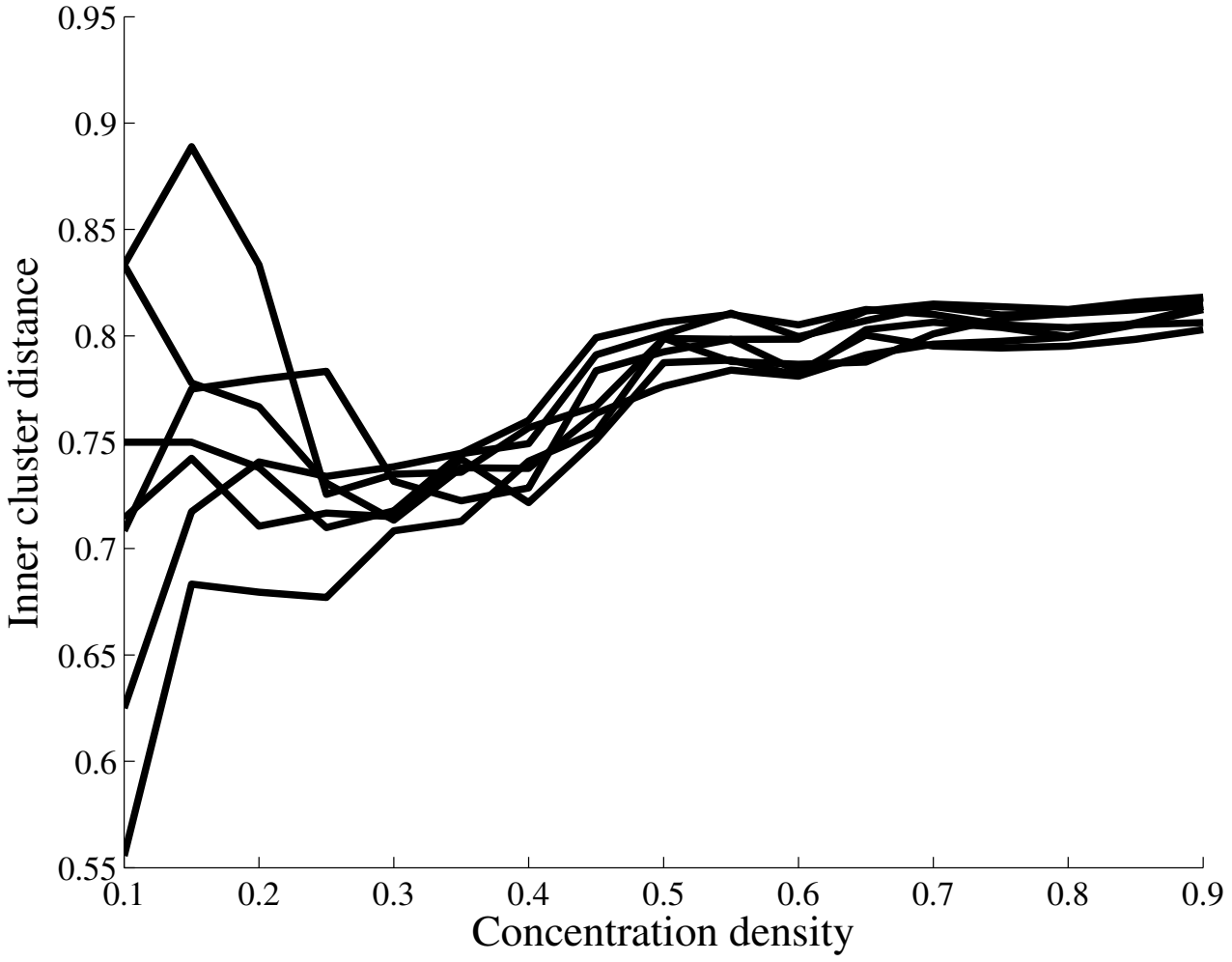
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
Inner cluster distance
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Concentration density

Figure 11

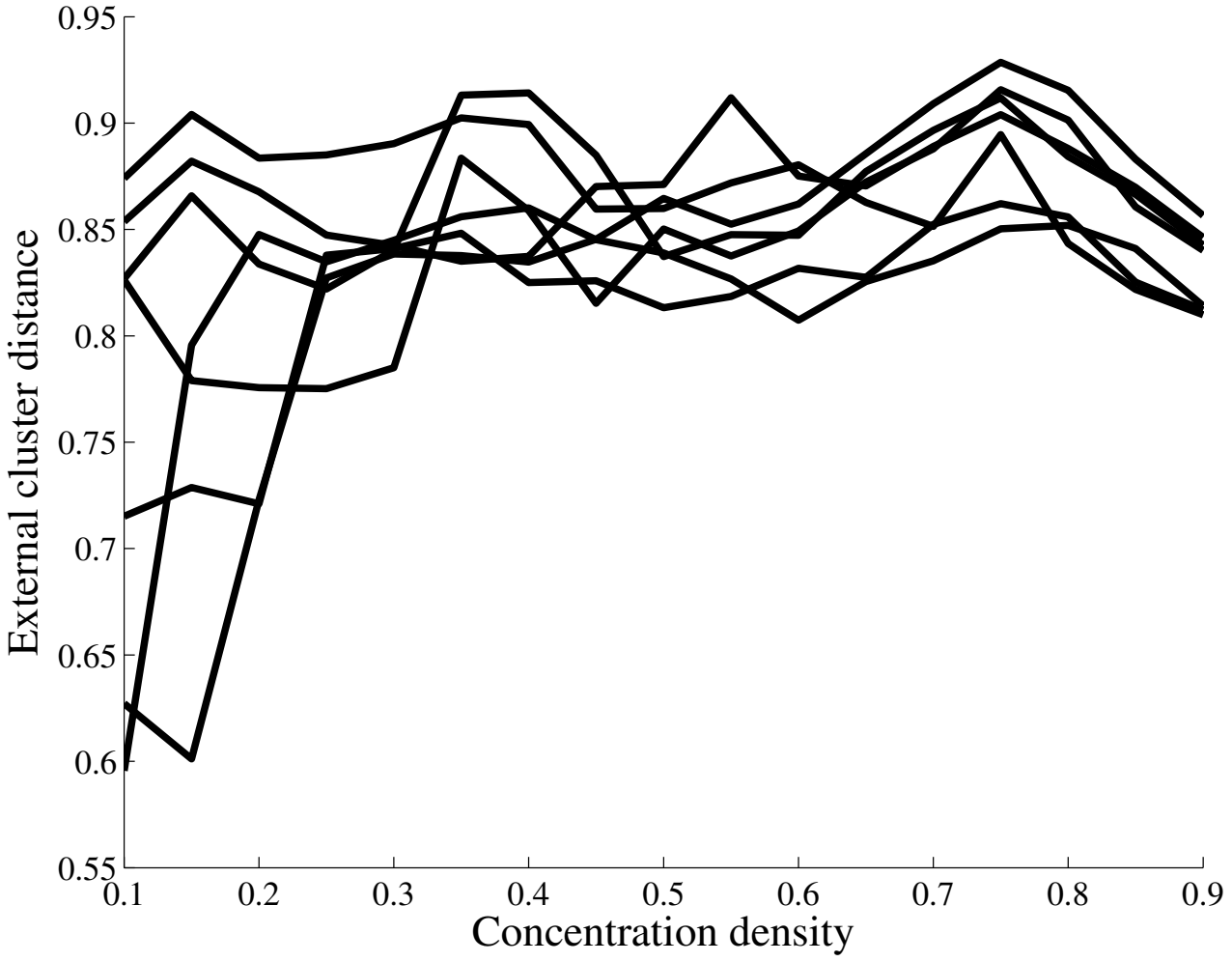
External cluster distance
Concentration density
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

Figure 12

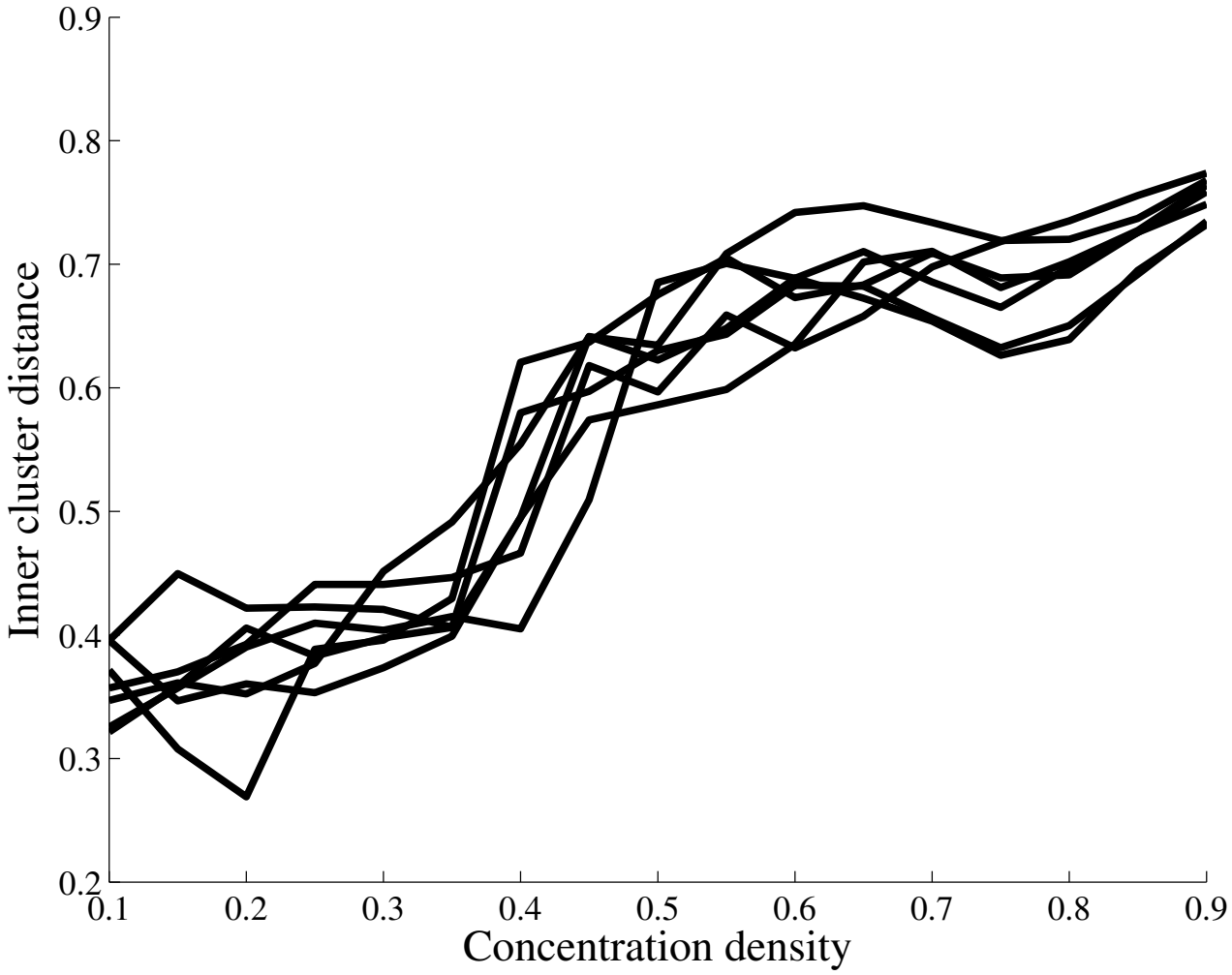
Inner cluster distance
Concentration density
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

Figure 13

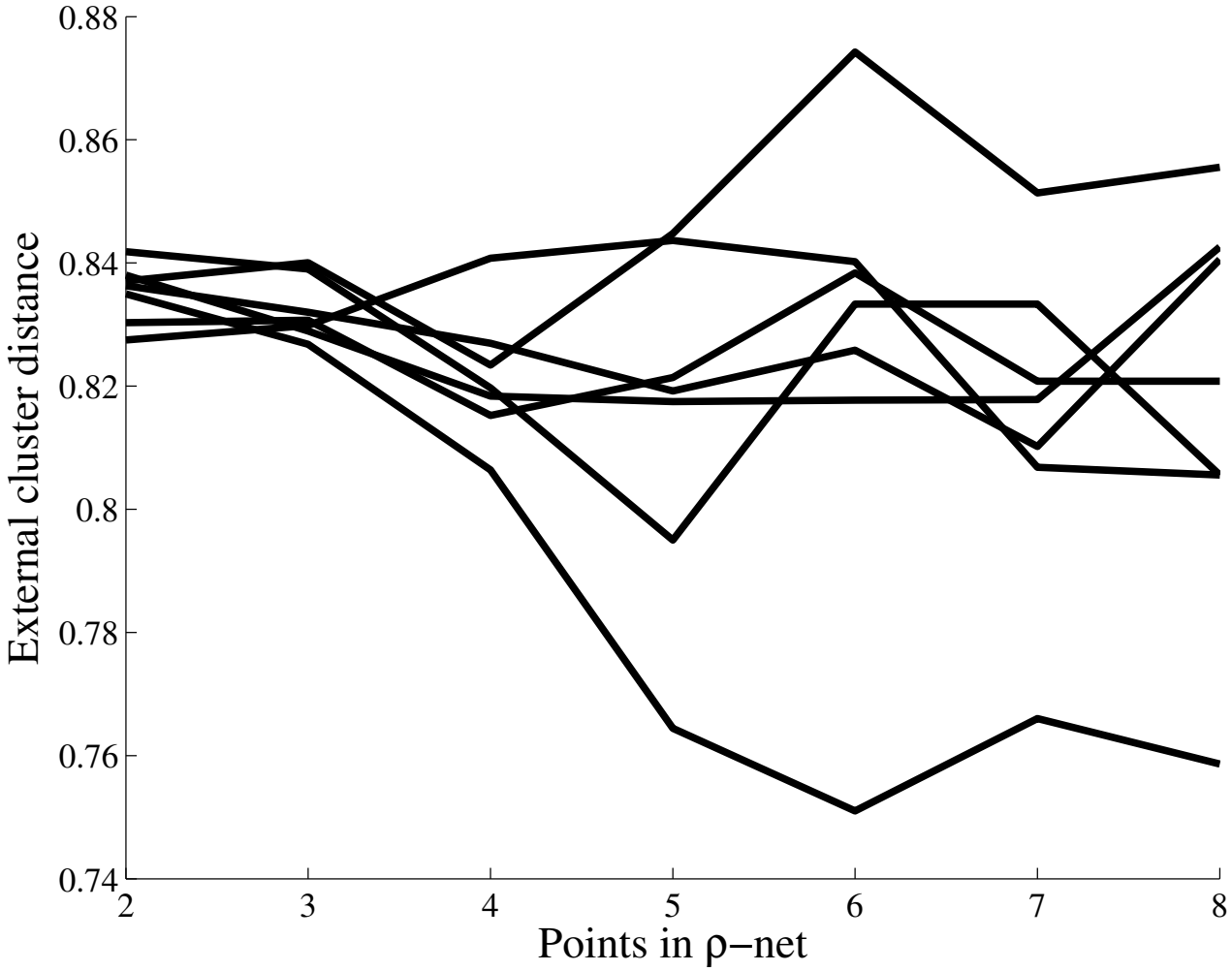
0.88
0.86
0.84
0.82
0.8
0.78
0.76
0.74
External cluster distance
2
3
4
5
6
7
8
Points in ρ−net

Figure 14

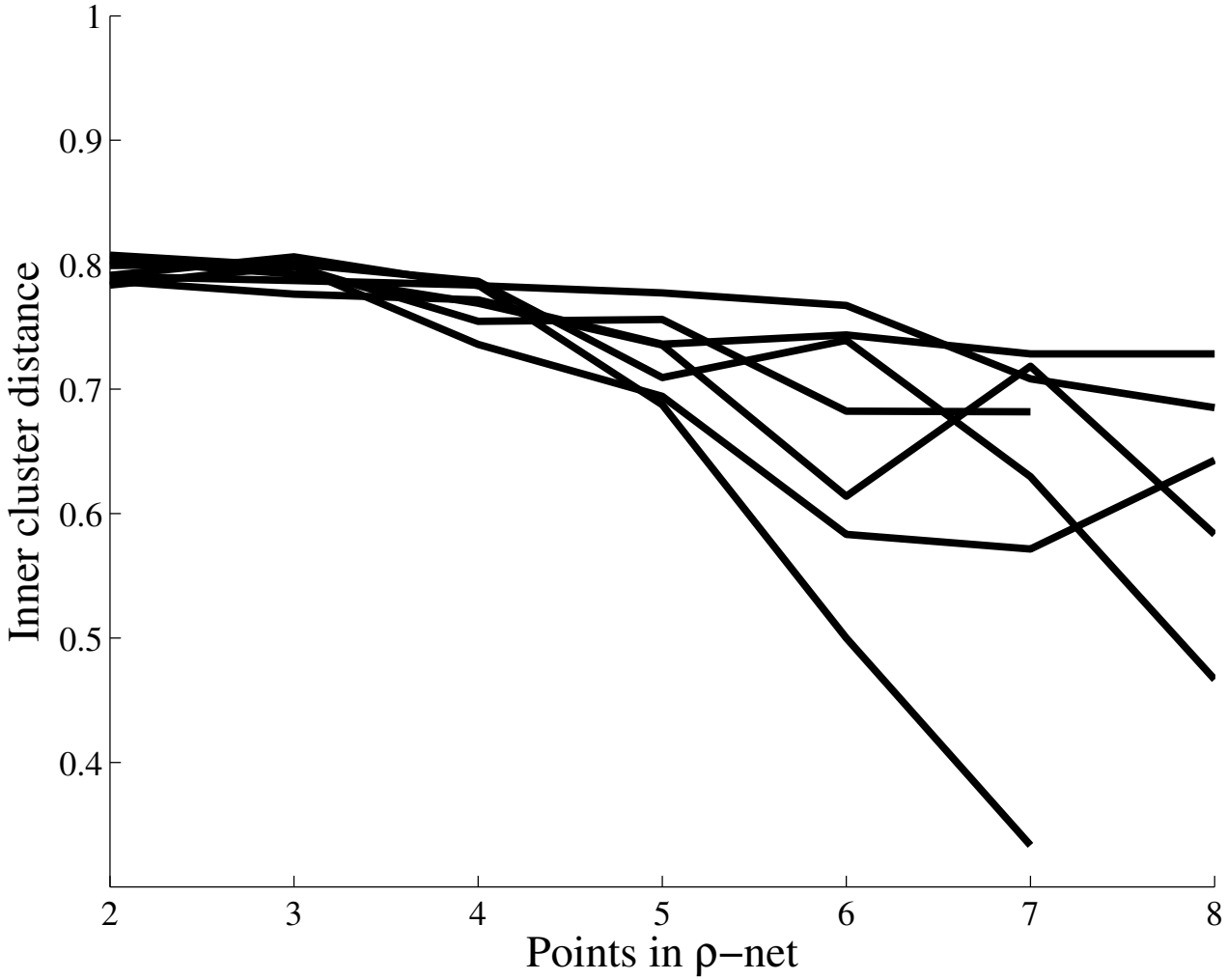
Inner cluster distance
1
0.9
0.8
0.7
0.6
0.5
0.4
2
3
4
5
6
7
8
Points in ρ−net

Figure 15

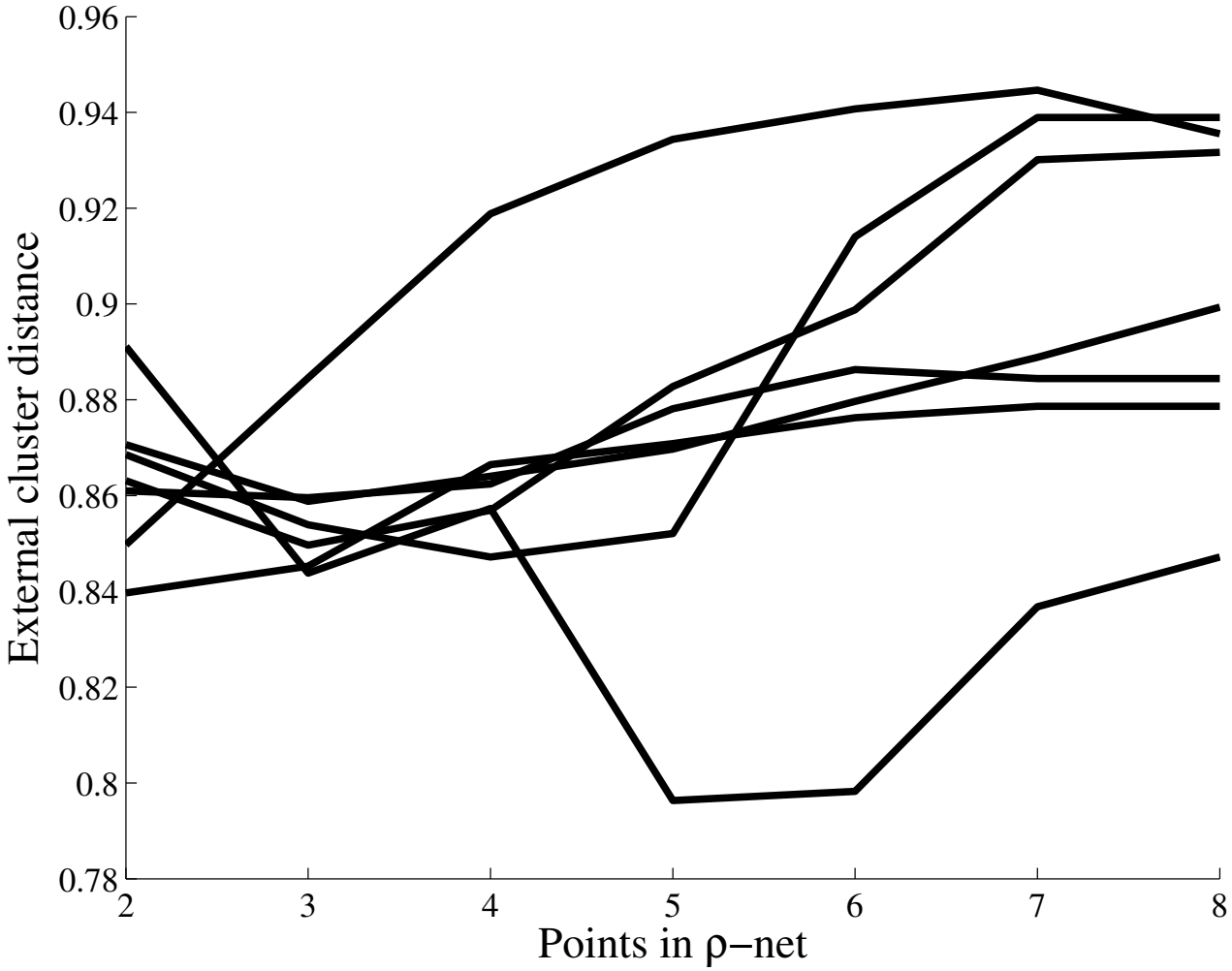
0.96
0.94
0.92
0.9
0.88
0.86
0.84
0.82
0.8
0.78
External cluster distance
2
3
4
5
6
7
8
Points in ρ−net

Figure 16

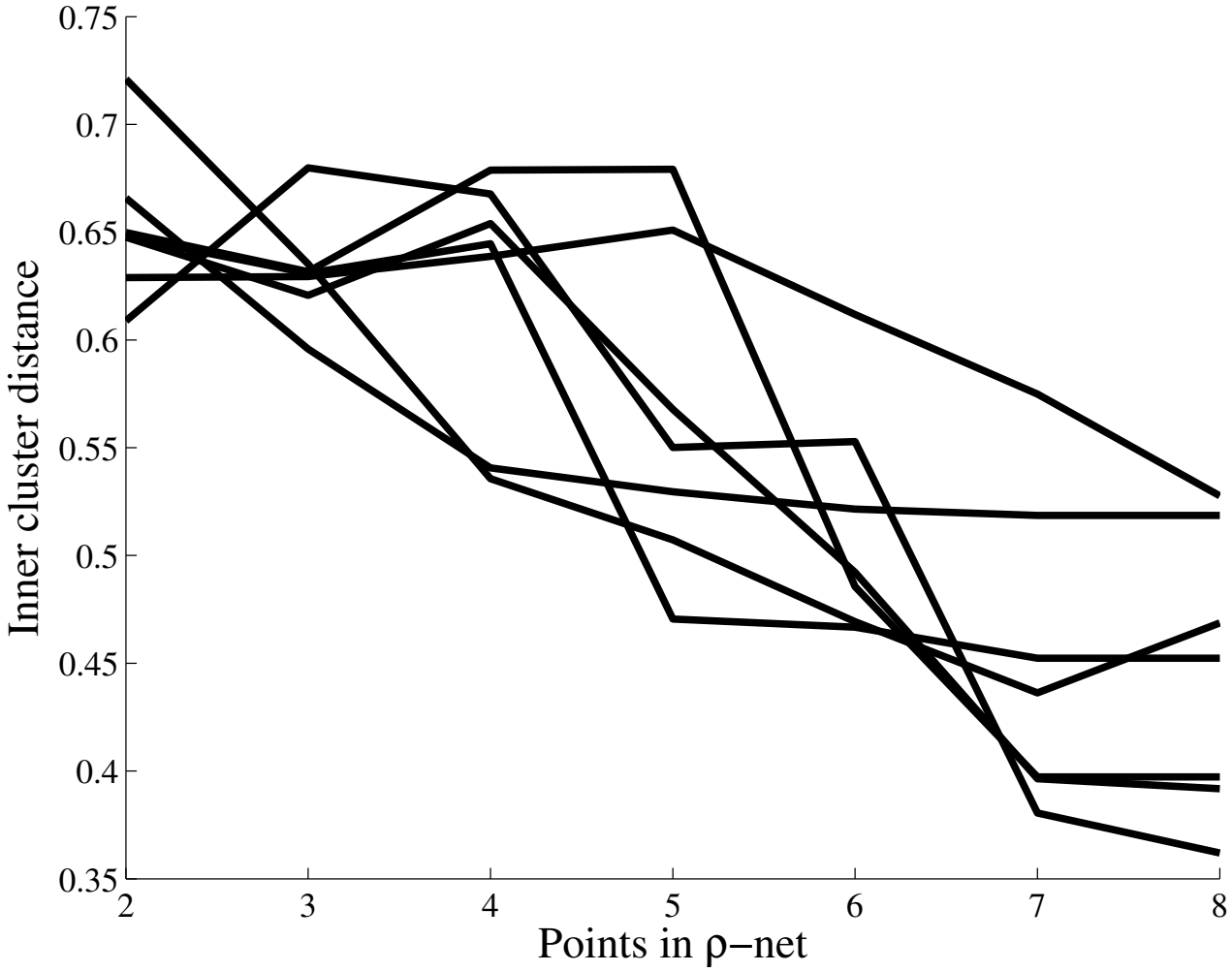
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
Inner cluster distance
2
3
4
5
6
7
8
Points in ρ−net

Figure 17