# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

**4,800**
Open access books available

**122,000**
International authors and editors

**135M**
Downloads

Our authors are among the

**154**
Countries delivered to

**TOP 1%**
most cited scientists

**12.2%**
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

## Interested in publishing with us?
## Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Chapter 1

# Clustering Algorithms for Incomplete Datasets

Loai AbdAllah and Ilan Shimshoni

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.78272

**Abstract**

Many real-world dataset suffers from the problem of missing values. Several methods were developed to deal with this problem. Many of them filled the missing values within fixed value based on statistical computation. In this research, we developed a new versions of the *k-means* and the *mean shift* clustering algorithms that deal with datasets with missing values without filling their values. We developed a new distance function that is able to compute distances over incomplete datasets. The distance was computed based only on the *mean* and *variance* of the data for each attribute. As a result, the runtime complexity of our computation was $O(1)$. We experimented on six standard numerical datasets from different fields. On these datasets, we simulated missing values and compared the performance of the developed algorithms using our distance and the suggested mean computations to other three basic methods. Our experiments show that the developed algorithms using our distance function outperform the existing *k-means and mean shift* using other methods for dealing with missing values.

**Keywords:** missing values, distance metric, weighted Euclidean distance, clustering, mean shift, k-means

## 1. Introduction

Missing values in data are common in real-world applications. They can be caused by human error, equipment failure, system-generated errors, and so on.

In this research, we developed two popular clustering algorithms to run over incomplete datasets: (1) k-means clustering algorithm [1] and (2) mean shift clustering algorithms [2].

Based on [3–6], there are three main types of missing data:

1. Missing completely at random (MCAR): when the missing value is not related to any other sample;
2. Missing at random (MAR): when the probability that a value is missing may depend on some known values but it does not depend on the other missing values;
3. Not missing at random (NMAR): when the probability that a known value is missing depends on the value that would have been observed.

There are two basic types of methods to deal with the problem of incomplete datasets. (1) Deletion: methods from this category ignore all the incomplete instances. These methods may change the distribution of the data by decreasing the volume of the dataset [7]. (2) Imputation: in these methods, the missing values were replaced with known value according to statistical computation. Based on these methods, we convert then incomplete data to complete data, and as a result, the existing machine learning algorithms can be run as they deal with complete data.

One of the most common approaches in this domain is the mean imputation (MI) method that replaces each incomplete data point with the mean of the data. There are several obvious disadvantages to this method: (a) using a fixed instance to replace all the incomplete instances will change the distribution of the original dataset and (b) ignoring the relationship among attributes will bias the performance of subsequent data mining algorithms. These problems were caused since we replace all the incomplete instances with a fixed one. On the other hand, a variant of this method is to replace the missing values only based on the distribution of the attributes. It means that the algorithm will replace each missing value with the mean of the of its attribute (MA) and the whole instance [8]. And in a case that the values were discrete, the missing value will be replaced by the most common (MCA) value in the attribute [9] (i.e., filling the unknown values of the attribute with the value that occurs most often for the same attribute). All those methods ignore the other possible values of the attribute and their distribution and represent the missing value with one value, that is, wrong in real-world datasets.

Finally, the *k*-**Nearest Neighbor Imputation** method [10, 11] estimates the values that should be replaced based on the *k* nearest neighbors based only on the known values. The main obstacle of this method is the runtime complexity.

We can summarize the main drawbacks of each suggested method as: (1) inability to approximate the missing value and (2) inefficiency to compute the suggested value. Based on our suggested method [12], the distance between two points, that they may include missing value, is not only efficient but also takes into account the distribution of each attribute.

To do that in the computation procedure, we take into account all the possible values of the missing value with their probabilities, which are derived from the attribute's distribution. This is in contrast to the MCA and the MA methods, which replace each missing value only with the mode or the mean of each attribute.

There are three possible cases between the values: (a) both of them are known: in this case, the distance will be computed as the Euclidean distance; (b) both of them are missing; and (c) one

value is missing. In the last two cases, the distance will be computed based only on the *mean* and the *variance* of the attribute. As a result, the runtime of the developed distance is $O(1)$ as the Euclidean distance.

In this research, we integrated this distance function in order to develop the *k*-means and the mean shift clustering algorithms. To this end, we derived more two formulas to compute the mean (for *k*-means algorithm) and for computing the gradient function of the local estimated density (for mean shift clustering algorithm).

The developed algorithms yield better results than the other methods and preserve the runtime of the algorithms which deals with complete data as can be seen in the experiments. We experimented on six standard numerical datasets from different fields from the Speech and Image Processing Unit [13]. Our experiments show that the performance of the developed algorithms using our distance function was superior to using other methods.

This chapter is organized as follows. A review of our distance function ($MD_E$) is described in Section 2. The mean computation is presented in Section 3. Section 3 describes several directions for integrating the ($MD_E$) distance and the computed *mean* within the k-means clustering algorithm. The mean shift clustering algorithm is presented in Section 4. Section 4.1 describes how to integrate the ($MD_E$) distance and the derived mean shift vector within the mean shift clustering algorithm. Experimental results of running the developed clustering algorithms are presented in Section 5. Finally, our conclusions and future work are presented in Section 6.

## 2. Our distance measure

Firstly, we will give a short preview to basic distance function that is able to compute distances between points with missing values developed by [2].

Let $A \subseteq \mathbb{R}^K$ be a set of points. For the $i$th attribute $A^i$, the conditional probability for $A_i$ will be computed according to the known values for this attribute from $A$ (i.e., $P(A^i) \sim \chi^i$), where $\chi^i$ is the distribution of the $i$th coordinate.

Given two sample points $X, Y \subseteq \mathbb{R}^K$, the goal is to compute the distance between them. Let $x^i$ and $y^i$ be the $i$th coordinate values from points $X$, $Y$, respectively. There are three possible cases for the values of $x^i$ and $y^i$:

1. Two values are known: the distance between them will be defined as the Euclidean distance.

2. One value is missing: Suppose that $x^i$ is missing and the value $y^i$ is given. Since the value of $x^i$ is unknown, we cannot compute the distance using the Euclidean distance equation. Instead, we compute the expectation of all the distances between the given value $y^i$ and all the possible values from attribute $i$ according to its distribution $\chi^i$.

   Therefore, we approximate the mean Euclidean distance ($MD_E$) between $y^i$ and the missing value $m^i$ as:

$$MD_E(m^i, y^i) = E\left[(x - y^i)^2\right] = \int p(x)(x - y^i)^2 dx = \left((y^i - \mu^i)^2 + (\sigma^i)^2\right).$$

That means, to measure the distance between known value $y^i$ and unknown value, the algorithm will compute the expectation distance for all the distances between $y^i$ and all the possible values of the missing value. These computations did not take into account the possible correlations between the missing values and the other known values (missing completely at random —MCAR) and the probability was computed according to the whole dataset. The resulting mean Euclidean distance will be:

$$MD_E(m^i, y^i) = \left((y^i - \mu^i)^2 + (\sigma^i)^2\right), \tag{1}$$

where $\mu^i$ and $(\sigma^i)^2$ are the *mean* and the *variance* for all the known values of the attribute.

**3.** Both values are missing: In this case, in order to measure the distance, we should compute all the distances between each possible pair of values one for each missing value $x^i$ and $y^i$. Both these values are selected from distribution $\chi^i$.

Then, we compute the expectation of the Euclidean distance between each selected value as we did for the one missing value problem. As a result, the distance is:

$$MD_E(x_i, y_i) = \int\int p(x)p(y)(x - y)^2 dxdy = \left((E[x] - E[y])^2 + \sigma_x^2 + \sigma_y^2\right).$$

As $x$ and $y$ belong to the same attribute, $E[x] = E[y] := \mu^i$ and $\sigma_x = \sigma_y := \sigma^i$. Thus:

$$MD_E(x^i, y^i) = 2(\sigma^i)^2. \tag{2}$$

As we mentioned, all these computations assume that the missing data is MCAR. However, in real-world datasets, the missing data are MAR. In this case, the probability $p(x)$ depends on the other observed values, and then, the distance will be computed as:

$$MD_E(m^i, y^i) = \int p(x|x_{obs})(x - y^i)^2 dx = \left(\left(y^i - \mu^i_{x|x_{obs}}\right)^2 + \left(\sigma^i_{x|x_{obs}}\right)^2\right),$$

where $x_{obs}$ denotes the observed attributes of point $X$, and $\mu^i_{x|x_{obs}}$ and $\left(\sigma^i_{x|x_{obs}}\right)^2$ are the conditional *mean* and *variance*, respectively.

On the other hand, in the case that the missing values are NMAR, the probability $p(x)$ that was used in Eq. (1) will be computed based on this information, and then, the distance will be:

$$MD_E(m^i, y^i) = \int p(x|m^i)(x - y^i)^2 dx = \left(\left(y^i - \mu^i_{x|m^i}\right)^2 + \left(\sigma^i_{x|m^i}\right)^2\right),$$

where $p(x|m^i)$ is the distribution of $x$ when $x$ is missing.

## 3. Mean computation

Since one of our goals is developing a k-means clustering algorithm over incomplete datasets, we need to derive a formula to compute the mean of a given set that may contain incomplete points. We decide to derive this equation based on our distance function $MD_E$.

Let $A \subseteq \mathbb{R}^K$ be a set of $n$ points that may contain points with missing values. Then, the *mean* of this dataset is defined as:

$$\overline{x} = \underset{x \in \mathbb{R}}{\arg\min} \sum_{i=1}^{n} \left(distance\left(x, p_i\right)\right)^2,$$

for any $x \in \mathbb{R}^K$, where $p_i \in A$ denotes each point from the set $A$, and $distance()$ is a distance function.

Let $f(x)$ be a multidimensional function: $f : \mathbb{R}^K :\rightarrow \mathbb{R}$ which is defined as:

$$f(x) = \sum_{i=1}^{n} \left(distance\left(x, p_i\right)\right)^2,$$

In our case, the $distance() = MD_E$. Thus,

$$f(x) = \sum_{i=1}^{n} \left(distance\left(x, p_i\right)\right)^2 = \sum_{i=1}^{n} \left( \underbrace{\sqrt{\sum_{j=1}^{K} MD_E\left(x^j, p_i^j\right)}}_{\text{The } MD_E() \text{ distance}} \right)^2 = \sum_{i=1}^{n} \sum_{j=1}^{K} MD_E\left(x^j, p_i^j\right),$$

where $x^j$ is the coordinate $j$ and $p_i^j$ is the coordinate $j$ in point $p_i$. Since each point $p_i$ may contain missing attributes, and according to the definition of the $MD_E$ distance in the previous section, $f(x)$ will be:

$$f(x) = \sum_{j=1}^{K} \left[ \underbrace{\sum_{i=1}^{n_j} \left(x^j - p_i^j\right)^2}_{\text{there are } n_j \text{ known coordinates}} + \underbrace{\sum_{i=1}^{m_j} \left(\left(x^j - \mu^j\right)^2 + \left(\sigma^j\right)^2\right)}_{\text{there are } m_j \text{ missing coordinates}} \right].$$

$\overline{x}$ is the solution of $f'(x) = 0$, and in a multidimensional case: $\overline{x}$ is the solution of $\nabla f = \vec{0}$, where

$$\nabla f = \left(f'_{x^1}, f'_{x^2}, \ldots, f'_{x^k}\right) = 0,$$

is the gradient of function $f$. Firstly, we will deal with one coordinate, and then, we will generalize it for the other coordinates.

$$\Rightarrow f'_{x^l} = 2\sum_{i=1}^{n_l}\left(x^l - p_i^l\right) + 2\sum_{i=1}^{m_l}\left(x^l - \mu^l\right) = 0$$

$$\Rightarrow nx^l = \sum_{i=1}^{n_l} p_i^l + m_l\mu^l \Rightarrow x^l = \frac{\sum_{i=1}^{n_l} p_i^l}{n} + \frac{m_l\mu^l}{n}$$

$$\Rightarrow x^l = \frac{n_l}{n}\frac{\sum_{i=1}^{n_l} p_i^l}{n_l} + \frac{n-n_l}{n}\mu^l = \mu^l.$$

Thus, we simply get:

$$x^l = \mu^l. \tag{3}$$

Repeating this for all the coordinates yields $\overline{x} = \left(\mu^1, \mu^2, \ldots, \mu^k\right)$. In other words, each coordinate of the mean is the mean of the known values of that coordinate.

In the same way, we derive a formula for computing the weighted mean for each coordinate $l$, yielding:

$$\overline{x}_w^l = \frac{\sum_{i=1}^{n_l} w_i x_i^l + \sum_{i=1}^{n_l} w_i \mu^l}{\sum_{i=1}^{n} w_i},$$

where $w_i$ is the weight of point $x_i$. It means, in order to compute the weighted mean of a set of numbers that some of them are unknown, we must distinguish between known and unknown values. If the value is known, we multiply it with its weight. On the other hand, if the value is missing, we replace it with the mean of the known values and then multiply it by the matching weight.

## 4. k-Means clustering using the $MD_E$ distance

Based on the derived formulas, the $MD_E$ distance and the *mean*, our aim in this research is to develop *k*-means clustering algorithms for incomplete datasets [1].

The $MD_E$ distance and the *mean* are general and can be integrated within any algorithm that computes distances or mean computation. In this section, we describe our proposed method to integrate those formulas within the framework of the *k*-means clustering algorithm.

We developed three different versions for k-means. For simplicity, we assume that all the points are from $\mathbb{R}^2$. We have two way to look about incomplete points. The first one considers each point as a single point, this version is similar to the GMM algorithm described in [14, 15]. On the other hand, the second way is to replace each incomplete point with a set of points according to the data distribution (these are the other two methods). As will be shown in our experiments, they outperform the first algorithm.

The k-means clustering algorithm is constructed from two basic steps: (1) associate each point with its closest centroid, and then, (2) update the centroid based on the new association from Eq. (1). Given dataset $D$ that may contain points with missing values. In the first step, the $MD_E$ distance is used to compute the distances between each data point and the centroids in order to associate each point with the closest centroid. This association is general for all the three versions. However, there are several possible ways to then compute the new centroids of the clusters. We use **Figure 1(a)** in order to illustrate those possibilities. In this example, we see two clusters (i.e., $C1$ was assigned to be the yellow cluster and $C2$ was assigned to be the brown cluster). Our goal is to calculate the centers of each cluster. As an example, we will deal only with $C1$. If all the instances do not contain missing values, the centroid will be computed based on the Euclidean *mean* formula, resulting in the magenta star.

However, when the associated points for a given cluster contain incomplete points, it is not clear how to compute the mean. In the given example, let $(x_0, ?)$ (i.e., the red star) be a point with a missing $y$ value and $x = x_0$. This point was associated with $C1$'s cluster using the $MD_E$ distance. It is important to note that we are able to associate incomplete points with closest centroid even though their geometric locations are unknown since we use the $MD_E$ distance.

On the other hand, using the $MD_E$ distance is similar to use the MA-method based on the Euclidean distance, the point $(x_0, ?)$ will be replaced with $\left(x_0, \mu_y\right)$. It is clear that the difference between the two methods is only the variance of known values in coordinate $y$, a fixed value that does not influence the association result.

**The naïve method** to compute the new centroid is by replacing the point with the missing value with all the possible points

$$(x_0)_{possible} = \left\{ \left(x_0, y_p\right) | y_p \in Y_{possible} \right\},$$

the set of all the possible points that satisfy $x = x_0$. And

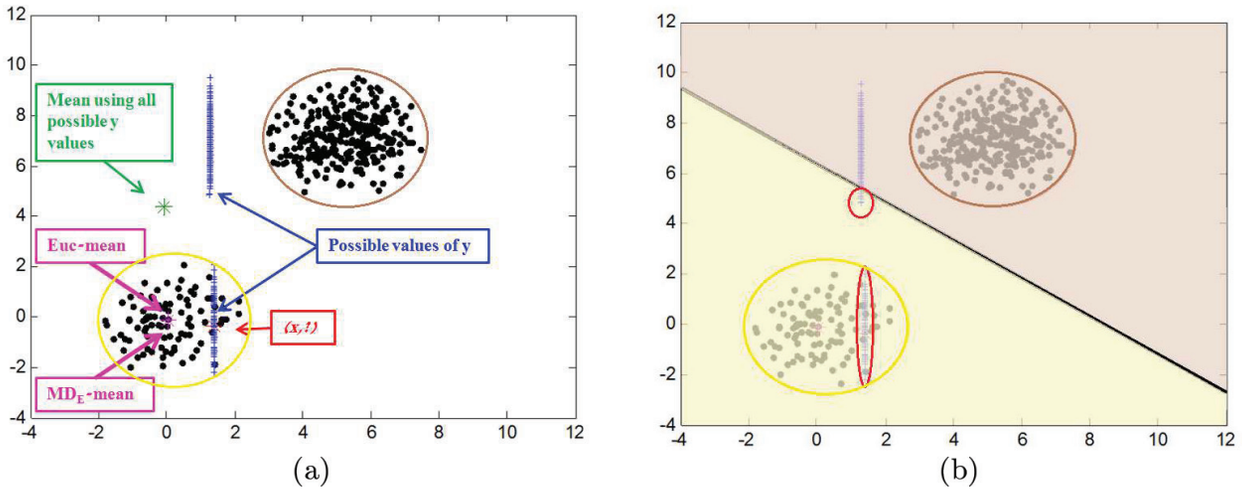


**Figure 1.** An example for computing the centroids for two clusters in a dataset with missing values. (a) shows the results of the different methods of computing the *mean*. (b) shows the Voronoi diagram.

$$Y_{possible} = \{y \in \mathbb{R} | \exists (x, y) \in D\},$$

denote all the possible values for attribute $Y$. And then computing the *mean* according to these points ($C1_{real}$ and $(x_0)_{possible}$), where each point from $C1_{real}$ has weight one and each point from $(x_0)_{possible}$ has weight $\frac{1}{|Y_{possible}|}$. Where

$$C1_{real} = \{(x, y) \in D | (x, y) \in C1\}$$

be the set of all the data points without missing values that are associated with the $C1$ cluster. As a result, the weighted *mean* of $C1$ is:

$$mean(C1) = \frac{\sum_{(x,y) \in C1_{real}} (x, y) + \left(x_0, \mu_y\right)}{|C1_{real}| + \sum \frac{1}{|Y_{possible}|}}. \tag{4}$$

This is identical to the Euclidean *mean* when the missing point is replaced with $\left(x_0, \mu_y\right)$ and is equivalent to the MA method when $\left(x_0, \mu_y\right)$ is associated with $C1$. As a result, the real centroid of the cluster (the magenta star) moves to the green star as described in **Figure 1(b)**, where not all the blue "+" marks are belonging to $C1$.

As a result, the mean computation must distinguish between two possible methods. The first method (which we call ***k-mean-***$MD_E$) takes into account all the possible points that their $y$ coordinates are the $y$ coordinates of the real data points from the yellow cluster in addition to the real points within the yellow circle. As a result, the *mean* of this set will be computed based on all the real points $C1_{real}$ and $C1_{(x_0)_{possible}}$ where,

$$C1_{(x_0)_{possible}} = \left\{ \left(x_0, y_p\right) \in (x_0)_{possible} | \exists (x, y) \in C1_{real} \wedge y = y_p \right\}.$$

Computing the new centroid using Eq. (3) yields not only the same centroid as using the Euclidean distance, but also preserves the runtime of the standard k-means using the Euclidean distance.

The second method (which we called ***k-mean-HistMD***$_E$): In this case, we first associate each of the points from $(x_0)_{Ypossible}$ with its nearest center, and after that compute a weighted *mean*. It means that to compute the *mean*, we will take into account all the real points $C1_{real}$, in addition to $PC1_{possible}$ where

$$PC1_{possible} = \left\{ \left(x_0, y_p\right) \in (x_0)_{possible} | \left(x_0, y_p\right) \in C1 \right\}.$$

According to this method, use all the points from $(x_0)_{possible}$ that are associated with the $C1$ cluster and not only the points from $(x_0)_{possible}$ whose $y$ coordinates are from the real points associated with that cluster. Since the weights are computed using the entire dataset, we cannot use Eq. (3). To this end, our suggested method for implementing the *mean* computation is simply to replace each point with a missing value with the $|Y_{possible}|$ points, each with a weight $\frac{1}{|Y_{possible}|}$, and run

weighted k-means on the new dataset. This method, in one hand, is simple to implement, but in the other hand, its runtime is high, since each point with, for example, a missing $y$ value will be replaced with all $|Y_{possible}|$ points. As a result, the size of the dataset will be:

$$|D_{real}| + (|D| - |D_{real}|) \cdot |Att_{possible}|,$$

where $D_{real}$ is the set of each data points that do not contain missing values. In order to reduce the runtime complexity, we turn to use Voronoi diagram. Based on Voronoi diagram, the data space is partitioned to $k$ subspaces (as can be seen in **Figure 1(b)**). Each point is associated with the subspace of the cluster in which it lies.

The third possibility is to divide the $y$ value space to several disjoint intervals. Where, each interval will be represented by its mean, and the weight of each interval will be the ratio between the number of points in the interval to the number of all possible points. This method we called ***k-mean-HistMD***$_E$. *k-mean-HistMD*$_E$ method approximates the two methods mentioned before that compute the weighted *mean*.

In conclusion, we have three methods:

- The naïve method which is equivalent to the MA method.
- k-means-$MD_E$
- k-mean-HistMD$_E$

These methods differ in their performance, efficiency, and the way they work.

## 5. Mean shift algorithm

In this section, we will describe another use case that integrates the derived distance function $MD_E$ within the framework of mean shift clustering algorithm. Firstly, we will give a short overview of the mean shift algorithm, and then, we will describe how we use $MD_E$ distance in this algorithm. Here, we only review some of the results described in [16, 17] which should be consulted for the details. Let $x_i \in \mathbb{R}^d$, $i = 1, \ldots, n$ is associated with a bandwidth value $h > 0$. The *sample point* density estimator at point $x$ is

$$\widehat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right). \tag{5}$$

Based on a symmetric kernel $K$ with bounded support satisfying

$$K(x) = c_{k,d} k\left(\|x\|^2\right) \qquad \|x\| \leq 1 \tag{6}$$

is a nonparametric estimator of the density at $x$ in the feature space. Where $k(x), 0 \leq x \leq 1$ is the *profile* of the kernel and the normalization constant $c_{k,d}$ assures that $K(x)$ integrates to one. As a result, the density estimator Eq. (5) can be rewritten as

$$\widehat{f}_{h,k}(x) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^{n} k\left( \left\| \frac{x - x_i}{h} \right\|^2 \right). \tag{7}$$

As a first step in the analysis is to find the modes of the density which are located among the zeros of the gradient $\nabla f(x) = 0$, of a feature space with the underlying density $f(x)$, and the mean shift procedure is a way to find these zeros without the need to estimate the density.

Therefore, the density gradient estimator is obtained as the gradient of the density estimator by capitalizing on the linearity of Eq. (7).

$$\widehat{\nabla f}_{h,K}(x) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (x - x_i) k'\left( \left\| \frac{x - x_i}{h} \right\|^2 \right). \tag{8}$$

Define $g(x) = -k'(x)$, then the kernel $G(x)$ is defined as:

$$G(x) = c_{g,d} g\left( \|x\|^2 \right).$$

Introducing $g(x)$ into Eq. (8) yields

$$\begin{aligned} \widehat{\nabla f}_{h,K}(x) &= \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (x_i - x) g\left( \left\| \frac{x - x_i}{h} \right\|^2 \right) \\ &= \frac{2c_{k,d}}{nh^{d+2}} \left[ \sum_{i=1}^{n} g\left( \left\| \frac{x - x_i}{h} \right\|^2 \right) \right] \left[ \frac{\sum_{i=1}^{n} x_i g\left( \left\| \frac{x - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g\left( \left\| \frac{x - x_i}{h} \right\|^2 \right)} - x \right], \end{aligned} \tag{9}$$

where $\sum_{i=1}^{n} g\left( \left\| \frac{x - x_i}{h} \right\|^2 \right)$ is assumed to be a positive number. Both terms of the product in Eq. (9) have special significance. The first term is proportional to the density estimate at $x$ computed with the kernel $G$. The second term

$$m_G(x) = \frac{\sum_{i=1}^{n} x_i g\left( \left\| \frac{x - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g\left( \left\| \frac{x - x_i}{h} \right\|^2 \right)} - x \tag{10}$$

is called the *mean shift vector*. The mean shift vector thus points toward the direction of maximum increase in the density. The implication of the mean shift property is that the iterative procedure

$$y_{j+1} = \frac{\sum_{i=1}^{n} x_i g\left( \left\| \frac{y_j - x_i}{h} \right\| \right)}{\sum_{i=1}^{n} g\left( \left\| \frac{y_j - x_i}{h} \right\| \right)} \quad j = 1, 2, \ldots \tag{11}$$

In real world, most often the convergence points of this iterative procedure are the local maxima (modes) of the density. All the points that share the same mode are clustered within the same cluster. Therefore, we get clusters as the number of modes.

### 5.1. Mean shift computing using the $MD_E$ distance

This section describes the way to integrate the $MD_E$ distance within the framework of the mean shift clustering algorithm. To achieve this mission, we will first compute the mean shift vector using the $MD_E$ distance. And then, we will integrate the $MD_E$ and the derived mean shift vector within the mean shift algorithm.

Using the derived $MD_E$ distance the density estimator in Eq. (7) will be written as:

$$\widehat{f}_{h,k}(x) = \frac{c_{k,d}}{nh^d}\sum_{i=1}^{n} k\left(\left\|\frac{x-x_i}{h}\right\|^2\right) = \frac{c_{k,d}}{nh^d}\sum_{i=1}^{n} k\left(\frac{\sum_{j=1}^{d} MD_E\left(x^j, x_i^j\right)^2}{h^2}\right). \tag{12}$$

Since each point $x_i$ may contain missing attributes, $\widehat{f}_{h,k}(x)$ will be:

$$\widehat{f}_{h,k}(x) = \frac{c_{k,d}}{nh^d}\sum_{i=1}^{n} k\left(\underbrace{\frac{\sum_{j=1}^{kn_i} MD_E\left(x^j, x_i^j\right)^2}{h^2}}_{\text{each } x_i \text{ has } kn_i \text{ known attributes}} + \underbrace{\frac{\sum_{j=1}^{unkn_i} MD_E\left(x^j, x_i^j\right)^2}{h^2}}_{\text{each } x_i \text{ has } unkn_i \text{ missing attributes}}\right).$$

According to the definition of the $MD_E$ distance, we obtain:

$$\widehat{f}_{h,k}(x) = \frac{c_{k,d}}{nh^d}\sum_{i=1}^{n} k\left(\frac{\sum_{j=1}^{kn_i}\left(x^j - x_i^j\right)^2}{h^2} + \frac{\sum_{j=1}^{unkn_i}\left(x^j-\mu^j\right)^2 + \left(\sigma^j\right)^2}{h^2}\right). \tag{13}$$

Now, we will compute the gradient of the density estimator in Eq. (13).

$$\begin{aligned}
\nabla\widehat{f}_{h,k}(x) &= \frac{c_{k,d}}{nh^{d+2}}\sum_{i=1}^{n}\left[\sum_{j=1}^{kn_i}\left(x^j - x_i^j\right)^2 + \sum_{j=1}^{unkn_i}\left(x^j-\mu^j\right)^2 + \left(\sigma^j\right)^2\right]' \\
&\quad \cdot k'\left(\frac{\sum_{j=1}^{kn_i}\left(x^j - x_i^j\right)^2}{h^2} + \frac{\sum_{j=1}^{unkn_i}\left(x^j-\mu^j\right)^2 + \left(\sigma^j\right)^2}{h^2}\right) \\
&= \frac{c_{k,d}}{nh^{d+2}}\sum_{i=1}^{n}\left[\sum_{j=1}^{kn_i}\left(x^j - x_i^j\right)^2\right]' \cdot k'\left(\frac{\sum_{j=1}^{kn_i}\left(x^j - x_i^j\right)^2}{h^2} + \frac{\sum_{j=1}^{unkn_i}\left(x^j-\mu^j\right)^2 + \left(\sigma^j\right)^2}{h^2}\right) \\
&\quad + \left[\sum_{j=1}^{unkn_i}\left(x^j-\mu^j\right)^2 + \left(\sigma^j\right)^2\right]' \cdot k'\left(\frac{\sum_{j=1}^{kn_i}\left(x^j - x_i^j\right)^2}{h^2} + \frac{\sum_{j=1}^{unkn_i}\left(x^j-\mu^j\right)^2 + \left(\sigma^j\right)^2}{h^2}\right).
\end{aligned}$$

In our computation, we will first deal with one coordinate $l$, and then, we will generate the computation for all the other coordinates.

$$\Rightarrow f'_{x^l} = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n_l} \left(x^l - x_i^l\right) \cdot k' \left( \frac{\sum_{j=1}^{kn_i} \left(x^j - x_i^j\right)^2}{h^2} + \frac{\sum_{j=1}^{unkn_i} \left(x^j - \mu^j\right)^2 + \left(\sigma^j\right)^2}{h^2} \right)$$
$$+ \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{m_l} \left(x^l - \mu^l\right) \cdot k' \left( \frac{\sum_{j=1}^{kn_i} \left(x^j - x_i^j\right)^2}{h^2} + \frac{\sum_{j=1}^{unkn_i} \left(x^j - \mu^j\right)^2 + \left(\sigma^j\right)^2}{h^2} \right)$$
$$= \frac{2c_{k,d}}{nh^{d+2}} [x^l \cdot \sum_{i=1}^{n} k' \left( \frac{\sum_{j=1}^{kn_i} \left(x^j - x_i^j\right)^2}{h^2} + \frac{\sum_{j=1}^{unkn_i} \left(x^j - \mu^j\right)^2 + \left(\sigma^j\right)^2}{h^2} \right)$$
$$- \sum_{i=1}^{n_l} x_i^l \cdot k' \left( \frac{\sum_{j=1}^{kn_i} \left(x^j - x_i^j\right)^2}{h^2} + \frac{\sum_{j=1}^{unkn_i} \left(x^j - \mu^j\right)^2 + \left(\sigma^j\right)^2}{h^2} \right)$$
$$- \sum_{i=1}^{m_l} \mu^l \cdot k' \left( \frac{\sum_{j=1}^{kn_i} \left(x^j - x_i^j\right)^2}{h^2} + \frac{\sum_{j=1}^{unkn_i} \left(x^j - \mu^j\right)^2 + \left(\sigma^j\right)^2}{h^2} \right),$$

where there are $n_l$ points for which the $x^l$ coordinate is known, and there are $m_l$ points where it is missing.

$$f'_{x^l} = \frac{2c_{k,d}}{nh^{d+2}} \cdot \left[ \sum_{i=1}^{n} g \left( \sum_{j=1}^{d} MD_E \left(x^j, x_i^j\right)^2 \right) \right]$$
$$\cdot \left[ \frac{\sum_{i=1}^{n_l} x_i^l \cdot g \left( \sum_{j=1}^{d} MD_E \left(x^j, x_i^j\right)^2 \right) + \sum_{i=1}^{m_l} \mu^l \cdot g \left( \sum_{j=1}^{d} MD_E \left(x^j, x_i^j\right)^2 \right)}{\sum_{i=1}^{n} g \left( \sum_{j=1}^{d} MD_E \left(x^j, x_i^j\right)^2 \right)} - x^l \right].$$

As a result, the mean shift vector using the $MD_E$ distance is defined as:

$$m_{MD_E,G}(x) = \frac{\sum_{i=1}^{n_l} x_i^l \cdot g \left( \sum_{j=1}^{d} MD_E \left(x^j, x_i^j\right)^2 \right) + \sum_{j=1}^{m_l} \mu^l \cdot g \left( \sum_{j=1}^{d} MD_E \left(x^j, x_i^j\right)^2 \right)}{\sum_{i=1}^{n} g \left( \sum_{j=1}^{d} MD_E \left(x^j, x_i^j\right)^2 \right)} - x^l. \quad (14)$$

Now, we can use this equation to run the mean shift procedure over datasets with missing values.

## 6. Experiments on numerical datasets

In order to measure performance of the developed clustering algorithm (i.e., k-means and mean shift), we compare their performance on complete datasets to its performance on

| Dataset | Dataset size | Clusters |
|---|---|---|
| Flame | $240 \times 2$ | 2 |
| Jain | $373 \times 2$ | 2 |
| Path-based | $300 \times 2$ | 3 |
| Spiral | $312 \times 2$ | 3 |
| Compound | $399 \times 2$ | 6 |
| Aggregation | $788 \times 2$ | 7 |

**Table 1.** Speech and Image Processing Unit Dataset properties.

incomplete data using the suggested distance function and then again using the existing methods (MCA, MA, and MI) within the standard algorithms.

To measure the similarity between two data clusterings, we decide to use the Rand index [18]. We use it in order to compare the results of the original clustering algorithms to the results of the other derived algorithms for incomplete datasets.

Our experiments use six standard numerical datasets from the Speech and Image Processing Unit [13]; dataset characteristics are shown in **Table 1**.

We produced the missing data by drawing randomly a set consisting of 10–40% of the data from each dataset. These sets are used as samples of incomplete data, where one attribute from each point was randomly selected to be assigned as missing value. For each dataset, we average the results over 10 different runs.

### 6.1. k-Means experiments

In the k-means algorithm, we developed two versions, k-means-$MD_E$ and k-means-$HistMD_E$; to cluster the incomplete datasets, we compare the performance of the k-means ($k$ is fixed for each dataset) clustering algorithm on complete data (i.e., without missing values) to its performance on data with missing values, using the $MD_E$ distance measure (k-means-$MD_E$ and k-means-$HistMD_E$) and then again using k-means-(MCA, MA, and MI).

As can be seen in **Figure 2**, the new algorithms that is based on the $MD_E$ distance outperformed the other existing algorithms on all the datasets. It occurred because in the MA MCA methods, the whole distribution of values is replaced by the mean or the mode of the distribution of known values, that is a fixed value. In our two developed algorithms, we use the distribution of the observed values in all the computation stages. This additional information, taking into account not only the mean of the attribute but also the variance, is probably the reason for the improved performance of our methods compared to the known heuristics.

### 6.2. Mean shift experiments

Mean shift clustering algorithm was tested using bandwidth $h = 4$ (because we saw that the standard mean shift worked well for this value).

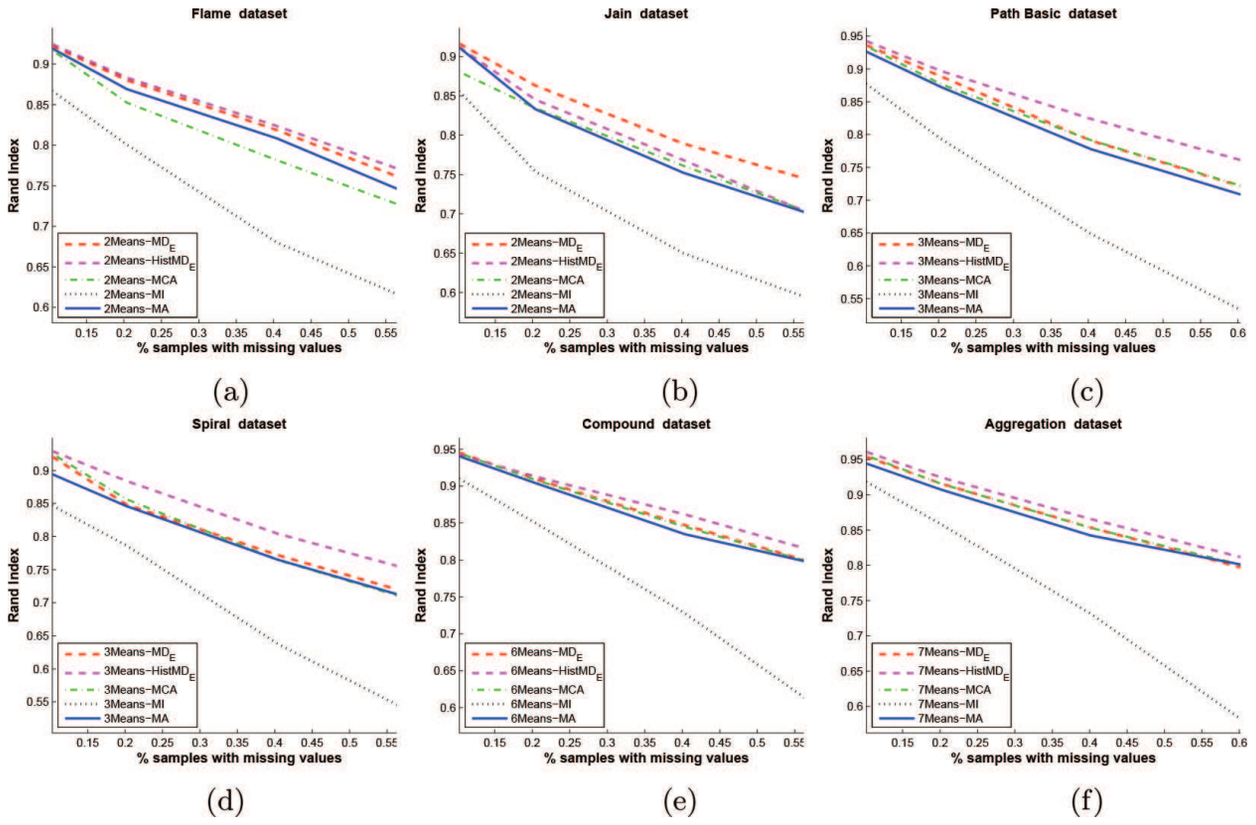


**Figure 2.** Results of k-means clustering algorithm using the different distance functions on the six datasets from the Speech and Image Processing Unit.

A resulting curve for the Rand index values was constructed for each dataset to evaluate how well the algorithm performed.

As can be seen in **Figure 3**, for all the datasets except the Jain dataset, the curves show that the new mean shift algorithm was superior and outperformed the other compared methods for all missing value percentages, while for the Jain dataset, its superiority became apparent only when the percent of the missing values was larger than 25%, as can be seen in **Figure 3(b)**. In addition, we can see that the $MS-MC$ method outperforms the $MS-MA$ method for the flame and path-based datasets, and the $MS-MC$ outperforms $MS-MA$ for the other datasets. As a result, we cannot decide unequivocally which algorithm is better. On the other hand, we obviously can state that the $MS-MD_E$ outperforms the other methods especially when the percentage of the missing values increases.

## 7. Conclusions

Missing values in data are common in real-world applications. They can be caused by human error, equipment failure, system-generated errors, and so on. Several methods were developed

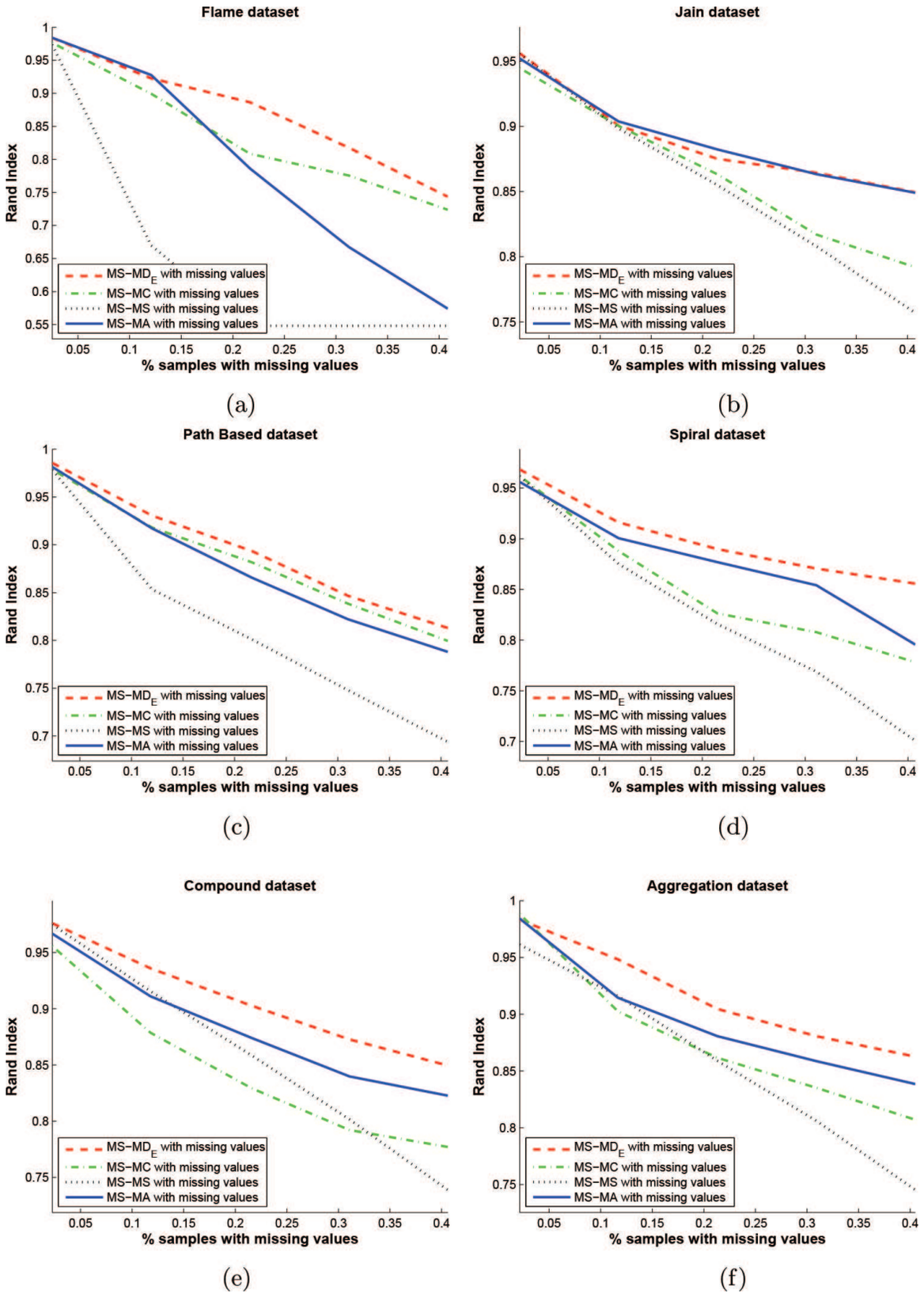


**Figure 3.** Results of mean shift clustering algorithm using the different distance functions on the six datasets from the Speech and Image Processing Unit.

to deal with this problem such as: filling the missing values with fixed values, ignoring sample with missing values, or dealing with the missing values by defining a distance function.

In this work, we have proposed a new mean shift clustering algorithm and two versions of the k-means clustering algorithm over incomplete datasets based on the developed $MD_E$ distance that was presented in [1, 2, 12].

The computational complexities of all the developed algorithms were preserved and they are the same as that of the standard algorithms using the Euclidean distance. The distance was computed based only on the *mean* and *variance* of the data for each attribute.

We experimented on six standard numerical datasets from different fields. On these datasets, we simulated missing values and compared the performance of the developed algorithms using our distance and the suggested mean computations to other three basic methods.

From our experiments, we conclude that the developed methods are more appropriate for measuring the mean, mean shift vector, and weighted mean for objects with missing values, especially when the percent of missing values is large.

## Author details

Loai AbdAllah[1]* and Ilan Shimshoni[2]

*Address all correspondence to: loai1984@gmail.com

1 Department of Information Systems, The Max Stern Yezreel Valley Academic College, Israel

2 Department of Information Systems, University of Haifa, Israel

## References

[1] Abedallah L, Shimshoni I. K-means over incomplete datasets using mean Euclidean distance. In: Proceedings of 12th International Conference on Machine Learning and Data Mining; 2016

[2] Abedallah L, Shimshoni I. Mean shift clustering algorithm for data with missing values. In: Proceedings of 14th International Conference of DaWaK; 2014. pp. 426-438

[3] Donders ART, van der Heijden GJMG, Stijnen T, Moons KGM. Review: A gentle introduction to imputation of missing values. Journal of Clinical Epidemiology. 2006;**59**(10):1087-1091

[4] Ibrahim JG, Chen M-H, Lipsitz SR, Herring AH. Missing-data methods for generalized linear models: A comparative review. Journal of the American Statistical Association. 2005;**100**(469):332-346

[5] Little RJA. Missing-data adjustments in large surveys. Journal of Business & Economic Statistics. 1988;**6**(3):287-296

[6] Little RJA, Rubin DB. Statistical Analysis with Missing Data. Hoboken, New Jersey: John Wiley & Sons; 2014

[7] Zhang S, Qin Z, Ling CX, Sheng S. Missing is useful: Missing values in cost-sensitive decision trees. IEEE Transactions on Knowledge and Data Engineering. 2005;**17**(12):1689-1693

[8] Magnani M. Techniques for dealing with missing data in knowledge discovery tasks. Obtido. 2004;**15**(01):2007. http://magnanim.web.cs.unibo.it/index.html

[9] Jerzy Grzymala-Busse, Ming Hu. A comparison of several approaches to missing attribute values in data mining. In: Proceedings of Rough Sets and Current Trends in Computing; Springer; 2001. pp. 378-385

[10] Zhang S. Shell-neighbor method and its application in missing data imputation. Applied Intelligence. 2011;**35**(1):123-133

[11] Batista G, Monard MC. An analysis of four missing data treatment methods for supervised learning. Applied Artificial Intelligence. 2003;**17**(5–6):519-533

[12] AbdAllah L, Shimshoni I. A distance function for data with missing values and its applications on KNN and k-means algorithms. International Journal Advances in Data Analysis and Classification

[13] Speech University of Eastern Finland and Image Processing Unit. Clustering dataset, http://cs.joensuu.fi/sipu/datasets/; 2008

[14] Hunt L, Jorgensen M. Mixture model clustering for mixed data with missing information. Computational Statistics and Data Analysis. 2003;**41**(3):429-440

[15] Ghahramani Z, Jordan M. Learning from incomplete data. Technical Report, MIT AI Lab Memo, (1509), 1995

[16] Comaniciu D, Meer P. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002;**24**(5):603-619

[17] Georgescu B, Shimshoni I, Meer P. Mean shift based clustering in high dimensions: A texture classification example. In: Proceedings of the 9th International Conference on Computer Vision; 2003. pp. 456-463

[18] Rand WM. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association. 1971;**66**(336):846-850

IntechOpen