POISSON DISTRIBUTION BASED INITIALIZATION FOR FUZZY CLUSTERING

Tomas Vintr[*], *Vanda Vintrova*[†], *Hana Rezankova*[‡]

Abstract: A quality of centroid-based clustering is highly dependent on initialization. In the article we propose initialization based on the probability of finding objects, which could represent individual clusters. We present results of experiments which compare the quality of clustering obtained by k-means algorithm and by selected methods for fuzzy clustering: FCM (fuzzy c-means), PCA (possibilistic clustering algorithm) and UPFC (unsupervised possibilistic fuzzy clustering) with different initializations. These experiments demonstrate an improvement in the quality of clustering when initialized by the proposed method. The concept how to estimate a ratio of added noise is also presented.

Key words: *Poisson distribution based initialization, k-means, fuzzy c-means, possibilistic clustering algorithm, unsupervised possibilistic fuzzy clustering*

Received: February 20, 2012
Revised and accepted: April 27, 2012

1. Introduction

Cluster analysis is represented by a set of multi-dimensional data analysis techniques used for object classification when objects are characterized by values of selected attributes. Methods of cluster analysis belong to a class of methods called *unsupervised learning*. Their aim is to divide a set of objects into subsets so that the similarity of objects within a subset and the difference of objects from different subsets are maximized. These subsets are called *clusters*.

[*] Tomas Vintr
University of Economics, Prague, Faculty of Informatics and Statistics, Department of Statistics and Probability, E-mail: `tomas.vintr@vse.cz`

[†] Vanda Vintrova
University of Economics, Prague, Faculty of Informatics and Statistics, Department of Statistics and Probability, E-mail: `vandav@email.cz`

[‡] Hana Rezankova
University of Economics, Prague, Faculty of Informatics and Statistics, Department of Statistics and Probability, E-mail: `rezanka@vse.cz`

In this paper we focus on *centroid-based clustering.* This class of clustering techniques is intended to cluster objects characterized by values of numeric attributes. As every object is represented by a numerical vector, it can be depicted as a point in a multidimensional space. From this point of view we will use some simplified expressions in the following text. A *position of an object* means a position of a point in the Euclidean space with coordinates representing values of attributes of an object relative to the origin of the coordinates. A *distance of two objects* is calculated as the Euclidean distance of two points at the position of these two objects. *Placing an object to the position* means to generate such an object so that the values of attributes are equivalent to the values of the coordinates of that position in the Euclidean space. A *space of objects* means such a compact subset of the Euclidean space such that every point with coordinates equivalent to the values of the attributes of any object from the set of objects belongs to it.

Since we test the clustering algorithms on generated sets of objects, a *cluster* in this text means a subset of a set of objects that was generated by a certain multivariate normal distribution. A *mean of a cluster* stands for a vector of means of the multivariate normal distribution that generated a cluster of objects.

In the centroid-based clustering techniques, the set of objects is divided into a certain number of nonempty clusters. Every cluster is represented by a vector of mean values of attributes called a *cluster prototype.* By an iterative shifting of objects between the subsets or by changing weights for calculation of the average, the algorithms try to find cluster prototypes similar to the mean of the clusters. There are three well known shortcomings of this class of clustering techniques. The first one is the following: if clusters have strongly different sizes (a number of objects belonging to an individual cluster), then larger clusters attract cluster prototypes and positions of some cluster prototypes in the output of the algorithms are either in a space between clusters or together close to the center of the large cluster, separating the large cluster into parts and losing information about smaller ones [3]. The second shortcoming is that user has to know a number of clusters as it is a parameter set *a priori.* A number of clusters is either estimated by a user who makes an expert estimation or there are different indexes for estimation of this parameter [12]. The third shortcoming is derived from the previous one. This is the problem with an *initialization*, that is a placing of cluster prototypes into a space of objects at the start of the algorithm. An initialization is mostly executed randomly either by placing cluster prototypes on positions of randomly chosen objects from a set of objects or by placing cluster prototypes on random positions in a space of objects.

We will focus on the initialization influence on a quality of clustering. We will test algorithms on datasets with exactly known number of identically numerous clusters generated by a multivariate normal distribution with identical covariance matrices $\mathbf{K} = \sigma^2\mathbf{I}$, where σ^2 is a variance of individual attributes of objects and $\mathbf{I}$ is an identity matrix. Tests will be executed with three types of initialization. The first initialization will place the cluster prototypes on the positions of means of clusters. This will be the referential, “perfect” initialization. The second initialization is the Poisson distribution based initialization and it is principal for this article. The algorithm is presented in the section Initialization. We want to prove that this Poisson distribution based initialization proposed by us is significantly better than widely used random initializations. For comparison, another initialization will be tested: placing cluster prototypes on random positions in a space of objects. A scale of experiments should also present limits of our algorithm and of tested clustering algorithms.

2. Centroid-Based Clustering

The basic method of centroid-based clustering is known as k-means [7], hereinafter HCM (Hard c-Means). The basic idea of HCM is an iterative minimization of the objective function

$$J_{HCM}\left(\mathbf{X};\mathbf{U}_{HCM};\mathbf{C}\right)=\sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}d_{ij}^{2}, \tag{1}$$

where $\mathbf{X}=\left\{\mathbf{x}_1,\ldots,\mathbf{x}_n\right\}$, $\mathbf{X}\subset\mathbf{R}^{dim}$, is a set of objects, $\mathbf{C}=\{C_1,\ldots,C_c\}$ is a set of clusters, $\mathbf{U}_{HCM}=(u_{ij})$ denotes a partition matrix with conditions

$$u_{ij}=\{0,1\}\,,\forall i\in\{1,\ldots,c\}\,,\forall j\in\{1,\ldots,n\}\,, \tag{2}$$

$$\sum_{i=1}^{c}u_{ij}=1,\forall j\in\{1,\ldots,n\}\,, \tag{3}$$

and $d_{ij}=\|\mathbf{x}_j-\mathbf{c}_i\|$ is a certain metric, where $\mathbf{c}_i$ is a cluster prototype of a cluster C_i.

As all objects of a dataset are assigned to clusters, HCM is very sensitive to the presence of outliers. Borders of clusters are between the positions of two neighbor cluster prototypes.

An approach of fuzzy clustering should soften a “sharp” border between clusters. The basic algorithm of fuzzy clustering is FCM (Fuzzy c-Means) [2]. The basic idea of FCM is an iterative minimization of the objective function

$$J_{FCM}\left(\mathbf{X};\mathbf{U}_{FCM};\mathbf{C}\right)=\sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}^{m}d_{ij}^{2}, \tag{4}$$

where $\mathbf{X}=\left\{\mathbf{x}_1,\ldots,\mathbf{x}_n\right\}$, $\mathbf{X}\subset\mathbf{R}^{dim}$, is a set of objects, $\mathbf{C}=\{C_1,\ldots,C_c\}$ is a set of clusters, $\mathbf{U}_{FCM}=(u_{ij})$ denotes a fuzzy partition matrix with conditions

$$\sum_{j=1}^{n}u_{ij}>0,\forall i\in\{1,\ldots,c\}\,, \tag{5}$$

$$\sum_{i=1}^{c}u_{ij}=1,\forall j\in\{1,\ldots,n\}\,, \tag{6}$$

$d_{ij}=\|\mathbf{x}_j-\mathbf{c}_i\|$ is a certain metric, where $\mathbf{c}_i$ is a cluster prototype of a cluster C_i, and $m>1$ is a fuzzifier.

An important parameter of the objective function is the fuzzifier. If its value is set on 1, the FCM algorithm reduces on the HCM algorithm [8]. Neither HCM nor FCM solves the problem of outliers, as the sums in (3) and (6) equal 1.

An algorithm, which aims to deal with a problem of outliers, is PCM (Possibilistic c-Means) [5]. In this algorithm, conditions for a fuzzy partitioning matrix were adjusted in a way that $0 < \sum_{i=1}^{c} u_{ij} \leq c, \forall j \in \{1, \ldots, n\}$, so that outliers have this sum as low as possible. To prevent a trivial solution when all u_{ij} equal 0, the objective function has been adjusted and now contains a controversial parameter. Either we replace the parameter with a constant, but for this replacement we need a profound knowledge of a dataset, or we use a computation recommended by the authors of the algorithm. In such a case the parameter is a function of u_{ij} which causes problem with derivation of the objective function and an impossibility to generate a partitioning matrix. Even though it is possible to generate a partitioning matrix by one or several iterations of the FCM algorithm, the PCM algorithm is computationally unstable [5], [1]. The qualitative problem of PCM is a strong tendency towards coincidences of clusters. If an initialization of two cluster prototypes is realized in a vicinity of the mean value of one cluster, it is highly probable that the positions of both cluster prototypes in the output of the algorithm shall be close to the mean value of this cluster.

The above mentioned algorithms represent the basic ideas of centroid-based clustering with a fixed number of subsets. We have to mention that using the fuzzifier is not the only way to soften sharp borders between clusters [8], but now we will not deal with the other way. We have decided to test only HCM, FCM, a newer, computationally stable version of PCM and the combination of FCM and PCM approaches.

Firstly, authors of PCM changed the objective function and the computation of the controversial parameter [6]. Later, this parameter was replaced by a constant and a new algorithm, called PCA (Possibilistic Clustering Algorithm), was proposed [14]. The objective function of PCA is

$$J_{PCA}\left(\mathbf{X}; \mathbf{U}_{PCA}; \mathbf{C}\right) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}^{2} + \frac{\beta}{m^{2}\sqrt{c}} \sum_{i=1}^{c} \sum_{j=1}^{n} \left(u_{ij}^{m} \ln u_{ij}^{m} - u_{ij}^{m}\right), \qquad (7)$$

where $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, $\mathbf{X} \subset \mathbf{R}^{dim}$, is a set of objects, $\mathbf{C} = \{C_1, \ldots, C_c\}$ is a set of clusters, $\mathbf{U}_{PCA} = (u_{ij})$ denotes a fuzzy partition matrix with conditions

$$\sum_{j=1}^{n} u_{ij} > 0, \forall i \in \{1, \ldots, c\}, \qquad (8)$$

$$0 < \sum_{i=1}^{c} u_{ij} \leq c, \forall j \in \{1, \ldots, n\}, \qquad (9)$$

$d_{ij} = \|\mathbf{x}_j - \mathbf{c}_i\|$ is a selected metric, where $\mathbf{c}_i$ is a cluster prototype of C_i, $m > 1$ is the fuzzifier and β is a positive parameter, which authors define as a *covariance of a set of objects*

$$\beta = \frac{1}{n}\sum_{j=1}^{n} \left\|\mathbf{x}_j - \bar{\mathbf{x}}\right\|^2, \tag{10}$$

where

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{j=1}^{n} \mathbf{x}_j. \tag{11}$$

In the meantime, there were attempts to create an algorithm which would keep good characteristics of FCM and PCM and remove negatives [9], [10]. Required characteristics have been obtained by adding up both objective functions. This principle was copied and used in a new algorithm UPFC (Unsupervised Possibilistic Fuzzy Clustering) [13]. Its objective function is a sum of objective functions of FCM and PCA, i.e.

$$\begin{aligned} J_{UPFC}\left(\mathbf{X};\mathbf{U}_{FCM};\mathbf{U}_{PCA};\mathbf{C}\right) = \sum_{i=1}^{c}\sum_{j=1}^{n}\left(au_{ij,FCM}^{m}d_{ij}^{2} + bu_{ij,PCA}^{m}d_{ij}^{2}\right) + \\ + \frac{\beta}{m^2\sqrt{c}}\sum_{i=1}^{c}\sum_{j=1}^{n}\left(u_{ij,PCA}^{m}\ln u_{ij,PCA}^{m} - u_{ij,PCA}^{m}\right), \end{aligned} \tag{12}$$

where $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, $\mathbf{X} \subset \mathbf{R}^{dim}$, is a set of objects, $\mathbf{C} = \{C_1, \ldots, C_c\}$ is a set of clusters, $\mathbf{U}_{FCM} = (u_{ij,FCM})$ denotes a fuzzy partition matrix with conditions

$$\sum_{j=1}^{n} u_{ij,FCM} > 0, \forall i \in \{1, \ldots, c\}, \tag{13}$$

$$\sum_{i=1}^{c} u_{ij,FCM} = 1, \forall j \in \{1, \ldots, n\}, \tag{14}$$

$\mathbf{U}_{PCA} = (u_{ij,PCA})$ denotes a fuzzy partition matrix with conditions

$$\sum_{j=1}^{n} u_{ij,PCA} > 0, \forall i \in \{1, \ldots, c\}, \tag{15}$$

$$0 < \sum_{i=1}^{c} u_{ij,PCA} \le c, \forall j \in \{1, \ldots, n\}, \tag{16}$$

$d_{ij} = \|\mathbf{x}_j - \mathbf{c}_i\|$ is a selected metric, where $\mathbf{c}_i$ is a volume prototype of C_i, $m, h > 1$ are the fuzzifiers,

$$\beta = \frac{1}{n}\sum_{j=1}^{n} \left\|\mathbf{x}_j - \bar{\mathbf{x}}\right\|^2, \tag{17}$$

where

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{j=1}^{n}\mathbf{x}_j, \tag{18}$$

and parameters a and b are the weights that determine influence of the two original algorithms.

3. Initialization

It was mentioned in the introduction that there are problems of clustering using centroid-based clustering methods when there is a different number of objects in clusters and a number of clusters is badly estimated. We propose an initialization method, firstly mentioned in [11], which, if there is the same number of objects in clusters and a number of clusters is well estimated, improves the quality of the clustering.

3.1 Datasets without noise

Assuming that a number of clusters in a set of objects is c and a number of objects in all clusters is equal. Let us chose a random sample of s objects from a set of objects. The probability of choosing y objects from one randomly chosen cluster can be obtained by the binomial distribution as

$$P(y) = \left(\begin{array}{c} n \\ y \end{array}\right)\left(\frac{1}{c}\right)^{y}\left(1-\frac{1}{c}\right)^{s-y}. \tag{19}$$

With the assumption that c and s are large enough, we can approximate it by the Poisson distribution

$$P(y) = e^{-\lambda}\frac{\lambda^y}{y!}, \tag{20}$$

where $\lambda = \frac{s}{c}$. Equation (20) allows us to estimate the parameter λ for a certain probability that among randomly chosen objects, at least one (resp. at least two) object(s) belong(s) to the *a priori* chosen cluster (see Tab. I).

P($y > 0$)	$\lambda = s/c$	P($y > 1$)	$\lambda = s/c$
0.90	**2.3**	0.90	**3.9**
0.95	**3.0**	0.95	**4.8**
0.99	**4.6**	0.99	**6.6**

Tab. I *Estimated parameter λ for chosen probability.*

If we set, for example, $\lambda = 4.6$ and randomly choose $s = 4.6c$ objects from a set of objects, we have chosen at least one object from every cluster at probability 99%, and for c large enough we can suppose that we have choosen at least one object from 99% of clusters. If we eliminate $(\lambda - 1)c$ redundant objects after choosing s objects from a set of objects, then there remains only one chosen object from (almost) every cluster. It is intuitive that if two points belong to the same cluster, their mutual distance is small, and oppositely if two points belong to two different clusters, their mutual distance is large.

We eliminate the redundant objects iteratively. In every iteration, for every chosen object we calculate a distance to its nearest neighbor and we arrange objects ascendingly according to these distances. We identify a couple of objects from the set of remaining selected objects with the smallest mutual distance. If there are two or more couples of objects with the same smallest mutual distance, then we choose a couple randomly as another couple will be chosen during the next iteration. Then we have to eliminate one of these two objects. We eliminate the one with a smaller sum of distances to all remaining chosen objects. At the end of this procedure, every cluster should contain only one chosen object and, at the same time, at least one object should be in every cluster (see Algorithm 1).

The reason to eliminate the one with a smaller sum of distances is that if there are two coincident clusters or if they are very close to each other, the one with a smaller sum of distances is probably closer to the chosen object from the closest cluster. Firstly, the one with a higher sum of distances represents its own cluster better, because it is hypothetically further from the position of the object selected from the closest cluster. Secondly, if there are two objects from different clusters which are close together, and each of them is the only one in its cluster, then the algorithm could eliminate one of them. The probability of occurrence of this error could be decreased when the positions of these objects are "on the opposite sides" of these clusters. Thirdly, it corresponds to the idea of "maximizing the differences of objects from different subsets".

```
Input: $s$ randomly selected objects from a set of objects
Output: $c$ objects, each representing one cluster
for $i = 1$ to $(\lambda - 1)\, c$ do
    Calculate the set of distances between every pair of selected objects
    $\{d_{ij}\}$;
    Find $min\,\{d_{ij}\} = d_{kl} = d_{lk}; i \neq j$;
    Find $min\,\{\sum d_{il}; \sum d_{ik}\} = \sum d_{im}$;
    Eliminate point $m$;
end
```

Algorithm 1: Elimination of redundant objects.

3.2 Datasets with Noise

Let us have a set of objects with a subset of objects that do not belong to any cluster. Suppose that these objects are in the space of objects generated by the uniform distribution. We will call this subset a *noise* or an *added noise*. Using a part of the algorithm described above, we can estimate an amount of noise in a set of objects. We set λ on a certain value and then we select $s = \lambda c$ objects from a set of objects. We sort them ascendingly according to the distance to the nearest neighbor from the selection. Suppose that within our selection, there are objects whose position has the characteristic that a distance to its nearest neighbor is significantly higher (for more details see Experiments) than is usual for most objects from the selection. We call these objects *lonely points*. A lonely point can be either the only object selected from a certain cluster or a noise. If there are more objects selected from a certain cluster, their mutual distance will be small. We estimate the ratio of the lonely points in the selection (see Fig. 2) and we compare it with the hypothetical ratio (see Fig. 1) for chosen value of parameter λ. If there is a larger amount of the lonely points, we suppose that there is an added noise in the dataset.

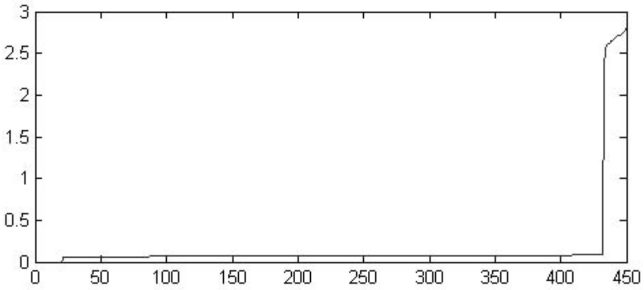


Fig. 1 *Graph of the distances from every selected object to its closest neighbor. Distance values (vertical axis) are sorted in an increasing order (serial numbers are on horizontal axis). Dataset with no added noise,* $c = 150$, $n = 7500$, $\lambda = 3$, $dim = 64$.

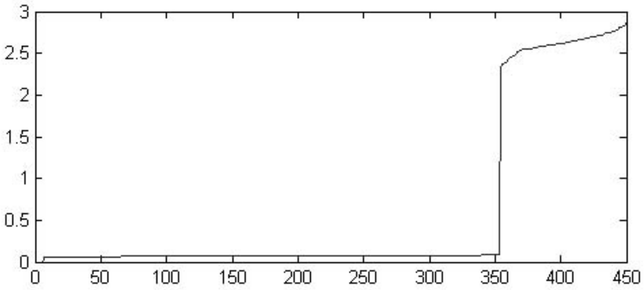


Fig. 2 *Graph of the distances of every selected object to its closest neighbor. Values of distances (vertical axis) are sorted in an increasing order (serial numbers are on horizontal axis). Dataset with 20 % added noise,* $c = 150$, $n = 9000$, $\lambda = 3$, $dim = 64$.

According to this measurement or more measurements with different λ we are able to manually estimate the ratio p of the noise. Afterwards we have to select $s = \lambda \frac{c}{1-p}$, where λ is large enough, objects from the set of objects and eliminate $100p\%$ of these objects with the largest distances to their nearest objects. Then we continue with elimination of redundant objects algorithm (see Algorithm 1).

4. Experiments

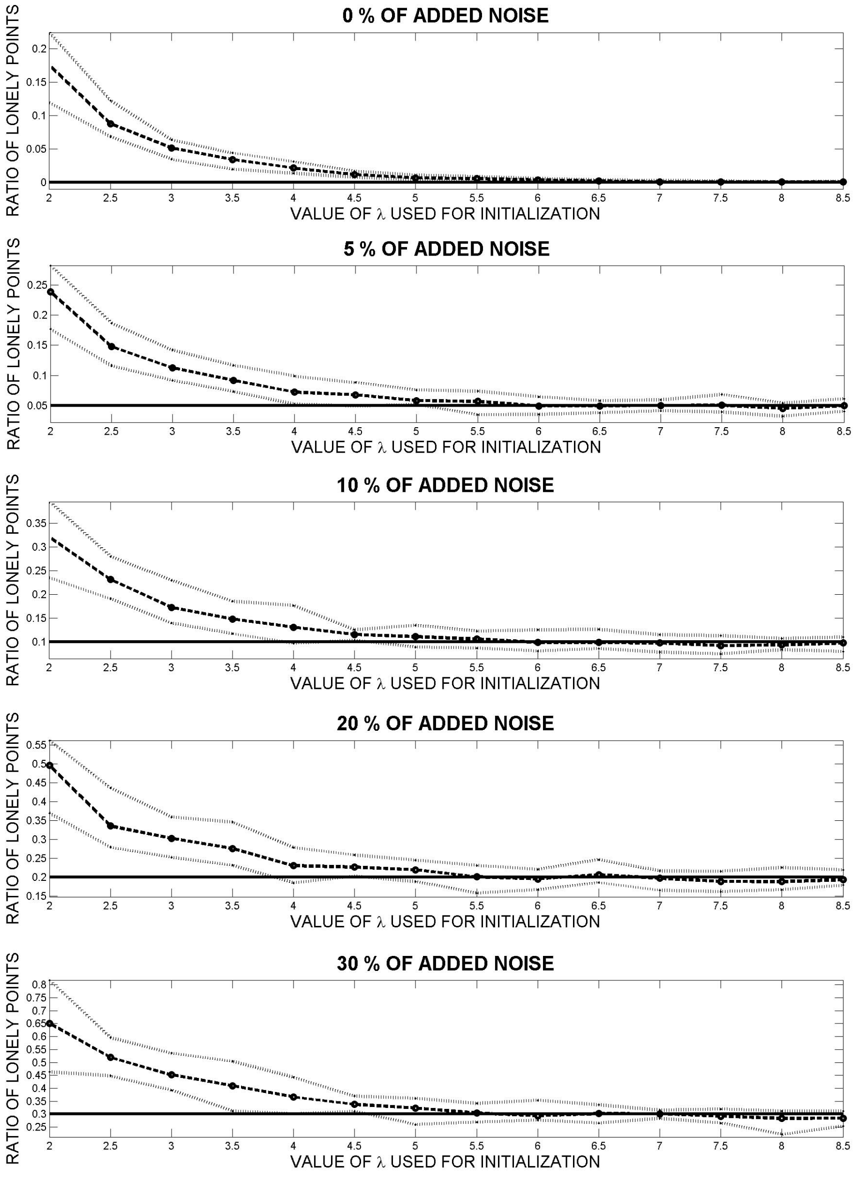


Fig. 3 *Dependency of the ratio of a number of lonely points in the selection to a number of objects, which are not lonely points, on the value of parameter λ. The solid line is the ratio of a number of added noise to a number of objects in clusters. The dashed line depicts the arithmetic mean and the dotted lines depict the range of the measurements of the ratio of lonely points.*

The algorithms were tested on different sets of objects. Every object of a set of objects is represented by a numeric vector of a relevant dimension. To generate a set of objects, we firstly generated vectors by a multivariate uniform distribution in a space of (hyper)cube with a center in the origin of coordinates and with a length of a side 1 (this (hyper)cube is understood as a space of objects). These vectors are understood as vectors of means of multivariate normal distribution, which consequently generates a given number of vectors. These vectors are understood as objects of the set of objects and form the required number of clusters.

The aim of the first experiment (Fig. 3) is to verify the estimation of the parameter λ (Tab. I) and to demonstrate the possibility to estimate the ratio of added noise. The sets of objects for this experiment are generated with the paramenters set as follows: the number of attributes of every object $dim = 64$, the number of clusters $c = 150$, the standard deviation of individual attributes of objects $\sigma = 0.00\bar{6}$ and the number of objects in every cluster 150. There are two parameters that are changing: the ratio of the number of objects belonging to the added noise to the number of objects belonging to the clusters and the value of the parameter λ.

There are five charts in the figure. Every chart depicts a dependency of the ratio of a number of lonely points in the selection to a number of objects, which are not lonely points, on the value of parameter λ. In these tests, the lonely points are objects from the selection whose distance to the nearest neighbor is more than twice larger than the median of all distances of selected objects to their nearest neighbor. Every ratio of lonely points was counted as an arithmetic mean of ten measurements on ten different sets of objects with the same value of λ. Ratios of lonely points are depicted as points connected by the dashed line. The solid line is a ratio of added noise. There are also minima and maxima of measurements (dotted lines). Every chart depicts outputs of measurements on the sets of objects with a different ratio of added noise (0%–30%, from the top to the bottom chart).

The aim of the further experiments (Figs. 5–12) is to compare the quality of clustering of the selected algorithms using different initializations and to demonstrate qualities of the Poisson distribution based initialization. The basic setting of sets of objects generation for these experiments is as follows: the number of attributes of every object $dim = 2$, the number of clusters $c = 150$, the standard deviation of individual attributes of objects $\sigma = 0.00\bar{6}$, the ratio of a number of added noise to a number of objects belonging to clusters 0% and the number of objects in every cluster 150 (Fig. 4). In every test one parameter is changing; for every value of this selected parameter a set of clustering is executed. Algorithms HCM, FCM, PCA and UPFC are applied. Every algorithm executes a clustering till it reaches a termination condition, nevertheless the minimal number of iterations is 11 and the maximal number of iterations is 100, after that it is terminated. Fuzzifiers in algorithms that contain them are set on $m, h = 2$, the parameters a and b in the algorithm UPFC are set on 1.

All the algorithms use the Euclidean metric to compute distances between objects. Every algorithm is tested three times for every setting, every time with different initialization. These initializations have the same starting positions of the volume prototypes for all the four tested algorithms for the given parameter setting of the set of objects. First initialization of volume prototypes is on positions

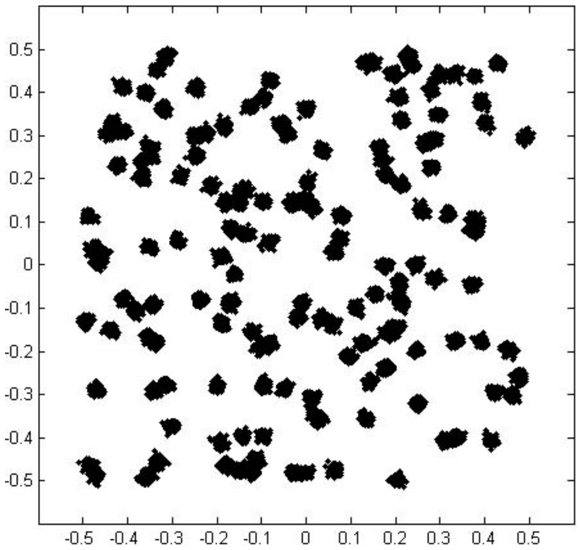


Fig. 4 *An example of a generated set of objects with basic setting. Depicted to the Euclidean space.*

of means of clusters. Second initialization is on random positions within a cube containing mean values of clusters. Third initialization is the Poisson distribution based initialization with parameter $\lambda = 6$ ($s = \lambda\frac{c}{1-p}$).

The quality of clustering is assessed using two criteria. If a mean of a cluster lies within 95% confidence interval of the cluster prototype which should represent the cluster after a termination of an algorithm, then this cluster prototype is labeled as a *well found center*. A cluster is labeled as a *not found cluster*, if no cluster prototype that has not been assigned to any cluster lies smaller than 3σ distance from a mean of this cluster. The results of the experiments are displayed in the series of charts.

There are four pairs of figures (Figs. 5 and 6, 7 and 8, 9 and 10, 11 and 12). Each of them represents the outputs of the experiments using one parameter as a variable. The first figure of each pair (Figs. 5, 7, 9, 11) depicts a dependency of the ratio of well found centers on the value of the variable. The second figure of each pair (Figs. 6, 8, 10, 12) depicts a dependency of the ratio of not found clusters on the value of the variable.

For every figure stands that in the top charts are the results of the algorithms using the initialization in positions of mean values of clusters, in the middle chart are the results when using the initialization on random positions in the space of objects and in the bottom chart are the results when using the Poisson distribution based initialization. For all these charts triangles signify results of the initialization, circles results of HCM, squares results of FCM, stars results of PCA and diamonds results of UPFC.

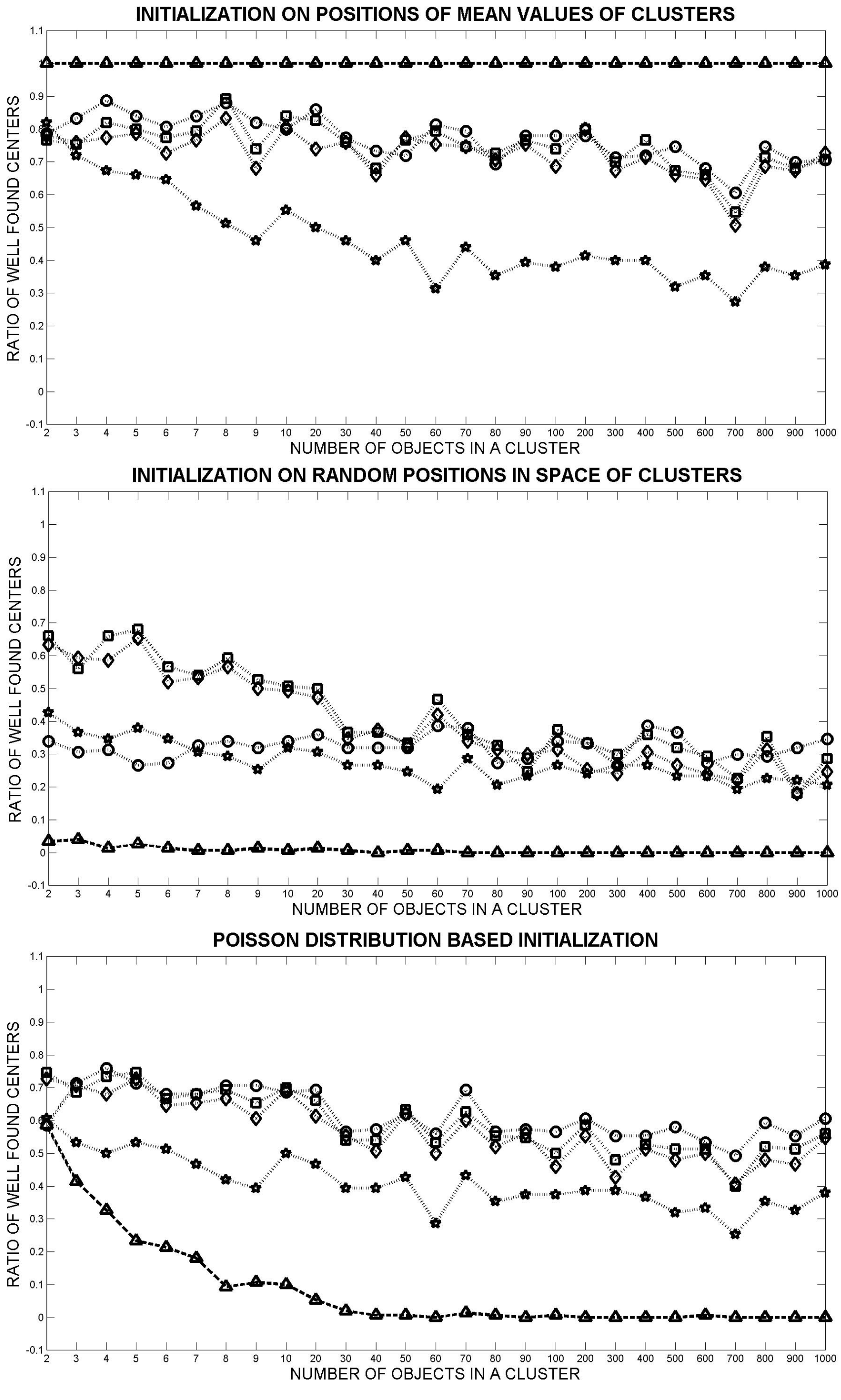


Fig. 5 *The test of influence of a number of objects in a cluster on the quality of clustering. Dependency of the ratio of well found centers on a number of points in a cluster.*

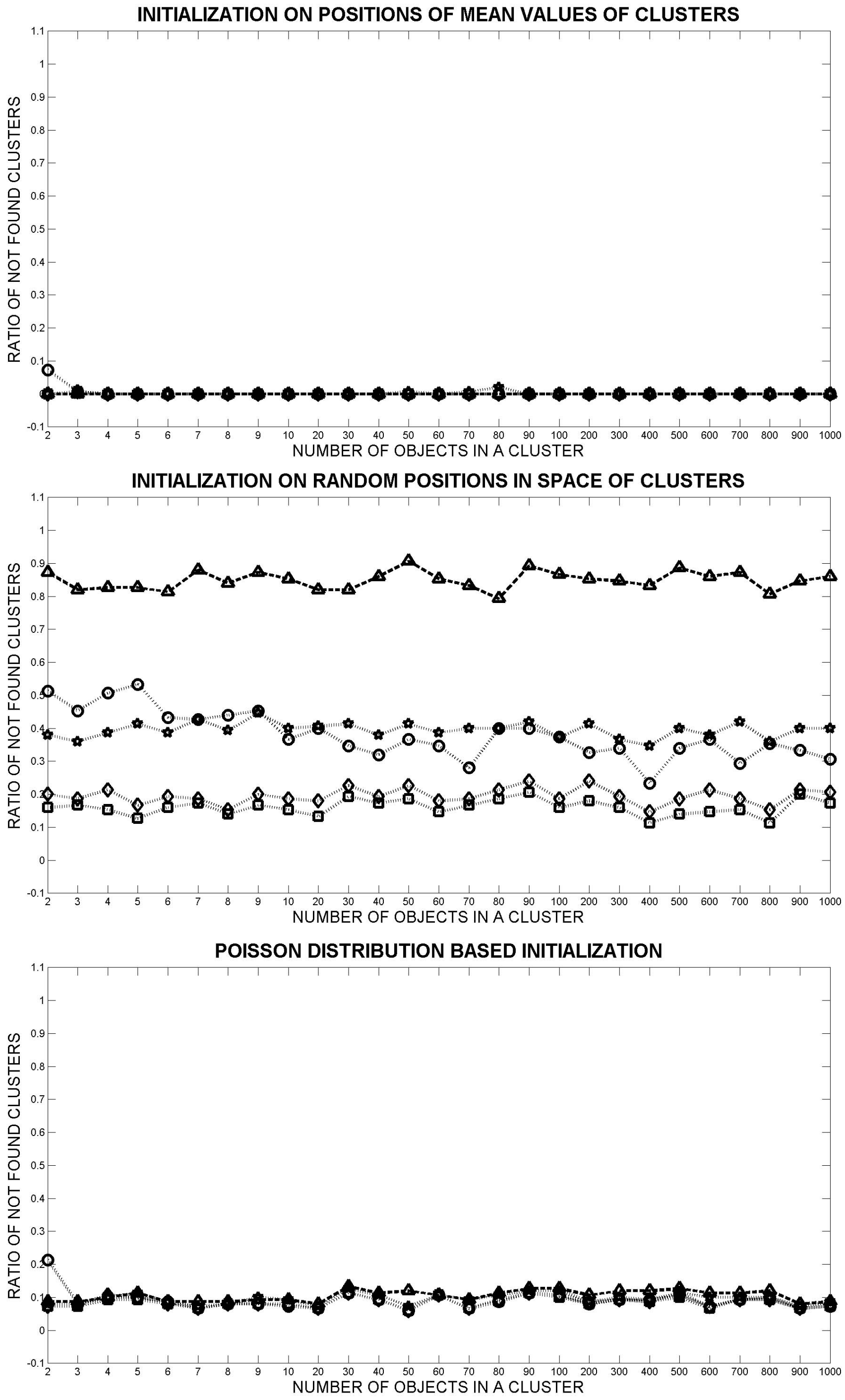


Fig. 6 *The test of influence of a number of objects in a cluster on the quality of clustering. Dependency of the ratio of not found clusters on a number of points in a cluster.*

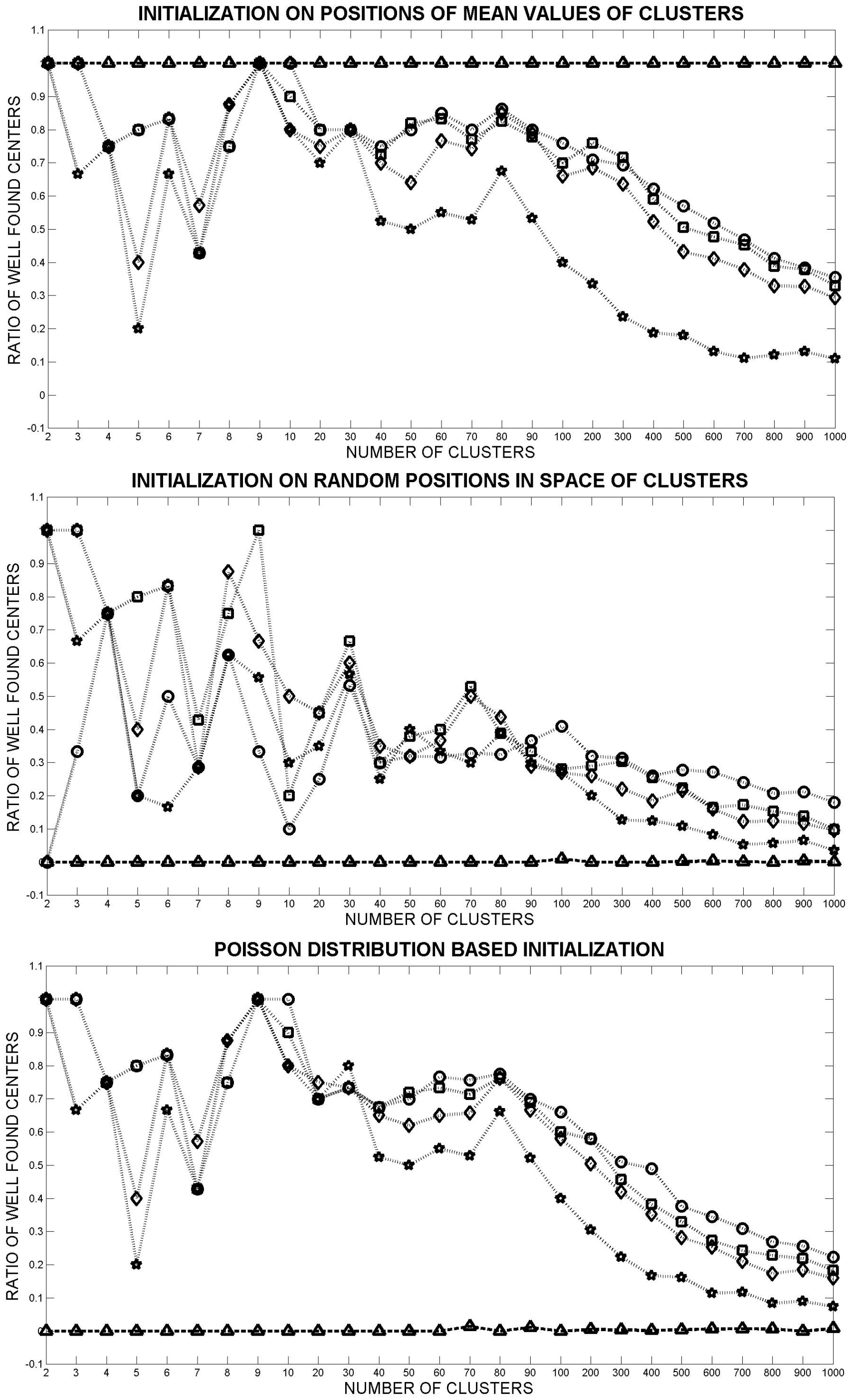


Fig. 7 *The test of influence of a number of clusters on the quality of clustering. Dependency of the ratio of well found centers on a number of clusters.*

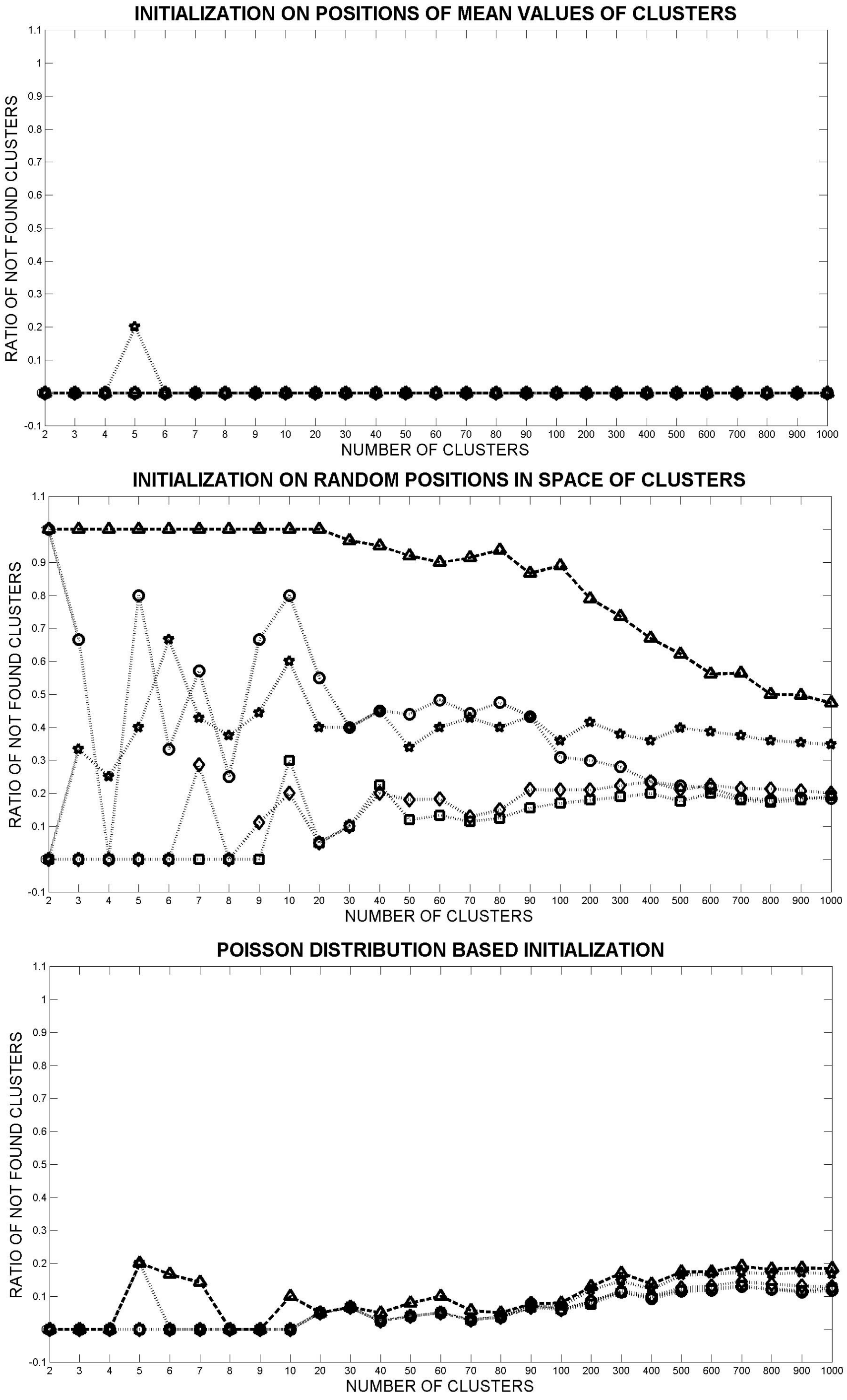


Fig. 8 *The test of influence of a number of clusters on the quality of clustering. Dependency of the ratio of not found clusters on a number of clusters.*

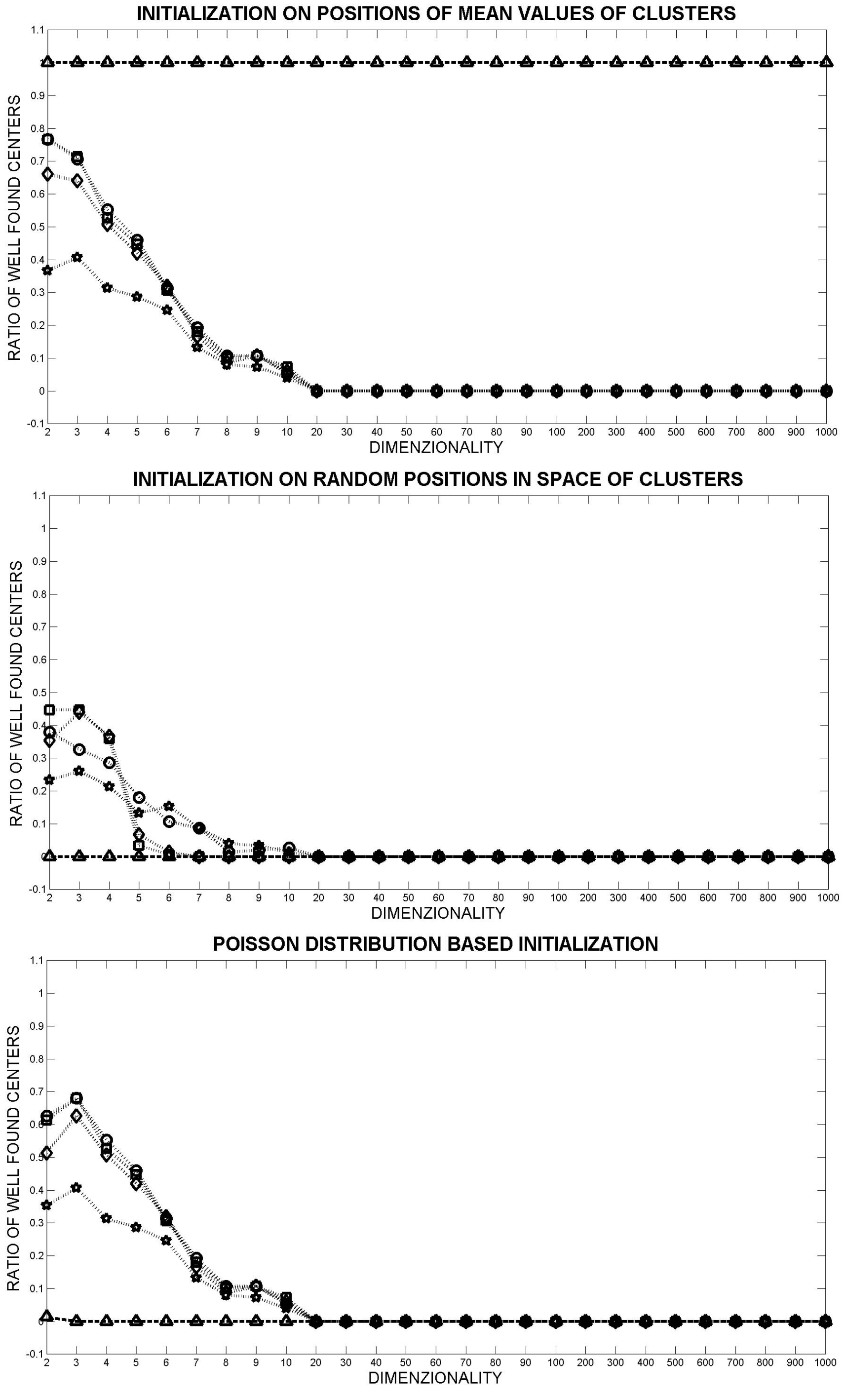


Fig. 9 *The test of influence of a dimensionality on the quality of clustering. Dependency of the ratio of well found centers on dimensionality.*

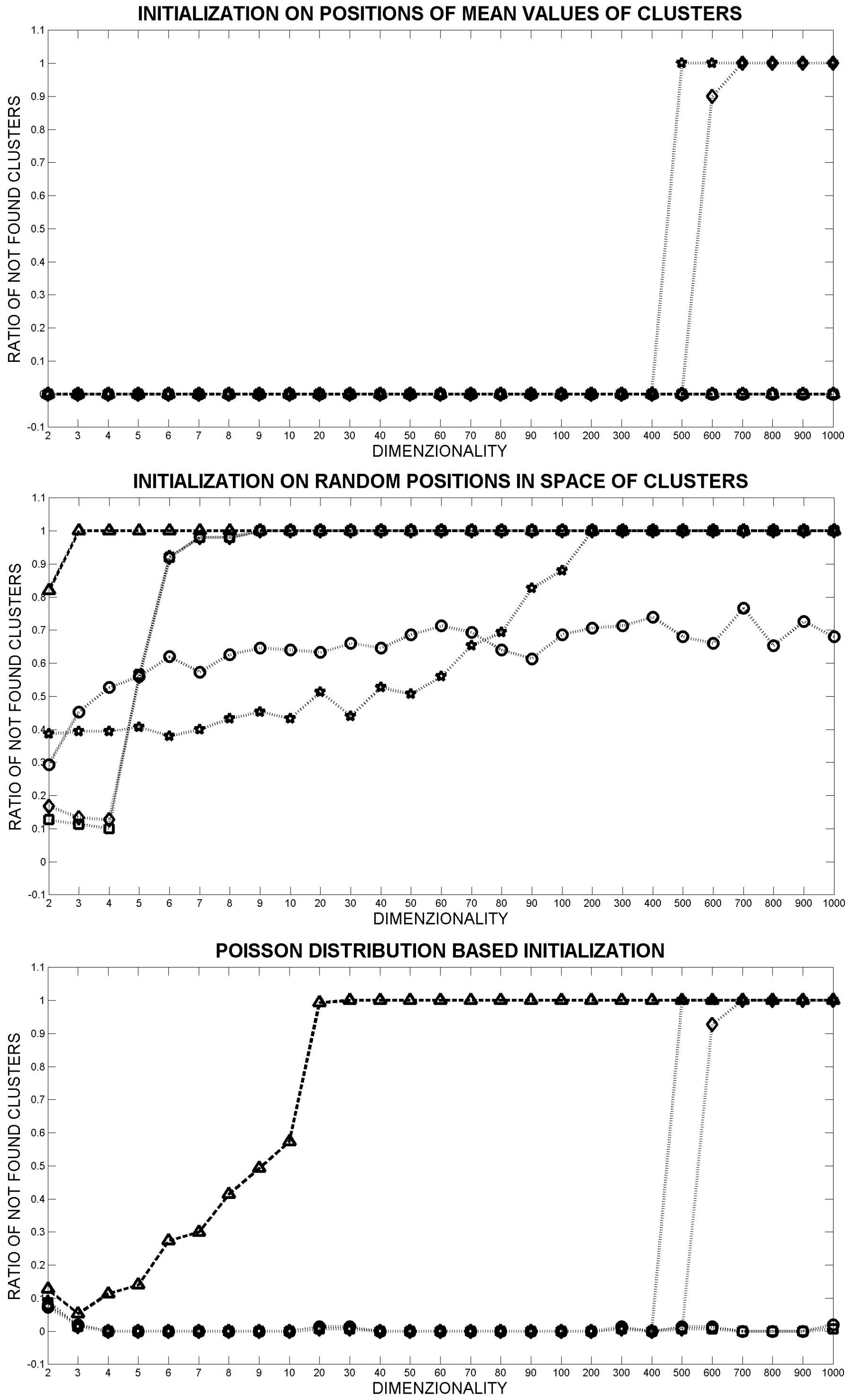
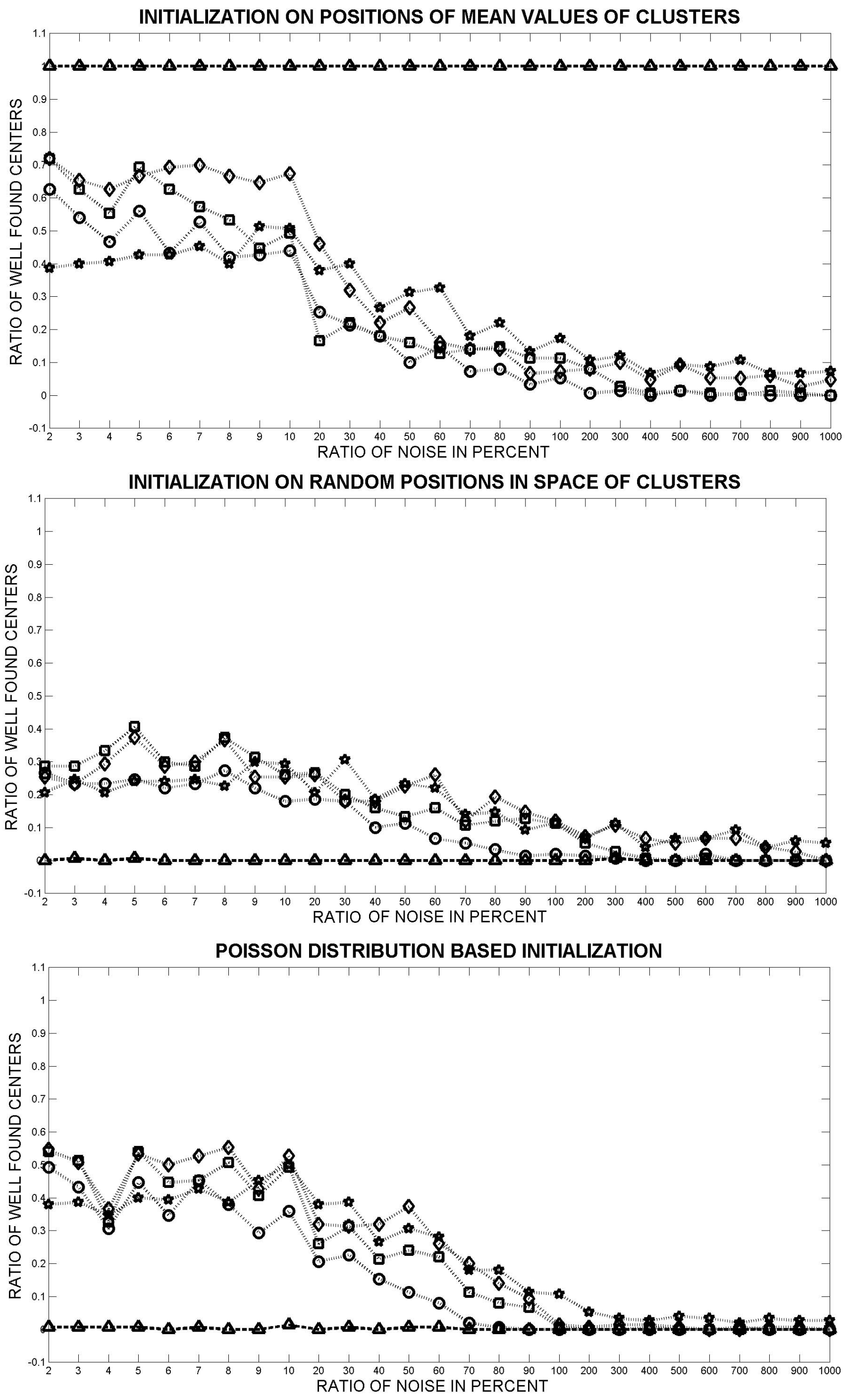


Fig. 10 *The test of influence of a dimensionality on quality of clustering. Dependency of the ratio of not found clusters on dimensionality.*



INITIALIZATION ON POSITIONS OF MEAN VALUES OF CLUSTERS

INITIALIZATION ON RANDOM POSITIONS IN SPACE OF CLUSTERS

POISSON DISTRIBUTION BASED INITIALIZATION

Fig. 11 *The test of influence of a ratio of added noise on the quality of clustering. Dependency of the ratio of well found centers on the ratio of added noise.*

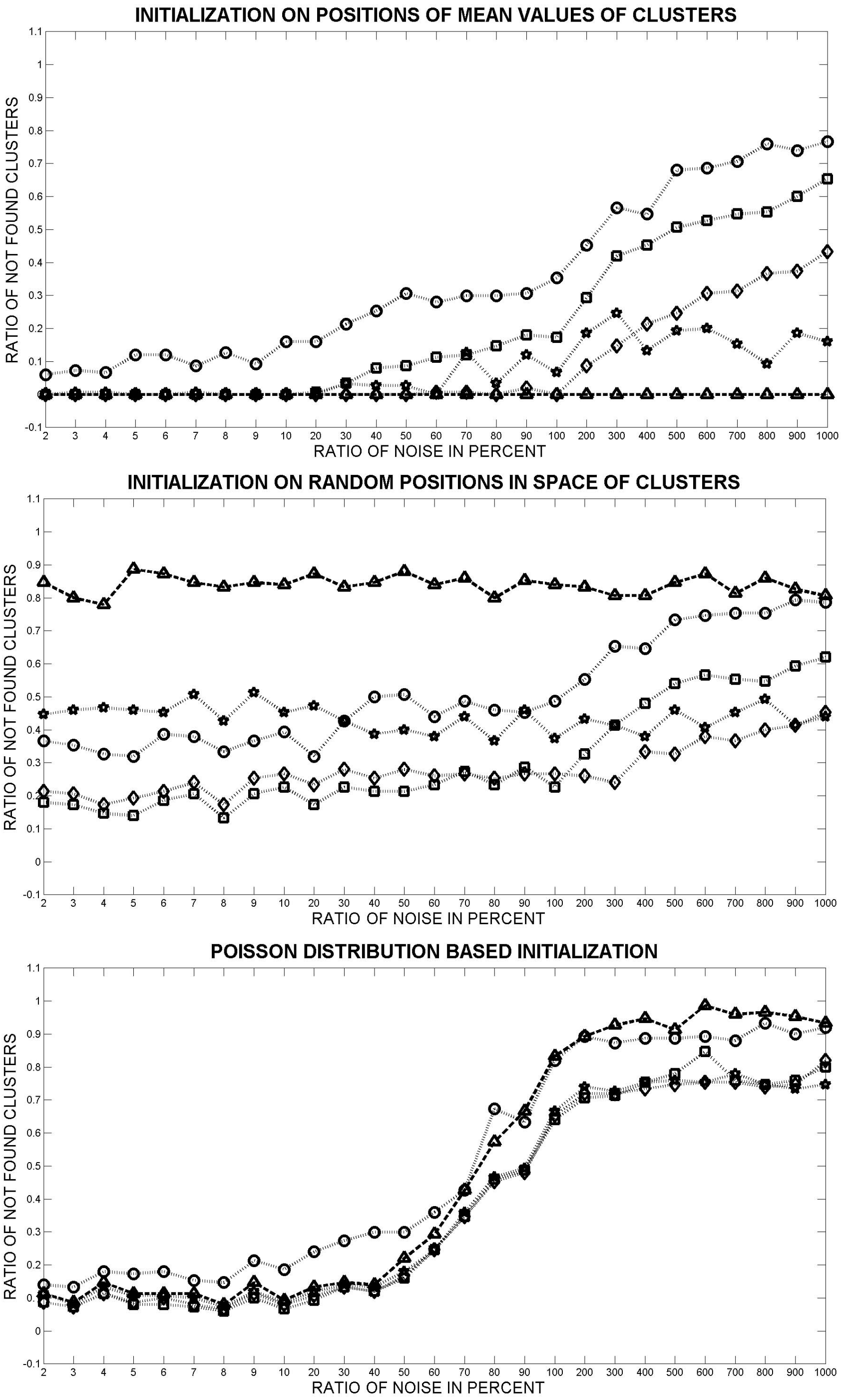


Fig. 12 *The test of influence of the ratio of added noise on the quality of clustering. Dependency of the ratio of not found clusters on the ratio of added noise.*

In the first experiment (Figs. 5 and 6), a number of objects in clusters is changing. In the second experiment (Figs. 7 and 8), a number of clusters is changing. In the third experiment (Figs. 9 and 10), a dimension of objects is changing. In the fourth experiment (Figs. 11 and 12), the variable is a ratio of the number of objects belonging to the added noise to the number of objects belonging to the clusters (2–1000%, where 1000% means that the amount of a noise is ten times larger than amount of the basic set of objects – the clusters).

5. Conclusion and Future Work

We have presented a new initialization method suitable for centroid-based clustering methods. There are five experiments which demonstrate the quality of this initialization. The first of them demonstrates that for higher values of λ the ratio of lonely points converges to the ratio of added noise. We have to note that this experiment is only a proof of a concept and that the outputs of this experiment do not represent a general way of how to estimate the ratio of added noise.

The four following experiments demonstrate the suitability of the initialization for centroid-based clustering methods, using it to explore the sets of objects which are similar to the tested ones. Firstly, we have to point out that the presented algorithms using this initialization were very successful in finding clusters. The results were comparable to the results of clustering with the known positions of the centers of the clusters for every setting of the sets of objects except for the sets of objects with the ratio of added noise higher than 0.5. Only for the ratio of added noise larger than 1 the results were worse than the results of the random initialization. The ratio of well found centers was always better than the ratio when using the random initialization (except for high dimensionality, where the ratio was zero for all the tested initializations).

In the outputs of these experiments, there are two interesting findings which were not subjects of the research. The quality of clustering of the set of objects with a small number of clusters is highly unpredictable, even if the cluster prototypes are initialized in the true centers of the clusters. This is very strange because these algorithms have been usually (and successfully) tested on the sets of objects with a small number of clusters. The second interesting finding is that the small number of objects in the clusters does not negatively affect the quality of clustering and that the quality of clustering is practically independent of the number of objects in clusters.

In the future work, we would like to apply fuzzy clustering methods with the Poisson distribution based initialization on the maps of autonomous robot navigation [4]. We expect that there should be a large number of small clusters. Also, we would like to improve the initialization for more general purposes, such as estimation of number of clusters or finding positions of different sized clusters.

Acknowledgement

This work was supported by project IGA VSE F4/6/2012.

References

[1] Barni M., Cappellini V., Mecocci A.: Comments on a possibilistic approach to clustering, IEEE Transactions on Fuzzy Systems, **4**, 3, 1996, pp. 393-396.

[2] Bezdek J. C.: Pattern Recognition with Fuzzy Objective Function Algorithms, Kluwer Academic Publishers, Heidelberg, 1981.

[3] Kaymak U., Setnes M.: Extended fuzzy clustering algorithms, ERIM report series Research in Management, 2000, No. ERS-2000-51-LIS.

[4] Krajnik T., Faigl J., Vonasek V., Kosnar K., Kulich M., Preucil L.: Simple, yet stable bearing-only navigation, Journal of Field Robotics, **27**, 5, 2010, pp. 511-533.

[5] Krishnapuram R., Keller J. M.: A possibilistic approach to clustering, IEEE Transactions on Fuzzy Systems, **1**, 2, 1993, pp. 98-110.

[6] Krishnapuram R., Keller J. M.: The possibilistic c-means algorithm: insights and recommendations, IEEE Transactions on Fuzzy Systems, **4**, 3, 1996, pp. 385-393.

[7] MacQueen J. B.: Some methods for classification and analysis of multivariate observations, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281-297.

[8] Miyamoto S., Ichihashi H., Honda K.: Algorithms for Fuzzy Clustering, Springer, 2008.

[9] Pal N. R., Pal K., Bezdek J. C.: A mixed c-means clustering model, Proceedings of the Sixth IEEE International Conference on Fuzzy Systems, 1997, pp. 11-21.

[10] Pal N. R., Pal K., Keller J. M., Bezdek J. C.: A possibilistic fuzzy c-means clustering algorithm, IEEE Transactions on Fuzzy Systems, **13**, 2005,4, pp. 517-530.

[11] Vintr T., Pastorek L., Vintrova V., Rezankova H.: Batch FCM with volume prototypes for clustering high-dimensional datasets with large number of clusters, Proceedings of the Third World Congress on Nature and Biologically Inspired Computing (NaBIC), 2011, pp. 427-432.

[12] Wang Y., Li C., Zuo Y.: A selection model for optimal fuzzy clustering algorithm and number of clusters based on competitive comprehensive fuzzy evaluation, IEEE Transactions on Fuzzy Systems, **17**, 3, 2009, pp. 568-577.

[13] Wu X., Wu B., Sun, J., Fu, H.: Unsupervised possibilistic fuzzy clustering, Journal of Information & Computational Science, **7**, 5, 2010, pp. 1075-1080.

[14] Yang M. S., Wu K. L.: Unsupervised possibilistic clustering, Pattern Recognition, **39**, 1, 2006, pp. 5-21.