Int. J. Appl. Math. Comput. Sci., 2010, Vol. 20, No. 1, 123–134
DOI: 10.2478/v10006-010-0009-3

A COMPLETE GRADIENT CLUSTERING ALGORITHM FORMED WITH KERNEL ESTIMATORS

PIOTR KULCZYCKI [*,**], MAŁGORZATA CHARYTANOWICZ [*,***]

[*] Systems Research Institute, Center for Stochastic Data Analysis Methods
Polish Academy of Sciences, ul. Newelska 6, 01–447 Warsaw, Poland
e-mail: {kulczycki,malgorzata.charytanowicz}@ibspan.waw.pl

[**] Department of Automatic Control and Information Technology, Faculty of Electrical and Computer Engineering
Cracow University of Technology, ul. Warszawska 24, 31–155 Cracow, Poland
e-mail: kulczycki@pk.edu.pl

[***] Institute of Mathematics and Computer Science
John Paul II Catholic University of Lublin, ul. Konstantynów 1 H, 20–708 Lublin, Poland
e-mail: mchmat@kul.lublin.pl

The aim of this paper is to provide a gradient clustering algorithm in its complete form, suitable for direct use without requiring a deeper statistical knowledge. The values of all parameters are effectively calculated using optimizing procedures. Moreover, an illustrative analysis of the meaning of particular parameters is shown, followed by the effects resulting from possible modifications with respect to their primarily assigned optimal values. The proposed algorithm does not demand strict assumptions regarding the desired number of clusters, which allows the obtained number to be better suited to a real data structure. Moreover, a feature specific to it is the possibility to influence the proportion between the number of clusters in areas where data elements are dense as opposed to their sparse regions. Finally, the algorithm—by the detection of one-element clusters—allows identifying atypical elements, which enables their elimination or possible designation to bigger clusters, thus increasing the homogeneity of the data set.

Keywords: data analysis and mining, clustering, gradient procedures, nonparametric statistical methods, kernel estimators, numerical calculations.

1. Introduction

Consider an m-element set of n-dimensional vectors:

$$x_1, x_2, \ldots, x_m \in \mathbb{R}^n. \tag{1}$$

Generally, the task of clustering relies upon the division of the above data set into subsets (clusters), each containing elements similar to one another, yet significantly differing from elements of other subsets (for a basic notion, see (Anderberg, 1973; Jain and Dubes, 1988; Everitt *et al*., 2001)). Such a comfortable, intuitively obvious definition is equally awkward both from a theoretical and a practical point of view, as it contains visible and hidden imprecisions. Above all there is no unambiguous definition of what denotes "similarity" (and, consequently, "difference") of elements, nor is it clear if the number of clusters is to be arbitrarily assumed or defined as a result of the structure of real data (1) itself, or how to measure the quality of the divisions imposed. If it is also taken into account that the mathematical apparatus does not have a natural methodology for solving such problems, the existence becomes obvious of an excessive number of heuristic iterative procedures, each of them characterized by different advantages and disadvantages, as well as certain properties which may be of benefit in some problems and of no profit in others.

Another kind of difficulty in the use of such heuristic iterative procedures is that many of these methods were developed 30–40 years ago, when computer access was the privilege of a selected group of specialists, in possession of deep knowledge necessary for comprehensive analysis of the obtained results. In many cases, a lot of conditions and parameters—among others such fundamentals as a stop criterion or the assumed number of

clusters—are left for the user to decide. A basis for the analysis of collected results was often presented by viewing their graphical representation, not easy even when using specialized visualization methods, particularly in the multidimensional case when $n > 2$.

The above is now in complete contradiction to the demands of the majority of contemporary users of these methods. The availability of computer technology means that they often do not have specialist knowledge for comprehensive analysis. The best solution for these users is to provide a complete algorithm, taking into account "automatic" procedures for fixing all quantities appearing there, both functions and parameters, as well as clear information concerning their influence on the obtained results and—in consequence—advantages and disadvantages arising from their potential changes.

In the now classic paper (Fukunaga and Hostetler, 1975), the authors formulated a natural idea of clustering, employing notable possibilities entering into widespread use of statistical kernel estimators at that time, today the main method of nonparametric estimation. The basis of the above concept is treating the data set (1) as a random sample obtained from an n-dimensional random variable, calculating the kernel estimator of the density of its distribution, and making the clear assumption that particular clusters correspond to modes (local maxima) of the estimator, and so "valleys" of the density function constitute a bordering of such clusters. The presented method was formulated as a general idea only, leaving the details—as was the generally accepted behavior of that time—to the painstaking analysis of the user. Its naturalness and clarity of interpretation allowed the method to be applied in many varied specialist tasks such as tracking, image segmentation, information fusion, and video processing (see (Zhang *et al*., 2005) for a list of examples), interesting mutations and supplements (see, e.g., (Yang *et al*., 2003) or (Cheng, 1995)), and even unwitting repetition of the same idea (Wang *et al*., 2004).

The aim of this paper is to present gradient clustering algorithm based on Fukunaga and Hostetler's concept in its complete form, suitable for direct use without requiring users to have a deeper statistical knowledge or to conduct laborious research. All parameters appearing here can be effectively calculated using convenient numerical procedures based on optimization criteria. Moreover, making use of a near-intuitive interpretation of the concept of the gradient algorithm itself, as well as its theoretical base—kernel estimators, an illustrative analysis of the significance of particular parameters will be given, and the effects achieved through their possible change with respect to the above mentioned optimal values, depending on conditions of the problem in question and user preferences.

The main feature of the algorithm under research is that it does not demand strict assumptions regarding the desired number of clusters, which allows the obtained number to be better suited to a real data structure. In the paper, the parameter directly responsible for the number of clusters will be indicated. At a preliminary stage its value can be calculated effectively using optimization criteria. It will also be shown how possible changes to this value (which may be performed but are not necessary) influence the increase or decrease in the number of clusters, although without defining their exact number. Moreover, the next parameter is indicated, and its value will influence the proportion between the number of clusters in dense and sparse areas of data set elements. Here also its value can be assumed based on optimization ground, or possibly subject to modifications with the goal of increasing the number of clusters in dense areas of data set elements while simultaneously reducing or even eliminating them from sparse regions, or vice-versa. This possibility is particularly worth underlining as practically non-existent in other clustering procedures.

By its nature, the algorithm commonly creates one-element clusters, which can be treated as atypical elements (Barnett and Lewis, 1994) in a given configuration of clusters. This could be the basis for eliminating elements which create them in order to increase the homogeneity of the data set. However, by the above mentioned modification of the appropriate parameter, leading to a reduction in the number of clusters in sparse areas, these elements may also be assigned to the closest clusters.

Moreover, the appropriate relation between the above mentioned two parameters permits a reduction or even elimination of clusters in sparse areas, usually without influencing the number of clusters in dense areas of data set elements.

The complete gradient clustering algorithm proposed in this paper has, of course, its application limits, mainly that it is not intended for tasks where the desired number of clusters is strictly defined. The calculation time becomes relatively great, which may cause difficulties with its use in tasks carried out in real time. For multidimensional problems, i.e., when $n > 5$, it might prove necessary to apply first procedures for the reduction of dimensionality. Typical concepts are described in the book (Larose, 2006, Chapter 1) and those dedicated to algorithms of data analysis and mining using the kernel estimators methodology, based on the simulated annealing method, are presented in (Kulczycki and Łukasik, 2009) and will soon be the subject of further research.

A preliminary version of this article was presented as (Kulczycki and Charytanowicz, 2008).

2. Statistical kernel estimators

Consider an n-dimensional random variable X with a distribution characterized by the density f. Its kernel estimator $\hat{f} : \mathbb{R}^n \to [0, \infty)$, calculated using experimentally obtained values for the m-element random sample

$x_1, x_2, \ldots, x_m$, in its basic form is defined as

$$\hat{f}(x) = \frac{1}{mh^n} \sum_{i=1}^{m} K\left(\frac{x - x_i}{h}\right), \tag{2}$$

where $m \in \mathbb{N}\backslash\{0\}$, the coefficient $h > 0$ is called a smoothing parameter, while the measurable function $K : \mathbb{R}^n \to [0, \infty)$ of unit integral $\int_{\mathbb{R}^n} K(x)\mathrm{d}x = 1$, symmetrical with respect to zero and having a weak global maximum in this place, is called the kernel. The choice of the kernel K and the calculation of the smoothing parameter h is made most often with the criterion of the mean integrated square error.

Thus, the choice of the kernel form has—from a statistical point of view—no practical meaning and, thanks to this, it becomes possible to take into account primarily properties of the obtained estimator (e.g., its class of regularity, assigned positive values) or aspects of calculations, advantageous from the point of view of the application problem under investigation (for a broader discussion, see (Kulczycki, 2005, Section 3.1.3; Wand and Jones, 1994, Section 2.7 and Section 4.5) and also (Muller, 1984) for individual aspects). The most popular in practice is the normal kernel:

$$K(x) = \frac{1}{2\pi^{n/2}} \exp\left(-\frac{x^{\mathrm{T}}x}{2}\right). \tag{3}$$

Note that it is differentiable to any degree and assumes positive values in the whole domain.

The fixing of the smoothing parameter h has significant meaning for the estimation quality. Too small a value causes a large number of local extremes of the estimator $\hat{f}$ to appear, which is contrary to the actual properties of real populations. On the other hand, too big values of the parameter h result in the overflattening of this estimator, hiding specific properties of the distribution under investigation. According to the universal cross-validation method (Kulczycki, 2005, Section 3.1.5; Silverman, 1986, Section 3.4.3), it can be calculated as a value realizing the minimum of the function $g : (0, \infty) \to \mathbb{R}$ in the form

$$g(h) = \frac{1}{m^2h^n} \sum_{i=1}^{m} \sum_{j=1}^{m} \widetilde{K}\left(\frac{x_j - x_i}{h}\right) + \frac{2}{mh^n} K(0), \tag{4}$$

while $\widetilde{K}(x) = K^{*2}(x) - 2K(x)$, whereas K^{*2} denotes a convolution square of the function K. For the normal kernel (3), we have

$$K^{*2}(x) = \frac{1}{(4\pi)^{n/2}} \exp\left(-\frac{x^{\mathrm{T}}x}{4}\right). \tag{5}$$

A range of other methods of calculating the smoothing parameter have been investigated under specific conditions. In particular, for the one-dimensional case one can recommend the simple and effective plug-in method (Kulczycki, 2005, Section 3.1.5; Wand and Jones, 1994, Section 3.6.1), although the above described universal cross-validation method can also be applied here.

In the case of the basic definition of the kernel estimator (2), the influence of the smoothing parameter on particular kernels is the same. Advantageous results are obtained thanks to the individualization of this effect, achieved through the so-called modification of the smoothing parameter. It relies on mapping the positive modifying parameters $s_1, s_2, \ldots, s_m$ on particular kernels, described as

$$s_i = \left(\frac{\hat{f}_*(x_i)}{\overline{s}}\right)^{-c}, \tag{6}$$

where $c \in [0, \infty)$, $\hat{f}_*$ denotes the kernel estimator without modification, $\overline{s}$ is the geometrical mean of the numbers $\hat{f}_*(x_1)$, $\hat{f}_*(x_2)$, …, $\hat{f}_*(x_m)$ and, finally, defining the kernel estimator with the modification of the smoothing parameter in the following form:

$$\hat{f}(x) = \frac{1}{mh^n} \sum_{i=1}^{m} \frac{1}{s_i^n} K\left(\frac{x - x_i}{hs_i}\right). \tag{7}$$

Thanks to the above procedure, the areas in which the kernel estimator has small values (e.g., in the range of "tails") are additionally flattened, and the areas connected with large values are peaked, which permits to better reveal individual properties of the distribution. The parameter c stands for the intensity of the modification procedure. Based on indications for the criterion of the integrated mean square error, the value

$$c = 0.5 \tag{8}$$

can be suggested.

Owing to the basic form of the kernel estimator (2), the smoothing parameter has the same influence on particular coordinates of this variable. Taking into account the possibility of sizable differences in scales of the above coordinates, for some of these the value of the parameter may turn out to be too small, whereas for others—too big. Because of this, a linear transformation is applied:

$$X = RY, \tag{9}$$

where the matrix R is positive definite. In practice, its two main forms are used: diagonal,

$$R = \begin{bmatrix} \sqrt{\mathrm{Var}(X_1)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sqrt{\mathrm{Var}(X_n)} \end{bmatrix}, \tag{10}$$

and general,

$$R = \sqrt{\mathrm{Cov}(X)}, \tag{11}$$

where $\mathrm{Var}(X_i)$ means the variance of the i-th coordinate, while $\mathrm{Cov}(X)$ stands for the covariance of the variable X. Following the transformation (9), the kernel estimator takes the form

$$\hat{f}(x) = \frac{1}{mh^n \det(R)} \sum_{i=1}^{m} K\left(R^{-1}\frac{x - x_i}{h}\right). \tag{12}$$

As a result, the scales of the particular coordinates become equal, while in the case of the general form (11), additionally, the shapes of kernels stretch out in a direction defined by proper correlation coefficients.

The naturalness and clarity of the kernel estimator concept allow us to easily adapt its properties to conditions of an investigated problem, e.g., by restricting the support of the function $\hat{f}$. The case of the left-sided boundary of a one-dimensional random variable, i.e., the condition $\hat{f}(x) = 0$ is to be fulfilled for every $x < x_*$, with any $x_* \in \mathbb{R}$ fixed, will be presented below. The concept of the proposed procedure consists of realizing a symmetrical "reflection" with respect to the boundary x_* of the fragment of any i-th kernel lying beyond the interval $[x_*, \infty)$ and treating it as a fragment of a kernel "caught" in the symmetrical "reflection" of the element x_i with respect to the boundary x_*, thus at the point $x_* - (x_i - x_*)$, so that $2x_* - x_i$. The basic form of the kernel estimator (2) may then be described as

$$\hat{f}(x) = \frac{1}{mh} \sum_{i=1}^{m} \chi_{[x_*,\infty)}(x) \left[K\left(\frac{x - x_i}{h}\right) + K\left(\frac{x + x_i - 2x_*}{h}\right) \right], \tag{13}$$

where $\chi_{[x_*,\infty)}$ denotes the characteristic function of the interval $[x_*, \infty)$. The parts of the particular kernels "cut off" beyond the assumed support are therefore "completed" inside the support in the direct neighborhood of the boundary, and so in the range of error most often accepted in practice.

The concepts described by (7), (12) and (13) can be joined in a natural manner.

Detailed information regarding kernel estimators is presented in the monographs (Kulczycki, 2005; Silverman, 1986; Wand and Jones, 1994). Examples of practical applications can be found in the publications (Kulczycki, 2007; 2008).

3. Complete gradient clustering algorithm

As in Introduction, consider an m-element set of n-dimensional vectors (1). It will be treated as a random sample obtained from the n-dimensional random variable X, with distribution having a density f. Using the methodology described in Section 2, a kernel estimator $\hat{f}$ can be created. Let us make a natural assumption that particular clusters are related to its modes, or local maxima of the function $\hat{f}$, and mapping onto them elements of the set (1) is realized by transposing those elements in the gradient direction $\nabla\hat{f}$, with an appropriate fixed step.

The above is carried out iteratively with the gradient clustering algorithm (Fukunaga and Hostetler, 1975), based on the classic Newton procedure (Kincaid and Cheney, 2002, Section 3.2), defined as

$$x_j^0 = x_j \quad \text{for } j = 1, 2, \ldots, m, \tag{14}$$

$$x_j^{k+1} = x_j^k + b\frac{\nabla\hat{f}(x_j^k)}{\hat{f}(x_j^k)} \quad \text{for } j = 1, 2, \ldots, m \text{ and } k = 0, 1, \ldots, k^*, \tag{15}$$

where $b > 0$ and $k^* \in \mathbb{N}\backslash\{0\}$. In practice, it is recommended that

$$b = \frac{h^2}{n+2} \tag{16}$$

(Fukunaga and Hostetler, 1975)[1].

In order to refine the above concept to the state of a complete algorithm, the following aspects need to be formulated and analyzed in detail:

1. formula for the kernel estimator $\hat{f}$,
2. setting a stopping condition (and, consequently, the number of steps k^*),
3. definition of a procedure for creating clusters and assigning to them particular elements of the set (1), after the last, k^*-th step,
4. analysis of the influence of the values of parameters on the obtained results.

The above tasks are the subjects of the following sections.

3.1. Formula of the kernel estimator. For the needs of further parts of the concept presented here, the kernel estimator $\hat{f}$ is assumed in a form with the modification of the smoothing parameter of standard intensity (8), as is linear transformation using the diagonal form of the matrix (10)[2].

[1]For ease of computations one can make use of $\nabla\hat{f}(x)/\hat{f}(x) = \nabla\ln(\hat{f}(x))$. Moreover, the value of this expression is sometimes obtained by computing the so-called mean shift—in this case, the gradient clustering algorithm is known in the literature as the mean shift algorithm (procedure); see, for example, (Cheng, 1995; Comaniciu and Meer, 2002; Yang *et al.*, 2003). The method of evaluating the above expression is of no relevance for further parts of the presented material.

[2]Using the general form of the transformation matrix (11) results in elongating kernels in one direction. This causes a difference in the rate of the convergence of the algorithm (14)–(15) with respect to the direction of the transposition of elements of the set (1), unjustified from the point of view of the clustering task and, consequently, interfering with the obtained results. Also for this reason, the product kernel (Kulczycki, 2005, Section 3.1.3; Wand and Jones, 1994, Section 4.2), very useful in practical applications, was rejected.

The kernel K is recommended in the normal form (3) due to its differentiability in the whole domain, convenience for analytical deliberations connected with the gradient, and assuming positive values, which in every case prevents from division by zero in the formula (15).

3.2. Setting a stop condition. It is assumed that the algorithm (14)–(15) should be finished, if after the consecutive k-th step the following condition is fulfilled:

$$|D_k - D_{k-1}| \leq aD_0, \tag{17}$$

where $a > 0$ and

$$D_0 = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} d(x_i, x_j), \tag{18}$$

$$D_{k-1} = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} d(x_i^{k-1}, x_j^{k-1}), \tag{19}$$

$$D_k = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} d(x_i^k, x_j^k), \tag{20}$$

while d means a Euclidean metric in $\mathbb{R}^n$. Therefore, D_0 and D_{k-1}, D_k denote sums of distances between particular elements of the set (1) before starting the algorithm as well as after the $(k-1)$-th and k-th steps, respectively. Primarily, it is recommended that

$$\alpha = 0.001. \tag{21}$$

A potential decrease in this value does not significantly influence the obtained results, although increases require individual verification of their correctness. The convergence of the above algorithm is proven in Appendix.

Finally, if after the k-th step the condition (17) is fulfilled, then

$$k^* = k \tag{22}$$

and, consequently, this step is treated as the last one.

3.3. Procedure for creating clusters and assigning particular elements to them. At this stage, the following set is investigated:

$$x_1^{k^*}, x_2^{k^*}, \ldots, x_m^{k^*}, \tag{23}$$

consisting of the elements of the set (1) after the k^*-th step of the algorithm (14)–(15). Following this, the set of mutual distances of the above elements

$$\left\{ d(x_i^{k^*}, x_j^{k^*}) \right\}_{\substack{i=1,2,\ldots,m-1 \\ j=i+1,i+2,\ldots,m}} \tag{24}$$

should be defined. Its size is given as

$$m_d = \frac{m(m-1)}{2}. \tag{25}$$

Taking (24) as a sample of a one-dimensional random variable, the auxiliary kernel estimator $\hat{f}_d$ of mutual distances of the elements of the set (23) ought to be calculated. Regarding the methodology of kernel estimators presented in Section 2, the normal kernel (3) is once again proposed, as is the use of the procedure of smoothing parameter modification with a standard value of the parameter (8), and additionally left-sided boundary of a support to the interval $[0, \infty)$.

The next task is to find—with suitable precision—the "first" (i.e., for the smallest value of an argument) a local minimum of the function $\hat{f}_d$ belonging to the interval $(0, D)$, where

$$D = \max_{\substack{i=1,2,\ldots,m-1 \\ j=i+1,i+2,\ldots,m}} d(x_i, x_j). \tag{26}$$

For this purpose, one should treat the set (24) as a random sample, calculate its standard deviation σ_d, and next take in sequence the values x from the set

$$\left\{0.01\sigma_d, 0.02\sigma_d, \ldots, [\text{int}(100D) - 1]0.01\sigma_d\right\}, \tag{27}$$

where $\text{int}(100D)$ denotes the integer part of the number $100D$, until finding the first (the smallest) of them which fulfils the condition

$$\hat{f}_d(x - 0.01\sigma_d) > \hat{f}_d(x) \text{ and } \hat{f}_d(x) \leq \hat{f}_d(x + 0.01\sigma_d). \tag{28}$$

This valueă[3] will be denoted hereinafter as x_d, and it can be interpreted as half the distance between "centers" of potential clusters lying closest together.

Finally, the clusters will be created. To this aim, one should:

1. Take the element of the set (23) and initially create a one-element cluster containing it.

2. Find an element of the set (23) different from the one in the cluster, closer than x_d; if there is such an element, then it should be added to the cluster, otherwise, proceed to Point 4.

3. Find an element of the set (23) different from elements in the cluster, closer than x_d to at least one of them; if there is such an element, then it should be added to the cluster and Point 3 repeated.

4. Add the obtained cluster to a "list of clusters" and remove from the set (23) elements of this cluster; if this so-reduced set (23) is not empty, return to Point 1, otherwise, finish the algorithm.

[3] If such a value does not exist, then one should recognize the existence of one cluster and finish the procedure. A similar suggestion may be made for the irrational, yet formally possible case where $m = 1$, as the set (24) is then empty.

The "list of clusters" so defined contains all clusters marked out in the above procedure. Therefore it becomes the complete gradient clustering algorithm in the basic form—its possible modifications and their influence on the obtained results will be presented in the next section.

3.4. Analysis of the influence of the values of parameters on the obtained results. It is worth repeating that the presented clustering algorithm did not require a preliminary, often arbitrary in practice, assumption concerning the number of clusters—their size depending solely on the internal structure of data, given as the set (1). In the application of the complete gradient clustering algorithm in its basic form, the values of the parameters used are effectively calculated taking optimization reasons into account. However, optionally—if the researcher makes a decision—by an appropriate change in values of kernel estimator parameters, it is possible to influence the size of the number of clusters, and also the proportion of their appearance in dense areas in relation to sparse regions of elements in this set.

In the example presented now, the elements of the set (1) have been generated pseudorandomly, for a distribution selected specially to highlight the above aspects.

As mentioned in Section 2, too small a value of the smoothing parameter h results in the appearance of too many local extremes of the kernel estimator, while too great a value causes its excessive smoothing. In this situation lowering the value of the parameter h with respect to that obtained by procedures based on the criterion of the mean integrated square error creates, as a consequence, an increase in the number of clusters. At the same time, an increase in the smoothing parameter value results in fewer clusters. It should be underlined that in both cases, despite having an influence on the size of the cluster number, their exact number will still depend solely on the internal structure of data. Based on research carried out, one can recommend a change in the value of the smoothing parameter by between -25% and $+50\%$. Outside this range, the obtained results require individual verification.

Figure 1 shows the illustratively chosen sample set of two-dimensional vectors. When applying the smoothing parameter value calculated with the cross-validation method (see Section 2), three clusters are obtained. Following a decrease in this value by 25%, their number grows to four, as one cluster divides in two. On the other hand, a 50% increase results in the identification of only two clusters, with two of the original clusters uniting.

Next, as mentioned in Section 2, the intensity of the modification of the smoothing parameter is implied by the value of the parameter c, given as standard by the formula (8). Its increase smoothes the kernel estimator in areas where elements of the set (1) are sparse and also sharpens it in dense areas—in consequence, if the value of the parameter c is raised, then the number of clusters in sparse areas of data decreases, while at the same time increasing in dense regions. Inverse effects can be seen in the case of lowering this parameter value. Based on research carried out we can recommend the value of the parameter c to be between 0 (meaning no modification) and 1.5. An increase greater than 1.5 requires individual verification of the validity of the obtained results. Particularly, it is recommended that $c = 1$.

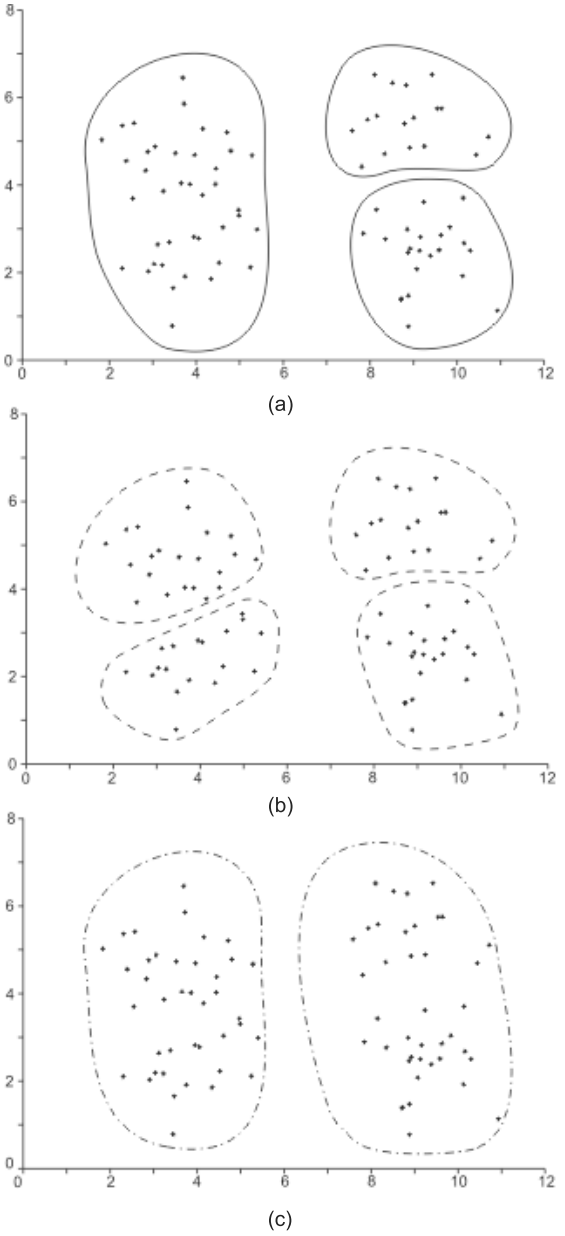

Fig. 1. Effects of changing the value of the smoothing parameter h: calculated by the cross-validation method (a), lowered by 25% (b), raised by 50% (c).

Figure 2 shows an illustrative sample set of two-dimensional vectors. When the standard value $c = 0.5$ is applied, four clusters are obtained—two in dense ar-

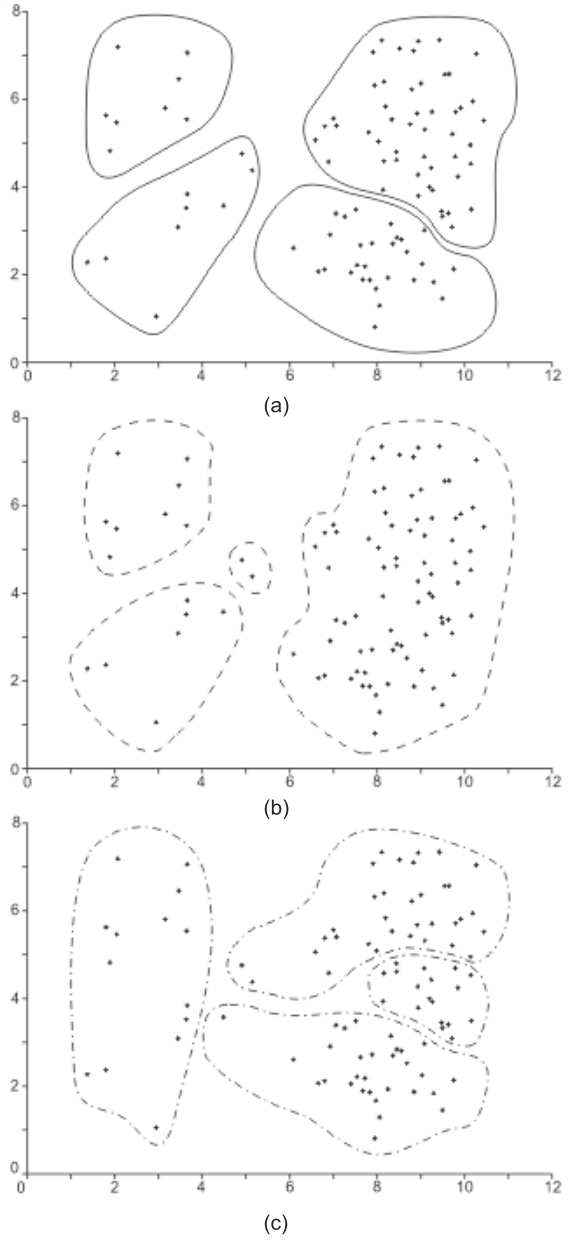

Fig. 2. Effects of differentiation intensity of smoothing parameter modification through changes in the value of the parameter c: standard value $c = 0.5$ (a), value lowered to $c = 0$ (b), value increased to $c = 1$ (c).

eas and two in sparse regions. If $c = 0$, there is no change in the number of clusters; however, the clusters in dense areas coalesce, and an additional cluster appears in sparse regions. Similarly, when $c = 1$, the number of clusters remains the same, but in dense areas the number increases to three, while decreasing to one in sparse regions.

Practice, however, often prevents changes to clusters in dense areas of data—the most important from an applicational point of view—while at the same time requiring a reduction or even elimination of clusters in sparse regions, as they frequently pertain to atypical elements (outliers) commonly arising due to various errors. Putting the above together, one can propose an increase in both the standard scale of the smoothing parameter modification (8) and the value of the smoothing parameter h calculated on the criterion of the mean integrated square error, to the value h^* defined by the formula

$$h^* = \left(\frac{3}{2}\right)^{c-0.5} h. \tag{29}$$

The joint action of both these factors results in a twofold smoothing of the function $\hat{f}$ in the regions where the elements of the set (1) are sparse. Meanwhile, these factors more or less compensate for each other in dense areas, thereby having small influence on the detection of these clusters. Based on research carried out, one can recommend a change in the value of the parameter c from 0.5 to 1.0. Increasing it to above 1.0 demands individual verification of the validity of the obtained results. Particularly it is recommended that $c = 0.75$.

Figure 3 once more shows an illustrative sample set of two-dimensional vectors. In the case of the standard value of the parameter $c = 0.5$, four clusters appear—two in dense areas and two in sparse regions. When $c = 0.75$ was assumed and, consequently, $h^* = (3/2)^{0.25} h \approx 1.11h$, the first two clusters remained unchanged, but the two peripheral ones united. For $c = 1$ and $h^* = 1.22h$ this was also then eliminated.

Finally, it is worth mentioning a possibility of reducing the set (24). In practice, it is too large not only because of the square dependence regarding the size of the set (1), occurring in the formula (25), but also due to the fact that the estimator $\hat{f}_d$ concerns a one-dimensional random variable, while $\hat{f}$, usually multidimensional, by nature demanding notably greater a sample size. For a very large size of the sample (24) it is worth using data-compression procedures well-known in literature, see, e.g., (Girolami and He, 2003; Pal and Mitra, 2004, Section 2.5).

4. Summary and application examples

The subject of this article is the gradient clustering algorithm, based on the natural assumption that if one treats a data set, given as n-dimensional vectors, as a sample obtained from an n-dimensional random variable, then particular clusters correspond to modes (local maxima) of its density estimator, while assigning particular data set elements to them takes place by transposing those elements in the direction of the density function gradient. A complete form was presented, suitable for direct use without requiring a deeper statistical knowledge or laborious research by users.

A basic characteristic of the complete gradient clustering algorithm investigated here is that a fixed number of clusters is not required, just an indication of its size,

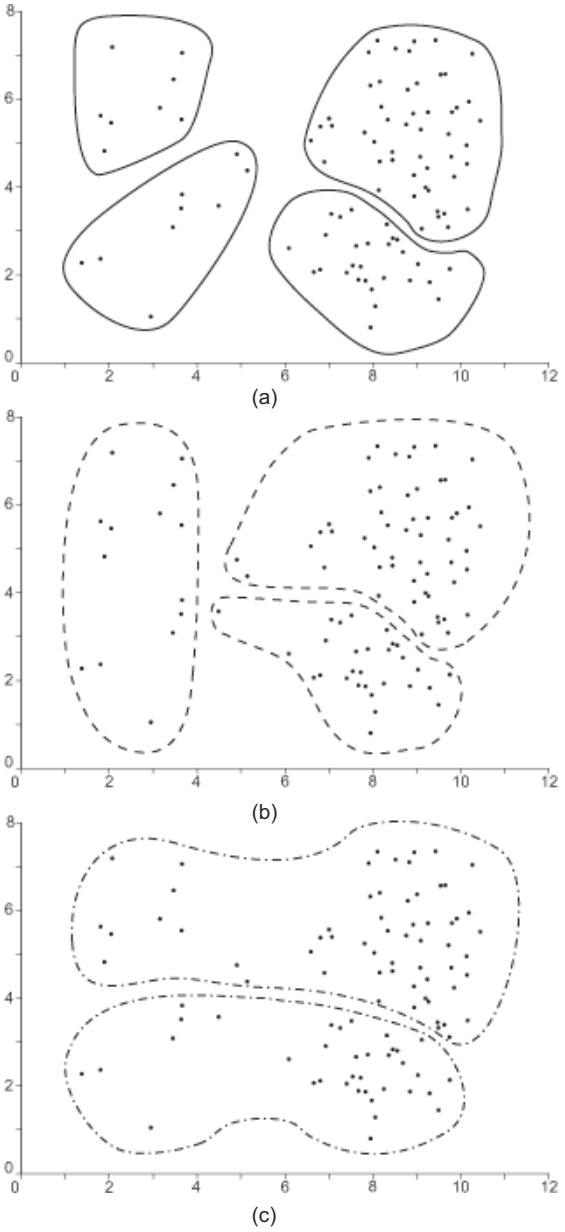

Fig. 3. Effects of simultaneous changes in the values of the parameters h and c: standard value $c = 0.5$ and h calculated by the cross-validation method (a), value increased to $c = 0.75$ and h calculated according to the formula (29) (b), value increased to $c = 1$ and h calculated according to the formula (29) (c).

which allows the number of clusters to be suited to the real structure of data. Applying the algorithm in its basic form does not require the user to supply arbitrary values for parameters, as they may be calculated using optimizing criteria; however, there also exists the possibility of their optional change. Thus, with a proper modification of parameter values, it is possible to influence the approximate quantity of clusters alone (although their exact number will depend on the internal structure of data), as well as—which is particularly worth underlining—the proportion of their appearances in dense as opposed to sparse areas of data set elements. Especially, it is possible practically not to intervene in the number of clusters in dense areas, at the same time significantly reducing, or even eliminating, clusters in sparse regions. The algorithm often creates one-element clusters, which indicates that they are atypical in a given data structure—it can be then homogenized by their elimination or by assigning them to the nearest clusters through the above mentioned appropriate change in parameter values.

Now, three application examples of the investigated algorithm will be presented, firstly for improving the quality of the kernel estimator of the distribution density, described in Section 2, next, for use in a classification task—referring to a real research problem from biology—for a set of real data available in classic literature, and then for the practical task of planning the marketing strategy of mobile phone operators.

Consider first the example of an eighty-element random sample illustrated in Fig. 4. In order to find a distribution density, the kernel estimators methodology, presented in Section 2, will be used, with the application of a normal kernel, a cross-validation method, a procedure for the modification of the smoothing parameter with standard intensity (8), and a linear transformation. So, the matrix R calculated for the sample considered is for the diagonal form (10):

$$R = \begin{bmatrix} 8.77 & 0 \\ 0 & 4.15 \end{bmatrix}, \tag{30}$$

and for the general form (11):

$$R = \begin{bmatrix} 8.77 & 1.41 \\ 1.41 & 4.15 \end{bmatrix}. \tag{31}$$

In the first case, this results in the contour lines of kernels being stretched horizontally by about 45%, while in the second case the direction undergoes a turn of about 30ř counter-clockwise. Thus, in both cases the shapes of kernels do not fit any clearly-outlined data subset and do not even represent any kind of "compromise" between them.

In order to improve the quality of the estimator, the random sample shown in Fig. 4 was submitted to the complete gradient clustering algorithm with $c = 1$ and the smoothing parameter value obtained using the formula (29), after which for each defined cluster a linear transformation matrix was calculated in its general form (11), giving the following:

$$R' = \begin{bmatrix} 0.18 & -0.22 \\ -0.22 & 4.74 \end{bmatrix}, \tag{32}$$

$$R'' = \begin{bmatrix} 3.14 & 3.21 \\ 3.21 & 3.57 \end{bmatrix}. \tag{33}$$

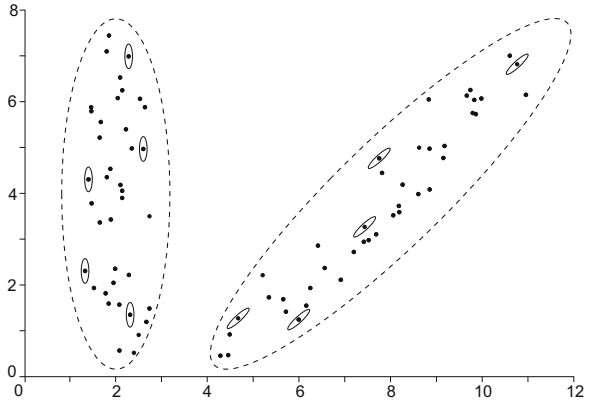

Fig. 4. Illustration of the improvement in the quality of the kernel estimator by the application of the local linear transformation matrices (32)–(33).

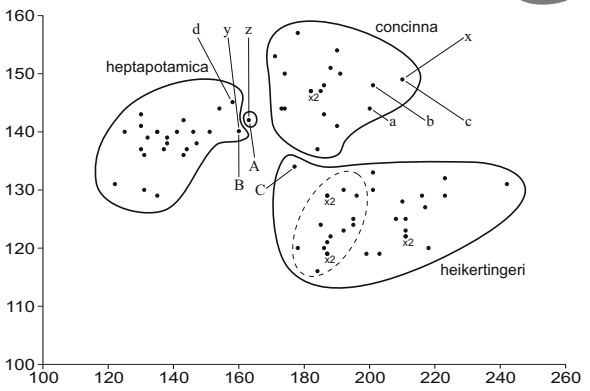

Fig. 5. Empirical data for three species of the beetle genus *Chaetocnema* (X-axis—width of the first joint of the tarsus, Y-axis—the front angle of the aedeagus).

Applying them separately to the obtained clusters, the shape of the kernels was appropriately stretched—for illustration, see Fig. 4 once more.

Knowing the theoretical distribution density from which the sample was generated, the mean integrated square error value was calculated. For the whole 80-element random sample, this error was only 2% less for the general form of the transformation matrix (31) than for the significantly simpler diagonal form (30). Such a small difference comes from the fact that—as mentioned above—in both cases the kernel shapes are almost equally poorly suited to the data structure. After using the complete gradient clustering algorithm and calculating the kernel estimators separately for the obtained clusters, the error decreased by about 50% for the general transformation matrices (32)–(33), and for the diagonal forms of the matrices—by about 20%. Both cases show evident advantages arising from the applications of preliminary clustering data with the help of the complete gradient clustering algorithm. Similar results occurred for other distributions, also multimodal, and for various random sample sizes.

The next example is based on real data referring to the often-found in Europe beetle of the genus *Chaetocnema*, existing in three speciesă: *conicinna*, *heikertingeri* and *heptapotamica*. The work (Lubischew, 1962, Tables 4–6), offers six features measured in 21, 31 and 22 males of the above species, respectively. The first feature, "width of the first joint of the tarsus in microns (the sum of measurements for both tarsi)", dominates, while the fourth, "the front angle of the aedeagus (1 unit = 7,5ř)" and the sixth, "the aedeagus width from the side (in microns)", are of medium importance. The remaining three are less significant. For the sake of graphic presentation and facilitating interpretation of the results obtained on a plane, the investigation below was limited to the two mentioned first.

Figure 5 shows points representing particular individually tested beetles (Lubischew, 1962, Tables 4–6). Some of them were characterized by identical measurements—in these cases corresponding points were given the symbol "$\times 2$". As a result of clustering using the complete gradient clustering algorithm for standard parameter values, the data set was divided into four clusters, with one of them containing one element.

Considering the task of classification with the clustering procedure, one can ascertain that three elements marked in Fig. 5 as A, B, C were wrongly assigned. The first of these forms the aforementioned one-element cluster, the second was wrongly included in the left-hand cluster representing the species *heptapotamica*, and the third was placed in the lower-right-hand *heikertingeri*, while all three should belong to the upper-right-hand cluster representing the species *conicinna*.

The basic conditioning of the problem—even without knowing the number of species—shows that the one-element cluster is erroneous. Following the instructions presented above, this cluster was eliminated in the typical way by taking $c = 1$ and the modification of the smoothing parameter h value, according to the formula (29). Thus the number of clusters was reduced to three. The point A was assigned to the left-hand cluster representing the species *heptapotamica*, which is not actually correct from the classification task point of view, as it should belong to the upper-right-hand cluster of *conicinna*. So the complete gradient clustering algorithm made three mistakes in classifying the beetle genus *Chaetocnema*. Looking at Fig. 5 it is worth noticing, however, that in the case of points A and B these mistakes are justified—the above points are placed very close to elements of the left-hand cluster of the species *heptapotamica*. With respect to the point C such diagnosis is not so unambiguous, though worth pointing out is the fact that it lies close to a significant concentration of elements from the lower-right-hand cluster representing *heikertingeri*.

For comparison, the same data were subjected to the classic k-means algorithm, available in statistical packages Statistica and SPSS. Standard parameter values as well as procedure forms were used during their running. It is worth stressing that the k-means algorithm availed of the *a priori* assumed correct number of clusters, which in many applications may not be known, or even such a "correct"—from a theoretical point of view—number might not exist at all (see the example for planning the marketing strategy for mobile phone operators, below). Thus, for the k-means algorithm from Statistica, the clustering process led to four erroneous classifications—wrongly assigned points are marked in Fig. 5 with the letters 'a', 'b', 'c' and 'd'—the first three were included in the lower-right-hand cluster representing the *heikertingeri* space, although they should be part of the upper-right-hand *conicinna*, while the fourth, actually belonging to the left-hand *heptapotamica*, was given to the upper-right-hand cluster of *conicinna*. The k-means algorithm from SPSS, however, generated 18 misclassifications. Here, a group of 15 elements in the left part of the lower-right-hand cluster representing the *heikertingeri* space—in Fig. 5 surrounded by a dashed line—were placed in the upper-right-hand cluster of *conicinna*, while for one element, marked in Fig. 5 by the letter 'x', the opposite classification error was the case. Moreover, the points 'y' and 'z' were put in the left-hand cluster representing the *heptapotamica* space, while they should actually belong to the upper-right-hand *conicinna*. When interpreting the relative positions of particular points in Fig. 5, in both cases errors created by the k-means algorithm are most often difficult to explain in terms that would be easy for people to understand.

The above comments can be generalized to the results of numerous tests carried out with the aim of comparing the complete gradient clustering algorithm investigated here with other classic clustering procedures besides k-means, e.g., hierarchical methods. It is difficult to confirm here the absolute supremacy of any one of them—to a large degree the advantage stemmed from the conditions and requirements formulated with regard to the problem under consideration, although the complete gradient clustering algorithm allowed greater possibilities of adjustment to the real structure of data and, consequently, the obtained results were more justifiable to a natural human point of view. A very important feature for practitioners was the possibility of firstly functioning using standard parameter values, and the option of changing them afterwards—according to individual needs—by the modification of two of them with easy and illustrative interpretations.

The complete gradient clustering algorithm was also successfully practically applied to planning the marketing strategy for mobile phone operators with respect to corporate clients. The aim of the research here was to investigate the appropriate behavior towards a given client, based on factors such as a mean monthly income from each SIM-card, the length of subscription, and the number of active SIM-cards. There was no *a priori* theoretical premise for setting the number of clusters characterizing particular types of clients. The data representing particular clients were divided into clusters, resulting in the possibility of defining a preferred marketing strategy with respect to each of them on the basis of fuzzy data obtained from experts.

A deep analysis and exploration of information contained in the client database not only allows the development of the best—from the client as well as the operator point of view—options for offering the most satisfaction possible to the former and at the same time the appropriate steering of the development of the latter, but also the acquisition of new clients. In particular, after the preliminary phase was performed for standard parameter values, the intensity of the modification of the smoothing parameter was increased by taking $c = 1$, with the aim of dividing the largest cluster containing over half of the elements and reducing small, and therefore less meaningful, clusters. Finally, 17 clusters were distinguished as a result for the complete gradient clustering algorithm, which was an acceptable size as far as further analysis was concerned, and consequently a change in the smoothing parameter value was not carried out, with the standard value remaining. The two largest clusters contained 27% and 23% of data base elements, with the next medium two in possession of 14% and 7%—most often elements of typical characteristics. The rest of the clusters accounted for less than 3%, most often atypical and firm-specific, although properly grouped. For details, see the publication (Kulczycki and Daniel, 2009).

The concept presented in this article is universal, and in particular cases the details may be refined, as an example see the different concepts of the stop criterion based on entropy applied in the works (Rodriguez and Suarez, 2006; Carreira-Perpinan, 2006).

Acknowledgment

Our heartfelt thanks go to our colleague Dr. Karina Daniel, whose research carried out for her Ph.D. thesis (Daniel, 2009) showed us the importance of the task presented here and gave credence to the usefulness of the obtained results.

References

Anderberg, M. R. (1973). *Cluster Analysis for Applications*, Academic Press, New York, NY.

Barnett, V. and Lewis, T. (1994). *Outliers in Statistical Data*, Wiley, Chichester.

Carreira-Perpinan, M. A. (2006). Fast nonparametric clustering with gaussian blurring mean-shift, *Proceedings of the International Conference on Machine Learning, Pittsburgh, PA, USA*, pp. 153–160.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**(8): 790–799.

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

Daniel, K. (2009). *Marketing strategy support method for a cell phone operator*, Ph.D. thesis, Systems Research Institute, Polish Academy of Sciences, Warsaw, (in Polish).

Everitt, B. S., Landau, S. and Leese, M. (2001). *Cluster Analysis*, Arnold, London.

Fukunaga, K. and Hostetler, L. D. (1975). The estimation of the gradient of a density function, with applications in pattern recognition, *IEEE Transactions on Information Theory* **21**(1): 32–40.

Girolami, M. and He, C. (2003). Probability density estimation from optimally condensed data samples, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(10): 1253–1264.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, NJ.

Kincaid, D. and Cheney, W. (2002). *Numerical Analysis*, Brooks/Cole, Pacific Grove, CA.

Kulczycki, P. (2005). *Kernel Estimators in Systems Analysis*, WNT, Warsaw, (in Polish).

Kulczycki, P. (2007). Kernel estimators in systems research, *in* P. Kulczycki, O. Hryniewicz and J. Kacprzyk (Eds), *Information Technologies in Systems Research*, WNT, Warsaw, pp. 79–105, (in Polish).

Kulczycki, P. (2008). Kernel estimators in industrial applications, *in* B. Prasad (Ed.), *Soft Computing Applications in Industry*, Springer-Verlag, Berlin, pp. 69–91.

Kulczycki, P. and Charytanowicz, M. (2008). A complete gradient clustering algorithm, *in* K. Malinowski and L. Rutkowski (Eds), *Control and Automation: Current Problems and Their Solutions*, EXIT, Warsaw, pp. 312–321, (in Polish).

Kulczycki, P. and Daniel, K. (2009). A method for supporting the marketing strategy of a mobile phone network provider, *Przegląd Statystyczny* **56**(2): 116–134, (in Polish).

Kulczycki, P. and Łukasik, S. (2009). Reduction of sample dimension and size for synthesis of a statistical fault detection system, *in* Z. Kowalczuk (Ed.), *Systems Detecting, Analysing and Tolerating Faults*, PWNT, Gdańsk, pp. 139–146, (in Polish).

Larose, D. T. (2006). *Data Mining Methods and Models*, Wiley, New York, NY.

Lubischew, A. A. (1962). On the use of discriminant functions in taxonomy, *Biometrics* **18**(4): 455–478.

Muller, H. G. (1984). Smooth optimum kernel estimators of densities, regression curves and models, *The Annals of Statistics* **12**(2): 766–774.

Pal, S. K. and Mitra, P. (2004). *Pattern Recognition Algorithms for Data Mining*, Chapman and Hall, London.

Rodriguez, R. and Suarez, A. G. (2006). A new algorithm for image segmentation by using iteratively the mean shift filtering, *Scientific Research and Essay* **1**(2): 43–48.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London.

Wand, M. P. and Jones, M. C. (1994). *Kernel Smoothing*, Chapman and Hall, London.

Wang, W. J., Tan, Y. X., Jiang, J. H., Lu, J. Z., Shen, G. L. and Yu, R. Q. (2004). Clustering based on kernel density estimation: Nearest local maximum searching algorithm, *Chemometrics and Intelligent Laboratory Systems* **72**(1): 1–8.

Yang, C., Duraiswami, R., DeMenthon, D. and Davis, L. (2003). Mean-shift analysis using quasi-newton methods, *Proceedings of the IEEE International Conference on Image Processing, Barcelona, Spain*, pp. 447–450.

Zhang, K., Tang, M. and Kwok, J. T. (2005). Applying neighborhood consistency for fast clustering and kernel density estimation, *Proceedings of the IEEE International Conference on Vision and Pattern Recognition, San Diego, CA, USA*, pp. 1001–1007.

Piotr Kulczycki is a professor at the Systems Research Institute of the Polish Academy of Sciences and the head of its Center for Statistical Data Analysis Methods, as well as at the Cracow University of Technology, where he is the head of the Department of Automatic Control and Information Technology. He has also held the position of a visiting professor at Aalborg University. The field of his scientific activity to date covers applicational aspects of information technology as well as data mining and analysis, mostly connected with the use of modern statistical methods and fuzzy logic in diverse issues of contemporary systems research and control engineering.

Małgorzata Charytanowicz is research scientist at the Systems Research Institute, Polish Academy of Sciences, as well as at the Department of Numerical Analysis and Programming Methods, John Paul II Catholic University of Lublin. She obtained her M.Sc. degree in mathematics from Maria Curie-Skłodowska University in Lublin, and her Ph.D. from the Systems Research Institute, Polish Academy of Sciences, in the area of computer science. Her research interests are programming methods, data analysis and its applications for medicine.

Appendix

Here a proof is provided that the algorithm presented in Section 3.2 converges, therefore that after a sufficient (finite) number of steps the condition (17) with the notations

(18)–(20) is fulfilled with probability 1. To this end, it is enough to show that

$$\lim_{k\to\infty} |D_k - D_{k-1}| = 0 \tag{34}$$

with probability 1, and therefore, for every $\epsilon > 0$ there exists $\widetilde{k} \in \mathbb{N}$ such that

$$\left| \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} d(x_i^{\widetilde{k}}, x_j^{\widetilde{k}}) - \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} d(x_i^{\widetilde{k}-1}, x_j^{\widetilde{k}-1}) \right| < \epsilon. \tag{35}$$

So let $\epsilon > 0$ be arbitrarily fixed. From the definition of the kernel estimator and the normal kernel form, it follows that the set $x \in \mathbb{R}^n$ such that $\nabla \hat{f}(x) = 0$ has measure zero. Because at the beginning of Section 3 it was assumed that elements of the set (1) are treated as realizations of a random variable with distribution having density, the probability that one of them belongs to this set is zero. Therefore, $\nabla \hat{f}(x_i) \neq 0$ for $i = 1, 2, \ldots, m$ with probability 1.

In the paper (Fukunaga and Hostetler, 1975) it was shown that in this case, for the normal kernel and the parameter b given by the formula (16), the algorithm (14)–(15) transposes the elements $x_1, x_2, \ldots, x_m$ to the proper modes of the estimator $\hat{f}$. A result is that for any fixed $i = 1, 2, \ldots, m-1$ and $j = i+1, i+2, \ldots, m$ there exists $\widetilde{k}_{i,j} \in \mathbb{N}$ such that for every natural k greater than $\widetilde{k}_{i,j}$ we have

$$\left| d(x_i^k, x_j^k) - d(x_i^{k-1}, x_j^{k-1}) \right| < \frac{2\epsilon}{m(m-1)} \tag{36}$$

(remember that a convergence sequence is a Cauchy one). As the number of factors in the sums appearing in the formula (35) equals $m(m-1)/2$, then denoting

$$\widetilde{k} = \max_{\substack{i=1,2,\ldots,m-1 \\ j=i+1,i+2,\ldots,m}} \widetilde{k}_{i,j}, \tag{37}$$

one obtains the condition (35), which finally completes this proof, establishing the convergence of the algorithm presented in Section 3.2 with probability 1.

Received: 2 October 2008
Revised: 10 April 2009