*Journal of Applied Statistics*
Vol. 39, No. 6, June 2012, 1211–1224

# The Complete Gradient Clustering Algorithm: properties in practical applications

Piotr Kulczycki[a,b*], Malgorzata Charytanowicz[a,c], Piotr A. Kowalski[a,b] and Szymon Lukasik[a,b]

*[a]Polish Academy of Sciences, Systems Research Institute, Centre of Information Technology for Data Analysis Methods, Warsaw, Poland; [b]Department of Automatic Control and Information Technology, Cracow University of Technology, Cracow, Poland; [c]Institute of Mathematics and Computer Science, Catholic University of Lublin, Lublin, Poland*

(*Received 11 May 2011; final version received 23 November 2011*)

The aim of this paper is to present a Complete Gradient Clustering Algorithm, its applicational aspects and properties, as well as to illustrate them with specific practical problems from the subject of bioinformatics (the categorization of grains for seed production), management (the design of a marketing support strategy for a mobile phone operator) and engineering (the synthesis of a fuzzy controller). The main property of the Complete Gradient Clustering Algorithm is that it does not require strict assumptions regarding the desired number of clusters, which allows to better suit its obtained number to a real data structure. In the basic version it is possible to provide a complete set of procedures for defining the values of all functions and parameters relying on the optimization criterions. It is also possible to point out parameters, the potential change which implies influence on the size of the number of clusters (while still not giving an exact number) and the proportion between their numbers in dense and sparse areas of data elements. Moreover, the Complete Gradient Clustering Algorithm can be used to identify and possibly eliminate atypical elements (outliers). These properties proved to be very useful in the presented applications and may also be functional in many other practical problems.

**Keywords:** data analysis and exploration; clustering; nonparametric methods; kernel estimators; seed production; mobile phone operator; fuzzy controller

## 1. Introduction

Clustering is becoming a fundamental procedure in data analysis. It lies between classical data analysis (where the aim of research is already recognized) and exploration data analysis, in which the subject of the investigated regularity is not known *a priori*, and its discovery constitutes an

[*]Corresponding author. Email: kulczycki@ibspan.waw.pl

ISSN 0266-4763 print/ISSN 1360-0532 online

http://dx.doi.org/10.1080/02664763.2011.644526
http://www.tandfonline.com

integral part of the research. In the first instance, clustering can be treated as classification, albeit without defined patterns, while the second views it as a division of the examined data set into clusters – subsets, each containing elements similar to others in the same subset, yet significantly differing from elements belonging to other subsets. For a basic notion, see the classical books, e.g. [4,9] as well as many interesting papers on a subject similar to the methodology used here, e.g. [8,17,23,24].

Clustering has no natural mathematical apparatus, such as for example differential calculus for investigating the extremes of a function. In this situation the ambiguity of an interpretation (important mainly in practical applications) as well as particular factors of the definition itself (e.g. the meaning of 'similarity' and consequently 'difference' of elements, or if it is clear that the number of clusters will be arbitrarily assumed or defined as a result of the structure of data itself or how to measure the quality of the divisions imposed) imply a huge variety of concepts and thus of clustering procedures. On one hand this significantly hinders the research, but on the other it allows to better suit the applied method to the specifics and requirements of a definite task. This also concerns the scale of 'automation' of a procedure – if the user wishes to select the values of parameters and directly influence the features of an obtained solution, or to gain in this case at least preliminary indications based on optimization criterions.

This paper aims to present the properties of the Complete Gradient Clustering Algorithm concerning its applicational potential, illustrated in examples of practical problems of bioinformatics, management and engineering, concerning the categorization of grains for seed production, the design of a marketing support strategy for a mobile phone operator and the synthesis of a fuzzy controller for the reduction of a rule set, respectively.

Consider the $m$ elements data set comprising $n$-dimensional vectors

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

treated here as a sample obtained from an $n$-dimensional real random variable. In the concept investigated here the natural assumption is made that particular clusters correspond to modes (local maxima) of the density function of distribution of this variable, and so the 'valleys' constitute a bordering of such clusters.

In the now classic paper [6] Fukunaga and Hostetler estimated such density using statistical kernel estimators – presently the main method of nonparametric estimation. In the framework of the numerical algorithm applied here, the elements of data set (1) are moved along the gradient of the density function until they concentrate in ever more clearly defined clusters. This method was formulated as a general idea only, leaving the details to the painstaking analysis of the user. Its positive features, naturalness and clarity of interpretation allowed the application of the method in many varied specialist tasks such as tracking, image segmentation, information fusion and video processing (see [29] for a list of examples) as well as creating interesting mutations and supplements (see e.g. [2,28]). In the literature, one can even find unintentional repetition of the same idea [26].

The Complete Gradient Clustering Algorithm presented in this paper is based on Fukunaga's and Hostetler's concept, and has been supplemented and given in its full form, suitable for direct use without requiring users to have a deeper statistical knowledge or conduct laborious research. It can be characterized by the following features:

1. all parameters can be effectively calculated using numerical procedures based on optimizing criteria;
2. the algorithm does not demand strict assumptions regarding the desired number of clusters, which allows the number obtained to be better suited to a real data structure;
3. the parameter directly responsible for the number of clusters is indicated; it will also be shown how possible changes – e.g. with regard to values calculated using optimizing criteria (see

point 1 stated above) – to this value, influence the increase or decrease in the number of clusters without, however, defining their exact number;

4. moreover, the next parameter can be easily indicated, the value of which will influence the proportion between the number of clusters in dense and sparse areas of elements of data set (1); here also the value of this parameter can be assumed based on optimizing criteria (see again point 1); it will also be shown here that potential lowering of the value of this parameter results in a decrease in the number of clusters in dense regions of data as the number of clusters in sparse areas increases, while a potential raise in its value has the opposite effect – increasing the number of clusters in dense areas while simultaneously reducing or even eliminating them from sparse regions of data set (1);
5. the appropriate relation between the two above-mentioned parameters allows for a reduction, or even elimination of clusters in sparse areas, practically without influencing the number of clusters in dense areas of data set elements;
6. the algorithm also creates small, even single-element clusters, which can be treated as atypical elements (outliers) in a given configuration of clusters, which makes possible their elimination or assignation to the closest cluster by a change – described in the previous point – in the values of the appropriate parameters.

The features in point 4, and in consequence 5, are particularly worth underlining as practically nonexistent in other clustering procedures. In practical applications, it is worth highlighting the implications of points 1 and 2, and also potentially 3. Unusual possibilities are offered by the property expressed in point 6.

## 2. Statistical kernel estimators

Let the $n$-dimensional random variable $X : \Omega \to \mathsf{R}^n$, with a distribution having the density $f$, be given. Its kernel estimator $\hat{f} : \mathsf{R}^n \to [0, \infty)$ is calculated on the basis of the $m$-elements random sample (1) experimentally obtained from the variable $X$, and is defined in its basic form by

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

where the measurable function $K : \mathsf{R}^n \to [0, \infty)$, symmetrical with respect to zero and having a weak global maximum in this point, fulfils the condition $\int_{\mathsf{R}^n} K(x)dx = 1$ and is called a kernel, whereas the positive coefficient $h$ is referred to as a smoothing parameter. It is worth noting that a kernel estimator allows the identification of density for practically every distribution, without any assumptions concerning its membership to a fixed class. Atypical, complex distributions, also multimodal, are regarded here as textbook unimodal.

Setting the quantities introduced in definition (1), i.e. choice of the form of the kernel $K$ as well as calculation of the value for the smoothing parameter $h$, is most often carried out according to the criterion of minimum of an integrated mean-square error. Broader discussion and practical algorithms are found in the books [11,21,25][1]. In particular, the choice of the kernel $K$ form has no practical meaning and thanks to this it is possible to first take into account properties of the estimator obtained (e.g. its class of regularity, boundary of a support, etc.) or aspects of calculations, advantageous from the point of view of the applicational problem under consideration. On the contrary, the value of the smoothing parameter $h$ has significant meaning for quality of estimation. Too small a value causes a large number of local extremes of the estimator $\hat{f}$ to appear, on the other hand, too big values of the parameter $h$ result in overflattening of this estimator – this property will be successfully used here later.

Practical applications may also use additional procedures, some generally improving the quality of the estimator, and others – optional – possibly fitting the model to an existing reality. For the first group one should recommend the modification of the smoothing parameter [11, Section 3.1.6; 21, Section 5.3.1] and a linear transformation [11, Section 3.1.4; 21, Section 4.2.1], while for the second, the boundaries of a support [11, Section 3.1.8; 21, Section 2.10].

Finally, the procedure of the modification of the smoothing parameter is outlined below, as it will be heavily used in the following. Thus, in the case of the basic definition of kernel estimator (1), the influence of the smoothing parameter on particular kernels is the same. Advantageous results are obtained thanks to the individualization of this effect, achieved by introducing the positive modifying parameters $s_1, s_2, \ldots, s_m$ mapped to particular kernels, whose value is given as

$$s_i = \left( \frac{\hat{f}_*(x_i)}{\bar{s}} \right)^{-c}, \tag{3}$$

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

$$\hat{f}(x) = \frac{1}{mh^n} \sum_{i=1}^{m} \frac{1}{s_i^n} K \left( \frac{x - x_i}{hs_i} \right). \tag{4}$$

Thanks to the above procedure, the areas in which the kernel estimator assumes small values are flattened and the areas connected with large values – peaked. The parameter $c$ stands for the intensity of the modification procedure – the greater its value, the stronger (more distinct) the above procedure. Based on indications for the criterion of the integrated mean-square error, the value

$$c = 0.5 \tag{5}$$

can be tentatively suggested.

Detailed information regarding kernel estimators can be found in the monographs [11,21,25]. Example practical applications are presented in the publications [12,13].

## 3. Complete gradient clustering algorithm

Consider – as in the Introduction – the $m$-elements set of $n$-dimensional vectors (1). This will be treated as a random sample obtained from the $n$-dimensional random variable $X$, with distribution having the density $f$. Using the methodology described in Section 2, the kernel estimator $\hat{f}$ can be created. (Note that if the kernel $K$ used is not only measurable – as was required in the definition – but is also differentiable, then this will also be a property of the obtained estimator $\hat{f}$, which ensures the existence of the gradient $\nabla \hat{f}$.) Take the natural assumption that particular clusters are related to its modes (i.e. the local maxima of the function $\hat{f}$), and elements of set (1) are mapped onto them by shifting in the gradient $\nabla \hat{f}$ direction, with the appropriate fixed step.

The above is carried out iteratively with the Gradient Clustering Algorithm [6], based on the classic Newtonian procedure [10, Section 3.2], defined as

$$x_j^0 = x_j \quad \text{for } j = 1, 2, \ldots, m, \tag{6}$$

$$x_j^{k+1} = x_j^k + b \frac{\nabla \hat{f}(x_j^k)}{\hat{f}(x_j^k)} \quad \text{for } j = 1, 2, \ldots, m \quad \text{and} \quad k = 0, 1, \ldots, k^*, \tag{7}$$

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

$$b = \frac{h^2}{n+2} \tag{8}$$

[6].[2]

Based on comprehensive research, the above concept was supplemented to its full form, useful for effective application without having deeper knowledge in statistics or laborious subject investigations. Thanks to the appropriate utilization of specific features and additional procedures for kernel estimators, the properties mentioned at the end of this paper's Introduction are obtained.

To construct the estimator, the normal kernel

$$K(x) = \frac{1}{(2\pi)^{n/2}} \mathrm{e}^{-x^{\mathrm{T}}x/2} \tag{9}$$

is applied, due to its differentiability in the whole domain, convenience for analytical considerations connected with gradient, and assuming positive values, which in every case guards against division by zero in formula (7). The procedure was used for modification of the smoothing parameter with standard intensity (5) and linear transformation [11, Section 3.1.4; 21, Section 4.2.1] with a matrix of diagonal form[3]

$$R = \begin{bmatrix} \sqrt{\mathrm{Var}(X_1)} & 0 & \cdots & 0 \\ 0 & \sqrt{\mathrm{Var}(X_2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{\mathrm{Var}(X_n)} \end{bmatrix}, \tag{10}$$

where $\mathrm{Var}(X_i)$ means the variance of the $i$th coordinate. Finally, the kernel estimator takes the form

$$\hat{f}(x) = \frac{1}{mh^n \det(R)} \sum_{i=1}^{m} \frac{1}{s_i^n} K\left(R^{-1} \frac{x - x_i}{hs_i}\right). \tag{11}$$

Next, it is assumed that algorithm (6)–(7) should be finished, if after the consecutive $k$th step particular elements move very little, and so if – in consequence – the following condition is fulfilled:

$$\frac{|D_k - D_{k-1}|}{D_0} \leq a, \tag{12}$$

where $a > 0$ and

$$D_0 = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} d(x_i, x_j), \tag{13}$$

$$D_{k-1} = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} d(x_i^{k-1}, x_j^{k-1}), \quad D_k = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} d(x_i^k, x_j^k), \tag{14}$$

while $d$ means the Euclidean metric in $\mathrm{R}^n$. Therefore, $D_0$ and $D_{k-1}$, $D_k$ denote sums of distances between particular elements of set (1) before starting the algorithm and after the ($k$–1)th and $k$th step, respectively. For correctness of formula (12), note that obviously $D_0 \neq 0$. Primarily, it is recommended that

$$a = 0.001. \tag{15}$$

The potential decrease in this value does not significantly influence the obtained results, although increases require individual verification of their correctness.

Finally, if after the $k$th step condition (12) is fulfilled, then

$$k^* = k \tag{16}$$

and consequently this step is treated as the last one.

Now, a procedure should be used for creating clusters and assigning particular elements to them For this purpose, the following set is investigated:

$$x_1^{k^*}, x_2^{k^*}, \ldots, x_m^{k^*}, \tag{17}$$

consisting of the elements of set (1) after the $k^*$th step of algorithm (6)–(7). Following this, the set of mutual distances of the above elements

$$\{d(x_i^{k^*}, x_j^{k^*})\}_{\substack{i=1,2,\ldots,m-1 \\ j=i+1,i+2,\ldots,m}} \tag{18}$$

should be defined. The number of elements of set (18) amounts to

$$m_d = \frac{m(m-1)}{2}. \tag{19}$$

Taking set (18) as a sample of a one-dimensional random variable, i.e. as sample (1), the auxiliary kernel estimator $\hat{f}_d$ ought to be calculated using the methodology described in Section 2. It can be interpreted as an estimator of distances between the elements of set (17). Normal kernel (9) is once again proposed, as is the use of the procedure of smoothing parameter modification with standard value of parameter (5), 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) local minimum of the function $\hat{f}_d$ belonging to the interval $(0, D)$, where

$$D = \max_{\substack{i=1,2,\ldots,m-1 \\ j=i+1,i+2,\ldots,m}} d(x_i, x_j). \tag{20}$$

For this purpose, one should treat set (18) as a random sample, calculate its standard deviation $\sigma_\mathrm{d}$, and next take in sequence the values $x$ from the set

$$\{0, 0.01 \cdot \sigma_d, 0.02 \cdot \sigma_d, \ldots, [\mathrm{int}(100 \cdot D) - 1] \cdot \sigma_\mathrm{d}\}, \tag{21}$$

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

$$\hat{f}_\mathrm{d}(x - 0.01\sigma_\mathrm{d}) > \hat{f}_\mathrm{d}(x) \quad \text{and} \quad \hat{f}_\mathrm{d}(x) \leq \hat{f}_\mathrm{d}(x + 0.01\sigma_\mathrm{d}). \tag{22}$$

Such calculated value[4] will be denoted hereafter as $x_\mathrm{d}$, 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 an element of set (17) and initially create a one-element cluster containing it;
2. find an element of set (17) different from the one in the cluster, closer than $x_\mathrm{d}$; if there is such an element, then it should be added to the cluster; in the other case – proceed to point 4;
3. find an element of set (17) different from elements in the cluster, closer than $x_\mathrm{d}$ 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 set (17) elements of this cluster; if this so-reduced set (17) is not empty, return to point 1; in the other case – finish the algorithm.

The ‘list of clusters’ defined in such a way contains all clusters marked out in the above procedure. Therefore, this becomes the Complete Gradient Clustering Algorithm in basic form – its possible modifications and their influence on the obtained results will be presented in the next section.

Finally, it is worth mentioning the possibility of reducing set (18). In practice, it is too large not only because of the square dependence regarding size of set (1), occurring in formula (19), but also due to the fact that the estimator $\hat{f}_{\mathrm{d}}$ concerns the one-dimensional random variable, while $\hat{f}$, usually the multidimensional, by nature demands a notably greater sample size. For a very large size of sample (18), it is worth using data-compression procedures well known in the literature, see e.g. [7, 19, Section 2.5].

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 [1,20] – when applied to the Complete Gradient Clustering Algorithm it proved to be similarly effective as the one based on formula (12), although more laborious in implementation.

## 4. Influence of the values of parameters on results obtained

It is worth repeating that the presented clustering algorithm did not require a preliminary, in practice often arbitrary, assumption concerning number of clusters – their size depending solely on the internal structure of data, given as 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 optimizing reasons into account. However, optionally – if the researcher makes the decision – by an appropriate change in values of kernel estimator parameters it is possible to influence the size of number of clusters, and also the proportion of their appearance in dense areas in relation to sparse regions of elements in this set.

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 $h$ in 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 of between $-25\%$ and $+50\%$. Outside this range, results obtained require individual verification.

Next, as mentioned in Section 2, the intensity of modification of the smoothing parameter is implied by the value of the parameter $c$, given as standard by formula (5). Its increase smoothness of the kernel estimator in areas where elements of set (1) are sparse, and also sharpens it in dense areas – as a 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 one 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 results obtained. Particularly it is recommended that $c = 1$.

Practice, however, often prevents changes to the clusters in dense areas of the 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 commonly arising due to various errors. Putting the above considerations together, one can propose an increase in both the standard scale of the smoothing parameter modification (5) as well as 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

$$h^* = \left(\frac{3}{2}\right)^{c-0.5} h. \tag{23}$$

The joint action of both these factors results in a twofold smoothing of the function $\hat{f}$ in the regions where the elements of set (1) are sparse. Meanwhile, these factors more or less compensate for each other in dense areas, thereby having practically no 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 results obtained. Particularly, it is recommended that $c = 0.75$.

More details with illustrative examples can be found in the paper [14].

## 5. Applicational examples

The algorithm presented in this paper was comprehensively tested both for random statistical data as well as generally available benchmarks. It was also compared with other well-known clustering methods, $k$-means and hierarchical procedures. 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 for 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 functioning using standard parameters values first, and the option of changing them afterwards – according to individual needs – by the modification of two of them with easy and illustrative interpretations. These properties were actively used in three projects from the domains of bioinformatics, management and engineering, which will be presented in detail in the following subsections.

### 5.1 *Categorization of grains for seed production*

Bioinformatics – a discipline concerning the application of mathematical and IT tools to solve problems of biological science – is now growing on an exceptionally dynamic and diverse scale. Opportunities are increasing thanks to the development and prevalence of computer technology which have resulted in a sudden increase in mutual understanding and cooperation in the frameworks of previously different research methods of hard and natural sciences. The results of investigations carried out as part of a larger project on the categorization of grains according to the geometric features of seeds, taken from X-ray images, for production purposes will be presented below.

For an illustrative and comparative presentation of aspects of research using the Complete Gradient Clustering Algorithm presented in this paper, an analysis will be made of a sample of harvested wheat grain originating from experimental fields explored at the Institute of Agrophysics of the Polish Academy of Sciences in Lublin. The examined group consisted of grains of three strains of wheat – *Kama*, *Rosa* and *Canadian* – with 70 of each type selected randomly for testing. A high-quality visualization of their internal structures was achieved using a soft X-ray technique, without destroying the subject material. After scanning the resulting pictures, the following seven geometric parameters of wheat kernels were obtained using the program GRAINS, specially created to this aim: area $A$, perimeter $P$, compactness $C = 4\pi A/P^2$, length of kernel, width of kernel, asymmetry coefficient and length of kernel groove. Each was thus represented by a seven-dimensional vector ($n = 7$), while their set comprises a 210-element sample (1). In the preliminary phase, the data dimensionality was reduced to 2 using Principal Components Analysis.

As a result of using the Complete Gradient Clustering Algorithm with the standard values of the smoothing parameter *h* and the intensity of its modification *c*, obtained by the mean-square criterion, seven clusters were found, of 76, 64, 57, 7, 3, 2, 1 elements each. It can be deduced that the first three represent the three used for the analysis investigated here, while the remaining four small clusters contain atypical elements, without excluding physically damaged. If one disregards the 13 units contained in these 4 small clusters (6% of the entire population), the number of correctly classified grains was, in order 91%, 97%, 88% for *Kama*, *Rosa* and *Canadian*, respectively. It is worth pointing out that the above results were obtained without the need for any *a priori* assumption as to the required number of clusters, information which may be difficult or even impossible to obtain in practical problems in biology.

If, however, a necessity is assumed to map every element to one of the larger clusters, then this can be achieved by appropriately changing the values of the parameters *h* and *c* to those obtained with optimization criterions. Thus, by successively increasing the value of the former, the number of local extremes of the kernel estimator falls, while decreasing the latter makes it impossible to divide the large clusters created in this way. In doing so, three large clusters are obtained for *h* increasing by 75% and *c* decreased to a value of 0.1. The number of correct classifications was for one strain slightly lower than that obtained earlier, and was 91%, 96%, 88% for particular strains, respectively, and was still reached without any arbitrary assumptions as to number of clusters required.

The above results were comparable to those for other methods, among others classic *k*-means, although in this case it did require additional correct information regarding the number of classified strains.

In summary, use of the Complete Gradient Clustering Algorithm, presented in this paper, allowed the correct classification of the grains of three strains of wheat without *a priori* information about their number. What is more, with standard parameters values, the above algorithm also enabled the identification of atypical elements, e.g. physically damaged and – following their elimination from the sample – a slight reduction in the number of misclassifications in the remaining part.

The above illustratory example, concerning three strains of wheat, can be generalized for other categorization tasks of seed produce of similar conditioning. This research was carried out in cooperation with Prof. Jerzy Niewczas and Slawomir Zak.

### 5.2 *Marketing support strategy for mobile phone operator*

The highly dynamic growth prevalent on the mobile phone network market naturally necessitates a company to permanently direct its strategy towards satisfying the differing needs of its clients, while at the same time maximizing its income. The uncontrollable nature of this kind of activity, however, can lead to a loss of coherence in treating particular clients, and their subsequent defection to competitors. To avoid this, a formal solution of global nature must be found. Below are presented the results of research prepared for a Polish mobile phone network operator, concerning long-term business clients, i.e. those with more than 30 SIM cards and an account history of at least 2 years.

In practice, there is a vast spectrum of quantities characterizing particular subscribers. Following detailed analysis of the economic aspects of the task under investigation here, it was taken that basic traits of business clients would be shown by three quantities: average monthly income per SIM card, length of subscription and number of active SIM cards. Thus, each of *m*-elements of a database $x_1, x_2, \dots, x_m$ is characterized by the following three-dimensional vector:

$$x_i = \begin{bmatrix} x_{i,1} \\ x_{i,2} \\ x_{i,3} \end{bmatrix} \quad \text{for } i = 1, 2, \dots, m, \tag{24}$$

where $x_{i,1}$ denotes the average monthly income per SIM card of the $i$th client, $x_{i,2}$ its length of subscription, and $x_{i,3}$ the number of active SIM cards.

In the initial phase, atypical elements of the set $x_1, x_2, \ldots, x_m$ (outliers) are eliminated, according to the procedure presented in the publication [16], based on kernel estimators methodology. The uniformity of the data structure is so increased, and it is worth underlining that effect is obtained by canceling only those elements which would not be of importance further in the procedure investigated.

Next clustering of the data set is performed, using the Complete Gradient Clustering Algorithm presented in this paper. This results in a division of the data set representing specific clients, into groups of similar nature. The results obtained for typical intensity of smoothing parameter modification (5) indicated that an excessive number of clusters of small sizes, located in areas of low density of sample elements, most often contain insignificant specific clients, and that an overly numerous main cluster contains over half the elements. In accordance with the properties of the algorithm used, this value was increased to $c = 1$. This gave the desired effect: the number of 'peripheral' clusters lowered significantly and the main cluster was split. The obtained number of clusters was satisfying, which led to any possible change in the value of the smoothing parameter $h$ becoming redundant. Finally, the sample, considered at this stage, containing 1639 elements was divided into 26 clusters of the following sizes: 488, 413, 247, 128, 54, 41, 34, 34, 33, 28, 26, 21, 20, 14, 13, 12, 10, two 4-element clusters, three of 3-elements, two of 2-elements and two of 1-element. It is worth noting the four clearly drawn groups: the first of these comprises two numerous clusters of 488 and 413-elements, next two medium-sized 247- and 128-elements, followed by small – nine clusters containing from 20 to 54 and lastly 13 clusters of less than 20 elements. Next began the elimination of these last clusters, with the exception however of those containing key clients (clusters of 14, 13 and 10-elements) as well as one where at least half of its elements were prestige clients (12-elements cluster). In the end, 17 clusters remained for further analysis.

Next for each of the above-defined clusters, an optimal – from the point of view of expected profit of the operator – strategy is created for treating subscribers belonging to it. With regard to the imprecise evaluation of experts used here, elements of fuzzy logic and preference theory [5] have been used – details are however beyond the scope of this paper.

It is worth pointing out that none of the above calculations must be carried out at the same time as negotiating with the client, but merely updated (in practice once every 1–6 months).

The client being negotiated with is described with the aid – in reference to formula (24) – of a three-dimensional vector, whose particular coordinates denote average monthly income per SIM card of that client, length of its subscription and the number of its active SIM, respectively. These data can relate to the subscriber history to date in a given network, when renegotiating contract terms, or in a rival network if attempting to take them over. Mapping of the client being negotiated to the proper subscriber group, from those obtained as a result of earlier-performed clustering, was carried out using Bayes classification also applying kernel estimators methodology (for subject bibliography see [15]). Due to the fact that the marketing strategies for particular clusters have already been defined, this finally completes the procedure for the algorithm to support the marketing strategy for a business client, investigated here.

The above method, researched with the cooperation of Dr Karina Daniel, was successfully implemented for the needs of a Polish network operator.

### 5.3 *Synthesis of fuzzy PID controller*

Fuzzy PID controllers are a valuable – from an applicational point of view – generalization of the commonly used, precisely examined and familiarized by practitioners classical PID feedback controllers. The fuzzy version is particularly useful for challenging systems, e.g. containing strong nonlinearities and uncertainties, thanks to the greater degree of freedom, such controllers can better fit the specifics of an object. On the other hand, however, too great a degree of freedom may cause difficulties in appropriately fixing their functions and parameters, implying an incorrectly working system, and in the extreme case impossible excessive expansion of its structure making it impossible to realize in practice. The problem of a suitably large, but not lowering quality, simplification of fuzzy PID controllers structures is therefore fundamentally significant in applicational engineering.

Investigated below are the fuzzy PID controllers in Takagi–Sugeno sense [27]. Their concept is built on the set (base) of $k$ fuzzy rules of the form

$$\text{IF } (x \text{ is } A_j) \text{ THEN } (y = f_j(x)) \quad \text{for } j = 1, 2, \ldots, k. \tag{25}$$

If – according to the character of the fuzzy approach – the element $x$ belongs to many sets to a degree defined by values of their membership functions, i.e. with $\mu_{A_j}(x)$, then finally $y$ takes the form of the normalized mean

$$y = \frac{\sum_{j=1}^{k} \mu_{A_j}(x) f_j(x)}{\sum_{j=1}^{k} \mu_{A_j}(x)}. \tag{26}$$

In the case of fuzzy PID controllers, the coordinates of the vector $x$ are connected with an error and its integral and derivative, while the variable $y$ constitutes a generated control. Even if one assumes the simple triangular or trapezoid membership functions $\mu_{A_j}$, and that the functions $f_j$ are linear, then the large number of parameters appearing in such a task may pose the threat of losing the possibility of correct effective fixing of their values. The appropriate reduction in the size of the fuzzy rules set (25) becomes therefore a fundamental problem, in particular for the complex applicational cases. To solve the task of reducing fuzzy rules, many contemporary IT methods are used, above all evolutionary algorithms, neuro-fuzzy systems or statistical approaches also, among which dominate concepts based on the clustering technique. The Complete Gradient Clustering Algorithm presented in this paper was applied successfully to this aim.

Let then be given the vector $\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right]$ and $m$ measurements of values obtained during operation of the system with the fuzzy PID controller in its primary form, i.e. without reducing the rules set:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}, \ldots, \begin{bmatrix} x_m \\ y_m \end{bmatrix}. \tag{27}$$

Treating the above set as random sample (1) one can perform clustering with the use of the Complete Gradient Clustering Algorithm presented in this paper. Let

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \end{bmatrix}, \begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{x}_{\tilde{m}} \\ \tilde{y}_{\tilde{m}} \end{bmatrix} \tag{28}$$

represent centers of $\tilde{m}$ clusters obtained in this way. Each of the element $\tilde{x}_i$ for $i = 1, 2, \ldots, \tilde{m}$ may be the basis of $i$th fuzzy rule with the respective membership function

$$\mu_i(x) = \exp\left( - \left\| \frac{x - \tilde{x}_i}{d} \right\|^2 \right), \tag{29}$$

where the 'scaling' parameter $d > 0$ characterizes the generalization ability resulting from the fuzzy inference concerning the control system under design. The experimental research carried out indicates that the value $d = \tilde{m}/2$ can be successfully used. As a consequence, formula (26) takes the following form:

$$y = \frac{\sum_{i=1}^{\tilde{m}} \mu_i(x) f_i(x)}{\sum_{i=1}^{\tilde{m}} \mu_i(x)}, \tag{30}$$

where $f_i$ are linear functions whose parameters may be calculated based on the classical least-squares estimation task.

This method was positively verified in numerous practical problems. Presented below are comparative results obtained for the control system of a hard-drive servo motor, presented in the paper [22]. Its following model was used:

$$\begin{bmatrix} \dot{s}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} 1 & 1.664 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 1.384 \\ 1.664 \end{bmatrix} u(t), \tag{31}$$

where $u$ constitutes actuator input (in volts), $s$ and $v$ are the position (in tracks) and velocity of the disk drive's head. The problem of accurate positioning was analyzed with $s(t)$ as an output. Typically for such applications, a controller of PD type was considered [18].

First, the standard PD fuzzy controller with 49-rules was tuned for quick response with the step reference signal. The 121-elements set (27) was obtained in this way:

$$\begin{bmatrix} e_1 \\ \dot{e}_1 \\ u_1 \end{bmatrix}, \begin{bmatrix} e_2 \\ \dot{e}_2 \\ u_2 \end{bmatrix}, \ldots, \begin{bmatrix} e_{121} \\ \dot{e}_{121} \\ u_{121} \end{bmatrix}, \tag{32}$$

where $e$ represents error, was treated as random sample (1) and subjected to the Complete Gradient Clustering Algorithm. As a result, the PD fuzzy controller with the base reduced to 38 rules was obtained.

To compare the results acquired using a classical PD feedback-controller, a fuzzy PD controller with full (unreduced) 49-element rule base [18], and the above investigated fuzzy controller with base reduced to 38 rules, for each of them the values were obtained for the root-mean-square-error index and the percentage overshoot for a response with the step reference signal. For the first value, the results were 0.291, 0.198, 0.111, respectively, for the second 78%, 92%, 15%. For both, the best results were provided by the use of the fuzzy PD controller with the rule set reduced using the Complete Gradient Clustering Algorithm. Similar results were achieved for other conditions and performance indexes.

Further testing was carried out for the system with the fuzzy controller with the rule set reduced by the Complete Gradient Clustering Algorithm, for various – different from those obtained with the integrated mean-square error criterion – values of the smoothing parameter $h$ and the intensity of modification $c$. The most advantageous results were achieved for the value of the latter, slightly lowered – with respect to optimal (5) – to $c = 0.25$. This effect can be interpreted by an increase in the number of peripheral clusters characterizing atypical states, 'dangerous' from the point of view of correct behavior of the system. Moreover, the main cluster generally contained even 80% elements of set (32), representing 'safe' states, and its potential division did not bring any positive changes. As before it was not necessary to alter – with respect to optimal – the value of the smoothing parameter $h$. It proves once again the Complete Gradient Clustering Algorithm adapts well to real data structures.

The presented concept was successfully implemented for the control of a robot under the authority of the Department of Automatic Control and Information Technology of the Cracow University of Technology.

## Notes

1. For calculating a smoothing parameter one can especially recommend the plug-in method in the one-dimensional case [11, Section 3.1.5; 21, Section 3.6.1] as well as the cross-validation method [11, Section 3.1.5; 21, Section 3.4.3] in the multidimensional. Comments for the choice of kernel may best be found in [11, Section 3.1.3; 25, Sections 2.7 and 4.5].
2. For convenience of calculating 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 a 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 [2,3,28]. The method of calculation of the above expression's value is of no relevance for further parts of the presented material.
3. Using the general form of transformation matrix $R = \sqrt{\mathrm{Cov}(X)}$ [11, Section 3.1.4; 21, Section 4.2.1] results in a lengthening of shape of kernels in one direction. This causes a difference in rate of convergence of algorithm (14) and (15) with respect to the direction of transposition of elements of set (1), non-justified from the point of view of the clustering task, and consequently interfering with obtained results. Also for this reason, the product kernel [11, Section 3.1.3; 25, Section 4.2], very useful in practical applications, was rejected.
4. 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 set (23) is then empty.

## References

[1] M.A. Carreira-Perpinan, *Fast nonparametric clustering with Gaussian blurring mean-shift*, Proceedings of the International Conference on Machine Learning, Pittsburgh, USA, 25–29 June 2006, pp. 153–160.

[2] Y. Cheng, *Mean shift, mode seeking, and clustering*, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1995), pp. 790–799.

[3] D. Comaniciu and P. Meer, *Mean shift: A robust approach toward feature space analysis*, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002), pp. 603–619.

[4] B.S. Everitt, S. Landau, and M. Leese, *Cluster Analysis*, Arnold, London, 2001.

[5] J. Fodor and M. Roubens, *Fuzzy Preference Modelling and Multicriteria Decision Support*, Kluwer, Dordrecht, 1994.

[6] K. Fukunaga and L.D. Hostetler, *The estimation of the gradient of a density function, with applications in pattern recognition*, IEEE Trans. Information Theory 21 (1975), pp. 32–40.

[7] M. Girolami and Ch. He, *Probability density estimation from optimally condensed data samples*, IEEE Trans. Pattern Anal. Mach. Intell. 25 (2003), pp. 1253–1264.

[8] M. Herbin, N. Bonnet, and P. Vautrot, *A clustering method based on the estimation of the probability density function and on the skeleton by influence zones. Application to image processing*, Pattern Recognition Lett. 17 (1996), pp. 1141–1150.

[9] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Englewood Cliffs, 1988.

[10] D. Kincaid and W. Cheney, *Numerical Analysis*, Brooks/Cole, Pacific Grove, 2002.

[11] P. Kulczycki, *Estymatory jadrowe w analizie systemowej*, WNT, Warsaw, 2005.

[12] P. Kulczycki, *Estymatory jadrowe w badaniach systemowych*, in *Techniki informacyjne w badaniach systemowych*, P. Kulczycki, O. Hryniewicz, and J. Kacprzyk, eds., WNT, Warsaw, 2007, pp. 79–105.

[13] P. Kulczycki, *Kernel estimators in industrial applications*, in *Soft Computing Applications in Industry*, B. Prasad, ed., Springer-Verlag, Berlin, 2008, pp. 69–91.

[14] P. Kulczycki and M. Charytanowicz, *A complete gradient clustering algorithm formed with kernel estimators*, Int. J. Appl. Math. Comput. Sci. 20 (2010), pp. 123–134.

[15] P. Kulczycki and P.A. Kowalski, *Bayes classification of imprecise information of interval type*, Control and Cybernetics 40 (2011), pp. 101–123.

[16] P. Kulczycki and C. Prochot, *Wykrywanie elementow odosobnionych za pomoca metod estymacji nieparametrycznej*, in *Badania operacyjne i systemowe: podejmowanie decyzji – podstawy teoretyczne i zastosowania*, R. Kulikowski, J. Kacprzyk, and R. Slowinski, eds., EXIT, Warsaw, 2004, pp. 313–328.

[17] C. Mauceri and D. Ho, *Clustering by kernel density*, *Comput. Economics* 29 (2007), pp. 199–212.

[18] R. Mudi and N.R. Pal, *A robust self-tuning scheme for PI and PD type fuzzy controllers*, IEEE Trans. Fuzzy Systems, 7 (1999), pp. 2–16.

[19] S.K. Pal and P. Mitra, *Pattern Recognition Algorithms for Data Mining*, Chapman & Hall, London, 2004.

[20] R. Rodriguez and A.G. Suarez, *A new algorithm for image segmentation by using iteratively the mean shift filtering*, Sci. Res. Essay 1 (2006), pp. 43–48.

[21] B.W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman & Hall, London, 1986.

[22] K.C. Tan, R. Sathikannan, W.W. Tan, and A.P. Loh, *Evolutionary design and implementation of a hard disk drive servo control system*, Soft Computing 11 (2007), pp. 131–139.

[23] T.N. Tran, R. Wehrens, and L.M.C. Buydens, *KNN-kernel density-based clustering for high-dimensional multivariate data*, Comput. Statist. Data Anal. 51 (2006), pp. 513–525.

[24] T. Vo Van and T. Pham-Gia, *Clustering probability distributions*, J. Appl. Stat. 37 (2010), pp. 1891–1910.

[25] M.P. Wand and M.C. Jones, *Kernel Smoothing*, Chapman & Hall, London, 1994.

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

[27] R.R. Yager and D.P. Filev, *Foundations of Fuzzy Modeling and Control*, Wiley, New York, 1994.

[28] C. Yang, R. Duraiswami, D. DeMenthon, and L. Davis, *Mean-shift analysis using quasi-Newton methods*, Proceedings of the IEEE International Conference on Image Processing, Vol. 3, Barcelona, Spain, 14–18 September, 2003, pp. 447–450.

[29] K. Zhang, M. Tang, and J.T. Kwok, *Applying neighborhood consistency for fast clustering and Kernel density estimation*, Proceedings of the IEEE International Conference on Vision and Pattern Recognition, Vol. 2, San Diego, USA, 20–25 June 2005, pp. 1001–1007.