Reduction of Dimension and Size of Data Set by Parallel Fast Simulated Annealing

Piotr Kulczycki and Szymon Łukasik

Abstract A universal method of dimension and sample size reduction, designed for exploratory data analysis procedures, constitutes the subject of this paper. The dimension is reduced by applying linear transformation, with the requirement that it has the least possible influence on the respective locations of sample elements. For this purpose an original version of the heuristic Parallel Fast Simulated Annealing method was used. In addition, those elements which change the location significantly as a result of the transformation, may be eliminated or assigned smaller weights for further analysis. As well as reducing the sample size, this also improves the quality of the applied methodology of knowledge extraction. Experimental research confirmed the usefulness of the procedure worked out in a broad range of problems of exploratory data analysis such as clustering, classification, identification of outliers and others.

Keywords Dimension reduction · Sample size reduction · Linear transformation · Simulated annealing · Data analysis and mining

P. Kulczycki (✉) · S. Łukasik
Systems Research Institute, Centre of Information Technology for Data Analysis Methods,
Polish Academy of Sciences, Warsaw, Poland
e-mail: Piotr.Kulczycki@ibspan.waw.pl

S. Łukasik
e-mail: Szymon.Lukasik@ibspan.waw.pl

P. Kulczycki · S. Łukasik
Department of Automatic Control and Information Technology,
Cracow University of Technology, Cracow, Poland

L. T. Kóczy et al. (eds.), *Issues and Challenges of Intelligent Systems and Computational Intelligence*, Studies in Computational Intelligence 530,
DOI: 10.1007/978-3-319-03206-1_19, © Springer International Publishing Switzerland 2014

1 Introduction

Contemporary data analysis avails of a broad and varied methodology, based on both traditional and modern—often specialized—statistical procedures, currently ever more supported by the significant possibilities of computational intelligence. Apart from the classical methods—fuzzy logic and neural networks, metaheuristics such as genetic algorithms, simulated annealing, particle swarm optimization, and ants algorithms [1] are also being applied here more widely. The proper combination and exploitation of the advantages of these techniques allows in practice for the effective solution to fundamental problems in knowledge engineering, particularly those connected with exploratory data analysis.

More and more frequently the process of knowledge acquisition is realized using multidimensional data sets of large size. This stems from the dynamic growth in the amount of information collected in database systems requiring permanent processing. The extraction of knowledge from extensive data sets is a highly complex task. Here difficulties are mainly related to limits in efficiency of computer systems—for large-sized samples—and problems exclusively connected with the analysis of multidimensional data. The latter arise mostly from a number of phenomena occurring in data sets of this type, known in literature as "the curse of multidimensionality". Above all, this includes the exponential growth in sample size necessary to achieve appropriate effectiveness of data analysis methods with increasing dimension (the empty space phenomenon), as well as the vanishing difference between near and far points (norm concentration) using standard Minkowski distances [2].

As previously mentioned, the data set size can be reduced mainly to speed up or make at all possible calculations. In the classical approach, this is realized mostly with sampling methods or advanced data condensation techniques. Useful algorithms have also been worked out allowing the problem to be simplified by decreasing its dimensionality. Therefore, let X denote a data matrix of dimension $m \times n$:

$$X = [x_1 \,|x_2| \cdots |x_m\,]^{\mathrm{T}} \tag{1}$$

with particular m rows representing the realizations of an n-dimensional random variable.[1] The aim of reducing a dimension is to transform the data matrix in order to obtain its new representation of the dimension $m \times N$, where N is considerably—from the point of view of conditioning of a problem in question—smaller than n. This reduction can be achieved in two ways, either by choosing N most significant coordinates/features (feature selection) or through the construction of a reduced set, based on initial features (feature extraction) [3]. The latter can be treated as more general—the selection is a particularly simplecase of extraction. Noteworthy among

[1] Particular coordinates of a random variable of course constitute one-dimensional random variables and if the probabilistic aspects are not the subject of research, then in data analysis these variables are given the terms "feature" or "attribute".

extraction procedures are linear methods, where the resulting data set Y is obtained through the linear transformation of initial data set (1), therefore using the formula

$$Y = X \cdot A, \tag{2}$$

where A is a matrix of dimension $n \times N$, as well as nonlinear methods for which the transformation can be described by the nonlinear function $g : \mathbb{R}^n \to \mathbb{R}^N$. This group also contains the methods for which such a functional dependence, expressed explicitly, does not exist. Comparisons of effectiveness of extraction procedures carried out in subject literature show that nonlinear methods, despite having more general mathematical apparatus and higher efficiency in the case of artificially generated specific sets of data, for real samples frequently achieve significantly worse results [4].

The goal of this paper is to develop a universal method of reducing dimension and size of a sample designed for use in data exploration procedures. The reduction of the dimension will be implemented using a linear transformation on the condition that it affects as little as possible the mutual positions of original and resulting samples' elements. For this aim a novel version of the heuristic method of parallel fast simulated annealing will be researched. Moreover, those elements of a random sample which significantly change their position following transformation will be eliminated or assigned less weight for the purposes of further analysis. This concept achieves an improvement in quality of knowledge discovery and—possibly—a reduction in sample size. The effectiveness of the presented method will be verified for fundamental procedures in exploratory data analysis: clustering, classification and detection of atypical elements (outliers).

2 Preliminaries

2.1 Reduction in Dimension and Sample Size

The dimension can be reduced in many ways. Correctly sorting the procedures applied here requires, therefore, a wide range of criteria to be taken into account. Firstly the aforementioned systematic for linear and nonlinear methods is associated with character of dependence between initial and reduced data sets. Most important of these, a reference linear procedure for dimension reduction even, is the Principal Component Analysis (PCA). Among nonlinear methods the most often mentioned is Multidimensional Scaling (MDS). Reduction procedures are often considered with respect to facility of description of mapping between initial and reduced data sets. This can be defined as *explicite* (which allows to generalize the reduction procedure on points not belonging to initial data set), as well as given only *implicite*, i.e. through reduced representation of elements of an initial data set. The type of method chosen has particular significance in the cases of data analysis tasks, where a continuous influx of new information is present—in this form of problem, the reduction methods belonging to the first of the above groups are preferred. The third division of transformation procedures is related to their level of relationship with the data analysis algorithms used in the next step. It is worth noting here universal techniques which, through analogy to machine learning methods, can be termed as unsupervised. These work autonomously, without using results of exploration procedures [5]. The second category concerns algorithms dedicated to particular techniques in data analysis, in particular considering class labels. Here are often used statistical methods [6] as well as heuristic procedures of optimization, e.g. evolutionary algorithms [7].

A reduction in data set size can be realized with a wide range of sampling or grouping methods. The former most often uses random procedures or stratified sampling [8]. The latter applies either classical clustering techniques or special procedures for data condensation problems. There exists also a significant number of methods for reducing size which take into account additional knowledge, for example concerning whether elements belong to particular classes [9, 10]. Moreover methods dedicated to particular analytical techniques, for example kernel estimators [11, 12], have been developed (see e.g. [13]).

The method presented in this paper is based on a concept of dimension reduction which is linear, *explicite* defined and of universal purpose. Its closest equivalents can be seen to be the Principal Component Analysis method (due to its linear and unsupervised nature), feature selection using evolutionary algorithms [14] and the projection method with preserved distances [15–17], with respect to the similar quality criterion.

A natural priority for the dimension reduction procedure is maintaining distances between particular data sets elements—a wide range of methods treat this as a quality indicator. Typical for this group of algorithms is the classic multidimensional scaling, also known as principal coordinates analysis. It is a linear method, which creates the analytical form of the transformation matrix A, minimizing the index

$$S(A) = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \left(d_{ij}^2 - \delta_{ij}(A)^2 \right), \tag{3}$$

where d_{ij} denotes the distance between the elements x_i and x_j of the initial data set, while $\delta_{ij}(A)$ are respective distances in the reduced data set. A different strategy is required when searching for a solution with different structural characteristics or performance indicator, or else a nonlinear relation between initial and reduced data sets. This type of procedure is termed multidimensional scaling (MDS), mentioned before. A model example of this is nonlinear Sammon mapping, which—thanks to the application of a simple gradient algorithm—allows to find a reduced representation of the investigated data set, ensuring minimization of the so-called Sammon stress:

$$S_S(A) = \frac{1}{\sum_{i=1}^{m-1} \sum_{j=i+1}^{m} d_{ij}} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \frac{(d_{ij} - \delta_{ij}(A))^2}{d_{ij}}. \tag{4}$$

Such a defined criterion enables more homogenous treatment of small and large distances [18], while the value $S_S(A)$ is further normalized to the interval [0,1].

An alternative index, also considered in the context of MDS is so-called raw stress, defined by

$$S_R(A) = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \left(d_{ij} - \delta_{ij}(A)\right)^2. \tag{5}$$

The multidimensional scaling methods are mostly nonlinear procedures. However, the task was undertaken to formulate the problem of minimization of indexes (4) and (5) with assumed linear form of transformation. The first example of this technique is the algorithm for finding linear projection described in the paper [17]. Here an iterative method of greatest descent is applied, which gives in consequence better results than PCA in the sense of index (4). A similar procedure was investigated for function (5), with the additional possibility to successively supplement a data set [16]. In both cases the applied approach did not account for the multimodality of the stress function. To avoid becoming trapped in a local minimum one can use the appropriate heuristic optimization strategy. In particular, for minimization of index (4), the paper [14] uses the evolutionary algorithm. The solution for this investigation is, however, only to choose the reduced features set. A more effective approach seems to be the concept of their extraction—being more general, it will be the subject of investigation in this paper.

In the construction of the algorithm presented here, an auxiliary role is played by an unsupervised technique of feature selection using to this aim an appropriate measure of similarity—index of maximal compression of information [19]. It is based on the concept of dividing features into clusters, with the similarity criterion of features defined by the aforementioned index. This division is based on the algorithm of k-nearest neighbors, where it is recommended that $k \cong n - N$. The number of clusters achieved then approaches N, although it is not strictly fixed, but in a more natural manner is adapted to a real data structure.

Another aspect of the procedure presented here is a reduction in size of sample (1). Conceptually, the closest technique is the condensation method [20]. It is unsupervised and to establish the importance of elements takes into account their respective distances. In this case the algorithm of k-nearest neighbors is also applied, where the similarity measure between sample elements is Euclidean distance. Within this algorithm, in the data set are iteratively found prototype points, or points for which the distance r to the kth nearest neighbor is the smallest. With every iteration, elements closer than $2r$ from the nearest prototype point, are eliminated.

2.2 Simulated Annealing Algorithm

Simulated annealing (SA) is a heuristic optimization algorithm, based on the iterative technique of local search with appropriate criterion for accepting solutions. This allows to establish a valid solution for every iteration, mostly using the quality index value for the previous and current iteration, and variable parameter called the annealing temperature, which decreases in time. In this way it becomes possible to accept a valid solution worse than the previous, thereby reducing the danger of the algorithm getting stuck at local minimums. In addition it is assumed that the probability of accepting a worse solution should decrease over time. All of the above traits contain the so-called Metropolis rule, which is most often applied as acceptance criterion in simulated annealing algorithms.

Let therefore $Z \subset \mathbb{R}^t$ denote the set of admissible solutions to a certain optimization problem, while the function $h : Z \to \mathbb{R}$ is its quality index, hereinafter referred to as cost. Furthermore, let $k = 0, 1, \ldots$ mean the number of iteration, whereas $T(k) \in \mathbb{R},\ z(k) \in Z,\ c(k) = h\left(z(k)\right),\ z_0(k) \in Z,\ c_0(k) = h\left(z_0(k)\right)$—respectively—temperature and solution valid for the iteration k and its cost, and also the best solution found to date and its cost. Under the above assumptions the basic variant of the SA algorithm can be described thus:

```
procedure Simulated_annealing
  begin
    Generate(T(1),z(0))
    c(0)= Evaluate_quality(z(0))
    z0(0)= z(0)
    c0(0)= c(0)
    k = 1
    repeat
      z(k)= Generate_neighbor(z(k-1))
      c(k)= Evaluate _quality(z(k))
      Δc = c(k) - c(k-1)
      z(k)= Metropolis_rule(Δc,z(k),z(k-1),T(k))
      if c(k) < c0(k-1)
            z0(k)= z(k)
            c0(k)= c(k)
        else
            z0(k)= z0(k-1)
            c0(k)= c0(k-1)
        Calculate(T(k+1))
      stop_condition = Check_stop_condition()
      k=k+1
    until stop_condition == FALSE
return kstop=k-1, z0(kstop), c0(kstop)
end
```

where the procedure for the Metropolis rule is realized by

```
procedure Metropolis_rule(Δc,z(k),z(k-1),T(k))
  if Δc < 0
    return z(k)
  else
    if Random_number_from_(0,1) < exp(-Δc/T(k))
```

```
        return z(k)
    else
        return z(k-1)
end
```

The SA algorithm requires in the general case the assumption of the appropriate initial temperature value, formula of its changes associated with an accepted method of generating a neighboring solution, as well as a condition for ending the procedure. However in particular applications one should also define other functional elements, such as method of generating the initial solution and form of the quality index. The first group of tasks will now be discussed, while the second—as specific for the application of the SA algorithm investigated here—will be the subject of detailed analysis in Sect. 3.

Numerous fundamental and applicational works have resulted in creation of many variants of the algorithm described here. Their main difference is the scheme for temperature changes and method for obtaining a neighboring solution. The standard approach is the classical simulated annealing algorithm, also known as the Boltzmann annealing algorithm (BA). This assumes an iterative change in temperature according to a logarithmic schedule and generation of a subsequent solution by adding to the current one the value of step $\Delta Z \in \mathbb{R}^t$, which is the realization of t-dimensional pseudorandom vector with normal distribution. The BA algorithm—although effective in the general case—has a large probability of acceptance of worse solutions, even in the final phase of the search process. This allows for the effective escape from local minimums of a cost function and guarantees asymptotic convergence to a global one [21], while also ensuring the procedure represents—in some sense—a random search of the space of admissible solutions. For the SA algorithm to be more deterministic in character, and at the same time keeping convergence to the optimal solution, the following scheme for temperature change can be applied:

$$T\left(k+1\right) = \frac{T\left(1\right)}{k+1}, \tag{6}$$

together with the generation of neighboring solution using a Cauchy distribution

$$g(\Delta z) = \frac{T\left(k\right)}{\left(\Delta \mathrm{z}^2 + T(k)^2\right)^{(t+1)/2}}. \tag{7}$$

The procedure defined by the above elements is called Fast Simulated Annealing (FSA) [22]. It will be a base—in the framework of this paper—for the dimension reduction algorithm.

The problem of practical implementation of FSA is the effective generation of random numbers with multidimensional Cauchy distribution. The simplest solution is the application for each dimension of the vector, of a one-dimensional number generator with the same distribution. This strategy was used in the Very Fast Simulated Annealing algorithm (VFSA), expanded later within the framework of a complex procedure of Adaptive Simulated Annealing [23]. Such a concept has, however, a fundamental flaw: the step vectors generated here concentrate near the axes of the coordinate system. An alternative could be to use a multidimensional generator based on the transformation of the Cartesian coordinate system to a spherical one. It is suggested here that the step vector $\Delta z = [\Delta z_1, \Delta z_2, \ldots, \Delta z_t]$ be obtained by generating first the radius r of the hypersphere, using the method of inverting the Cauchy distribution function described with the spherical coordinates, and then selecting the appropriate point on the t-dimensional hypersphere. The second phase is realized by randomly generating the vector $u = [u_1, u_2, \ldots, u_t]^T$ with coordinates originating from the one-dimensional normal distribution $u_i \sim N(0, 1)$, and then the step vector Δz:

$$\Delta z_i = r \frac{u_i}{|\mathrm{u}|}, \quad i = 1,\ 2, \ldots, t. \tag{8}$$

The presented procedure ensure a symmetric and multidirectional generation scheme, with heavy tails of distribution, which in consequence causes effective exploration of a solution space [24]. Taking the above into account, it has been applied in the algorithm investigated in this paper.

Establishing an initial temperature is vital for the correct functioning of the simulated annealing algorithm. It implies the probability of acceptance of a worse solution at subsequent stages of the search in the solutions space. Subject literature tends to suggest choosing the initial temperature so that the probability of acceptance of a worse solution at the first iteration, denoted hereinafter as $P(1)$, is relatively large. These recommendations are not absolute, however, and different proposals can be found in literature, for example close to 1.0 [25], around 0.8 [26] or even only 0.5 [27]. Often in practical applications of SA algorithms, the temperature value is fixed during numerical experiments [28]. An alternative is to choose a temperature according to a calculational criterion which has the goal of obtaining $T(1)$ the value on the basis of a set of pilot iterations, consisting of generating the neighbor solution $z(1)$ so that the assumed $P(1)$ value is ensured. For this purpose one can—analyzing the mean difference in cost between the solutions $z(1)$ and $z(0)$, denoted as $\overline{\Delta c}$ in the following—calculate the temperature $T(1)$ value by substituting $\overline{\Delta c}$ to the right-side of the inequality in the Metropolis rule defining the probability of the worse solutions acceptance:

$$\mathrm{P}\,(1) = e^{-\frac{\overline{\Delta c}}{T(1)}}, \tag{9}$$

thus in consequence

$$T\ (1) = -\frac{\overline{\Delta c}}{\mathrm{In}\ P(1)}. \tag{10}$$

The mean difference in cost can be replaced with e.g. the standard deviation of the cost function value, marked as $\overline{\sigma}_c$, also estimated on the basis of the set of pilot iterations [29]. A problem which appears in the case of SA algorithms dedicated to minimizing functions with real arguments (including the aforementioned BA, FSA, VFSA and ASA), is the dependence of the strategy for generating a neighbor solution on temperature. Therefore, both the standard deviation $\overline{\sigma}_c$ and the mean $\overline{\Delta c}$

are directly dependent on it. The application of formula (10) is not possible here and in the case of these algorithms, the initial temperature value is usually arbitrary. This paper proposes a different strategy based on the generation of a set of pilot iterations, allowing the value $T(1)$ to be obtained on the assumption of any value of initial probability of worse solutions acceptance.

As equally important as the choice of initial temperature is the determination of the iteration at which the algorithm should be terminated. The simplest—although not flexible and often requiring too detailed knowledge of the investigated task—stop criterion is reaching a previously assumed number of iterations or a satisfactory cost function value. An alternative could be to finish the algorithm when following a certain number of iterations, the best obtained solution is not improved, or the use of an appropriate statistical method based on the analysis of cost function values as they are obtained. The last concept is universal and—desirable among heuristic algorithms stop criterions—related to the expected result of their works. This usually consists of calculating the estimator of expected value of the global minimum $\hat{c}_{min}$ and finishing the algorithm in the iteration k, when the difference between it and the discovered smallest value $c_0(k)$ is not greater than the assumed positive ε, so if

$$|c_0\,(k)\;-\hat{c}_{min}\,|\leq \varepsilon. \tag{11}$$

One the most recent techniques using this type of strategy is the algorithm proposed in the work [30]. In order to calculate the value c_{min} an estimator is applied here based on order statistics [31]. This algorithm constitutes a universal and effective tool for a wide range of stochastic optimization techniques. Such a method, used as part of the FSA procedure, will now be described.

Let therefore $\{c_0(k), c_1(k), c_2(k), \ldots, c_r(k)\}$ denote the ordered non-decreasing set of r lowest cost function values, obtained during k iterations of the algorithm. In the case of an algorithm convergent on a global minimum, the condition $\lim_{\mathrm{k}\to\infty} c_j(k) = c_{min}$ is fulfilled for every $j \in \mathbb{N}$, while the sequences $c_j(k)$ can be applied to construct the aforementioned estimator value c_{min}. This estimator makes use of the assumption of asymptotic convergence of order statistic distribution to the Weibull distribution, and in the iteration k takes the general form:

$$\hat{c}_{min}\,(k) = c_0\,(k) - \frac{\frac{2t}{\beta} - 1}{r}\,(c_r\,(k) - c_0\,(k))\,. \tag{12}$$

The parameter β occurring above is termed a homogenous coefficient of the cost function h around its minimum. On additional assumptions, in calculational practice one can take $\beta = 2$ [32]. The confidence interval for the cost function minimum, at the assumed significance level $\delta \in (0, 1)$, is of the form

$$\left[c_0\,(k) - \frac{\left(1-(1-\delta)^{1/r}\right)^{\beta/t}}{1-\left(1-(1-\delta)^{1/r}\right)^{\beta/t}}\,(c_r\,(k) - c_0\,(k))\,, c_0\,(k)\right]. \tag{13}$$

The paper [30] suggests that point estimator (12) can be replaced by confidence interval (13) with the algorithm being stopped when the confidence interval width is less than the aforementioned, assumed value ε. Such an idea, modified for the specific problem under investigation, will be applied here.

The simulated annealing procedure can be easily parallelized, whether for required calculations, or in the scheme of establishing subsequent solutions. While parallelizing the SA algorithm is not a new idea, and was already investigated a few years after its creation [33], it needs to be adapted for particular applicational tasks [34]. At present the suitability of the Parallel Simulated Annealing (PSA) algorithm continuously increases with the common availability of multicore systems. In the algorithm worked out in this paper, a variant will be taken with parallel generation of neighbor solutions, assuming that the number of SA threads equals the number of available processing units.

3 Procedure for Reducing Dimension and Sample Size

The algorithm investigated in this paper consists of two functional components: a procedure for reducing the dimension and a way of allowing sample size to be decreased. They are implemented sequentially, with the second dependent on the results of the first. The reduction of sample size is optional here.

3.1 Procedure for Dimension Reduction

The aim of the algorithm under investigation is a decrease in the dimensionality of the data set elements, represented by the matrix X with the form specified by formula (1), so of the dimension $m \times n$, where m means the data set size, and n—the dimension of its elements. In consequence the reduced form of this data set is represented by the matrix Y of the dimension $m \times n$, while N denotes the assumed reduced dimension of elements, appropriately less than n. The procedure for reducing the dimension is based on linear transformation (2), with the matrix A given in form

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nN} \end{bmatrix}, \tag{14}$$

although for the purposes of notation used in the simulated annealing algorithm, its elements are denoted as the row vector

$$[a_{11},\ a_{12},\ \ldots,\ a_{1N},\ a_{21},\ a_{22},\ \ldots,\ a_{2N},\ \ldots\ ,a_{n1},\ a_{n2},\ \ldots\ ,\ a_{nN}], \tag{15}$$

which represents the current solution $z(k) \in \mathbb{R}^{n \cdot N}$ in any iteration k. In order to generate neighbor solutions, a strategy was used based on the multidimensional generator of the Cauchy distribution (formulas (7) and (8)). The quality of the obtained solution can be evaluated with the application of the cost function h, which is the function of the raw stress S_R given by dependence (5), where the matrix Y elements are defined on the basis of Eq. (2). The alternative possibility of using Sammon stress (4) for this purpose was also examined.

The developed procedure requires firstly that the basic parameters are specified: the dimension of the reduced space N, a coefficient defining directly the maximum allowed width of the confidence interval ε_w for the stop criterion based on the order statistics, the number of processing threads of the FSA procedure p_{thread}, initial scaling coefficient (length of step) for the multidimensional Cauchy generator T_{scale}, as well as the probability of acceptance of a worse solution $P(1)$ in the first iteration of the FSA algorithm.

Starting the algorithm requires moreover the generation of the initial solution $z(0)$. To this aim the feature selection procedure of [19], described in the previous section, was realized. Here $k = n - N$ should be assumed, which in consequence usually results in obtaining approximately N clusters. The aforementioned procedure described leads to getting the auxiliary vector $b \in \mathbb{R}^n$, the particular coordinates of which characterize the number of cluster, to which the coordinate from the original space was mapped, as well as the vector $b_r \in \mathbb{R}^n$ of binary values $b_r(i) \in \{0, 1\}$ for $i = 1, 2, \ldots, n$, defining whether a given feature was chosen as a representative of the cluster to which it belongs, in which case $b_r(i) = 1$, or not—then $b_r(i) = 0$. The auxiliary vectors b and b_r can be used in the considered algorithm for generating the initial solution in two ways:

1. Each of N features of the initial solution is a linear combination of features mapped to one of N clusters—to define the form of the matrix A one can use

$$\begin{cases} a_{ij} = 1, & \text{if } b(i) = j \\ a_{ij} = 0, & \text{if } b(i) \neq j \end{cases} \quad \text{for} \quad i = 1, 2, \ldots, n \quad \text{and} \quad j = 1, 2, \ldots, N\,. \tag{16}$$

2. Each of N features of the initial solution is given as representative for one of N clusters—the form of the matrix A is then defined as

$$\begin{cases} a_{ij} = 1, & \text{if } b_r(i) = 1 \quad \text{and} \quad b(i) = j \\ a_{ij} = 0, & \text{if } b_r(i) = 0 \end{cases} \quad \text{for} \quad i = 1, 2, \ldots, n \quad \text{and} \quad j = 1, 2, \ldots, N\,. \tag{17}$$

The possibility of applying both the above ways of generating an initial solution—the first called a linear combination of features and the second, referred to as features selection—is a subject of detailed experimental analysis concerning dimensional reduction, described in Sect. 4.

After generating the initial solution, in order to carry out the simulated annealing algorithm, the temperature $T(1)$ should be fixed in the first iteration. To this aim the technique presented in the previous section can be followed, allowing at the start to obtain the assumed initial value of the probability of worse solution acceptance $P(1)$. In the case of the algorithm for generating neighbor solutions, it is not recommended to use the relation resulting from equality (9). As mentioned in the previous section, this is implied by the dependence of a formula for generating a neighbor solution on the annealing temperature. In order to avoid this inconvenience, an additional coefficient T_{scale} was introduced, being the parameter of the Cauchy distribution in the first iteration of the FSA algorithm (also known as an initial step length), and furthermore the temperature occurring in the generating distribution was scaled. The coefficient T_{scale} is thus used as a parameter of the random numbers generator, with the aim of calculating a set of pilot iterations (the size of this set is assumed to be 100). These iterations consist of the generation of an appropriate number of transitions from $z(0)$ worse in the sense of cost function used, to the neighbor solution $z(1)$, and the establishment of the mean value of the cost difference $\overline{\Delta c}$ between $z(1)$ and $z(0)$. This value is inserted to formula (10), through which the initial temperature can be obtained. Moreover, in order to find the assumed shape of the generated distribution, in the first iteration of the FSA algorithm, the additional scaling coefficient is calculated:

$$c_{temp} = -\frac{\overline{\Delta \mathrm{c}}}{\mathrm{In}\ P(1)T_{scale}}. \tag{18}$$

In consequence, in the first iteration of the actual algorithm, in order to generate a neighbor solution, the scaled temperature $T(1)/c_{temp}$ (therefore T_{scale}) is used, and for the Metropolis rule—just the value $T(1)$. Similar scaling takes place during the generation of neighbor solutions in each consecutive iteration of the FSA algorithm. Thanks to this kind of operation it becomes possible to fix the initial probability of acceptance of a worse solution, which determined by the coefficient $P(1)$, while retaining the additional possibility of establishing—by assuming the value T_{scale}—the parameter of initial spread of values obtained from a pseudorandom numbers generator.

All iterations of the FSA algorithm have been parallelized using a strategy with parallel generation of neighbor solutions. So each of p_{thread} threads creates a solution neighboring the one established in the previous iteration $z(k-1)$. This occurs with the application of a random generator with multidimensional Cauchy distribution. For all threads, the annealing temperature is identical and equals $T(k)/c_{temp}$. Furthermore, every thread realizes the procedure for the Metropolis rule, accepting or rejecting its own obtained neighbor solutions.

The next two steps of the algorithm are performed in sequence. So, first the current solution is fixed for the SA algorithm. The procedure for this is to choose as a current solution either the best from those better than that found in the previous iteration obtained by different threads, or—if such a solution does not exist—random selection of one of the worse solutions. Calculated thus, the current solution, together with the temperature updated according to formula (6), is also used in the next iteration of the FSA algorithm as the current solution. This kind of strategy can be classified as a method of parallel processing based on speculative decomposition.

The last step performed as part of single iteration is verifying the criterion for stopping the SA procedure. To this aim the confidence interval for the minimum value of the cost function, given by formula (13), is calculated. The order statistics used for interval estimation have the order r assumed as 20, in accordance with the proposals of the paper [30]. As a significance level δ for the confidence interval defined by formula (13), a typical value 0.99 [35] is assumed. The width of the confidence interval is compared with the threshold value ε calculated at every iteration with

$$\varepsilon = 10^{-\varepsilon_w} c_0\left(k\right). \tag{19}$$

Finally, the simulated annealing procedure is terminated if

$$\frac{\left(1-\left(1-\delta^{1/\mathrm{r}}\right)^{\beta/t}\right)}{1-\left(1-\left(1-\delta^{1/\mathrm{r}}\right)^{\beta/t}\right)}\left(c_{\mathrm{r}}\left(\mathrm{k}\right)-c_0\left(\mathrm{k}\right)\right) > \ \varepsilon, \tag{20}$$

with notations introduced at the end of Sect. 2. Finding the threshold value ε based on formula (20) allows the adaptation of a such defined criterion to a structure of a specific data set under investigation. The sensitivity of the above procedure can be set by assuming the value of the exponent $\varepsilon_w \in \mathbb{N}$, one of the arbitrarily fixed parameters of the procedure worked out in this paper.

It is worth noting that the nature of the method presented here for dimension reduction enables establishment of the "contribution" which particular elements of the data set Y make to the final value $c_0\left(k_{stop}\right)$. This fact will be used in the procedure for reducing the sample size, which will be presented in Sect. 3.2.

3.2 Procedure for Sample Size Reduction

In the case of the dimension reduction method presented above, some sample elements may be subject to an undesired shift with respect to the others and, as a result, may noticeably worsen the results of an exploratory data analysis procedure in the reduced space $\mathbb{R}^N$. A measure of the location deformation of the single sample element x_i compared to the others, resulting from transformation (2), is the corresponding stress value $c_0(k_{stop})$ calculated for this point (called stress per point) [36]. In the case of raw stress it is given by

$$c_0(k_{stop})_i = S_R(A)_i = \sum_{\substack{j=1 \\ j\neq i}}^{m} (d_{ij} - \delta_{ij}(A))^2, \tag{21}$$

whereas for the Sammon stress it takes the form

$$c_0(k_{stop})_i = S_s(A)_i = \frac{1}{\sum_{i=1}^{m-1}\sum_{j=i+1}^{m} d_{ij}} \sum_{\substack{j=i \\ j\neq i}}^{m} \frac{d_{ij} - \delta_{ij}(A))^2}{d_{ij}}. \tag{22}$$

It is worth noticing that in both cases these values are nonzero, except for the case—unattainable in practice—of "perfect" matching of respective distances of elements in original and reduced spaces. The values $c_0(k_{stop})_i$ for particular elements can be used to construct a set of weights, defining the adequacy of their location in a reduced space.

Let therefore w_i represent nonnegative weight mapped with the element x_i. Taking the above into account, it is calculated according to the following formula:

$$w_i = \frac{m\frac{1}{c_0(k_{stop})_i}}{\sum_{i=1}^{m}\frac{1}{c_0(k_{stop})_i}}. \tag{23}$$

The normalization which occurs in the above dependence guarantees the condition

$$m = \sum_{i=1}^{m} w_i. \tag{24}$$

The weights in this form contain information as to the degree to which a given sample element changed its relative location compared to the rest, where the larger the weight, the more relatively adequate its location, and its significance should be greater for procedures of exploratory data analysis carried out in a space of reduced dimension.

The weights' values which are calculated on the basis of the above formulas can be used for further procedures of data analysis. They also allow the following method of reducing sample size. Thus, from the reduced data set Y one can remove those m_{el} elements, for which their respective weights fulfill the condition $w_i < W$ with assumed $W > 0$. Intuitively $W = 1$ is justified—taking into account formula (24), this results in the elimination of elements corresponding to the values w_i less than the mean.

In conclusion, conjoining the methods from Sects. 3.1 and 3.2 enables a data set with reduced dimension as well as size to be obtained, with the degree of compression implied by the parameters N and W values.

3.3 Comments and Suggestions

In the case of the procedure for reducing dimension and sample size presented here, efforts were made to limit the number of parameters, the arbitrary selection of which is always a significant practical problem for heuristic algorithms. At the same time, conditioning of data analysis tasks, in which the procedure investigated here will be applied, means that—from a practical point of view—it is useful to propose specific values of those parameters with an analysis of the influence of their potential changes.

One of the most important arbitrarily assumed parameters is the reduced space dimension N. It can be fixed initially using one of the methods for estimating a hidden dimension [37], or by taking a value resulting from other requirements, for example $N = 2$ or 3 to enable a suitable visualization of the investigated data set. It is worth remembering that the procedure applied earlier for generating an initial solution with the fixed parameter $k = n - N$, creates a solution which does not always have a dimensionality identical to the assumed (as mentioned in Sect. 3.2). If a strictly defined dimension of the reduced data set is required, one should adjust the parameter k by repeating the feature selection algorithm with its correctly modified value, or use the initial solution, generated randomly, of assumed dimension of reduced space.

It is also worth mentioning the problem of computational complexity of the procedure worked out here, in particular regarding calculation of the cost function value. In practice, the calculational time for the PSA algorithm increases exponentially with an increase in sample size. So, despite the heuristic algorithm being the only method available in practice to minimize the stress function S_S or S_R for data sets of large dimensionality and size, its application must, however, be limited to those cases which are in fact feasible. Therefore, although the number of simulated annealing treads can be fixed at will, it should take the available number of processing units into account. This allows efficient parallel calculation of a cost function value by particular threads.

It should also be noted that the subject algorithm, due to its universal character, can be applied to a broad range of problems in statistics and data analysis. An example, from the case of statistical kernel estimators [11, 12], is the introduction of generalization of the basic definition of the estimator of probability distribution density to the following form:

$$\hat{f}(y) = \frac{1}{h^n \sum_{i=1}^{m} w_i} \sum_{i=1}^{m} w_i K\left(\frac{y - y_i}{h}\right). \tag{25}$$

Such a concept allows not only a reduction of sample size (for removed elements $w_i = 0$ is assumed), but also alternatively—an improvement in quality of estimation in the reduced space without eliminating any elements from the initial data set. In the former case care should also be given to normalize the weights after eliminating parts of elements, to fulfill condition (24).

Weights w_i calculated in the above manner can also be introduced to modified classical methods of data analysis, such as a weighted k-means algorithm [39], or a weighted technique of k-nearest neighbors [40]. In the first case, the weights are activated in the procedure for determining centers of clusters. The location of the center of the cluster C_i, denoted by $s_i = [s_{i1}, s_{i2}, \ldots, s_{iN}]^{\mathrm{T}}$, is updated in every iteration if $\sum_{y_l \in C_i} w_l \neq 0$, according to formula

$$s_{ij} = \frac{1}{\sum_{y_l \in C_i} w_l} \sum_{y_l \in C_i} w_l y_{lj} \quad \text{for} \quad j = 1, 2, ..., N. \tag{26}$$

In the k-nearest neighbors procedure however, each distance from neighbors of any element from the learning set is scaled using the appropriate weight.

4 Summary and Final Remarks

The subject of this paper was a complete algorithm for reducing dimension and sample size, ready to use in a wide range of exploratory data analysis problems. It constitutes a universal, unsupervised linear transformation of a features space, with the aim of best maintaining distances between sample elements, additionally supplemented by a reduction in significance of those elements whose locations in relation to the rest have changed considerably. The foundation for this algorithm is an innovative version of the parallel fast simulated annealing procedure, with stop criterion based on order statistics, automatic generation of initial temperature and a multidimensional generator of pseudorandom numbers with Cauchy distribution. The sample size was reduced as a result of calculating weights for particular elements, with the possibility of continuous adjustment of this procedure's intensity, through establishing an appropriate—for an investigated problem—value for the compression coefficient.

The presented methodology underwent detailed numerical testing. The basic research was carried out on the functionality of the method worked out, in particular the sensitivity to its assumed version and parameters. In general one can note that the proposed algorithm is not particularly sensitive to the choice of these parameters, which may be said to be, in practice, its valuable property. Further testing compared results with selected reference methods, especially the classic PCA procedure and the aforementioned selection of features by evolutionary algorithms, in the range of reduction of sample size, joined with an algorithm for data compression as presented in the paper [20]. In general, the results achieved with the application of the procedures investigated in this paper were frequently better, often significantly, than the reference methods mentioned before.

It should be also noted that the particular functional components of the procedure presented here can be applied in other tasks of information processing. Thus, the parallel fast simulated annealing algorithm may be used successfully in a wide range of optimization problems, thanks to its universal structure and relatively intuitive selection of arbitrarily assumed parameters. What is more, the proposed procedure for reducing the sample size can equally be applied together with other, also nonlinear, strategies for dimension reduction.

Finally, it is worth stressing that, despite the calculational complexity of the proposed algorithm, its execution—thanks to the possibility of creating highly efficient parallel implementation—is not very time-consuming. Even for the most complex of the tested data sets, it took only a few minutes while, thanks to the application of an adaptive stop criterion, this time is fitted to the difficulty of the problem under analysis, thus eliminating the need to introduce arbitrary assumptions. And lastly, due to the use of linear transformation, which is easy to generalize, and the simple idea of a set of weights, it is possible to use the investigated method effectively in a wide range of contemporary data analysis problems, from the areas of engineering, medicine, economics and social sciences, to name a few.

A detailed description of the methodology presented here can be found in the work [40] as well as in the paper [41] which will appear soon.

References

1. Gendreau, M., Potvin, J.-Y.: Handbook of Metaheuristics. Springer, New York (2010)
2. Francois, D., Wertz, V., Verleysen, M.: The concentration of fractional distances. IEEE Trans. Knowl. Data Eng. **19**, 873–886 (2007)
3. Xu, R., Wunsch, D.C.: Clustering. Wiley, New Jersey (2009)
4. van der Maaten, L.J.P.: Feature extraction from visual data. Ph.D. thesis, Tilburg University (2009)
5. Bartenhagen, C., Klein, H.-U., Ruckert, C., Jiang, X., Dugas, M.: Comparative study of unsupervised dimension reduction techniques for the visualization of microarray gene expression data. BMC Bioinformatics, 11, paper no 567 (2010)
6. Yan, S., Xu, D., Zhang, B., Zhang, H.-J., Yang, Q., Lin, S.: Graph embedding and extensions: a general framework for dimensionality reduction. IEEE Trans. Pattern Anal. Mach. Intell. **29**, 40–51 (2007)
7. Rodriguez-Martinez, E., Goulermas, J.Y., Tingting, M., Ralph, J.F.: Automatic induction of projection pursuit indices. IEEE Trans. Neural Netw. **21**, 1281–1295 (2010)
8. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco (2006)
9. Kulczycki, P., Kowalski, P.A.: Bayes classification of imprecise information of interval type. Control Cybern. **40**, 101–123 (2011)
10. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. Mach. Learn. **38**, 257–286 (2000)
11. Kulczycki, P.: Estymatory jadrowe w analizie systemowej. WNT, Warsaw (2005)
12. Wand, M.P., Jones, M.C.: Kernel Smoothing. Chapman and Hall, London (1995)
13. Deng, Z., Chung, F.-L., Wang, S.: FRSDE: fast reduced set density estimator using minimal enclosing ball approximation. Pattern Recogn. **41**, 1363–1372 (2008)
14. Saxena, A., Pal, N.R., Vora, M.: Evolutionary methods for unsupervised feature selection using Sammon's stress function. Fuzzy Inform. Eng. **2**, 229–247 (2010)
15. Sammon, J.W.: A nonlinear mapping for data structure analysis. IEEE Trans. Comput. **18**, 401–409 (1969)
16. Strickert, M., Teichmann, S., Sreenivasulu, N., Seiffert, U.: 'DIPPP' online self-improving linear map for distance-preserving data analysis. 5th Workshop on Self-Organizing Maps (WSOM), Paris, 5–8 September 2005, pp. 661–668 (2005)
17. Vanstrum, M.D., Starks, S.A.: An algorithm for optimal linear maps. Southeastcon Conference, Huntsville, 5–8 April 1981, pp. 106–110 (1981)
18. Cox, T.F., Cox, M.A.A.: Multidimensional Scaling. Chapman and Hall, Boca Raton (2000)
19. Pal, S.K., Mitra, P.: Pattern Recognition Algorithms for Data Mining. Chapman and Hall, Boca Raton (2004)
20. Mitra, P., Murthy, C.A., Pal, S.K.: Density-based multiscale data condensation. IEEE Trans. Pattern Anal. Mach. Intell. **24**, 734–747 (2002)

21. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distribution and the Bayesian restoration in images. IEEE Trans. Pattern Anal. Mach. Intell. **6**, 721–741 (1984)
22. Szu, H., Hartley, R.: Fast simulated annealing. Phys. Lett. A **122**, 157–162 (1987)
23. Ingber, L.: Adaptive simulated annealing (ASA): lessons learned. Control Cybern. **25**, 33–54 (1996)
24. Nam, D., Lee, J.-S., Park, C.H.: N-dimensional Cauchy neighbor generation for the fast simulated annealing. IEICE Trans. Inf. Syst. **E87–D**, 2499–2502 (2004)
25. Aarts, E.H.L., Korst, J.H.M., van Laarhoven, P.J.M.: Simulated annealing. In: Aarts, E.H.L., Lenstra, J.K. (eds.) Local Search in Combinatorial Optimization. Wiley, Chichester (1997)
26. Ben-Ameur, W.: Computing the initial temperature of simulated annealing. Comput. Optim. Appl. **29**, 367–383 (2004)
27. Kuo, Y.: Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. Comput. Ind. Eng. **59**, 157–165 (2010)
28. Mesgarpour, M., Kirkavak, N., Ozaktas, H.: Bicriteria scheduling problem on the two-machine flowshop using simulated annealing. Lect. Notes Comput. Sci. **6022**, 166–177 (2010)
29. Sait, S.M., Youssef, H.: Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems. IEEE Computer Society Press, Los Alamitos (2000)
30. Bartkute, V., Sakalauskas, L.: Statistical inferences for termination of Markov type random search algorithms. J. Optim. Theory Appl. **141**, 475–493 (2009)
31. David, H.A., Nagaraja, H.N.: Order Statistics. Wiley, New York (2003)
32. Zhigljavsky, A., Zilinskas, A.: Stochastic Global Optimization. Springer-Verlag, Berlin (2008)
33. Azencott, R.: Simulated Annealing: Parallelization Techniques. Wiley, New York (1992)
34. Alba, E.: Parallel Metaheuristics: A New Class of Algorithms. Wiley, New York (2005)
35. Kendall, M.G., Stuart, A.: The Advanced Theory of Statistics; Vol. 2: Inference and Relationship. Griffin, London (1973)
36. Borg, I., Groenen, P.J.F.: Modern Multidimensional Scaling. Theory and Applications. Springer-Verlag, Berlin (2005)
37. Camastra, F.: Data dimensionality estimation methods: a survey. Pattern Recogn. **36**, 2945–2954 (2003)
38. Kerdprasop, K., Kerdprasop, N., Sattayatham, P.: Weighted k-means for density-biased clustering. Lect. Notes Comput. Sci. **3589**, 488–497 (2005)
39. Parvin, H., Alizadeh, H., Minati, B.: A modification on k-nearest neighbor classifier. Glob. J. Comput. Sci. Technol. **10**, 37–41 (2010)
40. Łukasik, S.: Algorytm redukcji wymiaru i licznosci proby dla celow procedur eksploracyjnej analizy danych. Ph.D. thesis, Systems Research Institute, Polish Academy of Sciences (2012)
41. Kulczycki, P., Łukasik, S.: An algorithm for reducing dimension and size of sample for data exploration procedures. In press (2014)