An Algorithm for Sample and Data Dimensionality Reduction Using Fast Simulated Annealing

Szymon Łukasik[1] and Piotr Kulczycki[2]

[1] Department of Automatic Control and Information Technology, Cracow University of Technology
ul. Warszawska 24, 31-155 Cracow, Poland
[2] Systems Research Institute, Polish Academy of Sciences,
ul. Newelska 6, 01-447 Warsaw, Poland
szymonl@pk.edu.pl
kulczycki@ibspan.waw.pl

Abstract. This paper deals with dimensionality and sample length reduction applied to the tasks of exploratory data analysis. Proposed technique relies on distance preserving linear transformation of given dataset to the lower dimensionality feature space. Coefficients of feature transformation matrix are found using Fast Simulated Annealing - an algorithm inspired by physical annealing of solids. Furthermore the elimination or weighting of data elements which, as an effect of above mentioned transformation, were moved significantly from the rest of the dataset can be performed. Presented method was positively verified in routines of clustering, classification and outlier detection. It ensures proper efficiency of those procedures in compact feature space and with reduced data sample length at the same time.

Key words: dimensionality reduction, sample reduction, linear transformation, fast simulated annealing, cluster analysis, classification, outlier detection

1 Introduction

Modern data analysis has in its disposal a variety of methods based on both traditional and modern statistical techniques reinforced by soft computing procedures. Here, beside classical tools like fuzzy logic, neural networks and genetic algorithms, recent metaheuristics like particle swarm optimization, ant colony algorithms or bees optimization are frequently in use. Proper connection of algorithms' advantages enables their effective application in problems of contemporary knowledge engineering and data mining in particular. The subject of presented research is a concept of using nature-inspired Simulated Annealing algorithm [7] for the purpose of data dimensionality and sample size reduction.

Recently, the subject of data analysis are more and more frequently high dimensional datasets with huge sample lengths. It is a result of growing amount

of information stored in data warehouses. Extraction of knowledge from such datasets is a very complicated task. Difficulties include mainly limitations of computer systems' performance when considering huge samples and methodological obstacles of high dimensional data analysis. The latter is connected with properties of such datasets referred in bibliography as "curse of dimensionality" (this term was used for the first time by Bellman in the context of control systems design) [21]. It includes exponential grow of sample size needed to achieve proper efficiency of data analysis with increasing dimensionality, so called "empty space phenomenon" and vanishing of distances between close and distant points when using typical Minkowski norm.

To overcome above-mentioned problems adequate reduction procedures were developed. Sample length reduction is performed usually by means of sampling techniques [2] or advanced data condensation routines [14] and its expected result is mainly speeding up calculation time associated with data mining process. Dimensionality reduction can be performed in numerous ways. Let X to denote $n \times m$ data matrix:

$$X = \left[x_1 \; x_2 \; ... \; x_m \right] \tag{1}$$

columns of which represent n dimensional sample elements for given probabilistic variable. Each dimension of such variable will be referred later in this paper as a feature. The aim of dimensionality reduction is a data transformation to a new $N \times m$ sized form, where N is significantly smaller than n. This can be achieved either by selecting most N significant features (feature selection) or by construction of a new set of N features based on the initial ones (i.e. by feature extraction). The second case is more general and will be considered in this work.

Among feature extraction procedures one can distinct: linear methods where synthesis of resulting dataset Y is performed by linear transformation:

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

with A being a transformation matrix of size $N \times n$ and nonlinear techniques where data transformation can be described by a nonlinear function $g : R^n \rightarrow R^N$ (or if such functional relationship does not exist). Details of feature transformation are usually established using some criterion which ensures maintaining critical data properties. It can be derived either from some general data characteristics (in unsupervised manner) or from the result of considered data analysis task (supervised feature extraction). One of the most widely used universal linear techniques of feature extraction is the principal components analysis (PCA). Conversely, multidimensional scaling (MDS) constitutes a typical representative of traditional nonlinear methods [6]. Studies on performance of routines belonging to both of above mentioned classes prove that even though nonlinear techniques possess more advanced mathematical background, they obtain often worse results in case of real-life datasets [12]. Apart from the performance the ability to create implicit mapping, which afterwards can be easily generalized to new data elements acquired dynamically, is also important in practical analytical tasks [10].

This paper introduces a new universal method of linear dimensionality reduction for use in exploratory data analysis. Dimensionality reduction is accomplished here by means of distance preserving linear transformation. The elements of the transformation matrix are to be determined using Fast Simulated Annealing. Additionally sample elements which as an effect of transformation significantly change their position could be eliminated or given lower weights. It can later serve in improvement of data analysis performance or sample length reduction.

The paper is organized as follows. Methodological preliminaries of the introduced method and its detailed description will be presented in the following Sections. As the performance of the technique under consideration was tested in clustering, classification and outlier detection procedures their short description will be given as well, followed by experimental results obtained in numerous testing trials. Finally some concluding remarks on the introduced method and planned further research will be given.

2 Methodological Preliminaries

2.1 Basic Exploratory Data Mining Tasks

First consider a problem of outlier detection. Such procedure is usually performed at the start of data exploration process to remove those elements from the sample which are found to be not representative. Usually it is performed by means of statistical approaches, e.g. using Grubbs test or Local Outlier Factor algorithm [3]. Measuring the performance of given procedure is difficult as usually it is not known in advance which element of the sample is atypical. However, if such knowledge is available, then the performance of the algorithm can be measured by:

$$I_{out} = \frac{c_o}{m} \tag{3}$$

where c_o is a number of correctly classified elements - either as an outlier or normal sample data point.

The task of cluster analysis is equivalent to such division of available data elements into subgroups (clusters) that elements belonging to each cluster are similar to each other and, at the same time, there is a significant dissimilarity between different clusters' elements. Numerous procedures have been developed to solve this problem. Among others K-means and DBSCAN algorithms can be named as popular ones [23]. If it is needed to compare different clustering solutions (or there exists a knowledge about cluster assignment) it is possible to use appropriate clustering indices e.g. Rand index:

$$I_{Rand} = \frac{a+b}{\binom{m}{2}} \tag{4}$$

with a and b being a number of data pairs which have been assigned to the same and different clusters in the both of analyzed solutions. If cluster number is

fixed, one can also try to form confusion matrix, align it properly to find cluster correspondence and calculate cluster preservation index I_{clust} [16].

Finally let us consider the task of classification, that is designating element $\tilde{x} \in R^n$ from the testing set to one of the fixed class with known set of representative patterns, similar to (1) (i.e. training set). Classification is often performed using instance-based learning methods e.g. k-nearest neighbor algorithm, along with more sophisticated statistical or computational intelligence procedures [19]. The efficiency of classification is evaluated by measuring its accuracy:

$$I_{class} = \frac{l}{m} 100\% \tag{5}$$

that is by a number of testing dataset elements l properly assigned to available classes, given as a ratio of overall dataset length. When precise division of the dataset into testing and training part is not explicitly given, one can use k-fold cross-validation, i.e. split available data into k sets and use one for evaluating purposes and the rest – for classifier learning. Whole process is usually repeated k times, although different variants of such validation can be found in the bibliography of the subject. Nevertheless in the case of cross-validation the average accuracy $\overline{I}_{class}$ is usually reported as a final result.

2.2 Fast Simulated Annealing

Simulated Annealing (SA) is a heuristic algorithm which can be used in various optimization problems. Its idea is based on metallurgic annealing process. The SA algorithm incorporates iterative local search with individual acceptance criterion. By means of this criterion current algorithm's solution is established, typically with usage of solution quality index from two consecutive iterations and variable decreasing in time parameter called temperature of annealing. Moreover it is assumed that non-zero probability of worse solution acceptance should be enforced. This probability ought to decrease in time and enable the algorithm to escape from pitfalls of local minima. In most generic variants of the SA algorithm Metropolis rule is used as above mentioned criterion [7].

The algorithm in particular application demands specifying few functional elements like generation of initial and neighbor solutions, initial temperature and scheme of its changes, solution quality index and finishing criterion. Some general remarks concerning these issues were made in [15]. It is worth to mention as well that SA can be effectively used in continuous optimization with specific variants of the algorithm developed precisely for that purpose, e.g. Boltzmann Annealing, Fast Simulated Annealing and Adaptive Simulated Annealing [5].

Fast Simulated Annealing (in short: FSA), used in dimensionality reduction algorithm described here, is a strategy which employs random moves obtained by using multidimensional Cauchy distributed random numbers [18]. Global convergence of the algorithm to the optimal solution as iteration number t approaches infinity, is maintained by using conservative logarithmic annealing temperature schedule.

3 Algorithm Description

3.1 Dimensionality Reduction

Concept of the dimensionality reduction technique from n to predetermined N-dimensional space is based on linear transformation (2), given in detail by:

$$\begin{bmatrix} y_{11} & y_{12} & ... & y_{1m} \\ y_{21} & y_{22} & ... & y_{2m} \\ \vdots & \vdots & & \vdots \\ y_{N1} & y_{N2} & ... & y_{Nm} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & ... & a_{Nn} \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & ... & x_{1m} \\ x_{21} & x_{22} & ... & x_{2m} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & ... & x_{nm} \end{bmatrix} . \quad (6)$$

Elements of transformation matrix A are found using Fast Simulated Annealing technique. Solution is represented as a vector:

$$z = [a_{11}, a_{12}, ..., a_{1n}, a_{21}, a_{22}, ..., a_{2n}, ..., a_{N1}, a_{N2}, ..., a_{Nn}]^T \in R^{nN} . \quad (7)$$

Solution quality index is given in the form of cost which is going to be minimized as a result of FSA algorithm. It can be represented in the form of:

$$g(z) = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \left(d_{ij} - \delta_{ij}(z)\right)^2 \quad (8)$$

or

$$g(z) = \frac{1}{\sum_{i=1}^{m-1} \sum_{j=i+1}^{m} d_{ij}} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \frac{\left(d_{ij} - \delta_{ij}(z)\right)^2}{d_{ij}} \quad (9)$$

where d_{ij} and δ_{ij} are distances (predominantly Euclidean) between sample points i and j in the initial and reduced feature space respectively. Both indices should enable achieving minimal difference of distances between sample elements in initial and reduced feature spaces, with additional emphasis put on small distances in the second cost function. Such formulations of solution quality indices are referred to as raw stress (8) and Sammon stress (9). Both were already used in nonlinear procedures of Multidimensional Scaling [4].

Initial solution is determined either randomly or by using feature selection algorithm presented in [13], with the second strategy being represented by two different variants. In general this alternative deterministic technique is based on the idea of feature space partition into clusters containing features which are similar to each other, with maximum information compression index being used as a similarity measure. Feature space clustering is performed using k-nearest neighbor algorithm, where k equal to $n - N$ should be assumed. As a result approximately N clusters are obtained. It is worth mentioning that a result of such initial solution's determination instead of being strictly fixed is customized to a real data structure. First approach of employing this feature selection algorithm to initial solution generation is based on N most representative features. The transformation matrix is formed in a way to retain them. It is achieved by using

1 and 0 properly as an indicative weights in the structure of A. Second strategy based on approach presented in [13] involves creating reduced feature set by linear combination of features included in each of "feature clusters". To implement it, the solution of feature selection is stored in auxiliary vector $v \in R^n$. Each element of v characterizes the number of cluster to which corresponding feature from the initial feature space was assigned. This vector is then transformed into transformation matrix A using following rule: $a_{ij} = 1$ if $v_j = i$ and $a_{ij} = 0$ otherwise.

Initial temperature of FSA is determined from preliminary set of pilot runs and it ensures approximate 0.7 probability of worse solution acceptance in the introductory phase of the algorithm. Annealing ends after fixed number of iterations and as its result matrix A minimizing solution quality indices (8) or (9) and transformed dataset Y are obtained. In the case of classification task the reduction is performed for training dataset and reduced evaluation set is synthesized using transformation matrix formed as a result of such procedure.

3.2 Weighting and Sample Length Reduction

Linear transformation of feature space in the form presented in the previous Subsection can seriously affect some data elements' relative position. Consequently the performance of data mining procedures in the reduced feature space can deteriorate significantly. As a countermeasure it is proposed to associate with each sample element a positive weight w_i normalized to ensure $\sum_{i=1}^{m} w_i = m$. Those weights are to be calculated using auxiliary parameters:

$$w_i^* = \frac{1}{\sum_{j=1, j\neq i}^{m} \left(d_{ij} - \delta_{ij}\right)^2} , \tag{10}$$

and performing normalization:

$$w_i = \frac{m w_i^*}{\sum_{i=1}^{m} w_i^*} \tag{11}$$

for $i = 1, ..., m$. Introduction of weights allows to take into account deformations in a relative data structure. Data elements with higher weights could then be treated as more adequate. Furthermore, one can use them as well to eliminate some data elements from the sample. It can be performed by removing instances with associated weights fulfilling following condition: $w_i < W$ where $W \in (0, +\infty)$ and then normalizing all weights (11). One can achieve in this way simultaneous dimensionality and sample length reduction with W serving as a data compression ratio.

4 Experimental Results

Proposed technique was verified for data exploration procedures based on four multidimensional example datasets taken from the UCI Machine Learning Repository [20].

Data dimensionality reduction routine was compared with PCA and unsupervised feature selection based on Evolutionary Algorithms [16] (first, best performing, variant of this algorithm was selected for this comparison). The latter was chosen for this study, because it employs Sammon stress as solution quality index. Heuristic procedure of Simulated Annealing as well as referenced techniques were executed in 10 independent trials (similarly to [16]). Each run was performed using 1000000 iterations as a stopping condition. Reduced feature space size N was selected according to [16], with the exception of $Vehicle$ dataset for which standard PCA-based intrinsic dimensionality estimation was employed.

Table 1. Dimensionality reduction for nearest-neighbor classification

	Glass		**WBC**	
	m=214, n=9, N=4		*m=683, n=9, N=4*	
	6 classes		*2 classes*	
	Accuracy [%]		**Accuracy [%]**	
	Average	**Std. dev.**	**Average**	**Std. dev.**
Initial FS	69.0	7.7	95.6	2.2
Raw, Linear combination	61.2	8.7	95.7	1.7
Raw, Feature selection	63.6	6.5	96.0	1.6
Raw, Random	62.1	7.7	96.0	1.4
Sammon, Linear combination	62.1	9.4	**96.2**	1.4
Sammon, Feature selection	**64.5**	4.4	95.9	1.5
Sammon, Random	61.4	9.9	96.0	1.2
EA-based [16]	64.8	4.4	95.1	0.8
PCA	57.6	9.9	96.3	1.8

	Wine		**Vehicle**	
	m=178, n=13, N=5		*m=846, n=18, N=5*	
	3 classes		*4 classes*	
	Accuracy [%]		**Accuracy [%]**	
	Average	**Std. dev.**	**Average**	**Std. dev.**
Initial FS	72.6	3.9	64.8	3.1
Raw, Linear combination	70.6	6.3	57.9	5.0
Raw, Feature selection	70.6	6.0	55.7	3.9
Raw, Random	**76.0**	5.1	**66.4**	3.9
Sammon, Linear combination	68.6	4.0	57.9	5.1
Sammon, Feature selection	70.9	6.0	56.2	6.4
Sammon, Random	75.4	6.1	64.9	3.5
EA-based [16]	72.8	1.0	60.8 [N=9]	1.5
PCA	70.9	8.4	46.9	5.5

Table 1 summarizes results obtained for classification performed using five-fold cross validation and the nearest-neighbor classifier. Reported values include classification accuracy in the initial feature space, six variants of the FSA-based

algorithm with different cost function and initial solution generation, as well as classification accuracy I_{class} obtained by referenced algorithms. It is important to stress that for EA-based technique reduced feature set is synthesized using both training and testing sets. In the case of the algorithm being described here out-of-sample extension is used. It allows to transform testing set using transformation matrix synthesized for the training set. Nevertheless, results obtained are comparable to the ones achieved by referenced techniques. It can be noticed however, that it is difficult to select in advance which variant of the algorithm will reach highest-performance.

To test the possibility of sample size reduction classifier based on the kernel density estimators (KDE) was used [9], as its structure is very easy to modify to include weights [8]. In the considered case weighting scheme alone does not have a positive effect on classificator's performance. It can be used though to eliminate elements which as an effect of dimensionality reduction have a negative impact on data mining process. Elimination of elements with weights lower than 0.5 leads in some cases to the improvement of classification accuracy (see Table 2). It is predominantly observed when the sample size is too small to perform KDE-based classification reliably in the initial, high dimensional feature space. It confirms well-known fact that kernel density estimation is seriously affected by the curse of dimensionality [17].

Table 2. Dimensionality and sample size reduction for KDE-based classification

	Glass		**WBC**	
	Accuracy [%]		**Accuracy** [%]	
	Average	**Std. dev.**	**Average**	**Std. dev.**
Initial FS	60.5	7.6	95.0	2.0
PCA	52.6	8.9	93.0	2.8
Reduced	63.8	10.5	92.4	2.4
Reduced + Sample size reduction	67.6	7.7	95.5	2.1
(Sample elements removed [%])	(8.7)	(1.8)	(10.3)	(2.1)

Finally cluster analysis and outlier detection experiments were performed. The preservation of cluster structure was indicated by cluster preservation index I_{clust}. In the case of outlier classification, its preservation was measured by (3). Values of both indices were reported for selected datasets in Table 3). Again, the technique under consideration achieved high accuracy of datasets structure preservation, comparable (or even better) to the one achieved by EA-based technique.

5 Conclusion

This paper introduces new dimensionality and sample reduction technique designed for tasks of exploratory data mining. Introductory studies on method's

Table 3. Dimensionality reduction with cluster and outlier preservation

		Glass		WBC	
		Preserved [%]		Preserved [%]	
		Average	Std. dev.	Average	Std. dev.
Cluster preservation	**Reduced**	71.2	9.7	98.1	0.4
	EA-based [16]	69.3	4.9	94.7	2.3
Outlier Preservation	**Reduced**	95.4	0.9	88.1	1.1

performance prove that it offers promising solution quality in reference to the state-of-art principal components analysis procedure and similar heuristic based feature selection strategy. One should note however it is not specifically suited and designed for very high dimensional problems with huge sample sizes, as the optimization phase of FSA has significant computational complexity. Its leads to exponential growth of computation time with increasing m. Nevertheless the method under consideration can be still used for data visualization and formulation of convenient data transformation, which can be later used in the data acquisition process. What is more, the possibility of practical implementation is significantly increased by employing simultaneous sample size.

Further studies on the subject will concern various improvements in Fast Simulated Annealing scheme (e.g. statistic termination criterion of the algorithm). As Simulated Annealing can be effectively parallelized (refer to [1] and [11]) this area of research is going to be explored as well. It will allow the algorithm application for larger datasets. In addition prospective research will concern further improvements in sample size reduction scheme and its usage in various standard data mining algorithms.

References

1. Alba E. (Ed.): Parallel Metaheuristics. John Wiley & Sons, New York (2005)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (1999)
3. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computing Surveys, 41, 15:1–15:58 (2009)
4. Cox, T.F., Cox, M.A.A.: Multidimensional Scaling. Chapman & Hall, Boca Raton (2000)
5. Ingber, L.: Adaptive simulated annealing (ASA): Lessons learned. Control and Cybernetics, 25/1, 33–54 (1996)
6. Jain, A.K, Duin, R.P.W., Mao, J.: Statistical Pattern Recognition: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22/1, 4–37 (2000)
7. Kirkpatrick, S., Gelatt, C.D., Vecchi, M. : Optimization by Simulated Annealing. Science, 220, 671–680 (1983)
8. Kowalski, P.A., Kulczycki, P.: Data sample reduction for classification of interval information using neural network sensitivity analysis. In: Dicheva, D., Dochev, D. (eds.) Artificial Intelligence: Methodology, Systems, and Applications. LNCS, vol. 6304/2010, pp. 271–272. Springer, Heidelberg (2010)

9. Kulczycki, P.: Kernel Estimators in Industrial Applications. In: Prasad, B. (ed.) Soft Computing Applications in Industry, pp. 69–91. Springer, Heidelberg (2008)
10. Lee, J.L., Verleysen, M.: Nonlinear Dimensionality Reduction. Springer, Heidelberg (2007)
11. Łukasik, S.: Parallel Computing of Kernel Density Estimates with MPI. In: Shi, Y., Albada, G.D.v., Dongarra, J., Sloot, P.M.A. (eds.) Computational Science. LNCS, vol. 4489, pp. 726–734. Springer, Heidelberg (2007)
12. Maaten, L.J.P.v.: Feature Extraction from Visual Data. PhD Thesis, Tilburg University, June 23rd 2009.
13. Mitra, P., Murthy, C.A., Pal, S.K.: Unsupervised Feature Selection using Feature Similarity. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24/4, 301–312 (2002)
14. Pal, S.K., Mitra, P.: Pattern Recognition Algorithms for Data Mining. Chapman and Hall, London (2004)
15. Sait, S.M., Youssef, H.: Iterative computer algorithms with applications in engineering. IEEE Computer Society, Los Alamitos (1999)
16. Saxena, A., Pal, N.R., Vora, M.: Evolutionary methods for unsupervised feature selection using Sammons stress function. Fuzzy Information and Engineering, 2/3, 229–247 (2010)
17. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman and Hall, London (1986)
18. Szu, H., Hartley, R.: Fast simulated annealing. Physics Letters A, 122/3-4, 157-162 (1987)
19. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson Addison-Wesley, Boston (2006)
20. UCI Machine Learning Repository, `http://archive.ics.uci.edu/ml/`
21. Verleysen M., François D.: The Curse of Dimensionality in Data Mining and Time Series Prediction. In: Cabestany, J., Prieto, A., Sandoval, F. (eds.) Computational Intelligence and Bioinspired Systems. LNCS, vol. 3512, pp. 758–770. Springer, Heidelberg (2005)
22. Wand, M.P., Jones, M.C.: Kernel Smoothing. Chapman and Hall, London (1995)
23. Xu, R., Wunsch, D.C.: Clustering. Wiley, New York (2009)