Dimensionality Reduction for Feature and Pattern Selection in Classification Problems

Zacharias Voulgaris, and George D. Magoulas
Birkbeck College, University of London
{zacharias, gmagoulas}@dcs.bbk.ac.uk

***Abstract*—Reducing the dimensionality of a dataset is an important and often challenging task. This can be done by either reducing the number of features, a task called feature selection, or by reducing the number of patterns, called data reduction. In this paper we propose methods that employ a novel concept called Discernibility for achieving these two tasks separately, with the aim to solve classification problems. The experimental results verify our claim that the proposed methods are a viable alternative for dimensionality reduction, for various datasets and a variety of classifiers.**

***Index Terms*— Dimensionality Reduction, Feature Selection, Data Reduction, Pattern Recognition, Discernibility**

I. Introduction

THE past five decades have been revolutionary in the way data acquisition has developed. This facilitated the accumulation of data, in both resolution (high number of features) and size (large number of patterns), rendering today's databases on the order of terabytes [1]. This gave rise to a constantly growing need for data reduction as well as the adoption of an eclectic attitude towards the attributes of certain datasets (particularly those containing microarray data). In this article we explore how we can achieve both data reduction and feature selection using the novel concept of Discernibility that we have introduced.

According to [2], there are two distinct approaches to Data Reduction (DR): the general purpose DR and the task-specific DR. Since in this research we are dealing with classification, we will focus on the latter approach (which as one would expect, yields better results for the classification task). Contrary to one would expect, the reduced datasets often exhibit better classification performance (accuracy rate). This is probably due to the reduction in complexity, which in the original dataset takes the form of redundant patterns and useless outliers.

The same reduction in complexity (for datasets with high number of features) is achieved via feature selection techniques. Particularly in biomedical sets involving gene expression data, the dimensionality involved is ultra-high (thousands of attributes), resulting to a redundancy in features that often has a negative impact on the classification process. This type of datasets, also referred to as microarray data, is also characterised by a limited number of patterns, which makes generalisation more difficult, rendering the classification process a rather challenging task.

Regarding the reduction of the feature set, a lot of research has taken place using a variety of methods. The methods used in the literature were applied mostly for classification with SVMs [3, 4, 5], although other approaches were also considered [6 - 11]. As regards the methods themselves, feature selection has been accomplished using Genetic Algorithms [3, 6], Locally Linear Embedding (LLE) [4], Recursive Salient Analysis (RSA) [5], the selection of a support set [7], a combination of PCA and the UKW clustering algorithm [8], t-tests [6, 9], Regression Splines (MARS) [9], Classification and Regression Trees (CART) [9], Random Forests [9], Linear Genetic Programs (LGP) [9], Neural Networks as a similarity measure [10], and clustering analysis [11]. All of the methods of the literature appear to work well, for one or the other dataset, for SVMs or the other classifiers used in each particular case. However, the underlying problem that all of them have is that there is no particular stopping criterion for the feature selection. The proposed approach in this paper attempts to overcome this drawback by introducing a threshold parameter for the feature selection process.

As regards the reduction of the patterns in a dataset, a lot of different techniques have been developed over the past few years. These include Statistical methods (such as Chi-square [2], Similarity [12] and Aggregation [1]), Geometrical methods [13], Clustering (both classic [14] and fuzzy [15]), Autoregressive (AR) techniques [16], PCA [15, 12], Singular Value Decomposition (which is one of the most popular methods) [17, 15], Wavelets (DWT) [16], Independent Component Analysis [12], Fourier Decomposition (DFT [16] and a variation of FA [18]), SOM [19] and an alternative data reduction approach (ALEV) [19]. The method proposed in this paper comes from a different perspective to offer a promising alternative as it is quite fast and simple to implement.

II. Discernibility Concept and Index of Discernibility

A. Concept Description and Calculation

Discernibility is a measure we developed for evaluating how easily distinguishable the classes of a dataset are. It works on two levels: the pattern one and the dataset one. In the latter case it takes the form of an index, which we call the Index of Discernibility.

For calculating Discernibility we make use of (hyper)spheres; this attempt assumes a fixed radius around each element of the dataset, which corresponds to the average distance between this and the rest of the elements of that class. Note that the radius depends on the class structure, so elements belonging to different classes may have different radii. Once the radius of an element is established, elements of the same class as the examined element that belong to its (hyper)sphere are identified and counted. The discernibility of that element is calculated by dividing the number of these elements by the number of total elements in the (hyper)sphere (see Figure 1).

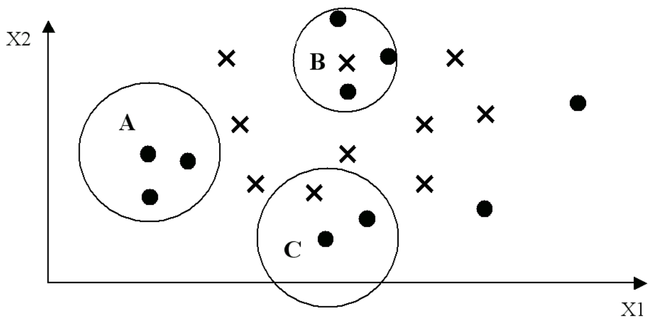

Fig. 1. Graphical representation of the Discernibility concept for three elements of a simple dataset with only two features, X1 and X2. In this example, the discernibility of the element in the centre of circle A is ID1 = 2 / 2 = 1, that of the element in the centre of circle C is ID2 = 1 / 2 = 0.5, while that of the element in the centre of circle B is ID3 = 0 / 3 = 0.

This concept should not be confused with the discernibility tables, or the indiscernibility concept, used in Rough Sets ([20, 21]). The latter are completely different both in function and in field of application.

After having calculated the discernibilities of all the elements of the dataset we can calculate the Index of Discernibility (ID). This is defined as the number of elements having discernibility higher than the threshold $th = 0.5$ divided by the total number of elements. Alike the Discernibility of single pattern, the Index of Discernibility always takes values between 0 and 1, and the higher it is, the more distinguishable the classes of the dataset are.

B. Applications

The Index of Discernibility has a variety of applications. Its primary use is as an evaluation measure of the "difficulty" of a dataset (the lower it is, the less distinguishable the classes are, and therefore the more difficult it is to classify accurately). In addition, it can be applied for evaluating individual features of it. This allows it to be used as a part for feature selection techniques [22], one of which we will describe in section III. Another application is its use for providing structural information on the dataset, which can be incorporated in various classifiers (in [23] we describe how this can be done for upgrading the kNN classifier). Moreover, by employing the individual discernibilities of the various patterns of each class of the dataset, we can perform data reduction (section IV). Furthermore, the Index of Discernibility can be used in combination with other metrics for the development of a Reliability measure.

III. Feature Selection Using the ID

The Index of Discernibility can be a useful heuristic for the development of feature selection methods. This is accomplished by either applying it to individual features of the dataset and filtering out the ones that exhibit a value lower than a given threshold, or to groups of features and finding a combination of them that maximises its value [22]. Several such methods have been created by the authors, one of which is presented below.

The feature selection method called fs3 is a simple technique which firstly evaluates the *na* features of a dataset (P, T) and then selects the ones that have an ID greater than a given threshold (*th*). However, as this is an absolute parameter, the number of features at the new feature set greatly depends on the dataset itself. Empirically, a value ranging from 0.7 to 0.85 is a good choice for the *th* parameter. Values less than 0.7 would result in an excessive number of features while values greater than 0.85 would limit the selection of the features to very few features or none in some cases. Of course, the selection of the *th* parameter is problem-dependent. To alleviate this problem one can apply the following rule of thumb: choose *th* to be equal to the mean or the median of the Discernibilities of the features of the dataset. This method's outputs are the reduced feature set (P2), and a list of the indices of these features (NFS). Afterwards, using the NFS array, we can perform the same feature selection (in practically no time) for the testing set. The operation of the method is demonstrated in the flowchart of Fig. 2.

Since the computations involved in this method are very limited and not time-consuming (because merely single features are evaluated), this method is very fast. This constitutes its main advantage. Also, if the threshold parameter is set to a reasonable value, the reduced feature set yielded is very robust and exhibits good classification performance. Furthermore, the fact that it has only one parameter (*th*) makes it relatively easy to fine-tune to a specific problem and the low sensitivity of this parameter allows the resulting feature set (P2) to be more stable in terms of the quality of the selected features.

Other approaches for feature selection based on Discernibility employ an N-dimensional evaluation of the feature sets, which takes into account the correlation of the selected features. However, such approaches can be very computationally expensive, especially in cases where there is a large number of features. Also, the improvement is small and often insignificant, rendering a not so attractive alternative.

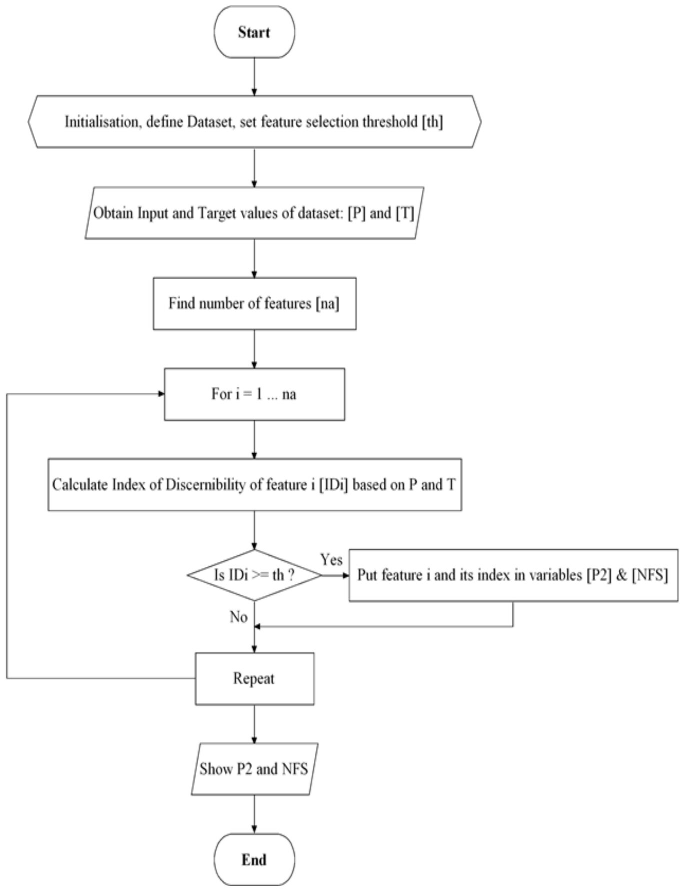

Fig. 2. Flow chart of the ID-based feature selection method (fs3).

IV. Discernibility-Based Data Reduction

Data reduction can be accomplished using the discernibility concept by assessing how "easily" distinguishable the various patterns of a dataset are, and then removing the ones having the highest discernibility, taking into account their distances to the other patterns of their classes. Contrary to the feature reduction techniques, in this application of Discernibility it is unwise to eliminate patterns below a given threshold, since there is a strong inter-dependency among all of them, as regards their discernibility status. Therefore, the patterns to be removed have to be selected carefully; otherwise there is a risk of distorting the class structure of the dataset (resulting to a drop in the accuracy rate of the classifiers). An analytical description of the operation of this method can be seen in the flowchart of Fig. 3.

From this flow chart, one can observe that there are two factors that are taken into account for the data reduction: the patterns' discernibilities Z and their distances DD to the other patterns, which are to be discarded. Therefore, a removed pattern has to be easy to distinguish (high discernibility) and be relatively far away (distance) from other patterns that will be removed. This allows the reduction of the patterns to be "smooth" and balanced.

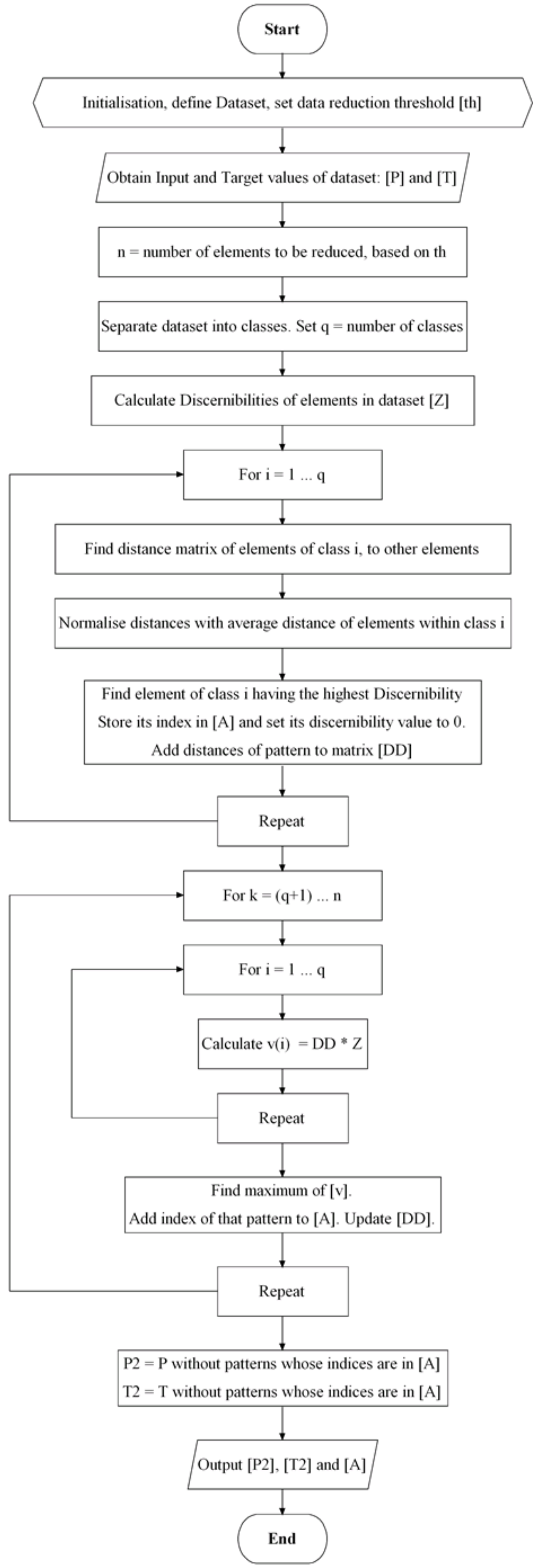

Fig. 3. Flow chart of the ID-based data reduction method.

V. Experiments, Results and Discussion

A. Feature Selection Experiments

Experiment Setup

We run two distinct sets of experiments. The first one involves testing the proposed feature selection schemes in terms of their dimensionality reduction capabilities, whilst the second one investigates how five classification algorithms perform based on the reduced feature sets.

The datasets used in our experiments were: the Diffuse Large B-Cell Lymphoma (DLBCL), the Colon Tumor (colon), the Central Nervous System (CNS) and the ALL-AML Leukaemia (leukaemia) problems. All of them were taken from [24] and are representative of a class of bioinformatics applications that employ microarray data. Also, these datasets are often used in the literature for feature selection [3-11]. Their characteristics are shown in Table 1 (this table also presents the dimensionality of the reduced feature sets that will be discussed in the next subsection). The large number of feature in the original datasets stems from the fact that these data involve gene expression profiles, which are by nature of this size. Datasets with dimensionality between 2000 to 4000 features are actually already reduced. Lastly, all of the features are numeric and continuous.

The classifiers used in our experiments are k Nearest Neighbour (kNN) [25], a variations of kNN (V-kNN) [23], Linear Discriminant Analysis (LDA) [25], the Gravity Model Classifier [26], and the Fuzzy kNN classifier [27] (we used the MATLAB implementation of the Fuzzy kNN that was produced by Emre Akbas in December 2006).

Regarding the experiments, they comprised of 50 runs, using the leave-one-out method. We used this method of validation due to the small number of patterns in each dataset (the largest one had 72 patterns). In addition, this method of validation is the one more widely used in the literature for datasets of this type.

TABLE I
Datasets characteristics and sizes of reduced feature sets

Dataset	No. of Patterns	Original no. of Features	Features after fs3
DLBCL	47	4026	45
Colon	62	2000	11
CNS	60	7129	19
Leukaemia	72	7129	112

Results and Discussion

Classification results with the original feature sets are presented in Table II. Note that for the LDA classifier, there are no results for the CNS and Leukaemia datasets, as our system (1.8 GHz processor with 2GB RAM) could not complete the relevant experiments because of the method's high computational demands. As regards the overhead of the feature selection process, the CPU time ranged from 6 to 23 seconds (depending on the dataset). The classification results for the reduced feature set for the various classifiers can be seen in Table III.

TABLE II
Feature Selection Experimental Results (original feature set)

Classifier Dataset, Ev. Measure		kNN	V-kNN	Fuzzy kNN	LDA	GMC
DLBCL	A.R.	55.32	48.94	55.32	63.83	51.06
	Time	1.01	16.88	0.96	5381.40	1.74
Colon	A.R.	83.87	79.03	85.48	51.61	64.52
	Time	0.89	15.82	0.72	882.40	1.53
CNS	A.R.	66.67	63.33	66.67	–	65.00
	Time	3.05	76.52	3.19	–	5.71
Leukaemia	A.R.	93.06	90.28	93.06	–	65.28
	Time	4.68	144.45	4.74	–	8.35

Note: A.R. = Accuracy Rate (%), Time = CPU time in seconds

TABLE III
Feature Selection Experimental Results (reduced feature set)

Classifier Dataset, Eval. Measure		kNN	V-kNN	Fuzzy kNN	LDA	GMC
DLBCL	A.R.	72.34	68.09	87.32	42.55	72.34
	Time	0.06	1.29	0.09	0.20	0.13
Colon	A.R.	85.48	90.32	90.32	77.42	83.87
	Time	0.05	2.29	0.06	0.19	0.25
CNS	A.R.	71.67	68.33	70.00	56.67	66.67
	Time	0.12	2.22	0.14	0.11	0.22
Leukaemia	A.R.	94.44	94.44	94.44	48.61	65.28
	Time	0.17	3.55	0.19	0.26	0.34

Note: A.R. = Accuracy Rate (%), Time = CPU time in seconds

As it can be observed from Tables II and III the average CPU time of the classification is reduced. This is very important, considering that the feature selection itself takes a not negligible amount of time. Another important point is that the average accuracy rate is significantly increased in most of the datasets, for most of the classifiers. This is something expected, since the original feature set contains a large number of useless features which not only do not aid the classification, but for many of the classifiers, they make it more difficult. So, by eliminating these features we end up with a relatively "easier" dataset to classify.

B. Data Reduction Experiments

Experiment Setup

For this series of experiments, we tested the data reduction method with 3 fairly large datasets and applied 4 different threshold values (1/10, 1/4, 1/3 and 1/2). The experiments comprised of 30 rounds of 10-fold cross validation. The datasets used were *clouds*, which contains 5000, 2-dimensional patterns [28], and the *pendigits* (10992, 16-dimensional patterns) and *magic* (19020, 11-dimensional) [29]. We experimented with the classifiers kNN, LDA, C4.5, Fuzzy kNN [27] and the Gravity Model Classifier (GMC) [26]. For each classifier we measured the Accuracy Rate before and after the data reduction, as well as the CPU time taken for each classification. The data reduction overhead was measured separately.

Results and Discussion

The results of the classifications for each one of the three datasets are shown in Tables IV–VI. The data reduction overhead ranged from 5 to 60 sec., depending on the dataset. It is generally higher for large and / or complex datasets (e.g. the *magic* dataset).

TABLE IV
RESULTS FOR THE *CLOUDS* DATASET

Classifier Reduction	Ev.Measure	kNN	C4.5	Fuzzy kNN	LDA	GMC
None	A.R.	88.19	65.40	86.02	50.00	86.73
	Time	0.59	5.10	0.71	<.01	1.64
1/10	A.R.	88.21	64.55	86.40	50.00	86.70
	Time	0.54	4.40	0.64	<.01	1.48
1/4	A.R.	88.24	67.77	86.87	50.00	86.13
	Time	0.47	5.17	0.53	<.01	1.24
1/3	A.R.	88.13	63.95	87.18	50.00	85.73
	Time	0.42	2.90	0.48	<.01	1.10
1/2	A.R.	87.93	63.97	87.41	50.00	83.40
	Time	0.34	1.96	0.37	<.01	0.84

Note: A.R. = Accuracy Rate (%), Time = CPU time in seconds

TABLE V
RESULTS FOR THE *PENDIGITS* DATASET

Classifier Reduction	Ev.Measure	kNN	C4.5	Fuzzy kNN	LDA	GMC
None	A.R.	97.60	79.47	88.02	82.19	81.13
	Time	23.88	394.31	24.76	0.09	57.98
1/10	A.R.	97.60	34.19	97.74	82.59	85.42
	Time	20.62	287.68	21.09	0.04	51.19
1/4	A.R.	97.66	46.66	97.66	82.08	87.94
	Time	16.43	206.69	16.49	0.03	40.98
1/3	A.R.	97.68	39.88	97.63	81.79	86.39
	Time	13.53	169.09	13.43	0.03	31.69
1/2	A.R.	97.40	37.34	97.37	81.36	73.44
	Time	7.47	102.07	7.99	0.03	22.52

Note: A.R. = Accuracy Rate (%), Time = CPU time in seconds

TABLE VI
RESULTS FOR THE *MAGIC* DATASET

Classifier Reduction	Ev.Measure	kNN	C4.5	Fuzzy kNN	LDA	GMC
None	A.R.	83.04	81.17	82.91	77.35	67.79
	Time	56.72	127.90	59.95	0.04	151.39
1/10	A.R.	83.29	81.05	83.09	78.22	67.79
	Time	48.88	108.02	52.56	0.04	130.26
1/4	A.R.	82.93	79.41	82.67	78.48	68.15
	Time	35.43	84.00	38.81	0.03	92.00
1/3	A.R.	82.64	79.17	82.36	78.62	68.69
	Time	28.58	70.41	30.77	0.02	83.06
1/2	A.R.	81.57	77.69	81.52	78.08	69.76
	Time	19.19	47.20	21.45	0.01	58.39

Note: A.R. = Accuracy Rate (%), Time = CPU time in seconds

From Tables IV, V and VI, it can be seen that in the majority of cases, both the Accuracy Rate and the CPU time are enhanced, for all of the classifiers. Also, in some cases (e.g. for the GMC classifier in the dataset *Magic*), the CPU time is decreased so much that even together with the data reduction overhead (which in this case is about 1 minute) it is still faster than with the original dataset for all of the reduction thresholds.

The reduced dataset is often very similar in appearance (i.e. it maintains the general structure of the patterns in the feature space), as it can be seen in Figure 4, for the dataset *Clouds*. This is because the patterns to be excluded are chosen in such a way that they are far apart from each other. Otherwise, if the Discernibility of each pattern were to be used as the sole decision rule (instead of the product of Discernibility and distance), the reduced dataset would not be as robust.

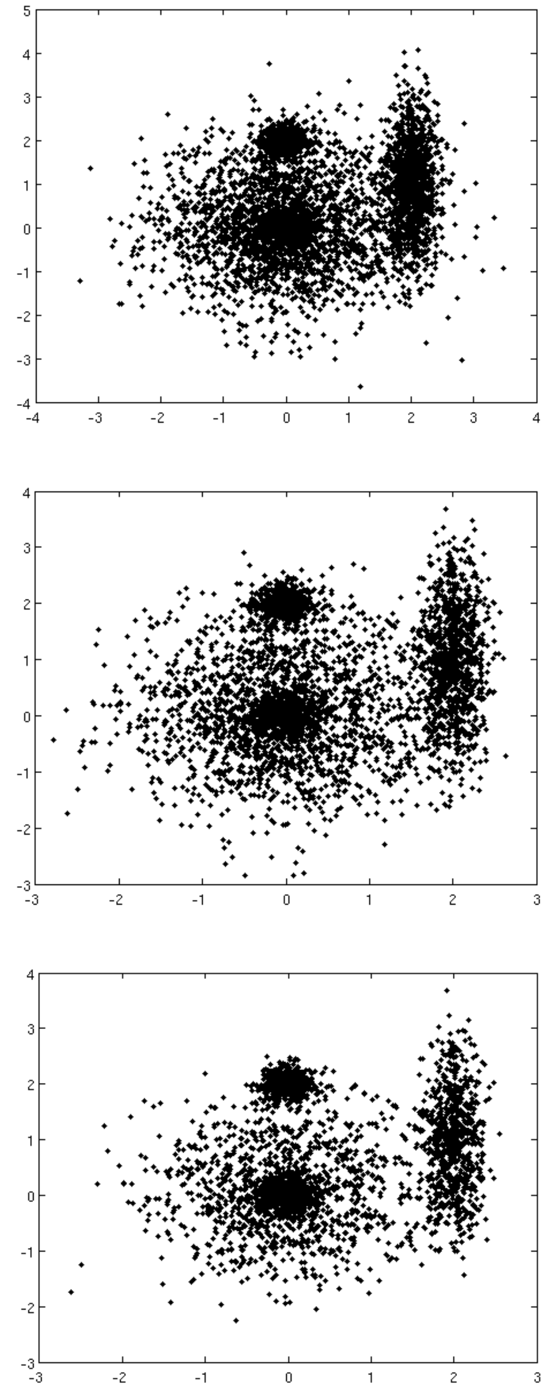

Fig. 4. *Clouds* dataset originally (top), with 1/10 reduction (middle) and with 1/3 reduction (bottom). The geometry of the dataset is generally preserved.

VI. Conclusion

An independent heuristic measure has been discussed for evaluating how distinguishable the classes of a dataset are. Among its various applications are methods for dimensionality reduction, both in terms of features and in term of patterns. The experiments we carried out illustrate that the methods developed by the authors perform well and that the Discernibility concept can be a robust tool for this type of work. Future work will include further research into the fine-tuning of the methods introduced as well as other applications of the Discernibility concept and its index.

References

[1] J. Skyt, C. S. Jensen and T. B. Pedersen, "Specification-based data reduction in dimensional data warehouses", in *Information Systems*, vol. 33, Issue 1, 2008, pp. 36-63.

[2] X. B. Li and V. S. Jacob, "Adaptive data reduction for large-scale transaction data", in *European Journal of Operational Research*, vol. 188, Issue 3, 2008, pp. 910-924.

[3] H. Frohlich and O. Chapelle, "Feature selection for support vector machines by means of genetic algorithms"., in *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*, 2003, pp. 142-148.

[4] S. Chao and C. Lihui, "Feature dimension reduction for microarray data analysis using locally linear embedding", in *Proceedings of 3rd Asia-Pacific Bioinformatics Conference,* 2005, pp. 211-217.

[5] L. Cao, C. K. Seng, Q. Gu and H. P. Lee, "Saliency analysis of support vector machines for gene selection in tissue classification", in *Neural Computing & Applications*, vol. 11, 2003, pp. 244-249.

[6] M. Lecocke and K. Hess, "An empirical study of univariate and GA-based feature selection in binary classification with microarray data", in *UT MD Anderson Cancer Center Department of Biostatistics Working Paper Series*, working paper 5, Available online at www.bepress.com/mdandersonbiostat/paper5 (last accessed February 2008), 2005.

[7] G. Alexe, G. Bhanot and B. Venkataraghavan, "A robust meta-classification strategy for cancer diagnosis from gene expression data", in Proc. IEEE Computational System Bioinformatics Conference, 2005, pp. 322-325.

[8] D. K. Tasoulis, V. P. Plagianakos and M. N. Vrahatis, "Unsupervised clustering in mRNA expression profiles", in *Computers in Biology and Medicine,* vol. 36, 2006, pp. 1126-1142.

[9] S. Mukkamala, Q. Liu, R. Veeraghattam and A. H. Sung, "Computational intelligent techniques for tumor classification (using microarray gene expression data)", in International Journal of Lateral Computing, vol. 2, no. 1, 2005, pp. 38-45.

[10] T. Sawa, L. Ohno-Machado, "A neural network-based similarity index for clustering DNA microarray data", in Computers in Biology and Medicine, vol. 33, 2003, pp. 1-15.

[11] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays", in Proc. Natl. Acad. Sci. USA, vol. 96, 1999, pp. 6745-6750.

[12] T. Thireou, J. L. R. Guivernau, V. Atlamazoglou, M. J. Ledesma, S. Pavlopoulos, A. Santos and G. Kontaxakis, "Evaluation of data reduction methods for dynamic PET series based on Monte Carlo techniques and the NCAT phantom", in *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 569, Issue 2, 2006, pp. 389-393.

[13] H. Song and H.Y. Feng, "A global clustering approach to point cloud simplification with a specified data reduction ratio", in *Computer-Aided Design*, In Press, Corrected Proof, Available online at: www.sciencedirect.com (last accessed March 2008), 2007.

[14] O. Okun and H. Priisalu, "Unsupervised data reduction", in *Signal Processing*, vol. 87, Issue 9, 2007, pp. 2260-2267.

[15] S. Russell and V. Yoon, "Applications of wavelet data reduction in a recommender system", in *Expert Systems with Applications*, vol. 34, Issue 4, 2008, pp. 2316-2325.

[16] K. Polat and S. Güneş, "A novel data reduction method: Distance based data reduction and its application to classification of epileptiform EEG signals", in *Applied Mathematics and Computation*, In Press, Corrected Proof, Available online at: www.sciencedirect.com, (last accessed March 2008), 2007.

[17] S. Vanlanduit, B. Cauberghe, P. Guillaume, P. Verboven and E. Parloo, "Reduction of large frequency response function data sets using a robust singular value decomposition", in *Computers & Structures*, vol. 84, Issue 12, 2006, pp. 808-822.

[18] J. Vanherzeele, P. Guillaume, S. Vanlanduit and P. Verboven, "Data reduction using a generalized regressive discrete Fourier series", in Journal of Sound and Vibration, vol. 298, Issues 1-2, 22 2006, pp. 1-11.

[19] W. Toplak, "A comparison of SOM and ALEV for data reduction purposes in transport telematics", in Proc. of 26th IASTED International Conference on Artificial Intelligence and Applications, CD Proceedings ISBN: 978-0-88986-710-9, 2008, pp. 181-186.

[20] S. Tan, Y. Wang and X. Cheng, "Text feature ranking based on rough-set theory", in IEEE/WIC/ACM International Conference on Web Intelligence, 2007.

[21] J. Komorowski, Z. Pawlak, L. Polkowski, A. Skowron, "Rough sets: a tutorial", in Rough-Fuzzy Hybridization: A New Method for Decision Making, Springer-Verlag, Singapore, 1998, Available online at http://citeseer.ist.psu.edu/komorowski98rough.html (last accessed February 2008).

[22] Z. Voulgaris and G. D. Magoulas, "A discernibility-based approach to feature selection for microarray data",in Proc. of IEEE International Conference of Intelligent Systems, Varna, Bulgaria, Sept. 2008, forthcoming.

[23] Z. Voulgaris and G. D. Magoulas, "Extensions of the k nearest neighbour methods for classification problems", in Proc. of 26th IASTED International Conference on Artificial Intelligence and Applications, CD Proceedings ISBN: 978-0-88986-710-9, 2008, pp. 23-28.

[24] Kent Ridge Bio-medical Data Set Repository, Available online at http://sdmc.lit.org.sg/GEDatasets/Datasets.html (last accessed January 2008).

[25] R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification (2nd ed.), John Wiley and Sons, 2001.

[26] D. Ruta and B. Gabrys, "Physical field models for pattern classification", in Soft Computing Journal, vol. 8, no. 2, 2003, pp. 126-141.

[27] J. M. Keller, M. R. Gray, and J. A. Givens, Jr., "A fuzzy k-nearest neighbor algorithm", in IEEE Transactions on Systems, Man, and Cybernetics, vol. 15, no. 4, 1985, pp. 580-585.

[28] ELENA Project Artificial Databases – clouds dataset, available on-line at http://www.dice.ucl.ac.be/neural-nets/Research/Projects/ELENA/databases/ARTIFICIAL/clouds/ (last accessed Mar 2008).

[29] UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/ (last accessed Mar 2008).