Proc. of the Intl. Conf. on Advances In Information Processing And Communication Technology - IPCT 2014.
Copyright © Institute of Research Engineers and Doctors. All rights reserved.
ISBN: 978-1-63248-021-7 doi: 10.15224/ 978-1-63248-021-7-70

Induction of LDA Oblique Decision Rules

Marcin Michalak

***Abstract*—This paper presents a new algorithm of decision rules with oblique conditions induction. It bases on the Fisher's Linear Discriminant Analysis as a tool of finding an initial classes separation. This technique has a good ability of oblique dependencies generalisation what reduces the number of decision rules and their complexities.**

***Keywords*—classification, machine learning, Linear Discriminant Analysis, decision rules, oblique decision rules**

I. Introduction

Machine learning is the wide branch of the computer science that contents algorithms and methods of building models from the data. The nature of the data and the goal of the analysis usually divide machine learning into two main groups: supervised and unsupervised learning.

In the case of the unsupervised learning, data examples do not contain any decision attribute. In the case of the supervised learning data examples contain two types of attributes: one (or more) is considered as the dependent from the remaining ones and is called the decision attribute or just the class. Then it is claimed that there exist some dependencies between conditional attributes and the class. In other words, supervised algorithms try to find (or describe) existing dependencies in the data. The name of the supervised learning comes from the fact, that the built model can be evaluated on the data which class labels are known.

Supervised methods are also divided into two groups due to the nature of the dependent attribute: if the decision attribute takes values from the finite set (set of discrete values) it is commonly named as the classification and the regression otherwise.

There are many of algorithms of classification. Just to mention the most important of them: artificial neural networks [16] , Support Vector Machines [4], decision trees [15], linear discriminant analysis [8]. As the most popular methods of regression apart from also applicable neural networks and Support Vector Machines, also splines [3], kernel estimators [21][27], radial basis functions [5], additive models [9], projection pursuit regression [9] should be mentioned.

All algorithms of the supervised and unsupervised learning can be also divided due to the criterion of the model being transparent (or just interpretable) for the user or in other words – the transparency of the decision making process.

Marcin Michalak

Institute of Informatics, Silesian University of Technology
ul. Akademicka 16
44-100 Gliwice, Poland

There are models of the classification (or regression) which only say "how" the model behaves and models, which say "how" and "why" the particular decision has been made. The most popular algorithms that work like "black boxes" (none explanation, just the answer) for the both – classification and regression tasks – are neural networks or Support Vector Machines.

When the interpretability of the classification model is important the induction of decision rules from the data examples (example based learning) becomes the most usable technique. Decision trees and decision rules can give quite good results in the most of cases but they have the strong limitation: the most of algorithms generate decision rules that are hyperrectangles which edges are parallel to axes of the coordinates system. This means that for two classes that are linearly separable with the hyperplane that is non-parallel to any coordinates system axis the generated set of rules will contain a lot of small rules, covering only the small part of the space and not describing any significant dependencies.

The main goal of the requested application is the development of the new algorithm of induction rules with oblique condition directly from the data, also with the usage of the Support Vector Machine and the new local measures of local classification error.

The paper is organised as follows: next section shows the state of art algorithms of induction of rules with oblique conditions and the authors' previous approaches in this area, then the description of decision rules, oblique decision rules and Linear Discriminant Analysis backgrounds are presented. Afterwards, the new idea of oblique decision rule induction is introduced, followed by the results of experiments and a short discussion. The paper ends with some final remarks and description of further works.

II. Related and Previous Works

The rule induction is a problem widely encountered in the literature. According to the goal of the analysis there are two groups of rule induction algorithms. First of them generates association rules and second of them decision rules. Association rules describe dependencies between attributes that do not give any decision. Methods of association rules induction can be also applied for the data with the decision attribute. Decision rules describe dependencies between conditional attributes and the class. In this project only the decision rules belong to the scope of interest.

The most popular approach of the rule induction is the "from coverage" strategy [6][7]. For every class rules are generated as long as all objects from this class are covered by at least one decision rule. It is also called the "separate and conquer" strategy [11] as two alternating steps can be pointed: learn the rule that covers (describes) the part of the given training examples and remove them from the training set (the separate step) and recursively learn another rule that covers some of the remaining examples (the conquer step).

The common feature of all generated decision rules is their shape in the data features space: they are hyperrectangles which edges are parallel to axes of the coordinates system. This language of description is sufficient in most of problems is insufficient for the data that represents oblique dependencies. But even if it is possible to separate two classes linearly with the hyperplane what is non-linear to all axes in the coordinates system (only one oblique rules per decision class is needed) the generated set of rule will contain a lot of rules. What is also significant there will be a lot of small rules in this set, rules that cover only the small part of the data space – possibly only the object that generated this rule and its very close neighbourhood. It has a very important consequence: the final set of rules should be the compromise between the accurate model, containing all rules, but very complex and the model containing only the best, strongest rules, but also having good ability of generating the dependencies in the data with the cost of an acceptable level of accuracy decrease.

Algorithms of induction of decision rule with oblique decisions can solve the problem presented above. In these methods the single condition of the rule is the fact of lying the point over or under the certain hyperplane. There are several propositions in the literature, proposing different ways of rule induction and some of them give oblique decision rules. Two main branches are based on the neural networks [14][24][25] and Support Vector Machines [1][10][12][13][22]. Also the induction of decision rules from the decision trees with the oblique conditions is popular [2][20].

As it was mentioned above, Support Vector Machines are considered as the starting point of oblique rule generating very often. One of the approaches [1] can be described as the extrapolation of Support Vector Machine results on the whole dataset space, which is the input of the standard decision rule induction. After the nonlinear Support Vector Machine are trained “empty regions” of the dataset space are filled with the artificial objects which class is Support Vector Machine dependent. Then mixed (original and artificial) set of objects generates hyperrectangular decision rules.

Fung et al. [10] base on linear Support Vector Machine results. After the data normalisation hyperrectangular decision rules covering the hyperplane introduced halfspace are searched. This means that linear Support Vector Machine results are just the base for hyperrectangular decision rule induction what is the opposite approach to the requested research.

The ITER algorithm [12] generates regression decision rules from any trained “black box” regression model like artificial neural networks or support vector machines but they also remain the hyperrectangular. Its later modification Minerva [13] assures rules to be non-overlapped but still non-oblique.

Very interesting approach is presented in [22] called SVM+Prototypes. Each class is divided into subclasses and their prototypes with the clustering algorithm. Then, for each subcluster the decision rule of the form of an ellipsoid or a hyperrectangle is inducted.

As the mentioned groups start from the non-rule classification methods, the CHIRA [26] starts from the set of hyperrectangular decision rules and postprocess them to the oblique ones. The algorithm merges rules iteratively, in pairs. It applies the procedure of determining convex hulls for regions in a feature space which are covered by aggregated rules. Two different steps of postprocessing are defined: rules joining and rules aggregation. The rules joining saves the hyperrectangular form of the rule as it is the result of merging two rules that contain the same attributes in elementary conditions. Rules aggregation builds the rule with oblique conditions. The aggregation procedure relies on the assumption that a single decision rule indicates a convex area in a particularly chosen feature subspace. CHIRA tries to iteratively join such areas in order to obtain larger convex regions, from which hyperplanes equations can be calculated. To determine convex region for initial rules, the algorithm extracts boundary points from elementary conditions, builds a set of rule vertices upon them and finally, applies convex hull calculation procedure.

The ADRED algorithm starts directly from the data to generate oblique rules [23]. In this approach each rule is generated for some defined cluster of objects. For each pair of axes a number a number of hyperplanes is generated. A hyperplane is considered as an optimal if it maximizes the number of negative points (points not from the class described by the generating rule) over the hyperplane and while keeping all positive points under the hyperplane.

In the previous works several two other approach of oblique decision rule induction were presented. First of them based on grid search of parameters of each oblique condition [19]. This caused very high computational complexity. In the second approach ([18] and improved in [17]) classes were divided into subclasses (with k-means algorithm) and each subclass was described with the one hyperrectangle which edges was parallel to PCA determined directions. It was not resistant to the overlapping and non-convex subclasses.

III. Decision Rules

Decision rule can be defined as the following logical formula:

$$\text{IF } cond_1 \wedge cond_1 \wedge \ldots \wedge cond_n \text{ THEN } class = c \qquad (1)$$

where $cond_i$ denotes some logical expression of the type a op A: a is a value of the variable, op is the one of logical operators (=; <; >; ≥; ≤; „in’’) and A is a constant (an interval for the „in’’ operator). The c in the rule conclusion represents one of the existing class labels.

The main goal of rule induction is to build dependencies that can be interpreted by the user or the domain expert. Focusing on this goal is limited by the rules ability of generalisation – we are not interested in big set of small, particular rules, explaining the nature of a very small number

Proc. of the Intl. Conf. on Advances In Information Processing And Communication Technology - IPCT 2014.
Copyright © Institute of Research Engineers and Doctors. All rights reserved.
ISBN: 978-1-63248-021-7 doi: 10.15224/ 978-1-63248-021-7-70

of objects. It is very common situation when we approve a small model accuracy decrease as the cost of limitation the number of rules or their complexity. Typical methods of the mentioned rule postprocessing step are rules filtering, joining and shortening.

A. *Oblique Decision Rules*

Most popular algorithms find decision rules that are hyperrectangles. This limitation avoids finding small sets of rules for datasets with typical oblique dependencies and implies finding big sets of smaller rules instead.

To define the oblique rule the definition of single oblique condition must be shown. Let's consider the *k*–dimensional space of objects. As the base of separating condition the following *k*–dimensional hyperplane can be used:

$$H_1x_1 + H_2x_2 + \ldots + H_kx_k + H_{k+1}=0 \qquad (2)$$

For an object *O* the following coefficient can be defined:

$$H(O)=H_1o_1 + H_2o_2 + \ldots + H_ko_k + H_{k+1} \qquad (3)$$

where o_1, o_2, …, o_k are coordinates of the object *O*. With this notions as the single oblique condition one of the following can be used:

$$H(O) > 0;\ H(O) < 0\ \ H(O) \geq 0\ \ H(O) \leq 0 \qquad (4)$$

B. *LDA Background*

Linear discriminant analysis (LDA) [8] is a statistical tool for finding the linear combination of objects' features that separates two classes from each other. In the original approach, the direction of linear projection of points which

This approach is based on assumptions that probability distribution functions of each class have normal distribution and equal covariances.

IV. LDA Oblique Decision Rules

In the previous paper [19] the grid search of oblique condition parameters was applied, what implied a big complexity of the algorithm. The current approach uses LDA as a much faster statistical tool for finding a single oblique condition parameters.

For each class a maximal number of rules *R* and maximal number of conditions in the rule *C* are assumed. The strategy of building a single rule is a „separate-and-conquer'' strategy. In the one step the maximal (from the number of oblique conditions point of view) rule is generated and in the second, all positive objects from the considered class (objects, covered by the newly created rule) are removed from the training set.

Single rule is created iteratively: while the rule is not maximally long or is not exact (covers only positive objects) the oblique condition with the LDA is searched. When the condition is found all objects below the hyperplane ($H(O) < 0$) are removed from the further rule generation.

When there are only two classes in the data (e.g. 1 and -1) it is easy to perform the rule induction – only two iterations, one for the class. In other cases – when there are more than two classes – in each iteration we have to change the problem into the binary as follows: for the classes from 1 to *n*, in *i*-th iteration change all labels different from *i* to -1 and all labels *i* to 1.

V. Case Study

Let us consider the following artificial dataset (taken from [26]). It is shown on the Fig. 1.

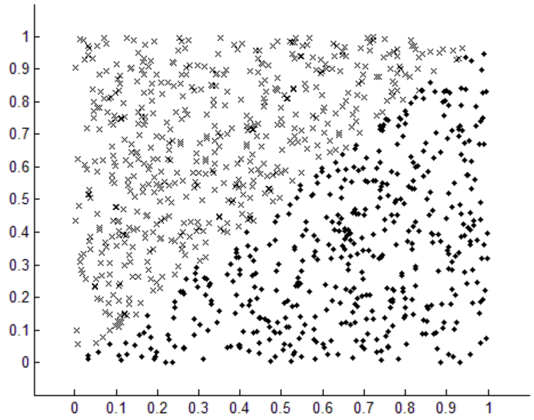

Figure 1. Two-dimensional artificial data (2D).

This data contains 1000 points randomly distributed in the square (0,0)x(1,1) which are assigned to two almost balanced classes (466 dots and 534 crosses). We can observe that class that point belongs to depends on the lying over (crosses) or under (dots) the line *y* = *x*. This cause that it is very hard to describe this simple dependence with rectangular decision rules.

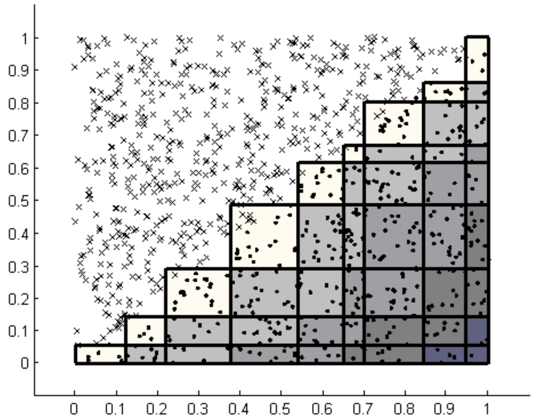

Figure 2. Visualisation of JRip generated decision rules.

A simple effort to cover the class of dots with the usage of JRip algorithm (a Weka implementation of RIPPER algorithm) is shown on the Fig. 2.

There are 9 rectangular and overlapping rules (the increase of the overlapping level of the data space is represented with the increase of the grey level in the image) describing the class of dots. From the other hand, the usage of LDA based oblique decision rules gives us only one rule (with three parameters) that can describe the considered class (as it is shown on the Fig. 3.).

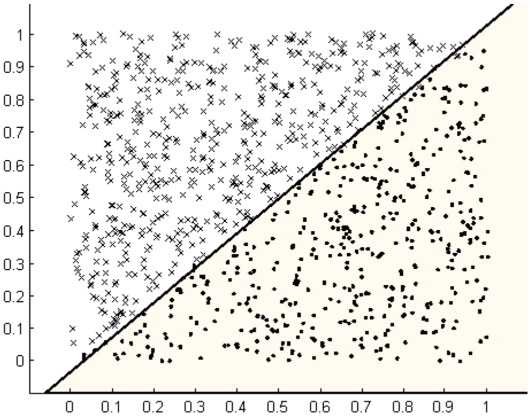

Figure 3. Visualisation of single oblique conition decision rule obtained with the LDA.

VI. Results

Experiments were performed on several commonly known benchmark datasets from the UCI ML Repository and three artificial datasets with the typical oblique dependencies from [26]. First of these three datasets was presented on the Fig. 1.

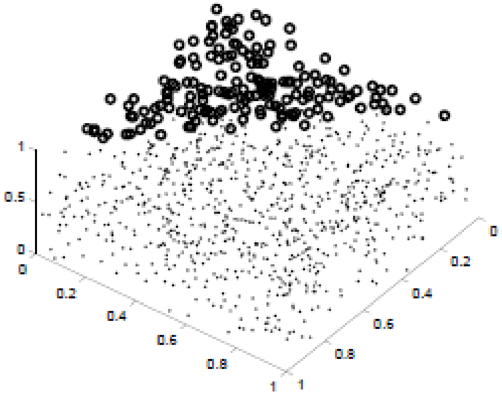

Figure 4. Three-dimensional artificial data (3D).

The second one contains three-dimensional data (called as 3D) with two unbalanced classes in the unitary cube: circles are grouped in one of the cube corners (Fig. 4.). Last one dataset are two-dimensional and balanced but the class of dots is not coherent (Fig. 5.).

The new method of rule induction was compared with the JRip algorithm, the Weka software implementation of the RIPPER [7].

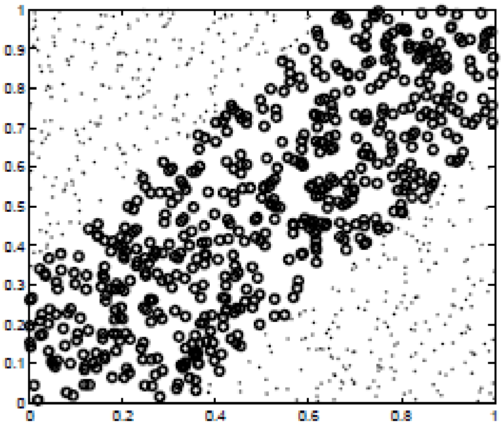

Figure 5. Visualisation of the second two-dimensional artificial data (2D2).

For each dataset a 10 cross-fold validation model of train and test iteration was performed and averaged results are presented in the Table I.

TABLE I. COMPARISON OF RESULTS OF JRIP AND LDA OBLIQUE RULES

Dataset	JRip			LDA Oblique Rules		
	acc [%]	#rules	#cond	acc [%]	#rules	#cond
2D	96	10	18	**99**	2	6
3D	95	8	19	**99**	4	28
2D2	96	9	18	**97**	6	36
Ripley	86	2	1	82	2	6
Balance	81	11	30	**91**	3	15
Breast	96	6	10	93	2	20
Bupa	65	3	4	**67**	6	84
Heart	79	5	8	**84**	2	28
Iris	95	3	4	93	5	45
Parkinson	88	5	9	83	2	46
Pima	75	3	5	**76**	2	18

The comparison is presented due to the accuracy of prediction (acc [%]), number of rules (#rules) and number of parameters of the model (#cond). This last parameter is the sum of conditions in all rectangular rules and the number of coefficients of all oblique conditions.

VII. Discussion

When the accuracy of prediction of two classifiers are compared for 7 datasets LDA Oblique Rules gives better results than JRip (accuracies of LDA OR typed with the bold font). It also must be stressed that JRip builds models with the default rule(class) – in case when an object does not recognize any rule it is classified as the default class (the usage of default rule). Therefore, the total number of rules and conditions of JRip results describes only the one class (to be more precise – the number of rules must be decreased by one, the number of conditions remains). Only two sets have more than two classes: Balance, Iris (marked with a underlined font). This means that Breast dataset is described by LDA OR with only one rule per decision class, while JRip needs five rules per class (assuming the comparable complexity of description of the default class).

VIII. Conclusions and Further Works

The usage of LDA as the statistical tool for induction of decision rules with oblique conditions gives several significant remarks. First – it is not appropriate tool for datasets which classes are wrapping themselves. It is a little bit weaker assumption that convex classes: classes can be concave but their convex complements intersection must be an empty set.

This limitation is strongly connected with the limitation of the LDA method, mentioned in the beginning of the paper. What is very important to stress is all assumptions are global. This means that any globally based measures cannot give satisfactory results of oblique decision rules induction in cases of wrapping classes.

It seems that this approach of direct oblique decision rules needs the locally based measure of linear separation of classes, which will help to improve the final decision rules qualities.

References

[1] N. Barakat, and J. Diederich, “Learning-based Rule-extraction from Support Vector Machines”, In 14th Int. Conf. on Computer Theory and App. ICCTA 2004 Proceedings, Alexandria, Egypt, 107-112, 2004

[2] K.P. Bennet, and J.A. Blue, “A Support Vector Machine Approach to Decision Trees”, Proc. of the IEEE Int. Joint Conf. on Neural Netw., pp. 2396–2401, 1998.

[3] C. de Boor, A Practical Guide to Splines, Springer, 2001

[4] B. Boser, I. Guyon, and V. Vapnik, “A Training Algorithm for Optimal Margin Classifiers”, In: Proc. of the 5th Ann. Workshop on Comput. Learn. Theory, pp. 144–152, 1992.

[5] M. Buhmann, Radial Basis Functions: Theory and Implementations, Cambridge University Press, 2003

[6] P. Clark, and T. Niblett, ”The CN2 Induction Algorithm”, Mach. Learn., vol. 3(4), pp. 261–283, 1989.

[7] W.W. Cohen, “Fast Effective Rule Induction”, Proc. of the Twelfth Int. Conf. on Mach. Learn., pp. 115–123, 1995.

[8] R.A. Fisher, ”The Use of Multiple Measurements in Taxonomic Problems”, Ann. of Eugen., vol. 7(2), pp. 179-188, 1936.

[9] J. Friedman, and W. Stuetzle, “Projection Pursuit Regression”. J. of the Am. Stat. Ass., vol. 76, pp. 817 – 823, 1981.

[10] G. Fung, S. Sandilya, and R. Rao, “Rule Extraction from Linear Support Vector Machines”. In Proc. of the 11th ACM SIGKDD Int. Conf. on Know. Discov. in Data Min., pp. 32–40, 2005.

[11] J. Fürnkranz, “Separate-and-conquer Rule Learning”, Artif. Intell. Rev., vol. 13(1), pp. 3–54, 1999.

[12] J. Huysmans, B. Baesens, and J. Vanthienen, “ITER: an Algorithm for Predictive Regression Rule Extraction”, Lect. Notes in Comp. Sci., vol. 4081, pp. 270–279, 2006.

[13] J. Huysmans, R. Setiono, B. Baesens, and J. Vanthienen, “Minerva: Sequential Covering for Rule Extraction”, IEEE Trans. On Syst., Man, and Cybern.—Part B: Cybern., vol. 38(2), pp. 299-309, 2008.

[14] D. Kim, and J. Lee, “Instance-Based Method to Extract Rules from Neural Networks”, Lect. Notes in Comp. Sci., vol. 2130, pp. 1193-1198, 2001.

[15] R. Kohavi, and J. Quinlan, Data Mining Tasks and Methods: Classification: Decision-tree Discovery, in: Handbook of Data Mining and Knowledge Discovery, Oxford University Press, 2003

[16] W. McCulloch, and W. Pitts, “A Logical Calculus of the Ideas Immanent in Nervous Activity” , Bull. of Math. Biophys., vol. 5, pp. 115-133, 1943.

[17] M. Michalak, and K. Nurzyńska, “Advanced Oblique Rule Generating Based on PCA”, Lect. Notes in Comp. Sci., 2014 (in press)

[18] M. Michalak, and K. Nurzyńska, “PCA Based Oblique Decision Rules Generating”, Lect. Notes in Comp. Sci., vol. 7824, pp. 198-207, 2013.

[19] M. Michalak, M. Sikora, and P. Ziarnik, “ORG – Oblique Rules Generator”, Lect. Notes in Comp. Sci., vol. 7268, pp. 152-159, 2012.

[20] S. Murthy, S. Kasif, and S. Salzberg, “A System for Induction of Oblique Decision Trees”, J. of Artif. Intell. Res., vol. 2, pp. 1-32, 1994.

[21] N. Nadaraya, “On Estimating Regression”, Theory of Probab. and Its App., vol. 9(1), pp. 141–142, 1964.

[22] H. Nunez, C. Angulo, and A. Catala,”Rule Extraction from Support Vector Machines”, Proc. of Eur. Symp. on Artif. Netw., pp. 107-112, 2002.

[23] Z. Raś, A. Daradzińska, and X. Liu, “System ADReD for Discovering Rules Based on Hyperplanes”, Eng. App. of Artif. Intell., vol. 17(4), pp. 401–406, 2004.

[24] E. Saad, and D. II Wunsch, “Neural Network Explanation Using Inversion”, Neural Netw., vol. 20, pp. 78-93, 2007.

[25] R. Setiono, and B. Baesens, and C. Mues, “Rule Extraction from Minimal Neural Networks for Credit Card Screening”, Int. J. of Neural Syst., vol. 21(4), pp. 265-76, 2011.

[26] M. Sikora, and A. Gudyś, “CHIRA – Convex Hull Based Iterative Algorithm of Rules Aggregation”, Fundam. Inform., vol. 123, pp.143-170, 2013.

[27] G. Watson, “Smooth Regression Analysis”. Sankhya - The Indian J. of Stat., vol 26(4), pp. 359–372, 1964.

About Author:

M. Michalak (1981) received his M.Sc. Eng. and Ph.D. degree in the field of computer science from the Silesian University of Technology (SUT), Poland in 2005 and 2009, respectively. Since 2009 he is a postdoctoral fellow in the Faculty of Automatic Control, Electronics and Computer Science at SUT and in the years 2009 – 2012 he was an engineer (later the Assistant Professor) at Central Mining Institute, Poland. He was also employed in the coal mining industry as the specialist of the data analysis. He is an author of over 50 publications from the fields of data mining, machines diagnosis, binary biclustering, rough sets theory, time series analysis, multi-spectral images analysis, oblique rule induction and many others.