Gaining Insights from Symbolic Regression Representations of Class Boundaries

Ingo Schwab
Karlsruhe University of Applied Sciences
Karlsruhe, Germany
Ingo.Schwab@hs-karlsruhe.de

Norbert Link
Karlsruhe University of Applied Sciences
Karlsruhe, Germany
Norbert.Link@hs-karlsruhe.de

***Abstract*- In this paper, we propose a generalization of the well-known regression analysis to fulfill supervised classification aiming to produce a learning model which best separates the class members of a labeled training set. The class boundaries are given by a separation surface which is represented by the level set of a model function. The separation boundary is defined by the respective equation. The model is represented by mathematical formulas and composed of an optimum set of expressions of a given superset. We show that this property gives human experts additional insight in the application domain. Furthermore, the representation in terms of mathematical formulas (e.g., the analytical model and its first and second derivative) adds additional value to the classifier and enables to answer questions, which other classifier approaches cannot. The symbolic representation of the models enables an interpretation by human experts.**

Keywords-Classification; Symbolic Regression; Knowledge Management; Data Mining; Pattern Recognition.

I. INTRODUCTION

Supervised classification algorithms aim at assigning a class label for each input example. Given a training dataset of the form (x_i, y_i), where $x_i \in \Re^n$ is the *i*th example and $y_i \in \{-1,+1\}$ is the *i*th class label in a binary classification task. A model φ is learned, so that $\varphi(x_i) = y_i$ for new unseen examples. In fact, it is an optimization task and the learning process is mainly data driven. It results in an adaptation of the model to reproduce the data with as few errors as possible. Several algorithms have been proposed to solve this task and the result of the learning process is an internal knowledge model φ.

There are basically two ways to represent the knowledge of model φ. The first approach includes algorithms like Naïve Bayes Classifiers, Hidden Markov Models or Belief Networks [1]. The main idea is to represent it as probability distribution. The classification boundary is the intersection of the posterior probabilities in Bayes decision theory.

The other approach for representing the knowledge is to determine a surface in the feature space which separates the different classes of the training data as good as possible. The decision surface is represented by parameterized functions which can be the sum of weighted base functions of one function class. Examples include the logistic functions and radial basis functions, which can be used in Neural Networks and Support Vector Machines [1].

It is important to point out that the base functions are closely linked to the used classifiers. Our approach further refines this idea (see Sections II and III). Again, the decision surface is determined by a level surface of a model function. But, in this case, the function is composed of arbitrary mathematical symbols, forming a valid expression of a parameterized function. This approach allows the human users of the system to control the structure and complexity of the solutions.

Following this idea, we try to find solutions which are as short (and understandable) as possible. Additionally, the selected solutions should model the dataset as good as possible. Clearly, this is a contradiction and of the nature of multiobjective decision making. Therefore, we select all good compromises of the pareto front [4] and sort them by complexity. This approach extends the concept presented in [11] and helps human experts to choose the best compromise. Standard classification approaches (e.g., Neural Networks) in which the structure of the base functions is predefined are not able to reduce their structural complexity. In most nontrivial applications they are not understandable to the human expert and the represented knowledge can therefore not be refined and reused for other purposes [2].

There are many different ways to further subdivide this class of learning algorithms (e.g., greedy and lazy, inductive and deductive [5]). In this paper we focus on the symbolic and subsymbolic paradigm (see [2][3] for more details) and its consequences for the reusability of the model φ and the inherent learned knowledge. This subdivision separates the approaches with symbolic representations in which the knowledge of model φ is characterized by explicit symbols, whereas subsymbolic are associated with continuous representations. One of the main disadvantages of subsymbolic classifiers (e.g., Neural Network or SVM) is that the class of classifiers includes rather the properties of a black box and the learned model cannot be interpreted or reformulated.

The main advantages of our approach (see Table I) are determined by the inherent nature of mathematical formulas and there are many rules to reformulate, simplify and derive additional information from them (e.g., first and second derivative). In fact, reformulating mathematical formulas is one of the most important areas of mathematics. For the black box character of the subsymbolic learning algorithms such rules simply do not exist.

The remaining part of this paper is arranged as follows. In Section II, the proposed Symbolic Regression Algorithm is presented. Section III summarizes our approach and shows how to generalize the regression task to classification. Furthermore, the main advantages of the approach are briefly shown. Section IV explains some of our experiments and Section V concludes.

II. BACKGROUND AND RELATED WORK

A. Symbolic vs. Subymbolic Representation

As Smolensky [3] noted, the term subsymbolic paradigm is intended to suggest symbolic representations that are built out of many smaller constitutes: “Entities that are typically represented in the symbolic paradigm by symbols are typically represented in the subsymbolic paradigm by a large number of subsymbols” (p.3). From this point of view the syntactic role of subsymbols can be described that subsymbols participate in numerical computation. In contrast operations in the symbolic paradigm that consist of a single discrete operation are often achieved in the subsymbolic paradigm as the result of a large number of much finer-grained numerical operations. One well known problem with subsymbolic networks which have undergone training is that they are extremely difficult to interpret and analyze. In [2], it is argued that it is the inexplicable nature of mature networks.

B. Pareto Front

In this subsection we discuss the Pareto Front or Pareto Set in multiobjective decision making [4]. This area of research has a strong impact on machine learning and data mining algorithms.

Many problems in the design of complex systems are formulated as optimization problems, where design choices are encoded as valuations of decision variables and the relative merits of each choice are expressed via a utility or cost function over the decision variables.

In most real-life optimization situations, however, the cost function is multidimensional. For example, a car can be evaluated according to its cost, size, fuel consumption, storage room, and a configuration *s* which is better than *s*' according to one criteria, can be worse according to another. Consequently, there is no unique optimal solution but rather a set of efficient solutions, also known as pareto solutions, characterized by the fact that their cost cannot be improved in one dimension without being worsened in another. In machine learning algorithms the competing criteria are the prediction accuracy and the size of the learning model.

The set of all Pareto solutions, the Pareto front, represents the problem trade-offs, and being able to sample this set in a representative manner is a very useful aid in decision making. In other words the solutions are ordered by complexity. Through the symbolic representation the human expert is able to interpret the solutions of the pareto front (see section IV c).

C. Classical Regression Analysis and Symbolic Regression

Regression analysis [7] is one of the basic tools of scientific investigation enabling identification of functional relationship between independent and dependent variables. The general task of regression analysis is defined as identification of a functional relationship between the independent variables $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ and dependent variables $\mathbf{y} = [y_1, y_2, \ldots, y_m]$, where n is a number of independent variables in each observation and m is a number of dependent variables.

The task is often reduced from an identification of a functional relationship f to an identification of the parameter values of a predefined (e.g., linear) function. That means that the structure of the function is predefined by a human expert and only the free parameters are adjusted. From this point of view Symbolic Regression goes much further.

Like other statistical and machine learning regression techniques Symbolic Regression also tries to fit observed experimental data. But unlike the well-known regression techniques in statistics and machine learning, Symbolic Regression is used to identify an analytical mathematical description and it has more degrees of freedom in building it. A set of predefined (basic) operators is defined (e.g., add, multiply, sin, cos) and the algorithm is mostly free in concatenating them. In contrast to the classical regression approaches which optimize the parameters of a predefined structure, here also the structure of the function is free and the algorithm both optimizes the parameters and the structure of the base functions.

There are different ways to represent the solutions in Symbolic Regression. For example, informal and formal grammars have been used in Genetic Programming to enhance the representation and the efficiency of a number of applications including Symbolic Regression [8].

Since Symbolic Regression operates on discrete representations of mathematical formulas, non-standard optimization methods are needed to fit the data. The main idea of the algorithm is to focus the search on promising areas of the target space while abandoning unpromising solutions (see [4][9] for more details). In order to achieve this, the Symbolic Regression algorithm uses the main mechanisms of Genetic and Evolutionary Algorithms. In particular, these are mutation, crossover and selection [9] which are applied to an algebraic mathematical representation.

The representation is encoded in a tree [9] (see Figure 1). Both the parameters and the form of the equation are subject to search in the target space of all possible mathematical expressions of the tree. The operations are nodes in the tree (Figure 1 represents the formula 6x+2) and can be mathematical operations such as additions (add), multiplications (mul), abs, exp and others. The terminal values of the tree consist of the function's input variables and real numbers. The input variables are replaced by the values of the training dataset.

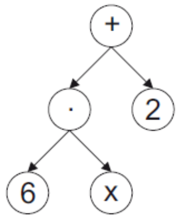

Figure 1. Tree representation of the equation 6x+2.

In Symbolic Regression, many initially random symbolic equations compete to model experimental data in the most promising way. Promising are those solutions which are a good compromise between correct prediction quality of the experimental data and the length of the symbolic complexity.

Mutation in a symbolic expression can change the mathematical type of formula in different ways. For example, a div is changed to an add, the arguments of an operation are replaced (e.g., change 2*x to 3*x), an operation is deleted (e.g., change 2*x+1 to 2*x), or an operation is added (e.g., change 2*x to 2*x+1).

The fitness objective in Symbolic Regression, like in other machine learning and data mining mechanisms, is to minimize the regression error on the training set. After an equation reaches a desired quality level of accuracy, the algorithm returns the best equation or a set of good solutions (the pareto front). In many cases the solution reflects the underlying principles of the observed system.

III. PROPOSED METHOD

This section explains our knowledge acquisition workflow (see Figure 2). The core of the workflow is structured in 4 steps.

1. The human expert defines the set of base functions. The functions should be adapted to the domain problem. For example many geometrical problems are much easier to solve with trigonometric base functions.
2. The second step in the workflow is the main optimization process (see [12], section II c. and III a. of this paper for more details). Symbolic Regression is used to solve this task. It should be noted, however, that other optimization algorithms which can handle discrete black-box optimization can be used for this task.
3. A human expert can interpret and reformulate the solutions of the pareto front (see section IV b.).
4. The knowledge can be transferred to other domains.

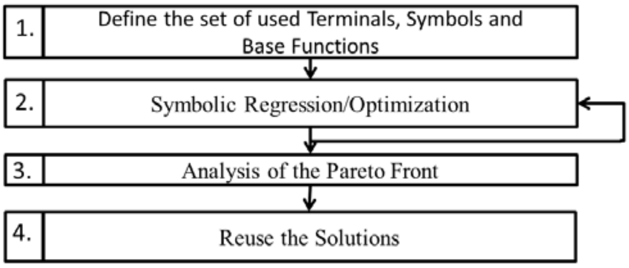
1.
Define the set of used Terminals, Symbols and Base Functions
2.
Symbolic Regression/Optimization
3.
Analysis of the Pareto Front
4.
Reuse the Solutions

Figure 2. The knowlede acquisition workflow.

A. From Regression to Classification

In this subsection the symbolic regression algorithm is generalized to a symbolic regression classification.

First, an activation function is defined. In our approach it is a step function which is defined as $\Phi(z)=\begin{cases}1\, iff\ z\geq 0\\ 0\, iff\ z<0\end{cases}$.

Given is a training set of N feature vectors $\{\overrightarrow{x_i}\}_{i=1}^N$ and assigned class label $\{y_i\}_{i=1}^N$, $y_i\in[0,1]$. The main challenge and computer time consuming task is to find a function f which transforms the input space in the way that $\Phi(f(\vec{x}))=\vec{y}$ with as few errors as possible. In other words, a function $f(\vec{x})$ is sought with $f(\vec{x})=0$ separating the areas of the feature space, where the vectors of the different classes are located. The zero-crossing $f(\vec{x})=0$ therefore defines the decision surface. So far, the approach is Perceptron-like [6]. Instead of replacing the step-functions by continuous and differentiable base functions to allow cost function optimization, Symbolic Regression is used to optimize the cost function $J=\sum_{i=1}^{N}(\Phi[f(\overrightarrow{x_i})]-y_i)^2$ and therefore to find $f(\vec{x})$.

The main advantage of this approach is due to the fact that complexity and interpretability of the solution can be controlled by the user by the set of allowed operations and by selecting the appropriate complexity by means of the pareto front. Further approach advantages (see the next subsection) are consequences of this property.

B. Advantages

In this subsection we summarize the additional advantages of the proposed approach. It should be noted that all mathematical reformulations of the classifier do not change its behavior in classification.

To be understandable to human experts our approach tries to find solutions which are as simple as possible. The pareto front [4] sorts the solutions by complexity and prediction quality.

One of the main advantages is that this approach enables to calculate the first derivative of the classifier. One scenario could be in engineering technologies or medical systems.

TABLE I. ADVANTAGES

Can be interpreted by human experts
Can be reused in other domains
Knowledge Base
Rules to simplify and reformulate. The reformulations do not change its bahaviour.
Analytical Boundary Detection
Analytical Gradient Calculation
Blocks of analytical expert knowledge can be used

For example it could be the task to learn when a workpiece is damaged or when there is a risk of an illness. The general learning approaches enable only to predict the class (e.g., defect or no defect). With the first derivative which can be analytically calculated by our approach we can also say which attributes of the classifier should be changed (and in which direction) to leave the undesired class as soon as possible.

IV. EXPERIMENTS AND RESULTS

This section discusses and demonstrates some of the conducted experiments. First we show two experiments based on artificial datasets while the third described experiment is based on a real-word dataset.

A. *First Experiment*

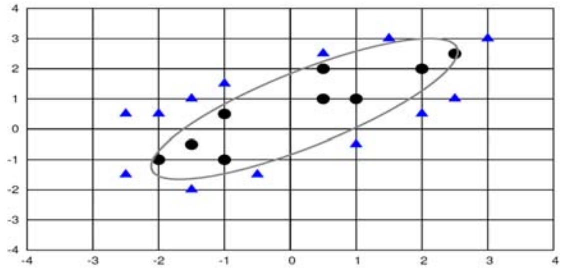

Figure 3. First dataset.

Figure 3 shows the data of a two class learning task in a two-dimensional plot. The first class is represented by the circles and the second by the triangles. The zero-crossing $f(\vec{x}) = 0$ decision boundary of the different classes of formula 1 (calculated by our Symbolic Regression algorithm) is displayed in Figures 3 and 4 by the parabola. In order to find interpretable formulas we restricted the search on using add, sub, mul and all real numbers as operators.

$$f(x, y) = 1.54516 + y + 1.63312\,xy - x^2 - y^2 - 0.672694\,x \quad (1)$$

As diskussed in Section III, it is easy for a human expert to interprete this solution. It is a representation of a ellipse. With this knowledge the user can conlude much more about the domain. The additional knowledge includes conclusions about the decision area. Based on their high complexity black box machine learning algorithms usually give no additional insight into its bahavior.

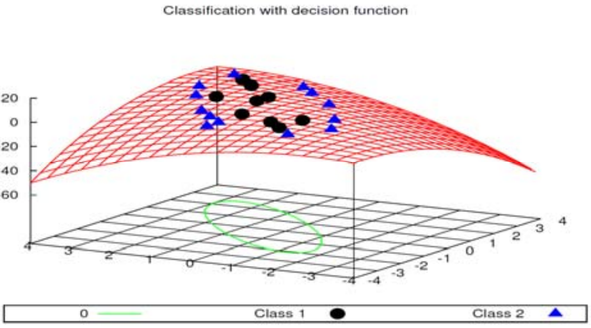

Figure 4. The tranformation of the feature space.

As a result of the interpretable analytical solution (formula 1) we know that there is only one decision boundary (the zero-crossing). This knowledge is essential for some domains (application scenarios can include medical or other critical domains) which require robust classifiers. This robustness includes predictable behaviour to unknown datasets which include so far uncovered areas of the feature space.

B. *Second Experiment*

The second experiment is based on the well-known spiral dataset [10][12]. The problem to distinguishing two intertwined spirals is a non-trivial one. Figure 5 depicts the 970 patterns that form the two intertwined spirals. These patterns were provided in [10].

This experiment is an example of the way in which additional human expert knowledge can improve the quality of the found solutions (see section III). For a human expert it is obvious that the problem is periodic. To find good and short models it is therefore essential to add periodic and trigonometric base functions. Therefore, we allowed the algorithm to use additions (add), substractions (sub), divisions (div), multiplications (mul), sin, cos and all real numbers. Several correct problem solving solutions had been found by our system for this classifictaion problem. One of them is formula (2) (the numbers in the formula are rounded using 3 fractional digits) which is able to classify the spiral dataset without an error and figure 6 shows the three-dimensional plot of the function. To the best of our knowledge it is one of the shortest known solutions to solve this classification tasks.

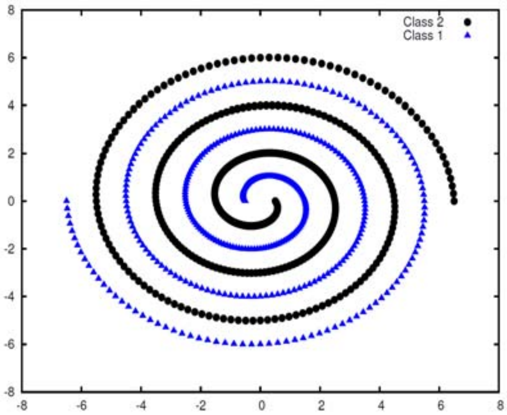

Figure 5. The spiral dataset.

$$f(x,y) = \sin\left(3{,}35\,x + \frac{y}{\dfrac{0{,}042}{\dfrac{x}{y} + 0{,}005 * \dfrac{y}{x}} + \dfrac{x}{y} - 0{,}0356}\right) \quad (2)$$

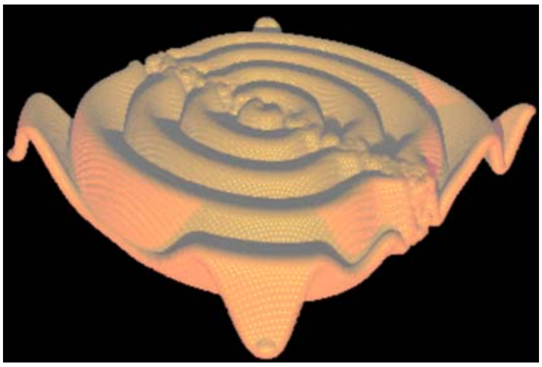

Figure 6. The three-dimensional plot of function 3.

C. *Real Life Dataset – Haberman's Survival Data Set*

The Habermans's Survival Dataset dataset contains cases from a medical study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone breast cancer surgery [13][14][15][16].

It consists of 4 attributes:
1. Age of patient at time of operation (age).
2. Patient's year of operation (the year of the operation).
3. Number of positive axillary nodes detected (nodes).
4. The survival status (class attribute) .

Table II summarizes the rules of the pareto front of one run found by our Symbolic Regression system [12]. The formulas are ordered by complexity. It should be mentioned that repeating this procedure can result in different solutions.

complexity	accuracy	formula
13	0.478355	*f*(age,operation,nodes) = operation/(2.05368*age*nodes - 83.8188*nodes - 154.58)
9	0.493074	*f*(age,operation,nodes) = nodes*nodes/(age - 43.7473) - 6.15
7	0.493074	*f*(age,operation,nodes) = age - 71/nodes - 41.35
5	0.52987	*f*(age,operation,nodes) = nodes - 469.83/age
3	0.544589	*f*(age,operation,nodes) = nodes - 8.69
1	0.596104	*f*(age,operation,nodes) = 0

TABLE II. RULES.

As a simple showcase to show how additional insights in a domain can be gained we consider the formula *f*(age,operation,nodes) = age - 71/nodes - 41.35 (complexity 7) in Table II. It can be reformulated by age = 71/nodes + 41.35. A human user knows that the number of axillary nodes cannot have negative values. This implies that if the age of the patient is less than 41.35 the survival status is greater than 50 percent. This simple explample shows, that reformulating and adding additional domain knwoledge adds further insight. New knowledge is derived and it can be used in another context. This procedure is however, only possible on the basis of the symbolic and interpretable representation of the formulas (see section II).

V. CONCLUSIONS AND FUTURE WORK

In this paper, we showed a generalization of the well-known regression task to classification problems. Furthermore, the focus was set on understandable solutions achieved via Symbolic Regression which enable human experts to redefine and reuse the knowledge. Very important is that mathematically correct reformulations of the classifier formulas do not change its properties. Additional knowledge can be derived by reformulation the formulas. The power of our approach has been shown in experiments. Future work will focus on how the developed techniques can be transferred to other domains. Additionally, we will cooperate with our industrial partners to put the approaches into practice.

REFERENCES

[1] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern Classification", 2nd ed., Wiley Interscience, 2000.

[2] D. Robinson, "Implications of Neural Networks for How We Think about Brain Function", in Behavioral and Brain Science, 15, pp. 644-655, 1992.

[3] P. Smolensky, "On the Proper Treatment of Connectionism", in Behavioral and Brain Sciences, 11, pp. 1-74, 1988.

[4] R. E. Steuer, ”Multiple Criteria Optimization: Theory, Computations, and Application”. New York: John Wiley & Sons, 1986.

[5] J. H. Holland, K. J. Holyoak, R. E. Nisbett, and P. R. Thagard, “Induction: Processes of Inference, Learning, and Discovery”. Cambridge, MA, USA, 1989.

[6] W. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity”. Bulletin of Mathematical Biophysics, pp. 115-133, 1943.

[7] D. A. Freedman, ”Statistical Models: Theory and Practice, Cambridge University Press, 2005.

[8] M. O'Neill and C. Ryan, "Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language"; Kluwer Academic Publishers, Dordrecht Netherlands, 2003.

[9] J. R. Koza, ”Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge, MA, USA: MIT Press, 1992.

[10] K. Lang and M. Witbrock, ”Learning to tell two spirals apart”. Proceedings of 1988 Connectionists Models Summer School. Morgan Kaufmann, San Mateo CA, pp. 52-59, 1989.

[11] J. K. Kishore, L. M. Patnaik, V. Mani, and V. K. Agrawal “Application of Genetic Programming for Multicategory Pattern Classification”. IEEE Transactions on Evolutionary Computation, 4 (3). pp. 242-258, 2000.

[12] I. Schwab and N. Link, “Reusable Knowledge from Symbolic Regression Classification”, Genetic and Evolutionary Computing (ICGEC 2011), 2011.

[13] S. J. Haberman. Generalized Residuals for Log-Linear Models, Proceedings of the 9th International Biometrics Conference, Boston, pp. 104-122, 1976.

[14] J. M. Landwehr, D. Pregibon, and A. C. Shoemaker, Graphical Models for Assessing Logistic Regression Models (with discussion), Journal of the American Statistical Association 79: pp. 61-83, 1984.

[15] W.-D., Lo. “Logistic Regression Trees”, PhD thesis, Department of Statistics, University of Wisconsin, Madison, WI, 1993.

[16] A. Frank, A. Asuncion. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2010.