Applied Soft Computing xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Hybridizing Cartesian Genetic Programming and Harmony Search for adaptive feature construction in supervised learning problems

Andoni Elola [a], Javier Del Ser [a,b,c,*], Miren Nekane Bilbao [a], Cristina Perfecto [a], Enrique Alexandre [d], Sancho Salcedo-Sanz [d]

[a] University of the Basque Country UPV/EHU, Alameda Urquijo S/N, 48013 Bilbao, Spain
[b] TECNALIA, P. Tecnologico Bizkaia, Ed. 700, 48160 Derio, Spain
[c] Basque Center for Applied Mathematics (BCAM), 48009 Bilbao, Spain
[d] Universidad de Alcalá, 28805 Alcalá de Henares, Madrid, Spain

ARTICLE INFO

Article history:
Received 14 April 2016
Received in revised form
28 September 2016
Accepted 29 September 2016
Available online xxx

Keywords:
Feature construction
Supervised learning
Cartesian Genetic Programming
Harmony Search

ABSTRACT

The advent of the so-called Big Data paradigm has motivated a flurry of research aimed at enhancing machine learning models by following very diverse approaches. In this context this work focuses on the automatic construction of features in supervised learning problems, which differs from the conventional selection of features in that new characteristics with enhanced predictive power are inferred from the original dataset. In particular this manuscript proposes a new iterative feature construction approach based on a self-learning meta-heuristic algorithm (Harmony Search) and a solution encoding strategy (correspondingly, Cartesian Genetic Programming) suited to represent combinations of features by means of constant-length solution vectors. The proposed feature construction algorithm, coined as Adaptive Cartesian Harmony Search (ACHS), incorporates modifications that allow exploiting the estimated predictive importance of intermediate solutions and, ultimately, attaining better convergence rate in its iterative learning procedure. The performance of the proposed ACHS scheme is assessed and compared to that rendered by the state of the art in a toy example and three practical use cases from the literature. The excellent performance figures obtained in these problems shed light on the widespread applicability of the proposed scheme to supervised learning with legacy datasets composed by already refined characteristics.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Predictive analytics are broadly conceived as the family of supervised machine learning models aimed at inferring unknown outcomes from a system based on a set of observed variables or features [1]. Albeit supervised learning models date back to several decades ago, predictive analytics have nowadays regained momentum by virtue of the *in crescendo* availability of data in most fields of knowledge. Hot topics such as Intelligent Systems [2] and Big Data [3,4] evince the increasing relevance of predictive modeling among different disciplines and the subsequent need for enhancing and innovating through all its compounding processing steps [5]: (1) data preparation and cleansing, with different strategies to impute missing and/or illegal data depending on their alphabet; (2) novelty/outlier detection; (3) feature processing, where the original dataset is processed/transformed/filtered so as to describe the essential features of the data and reduce the complexity of the subsequent predictive model; (4) model selection, where diverse alternatives have so far been reported in the literature characterized by different controlling parameters, training algorithms, discriminative capability and generalization properties; (5) model tuning; and (6) model performance assessment when predicting a set of unseen examples. It is only by thoroughly elaborating on each of the above steps that a good predictive model is generated.

This manuscript gravitates on the third processing step as enumerated above: feature processing. The literature has been specially profitable in this regard, with *de facto* classifications depending on the selective or constructive nature of the feature processing approach at hand. On one hand, *feature selection* schemes essentially select a subset of the original features by following strategies (filter, wrapper or embedded methods). Interestingly for the scope of this manuscript meta-heuristically empowered feature selection schemes have lately come into scene

* Corresponding author at: TECNALIA, P. Tecnologico Bizkaia, Ed. 700, 48160 Derio, Spain.
E-mail address: javier.delser@tecnalia.com (J. Del Ser).

http://dx.doi.org/10.1016/j.asoc.2016.09.049
1568-4946/© 2016 Elsevier B.V. All rights reserved.

Please cite this article in press as: A. Elola, et al., Hybridizing Cartesian Genetic Programming and Harmony Search for adaptive feature construction in supervised learning problems, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.09.049

in a diversity of scenarios [6–13] with particular emphasis in Energy applications [14,15] and Bioinformatics [16,17]. On the other hand, feature extraction/construction or dimensionality reduction algorithms transform the original dataset to a feature space of fewer dimensions, which can be done by resorting to elements from linear statistics [18] or newer findings in the field of non-linear manifold learning and low-dimensional embedding [19].

This research work focuses on this second category, specifically on the construction of features via wrapper methods. This class of methods are of paramount utility when dealing with legacy datasets, i.e. datasets whose compounding features result from raw information preprocessed through application-specific signal processing stages. In such situations there is no access to the original data from which such features were extracted, hence jeopardizing the adoption of embedded schemes with known potential in highly multidimensional datasets (e.g. deep learning). The scope is also placed on the readability of the constructed features, which not only is useful for assessing mathematical properties therefrom (e.g. trends, correlations), but also becomes a requirement for certain application scenarios where supervision by a higher-level entity and/or the preservation of privacy are crucial, such as the risk assessment in bank insurance, the diagnosis of diseases and the personalized prescription of medical treatments. From a technical perspective this sought explicitness for the constructed feature set can be provided by Evolutionary Programming [20], a branch of Evolutionary Computation that aims at iteratively refining *computer programs* based on a measure of their quality or fitness. In the context of mathematical programs, this term stands for a combination or function of different variables (features) based on an alphabet of operator functions (e.g. +, −, ×, ÷). Such programs can be represented as tree structures, which can be in turn evolved via evolutionary crossover and mutation processes towards regions of progressively higher optimality as measured by the fitness function at hand. When put in the context of feature construction, each evolved program represents a combination of features (i.e. a newly constructed feature), whereas the fitness function is given by the performance of the wrapped predictive model when trained with the evolved feature set. Indeed this has been the technical approach followed by a number of contributions by the research community where the good performance of Evolutionary Programming has been evinced in diverse practical applications of predictive modeling (see [21–30] and the comprehensive survey in [31]).

The work presented in this paper takes a step further in the state of the art in the above field by proposing a novel wrapper approach based on the combination of Cartesian Genetic Programming [32] and Harmony Search (hereafter denoted as HS, [33]). On the one hand, Cartesian Genetic Programming permits to encode (represent) programs by means of strings of integers, which numerically encode the operators that relate variables to each other, their connections to the set of input features and the resulting output features fed to the model. On the other hand, Harmony Search is a meta-heuristic solver that has been widely shown to outperform other bio-inspired optimization algorithms in many applications [34]. In this manuscript we propose to blend together these two techniques to yield a feature construction wrapper that in addition, exploits information about the predictive relevance of the produced feature set so as to enhance the convergence properties of the overall search process. The performance of the derived feature construction scheme is evaluated over four supervised learning problems – namely, the well-known `WINE` dataset, leaf-based plant classification (`LEAF`, [35]), classification of radar returns from the ionosphere (`IONOSPHERE`, [36]) and vehicle type recognition (`VTR`, [37]) – with results that dominate the best scores obtained to date. To the best of the authors' knowledge, this is the first contribution in the literature hybridizing Cartesian Genetic Programming with Harmony Search for feature construction in supervised learning.

The rest of the paper is structured as follows: Section 2 formally poses the construction of explicit features in supervised learning scenarios as a mathematical optimization problem. Next, Section 3 and subsections therein delves into the proposed algorithmic approach by outlining its overall working procedure and detailing each of its compounding modules. Experimental results over the four considered datasets are presented and discussed in Sections 4 and 5 and, finally, Section 6 ends the paper by drawing conclusions and sketching several lines of future research.

2. Feature construction as an optimization problem

Mathematically speaking a supervised learning problem departs from a set of available data instances $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^{N}$, with N denoting the number of instances or examples, x_n^d the d-th feature for example n and $D \doteq |\mathbf{x}_n|\ \forall n \in \{1, \ldots, N\}$ the number of features or *dimensionality*. Since we deal with supervised learning, samples in $\mathbf{X}$ are associated to a value of the target variable to be predicted, which are all collected in the label vector $\mathbf{y} \doteq \{y_n\}_{n=1}^{N}$. The goal of a supervised learning algorithm is to infer the pattern relating $\mathbf{x}_n$ to its corresponding label y_n. This can be accomplished by a model $M_{\boldsymbol{\theta}} : \mathcal{X}^D \mapsto \mathcal{Y}$ that maps a given data instance or sample $\mathbf{x}$ to its estimated target variable y. The model $M_{\boldsymbol{\theta}}(\cdot)$ can be constructed (trained, learned) based on a set of training examples $\{\mathbf{X}^{tr}, \mathbf{y}^{tr}\} \subset \{\mathbf{X}, \mathbf{y}\}$ and the parameters $\boldsymbol{\theta}$ of the model at hand. The remaining data samples $\mathbf{X}^{test} \doteq \mathbf{X} - \mathbf{X}^{tr}$ and their supervised labels are left out for testing the predictive performance $\Psi(\mathbf{y}^{test}; \mathbf{y}^{pred})$ of the model when processing unseen data. The design is hence to maximize this performance measure. To this end, a common practice is to randomly partition the training set into K folds, retain a subset for validating the model and use the remaining $K-1$ subsets as training data. By repeating this process K times a cross-validated estimate of model prediction performance can be computed, which is utilized for tuning the parameter configuration $\boldsymbol{\theta}$ of the model.

In this manuscript the maximization of the average performance of the model is approached by constructing a new feature set $\mathbf{X}' \doteq \{\mathbf{x}'_n\}_{n=1}^{N}$ with $D' \doteq |\mathbf{x}'_n|$, such that a model $M' : \mathcal{X}^{D'} \mapsto \mathcal{Y}$ can be trained to yield a better performance score than the same model when fed with the original set of features $\mathbf{X}$. As mentioned in Section 1 the scope of this work is placed on the construction of readable features based on the original data $\mathbf{X}$ and a set of explicit mathematical operators $\mathcal{F} \doteq \{f_1, \ldots, f_P\}$. Each sample $\mathbf{x}_n \in \mathbf{X}$ is mapped to the space spanned by a set of constructed features $\{x^{d,\prime}\}_{d=1}^{D'}$, each expressed as an combination of the original feature set $\{x^d\}_{d=1}^{D}$ through a operator subset $\mathcal{F}^{d,\prime} \subseteq \mathcal{F}$. For instance, if $\mathcal{F} = \{+, -, \times, \div, \cos, \sin, \exp\}$ and $D = 7$, examples of constructed features with $D' = 3$ could be given by

$$x^{1,\prime} = \cos(x^1 + x^2) + x^3 \times \exp(x^5 - x^2) \quad (\mathcal{F}^{1,\prime} = \{+, -, \times, \cos, \exp\}), \tag{1}$$

$$x^{2,\prime} = \exp(x^3 - \sin(x^5)) \quad (\mathcal{F}^{2,\prime} = \{-, \cos, \sin\}), \tag{2}$$

$$x^{3,\prime} = \cos(x^2 \times (x^1 - x^7)/x^4) \quad (\mathcal{F}^{3,\prime} = \{-, \times, \div, \cos\}). \tag{3}$$

It should be clear that any given operator subset $\mathcal{F}^{1,\prime}$ can result in different expressions, as the original features involved and their relative order in the expression may vary without impacting on the operator subset at hand. This is the reason why such combinations (*programs*) are often represented as trees, with variables (original features) located at leaf nodes and operators at intermediate nodes. The transformed dataset $\mathbf{X}'$ is next used for training and tuning a

Please cite this article in press as: A. Elola, et al., Hybridizing Cartesian Genetic Programming and Harmony Search for adaptive feature construction in supervised learning problems, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.09.049

Download English Version:

https://daneshyari.com/en/article/4963500

Download Persian Version:

https://daneshyari.com/article/4963500

Daneshyari.com