Expert Systems with Applications 38 (2011) 12930–12938

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Combining functional networks and sensitivity analysis as wrapper method for feature selection

Noelia Sánchez-Maroño *, Amparo Alonso-Betanzos

Laboratory for Research and Development in Artificial Intelligence (LIDIA), Department of Computer Science, Faculty of Informatics, University of A Coruña, Campus de Elviña s/n, 15071 A Coruña, Spain

ARTICLE INFO

Keywords:
Feature selection
Wrapper methods
Machine learning
Neural networks

ABSTRACT

In this paper, a new wrapper method for feature selection, namely IAFN-FS (Incremental ANalysis Of VAriance and Functional Networks for Feature Selection) is presented. The method uses as induction algorithm the AFN (ANOVA and Functional Networks) learning method; follows a backward non-sequential strategy from the complete set of features (thus allowing to discard several variables in one step, and so reducing computational time); and is able to consider "multivariate" relations between features. An important characteristic of the method is that it permits the user the interpretation of the results obtained, because the relevance of each feature selected or rejected is given in terms of its variance. IAFN-FS is applied to several benchmark real-world classification data sets showing adequate performance results. Also, a comparison with the results obtained by other wrapper methods is carried out, showing that the proposed method obtains better performance results in average.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

Feature extraction addresses the problem of finding the most compact and informative set of features of a given problem, to improve the efficiency or data storage or processing. The problem of feature extraction is decomposed in two steps: feature construction and feature selection. Feature construction methods complement the human expertise to convert "raw" data into a set of useful features. It may be considered a preprocessing transformation that may include: standardization, normalization, discretization, signal enhancement, extraction of local features, etc. Some construction methods do not alter the space dimensionality, while others enlarge it, reduce it or can act in either direction. But one should be aware of not losing information at the feature construction stage. Guyon and Elisseeff (2003) argued that it is always better to err on the side of being too inclusive rather than risking to discard useful information. Adding many features seems reasonable but it comes at a price: it increases the dimensionality of the patterns and thereby immerses the relevant information into a sea of possibly irrelevant, noisy or redundant features. Feature selection is the process in which the number of initial features is reduced and a subset of them that retain enough information for obtaining good, or even better, performance results is selected.

Feature selection is primarily performed to select relevant and informative features, but it can have other motivations, including:

- *general data reduction*, to limit storage requirements and increase algorithm speed,
- *feature set reduction*, to save resources in the next round of data collection or during utilization,
- *performance improvement*, to gain in predictive accuracy,
- *data understanding*, to gain knowledge about the process that generated the data or simply visualize the data,
- *diminish temporal resources*, providing faster and most-effective models.

There are several ways of classifying feature selection methods. The most common taxonomy is classifying them in wrapper and filter methods (Kohavi & John, 1997). Filter methods rely on general characteristics of the training data in order to provide a complete order of the features using a relevance index, without optimizing the performance of a predictor (Guyon & Elisseeff, 2003). Wrapper methods use a learning algorithm, along with a statistical re-sampling technique such as cross-validation to score subsets of features according to their predictive value (Kohavi & John, 1997). Wrapper methods are usually more expensive computationally, but also result in better performance (Blum & Langley, 1997).

Another important classification of feature selection methods is the distinction between univariate and multivariate algorithms. At this respect, many filters (ANOVA, t-test, Kolmogorov–Smirnoff test, etc.), and also some wrappers (Setiono & Liu, 1997; Yu & Chen, 2005; Yu et al., 2005) rank/select the features according to some consistency or redundancy measure that assumes feature independence. This assumption has some limitations:

* Corresponding author.
E-mail address: noelia.sanchez@udc.es (N. Sánchez-Maroño).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.04.089

- features that are not individually relevant may become relevant in the context of others;
- features that are individually relevant may not all be useful because of possible redundancies.

Some existing methods do not effectively capitalize on the nature of multivariate associations between classes. This is especially important for data set with overlapping patterns which cannot be efficiently separated based on distance measure or decision boundaries. So-called "multivariate" methods take into account feature dependencies (Guyon, Weston, Barnhill, & Vapnik, 2002; Lai, Reinders, & Wessels, 2006). Multivariate methods potentially achieve better results because they do not make simplifying assumptions of variable/feature independence. One justification of multivariate methods is that features that are individually irrelevant may become relevant when used in combination (example XOR problem). Another justification of multivariate methods is that they take into account feature redundancy and yield more compact subsets of features.

Wrappers use different strategies to perform a search through the space of feature subsets. The size of this space is exponential in the number of features, and thus an exhaustive search is intractable in most real situations. Mainly, search strategies can be backward or forward, although there are some new methods that start at an intermediate point. In a forward selection (Bo & Jonassen, 2002) method one starts with an empty set of features and progressively adds features yielding to the improvement of a performance index. In a backward elimination procedure one starts with all the features and progressively eliminates the least useful ones. Both procedures are reasonably fast and robust against overfitting, and both provide nested feature subsets. However, they may lead to different subsets and, depending on the application and the objectives, one approach may be preferred over the other one. Backward elimination procedures may yield better performance but at the expense of possibly larger feature sets. However, if the feature set selected is reduced too much, performance of the method may diminish drastically.

Although most recent applications of feature selection methods are in very high dimensional spaces with a comparatively reduced number of samples (Guyon & Elisseeff, 2003), there are still several issues that deserve attention in wrappers, such as the stability of the selection method and the use of knowledge to guide the search. In this paper, a wrapper algorithm for feature selection is presented. The method, called IAFN-FS (Incremental ANOVA and Functional Networks Feature Selection), is based on functional networks (Castillo, Cobo, Gutiérrez, & Pruneda, 1998) and analysis of variance decomposition. Functional networks (FN) are a generalization of neural networks that bring together domain knowledge, to determine the structure of the problem, and data, to estimate the unknown functional neurons (Castillo et al., 1998). So, knowledge can be used to guide the feature selection. The IAFN-FS method follows a backward strategy and it considers "multivariate" relations between features. Besides, IAFN-FS presents several other advantages, such as that it allows to discard several variables in just one step. Another important advantage of the method is that it permits the user the interpretation of the results obtained, because the relevance of each feature selected or rejected is given in terms of variance. The proposed method is applied to real-world classification data sets of the UCI Learning repository, and its performance results are compared to those obtained by other wrapper methods. As it will be shown, IAFN-FS exhibits good accuracy results while maintaining a reduced set of variables.

The paper is structured as follows: Section 2 gives a brief introduction to Functional Networks and describes the ANOVA decomposition including how to obtain the sensitivity indices. Section 3 describes the proposed wrapper method for feature selection. Section 4 presents the results achieved over different benchmark data sets and a comparative study is carried out; and finally, Section 5 is devoted to conclusions and future work.

2. The ANOVA and Functional Networks method

The ANOVA and Functional Networks (AFN) method is based on ANOVA (*ANalysis Of VAriance*) decomposition and functional networks (FN). The ANOVA decomposition permits global sensitivity indices to be obtained for each variable and for each interaction between variables (Castillo, Sánchez-Maroño, Alonso-Betanzos, & Castillo, 2007). The topology of a functional network is derived from this information, and low and high order interactions among variables are easily determined using ANOVA. Therefore, the AFN method permits a simplified topology for a functional network to be learnt from the sample data available.

Any square integrable function $f(x_1,x_2,\ldots,x_n)$ can always be written as the sum of the 2^n orthogonal summands Sobol (2001):

$$\begin{aligned} y &= f(x_1,\ldots,x_n) \\ &= f_0 + \sum_{i_1=1}^{n} f_{i_1}(x_{i_1}) + \sum_{i_1=1}^{n-1}\sum_{i_2=i_1+1}^{n} f_{i_1 i_2}(x_{i_1},x_{i_2}) \\ &\quad + \sum_{i_1=1}^{n-2}\sum_{i_2=i_1+1}^{n-1}\sum_{i_3=i_2+1}^{n} f_{i_1,i_2,i_3}(x_{i_1},x_{i_2},x_{i_3}) + \cdots \\ &\quad + \sum_{i_1=1}^{1}\sum_{i_2=2}^{2}\cdots\sum_{i_n=n}^{n} f_{i_1 i_2\ldots i_n}(x_{i_1},x_{i_2},\ldots,x_{i_n}), \end{aligned} \tag{1}$$

where the term f_0 is a constant and corresponds to the function with no arguments. Therefore, the AFN method estimates the desired output y of a given problem by approximating each functional component $f_{i_1 i_2\ldots i_k}$ in (1) as:

$$f_{i_1 i_2\ldots i_k}(x_{i_1},x_{i_2},\ldots,x_{i_k}) \approx \sum_{j=1}^{k_{i_1 i_2\ldots i_k}} c_{i_1 i_2\ldots i_k;j}\, p_{i_1 i_2\ldots i_k;j}(x_{i_1},x_{i_2},\ldots,x_{i_k}), \tag{2}$$

where $c_{i_1 i_2\ldots i_k;j}$ are parameters to be estimated and $p_{i_1 i_2\ldots i_k;j}$ is a set of orthonormalized basis functions. There exists several alternatives to choose those functions (Castillo et al., 2007); one possibility consists of using one of the families of univariate orthogonal functions, for example, Legendre polynomials, forming tensor products with them and selecting a subset of them.

Then, the $c_{i_1 i_2\ldots i_k;j}$ parameters are learnt by solving an optimization problem, such as:

$$\textit{Minimize} \quad J = \sum_{s=1}^{m} \epsilon_s^2 = \sum_{s=1}^{m} (y_s - \hat{y}_s)^2 \tag{3}$$

being m the number of available samples, y_s the desired output for the sample s and $\hat{y}_s$ the estimated output obtained by:

$$\begin{aligned} \hat{y}_s &= f_0 + \sum_{i_1=1}^{n}\sum_{j=1}^{k_{i_1}} c_{i_1;j}\, p_{i_1;j}(x_{i_1 s}) + \sum_{i_1=1}^{n-1}\sum_{i_2=i_1+1}^{n}\sum_{j=1}^{k_{i_1 i_2}} c_{i_1;j}\, p_{i_1 i_2;j}(x_{i_1 s},x_{i_2 s}) + \cdots \\ &\quad + \sum_{i_1=1}^{1}\sum_{i_2=2}^{2}\cdots\sum_{i_n=n}^{n}\sum_{j=1}^{k_{i_1 i_2\ldots i_n}} c_{i_1 i_2\ldots i_n;j}\, p_{i_1 i_2\ldots i_n}(x_{i_1 s},x_{i_2 s},\ldots,x_{i_n s}). \end{aligned} \tag{4}$$

As the method uses a set of orthonormalized functions (2), once the parameters $c_{i_1 i_2\ldots i_k}$ are learnt by solving (3), global sensitivity indices (*GSI*) for each functional component can be directly derived as:

$$GSI_{i_1 i_2\ldots i_k} = \sum_{j=1}^{k_{i_1 i_2\ldots i_k}} c^2_{i_1 i_2\ldots i_k;j}; \quad \forall (i_1 i_2 \ldots i_k). \tag{5}$$

دریافت فوری ← متن کامل مقاله

ISIArticles
مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات