Neurocomputing 74 (2010) 178–190

Contents lists available at ScienceDirect

Neurocomputing

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

Rule extraction from support vector machines: A review

Nahla Barakat[a,*], Andrew P. Bradley[b]

[a] Department of Applied Information Technology, German University of Technology in Oman, Oman
[b] School of Information Technology and Electrical Engineering (ITEE), The University of Queensland, St. Lucia, QLD 4072, Australia

ARTICLE INFO

Article history:
Received 20 March 2009
Received in revised form
22 January 2010
Accepted 22 February 2010
Communicated by S. Mitra
Available online 27 March 2010

Keywords:
Machine learning
Data mining
Knowledge discovery
Information extraction
Pattern recognition applications
SVMs

ABSTRACT

Over the last decade, support vector machine classifiers (SVMs) have demonstrated superior generalization performance to many other classification techniques in a variety of application areas. However, SVMs have an inability to provide an explanation, or comprehensible justification, for the solutions they reach. It has been shown that the '*black-box*' nature of techniques like artificial neural networks (ANNs) is one of the main obstacles impeding their practical application. Therefore, techniques for rule extraction from ANNs, and recently from SVMs, were introduced to ameliorate this problem and aid in the explanation of their classification decisions. In this paper, we conduct a formal review of the area of rule extraction from SVMs. The review provides a historical perspective for this area of research and conceptually groups and analyzes the various techniques. In particular, we propose two alternative groupings; the first is based on the SVM (model) components utilized for rule extraction, while the second is based on the rule extraction approach. The aim is to provide a better understanding of the topic in addition to summarizing the main features of individual algorithms. The analysis is then followed by a comparative evaluation of the algorithms' salient features and relative performance as measured by a number of metrics. It is concluded that there is no one algorithm that can be favored in general. However, methods that are kernel independent, produce the most comprehensible rule set and have the highest fidelity to the SVM should be preferred. In addition, a specific method can be preferred if the context of the requirements of a specific application, so that appropriate tradeoffs may be made. The paper concludes by highlighting potential research directions such as the need for rule extraction methods in the case of SVM incremental and active learning and other application domains, where special types of SVMs are utilized.

1. Introduction

Over the last three decades, data mining and machine learning techniques have been remarkably successful in extracting interesting knowledge and hidden *patterns* from the ever growing databases. In this context, Frawley et al. [1] provides the following definition for the term 'pattern':

> Given a set of facts (data) F, a Language L, and some measure of certainty C, a pattern S is a statement S in L that describes relationships among a subset Fs of F with certainty C, such that S is simpler (in some sense) than the enumeration of all facts in Fs. [1]

Support vector machines and ANNs are among the most successful machine learning techniques applied in this area [2–10]. However, the superior performance of SVMs and ANNs comes with a significant drawback: the language, L, in which they learn patterns from data is not comprehensible to humans; hence they do not provide an explanation or a comprehensible justification for the knowledge they learn. This has been shown to be one of the main obstacles impeding their practical application [11–13].

1.1. Rule extraction from SVMs: the motivation

Explanation has been one of the most important topics that has attracted researchers' attention, not only in philosophy, where several theories of explanation have been proposed [14], but also in the area of artificial intelligence. In particular, since the early introduction of expert systems, and continuing with case-based reasoning, explanation is still considered one of the most important criteria that influence the acceptance of these techniques by end users [15–18]. Similarly, it has also been shown that the explanation of a classification decision is a crucial requirement for the acceptance of *black-box* models by end users, especially in areas like medical diagnosis and prognosis [11–13,19–22]. Therefore, several methods have been introduced

[*] Corresponding author.
E-mail addresses: n.barakat@uq.edu.au, nahlabarakat@gmail.com (N. Barakat), bradley@itee.uq.edu.au (A.P. Bradley).

0925-2312/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2010.02.016

for rule extraction from ANNs [23], and more recently from SVMs [22,24–37].

One could argue that rule extraction from SVMs is a narrow area of research, which is not expected to grow in the future, given that to date rule extraction from ANNs has not been particularly successful. However, over the past 6 years, this topic has attracted an ever increasing number of researchers from a variety of domains [13,21,22,36–39]. Furthermore, there is a significantly growing trend in the usage of SVMs and continuing development of new types of kernels e.g., [40–44]. Explanation can also help in tasks like image annotation and document mapping, where SVMs have been widely used in building and mapping ontologies for the semantic web [45–47]. Discovering concept drift in active learning by SVMs [48] is also a significant domain which can benefit from the comprehensibility provided by rule extraction algorithms.

1.2. *Rule extraction from SVMs: the problem*

The task of rule extraction from SVMs is to devise rules from the model (SVM) rather than directly from the data. Therefore, an explanation of the patterns learned and embedded in their structure (model support vectors (SVs) and their associated parameters) is revealed and provided to the end users in a comprehensible form.

1.3. *Overview*

In this paper we conduct the first formal review for the area of rule extraction from SVMs. The survey conceptually groups and analyzes the literature at a higher level of abstraction, which outlines differing fundamental approaches to the topic, rather than summarizing the main features of individual algorithms, as in [37,49]. This is then followed by a comparative evaluation of the methods salient features and a benchmark of results published to date.

The paper is organized as follows: Section 2 provides a brief introduction to SVMs followed by the review of rule extraction algorithms in Section 3. Section 4 briefly summarizes the classification of rule extraction from ANNs, while similarities between ANNs and SVMs are highlighted in Section 5. In Section 6, we discuss and compare the main features of different algorithm. Some potential directions for future research are also highlighted in Section 7, followed by a summary and conclusions in Section 8. Abstract representations for individual algorithms' main modules are also provided in Appendix A.

2. Support vector machine classifiers: an overview

Support vector machines belong to the maximum margin classifier family and are based on the principle of structural risk minimization (SRM). The SRM principle aims to select a hypothesis function with low capacity from a nested sequence of functions that simultaneously minimizes both the true error rate (classification error on unseen examples), and empirical (training set) error rate [50].

Given the training data set $\{x_i,y_i\}$, $i=1,\ldots,l$, $y_i\in\{-1,1\}$, $x_i\in\Re^n$ (where x_i is an input feature vector of dimensionality n, and y_i the corresponding class label), an SVM finds the optimal separating hyper-plane with the largest margin [50]. Eqs. (1) and (2) represent the separating hyper-planes in the case of separable data sets:

$$x_i w+b\geq +1 \quad \text{for } y_i=+1 \tag{1}$$

$$x_i w+b\leq -1 \quad \text{for } y_i=-1 \tag{2}$$

For the linearly separable case, finding a maximum separating margin $d(d=2/(\|w\|))$ is a constrained optimization problem represented by

$$\text{minimize } \tfrac{1}{2}\|w\|^2 \quad \text{subject to } y_i(wx_i+b)\geq 1 \tag{3}$$

2.1. *Soft margin linear SVMs*

The soft margin hyper-plane can be obtained by relaxing the optimization problem in (3), through the introduction of ξ_i (positive slack variable) [51] to allow for errors on the training set. Including ξ_i, Eqs. (1) and (2) are modified as follows:

$$x_i w+b\geq +1-\xi_i \quad \text{for } y_i=+1 \tag{4}$$

$$x_i w+b\leq -1+\xi_i \quad \text{for } y_i=-1, \quad \xi_i\geq 0 \quad \forall_i \tag{5}$$

Introducing the regularization parameter C controls the tradeoff between the margin maximization and the training error [51] the optimization problem in (3) is then modified as follows:

$$\text{minimize } \frac{1}{2}\|w\|^2+C\sum_{i=1}^{l}\xi_i \tag{6}$$

Subject to $y_i(wx_i+b)\geq 1-\xi_i, \quad \xi_i\geq 0$

where $\sum_{i=1}^{l}\xi_i$ the upper bound on the training error and C is a regularization parameter [51]. The optimization problem in (6) is called the primal problem. To find an easier solution to this problem it is transformed to its dual [51], by introducing α, the Lagrange multipliers (dual variables) which are the fundamental unknowns in the dual optimization problem [51]. Hence the problem in (6) becomes

$$\text{maximize } w(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i y_i\alpha_j y_j\langle x_i,x_j\rangle$$

$$\text{such that } C\geq\alpha_i\geq 0 \quad \forall_i, \quad \sum_{i=1}^{l}\alpha_i y_i=0 \tag{7}$$

Solving for α, training examples with non-zero α are called support vectors (SVs) and the hyper-plane is being completely defined by the SVs only. As shown in (7), the C parameter constitutes the upper bound on α_i. As a result, three types of support vectors are characterized based on the values of α_i *and* ξ_i as follows:

- Support vectors with $\alpha_i<C$, those are the SVs which lie outside the margin d and will be correctly classified.
- Support vectors with $\alpha_i=C$, and $\xi_i>1$, these SVs lies at the wrong side of the hyper-plane, and represent errors (will be misclassified points by the hyper-plane).
- Support vectors with $\alpha_i=C$, and $0<\xi_i\leq 1$, these SVs will lie inside the margin d (i.e., closer than $1/(\|w\|)$ from the hyper-plane).

2.2. *Non-linear SVMs*

In the case of non-linear models, SVMs use kernel functions[1] to map non-linearly separable data in the input space $x_i\in\Re^n$ to be

[1] A function that enables direct computation of the inner product $\langle\Phi(x_i)\Phi(x)\rangle$ in feature space as a function of original input points, where Φ is a non-linear map from input space to a feature space (inner product) [51].