Applied Soft Computing 31 (2015) 185–195

Contents lists available at ScienceDirect

Applied Soft Computing

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

The use of vicinal-risk minimization for training decision trees

Yilong Cao*, Peter I. Rockett

Department of Electronic and Electrical Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, UK

ARTICLE INFO

Article history:
Received 4 July 2014
Received in revised form 6 February 2015
Accepted 25 February 2015
Available online 16 March 2015

Keywords:
Decision trees
Vicinal-risk minimization
Decision trees
Classification

ABSTRACT

We propose the use of Vapnik's vicinal risk minimization (VRM) for training decision trees to approximately maximize decision margins. We implement VRM by propagating uncertainties in the input attributes into the labeling decisions. In this way, we perform a global regularization over the decision tree structure. During a training phase, a decision tree is constructed to minimize the total probability of misclassifying the labeled training examples, a process which approximately maximizes the margins of the resulting classifier. We perform the necessary minimization using an appropriate meta-heuristic (genetic programming) and present results over a range of synthetic and benchmark real datasets. We demonstrate the statistical superiority of VRM training over conventional empirical risk minimization (ERM) and the well-known C4.5 algorithm, for a range of synthetic and real datasets. We also conclude that there is no statistical difference between trees trained by ERM and using C4.5. Training with VRM is shown to be more stable and repeatable than by ERM.

1. Introduction

Decision trees are a popular and widely used classification paradigm, largely due the ease with which the trained classifiers can be interpreted. Unfortunately, it has been shown that constructing an optimal decision tree (DT) is an NP-complete problem [1] and so a number of greedy heuristics have been proposed over the years, probably the foremost being the C4.5 algorithm [2], which seek sequentially to maximize information gain at each node in the tree. Typically with C4.5, a DT is trained to the point of over-fitting and then pruned with a second heuristic to improve generalization.

As an alternative training method for DTs, a wide range of meta-heuristics have been explored; see [3] for a recent survey. In this work we have used genetic programming (GP) since, as a population-based stochastic search method, GP has been shown to be well-suited to finding approximate solutions to NP-hard problems, such as DT training. Koza [4] seems to have been the first to propose GP for this purpose although see [5] for a comprehensive review.

Previous work on the evolutionary training of DTs has clearly established that credible decision trees can be induced by minimizing *empirical risk* [6] (i.e., misclassification error, or the fraction of patterns incorrectly classified) over some training set. Notwithstanding, there is a dearth of work which *quantitatively* compares GP results with conventional tree induction methods, such as C4.5. A number of authors—for example, [7]—have noted that GP-induced trees can give smaller misclassification errors compared to C4.5 but smaller errors do not necessarily denote any statistical significance. Since conventional tree induction methods are greedy algorithms, one would expect sub-optimal performance. The theoretical advantage of meta-heuristic methods is that they have been demonstrated to provide good—although not necessarily optimal—solutions to NP-hard problems with acceptable computing times. Evolutionary methods could therefore be expected to out-perform greedy methods, in general.

The principal and novel contribution of this paper is to introduce a new risk functional, vicinal risk, for training DTs, which addresses the challenging issue of how to regularize tree structures. To support this, we present statistically founded comparison between trees induced using the new risk and conventional methods. Since, we believe, minimizing new risk functional can only be carried-out using an appropriate meta-heuristic, this paper reports the application of soft computing to an important problem in machine learning.

In general, training in machine learning is ill-posed [6] and empirical risk minimization (ERM) does not necessarily produce best generalization over an unseen test set, a problem which is exacerbated by small datasets; it is this 'small data' scenario we explicitly address in this paper. The deficiencies of ERM are illustrated in Fig. 1 for the trivial case of classifying linearly separable patterns with a plane. Reduction of the ER to zero can be achieved by any of the infinite number of hyperplanes passing between the two groups of patterns. In particular, hyperplane 'B'—although minimizing the risk to zero—lacks the robustness to cope with even small noise perturbations of the pattern attributes. It is obvious that hyperplane 'A' will deliver the greatest 'margin' against noise. Since empirical risk minimization (ERM) does not necessarily produce acceptable margins, this motivates us to investigate a superior risk functional and apply it to decision trees.

The principal contribution of this paper is to report the induction of decision trees using *vicinal risk minimization* (VRM) [8] which displays significantly greater stability than ERM. Since this new risk is a continuous function, it is able to discriminate between the competing decision surfaces in Fig. 1 which (discrete) ERM cannot distinguish. We make statistical comparisons with decision trees induced with VRM and conventional ERM using the same GP method. As an underpinning baseline, we compare the above results with trees induced using the popular, deterministic C4.5 algorithm. We demonstrate that VRM is able to produce decision trees with superior generalization performance compared to C4.5, and GP-trained trees which minimize ER.

In Section 2 we describe the adaptation of VRM to decision trees. In Section 3 we discuss related work on decision trees and their training using genetic programming (GP); we review genetic programming and its single and multiple objective variants.

* Corresponding author. Tel.: +44 114 222 5589.
E-mail addresses: naodai.mmx@gmail.com (Y. Cao), p.rockett@sheffield.ac.uk (P.I. Rockett).

http://dx.doi.org/10.1016/j.asoc.2015.02.043
1568-4946/© 2015 Elsevier B.V. All rights reserved.

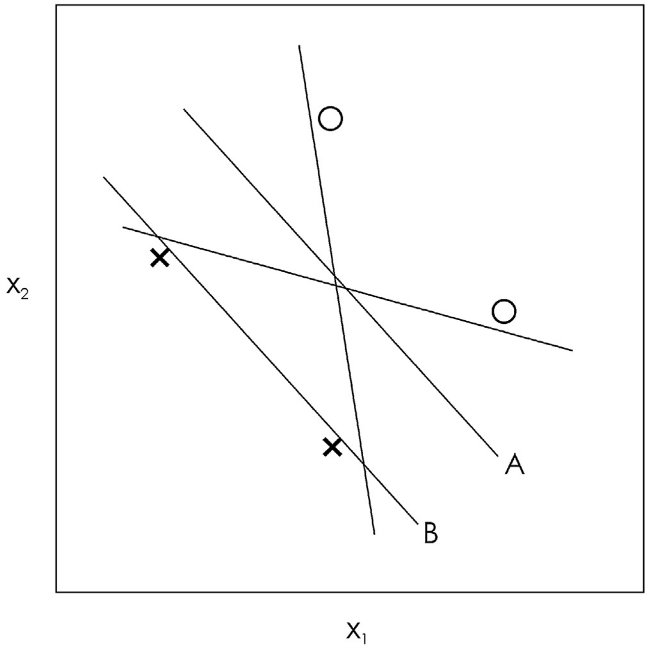

Fig. 1. Illustration, for the simple case of classifying linearly separable patterns, of the deficiency of minimizing empirical risk. The crosses and circles represent patterns of differing classes.

We outline our experimental methodology in Section 4, and present experimental results in Section 5. We offer further insights into the application of VRM to decision trees in Section 6; Section 7 concludes the paper.

2. Vicinal risk

Starting with the development of vicinal risk for a conventional scoring classifier given by Vapnik [8], for some set of ℓ training data, $\mathcal{D}=\{\mathbf{x}_1\to y_1,\mathbf{x}_2\to y_2,\ldots,\mathbf{x}_\ell\to y_\ell\}$ drawn from a data distribution $P(\mathbf{x}, y)$, where $\mathbf{x}_i\in\mathbb{R}^N$ and $y\in\{-1,+1\}$, the task of training a scoring classifier is to select some discriminant function $f(\mathbf{x})\to y$. We desire to select the $f(\mathbf{x})$ which minimizes the expected risk $R(f)$, which will ensure optimum generalization over future unseen examples drawn from $P(\mathbf{x}, y)$ where:

$$R(f)=\int L[f(\mathbf{x}),y]dP(\mathbf{x},y) \tag{1}$$

and L is some loss function. Unfortunately, $P(\mathbf{x}, y)$ is not known in practice and so the conventional approach has been to approximate $P(\mathbf{x}, y)$ using the set of samples $\{\mathbf{x}_i, y_i\}i\in[1,\ldots,\ell]$:

$$P(\mathbf{x},y)\approx\frac{1}{\ell}\sum_{i=1}^{\ell}\delta(\mathbf{x}-\mathbf{x}_i) \tag{2}$$

where δ is the Dirac delta function, and to minimize the *empirical risk*, R_{emp} (i.e., the expected 0/1 loss) over the training set. We can conveniently take the loss function to be [8]:

$$L[f(\mathbf{x}),y]=H[-yf(\mathbf{x})] \tag{3}$$

where H is the Heaviside step function. Thus for **x**-values which would give rise to a misclassification, (3) is unity; conversely, for **x**-values which yield correct classification, the loss is zero. Thus, empirical risk, R_{emp} is defined as:

$$R_{emp}(f)=\frac{1}{\ell}\sum_{i=1}^{\ell}H[-y_if(\mathrm{x}_\mathrm{i})] \tag{4}$$

As is clear from Section 1, the fundamental shortcoming of the 0/1 loss is due to its discrete nature, in particular, that a pattern is either classified correctly, in which case it contributes zero to the cumulative loss, or the pattern is misclassified and so contributes unity to the loss. Crucially, no account is taken of the *margin* by which a pattern is misclassified (or indeed, correctly classified). A misclassified pattern which is just the wrong side of a decision surface is weighted equally with a pattern that is a very large distance from the decision surface; intuitively, the latter case should be treated as more serious than the former. As a logical consequence, a pattern's distance from the decision surface should weight its contribution to the loss.

Vapnik [8] has motivated *vicinal risk* by assuming that the (unknown) data distribution is locally 'smooth' in which case $P(\mathbf{x}, y)$ can be approximated by placing a *vicinity function* on each training datum—this process can be thought of as either resampling or, equivalently, interpolating $\mathcal{D}$. Since the shortcomings of 0/1 loss are due to its discrete nature, smoothing the training set will have the effect of stabilizing the training process. Vapnik [8] described two possible types of vicinity functions, *hard* and *soft*. Hard vicinity functions have an abrupt cutoff at some distance from a training datum—under a 2-norm, this would be a ball or hypersphere centered on each datum. Whereas a hard vicinity function has a constant, non-zero value up to the cutoff distance and zero beyond, a soft vicinity function, such as a Gaussian kernel, typically has a peak value at the training datum and a monotonically reducing value with increasing distance from the datum. Entirely equivalently, placing a kernel over each training datum can be viewed as approximating $P(\mathbf{x}, y)$ using a Parzen windows density estimator [9,10] for which a Gaussian kernel is a natural choice. Here we develop the soft vicinity function approach because: (i) it is more tolerant of the setting of scale of the kernel and (ii) there is a technical requirement with hard vicinity functions that they do not overlap in pattern space [8].

Taking the loss function given in (3), analogous to minimizing (1), we wish to select the f which minimizes the vicinal risk, R_{VR} which is the expectation of (3) over the data distribution. Writing this functional in modified form from that given by Vapnik [8]:

$$R_{\mathrm{VR}}(f)=\int L[f(\mathbf{x}),y]dP(\mathbf{x},y) \tag{5}$$

$$\approx\frac{1}{\ell}\sum_{i=1}^{\ell}\int H[-y_if(\mathbf{x})]G(\mathbf{x}|\mathbf{x}_i,\sigma_i^2)d\mathbf{x} \tag{6}$$

where $G()$ is the Gaussian kernel of variance σ_i^2 placed on the ith datum. Here $P(\mathbf{x}, y)$ is approximated by the Parzen windows estimate of a sum of Gaussians. The integral within (6) has a straightforward interpretation as the hypervolume, in the N-dimensional pattern space, of the portion of the ith kernel which falls on the 'wrong' side of the decision surface and would hence give rise to misclassification. A number of properties of vicinal risk minimization (VRM) is apparent:

- Under VRM, we seek to minimize a continuous function (6), thereby removing the problem with 0/1 loss of being discrete. Patterns contribute to the loss depending on their distance from the decision surface, or more strictly, the hypervolume of the kernel function falling on the 'wrong' side of the decision surface. It is clear that correctly-classified patterns a long way from the decision surface will make a very small contribution to the loss and will hence have a minimal influence on the placement of the decision surface—this is highly desirable since only data in the vicinity of the decision surface run the risk of misclassification and should 'negotiate' the location of the decision surface.