Pattern Recognition Letters 32 (2011) 2136–2142

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

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

Risk function estimation for subproblems in a hierarchical classifier ☆

I.T. Podolak *, A. Roman

Institute of Computer Science, Jagiellonian University, Łojasiewicza 6, 30-348 Cracow, Poland

ARTICLE INFO

Article history:
Received 19 November 2010
Available online 7 September 2011
Communicated by J.A. Robinson.

Keywords:
Classification
Ensemble methods
Hierarchical classifier
Problem partitioning
Clustering algorithms
Risk estimation

ABSTRACT

One of the solutions to the classification problem are the ensemble methods, in particular a hierarchical approach. This method bases on dynamically splitting the original problem during training into smaller subproblems which should be easier to train. Then the answers are combined together to obtain the final classification. The main problem here is how to divide (cluster) the original problem to obtain best possible accuracy expressed in terms of risk function value. The exact value for a given clustering is known only after the whole training process. In this paper we propose the risk estimation method based on the analysis of the root classifier. This makes it possible to evaluate the risks for all subproblems without any training of children classifiers. Together with some earlier theoretical results on hierarchical approach, we show how to use the proposed method to evaluate the risk for the whole ensemble. A variant, which uses a genetic algorithm (GA), is proposed. We compare this method with an earlier one, based on the Bayes law. We show that the subproblem risk evaluation is highly correlated with the true risk, and that the Bayes/GA approaches give hierarchical classifiers which are superior to single ones. Our method works for any classifier which returns a class probability vector for a given example.

1. Introduction

Let $\mathcal{D} = \{(x_i, c_i)\}_{i=1}^{N}$ be a finite training set, where $x_i \in X$ are attribute vectors and c_i are target values from a finite set of K classes $\mathcal{C} = \{C_1, \ldots, C_K\}$. A classification problem is to find a hypothesis $f : X \to \mathcal{C}$, such that $f(x_i) = c_i + \epsilon$, for some small ϵ, i.e. generalizes well the data given in $\mathcal{D}$.

The hierarchical classifier (HC) was introduced earlier by one of the authors (Podolak, 2008; Podolak and Bartocha, 2009), as a classification model. HC builds a tree-like structure with simple classifiers in all nodes, which are to solve the problem given with a training set $\mathcal{D}$. The training algorithm is recursive: first a simple classifier Cl is built to solve the problem given with $\mathcal{D}$. The resulting Cl does not need to have a very low error, actually it is only supposed to be *weak* (Schapire, 1990; Schapire and Singer, 1999). In order to strengthen the final accuracy, the problem is split into a set of subproblems. By a subproblem $\mathcal{I} \subset \{1, \ldots, K\}$ we understand the task of finding a hypothesis $f_{\mathcal{I}} : X \to \mathcal{C}_{\mathcal{I}}$, where $\mathcal{C}_{\mathcal{I}} \subsetneq \mathcal{C}$, such that $k \in \mathcal{I}$ iff $C_k \in \mathcal{C}_{\mathcal{I}}$. To denote a subproblem we shall use $\mathcal{I}$ and $\mathcal{C}_{\mathcal{I}}$ interchangeably. Each subproblem $\mathcal{I}$ is solved with a new classifier which returns a class probability vector for classes from $\mathcal{C}_{\mathcal{I}}$. An important feature of the HC model is that subproblems are constructed by recursive partitioning of $\mathcal{C}$ into $\mathcal{C}_{\mathcal{I}}$ class subsets that may *overlap*, strengthening the whole model.

All classifiers in the tree are *weak*. For a K-class problem, a weak classifier has the accuracy only high enough so that the probability of the true class to have the highest activation is greater than $1/K$ (Podolak and Roman, 2009). Thanks to that the individual node training is computationally cheap and effective.

The accuracy of HC depends on both the node classifiers accuracy and the subproblems structure – this is the first top-down stage, see Fig. 1. An input pattern x is first classified by the root classifier Cl^0, then passed on to children classifiers, each of which is responsible for building a hypothesis for a different subproblem. After reaching the leaf nodes the activations are passed back to the root in a bottom-up way. Then, the final classification is computed using one of the evaluation methods. In this paper we use the weighted sum of the children answers:

$$Cl^{HC}(x) = \sum_{l} w_l(x) Cl^l(x),$$

where $w_l(x)$ is the weight of l-th child classifier for a given input attribute vector x. The value returned by each Cl^l classifier (together with the root one) is a probability vector $Cl(x) = (Cl_1(x), \ldots, Cl_K(x))$ (if $c_k \notin \mathcal{C}_{\mathcal{I}}$, then $Cl_k(x) = 0$ for classifier Cl built for subproblem $\mathcal{C}_{\mathcal{I}}$).

The clue to the high overall HC accuracy lies in proper partition into subproblems. This partition is based on the training results of

☆ This research was partially funded by Polish National Science Center Grant No. 6548/B/T02/2011/40.

* Corresponding author.
E-mail addresses: igor.podolak@uj.edu.pl (I.T. Podolak), adam.roman@uj.edu.pl (A. Roman).

0167-8655/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.08.002

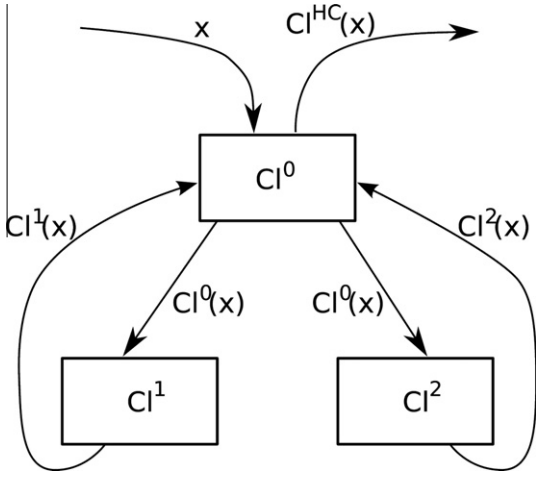

Fig. 1. Data flow in a very simple HC.

the current classifier node. Therefore, it can be thought of as a boosting type approach (Schapire, 1990). After partitioning into subproblems, classifiers defined for all of them are trained independently. In other words, the subproblems are found *before* the children classifiers are trained and the accuracy of the whole HC will be known only *after* all classifiers are trained. The accuracy is described in terms of so-called risk function. To obtain the best possible partition we need to estimate the HC risk. To compute it, we need the subclassifiers risk values. We need to evaluate them without training the subclassifiers.

The aim of this paper is to provide such risk evaluation method for all subproblems to estimate the risk of the whole HC and provide some cost function to find the optimal division into subproblems. This evaluation of the children risks will be based on the risk of the current classifier.

2. The HC structure

Recall that HC is trained using a dataset $\mathcal{D} = \{(x_i, c_i)\}_{i=1}^{N}$, where $x_i \in X$ is an attribute vector and $c_i \in \mathcal{C}$ is the true class of x_i, where $\mathcal{C} = \{C_1, \ldots, C_K\}$.

Definition 1. HC built for a K-class problem given with $\mathcal{D}$ is a tuple $HC = (V, V^0, child)$ such that $V = \{(Cl^l, F^l)\}$ is a set of nodes, each consisting of a classifier Cl^l and a binary clustering matrix F^l, such that

$$F^l = \begin{cases} K \times L^l & \text{matrix if } |child(V^l)| > 0 \\ \emptyset & \text{otherwise,} \end{cases}$$

where $child : V \to 2^V$ and $L^l = \text{vert}child(V^l)|$. V^0 is called the root node. F^l represents the subproblems for Cl^l, *i.e.* a class C_k belongs to a subproblem j iff $F^l_{kj} = 1$.

Definition 2. A loss function $\ell : X \times \mathcal{C} \times \mathcal{C} \to [0, \infty)$ returns the cost of classification of an example, such that $\ell(x, C, y(x)) = 0 \Leftrightarrow y(x) = C$, where $y(x)$ is the classifier prediction and C the true class.

Definition 3. The risk function $R[f] = \mathbb{E}[\ell(f)]$ is the expected value of the loss function, where f is the hypothesis implemented by the classifier.

The training algorithm for HC recursively trains a classifier on the given dataset, then finds a best possible partition into subproblems (defined with the clustering matrix F). Dataset is then partitioned into subproblem datasets and the whole training is repeated until no more divisions can be done. This process is described in Algorithm 1.

Algorithm 1: Hierarchical classifier training. *stopCondition*() may be a function of the number of classes, error level, tree depth, etc. *ncolumns*(F) returns the number of subproblems defined with F.

Algorithm: $HC(\mathcal{D})$
Input: $\mathcal{D}$ – training dataset
Output: HC classifier
$Cl \leftarrow trainClassifier(\mathcal{D})$
$M \leftarrow misclassificationMatrix(Cl, \mathcal{D})$
$F \leftarrow findClustering(M)$
$V \leftarrow (Cl, F)$
if $ncolumns(F) > 1$ **then**
  $child(V) \leftarrow \emptyset$
  for $l = 1$ **to** $ncolumns(F)$ **do**
    $\mathcal{C}_l = \{C_i : f_{il} = 1, i = 1, \ldots, K\}$
    $\mathcal{D}_l = \{(x, c) \in \mathcal{D} : c \in \mathcal{C}_l\}$
    if not $stopCondition(\mathcal{C}_l)$ **then**
      $V_l \leftarrow HC(\mathcal{D}_l)$
      $child(V) \leftarrow child(V) \cup \{V_l\}$
    end
  end
end
return (V)

The F matrix is found using only the *misclassification* matrix of Cl (frequently called a confusion matrix, e.g. in (Parker, 2001)).

Definition 4. A misclassification matrix $M = \{m_{ij}\}_{i,j=1}^{K}$ for a classifier Cl, is a stochastic $K \times K$ matrix such that $m_{ij} = P(pr = C_j | tr = C_i)$ (pr stands for *predicted*, tr stands for *true*), i.e. the likelihood that an example from some true class C_i is predicted by Cl as being from class C_j.

F matrix defines a number of subproblems, each characterized by a different set of classes. Subproblem consists of classes, examples of which were similarly classified with the parent Cl and which are found by a clustering algorithm *findClustering(M)* (Podolak, 2008; Podolak and Bartocha, 2009). This follows a hypothesis, that if a classifier is weak, then classes incorrectly classified are usually predicted to belong to classes which share some common characteristics and are predominant (Podolak and Roman, 2009). There are different approaches to classifiers selection for ensembles, eg. by measuring their competence (Woloszynski and Kurzynski, 2011) or clustering. Authors have proposed a number of clustering algorithms, mainly based on some machine learning approaches (Podolak, 2008). In these methods the problem is to define an appropriate cost function for clustering. Simple cost functions may come from some heuristics assumed. On the other hand, better results can be obtained if $R[HC]$ risk estimation itself is used. But to compute $R[HC]$, subproblem classifiers risk value is needed. Hence the approach proposed in this paper to evaluate it.

3. Subproblem risk evaluation

Let $\mathcal{M}$ be some training algorithm for a K-class classification problem and let Cl be a classifier built with $\mathcal{M}$ using training set $\mathcal{D}$.

Problem 1. Let a subproblem $\mathcal{I}$ be given. Is it possible to predict $R[Cl_{\mathcal{I}}]$ using some properties of $\mathcal{M}$?

3.1. Naïve solution

It may be noticed, that if $\mathcal{M}$ is used to train both the parent and child classifiers, then both would have some common characteris-