Pattern Recognition 47 (2014) 3429–3437

Contents lists available at ScienceDirect

Pattern Recognition

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

Random Forests with ensemble of feature spaces

Le Zhang [1], Ponnuthurai Nagaratnam Suganthan *

School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, 639798, Singapore

ARTICLE INFO

Article history:
Received 10 April 2013
Received in revised form
19 December 2013
Accepted 1 April 2014
Available online 12 April 2014

Keywords:
Ensemble
Classification
Diversity
Transformation
Rotation

ABSTRACT

Random Forests receive much attention from researchers because of their excellent performance. As Breiman suggested, the performance of Random Forests depends on the strength of the weak learners in the forests and the diversity among them. However, in the literature, many researchers only considered pre-processing of the data or post-processing of the Random Forests models. In this paper, we propose a new method to increase the diversity of each tree in the forests and thereby improve the overall accuracy. During the training process of each individual tree in the forest, different rotation spaces are concatenated into a higher space at the root node. Then the best split is exhaustively searched within this higher space. The location where the best split lies decides which rotation method to be used for all subsequent nodes. The performance of the proposed method here is evaluated on 42 benchmark data sets from various research fields and compared with the standard Random Forests. The results show that the proposed method improves the performance of the Random Forests in most cases.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Ensemble classifiers or multiple classifier systems (MCS) have been actively investigated in applied statistics [1], machine learning [2] and pattern recognition [3]. Several studies show that combining multiple weak classifiers into one aggregated classifier leads to better classification performance than that can be obtained from any of the weak individuals [4]. In the literature, there exist several different strategies to build the ensemble classifiers, i.e. (i) build each weak classifier with different dataset sampled from the original training samples or different subspaces form the training data, (ii) the aggregation strategy for the output of the classifiers, (iii) employing different learning algorithms to obtain different base classifiers and (iv) hybridization of the above 3 strategies.

The methods belonging to the first strategy include data sampling schemes [5], subspace selection [6], multiple kernels [7], transformation methods [8]. The second strategy for designing an ensemble involves the use of fusion rules to aggregate the output of the weak individuals, which ranges from simple average voting to more complex combination rules [9–11]. The method corresponding to the third strategy is hybrid ensembles, where different types of classification algorithms are combined [12]. Then one can easily use hybridization version of the above methods to yield complicate ensemble classifiers. The most popular mechanisms to build the ensemble based classifiers are non-hybrid which generally works by means of firstly generating an ensemble of base classifiers via applying a given base algorithm to different permutated training subsets and/or different subspaces, and then the output of the ensemble is aggregated in a suitable way.

There are many well-documented method in the literature, among which bagging (bootstrap aggregating) is the most widely used [5]. Although many variations have been proposed [13,14,4], Breiman's original idea is still widely used in building the ensemble classifier. In bagging, each weak classifier is trained on bootstrap samples of the original training samples. The output of the ensemble is obtained by means of uniformly voting, which means that the unlabeled test data is assigned the label with the highest number of votes among the weak learners.

The random subspace method (RSM) [6] is a method of combining models. Learning machines are trained on randomly chosen subspaces of the original input space (i.e. the training set is sampled in the feature space). The outputs of the models are then combined, usually by a simple majority vote.

Proposed by Breiman, Random Forests combine bagging and random subspace, which has demonstrated high classification performance in many field of research [15–18]. As mentioned above, Random Forests combine bagging and a specific form of a random subspace method where random subspace is conducted at each node of the classification and regression tree (CART) [19]. Random Forests use recursive partitioning to generate many classification and regression trees and then aggregate the results. Each tree is independently constructed using a bootstrap sample of the training data. Specifically, each tree is constructed using the following method:

* Corresponding author. Tel.: +65 67905404.
E-mail addresses: Lzhang027@e.ntu.edu.sg (L. Zhang), epnsugan@ntu.edu.sg (P.N. Suganthan).
[1] Tel.: +65 84231761.

http://dx.doi.org/10.1016/j.patcog.2014.04.001
0031-3203/© 2014 Elsevier Ltd. All rights reserved.

Training phase: Given X – the object in the training data set (an $N \times m$ matrix, where N is the number of the training data, $m$is the dimension of each data). Y – the labels of the training set(an $N \times 1$ matrix). L – the ensemble size, which means the number of trees in the forests. T_i (each random tree in the Random Forests, $i = 1 \ldots L$). *mtry* – the number of features randomly selected to split in each non-leaf node.

1. For $i = 1 \ldots L$
2. Generate the training set for T_i by sampling N times from all N available training cases with replacement.
3. At each node the best split is calculated using the *mtry* randomly chosen features in the training set for T_i.
4. Go to step 3 until T_i is fully grown without being pruned.

Classification phase: For a given sample, it is pushed down each tree in the forests and each tree in the forests will give one vote on the label of this sample. In this case, the predicted label of this sample is determined as the one which has the most votes in the forests.

More recently, additional properties of the Random Forests have gained interest, for example in feature selection [20–23], or explorative analysis of sample proximities [24]. Also there are researchers who attempted to improve the performance of Random Forests by either performing feature selection firstly [25] or finding a more suitable way to aggregate the results of the ensemble members [26]. The method in the first paper, which is actually a pre-processing method, works by comparing various feature selection methods for Random Forests in order to predict antifreeze proteins from sequence-derived properties. In the second paper, the author evaluated their weighted voting method which belongs to the post-processing method mentioned before on 17 UCI dataset.

Oblique Random Forests [27] use the idea of ridge regression to calculate the best splitting at each node. In fact, their Oblique Random Forests is based on the projection method which project the data at each node into another feature space, then the best split is calculated in that space. We can still find some details in the previous work of Breiman about two versions of CART: one generated from "orthogonal" trees with threshold on individual feature in every split and one from "oblique" trees separating the feature space by combining the selected feature values, which can be viewed as a transformation method. However, he did not propose a systematic method to combine the feature in Random Forests.

Ye et al. [28] builds Stratified Random Forests to deal with the high dimensional data. In their work, they divide the features of the data into two groups. One group will contain strong informative features and the other weak informative features. Then for feature subspace selection, they randomly select features form the two groups proportionally. Fisher discriminant projection is employed in their work to divide the feature into two groups. They find the most discriminative eigenvector W, or in other words, the eigenvector W corresponding to the largest eigenvalue firstly. Suppose $W = (W_1 \ldots W_n)$, then the absolute value of W_i is selected to measure the importance of the ith feature. However, SRF can only deal with binary problems. So we have to do the decomposition work such as one-against-one SRF or one-against-all SRF when deal with the multi-class problem. So in this case, SRF lose its advantage compared to standard Random Forests. Furthermore, dimensionality reduction or feature extraction is usually performed before the classification stage for real-life applications. There will be less weak informative features after dimensionality reduction.

In the present work, we propose to build the Random Forests with two projection methods. We firstly suggest the Rotation Random Forests which aim at building individual trees with high accuracy and diversity. The main idea is to apply transformation method to transform the data at each node to another space when computing the best split at this node. We have chosen Principal Component Analysis (PCA) [29] and Linear Discriminate Analysis (LDA) [30] in this study for transformation. In this study, we name these two versions of Rotation Random Forests as PCA based Random Forests and LDA based Random Forests and the member of the two ensembles as PCA based CART and LDA based CART respectively. We have to mention that the LDA based Random Forests is different from the Discriminant Random Forests [31] in the literature. The Discriminant Random Forests tries to obtain c-1 Discriminant vectors in each node (c is the number of the class) while there is no dimension reduction at each node in LDA based Random Forests. The detail will be discussed in the following section. We also test the Discriminant Random Forests with several datasets .The performances of this method are far from satisfying and hence are not reported in this study. As PCA and LDA perform well in the field of pattern classification [32] and more specifically, ensemble learning [33,34,8] , we introduce a new method to integrate the two types of Rotation Random Forests and the standard Random Forests into an ensemble. We will consider building the Random Forests that consist of PCA based CART and LDA based CART and the standard CART (orthogonal trees as proposed by Breiman), which will be called Random Forests with ensemble of feature spaces here.

The rest of the paper is organized as follows: Section 2 explains the two versions of Rotation Random Forests and Random Forests with ensemble of feature spaces. Section 3 presents the experiment to compare the standard Random Forests with Rotation Random Forests and Random Forests with ensemble of feature spaces. In Section 4, kappa-error diagram will be plotted to illustrate the strength and diversity between individual classifiers to show the improvement of our method to Breiman's standard Random Forests. Section 5 presents our conclusions and outlines direction of future work.

2. Rotation random forests and random forests with ensemble of feature spaces

The main mechanism behind the Rotation Random Forests is to transform or rotate the data feature space at each node to another space. Note that this is different from applying a transformation to the whole of the data before generating the Random Forests model, as we proposed to apply the transformation at each node. Since we are likely to choose a different subspace of features at each node while building a CART, the transformation at each node can be totally different. This can yield improved diversity between each pair of weak classifiers. Here we introduce two versions of Rotation Random Forests: PCA based Random Forests which use PCA to transform the data at each node and LDA based Random Forests which use LDA to transform data at each node. Furthermore, we ensemble the two transformation based CART trees and the standard CART tree to form an overall Random Forests, called Random Forests with ensemble of feature spaces. The following steps present the construction of the PCA based Random Forests.

Training phase: Given X – the object in the training data set (an $N \times m$ matrix, where $N$is the number of the training data, m is the dimension of each data). Y – the labels of the training set (an $N \times 1$ matrix). L – the ensemble size, which means the number of trees in the forests. T_i: (each random tree in the Random Forests, $i = 1 \ldots L$). *mtry* – The number of features randomly selected to split in each non-leaf node.

1. For $i = 1 \ldots L$.
2. Generate the training set for T_i by sampling N times from all Navailable training cases with replacement.

Download English Version:

https://daneshyari.com/en/article/530684

Download Persian Version:

https://daneshyari.com/article/530684

Daneshyari.com