Pattern Recognition 58 (2016) 149–158

Contents lists available at ScienceDirect

Pattern Recognition

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

A non-parametric approach to extending generic binary classifiers for multi-classification

Venkataraman Santhanam*, Vlad I. Morariu, David Harwood, Larry S. Davis

UMIACS, University of Maryland College Park, 2126 AVW Building, College Park, MD 20742, USA

ARTICLE INFO

Article history:
Received 27 May 2015
Received in revised form
3 April 2016
Accepted 13 April 2016
Available online 22 April 2016

Keywords:
Multi-classification
Ensemble method
One-vs-one
Orthogonal subspace
Non-parametric density estimation

ABSTRACT

Ensemble methods, which combine generic binary classifier scores to generate a multi-classification output, are commonly used in state-of-the-art computer vision and pattern recognition systems that rely on multi-classification. In particular, we consider the *one-vs-one* decomposition of the multi-class problem, where binary classifier models are trained to discriminate every class pair. We describe a robust multi-classification pipeline, which at a high level involves projecting binary classifier scores into compact orthogonal subspaces, followed by a non-linear probabilistic multi-classification step, using Kernel Density Estimation (KDE). We compare our approach against state-of-the-art ensemble methods (DCS, DRCW) on 16 multi-class datasets. We also compare against the most commonly used ensemble methods (VOTE, NEST) on 6 real-world computer vision datasets. Finally, we measure the statistical significance of our approach using non-parametric tests. Experimental results show that our approach gives a statistically significant improvement in multi-classification performance over state-of-the-art.

1. Introduction

For the task of multi-classification, most research focuses on one of the following:

- Developing a dedicated multi-class classifier (*e.g.* random forests [1]).
- Extending an existing binary classifier to deal with multiple classes, via an internal joint optimization step over multiple classes (*e.g.* SVC by Crammer/Singer [2], Structured SVM [3]).
- Dividing the multi-classification problem into a set of binary classification problems, followed by an *ensemble method* to combine the binary classifier scores into a multi-classification output.

We focus on the last item in the list. Binary classifiers are typically easier to build, faster to train/test and have much simpler decision boundaries when compared to dedicated multi-class classifiers. Moreover, for complex multi-class problems with a large number of classes and/or high feature dimensionality, the more practically viable option is usually to divide the multi-classification problem into several smaller easy-to-solve binary classification problems.

Several binary decomposition strategies exist for the multi-class problem, with the most popular ones being the *one-vs-all* (*OVA* [4]) and *one-vs-one* (*OVO* [5]) schemes. In *OVA*, a binary classifier is trained for each class, designed to distinguish it from the remaining classes. *OVO*, on the other hand, trains a binary classifier to distinguish between every pair of classes. Yet another decomposition strategy, that can be viewed as a generalization of *OVO* and *OVA*, is the *error-correcting-output-code* (*ECOC* [6]) framework, in which each class is assigned a unique fixed length binary codeword, after which a binary classifier for each bit position is trained, based on the codewords for all the classes. Minimal design *ECOCs* offering competitive performance have also been proposed [7], for which the number of binary classifiers required is sub-linear in the number of classes.

Rifkin and Klautau argued that *OVA* can match *OVO* performance, provided that the binary classifiers are well tuned [8]. Their analysis is however restricted to regularized classifiers such as *SVMs*, and is not applicable for generic binary classifiers. The *OVO* scheme typically provides better results than the *OVA* or *ECOC* scheme [3,9]. *OVO* is also more robust over various choices of binary classifiers and provides better scalability with a relative performance boost over *OVA* as the number of classes increases [10]. *OVO* is also surprisingly faster to train than *OVA* (and sometimes even *ECOC*), despite training more binary classifiers. This is because each *OVO* binary classifier is trained only on samples from a specific pair of classes, whereas each binary classifier in *OVA* or *ECOC* is trained using samples from all classes. Furthermore, when parallel computing is available, all *OVO* pairwise classifiers can be

* Corresponding author. Tel.: +1 301 346 4144.
E-mail addresses: venkai@umiacs.umd.edu (V. Santhanam), morariu@umiacs.umd.edu (V.I. Morariu), harwood@umiacs.umd.edu (D. Harwood), lsd@umiacs.umd.edu (L.S. Davis).

http://dx.doi.org/10.1016/j.patcog.2016.04.008
0031-3203/

trained in a massively parallel fashion, even for a very high number of classes.

After a binary decomposition of the multi-classification problem, the resulting binary classifier scores are aggregated to yield a final multi-classification output by strategies that are referred to as *ensemble methods* [11]. Our contribution is a new *OVO ensemble* method using *KDE* over *PCA projections* of binary classifier scores that is robust, yields probabilistic multi-class outputs and outperforms the most commonly used alternatives *VOTE/NEST*.

The rest of the paper is organized as follows: Section 2 details related work in *OVO ensemble methods*. Section 3 introduces notations, formalizes the notion of an ensemble method and describes our proposed approach. Section 4 is dedicated to *Kernel Density Estimation* (*KDE*), along with a *PCA projection* based approximation to multi-variate *KDE* which we use to obtain our probabilistic multi-class decisions. Sections 5 and 6 describe the experiments to test our approach against the state-of-the-art and contain a discussion of the results. Finally, we conclude in Section 7 with a summary of our proposed approach, its novel contributions and potential future improvements.

2. Related work

The most common *OVO ensemble method* is the *naïve voting scheme* (*VOTE* [12]), where all pairwise binary classifiers vote for the class of an unseen sample. The class with the highest number of votes is chosen as the predicted class. An improvement to *VOTE* is the *nesting one-vs-one scheme* (*NEST* [13]), which augments *VOTE* with a recursive tie-breaking scheme for instances where there are ties for the class with the highest vote. Several other *ensemble methods* have been proposed for *OVO* schemes, such as *weighted voting* (*WV* [14]), *Pairwise Coupling* (*PC* [15,16]), *decision directed acyclic graph* (*DDAG* [17]), *learning valued preference for classification* (*LVPC* [18]), *preference relations solved by non-dominance criterion* (*ND* [19]) and *binary tree of classifiers* (*BTC* [20]).

An excellent overview and a detailed experimental study of these *OVO ensemble methods* for various choices of binary classifiers are provided in [10]. Their results indicate that there is no "one method which performs best for all binary classifiers." The methods which perform consistently well, regardless of the choice of binary classifiers, are in fact the ones that are the simplest to explain: *VOTE* and *NEST*. Most of the popular machine learning software libraries used extensively by researchers, such as *LibSVM*, use *VOTE* as their *ensemble* method.

2.1. Limitations of VOTE, NEST

For a multi-classification problem with K classes, the *OVO* scheme trains $K(K-1)/2$ binary classifiers to distinguish between each pair of classes. For a test sample belonging to a class A, there will be $(K-1)(K-2)/2$ classifiers that have never seen any sample from class A. The predictions of any of these classifiers for the sample become arguably questionable. This is a recurring issue for all *OVO* ensemble methods, often referred to as the *non-competence problem* [21].

Considering the fact that both the *VOTE* and *NEST* methods disregard the relative magnitudes of the classifier scores completely and focus only on the binary predictions (votes), the vote given by a *non-competent* classifier is given the same weight as the vote carried by a *competent* one, which may affect results negatively. The success of the *VOTE* or *NEST* method, despite this limitation, is justified by the inherent redundancy in the *OVO* framework, with the rationale that the $(K-1)$ *competent* classifier votes more than compensate for the apparently random votes of the *non-competent classifiers*, which are usually not directed in favor of any one particular class.

However, there may be cases where this assumption is violated, especially for samples that tend to be confused between a pair and small subset of classes. Such samples tend to have several classes with a comparable (albeit low) vote count. The votes of *non-competent classifiers*, in this case, tend to have a stronger influence on the final prediction.

In order to overcome this limitation, a mechanism needs to be devised to efficiently utilize the information contained in the magnitudes of the scores. However, the raw scores for different *OVO* classifiers are not calibrated, centered at different thresholds, and have a potentially different scale and range. So, it would be unwise to use raw score magnitudes in any multi-classification scheme. The most intuitive way to make the scores of different *OVO* classifiers comparable is to convert them to probabilities.

2.2. Addressing the non-competence problem

A couple of approaches, *DCS* [22] and *DRCW-OVO* [23], have been recently proposed which specifically aim to reduce the *non-competence* effect. Both these techniques focus on *removal* (*DCS*) or *reweighting* (*DRCW-OVO*) of *non-competent* classifiers and can in fact be used as a pre-processing step for any subsequent *OVO* ensemble method (in place of *weighted voting* used in *DCS* and *DRCW*).

In *DCS* (*Dynamic Classifier Selection*), 3K nearest neighbors in the training data are first computed for each test sample based on the original feature space. An *i-vs-j OVO* classifier is flagged as *non-competent* (and removed) with respect to a given test sample, if its 3K neighbors contain no samples from either of class i or class j.

In *DRCW-OVO* (*Distance-based Relative Competence Weighting*), distances $(d_1, \ldots, d_K)$ are computed for each test sample, where d_i is the average distance to the 5 nearest neighbors from class i training samples in the original feature space. Subsequently, the score s_{ij} of each *i-vs-j OVO classifier* is reweighted as $s_{ij}^* = s_{ij} * d_j^2/(d_i^2 + d_j^2)$. The rationale behind this transformation is that the scores for the *competent* classifiers will be skewed more in favor of the true class whereas the scores of the *non-competent* classifiers will not be affected as much (due to the ratio $d_j^2/(d_i^2 + d_j^2)$ being closer to 0.5).

2.3. Probability estimates for binary classification

The sign of a binary classifier score represents the predicted class, while its magnitude crudely encodes the confidence of the predicted decision. Several *classifier specific* methods that convert scores to probabilities exist. For max-margin classifiers, such as *Linear SVM*, techniques such as *Platt scaling* convert *scores to probabilities* by means of a *sigmoidal* function [24,25]. On the other hand, for classifiers such as *Naïve Bayes*, *Platt scaling* performs poorly while *isotonic regression* [26] gives better probability estimates [27]. It is also shown in [27], that neither *Platt scaling* nor *isotonic regression* performs well for classifiers such as neural nets, bagged trees, and logistic regression which already provide well calibrated scores.

2.4. Probability estimates for multi-classification

Obtaining multi-class probability estimates is a much trickier proposition. For the *OVO* scheme, we are required to estimate a probability for each class given all the pairwise binary classifier scores. Previously proposed methods typically employ a two-step strategy: convert each binary classifier score to a probability r_{ij}, and then combine r_{ij}'s to obtain the multi-class probabilities p_i's. The combination strategy is formulated as an optimization