Pattern Recognition 44 (2011) 2656–2668

Contents lists available at ScienceDirect

Pattern Recognition

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

A probabilistic model of classifier competence for dynamic ensemble selection

Tomasz Woloszynski *, Marek Kurzynski

Chair of Systems and Computer Networks, Faculty of Electronics, Wroclaw University of Technology, Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland

ARTICLE INFO

Article history:
Received 26 August 2010
Received in revised form
16 March 2011
Accepted 17 March 2011
Available online 29 March 2011

Keywords:
Probabilistic modelling
Classifier competence
Multiple classifier system
Beta distribution

ABSTRACT

The concept of a classifier competence is fundamental to multiple classifier systems (MCSs). In this study, a method for calculating the classifier competence is developed using a probabilistic model. In the method, first a randomised reference classifier (RRC) whose class supports are realisations of the random variables with beta probability distributions is constructed. The parameters of the distributions are chosen in such a way that, for each feature vector in a validation set, the expected values of the class supports produced by the RRC and the class supports produced by a modelled classifier are equal. This allows for using the probability of correct classification of the RRC as the competence of the modelled classifier. The competences calculated for a validation set are then generalised to an entire feature space by constructing a competence function based on a potential function model or regression. Three systems based on a dynamic classifier selection and a dynamic ensemble selection (DES) were constructed using the method developed. The DES based system had statistically significant higher average rank than the ones of eight benchmark MCSs for 22 data sets and a heterogeneous ensemble. The results obtained indicate that the full vector of class supports should be used for evaluating the classifier competence as this potentially improves performance of MCSs.

1. Introduction

Multiple classifier systems (MCSs) were shown to outperform single classifiers for a wide range of classification problems [1–4]. The reason is that a combination of classifiers reduces risks associated with picking an inadequate single classifier, choosing a space of classifiers not containing the optimal classifier, and falling into local error minima during training [5–7]. However, in order for an ensemble of classifiers to perform better than an individual classifier, the ensemble has to be diverse (i.e. the classifiers have to make independent errors) and the combination method used has to effectively exploit that diversity [8,9]. One possible way to achieve diversity of the ensemble is to generate different training sets for the classifiers through, for examples, bootstrapping [10], boosting [11] and random subspaces [12]. Another way is to use a heterogeneous ensemble of structurally diverse classifiers [4,13,14].

For a combination of classifiers, two approaches used are classifier fusion (CF) and classifier selection (CS). In the CF approach, a test object is classified using a combination function and all classifiers in the ensemble. The combination functions used are sum, product, maximum, minimum, majority voting, fuzzy integral, and others [6,8,9]. However, redundant and inaccurate classifiers in the ensemble can adversely affect performance of a system based on the combination functions. This is because redundant classifiers reduce diversity of the ensemble and subsequently they only increase complexity of the system [13]. Also, performance of the system is unlikely to improve if inaccurate classifiers are included in the combination process. To remedy this, ensemble pruning (EP) methods have been developed [15–18]. The methods are based on selecting and combining a subset of classifiers from the ensemble instead of combining all. The selection criteria used are diversity [6,19] and performance [20] of the selected subset, and a mixture of the two [13,21]. For small ensembles, the optimal subset can be found through exhaustive search. For large ensembles, a quasi-optimal subset is found using heuristic and hill-climbing optimisation algorithms, e.g. genetic algorithms [13,15], reinforcement learning [22] and quadratic integer programming [23]. However, the subset selection in the EP methods is independent on the location of the test object in a feature space. Consequently, there may exist a different subset that locally performs better than the subset selected globally.

In the CS approach, the test object is classified by a single classifier that is statically or dynamically selected from the

* Corresponding author.
E-mail addresses: tomasz.woloszynski@pwr.wroc.pl (T. Woloszynski), marek.kurzynski@pwr.wroc.pl (M. Kurzynski).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.03.020

ensemble. In the static classifier selection, first each classifier in the ensemble is assigned with a region of competence in the feature space during training. Then, the classifier assigned with the region of competence that contains the test object is selected [24,25]. In the dynamic classifier selection (DCS), first a competence of each classifier is evaluated for the test object and then the most competent classifier is selected [6,26,27]. For the calculation of the competence, various performance estimates are used [26–29]. One drawback of the CS methods is that their performance depends solely on an accurate estimation. Another drawback is that the region of competence may be more difficult to estimate than the optimal decision boundary for some simple classification problems.

Recently, dynamic ensemble selection (DES) methods that use a mixture of the CF and CS approaches have been introduced [30,31]. The methods select and combine a subset of classifiers from the ensemble for each test object. The selection criteria used are performance of individual classifiers [30] and subsets of classifiers [31]. There are also DES methods that do not require the ensembles of trained classifiers, e.g. mixtures of experts (MoE) [32]. In the MoE, classifiers are trained and their competences are calculated in a coupled manner. However, many of the DCS and DES methods are ad hoc or heuristic. Consequently, it is difficult to draw sound conclusions about possible improvements for either a specific method or MCSs in general. For this reason, there is a growing interest in the theoretical explanation and justification of approaches, methods and concepts used for classifier combination [6,8,33].

In this paper, the classifier competence is studied using a probabilistic model. The study is the continuation of the previous work on competence measures for DES based systems [34–36]. In the previous work, a support given by a classifier for the correct class was modelled by a random variable and a competence measure based on the modelling was developed. However, not all values of the support could be modelled and the measure could not be used to evaluate worse-than-random classifiers. In this study, a unified modelling of the full vector of class supports is derived. Using the modelling, a competence measure is developed that can be used to evaluate any classifier in the ensemble. Three DCS and DES based systems were constructed using the measure developed. Performance of the systems was compared against two classical combination methods (single best and majority voting) and six DCS and DES based systems such as DCS-potential function estimate (DCS-PFE) [26], DCS-local accuracy (DCS-LA) [27], DCS-modified local accuracy (DCS-MLA) [28], DCS-multiple classifier behaviour (DCS-MCB) [29], DES-K nearest oracles eliminate (DES-KE) [30], and mixtures of experts (MoE) [32]. For the comparison, 22 benchmark data sets from the UCI Machine Learning Repository [37], the Ludmila Kuncheva Collection [38] and the ELENA project [39] were used.

This paper is organised as follows. In Section 2, a probabilistic model of the classifier competence is developed. Section 3 describes the systems that were constructed using the model. The experiments conducted are shown in Section 4 and the results with discussion are presented in Section 5. The paper is concluded in Section 6.

2. Theoretical framework

2.1. Classifier ensemble

Let a set of trained classifiers $\Psi=\{\psi_1,\ldots,\psi_L\}$, called a classifier ensemble, be given and let a classifier ψ_l, $l=1,\ldots,L$ be a function $\psi_l:\mathcal{X}\to\mathcal{M}$ from a feature space $\mathcal{X}\subseteq\mathbb{R}^n$ to a set of class labels $\mathcal{M}=\{1,\ldots,M\}$. A canonical model of classification is assumed [6,40], where the classifier ψ_l produces a vector of class supports $[d_{l1}(x),\ldots,d_{lM}(x)]$ for a feature vector $x\in\mathcal{X}$. It is further assumed, without loss of generality, that $\sum_{j=1}^{M} d_{lj}(x)=1$ and $d_{lj}(x)\geq 0$. Classification is made according to the maximum rule

$$\psi_l(x)=i \;\Leftrightarrow\; d_{li}(x)=\max_{j\in\mathcal{M}} d_{lj}(x). \tag{1}$$

The ensemble Ψ is used for classification through a combination function which, for example, can select a single classifier or a subset of classifiers from the ensemble, it can be independent or dependent on the feature vector x (in the latter case the function is said to be dynamic), and it can be non-trainable or trainable. For the dynamic combination functions, the concept of a classifier competence is frequently used. A competence function $c(\psi_l,x)$ estimates performance of the classifier ψ_l for x and it usually takes values in the interval [0,1], where the value of 0 (1) indicates the least (the most) competent classifier. Ideally, the function should be easy to calculate for arbitrary numbers of classes, features, and classifiers and it should be independent on the combination function and the methods used for constructing classifiers in the ensemble. In this study, a trainable competence function with the above properties is developed using a probabilistic model. For the training of the competence function, it is assumed that a validation set $V=\{(x_1,j_1),\ldots,(x_N,j_N)\}$ containing pairs of feature vectors and their corresponding class labels is available. For the existing DCS and DES based systems, the function developed would replace the module that calculates the classifier competences using the validation set.

2.2. Measuring the classifier competence

A natural competence measure of the classifier ψ_l for the feature vector x is the probability of correct classification $P_c(\psi_l|x)$. The probability can be written as

$$P_c(\psi_l|x)=\sum_{j=1}^{M}\Pr\{x \text{ belongs to the } j\text{-th class} \wedge \psi_l(x)=j\}, \tag{2}$$

where $\Pr\{S\}$ is the probability that a statement S is true. However, the probability is equal to 0 or 1 unless at least one of the two terms inside the probability operator in (2) is a random event. This is true in one of the two following cases[1]:

1. A probabilistic model of classification is used, where feature vectors and class labels are realisations of a random variable pair (X,J). Using the model, the probability (2) becomes

$$P_c(\psi_l|x)=\Pr\{x \text{ belongs to the } j\text{-th class}\},\quad \text{where } \psi_l(x)=j. \tag{3}$$

2. The classifier ψ_l assigns the class label j to the feature vector x in a stochastic manner. In this case, the probability (2) becomes

$$P_c(\psi_l|x)=\Pr\{\psi_l(x)=j\},\quad \text{where } j \text{ is the class label of } x. \tag{4}$$

There are problems, however, in both cases. First, the probabilistic model of classification is often used to construct some of the classifiers in the ensemble (as it is in this study) and therefore, it should not also be used to construct the competence function. This is because no one should be a judge in their own cause[2], meaning that the use of the same learning paradigm to construct a classifier and to evaluate its competence is unfair to the

[1] The case where both terms are random events is not considered.

[2] *Nemo iudex in causa sua*, a fundamental principle of natural justice ensuring fairness of judgement.