Pattern Recognition Letters 111 (2018) 36–42

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Restricted Set Classification with prior probabilities: A case study on chessboard recognition

Ludmila I. Kuncheva*, James H.V. Constance

School of Computer Science, Bangor University, Dean Street, Bangor, Gwynedd LL57 1UT, United Kingdom

ARTICLE INFO

Article history:
Received 6 June 2017
Available online 12 April 2018

Keywords:
Classification methodology
Restricted Set Classification
Simultaneous classification
Image recognition
Chess piece recognition

ABSTRACT

In the Restricted Set Classification approach (RSC), a set of instances must be labelled simultaneously into a given number of classes, while observing an upper limit on the number of instances from each class. In this study we expand RSC by incorporating prior probabilities for the classes and demonstrate the improvement on the classification accuracy by doing so. As a case-study, we chose the challenging task of recognising the pieces on a chessboard from top-view images, without any previous knowledge of the game. This task fits elegantly into the RSC approach as the number of pieces on the board is limited, and each class (type of piece) may have only a fixed number of instances. We prepared an image dataset by sampling from existing competition games, arranging the pieces on the chessboard, and taking top-view snapshots. Using the grey-level intensities of each square as features, we applied single and ensemble classifiers within the RSC approach. Our results demonstrate that including prior probabilities calculated from existing chess games improves the RSC classification accuracy, which, in its own accord, is better than the accuracy of the classifier applied independently.

1. Introduction

Restricted Set Classification (RSC) refers to the following problem. Given is a set containing m instances, $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, where $\mathbf{x}_j \in \mathbb{R}^n$, $j = 1, \ldots, m$, is a data point in some n-dimensional space. Each instance must be labelled in one of c classes from the set $\Omega = \{\omega_1, \ldots, \omega_c\}$. It is known that the maximum number of instances from class ω_i, present within X, is k_i, $i = 1, \ldots, c$. Thus the cardinality of X must satisfy $1 \leq |X| \leq \sum_{i=1}^{c} k_i$.

The solution to this problem is not straightforward. If a classifier is trained and then applied for labelling the instances in X (called the 'independent classifier'), the obtained labels are not guaranteed to meet the count constraints. Incorporating these constraints into the classification process has been shown to lead to an improvement on the accuracy of the independent classifier [14,15]. Here we hypothesise that a further improvement can be achieved if prior probabilities depending on the whole of X are considered by the RSC set classifier.

Examples of real-life RSC problems include recognising people in a group (e.g., for attendance monitoring of the students in a class [1] or for tracking [17]) and identification of animals for the purposes of monitoring and conservation [7,13]. A particularly suitable application is identifying the pieces on a chessboard from an image. When classifying chess pieces *together*, we can take advantage of the knowledge that there can only be a given number of objects from each class. For example, there can be at most eight white pawns on the board. In this paper, we chose chessboard recognition as an example to demonstrate the expected improvement on the classification accuracy of the independent classifier when using prior probabilities.

The rest of the paper is organised as follows. The RSC approach is detailed in Section 2. Our proposed extension is described in Section 3. Section 4 contains our case-study which demonstrates the improvement of the proposed approach over the original RSC in recognising chess pieces on a board. Section 5 offers our conclusions and ideas for future work.

2. Restricted Set Classification (RSC)

RSC is detailed in Algorithm 1. The RSC approach operates by applying a pre-trained classifier D to X to acquire estimates of the posterior probabilities for every instance within, and making an optimal label assignment while observing the count restriction. The classifier D is termed *the independent classifier* as it is trained on independent, identically distributed (i.i.d.) data, and is oblivious to any count limits. This can be any classifier which returns estimates of the posterior probabilities, $D(\mathbf{x}) = \{P_D(\omega_1|\mathbf{x}), \ldots, P_D(\omega_c|\mathbf{x})\}$. Denoting the space of probability distri-

* Corresponding author. .
E-mail address: l.i.kuncheva@bangor.ac.uk (L.I. Kuncheva).

https://doi.org/10.1016/j.patrec.2018.04.018
0167-8655/

Algorithm 1: Restricted Set Classification.

Input: Pre-trained classifier $D : \mathbb{R}^n \to \mathcal{P}(\Omega)$, the allowed number of instances from each class $K = \{k_1, \ldots, k_c\}$, a set of instances to be classified together $X = \{\mathbf{x}_1, ..., \mathbf{x}_m\}$, $\mathbf{x}_i \in \mathbb{R}^n$.

Output: Labels L for the instances in X.

```
// acquire probability matrix P^p
for i ← 1, ..., m do
    P^p(i, 1 : c) ← D(x_i)

// construct augmented probability matrix P^a
P^a ← ∅.
for i ← 1, ..., m do
    cc ← 1 // column counter
    for j ← 1, ..., c do
        for k ← 1, ..., k_i do
            P^a(i, cc) ← P^p(i, j)
            cc ← cc + 1

// find optimal label assignment M
M ← hungarian-assignment(−log(P^a))
L ← retrieve-labels(M)
Return L.
```

butions over Ω by $\mathcal{P}(\Omega)$, we have $D : \mathbb{R}^n \to \mathcal{P}(\Omega)$. It is desirable that these estimates are well calibrated [5].

D can be a single classifier or a classifier ensemble itself, as long as the output is a probability distribution. Straightforward estimates of the posterior probabilities from a classifier ensemble are the proportions of votes for the respective classes.

The posterior probability estimates for all instances in X are organised in an $m \times c$ "probability matrix" P^p, where row i represents the probability distribution obtained from D for instance $\mathbf{x}_i \in X$. Subsequently, an augmented probability matrix, P^a is constructed by repeating each column of P^p as many times as the number of allowed instances from the corresponding class. For example, if $k_1 = 3$ and $k_2 = 4$, the first three columns of P^a will be copies of the first column of P^p, followed by four copies of the second column of P^p. Thus the size of P^a is $m \times q$, where $q = \sum_{i=1}^{c} k_i$. We have previously proved [14,15] that the optimal assignment guaranteeing the minimum Bayes error in labelling the whole of X requires that the product of the posterior probabilities is maximum, that is

$$\langle \omega_1^*, \omega_2^*, \ldots, \omega_m^* \rangle = \arg \max_{\langle \omega_1^{(a)}, \omega_2^{(a)}, \ldots, \omega_m^{(a)} \rangle} \prod_{i=1}^{m} P\left(\omega_i^{(a)} | \mathbf{x}_i\right), \tag{1}$$

where $\omega_i^{(a)}$ is the class label assigned to $\mathbf{x}_i$, and ω_i^* is the optimal label. This optimisation must be carried out subject to the condition that the number of labels for class ω_j in the returned set must be no greater than the restriction constant k_j, $j = 1, \ldots, c$. The construction of the augmented matrix with posterior probabilities guarantees the compliance with the constraints. In order to find the optimal $\langle \omega_1^*, \omega_2^*, \ldots, \omega_m^* \rangle$, we need a matching procedure. The Hungarian algorithm finds the optimal match which *minimises* the *sum* (or cost) of assignments. Therefore, in order to use this algorithm we convert the product in Eq. (1) into a sum of logarithms. As we are seeking to *maximise* this sum while the algorithm looks for minimum cost, we submit to the Hungarian algorithm the matrix with the negative logarithms P^a.

The output of the Hungarian algorithm is a binary matrix M of the same size as P^a ($m \times q$), containing 1s where rows are assigned the column label, and 0s elsewhere. Each row (instance in X) has one and only one assigned column. The class label of the instance is retrieved by identifying which class label has given rise to the column in P^a. In the above example, if a column between 1 and 3 contains the 1 for the row, the label for the instance is ω_1. Alternatively, if the 1 is in one of the columns between 4 and 7, class ω_2 will be retrieved.

The theoretical grounds and empirical evidence that the RSC works better than m independent applications of D to the elements of X are given in the original work [15]. Here we are interested in extending RSC to incorporate prior probabilistic information, as proposed next.

3. Incorporating a conditional prior into RSC

Suppose that by analysing a large prior database, we were able to obtain prior probabilities depending on some parameter of the set of instances X. This parameter can be, for example, the cardinality of X or some relationship between the instances in X, $\theta = \theta(X)$. Say, we are recognising the students in a class from a photo of the classroom. While the students can sit wherever they choose in the classroom, some usually pick the same seats. We can use a parameter such as

$\theta =$ Sitting in the first row? (y/n),

and pre-calculate a prior probability for each student (class) conditioned on θ. The appearance of the student's face in the photo, which would be their feature vector $\mathbf{x}$, will not depend on θ.

Denote by $P_P(\omega_i|\theta)$ the conditional prior probability for class ω_i, $i = 1, \ldots, c$. To integrate this probability within the probabilities obtained from the independent classifier, P_D, we use

$$P(\omega_k|\mathbf{x}, \theta) = \frac{P(\mathbf{x}, \theta|\omega_k)P(\omega_k)}{P(\mathbf{x}, \theta)}$$

Assuming independence between $\mathbf{x}$ and θ,

$$\begin{aligned} P(\omega_k|\mathbf{x}, \theta) &= \frac{P(\mathbf{x}|\omega_k)\, P(\theta|\omega_k)\, P(\omega_k)}{P(\mathbf{x})\, P(\theta)} \\ &= \underbrace{\frac{P(\mathbf{x}|\omega_k)\, P(\omega_k)}{P(\mathbf{x})}}_{\text{posterior}} \frac{P(\theta|\omega_k)}{P(\theta)} \end{aligned}$$

Multiplying and dividing by $P(\omega_k)$,

$$\begin{aligned} P(\omega_k|\mathbf{x}, \theta) &= P(\omega_k|\mathbf{x}) \frac{P(\theta|\omega_k)P(\omega_k)}{P(\theta)} \frac{1}{P(\omega_k)} \\ &= P(\omega_k|\mathbf{x}) \frac{P(\omega_k|\theta)}{P(\omega_k)}. \end{aligned}$$

Any estimate of the probabilities can be plugged in this equation. In our case:

$$P_E(\omega_k|\mathbf{x}, \theta) = \underbrace{P_D(\omega_k|\mathbf{x})}_{\text{from } D} \quad \underbrace{\frac{P_P(\omega_k|\theta)}{P_P(\omega_k)}}_{\text{from the prior database}}.$$

We may wish to control the influence of the conditional prior probability on the final posterior probability. Therefore we introduce a tunable parameter, $\beta \in [0, 1]$, as follows:

$$P_E(\omega_k|\mathbf{x}, \theta) = P_D(\omega_k|\mathbf{x}) \left[\frac{P_P(\omega_k|\theta)}{P_P(\omega_k)} \right]^{\beta}. \tag{2}$$

This probability distribution across the class labels Ω should be calculated for each instance $\mathbf{x}_j \in X$ and used instead of $D(\mathbf{x}_j)$ in constructing P^p in Algorithm 1.

Note that the conditional prior is only available in relation to the whole set X. Arguably, this probability extension can be thought of as coming from an extra classifier built upon an alternative feature space containing only θ.