Single Classifier Based Multiple Classifications

Albert Hung-Ren Ko and Robert Sabourin
ETS, University of Quebec
1100 Notre-Dame West Street, Montreal, Quebec, H3C 1K3 Canada

Abstract. In this paper, a Single Classifier-based Multiple Classification Scheme (SMCS) is proposed as an alternative multiple classification scheme. The SMCS uses only a single classifier to generate multiple classifications for a given test data point. Because of the presence of multiple classifications, classification combination schemes, such as majority voting, can be applied, and so the mechanism may improve the recognition rate in a manner similar to that of Multiple Classifier Systems (MCS). The experimental results confirm the validity of the proposed SMCS as applicable to many classification systems.

1 Introduction

Most EoCs are created so that an abundance of diverse classifiers is generated, and subsequently an optimal subset of classifiers is selected. By partially omitting selected samples from a sample pool for each classifier training operation, we create different data subsets [458]. Then, by using these data subsets to train classifiers, every classifier will be different from the others. Multiple classifiers yield multiple class labels for a given test sample, and we can combine these multiple class labels into a single class label. Given that each classifier actually draws a boundary between classes, the MCS obtains a new boundary by applying a fusion function, that is, de facto, a combination of different boundaries drawn by different classifiers.

In this paper, we propose an unconventional Single-Classifier-based Multiple Classification Scheme (SMCS) approach that is similar to the MCS, but without the need to train multiple classifiers. We propose a mechanism that achieves multiple classifications with a single classifier, and so benefits from the logic of an MCS without repetitive classifier training and without classifier selection. Given a test sample to classify and some training samples, our method divides the training samples into two groups: one containing what we call reference samples, and the other containing what we call evaluation samples. We use different reference samples to generate different pseudo test data points, each of which constitutes a different combination of an original test sample and some reference samples (Fig. 1), and we use evaluation samples to select adequate reference samples for pseudo test data point generation. Because we use different reference samples to generate pseudo test data points, data diversity is extracted from the original training data in a way similar to that in MCS. Furthermore, because of the generation of multiple pseudo test data points for an original test sample, we can obtain multiple classifications for that test sample. Consequently, traditional classification combination schemes in the MCS, such as the majority voting fusion function, can be implemented.
to generate a final class label. The proposed method can somehow be related to local learning [1], in which local information is exploited to facilitate classification task.

Note that the generation of pseudo data points to improve classification accuracy is not new [6]. The generation of artificial training examples, known as virtual examples, have been proposed for Support Vector Machine (SVM) [2][10]. However, this is different from the proposed methods in three perspectives: a) The virtual examples are to generate virtual training data, whereas the proposed method is to generate pseudo test data; the scopes are different. b) The virtual examples are generated so that the learning machine will extract the invariances from the artificially enlarged training data [10], whereas our proposed method is to generate pseudo test data points so that the learning machine can combine them and enhance accuracy; the purposes are different. c) Virtual examples are designed specifically for SVM, whereas the proposed method is suitable for all kinds of classifiers; the scales are different.

Also Note that there are some fundamental differences between the MCS and the SMCS. In the MCS, we benefit from the fact that each classifier has a different perception of how a test sample should be classified. Because the decision boundary made by each classifier is different, there is diversity among decision boundaries drawn by different classifiers. Given that classifiers make different errors on different test samples, diversity can actually help improve classification accuracy. So, in the MCS, one of the core issues is to generate, select, and combine multiple classifiers, such that the combined decision boundary is better than any existing single boundary. In the SMCS, we not only try to find a better decision boundary, but one with the potential to be close to the oracle and not constrained by an existing classifier boundary and the number of classifiers. In designing the SMCS, we acknowledged the fact that a decision boundary drawn by a classifier might never be optimal; so, instead of refining several existing decision boundaries by combining them, we are trying to explore and make use of information in the neighborhood of a single decision boundary. In this way, we are looking for diversity that is already present in the neighborhood, rather than trying to benefit from diversity embedded in different classifiers. Consequently, diversity is extracted not from diverse decision boundaries, but from diverse pseudo test data points. The core issue is then to adequately generate, select, and combine multiple pseudo test data points for a test sample, rather than generating, selecting, and combining multiple classifiers.

We focus on two main questions in this paper:

1. Can we extract diversity from a dataset without training multiple classifiers?
2. Can multiple classification without multiple classifiers enhance accuracy?

2 Proposed Method

Given a training dataset \( \mathbf{X} \), we first divide the training samples into \( N \) reference samples \( \mathbf{X}_r = \{x_1, x_2, \ldots, x_N\} \), and \( M \) evaluation samples \( \mathbf{X}_e = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_M\} \), a single classifier \( C_X \) trained by all the available training samples \( \mathbf{X} \), and a test data point \( \tilde{x}_t \). The mechanism involves the creation of \( K \) pseudo test data points, \( \tilde{X}_t = \tilde{x}_{1,t}, \tilde{x}_{2,t}, \ldots, \tilde{x}_{K,t} \), which would result in \( K \) corresponding classification outputs \( \hat{y}_{1,t}, \hat{y}_{2,t}, \ldots, \hat{y}_{K,t} \) after being classified by \( C_X \).