Confidence Boosting: Improving the Introspectiveness of a Boosted Classifier for Efficient Learning

Rudolph Triebel Hugo Grimmett Ingmar Posner
Mobile Robotics Group, Dep. of Engineering Science, Univ. of Oxford, OX1 3PJ Oxford, UK
{rudi, hugo, ingmar}@robots.ox.ac.uk

Abstract— **This paper concerns the recently introduced notion of introspective classification. We introduce a variant of the point-biserial correlation coefficient (PBCC) as a measure to characterise the introspective capacity of a classifier and apply it to investigate further the introspective capacity of boosting – a well established, efficient machine learning framework commonly used in robotics. While recent evidence suggests that boosting is prone to providing overconfident classification output (i.e. it has a low introspective capacity), we investigate whether optimising this criterion directly leads to an improved introspective capacity. We show that with only a slight modification in the AdaBoost algorithm the resulting classifier becomes less confident when making incorrect predictions, rendering it significantly more useful when it comes to efficient robot decision making.**

I. Introduction

Machine learning algorithms are changing the face of mobile robotics. Their reach now extends to a significant number of robotic tasks including perception, planning, navigation, and manipulation. As a result, most modern mobile robotic systems already rely to a significant degree on machine learning methods. However, for these methods to be useful in robotics, some very specific requirements have to be beyond those commonly considered in other research areas. In particular, these requirements include efficiency in terms of memory requirements, computation time and energy consumption as well as plasticity and robustness in order to provide true long-term autonomy. Algorithms that particularly meet this latter criterion are often identified as *online* or *life-long* learning methods.

In this work, we focus on *boosting*, a machine learning framework commonly used in robotics both for its efficiency and often competitive classification performance. Here we investigate its usefulness in *mission-critical* applications, where a single error in the classification can have desastrous consequences for the entire mission. As an example, consider an autonomous car that fails to detect a red traffic light. As was shown by Grimmett *et al.* [1], these applications require, in addition to a low rate of false detections (especially false negatives), a classifier that is able to provide a realistic estimate of its classification *confidence* along with the predicted class label. Classifiers with this capability are denoted as *introspective*. In this work, we investigate this further and particularly address two main questions. Firstly, what could be a good measure of this relationship between confidence and correctness? And secondly, can we use such a measure to improve the introspective capabilities of one of the most prevalent classification algorithms in robotics? The preliminary findings we present here suggest that while boosting may not be as intrinsically introspective as, for example, a Gaussian Process Classifier (GPC) [1], its introspective capacity can be increased significanly by optimising it explicitly as part of the standard Boosting framework. We point out that we do not provide any theoretical proofs here, but instead present empirical results along the lines of those presented in [1].

Our modification specifically applies to the standard AdaBoost [2] algorithm. However, our findings are also likely to be replicable for other variants of boosting such as LogitBoost, GentleBoost [3], or robust variants [4].

A. Related Work

Apart from the seminal work on boosting in general [2]–[4], most of the references related to this work are already given in [1]. Boosting has a long and successful track record in mobile robotics (see, for example, the work of Martinez Mozos *et al.* [5], [6]). Introspection has been recently introduced by Grimmett *et al.* [1].

This paper further investigates the introspective capabilities of one particular classification framework.

II. Approach

In this section we first describe the standard binary AdaBoost [2] algorithm (see Algorithm 1). Then, we introduce Confidence Boosting, our new introspective variant of AdaBoost. It uses an empirical measure of “introspectiveness”, which is a new idea to quantify and assess this property in classifiers. We propose a variant of the point-biserial correlation coefficient (PBCC) for this measure, which we briefly explain.

A. Standard Boosting

The main principle of boosting is to assign weights to the n training data points $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ with $\mathbf{x}_i \in \mathbb{R}^d$ and to run a given number m of training rounds through the data, where at each training round a classifier is obtained that particularly focusses on the misclassified samples from the previous rounds. This is done by updating the data weights according to a classification loss function, which is usually the 01-loss. In more detail the steps are: first, the weights are all equally initialized with $1/n$. Then, in each round a *weak* classifier $f_i : \mathbb{R} \rightarrow \{-1, 1\}$ is learned from the weighted training data, and its training error ϵ_i is computed. Here, $I()$ denotes the identity function, which is 1 if the argument is true and 0 otherwise. From the training error the coefficient α_i is computed and the data weights are updated so that the misclassified points obtain a higher weight while the weights of the other points remain unchanged. The obtained coefficients α_i are then used to classify a new test datum $\mathbf{x}^*$ using the weighted sum $\sum_{i=1}^{m} \alpha_i f_i(\mathbf{x}^*)$, which is simply tested for its sign: if it is positive, the predicted class label is 1, otherwise it is -1.

Algorithm 1: AdaBoost for binary classification

```
  Data: training data $(\mathcal{X}, \mathbf{y})$ consisting of $n$ labeled
        feature vectors, where $y_j \in \{-1, 1\}$
  Input: Number $m$ of training rounds
  Output: coefficients $(\alpha_1, \ldots, \alpha_m)$
1 $\mathbf{w}^{(1)} \leftarrow (1/n, \ldots, 1/n)$
2 for $i \leftarrow 1$ to $m$ do
3     $f_i \leftarrow$ LearnWeakClassifier $(\mathbf{w}, \mathcal{X}, \mathbf{y})$
4     $\epsilon_i \leftarrow \frac{\sum_{j=1}^{N} w_j^{(i)} I(f_i(\mathbf{x}_j) \neq y_i)}{\sum_{j=1}^{N} w_j^{(i)}}$
5     $\alpha_i \leftarrow \ln\left(\frac{1-\epsilon_i}{\epsilon_i}\right)$
6     for $j \leftarrow 1$ to $n$ do
7         $w_j^{(i+1)} \leftarrow w_j^{(i)} \exp(\alpha_i I(f_i(\mathbf{x}_j) \neq y_j))$
8     end
9 end
```

B. Confidence Boosting

The main benefits of the boosting algorithm are its arbitrarily small training error (it decreases monotonically with the number of training rounds), and its very efficient inference step. However, as was shown in [1], in terms of introspection the standard boosting algorithm performs much worse than other classification algorithms such as the Gaussian Process classifier (GPC), which means that it tends to be overconfident in its class predictions. This can for example be seen when the algorithm is first trained on two classes, and then elements of a third, unseen class are presented in the classification step. In that case, the algorithm returns class predictions with a very high certainty, although all predicted class labels must be incorrect. As a consequence, standard boosting can not be used in situations where the class label uncertainty is needed for further processing, e.g. to detect potential misclassifications or for active learning. In the experimental section, we will give more evidence for this.

To address this issue, we first have a closer look at line 3 in Algorithm 1: Here, a weak classifier f_i is determined that assigns class labels to given input data. The only requirement for f_i to be a weak classifier is that the weighted training error ϵ_i computed in line 4 is not larger than 0.5. One simple example for a weak classifier, which is often used in boosting, is the *decision stump*, which operates on a projection of the data onto a single feature dimension k and determines a threshold θ and an orientation $s \in \{-1, 1\}$ so that most of the positively labeled training points are on the positive side of the resulting decision boundary, i.e.

$$|\{(x_j^k, y) \forall j = 1, \ldots, n \mid s(x_j^k - \theta) \geq 0\}| \geq \frac{n}{2}, \tag{1}$$

where k is a fixed dimension of the feature vector $\mathbf{x}_j$, and $|.|$ denotes the size of a set. To find a weak classifier f_i, one common method is to loop over all feature dimensions $k = 1, \ldots, d$ and to use the decision stump that provides the smallest weighted training error, i.e. ϵ_i is then the smallest over all dimensions. The benefit of this is that the number of training points that need re-weighting is smallest and that the overall training error decreases fast. However, for an introspective classifier, one is more interested in a realistic relation between a correct classification and one with low uncertainty, rather than in a low training error. In the next section, we will give more details how this "introspectiveness" relation can be formulized. For now, we just state that all decision stumps can be used as a weak classifier, because they all return a weighted training error less than 0.5. Thus, if we choose the one that is most introspective in a given sense, instead of the one with the smallest training error, then the strong classifier that results from boosting is more likely to be introspective, too. This is the main idea of confidence boosting.

Of course, this brings also some drawback: as we don't choose the optimal decision stump in terms of classification performance, the resulting strong classifier will usually also perform worse. However, this can be adressed by simply increasing the number of used decision stumps, because the training error still decreases in each training round, although at a slower rate. Thus, the aim of obtaining an introspective classifier is traded off with the need to reduce the training error. This suggests a weighted sum of classification performance and introspectiveness as an assessment method for decision stumps, however in this first version of the algorithm we only consider the introspective part to avoid introducing a parameter for the algorithm. To measure introspectiveness, we suggest a function similar to the point-biserial correlation coefficient, which is described next.

C. The Point-Biserial Correlation Coefficient

In many probabilistic reasoning applications the problem arises how to measure the relation between two random variables, and there are a number of different measures in the literature that can provide such a relation. In principle, these measures try to answer questions like: "how strong is the statistical dependence between the variables?", or "how much information does one variable give about the other?", or "how much are the variables correlated?". Examples of these measures are the Kullback-Leibler (KL-) divergence, the mutual information (MI), or the correlation coefficient. However, most of these measures require both random variables to be continuous, whereas in our case we want to relate the discrete, binary variable of "classification correctness" with the continuous variable "classification uncertainty". The intuition behind this is that a classifier that is very often correct when it is certain and only incorrect when it is uncertain should be denoted as introspective. One way to determine such a relation is by using the *Point-Biserial Correlation Coefficient* (PBCC), a variant of the standard correlation coefficient. The PBCC is defined as follows:

$$r_{pb} := \frac{\mu_1 - \mu_2}{\sigma_n} \sqrt{\frac{n_1 n_2}{n^2}}, \tag{2}$$

where μ_1 and μ_2 are the mean values of the continous variable for those parts of the data, for which the binary variable is either 0 or 1, respectively. In our case, these are the average uncertainties for the incorrectly and the correctly classified data points. Furthermore, σ_n is the standard deviation of the continuous variable, i.e. the classification uncertainty, and n_1 and n_2 are the numbers of incorrectly and correctly classified samples with $n = n_1 + n_2$.

While the PBCC provides a good measure of introspectiveness in cases where there are enough correctly and incorrectly classified samples, it has the drawback that its range decreases when theses numbers are very unbalanced, for example when there are no incorrectly classified samples. In that case, the PBCC is 0, although the correctly classified samples can all be very certain, in which case the classifier would be more introspective than the PBCC suggests. Therefore, in our experiments, we use a simpler version of the PBCC, which only considers the first term, i.e.:

$$r^*_{pb} := \frac{\mu_1 - \mu_2}{\sigma_n}. \tag{3}$$

In the following, we will refer to this measure as the simplified PBCC (sPBCC).

To summarize, the modified version of the function `LearnWeakClassifier` trains a decision stump for a projection of the training data onto each dimension $k = 1, \dots, d$ and chooses the one that maximises either r_{pb} or r^*_{pb}. This requires a measure of uncertainty from a class label prediction returned from a decision stump, but these only return either 0 or 1, depending on whether the feature is on the positive or the negative side of the decision boundary. To obtain a probability value, we apply a sigmoid function to the distance of the feature value from the decision boundary of the stump, specifically we use the cumulative Gaussian function. As a result, we obtain a probability of a given test point $\mathbf{x}_j$ to have label 1. From this probability we then compute the *normalized entropy* to obtain an uncertainty estimate for the predicted class label, as was already done in [1].

III. Experimental Results

The aim of our experiments is two-fold: first we see whether our simplified PBCC measure is consistent with our intuitive notion of introspection. Secondly, we show that ConfidenceBoost, our modified version of AdaBoost, leads to a more introspective classifier. Our feature selection is the same as presented in [1], namely a dictionary-based correlation of randomly chosen image patches (originally introduced by Torralba *et al.* [7]). We also use the GTSRB data set, which comprises images of road signs in urban environments.

Classifier	Precision	Recall	Accuracy	sPBCC
SE GPC	1.000	1.000	1.000	-0.720
RBF SVM	1.000	1.000	1.000	-0.959
Linear GPC	1.000	1.000	1.000	-0.863
Linear SVM	1.000	1.000	1.000	-1.270
LogitBoost	1.000	1.000	1.000	-196189.581
ConfBoost	1.000	0.995	0.997	9.113

TABLE I: Classification performance when separating *stop* sign from the *lorries prohibited* signs. In addition to precision, recall, and accuracy, we also report the sPBCC value, which quantifies the introspective capabilities of a classifier. We can see that the GP classifiers are more introspective than the SVMs according to that measure, which underlines our findings from [1]

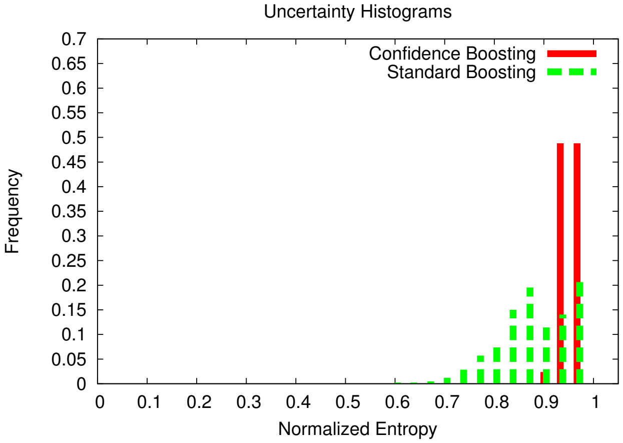

Fig. 1: Histogram of normalised entropies for the probabilities of the predicted class labels, where an unknown class was shown to the AdaBoost and the ConfidenceBoost classifier. Both classifiers are unconfident about the given class labels, but ConfidenceBoosting shows this to a larger extent. Note that the histograms are normalized so that the sum of all bins is 1.

In our first experiment, we compute the sPBCC value for the classifiers that were already used in [1]: the GPC and SVM, both with linear and squared-exponential (SE) kernels, LogitBoost, and our new ConfidenceBoost algorithm. We train each on 200 instances of *stop* and *lorry prohibited* sign, and test on an equally-sized set of the same classes. The results are shown in Table I.

We can see that both GPCs have a higher correlation between false classification and uncertainty than their SVM equivalents, as we would expect given their already established introspective capacity. We also see that the ConfBoost algorithm also performs very highly in this regard.

In the second experiment, we train both the standard AdaBoost and ConfidenceBoost on the same two classes of road sign, and test on a novel third class – we use the *roadworks ahead* sign – computing the histogram of normalised entropies for both. These histograms compare the distribution over the uncertainties in the class predictions of the classifiers, and can be seen in Fig. 1. They show that both classifiers are fairly uncertain about the predicted class labels, which is reasonable given that the presented data are from a class unseen during training. However, the labels returned from ConfidenceBoost are even more uncertain, leading to a more realistic assessment of the classification result. Hence, we can conclude that the ConfidenceBoost algorithm leads to a more introspective classifier.

IV. Conclusions and Future Work

The two contributions of this work are a way to quantify the introspectiveness of a classifier – a notion that only recently has been introduced, and a simple method to improve the introspective capabilities of the standard AdaBoost algorithm. However, many new questions arise from that, where one is that of a more theoretical foundation for the experimental results shown here. Another one addresses the implications of our findings in a more general way, such as: can this method be applied to other classification algorithms? For example, one could think of using other established methods for weak classification, thereby optimizing them in the introspective sense. Our preliminary results at least justify some further research along these lines.

Acknowledgment

This work is funded under the European Community's Seventh Framework Programme (FP7/2007-2013 V-CHARGE) under Grant Agreement Number 269916 and by the UK EPSRC Grant Number EP/J012017/1.

References

[1] H. Grimmett, R. Paul, R. Triebel, and I. Posner, "Knowing when we dont know: Introspective classification for mission-critical decision making," in *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2013, to appear.

[2] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, 1997.

[3] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[4] P. Li, "Robust logitboost and adaptive base class (abc) logitboost," in *Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, 2010.

[5] O. M. Mozos and W. Burgard, "Supervised learning of topological maps using semantic information extracted from range data," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, October 2006, pp. 2772–2777.

[6] O. M. Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard, "Supervised semantic labeling of places using information extracted from sensor data," *Robotics and Autonomous Systems (RAS)*, vol. 55, no. 5, pp. 391–402, May 2007.

[7] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing visual features for multiclass and multiview object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 854–869, May 2007.