OBLIQUE RANDOM FOREST BASED ON PARTIAL LEAST SQUARES APPLIED TO PEDESTRIAN DETECTION

Artur Jordão Lima Correia, William Robson Schwartz

Smart Surveillance Interest Group, Computer Science Department
Universidade Federal de Minas Gerais, Minas Gerais, Brazil

ABSTRACT

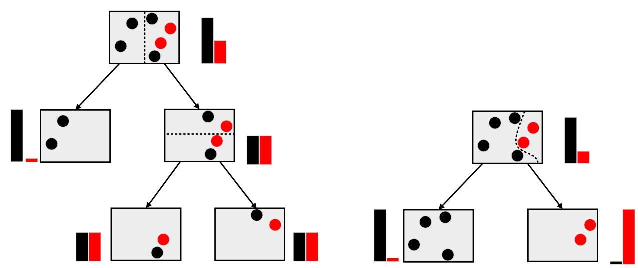
The increasing popularity of approaches based on random forest in computer vision tasks is due to its simplicity and flexibility with complex data. Random forest is a set of decision trees that can be divided in two subsets according to the view of the feature descriptors provided as input: orthogonal and oblique. In the former, the feature space is separated orthogonally (axis-aligned) by a single feature at a time. In the latter, it separates the space by oriented hyperplanes, which usually provides better data modeling. This work proposes a novel oblique random forest associated with Partial Least Squares to perform the oblique split. We validate the proposed approach, referred to as oRF-PLS, on the challenge INRIA Person dataset. Experimental results demonstrate that the proposed method outperforms traditional state-of-the-art detectors. In addition, we demonstrate that PLS is a more suitable choice to build oblique random forest than SVM, being faster and producing more accurate forests.

Index Terms— Pedestrian detection, random forest, oblique random forest, partial least squares, machine learning.

1. INTRODUCTION

Pedestrian detection is one of the most important computer vision tasks focusing on surveillance being used as a preliminary step for solving other problems, such as intelligent vehicles, person re-identification and advanced robotics. However, this task has many challenges, such as changing articulated pose, distinct illumination conditions, variance in clothing styles and frequent occlusion among pedestrians [1].

According to Benenson et al. [2], the most promising pedestrian detection methods are based on deep learning and random forest. Despite accurate, deep learning approaches (commonly convolutional neural networks) require a powerful hardware architecture and considerable amount of samples to learn a model. Moreover, the best results associated to such approaches are comparable with simpler methods. On the other hand, random forest approaches are able to run on simple CPU architecture and learn with a fewer amount of samples. The increasing number of studies based on this classifier is due several advantages that this approach presents including fast to test, probabilistic output and it naturally treats problems with more than two classes [3].



(a) (b)

Fig. 1. Decision tree split types, the bars represent the information gain. (a) Orthogonal split, tree with depth 2. (b) Oblique split, tree with depth 1 (best visualized in color).

Following the definition of Breiman [4], a random forest is a set of decision trees, in which the response is a combination of all tree responses of the forest. It can be classify according to the type of the decision tree that it is composed: orthogonal or oblique. In the former (Figure 1(a)), each tree node creates an axis-aligned boundary decision, in other words, it divides the data selecting an individual feature. Therefore, it usually does not provide accurate results since the data are not linearly separable in most of the cases. The latter ((Figure 1(b))) separates the data by oriented hyperplanes, providing better data separation and shallower trees [5].

Inspired by the aforementioned features, we propose a novel oblique random forest associated with Partial Least Squares (PLS), called *oRF-PLS*. PLS is a popular technique to dimensionality reduction and has been shown adequate to human detection [6, 7]. Schwartz et al. [6] employed the PLS purely as dimensionality reduction technique to project the high features space onto a low dimensional latent space and present these latent features to a Quadratic Discriminant Analysis (QDA) classifier discriminate humans from background. Different from [6], in this paper we propose to use PLS as regression method to find a boundary surface to separate positive and negative examples at each tree node (Figure 1(b)) that composes an oblique random forest.

To evaluate the gain provided by the proposed oRF-PLS, we utilize Support Vector Machine (SVM) to build the oblique random forest (oRF-SVM) and we compare our method with other traditional classifiers employed in pedestrian detection, e.g, linear SVM [8] and the QDA [6]. According to the experimental results, the proposed oRF-PLS is more accurate and statically faster than the oRF-SVM. Furthermore, it achieves comparable state-of-the-art results when compared with traditional methods based on HOG feature descriptors.

2. RELATED WORKS

In the last decade, several efforts have been performed to improve the features and classifiers to distinguish humans from background [1, 9, 2]. In the next paragraphs, we describe strategies that address these two issues.

Ko et al. [10] combined a cascade of random forests based on textural features. To extract texture information, the center-symmetric local binary patterns (CS-LBP) was employed, where each cascade level received a texture type provided by a different filter utilized to generate the CS-LBP descriptor. Inspired by [10], Koet et al. [11] proposed a modified cascade of random forests, in which the first stage employed Haar-like features and in the second stage, oriented center symmetric-local binary patterns (OCS-LBP) features. To improve the performance of their method, the authors delimited the search regions applying the divide-and-conquer paradigm on the scale pyramid.

Although texture is an important cue to find pedestrians in images, pose variation and occlusion often worsen the pedestrian detection results. Aiming at addressing such problems, Marin et al. [12] proposed an approach based on feature selectors. These feature selectors describe random patches of a window. For each feature selector, a linear SVM is learned, followed by thresholding operation that splits the data. To improve the detection performance, a bootstrapping procedure with the following steps was proposed. For each iteration, the current ensemble, F, searches for *hard negatives*, which consist of negative samples classified as positives (based on a thresholding). Afterwards, n new weak classifiers are trained and inserted into F. This procedure is repeated iteratively for k times. Due to accurate results achieved by this procedure, in this work we are using the same bootstrapping algorithm for oblique random forests.

In some computer vision tasks, a preprocessing step is applied on the data before applying the split procedure in the node [13, 14]. Nam et al. [13] demonstrated that oblique splits can generalize better the data than orthogonal split on raw features. Their work was inspired by [14], where a preprocessing on the features was proposed. This preprocessing consists in decorrelating the features, enabling the use of orthogonal splits instead oblique. Qiu et al. [15] also utilized a transformation on the features. The idea consists in learning a transformation matrix to maximize the variation inter-class and at the same time minimizes the intra-class variation, similarly to LDA. To learn this transformation, the authors used the nuclear norm, where each tree node uses the arriving sample as example to learn the transformation matrix.

Different from [15, 13], in this paper we are interested only on oblique random forests aiming at evaluating the PLS as an alternative to perform the tree splitting.

3. PROPOSED APPROACH

This section starts by describing the features of the oblique random forest as well as its building process. Then, we describe how to employ the PLS and SVM at the oblique random forest, respectively.

3.1. Oblique random forest

Figure 1 summarizes the main advantage provided by oblique random forest (oRF). As can be observed, the samples are separated by oriented hyperplanes (Figure 1 (b)), achieving a better partitioning of the space that induces shallower trees.

The steps performed to construct the decision trees that compose the oRF are the following. First, we employ feature selection on the data received by the tree. As noticed by Breiman [4], this technique ensures diversity among the trees, presenting an important contribution to improve the accuracy. In particular, the bagging mechanism also provides diversity on the random forest [4]. However, as reported by [3], several works are abandoning the use of such method. In this work, we discard the use of bagging since a considerable number of samples is required to build each oblique decision tree. Second, a starting node (root), Rj, is created with all data presented to it. The creation of a node estimates a decision boundary (hyperplane) to separate the presented samples according to their classes. Finally, the data samples are projected onto the estimated hyperplane and a threshold τ is applied on its projected values, splitting the samples between in two children (R_{jr}, R_{jl}). The samples below this threshold are sent to the left child, R_{jr}, and samples equal or above to the threshold are sent to its right child, R_{jr}. This procedure is recursively repeated until the tree reaches a specified depth or another stopping criterion.

To estimate the threshold that better separates the data samples, we employ the *gini index* as quality measure. The *gini index* is computed in terms of

$$\Delta L(R_j, s) = L(R_j) - \frac{\mid R_{jls} \mid}{\mid R_j \mid} L(R_{jls}) - \frac{\mid R_{jrs} \mid}{\mid R_j \mid} L(R_{jrs}), \tag{1}$$

where

$$L(R_j) = \sum_{i=1}^{K} p_i^j (1 - p_i^j), \tag{2}$$

in which $s \in S$ (S is a set of thresholds), K represents the class number and p_i^j is the label of class i at the node j. We choose *gini index* because it produces an extremely randomized forest [3].

Once the trees have been learned, given a testing sample v, each node sends it either to the right or to the left child according to the threshold applied to the projected sample. For a tree, the probability of a sample to belong to class c is estimated combining the responses of the nodes in the path from the root to the leaf that it reaches at the end. The prediction of the random forest to a given samples v is performed by aggregating the predictions of the trees by arithmetic average.

3.2. Oblique random forest with PLS and SVM

To build each node in an oblique decision tree associated with PLS, the samples P received by a node have its dimension reduced to a latent space $p\!-\!dimensional$ using PLS. The value to p is set by cross-validation (see Section 4.1). Subsequently, the regression coefficients β are estimated using

$$\beta = W(P^T W)^{-1} T^T y, \tag{3}$$

where W is a matrix of weights, in which the columns represent the contribution of each feature of the samples P according to their labels y and T is the matrix of the variables in the latent space. Finally, the best threshold to split the training data samples, is obtained using the *gini index* on the regression values given by $y_{v_i} = \bar{y} + \beta^T v_i$, where $\bar{y}$ represents the mean sample of y.

The difference to build the oRF-SVM is that the received samples data do not have their dimensionality reduced and instead compute the regression coefficients a linear SVM[1] is learned at each tree node. The remaining of the process is equal. This way, the approaches can be compared only in terms of better data separation and generalization.

4. EXPERIMENTAL EVALUATION

To measure the detection accuracy, we employed the standard protocol evaluation used by state-of-the-art called *reasonable set* (a detailed discussion regarding this protocol of evaluation can be find in [16, 9]). This protocol measure the log-average miss rate of the area under the curve on the interval from 10^{-2} to 10^0 (low values are better). However, in some experiments we report the results using the interval from 10^{-2} to 10^{-1}. The area under curve in this interval represents a very low false positive rate (that is a requirement to applications that may cause damage to environment as well to humans, e.g., surveillance, robotics and transit safety), thus we evaluate the methods under a more rigorous detection.

At the calibration stage of the oRFs parameters we utilized the TUD pedestrian [17] dataset as validation set. To

[1]We are using linear SVM because it has been shown appropriate to pedestrian detection [8, 9, 2].

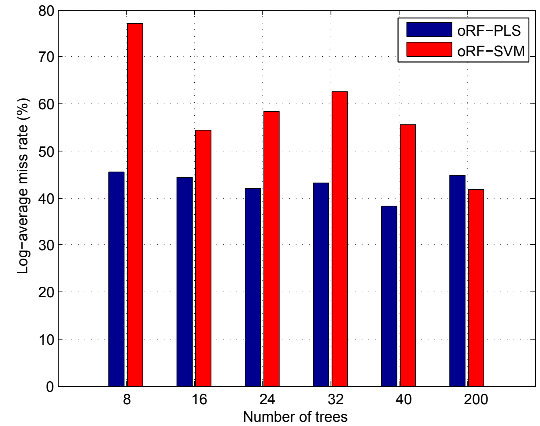


Fig. 2. Log-average miss-rate (low is better) on the validation set in function of the number of trees.

compare our approach with state-of-the-art detectors we are using the traditional and challenger INRIA dataset [8].

Through of the experiments, we are using the HOG features following the setup proposed by Dalal and Triggs[8], where the descriptor size is 3780 features. We present these features to both oRFs providing a comparison not influenced by the features.

4.1. Tree parameters

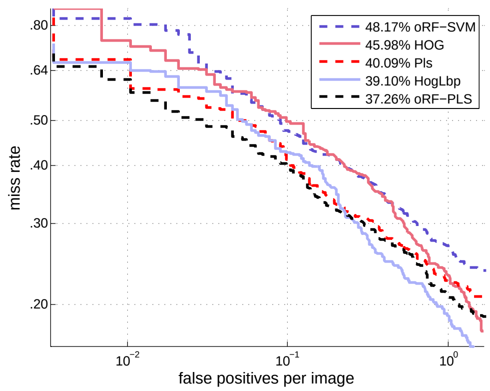
In this experimental validation, we focus on the impact of two factors in our forests: numbers of trees and number of selected features. We are using the term nF to denote the number of features randomly selected (from 3780 features available) to create a tree node (as explained in Section 3.1).

To both the oRFs (based on PLS and SVM), the maximum depth allowed at the growing stage of the tree is 5. In some preliminary experiments, we noticed that increasing the depth does not improve the gain considerably. On the validation dataset, the best parameters to oRF-SVM were using 200 trees and nF = 400, where it achieved a log-average miss rate of 41.67%. The oRF-PLS obtained the best results with 40 trees and nF = 550, with the log-average miss rate of 38.18%.

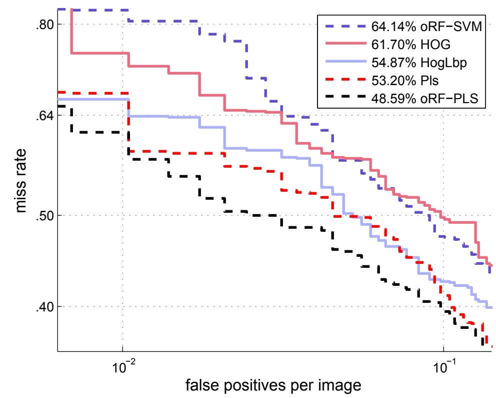
Differently from oRF-SVM, the oRF-PLS has an extra parameter to be tuned, the dimensionality reduction value, p, required by PLS technique (see Section 3.2). On the validation dataset, the best value found to p was of 6. In our experiments, we noticed that varying p slightly, increases substantially the log-average miss rate. For instance, modifying p of 6 to 8 the log-average miss rate goes from 38.18% to 42.98%. Therefore, this is a crucial parameter for oRF-PLS.

4.2. Influence of the number of trees

Figure 2 shows the log-average miss rate obtained by each approach on the validation dataset, according to number of trees composing the forest. As can see noticed, at the same



(a)



(b)

Fig. 3. Comparison of our oRF-PLS approach with the state-of-the-art. (a) Results with log-average miss-rate in the interval from 10^{-2} to 10^{0} (standard protocol). (b) Results with log-average miss-rate in the interval from 10^{-2} to 10^{-1}.

number the trees (except 200), the detection accuracy of oRF-PLS outperforms the oRF-SVM. To achieve competitive results, the oRF-SVM demands a considerable number of trees, which renders an extremely high computational cost (see Section 4.3). In addition, by computing the standard deviation of the log-average miss rate, we can notice that the oRF-SVM is more sensitive to variation of the number of trees to presenting a standard deviation of 10.58% while our proposed method presented a standard deviation of 2.42%.

According to the results in Figure 2, the use of PLS to build oRF is more adequate than use the SVM, producing smaller and more accurate forests.

4.3. Time issues

In this experiment, we show that the proposed oRF-PLS is faster than oRF-SVM. For this purpose, we performed a statistical test between the time (in seconds) to run the complete pipeline detection on an image of 640×480 pixels. To each approach, we execute the pipeline 10 times and compute its confidence interval (IC) using 95% of confidence. The oRF-PLS obtained an IC of $[270.2, 272.44]$ against $[382.72, 392.72]$ to oRF-SVM. As can be observed, the confidence intervals does not overlap, showing that the methods present statistical differences at the execution time. This result confirms that even the oblique decision tree based on SVM is faster than the oblique decision tree based on PLS, its forest becomes slower because more trees are required to provide an accurate detection.

4.4. Comparison with state-of-the-art

Our last experiment compares the proposed oRF-PLS with traditional state-of-the-art pedestrian detectors. As showed in Figure 3, our proposed method outperforms common classifiers used in pedestrian detection. Besides, when compared to oRF-SVM, our method outperforms in 10.91 and 15.55 percentage points (p.p) using the area in 10^{-2} to 10^{0} and 10^{-2} to 10^{-1}, respectively. A substantial improvement. In addition, the oRF-PLS was able to outperform a robust partial occlusion method, the HOG+LBP [18], in 1.84 and 6.28 p.p to the area in 10^{-2} to 10^{0} and 10^{-2} to 10^{-1}, respectively.

Regarding to results shown in this section, we conclude that the proposed oRF-PLS is able to obtain better results when compared with traditional classifiers applied to pedestrian detection.

5. CONCLUSIONS AND FUTURE WORK

This work presented a novel type of oblique random forest, the oRF-PLS. The proposed method employs the PLS as alternative to find a boundary decision which splits the data in each node at the decision tree. Experimental results demonstrated that shallower trees and a smaller forest are produced when using the PLS compared to the ones produced using SVM. Furthermore, our proposed method outperformed traditional classifiers used in pedestrian detection. As future work, we intend apply a random forest cascade with different features and use the PLS as transformation on raw features before present himself to orthogonal forest, as well as, extend the method to other object recognition tasks.

6. ACKNOWLEDGMENTS

The authors would like to thank the Brazilian National Research Council – CNPq (Grant #477457/2013-4), the Minas Gerais Research Foundation – FAPEMIG (Grants APQ-01915-14 and PPM-00025-15) and the Coordination for the Improvement of Higher Education Personnel – CAPES (DeepEyes Project).

7. REFERENCES

[1] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, 2009.

[2] R. Benenson, M. Omran, J. Hosang, , and B. Schiele, "Ten years of pedestrian detection, what have we learned?," in *ECCV, CVRSUAD workshop*, 2014.

[3] A. Criminisi and J. Shotton, *Decision Forests for Computer Vision and Medical Image Analysis*, Springer Publishing Company, Incorporated, 2013.

[4] Leo Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[5] Bjoern H. Menze, B. Michael Kelm, Daniel Nicolas Splitthoff, Ullrich Köthe, and Fred A. Hamprecht, "On oblique random forests.," in *ECML/PKDD (2)*, 2011.

[6] William Robson Schwartz, Aniruddha Kembhavi, David Harwood, and Larry S. Davis, "Human detection using partial least squares analysis," in *ICCV 2009*, 2009.

[7] Victor Hugo Cunha de Melo, Samir Leao, David Menotti, and William Robson Schwartz, "An optimized sliding window approach to pedestrian detection.," in *ICPR*, 2014.

[8] Navneet Dalal and Bill Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proc. of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, vol. 1 of *CVPR '05*, pp. 886–893.

[9] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona, "Pedestrian detection: An evaluation of the state of the art," *PAMI*, vol. 34, 2012.

[10] Byoung Chul Ko, Deok-Yeon Kim, Ji-Hoon Jung, and Jae-Yeal Nam, "Three-level cascade of random forests for rapid human detection," *Optical Engineering*, vol. 52, no. 2, pp. 027204–027204, 2013.

[11] Byoung Chul Ko, Mira Jeong, and JaeYeal Nam, "Fast human detection for intelligent monitoring using surveillance visible sensors," *Sensors*, vol. 14, no. 11, pp. 21247–21257, 2014.

[12] Javier Marín, David Vázquez, Antonio M. López, Jaume Amores, and Bastian Leibe, "Random forests of local experts for pedestrian detection," in *ICCV*, 2013.

[13] Woonhyun Nam, Piotr Dollár, and Joon Hee Han, "Local decorrelation for improved pedestrian detection," in *NIPS*, 2014.

[14] Bharath Hariharan, Jitendra Malik, and Deva Ramanan, "Discriminative decorrelation for clustering and classification.," in *ECCV (4)*, 2012, pp. 459–472.

[15] Qiang Qiu and Guillermo Sapiro, "Learning transformations for classification forests," 2013, vol. abs/1312.5604.

[16] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona, "Pedestrian detection: A benchmark," in *CVPR*, 2009.

[17] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele, "People-tracking-by-detection and people-detection-by-tracking.," in *CVPR*, 2008.

[18] Xiaoyu Wang, Tony X. Han, and Shuicheng Yan, "An HOG-LBP human detector with partial occlusion handling," in *ICCV*, 2009.