Feature Set Search Space for FuzzyBoost learning*

Plinio Moreno, Pedro Ribeiro, and José Santos-Victor

Instituto de Sistemas e Robótica & Instituto Superior Técnico
Portugal
{plinio,pedro,jasv}@isr.ist.utl.pt

Abstract. This paper presents a novel approach to the weak classifier selection based on the GentleBoost framework, based on sharing a set of features at each round. We explore the use of linear dimensionality reduction methods to guide the search for features that share some properties, such as correlations and discriminative properties. We add this feature set as a new parameter of the decision stump, which turns the single branch selection of the classic stump into a fuzzy decision that weights the contribution of both branches. The weights of each branch act as a confidence measure based on the feature set characteristics, which increases the accuracy and robustness to data perturbations. We propose an algorithm that consider the similarities between the weights provided by three linear mapping algorithms: PCA, LDA and MMLMNN [14]. We propose to analyze the row vectors of the linear mapping, grouping vector components with very similar values. Then, the created groups are the inputs of the FuzzyBoost algorithm. This search procedure generalizes the previous temporal FuzzyBoost [10] to any type of features. We present results in features with spatial support (images) and spatio-temporal support (videos), showing the generalization properties of the FuzzyBoost algorithm in other scenarios.

1 Introduction

Boosting algorithms combine efficiency and robustness in a very simple and successful strategy for classification problems. The advantages of this strategy have led several works to improve the performance of boosting on different problems by proposing modifications to the key elements of the original AdaBoost algorithm [5]: (i) the procedure to compute the data weights, (ii) the selection of the base classifier and (iii) the loss function it optimizes.

The focus of this work is the careful selection of the weak (base) classifier. Since the weak classifier could be any function that performs better than chance, the choice of the weak classifiers is usually motivated by the particular context of the problem. When the objective is to find meaningful sets of data samples,

* This work was supported by FCT (ISR/IST plurianual funding through the PIDDAC Program) and partially funded by EU Project First-MM (FP7-ICT-248258) and EU Project HANDLE (FP7-ICT-231640).

several dimensions of the original samples are gathered to build augmented base classifiers. This approach has been followed by the SpatialBoost [2], the TemporalBoost [11] and the temporally consistent learners used in [10]. A general drawback of these works is the specificity of the application for which they are constructed. The aim of this work is to generalize the fuzzy decision function of [10] to any type of data, while maintaining its main advantages. The generalized fuzzy decision function selects jointly the usual parameters of a decision stump and the set of features to use, a procedure that turns the single branch selection of the decision stump into a linear combination of the branches. Such a combination of the stump branches is commonly referred to as a fuzzy decision on [7,6,12]. Moreover, [8] shows empirically that the fuzzy tree decreases the variance and consequently improves the classification output.

The generalization of the fuzzy decision function brings a difficult problem to solve: the selection of the feature set. Exhaustive search is prohibitive, so we propose an algorithm that extracts the feature sets from (linear) dimensionality reduction techniques. These techniques map the original feature space to a more meaningful subspace, by the minimization of a cost function. Thus, the linear mapping contains relevant information about the similarity between feature dimensions on the original space. We analyze the rows of the linear mapping, selecting the components with very similar values and disregarding components with very low values. We consider three dimensionality reduction algorithms: Principal Components Analysis (PCA), Linear Discriminant Analysis (LDA) and Multiple Metric Learning for large Margin Nearest Neighbor (MMLMNN) Classification [14]. We apply the feature set search for fuzzy decision stumps in two data domains: spatial (face and car detection) and spatio-temporal (moving people and robots).

2 The FuzzyBoost algorithm

Boosting algorithms provide a framework to sequentially fit additive models in order to build a final strong classifier, $H(x_i)$. The final model is learned by minimizing, at each round, the weighted squared error

$$J = \sum_{i=1}^{N} w_i (y_i - h_m(x_i))^2, \tag{1}$$

where $w_i = e^{-y_i h_m(x_i)}$ are the weights and N the number of training samples. At each round, the optimal weak classifier is then added to the strong classifier and the data weights adapted, increasing the weight of the misclassified samples and decreasing correctly classified ones [13].

In the case of GentleBoost it is common to use simple functions such as decision stumps. They have the form $h_m(x_i) = a\delta\left[x_i^f > \theta\right] + b\delta\left[x_i^f \leq \theta\right]$, where f is the feature index and δ is the indicator function (i.e. $\delta[condition]$ is one if *condition* is *true* and zero otherwise). Decision stumps can be viewed as decision trees with only one node, where the indicator function sharply chooses branch a

or b depending on threshold θ and feature value x_i^f. In order to find the stump at each round, one must find the set of parameters $\{a, b, f, \theta\}$ that minimizes J w.r.t. h_m. A closed form for the optimal a and b are obtained and the value of pair $\{f, \theta\}$ is found using an exhaustive search [13].

2.1 Fuzzy weak learners optimization

We propose to include the feature set as an additional parameter of the decision stump, as follows:

$$h_m^*(x_i) = \frac{1}{||F||}\left(a\, F^T \delta\left[x_i > \theta\right] + b\, F^T \delta\left[x_i \leq \theta\right]\right), \tag{2}$$

where $x_i \in \mathbb{R}^d$ and the vector $F \in \mathbb{Z}_2^m, m \in \{1, \ldots, d\}$, so the non-zero components of F define a feature set. The vector F chooses a group of original sample dimensions that follow the indicator function constraints of Eq. (2) in order to compute the decision stump $h_m^*(x_i)$. Note that by choosing $m = 1$, Eq. (2) becomes the classic decision stump and the selection of different m values induce different decision stump functions. In this work we choose $F \in \mathbb{Z}_2^d$, thus we do not assume any a priori information about the samples x. Eq. (2) can be rearranged in order to put a and b in evidence,

$$h_m^*(x_i) = a\,\frac{F^T \delta\left[x_i > \theta\right]}{||F||} + b\,\frac{F^T \delta\left[x_i \leq \theta\right]}{||F||}. \tag{3}$$

From Eq. 3 it is easier to see that the the selector F is replacing the indicator function (i.e. a true or false decision) by an average of decisions. The new functions are:

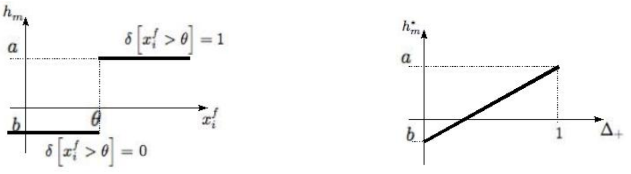
$$\mathit{\Delta}_+(x_i, \theta, F) = \frac{F^T \delta\left[x_i > \theta\right]}{||F||}, \quad \mathit{\Delta}_-(x_i, \theta, F) = \frac{F^T \delta\left[x_i \leq \theta\right]}{||F||}, \tag{4}$$

and they compute the percentage of features selected by F that are above and below the threshold θ. The functions $\mathit{\Delta}_+$ and $\mathit{\Delta}_- = 1 - \mathit{\Delta}_+$ of Eq. 4 sample the interval [0 1] according to the number of features selected by F (i.e. according to $||F||$). For example, if $||F|| = 2$ this yields to $\mathit{\Delta} \in \{0,\ 1/2,\ 1\}$, if $||F|| = 3$ to $\mathit{\Delta} \in \{0,\ 1/3,\ 2/3,\ 1\}$ and so on. The new weak learners, the fuzzy decision stumps, are expressed as $h_m^*(x_i) = a\mathit{\Delta}_+ + b\mathit{\Delta}_-$.

We illustrate in Fig. 1 the difference between the classic decision stumps and our proposed fuzzy stumps. The response of the decision stump is either a or b according to the feature point x_i^f, while the fuzzy stump response is a linear function of $\mathit{\Delta}_+$ that weights the contribution of the decisions a and b, thus the name fuzzy stump.

Replacing the fuzzy stumps of Eq. 3 in the cost function (Eq. 1), the optimal decision parameters a and b are obtained by minimization,

$$a = \frac{\bar{\nu}_+\bar{\omega}_- - \bar{\nu}_-\bar{\omega}_\pm}{\bar{\omega}_+\bar{\omega}_- - \left(\bar{\omega}_\pm\right)^2}, \qquad b = \frac{\bar{\nu}_-\bar{\omega}_+ - \bar{\nu}_+\bar{\omega}_\pm}{\bar{\omega}_+\bar{\omega}_- - \left(\bar{\omega}_\pm\right)^2}, \tag{5}$$



(a) (b)

Fig. 1. Response of the weak learners: (a) decision stumps and (b) fuzzy stumps.

with $\bar{\nu}_+ = \sum_i^N w_i y_i \Delta_+^T$, $\bar{\nu}_- = \sum_i^N w_i y_i \Delta_-^T$,
$\bar{\omega}_+ = \sum_i^N w_i \Delta_+^T$, $\bar{\omega}_- = \sum_i^N w_i \Delta_-^T$, $\bar{\omega}_\pm = \sum_i^N w_i \Delta_-^T \Delta_+^T$.

There is no closed form to compute the optimal θ and F, thus exhaustive search is usually performed. Although finding the optimal θ is a tractable problem, the search for the best F is NP-hard thus generally impossible to perform. This problem can be viewed as a feature selection problem with the objective of choosing, at each boosting round, the set of features that minimizes the error. In order to guide the search and reduce the number of possible combinations we apply dimensionality reduction algorithms in the original feature space, using the projection matrix in order to find feature set candidates.

2.2 The search space for the feature set

Linear dimensionality reduction techniques aim to find a subspace where regression and classification tasks perform better than in the original feature space. These feature extraction methods aim to find more meaningful and/or discriminative characteristics of the data samples by minimizing task-defined cost functions. Finally, the original features are substituted by the transformed ones in order to perform classification.

Although the dimensionality reduction methods differ in the way that they use labeled or unlabeled data to compute the linear transformation of the input, the projection vectors of all methods code the way that the original feature space should be combined to create a new dimension. Since the linear mapping contains relevant information about the correlations between dimensions of the original feature space, we propose to analyze each projection vector of the mapping by selecting vector components with similar values. Our rationale follows the weight similarity approach: if the weight of a dimension in the projection vector is similar to other dimension(s), this implies some correlation level between those dimensions. We apply this idea to three projection algorithms: PCA, LDA and MMLMNN.

The idea behind PCA is to compute a linear transformation $x^* = Lx$ that projects the training inputs into a variance-maximizing subspace. The linear transformation L is the projection matrix that maximizes the variance of the projected inputs, and the rows of L are the leading eigenvectors of the input data covariance matrix.

LDA uses the labels information to maximize the amount of between-class variance relative to the amount of within-class variance. The linear transformation $x^* = Lx$ outputted by LDA is also a projection matrix, and the rows of L are the leading eigenvectors of the matrix computed as the ratio of the between-class variance matrix and the within-class variance matrix. Unlike PCA, LDA operates in a supervised setting, restricting the number of linear projections extracted to the number of classes present in the problem.

The recently proposed MMLMNN method [14] attempts to learn a linear transformation $x^* = Lx$ of the input space, such that each training input should share the same labels as its k nearest neighbors (named target neighbors) and the training inputs with different label (named impostors) should be widely separated. This two terms are combined into a single loss function that has the competing effect of attracting target neighbors on one hand, and repel impostors on the other (see [14] for details).

Computing *F* from *L* Given a linear mapping L computed by PCA, LDA or MMLMNN, we scale the values of the projection matrix as follows: $\mathcal{L}_{ij} = \frac{|L_{ij}|}{\max(L)}$. The scaling ensures that $0 < \mathcal{L}_{ij} \leq 1$, which allow us to define lower thresholds (s_0 in Alg. 1) and number of intervals (n_s in Alg. 1) that have the same meaning for all the linear mappings. The algorithm for generating the feature sets is as follows:

```
  input : $s_0$ lower threshold, $n_s$ number of intervals, $\mathcal{L}$ projection matrix
  output: $F_j \quad j = 1 \ldots n_s$
1 for each projection (row) vector $\mathcal{L}_i$ do
2     compute $\Delta_s = (\max(\mathcal{L}_i) - s_0)/n_s$;
3     for $j = 1 \ldots n_s$ do
4         compute $s_j = s_0 + (j-1)\Delta_s$;
5         $F_j = \delta[s_j \leq \mathcal{L}_i < s_j + j\Delta_s]$;
6     end
7 end
```

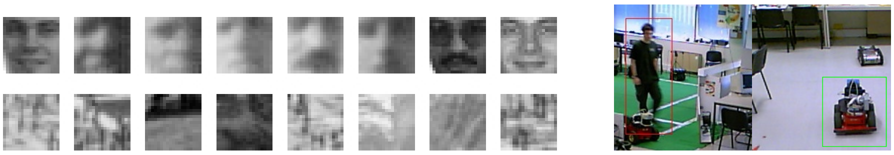
Algorithm 1: Generation of feature sets F of Eq. (2) from a scaled linear mapping $\mathcal{L}$

The lower threshold $s_0 \in [0, 1[$ removes components of $\mathcal{L}_i$ having low projection weights, which are the less meaningful dimensions. The number of intervals $n_s \in \mathbb{N}$ defines the criterion to group dimensions with similar weights (line 5 of Alg. 1), so a high number of intervals will group a few dimensions and a low number of intervals will generate a larger feature set F_j. In order to see the effect of several choices of s_0 and n_s, we apply the Algorithm 1 using several pairs (s_0, n_s) for each linear mapping L.

3 Experimental results

We evaluate the recognition rate difference between the decision stumps and the fuzzy stumps on two binary problems: (i) face vs. background and (ii) people vs.

robot discrimination. We use the CBCL MIT face database [1] and the people vs. robot database introduced in [4]. Figure 2 shows some examples of each class for both datasets.



(a) CBCL MIT face dataset (b) People vs. robot

Fig. 2. Positive and negative examples of the datasets used in this paper

Parameter selection of the feature search We define a set of pairs (s_0, n_s) for each linear mapping L in order to see the effect of the parameter selection in the performance of the generated feature sets in the FuzzyBoost algorithm. We set three low thresholds $s_0 \in \{0.1, 0.2, 0.3\}$, and for each s_0 we set three number of intervals, as follows: (i) $(0.1, 9)$, $(0.1, 18)$ and $(0.1, 27)$ for the first s_0, (ii) $(0.2, 8)$, $(0.2, 16)$ and $(0.2, 24)$ for the second s_0 and (iii) $(0.3, 7)$, $(0.3, 14)$ and $(0.3, 21)$ for the third s_0. The rationale behind this choice is to have Δ_s intervals with the same length across the different s_0 values, which allows to evaluate the pairs (s_0, n_s) fairly. For each pair (s_0, n_s), we apply the Alg. 1 on the three projection methods in order to generate the feature sets F. Then, F is applied on the weak learner selection of Eq. (2) in a fixed number of rounds $M = 1000$. The quantitative evaluation is the maximum recognition rate attained on the testing set.

Faces database The feature vector in this problem is constructed with the raw images (19x19 pixels). This is a high dimensional space, where the linear mappings are not able to find feature sets that provide a large improve when compared to GentleBoost. Nevertheless, Figure 3 shows that MMLMNN performs better than GentleBoost for most of the tests and a lot better than its competitors PCA and LDA.

Robot versus people database We apply two types of features: The weighted histogram of the Focus Of Attention (FOA) feature [9] and the Motion Boundary Histogram (MBH) [3] using a polar sampling grid cell. The spatio-temporal volume is constructed by stacking the feature vector of the current frame with the vectors of the previous four frames. The FOA feature is a $64d$ vector per frame, and the MBH is a $128d$ feature vector per frame. Thus, the spatio-temporal feature based on MBH lies in a very high dimensional space. Figs. 4 and 5 show the results of the FOA and MBH features respectively. We notice the same

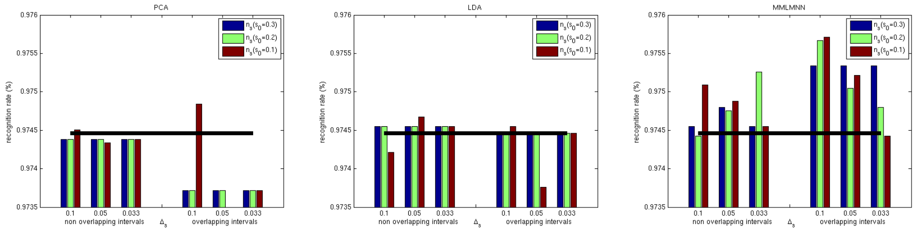


Fig. 3. Recognition rate of the FuzzyBoost algorithm for feature sets generated from: PCA, LDA and MMLMNN

trend of the previous tests, where MMLMNN performs better than the other dimensional reduction algorithms. Remark also the large gap improvement when using the FOA feature against the GentleBoost in Fig. 4. On the other hand, little improvement is achieved with the MBH feature. We believe that is more difficult to find the right spatio-temporal groupings in this feature space and it looks like the simple stacking of the feature vectors is able to attain good classification results. On the other hand, the FuzzyBoost is able to improve the performance of the FOA feature even higher than the GentleBoost with the stacked MBH features of five frames.

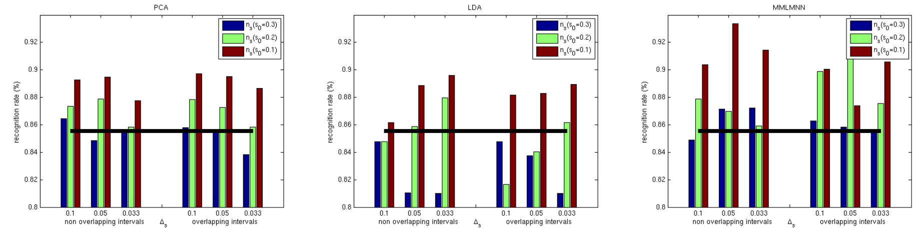


Fig. 4. Recognition rate of the FuzzyBoost algorithm for feature sets generated from: PCA, LDA and MMLMNN, FOA weighted histogram over 5 frames

4 Conclusions

We introduce the generation of appropriate feature sets for the FuzzyBoost algorithm. The algorithm for feature set generation analyzes the row vectors of any linear dimensionality reduction algorithm in order to find feature dimensions with similar vector components. We generalize the formulation of the fuzzy decision stump, which now can be applied to any learning problem. We present two types of domains where the fuzzy decision stump brings robustness and generalization capabilites, namely: face recognition and people vs. robot detection by their motion patterns.

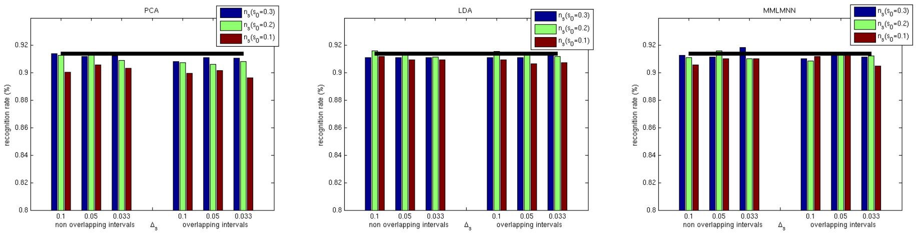


Fig. 5. Recognition rate of the FuzzyBoost algorithm for feature sets generated from: PCA, LDA and MMLMNN, MBH feature over 5 frames

References

1. CBCL face database #1, MIT center for biological and computation learning, `http://cbcl.mit.edu/projects/cbcl/software-datasets/FaceData2.html`
2. Avidan, S.: Spatialboost: Adding spatial reasoning to adaboost. In: In Proc. European Conf. on Computer Vision. pp. 386–396 (2006)
3. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: European Conference on Computer Vision (2006)
4. Figueira, D., Moreno, P., Bernardino, A., Gaspar, J., Santos-Victor, J.: Optical flow based detection in mixed human robot environments. In: Proc. of ISVC (2009)
5. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)
6. Jang, J.S.R.: Structure determination in fuzzy modeling: a fuzzy cart approach. In: Proceedings IEEE Conference on Fuzzy Systems. pp. 480–485 vol.1 (Jun 1994)
7. Janikow, C.Z.: Fuzzy decision trees: issues and methods. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 28(1), 1–14 (Feb 1998)
8. Olaru, C., Wehenkel, L.: A complete fuzzy decision tree technique. Fuzzy Sets and Systems 138(2), 221–254 (2003)
9. Pla, F., Ribeiro, P.C., Santos-Victor, J., Bernardino, A.: Extracting motion features for visual human activity representation. In: Proceedings of the IbPRIA'05 (2005)
10. Ribeiro, P.C., Moreno, P., Santos-Victor, J.: Boosting with temporal consistent learners: An application to human activity recognition. In: Proc. of ISVC. pp. 464–475 (2007)
11. Smith, P., da Vitoria Lobo, N., Shah, M.: Temporalboost for event recognition. In: International Conference on Computer Vision. vol. 1, pp. 733– 740 (October 2005)
12. Suarez, A., Lutsko, J.F.: Globally optimal fuzzy decision trees for classification and regression. IEEE Transactions on PAMI 21(12), 1297–1311 (Dec 1999)
13. Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing visual features for multiclass and multiview object detection. IEEE PAMI 29(5), 854–869 (2007)
14. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. 10, 207–244 (June 2009)