WEIGHTED ENSEMBLE BOOSTING FOR ROBUST ACTIVITY RECOGNITION IN VIDEO

Yuri Ivanov
Mitsubishi Electronic Research Lab
201 Broadway,
Cambridge, MA 02139
yivanov@merl.com

Raffay Hamid
College of Computing
Georgia Institute of Technology
Atlanta GA 30318
raffay@cc.gatech.edu

September 3, 2006

Abstract. In this paper we introduce a novel approach to classifier combination, that we term *Weighted Ensemble Boosting*. We apply the proposed algorithm to the problem of activity recognition in video, and compare its performance to different classifier combination methods. These include Approximate Bayesian Combination, Boosting, Feature Stacking, and the more traditional Sum and Product rules. Our proposed Weighted Ensemble Boosting algorithm combines Bayesian averaging strategy with the boosting framework, finding useful conjunctive feature combinations and achieving lower error rate than the traditional boosting algorithm. The method demonstrates a comparable level of stability with respect to the classifier selection pool. We show the performance of our technique with a set of 6 types of classifiers in an office setting detecting 7 classes of typical office activities.

Key words: Activity Recognition, Ensemble Learning, Computer Vision

1. Introduction and Previous Work

Recognition of human activities has played a central role in the design of intelligent surveillance system and high-level computer vision in general [4, 8]. One of the main difficulties in these areas remains the fact that no universal set of events exists that describes all aspects of our lives. This implies that the designer of an intelligent vision system needs to have at his disposal a convenient way to formulate events of interest that can be built up from smaller more general components. Since the set of events may not be known *a priori*, a mechanism for combining these general features is necessary to produce the final classifier.

One possible path towards such a mechanism is to explore techniques of classifier combination. The main purpose of combining classifiers (from hereon referred to as "weak classifiers") is to pool their individual outputs to produce a "strong classifier" that is more robust and accurate than each individual weak classifier.

Techniques for classifier combination are well explored in the literature (see *e.g.* [1, 3, 7], and the references therein). Due to their simplicity, sum, voting and product combination rules have been successfully used and analyzed by Pekalska [3] and Kittler [7], among others. Bilmes and Kirchhoff [6] note that the product rule is optimal when the classifiers in the ensemble are correlated, while sum rule is preferred if they are not. Combination techniques have been proposed in the past that investigate the Bayesian view of evidence integration [14].

A different approach to classifier combination is taken by Boosting, which has been profoundly successful in vision applications of face detection and recognition [10]). Recently, there have been efforts to extend the Boosting framework to incorporate temporal features, such as pedestrian detection by Viola and Jones [11] and temporal boosting by Smith *et. al* [9].

Our work continues on this trend, augmenting the greedy boosting framework with a robust conjunctive feature selection mechanism, which exploits correlations between the individual classifiers. In this work we introduce a novel classifier combination method - Weighted Ensemble Boosting. We train 6 multi-class event classifiers to detect 7 typical office activities and study the effect of the combination method on the final classifier performance.

2. Methods for classifier combination

This section describes several techniques for combining classifiers and introduces a novel *Weighted Ensemble Boosting* combination method. Throughout this paper it is assumed that the output of each weak classifier is represented by a posterior probability. That is, each classifier outputs a vector of probabilities of each class, $P_k(\tilde{\omega}|x)$, where k is the classifier index, and $\tilde{\omega}$ a class label. Each combination method accepts a set of K vectors $P_k(\tilde{\omega}|x)$ and outputs a distribution $P(\omega|x)$, a function of the individual classifier outputs, which is the basis for the final decision. In the remainder of this section we briefly describe the combination methods that we will be using, followed by the introduction of the main contribution of this paper in Section 3.

2.1. Feature Stacking

Feature Stacking uses a "super-classifier" trained on the outputs of weak classifiers stacked into a single vector. That is, the input for the top level classifier, $\tilde{X}$ is formed as follows:

$$\tilde{X} = \left(P_1(\omega|x)^T, P_2(\omega|x)^T, \ldots P_K(\omega|x)^T\right)^T \tag{1}$$

Then the classifier is trained on the pairs of data, (X_i, Y_i), where Y_i is the class label of the i-th data point. We show results of this approach using an RBF-kernel Support Vector Machine in the experimental section.

2.2. Approximate Bayesian Combination

Some of the previously work uses a measure of the classifier confidence to weigh predictions of each classifier for each of the classes as the Approximate Bayesian combination rule [15] [14]. This confidence is expressed by the distribution $P(\omega|\tilde{\omega})$ (a matrix), computed on a *validation subset* of the training data, where ω is the true class identity and $\tilde{\omega}$ - a prediction of the weak classifier. Then the Approximate Bayesian combination rule is derived as a confidence-weighted average:

$$P_a(\omega_i|x) = \sum_{k=1}^{K} w_k \underbrace{\sum_{j=1}^{J} P_k\left(\omega_i|\tilde{\omega}_j\right) P_k\left(\tilde{\omega}_j|x\right)}_{P_k(\omega_i|x)} \tag{2}$$

where $P_k(\tilde{\omega}|x)$ is the prediction of the individual classifier, and w_k is the weight of each classifier. The essence of equation 2 is that prediction of each classifier is weighted in accordance to the confidence that it has for each class.

An apparent difficulty with Equation 2 is choosing the value of w_k. This weight represents an output of an external "critic", which has been used in some earlier work [15] [14]. In the experiments of this paper we set it uniformly to $1/K$, as we are assuming the availability of pre-segmented data.

2.3. Boosting

Boosting is a popular ensemble method used for classification. In this paper we use the multi-class boosting algorithm of Freund and Schapire, [13]. In this algorithm the classification decision in the ensemble of weak classifiers is made on the basis of their weighted vote. During training weak classifiers are examined in turn with replacement. At every iteration a greedy selection is made - one classifier that gives a minimal error rate on the data, misclassified at the previous iteration is selected and its weight is updated as a function of its error rate. The process continues until the set number of iterations, T is reached, or no further improvement to the error rate can be made.

The resulting ensemble is a list of weights that is applied to the classifiers at the classification stage. With a slight change in the original notation, designating the selection pool of K classifiers as $\{f_k\}_K$, the resulting list of classifiers, H_B is typically a subset of the pool, where classifiers with vanishing weights are removed[1]:

$$H_B = Boost\left[\{f_k\}_K\right] \tag{3}$$

Formally, the posterior probability of i-th class can be expressed as the weighted sum of the votes. Using $[\![\cdot]\!]$ to designate an indicator function taking a value of 1 if the predicate argument holds and 0 otherwise:

$$P_b(\omega_i|x) \propto \sum_{k=1}^{K} W_k \left[\!\left[\arg\max P_k\left(\tilde{\omega}|x\right) = i\right]\!\right] \tag{4}$$

where the weight W_k is the aggregate weight of the k-th classifier, from the original list of T classifiers picked by $M1$:

$$W_k = \sum_{t=1}^{T} w_t \left[\!\left[f_t = f_k\right]\!\right] \tag{5}$$

Equation 5 states that the weight of the k-th classifier is the sum of weights of all instances where classifier f_k was picked by the algorithm.

[1] Compared to the original notation we aggregate the weights of the individual classifiers in the list. Unlike [13], we generate a list of *unique* classifiers. Note that this is only a notational change, made for the sake of coherency with the rest of the paper.

3. Weighted Ensemble Boosting

We present the Weighted Ensemble Boosting algorithm that is based on the combination of the ideas of forming classifier combinations by Bayesian averaging and boosting. The classifier pool for the boosting algorithm is augmented with all possible combinations of subsets of the weak classifiers. This is similar in spirit to identifying joint features in feature selection methods. We try all sub-combinations and let boosting algorithm select the ones that result in the maximum increase in accuracy. The combined classifier pool is composed of the original classifiers, $\{f_k\}_K$, and all sub-ensembles:

$$H_E = Boost\left[\bigcup\left(\{f_k\}_K, \left\{f_n^\beta\right\}_{N\times \mathrm{B}}\right)\right] \tag{6}$$

The combinations are formed as shown in equation 2 for all pairs, triples, etc. of the original classifiers. Additionally, a nonlinear transformation is applied to these sub-ensembles:

$$P_n^\beta(\omega_i|x) = \frac{\exp\left(\beta \sum\limits_{j\in S_n} P_j(\omega_i|x)\right)}{\sum\limits_{c=1}^{C}\left[\exp\left(\beta \sum\limits_{k\in S_n} P_k(\omega_c|x)\right)\right]} \tag{7}$$

where $P_k(\omega_j|x)$ is the "skewed" weak classifier, shown in Equation 2. S_n is the n-th classifier tuple. For an exhaustive enumeration of sub-ensembles, the total number of these tuples is given by the following relation:

$$N = \sum_{n=1}^{K}\binom{K}{n} = 2^K - 1 \tag{8}$$

In our experiments we use 6 classifiers with 8 different values of β, which results in $N = 63$ classifiers for every value of β, bringing the total size of the classifier set to 510.

4. Experiments

To evaluate the Weighted Ensemble Boosting we built several simple classifiers and combined their outputs as described in Section 2. Each classifier individually is perhaps not the best choice for identifying each of the events, but, just like in boosting algorithms, our objective is to derive a robust strong prediction from largely generic motion-based models.

4.1. Evaluation Data & Weak Classifiers

To perform the evaluation we created a data set of short video clips recorded in the office environment. The clips contain events from seven categories. The list of these events and the sizes of their training and testing data sets are given in table 1. In the setting of our experiments,

each classifier outputs a single decision for the entire video clip, rather than for every individual frame.

In our experiments we extract 6 types of features from the video stream - static as well as dynamic. For each feature we build an appropriate multi-class classifier. The list of features and the corresponding classifiers is given in table 2. These weak classifiers classify each video clip as containing a single example of an isolated activity, and are combined to form a strong classifier by the methods listed in Section 2.

	Training	Validation	Testing
Board	8	7	7
Meeting	8	14	7
Phone	7	6	6
Reading	10	7	7
Typing	8	8	6
Walk In	7	8	6
Walk Out	7	8	6

Tab. 1. Distribution of the number of sequences used in the training, validation and the testing data sets for the different event classes.

Feature	Classifer
Dominant Flow Histogram	KNN
Dominant Flow Dynamics	Discrete HMM
All Anchor Distances	Continuous HMM
Closest Anchor Distance	Continuous HMM
Blob Trajectory	Continuous HMM
Background Difference Energy	Gaussian

Tab. 2. Features and Classifiers used.

4.1.1. Dominant Flow Histogram

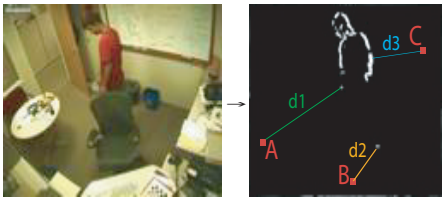
The scene flow classifier identifies sets of dominant motions in the scene and use them as features for discrimination of Scene activities. We compute the optical flow using Horn and Schunck algorithm [5]. For each frame we build a histogram of directions of the flow field, which is then used as a frame feature. We further quantize the set of flow histograms for a training data by estimating 3 prototypes, for each class with K-means algorithm. We denote k-th prototype of c-th class by H_k^c.

Each incoming frame is matched against these models and the closest matching class is selected as a winner for the frame. To produce the posterior distribution for the entire video clip the results for all frames in the test sequence are aggregated. Denoting the flow histogram of t-th frame by H_t, the vote of the classifier for t-th frame of the video is the class of the closest prototype:

$$L_t = \arg\min_c \left(\min_k \sqrt{(H_t - H_k^c)^2} \right) \tag{9}$$

Then the classifier output for the entire video sequence is the average vote for each class:

$$P(\tilde{\omega}|x) = \frac{1}{T} \sum_{t=1}^{T} [\![L_t = i]\!] \tag{10}$$



Input Frame Anchor Distances

Fig. 1. An illustration of the All-Anchor-Distance feature calculation. Left: anchor objects are selected by hand; right: distance to the closest point of the moving object is calculated for every anchor.

Tracked Blobs Aggregate Blob

Fig. 2. Illustration of the aggregate blob tracking. Individual blobs are found by the tracker. The blobs subsequently joined into the aggregate that we track.

4.1.2. Dominant Flow Dynamics

We model the flow dynamics of a video event with a discrete Hidden Markov Model. The alphabet of this model is computed by Vector Quantization of the flow histograms of the entire training video set with the K-means algorithm. For this paper we use an alphabet consisting of 20 prototypes.

A discrete 5-state HMM is trained for each event using the labeled training data where each input frame is represented by the index of the closest prototype flow of the alphabet. At the recognition phase, the likelihood of a sequence, $p(x|\tilde{\omega})$ is computed for each event model by the corresponding HMM and the posterior probability, $P(\tilde{\omega}|x)$ is calculated using Bayes rule.

4.1.3. All Anchor Distances

We define several anchor objects in the image and compute the distances from the closest point of the reference object to the closest point of the moving blob. The dynamics of these distances are computed with a continuous 5 state HMM (see an illustration in figure 1). The likelihood of a sequence is computed for each HMM and the result is converted to the posterior probability via the Bayes rule.

4.1.4. Closest Anchor Distance

It is expected that some behaviors are not tied to a particular object, but to any object. For instance, writing on the board would happen near a board, which requires a fixed reference object. In contrast, activity related to stealing a car can happen near any car in the parking lot. We try to model this type of activity by classifying dynamics of a moving blob with respect to the *closest* object (similar to [12]). Subsequently, the sequence likelihood is computed for each HMM and the Bayes rule is used to obtain the posterior probability.

4.1.5. Blob Trajectory

We also model the aggregate features of the moving objects in the frame. We do not consider motions of separate objects or object parts - instead we track the trajectory of the mean of all

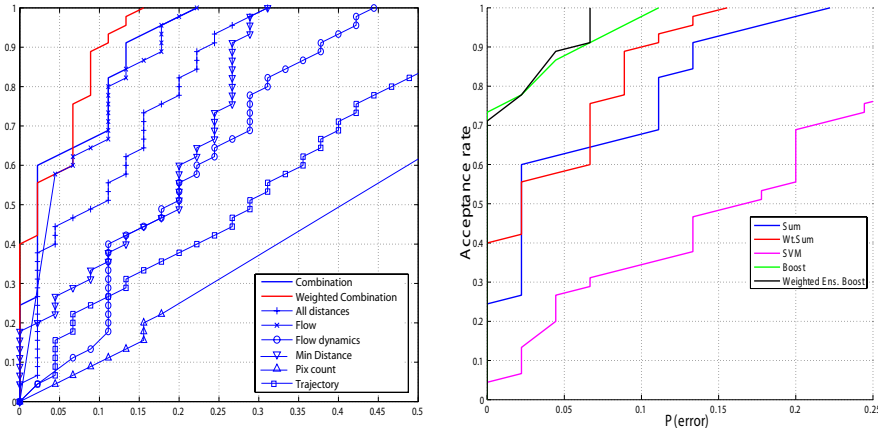


Fig. 3. a) Performance curves of the individual classifiers and their averaged combinations. b) Performance curves for various combination techniques compared with Weighted Ensemble Boosting.

moving pixels in the video. This allows us to use a stock tracker without much regard to its parameter settings and avoid estimation of the number of people in the scene and resolving of the related ambiguities.

The aggregate object is the union of all blobs that are found by a MeanShift tracking algorithm, [2]. We find the bounding box that covers the aggregate and extract the following features:

1. area of the aggregate bounding box;
2. the shortest distance of the aggregate bounding box to the anchor points;
3. x- and y- velocities;
4. aspect ratio of the bounding box.

We use a continuous 5-state HMM to estimate the feature dynamics. After the features are extracted, we train and test the classifier in the same way as described above.

4.1.6. Background Difference Energy

As a meeting involves multiple individuals, this classifier makes an implicit assumption that the more pixels are labeled as foreground, the more likely it is that a meeting is taking place. Using a set of training background images, we divide each image into a grid of 5x5 patches and fit a Gaussian Mixture Model to the chromatic content of the set of patches.

Using this background model we calculate the likelihood map for the training data containing 0, 1 and 2 people. Using the estimate of the number of people we classify a new frame as a meeting if number of people is estimated to be 2, *i.e.*:

$$P(\tilde{\omega}_i|x) = \begin{cases} P(N=2|x) & \text{, if } \tilde{\omega}_i = \text{“meeting”} \\ \left(1 - P(N=2|x)\right) / \left(K-1\right) & \text{, otherwise} \end{cases} \tag{11}$$

where N is the number of people and K the number of events.

4.2. Results

Our data set, collected in an office setting, contained from 19 to 29 video clips of each of the 7 actions listed in the table 1. Figure 3 shows the performance of each classifier as well as the two averaging combination techniques. Each curve is the plot of acceptance rate of a *multi-class* classifier against its error rate. It express the acceptable uncertainty of the classifier against its error rate, *i.e.*, how much data could one reject in order to achieve a guaranteed error rate.

Figure 3 shows that the worst classifier in the set performs just above chance, while the best one shows the error rate of about 77%. It can also be observed that the flow dynamics provides the best performance in the average sense, while the rest of the classifiers are approximately ordered as follows: `Average-Flow`, `All-Distances`, `Min-Distance`, `Flow-Dynamics`, `Motion-Trajectory`, `Pix-Count`, with the latter providing almost no information for classification of the recorded events.

However, error rates here indicate how frequently mistakes are made, regardless of which class they relate to. For instance, the pixel count classifier is fairly accurate in identifying meetings, but is not useful for any other event, which results in a very low overall accuracy. It is however frequently used by boosting algorithm in a combination with others as it identifies one event well.

Simple averaging of the classifier scores provides performance closely related to the best classifier in the set. Using the Approximate Bayesian Combination further increases the accuracy by teasing out the effects of each classifier on identification of each class individually. We applied all combination methods described in Section 2 to the office data set. The resulting performance curves are shown in figure 3.

The commonly used score averaging resulted in a slight improvement of performance at the lower error rates as compared to the best overall classifier in the set (see figure 3). This combination provides the baseline of 78% error rate for our further experiments.

The figure also shows the dismal performance of SVM as a "super-classifier" for the late fusion on our data set. For a set of 6 classifiers and 7 activities, the input space has 42 dimensions. Using cross-validation we found that RBF-kernel classifier shows the best, albeit poor performance. The main reason for that is the small size of our data set. In fact SVM over-fits the training data, providing practically no generalization. Reducing the dimensionality by PCA did not give significant improvement in performance.

Approximate Bayesian method improves the performance by 6% over the baseline to 84%. Boosting of the original set of 6 classifiers pushes the plank further to 89%. Finally, weighted ensemble boosting achieves the accuracy of 93%, which is a 15% improvement over the baseline (Table 4).

Figure 5 shows the "Leave-One-Out" (LOO) stability of the combination techniques used in the experiments. The LOO measure here quantifies the stability of the combined classifier with respect to change in the ensemble composition. The figure shows how much the chosen combination depends on the strongest classifier being present in the set. For this experiment classifier sets are formed by removing one classifier out of the ensemble at random. Note that

Combination	Accuracy
Sum	78%
Wt.Sum	84%
SVM	56%
Boost	89%
Wt.Ens.Boost	93%

Fig. 4. Summary of performance of combination methods.

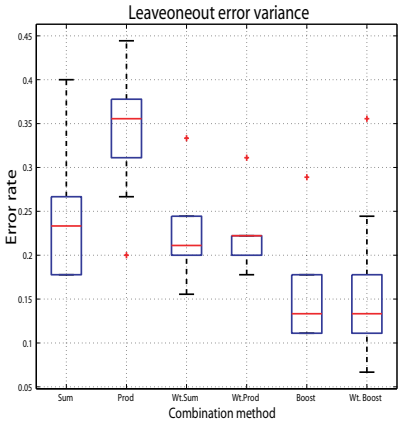


Fig. 5. Leave-One-Out combination stability. Each box signifies the variance of the strong classifier built from a reduced set of weak ones.

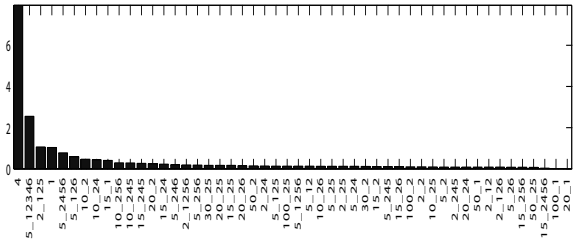


Fig. 6. Aggregated sub-ensemble weights. The labels along the y-axis encode the value of β in Equation7 and the indices of the weak classifiers used in the sub-ensemble.

in the case of the weighted ensemble boosting - the "root", classifier is taken out, which results in the reduction of the overall ensemble in size by $1 + |\mathrm{B}| \times 2^{\mathrm{K}-1}$. In our case of 6 classifiers and 8 values of β, the total number reduces from 510 to 253. The figure shows the distribution of errors as a result of 100 tests [2].

As is clear from figure 5 performance of boosting and Weighted Ensemble Boosting is the most stable of combinations we have tested. In addition to the methods discussed in the Section 2 we show the stability of the traditional product rule as well as a weighted product rule introduced in [14]. Such methods are sensitive to the degree of correlation between classifier and, as our experiments show, demonstrate high variance of the error estimate on our data.

Finally, figure 6 shows the sorted aggregated weights of the weighted sub-ensembles selected

[2] Even for fixed data split, the error rate will vary between the runs due to the tie-breaking randomization.

by the boosting algorithm. Each label on the y-axis encodes the type of the classifier selected by boosting. A single number stands for the original classifier (in the order shown in the legend of figure 3), while the rest of the labels have the two-part format. The first part indicates the value of β used in Equation 7, and the second part - a set of indices of classifiers used in the sub-ensemble, *e.g.*, "4" indicates the 4-th classifier from the list ("Min.Distance"), while "20_145" stands for a combination of classifiers 1, 4 and 5 with the value of $\beta = 20$. Interestingly, the classifier 4, that individually was rated only 3-rd best received the highest weight.

5. Conclusions and Future Work

In this work we introduced a novel method of performing classifier combination -Weighted Ensemble Boosting. We tested this method on a data set collected in an office environment and compared its performance to various combination methods. In our future work we plan to perform an extensive evaluation of the weighted ensemble boosting on problems with very large number of features and classes. We also plan on exploring the techniques that would allow us to avoid the exhaustive evaluation of all feature combinations for forming sub-ensembles.

References

[1] Ross A. and Jain A. Information fusion in biometrics. *Pattern Recognition Letters, Vol. 24, Issue 13, pp. 2115-2125*, 24:2115–2125, Sep 2003.

[2] Comaniciu D., Ramesh V., and Meer P. Real-time tracking of nonrigid objects using mean shift. *CVPR*, pages 673–678, 2000.

[3] Pekalska E., Duin R., and Skurichina M. A discussion on the classifier projection space for classifier combining. In *3rd International Workshop on Multiple Classifier Systems*, pages 137–148, Cagliari, Italy, 2002. Springer Verlag.

[4] Bobick A. F. Movement, activity, and action: The role of knowledge in the perception of motion. In *Philosophical Transactions Royal Society London B*. Royal Philosophical Soviety, 1997.

[5] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[6] Bilmes J. and Kirchhoff K. Directed graphical models of classifier combination: Application to phone recognition. In *Intl. Conference on Spoken Language Processing*, Beijing, China, 2000.

[7] Kittler J., Li Y., Matas J., and Sánchez R. Combining evidence in multimodal personal identity recognition systems. In *Intl . Conference on Audio- and Video-Based Biometric Authentication*, Crans Montana, Switzerland, 1997.

[8] Aggarwal J.K. and Cai Q. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428–440, 1999.

[9] Smith P., Lobo N., and Shah M. Temporal boost for event recognition. In *ICCV*, Beijing, China, 2005.

[10] Viola P. and Jones M. Robust real-time object recognition. In *ICCV*, Vancouver, Canada, 2001.

[11] Viola P., Jones M.J., and Snow D. Detecting pedestrians using patterns of motion and appearance. In *ICCV*, pages 734–741, 2003.

[12] Morris R. and Hogg D. C. StatisticalModels of object interaction. In *Workshop on Visual Surveillance*, Bombay, India, 1998. IEEE.

[13] Freund Y. and Schapire R. A decision-theoretic generalization of on-line learning and an application to boosting. *Computer and System Sciences*, 55(1):119–139, 1997.

[14] Ivanov Y. Multi-modal human identification system. In *Workshop on Applications of Computer Vision*, Breckenridge, CO, 2004.

[15] Ivanov Y., Heisele B., and Serre T. Using component features for face recognition. In *International Conference on Automatic Face ang Gesture Recognition*, Seoul, Korea, 2004.